پیشگفتار

این کتاب در *مورد تحلیل داده‌ها* با تمرکز خاص بر روی *تمرین مدل‌سازی پیش‌بینی* است. اصطلاح مدل‌سازی پیش‌بینی ممکن است ارتباط‌هایی مانند یادگیری ماشین، تشخیص الگو و داده‌کاوی را ایجاد کند. در واقع، این به‌عنوان ­انجمن‌ها مناسب هستند و روش‌های ذکر شده توسط این اصطلاحات بخشی جدایی ناپذیر از فرآیند مدل‌سازی پیش‌بینی است. اما مدل‌سازی پیش‌بینی بسیار فراتر از ابزارها و تکنیک‌های کشف ­الگوها در داده‌ها است. عمل مدل‌سازی پیش‌بینی، فرآیند توسعه یک مدل را به‌گونه‌ای تعریف می‌کند که بتوانیم دقت پیش‌بینی مدل را در داده‌های آینده که هنوز دیده نشده است، درک کرده و کمی کنیم. *کل* فرآیند تمرکز این کتاب است.

ما در نظر داریم که این کار راهنمای تمرین‌کننده‌ای برای فرآیند مدل‌سازی پیش‌بینی ­باشد و مکانی باشد که فرد بتواند در مورد این رویکرد بیاموزد و در مورد بسیاری از مدل‌های رایج و مدرن و قدرتمند شهودی به دست آورد. مجموعه‌ای از تکنیک‌های آماری و ریاضی مورد بحث قرار گرفته‌اند، اما انگیزه ما تقریباً در هر موردی این است که تکنیک‌ها را به گونه‌ای توصیف کنیم که به توسعه شهود برای نقاط قوت و ضعف آن‌ها به جای پیدایش ریاضی و زیربنای آن کمک کند. در بیشتر موارد ما از معادلات پیچیده اجتناب می‌کنیم، اگرچه چند استثناء ضروری وجود دارد. برای ­درمان‌های نظری بیشتر مدل‌سازی پیش‌بینی، پیشنهاد می‌کنیم [هستی و همکاران ( 2008](#bookmark1018) ) و [اسقف](#bookmark1011) [( 2006](#bookmark1011) ). برای این متن، خواننده باید اطلاعاتی در مورد آمارهای پایه، از جمله واریانس، همبستگی، رگرسیون خطی ساده و آزمون فرضیه‌های پایه (مثلاً *p* -values و آمار آزمون) داشته باشد.

فرآیند مدل‌سازی پیش‌بینی ذاتاً عملی است. اما در طول جستجوی مجدد ما ­برای این اثر، متوجه شدیم که بسیاری از مقالات و متون، خواننده را از بازتولید نتایج باز می‌دارد، یا به این دلیل که داده‌ها آزادانه ­در دسترس نبودند یا به این دلیل که نرم‌افزار در دسترس نبود یا فقط برای خرید در دسترس بود. [Buckheit و Donoho](#bookmark1012) [( 1995](#bookmark1012) ) نقد مرتبطی از حجاب علمی سنتی ارائه می‌دهد:

مقاله‌ای در مورد علوم محاسباتی در یک نشریه علمی، خود بورسیه نیست، بلکه صرفاً تبلیغ بورسیه است. بورس تحصیلی واقعی محیط توسعه نرم‌افزار کامل و ­مجموعه کامل دستورالعمل‌هایی است که ارقام را ایجاد می‌کند.

بنابراین، هدف ما این بود که تا حد امکان عملی باشیم و خوانندگان را قادر به بازتولید نتایج با دقت معقول کنیم و همچنین بتوانیم به‌طور طبیعی رویکرد مدل‌سازی پیش‌بینی را به داده‌های خود گسترش دهیم. علاوه بر این، ما از زبان R [(](#bookmark1024) Ihaka ­[and Gentleman 1996](#bookmark1019) ؛ R Development Core Team [2010](#bookmark1024) )، یک نرم‌افزار رایگان در دسترس برای محاسبات آماری و ریاضی، برای تمام مراحل فرآیند مدل‌سازی پیش‌بینی استفاده می‌کنیم. تقریباً تمام مجموعه داده‌های نمونه در بسته‌های R موجود هستند. بسته AppliedPredictiveModeling R شامل بسیاری از مجموعه داده‌های مورد استفاده در اینجا و همچنین اسکریپت‌های R برای بازتولید تحلیل‌های هر فصل است.

ما به چند دلیل R را به‌عنوان موتور محاسباتی این متن انتخاب کردیم. First R به صورت رایگان در دسترس است (اگرچه نسخه‌های تجاری آن وجود دارد) برای چندین ­سامانه عامل. دوم [،](#bookmark1016) تحت *مجوز عمومی عمومی منتشر شده است (* بنیاد نرم‌افزار آزاد [ژوئن 2007 ) که نحوه](#bookmark1016) توزیع مجدد برنامه را تشریح می‌کند. ­تحت این ساختار هر کسی آزاد است که کد منبع را بررسی و اصلاح کند. به دلیل ماهیت منبع باز، ده‌ها مدل پیش‌بینی قبلاً از طریق بسته‌های آزادانه در دسترس پیاده‌سازی شده اند. علاوه بر این، R دارای قابلیت‌های گسترده و قدرتمندی برای فرآیند مدل‌سازی کلی پیش‌بینی است. خوانندگانی که با R آشنایی ندارند می‌توانند آموزش‌های متعددی را به صورت آنلاین پیدا کنند. ما همچنین یک معرفی و راهنمای راه اندازی R را در پیوست ارائه می‌دهیم.

چند موضوع وجود دارد که ما زمان و/یا فضایی برای اضافه کردن نداشتیم، مهم‌ترین آنها: مدل‌های افزایشی تعمیم‌یافته، مجموعه‌ای از مدل‌های مختلف، مدل‌های شبکه، مدل‌های سری زمانی و چند مورد دیگر.

همچنین یک وب سایت برای کتاب وجود دارد:

[http://appliedpredictivemodeling. com/](http://appliedpredictivemodeling.com/)

که حاوی اطلاعات مربوطه خواهد بود.

این کار بدون کمک و ­شکنجه افراد بسیاری از جمله: والتر اچ. کارتر، جیم گرت، کریس جنینگز، پل هارمز، کریس کیفر، ویلیام کلینگر، دایجین کو، ریچ مور، دیوید نوهاسر، دیوید پاتر ممکن نبود. ، دیوید پاین، ویلیام راینز، آرنولد استرومبرگ و توماس ویدمار. ما همچنین می‌خواهیم از راس کوینلان برای کمکش در مورد Cubist و C5. 0 و بررسی توصیفات ما از این دو تشکر کنیم. در اسپرینگر، مایلیم از مارک اشتراوس و هانا براکن و همچنین داوران: وینی بوناتو، توماس میلر، راس کوینلان، اریک سیگل، استن یانگ و یک منتقد ناشناس تشکر کنیم. در آخر، مایلیم از خانواده هایمان ­برای حمایتشان تشکر کنیم: میراندا کوهن، استفان کوهن، بابی کوهن، رابرت کوهن، کارن کوهن و مری آن کوهن. وارن و کی جانسون؛ و والری و ترومن جانسون.

فصل 1

مقدمه

هر روز مردم با سؤالاتی مانند "امروز چه مسیری را برای رسیدن به محل کار باید طی کنم؟" مواجه می‌شوند. "آیا باید به یک شرکت مخابراتی تلفن همراه دیگر سوئیچ کنم؟" "چگونه باید پولم را سرمایه گذاری کنم؟" یا "آیا سرطان خواهم گرفت؟" این سؤالات نشان دهنده تمایل ما به دانستن رویدادهای آینده است و ما به‌طور جدی می‌خواهیم بهترین تصمیم را برای آن آینده بگیریم.

ما معمولا بر اساس اطلاعات تصمیم می‌گیریم. در برخی موارد ما داده‌های عینی و ملموس داریم، مانند ترافیک صبحگاهی یا گزارش آب و هوا. مواقع دیگر از شهود و تجربه استفاده می‌کنیم، مانند «امروز صبح باید از پل دوری کنم، زیرا معمولاً وقتی برف می‌بارد باتلاق می‌شود» یا «من باید آزمایش PSA بدهم زیرا پدرم سرطان پروستات دارد». در هر صورت، با توجه به اطلاعات و تجربه‌ای که در حال حاضر داریم، رویدادهای آینده را پیش‌بینی می‌کنیم و بر اساس آن پیش‌بینی‌کننده‌ها تصمیم می‌گیریم.

از آنجایی که اطلاعات از طریق اینترنت و رسانه‌ها به آسانی در دسترس است، تمایل ما برای استفاده از این اطلاعات برای کمک به تصمیم‌گیری ­تشدید شده است. و در حالی که مغز انسان می‌تواند به‌طور آگاهانه و ناخودآگاه حجم وسیعی از داده‌ها را جمع‌آوری کند، نمی‌تواند حجم بیشتری از اطلاعات مرتبط و قابل دستیابی را برای مسأله مورد نظر پردازش کند. برای کمک به فرآیندهای تصمیم‌گیری، اکنون به ابزارهایی مانند Google برای فیلتر کردن میلیاردها صفحه وب برای یافتن مناسب‌ترین اطلاعات برای جستجوهایمان، WebMD برای تشخیص بیماری‌هایمان بر اساس علائم و E\*TRADE برای غربالگری هزاران مورد روی می‌آوریم. سهام و شناسایی بهترین سرمایه گذاری برای پرتفوی ما.

این سایت‌ها و همچنین بسیاری دیگر، از ابزارهایی استفاده می‌کنند که اطلاعات فعلی ما را می‌گیرند، داده‌ها را به دنبال الگوهای مرتبط با مسأله ما می‌گردند و پاسخ‌ها را برمی‌گردانند. فرآیند توسعه این نوع ابزارها در تعدادی از زمینه‌ها مانند شیمی، علوم کامپیوتر، فیزیک و آمار تکامل یافته است و به آن‌ها «یادگیری ماشینی»، «هوش مصنوعی»، «تشخیص الگو»، «داده‌کاوی» گفته می‌شود. "تحلیل پیش‌بینی" و "کشف دانش". در حالی که هر زمینه با استفاده از دیدگاه‌ها ­و مجموعه ابزارهای مختلف به مسئله نزدیک می‌شود، هدف نهایی یکسان است: *ایجاد یک پیش‌بینی دقیق.* برای این کتاب، ما این اصطلاحات را در عبارت رایج *مدل‌سازی پیش‌بینی ترکیب می‌*کنیم.

گیسر [( 1993](#bookmark1016) ) مدل‌سازی پیش‌بینی را این‌گونه تعریف می‌کند: «فرآیندی که توسط آن یک مدل ایجاد یا انتخاب می‌شود تا سعی شود احتمال یک نتیجه را به بهترین شکل پیش‌بینی کند». ما این تعریف را کمی تغییر می‌دهیم:

مدل‌سازی پیش‌بینی: فرآیند توسعه یک ابزار یا مدل ریاضی که یک پیش‌بینی دقیق ایجاد می‌کند

استیو لوی از مجله *Wired* اخیراً درباره حضور فزاینده مدل‌های پیش‌بینی نوشت [( لوی 2010](#bookmark1020) )، «نمونه‌هایی [از هوش مصنوعی] را می‌توان در همه جا یافت: ماشین جهانی Google از هوش مصنوعی برای تفسیر پرسش‌های مرموز انسانی استفاده می‌کند. شرکت‌های کارت اعتباری از آن برای ردیابی کلاهبرداری استفاده می‌کنند. نتفلیکس از آن برای توصیه فیلم به مشترکین استفاده می‌کند. و سامانه مالی از آن برای مدیریت میلیاردها معامله استفاده می‌کند (فقط با سقوط گاه به گاه). نمونه‌هایی از انواع سوالاتی که می‌خواهید پیش‌بینی کنید عبارتند از:

این کتاب چند نسخه به فروش می‌رسد؟

آیا این مشتری کسب و کار خود را به شرکت دیگری منتقل می‌کند؟

خانه من در بازار فعلی چقدر به فروش می‌رسد؟

آیا فرد بیماری خاصی دارد؟

بر اساس انتخاب‌های گذشته، کدام فیلم‌ها این بیننده را مورد توجه قرار می‌دهند؟

آیا باید این سهام را بفروشم؟

چه افرادی را باید در سرویس دوستیابی آنلاین خود مطابقت دهیم؟

آیا ایمیل اسپم است؟

آیا این بیمار به این درمان پاسخ می‌دهد؟

شرکت‌های بیمه، به‌عنوان مثال دیگر، باید خطرات ­احتمالی دارندگان بیمه نامه خودرو، سلامت و زندگی را پیش‌بینی کنند. سپس از این اطلاعات برای تعیین اینکه آیا یک فرد بیمه نامه‌ای دریافت می‌کند یا خیر و اگر چنین است، با چه حق بیمه‌ای استفاده می‌شود. مانند شرکت‌های بیمه، دولت‌ها نیز به دنبال پیش‌بینی خطرات هستند، اما به منظور محافظت از شهروندان خود. نمونه‌های اخیر مدل‌های پیش‌بینی دولتی ­شامل مدل‌های بیومعیار برای شناسایی مظنونین به ترور، مدل‌های کشف تقلب [( وستفال 2008](#bookmark1027) [)](#bookmark1025) و مدل‌های ناآرامی و آشفتگی ( Shachtman [2011](#bookmark1025) ) است. حتی سفر به خواربارفروشی یا پمپ بنزین [مکان‌های روزمره که اطلاعات خرید ما در آن جمع‌آوری و تحلیل می‌شود تا بفهمیم چه کسی هستیم و چه می‌خواهیم [( دوهیگ 2012](#bookmark1015) )] ما را وارد دنیای مدل‌سازی پیش‌بینی می‌کند و ما اغلب حتی نمی‌دانند که ما آن را وارد کرده ایم. مدل‌های پیش‌بین اکنون در *وجود ما نفوذ کرده‌اند.*

در حالی که مدل‌های پیش‌بینی ما را به سمت محصولات رضایت‌بخش‌تر، درمان‌های متخصصی بهتر و سرمایه‌گذاری‌های سودآورتر راهنمایی می‌کنند، به‌طور مرتب پیش‌بینی‌کننده‌های نادرست ایجاد می‌کنند و پاسخ‌های اشتباه ارائه می‌کنند. به‌عنوان مثال، بسیاری از ما ایمیل مهمی را به دلیل یک مدل پیش‌بینی (معروف به فیلتر ایمیل) دریافت نکرده ایم که پیام را به اشتباه به‌عنوان هرزنامه شناسایی کرده است. به‌طور مشابه، ­مدل‌های پیش‌بینی (مثلاً مدل‌های تشخیصی متخصصی) بیماری‌ها را اشتباه تشخیص می‌دهند و مدل‌های پیش‌بینی (معروف به الگوریتم‌های مالی) به اشتباه سهامی را خرید و فروش می‌کنند که سود را پیش‌بینی می‌کند، در حالی که، در واقع، ضرر می‌یابد. این آخرین نمونه از ­مدل‌های پیش‌بینی که اشتباه پیش رفت، بسیاری از سرمایه‌گذاران را در سال 2010 تحت تأثیر قرار داد. کسانی که بازار سهام را دنبال می‌کنند احتمالاً با "سقوط ناگهانی" در 6 می‌2010 آشنا هستند که در آن بازار به سرعت بیش از 600 امتیاز را از دست داد، سپس بلافاصله دوباره به دست آورد. آن نقاط پس از ماه‌ها بررسی، کمیسیون معاملات آتی کالا و کمیسیون بورس و اوراق بهادار یک مدل الگوریتمی اشتباه را به‌عنوان علت سقوط شناسایی کردند [(](#bookmark1026) کمیسیون معاملات آتی کالای ایالات متحده و کمیسیون بورس و اوراق بهادار ایالات متحده [2010](#bookmark1026) ).

تا حدی ناشی از خرابی فلاش و سایر خرابی‌های مدل‌های پیش‌بینی، [رودریگز ( 2011](#bookmark1024) ) می‌نویسد، "مدل‌سازی پیش‌بینی، فرآیندی که توسط آن یک مدل ایجاد یا انتخاب می‌شود تا احتمال یک نتیجه را به بهترین شکل پیش‌بینی کند، اعتبار را به‌عنوان یک ابزار پیش‌بینی از دست داده است. " او فرض می‌کند که مدل‌های پیش‌بینی به‌طور منظم شکست می‌خورند زیرا متغیرهای پیچیده‌ای مانند رفتار انسان را در نظر نمی‌گیرند. در واقع، توانایی‌های ما برای پیش‌بینی یا تصمیم‌گیری توسط دانش حال و گذشته ما محدود شده و تحت تأثیر عواملی قرار می‌گیرد که در نظر نگرفته ایم. این واقعیت‌ها محدودیت‌های هر مدلی هستند، اما این واقعیت‌ها نباید ما را از تلاش برای بهبود روند خود و ساختن مدل‌های بهتر باز دارد.

تعدادی از دلایل رایج برای شکست مدل‌های پیش‌بینی وجود دارد و ما در فصل‌های بعدی به هر یک از آنها می‌پردازیم. مقصران رایج عبارتند از (1) پیش پردازش ناکافی داده‌ها، (2) اعتبار سنجی مدل ناکافی، (3) برون یابی ناموجه (به‌عنوان مثال، استفاده از مدل برای داده‌هایی که در فضایی قرار دارند که مدل هرگز ندیده است)، یا مهمتر از همه، (4) بیش برازش مدل با داده‌های موجود. علاوه بر این، مدل‌سازان پیش‌بینی اغلب تنها مدل‌های نسبتا کمی را هنگام جستجوی روابط پیش‌بینی بررسی می‌کنند. این معمولاً به دلیل ترجیح، دانش یا تخصص مدل‌سازان برای تنها چند مدل است یا فقدان نرم‌افزار موجود که آنها را قادر می‌سازد طیف وسیعی از تکنیک‌ها را کشف کنند.

با ارائه راهنمای گام به گام فرآیند ساخت مدل، به مدل‌سازان پیش‌بینی کمک کند تا مدل‌های قابل اعتماد و قابل اعتمادی را تولید کنند ­و دانش شهودی از طیف گسترده‌ای از مدل‌های رایج ارائه دهد. اهداف این کتاب عبارتند از:

* اصول اساسی برای ساخت مدل‌های پیش‌بینی
* توضیحات شهودی بسیاری از روش‌های مدل‌سازی پیش‌بینی رایج ­برای مسائل طبقه‌بندی و رگرسیون
* اصول و مراحل اعتبارسنجی یک مدل پیش‌بینی
* کد رایانه‌ای برای انجام کارهای اساسی لازم برای ساخت و اعتبارسنجی مدل‌های پیش‌بینی

برای نشان دادن این اصول و روش‌ها، از مجموعه متنوعی از مثال‌های واقعی از امور مالی گرفته تا داروسازی استفاده می‌کنیم که ­به تفصیل در بخش توضیح می‌دهیم.  [1. 4 .](#bookmark77) اما قبل از توصیف داده‌ها، ابتدا واقعیتی را بررسی می‌کنیم که با تکنیک‌های مدل‌سازی پیش‌بینی مواجه است: مبادله بین پیش‌بینی و تفسیر.

پیش‌بینی در مقابل تفسیر

برای مثال‌های ذکر شده در بالا، احتمالاً داده‌های تاریخی وجود دارد که می‌توان از آنها برای ایجاد یک ابزار ریاضی برای پیش‌بینی موارد نادیده آینده استفاده کرد. علاوه بر این، مهمترین هدف این مثالها درک این نیست که چرا چیزی رخ خواهد داد (یا نخواهد شد). در عوض، ما در درجه اول علاقه مند به پیش‌بینی دقیق ­شانس‌هایی هستیم که اتفاقی می‌افتد (یا نمی‌شود). توجه داشته باشید که تمرکز این نوع مدل‌سازی، بهینه‌سازی دقت پیش‌بینی است. به‌عنوان مثال، ما واقعاً برایمان مهم نیست که چرا یک فیلتر ایمیل فکر می‌کند یک پیام هرزنامه است. در عوض، ما فقط اهمیت می‌دهیم که فیلتر به‌طور دقیق هرزنامه‌ها را حذف کند و پیام‌هایی را که دوست داریم به صندوق پست ما منتقل کنند. به‌عنوان مثال دیگر، اگر من خانه‌ای می‌فروشم، علاقه اصلی من این نیست که چگونه یک وب سایت (مانند zillow. com ) ارزش آن را برآورد کرد. ­در عوض، من به شدت به آن علاقه دارم zillow. com به درستی قیمت خانه را تعیین کرده است. کاهش ارزش گذاری منجر به پیشنهادات کمتر و قیمت فروش کمتر خواهد شد. در عوض، ارزش گذاری بیش از حد ممکن است خریداران بالقوه را دور کند.

تنش بین پیش‌بینی و تفسیر نیز در زمینه متخصصی وجود دارد. به‌عنوان مثال، فرآیندی را در نظر بگیرید که یک بیمار سرطانی و متخصص هنگام فکر کردن به تغییر درمان‌های درمانی با آن مواجه می‌شوند. عوامل زیادی وجود دارد که متخصص و بیمار باید در نظر بگیرند مانند برنامه دوز، عوارض جانبی بالقوه و میزان بقا. با این حال، اگر بیماران به اندازه کافی از درمان جایگزین استفاده کرده باشند، می‌توان اطلاعات مربوط به بیماری، تاریخچه درمان و جمعیت شناسی را در مورد این بیماران جمع‌آوری کرد. همچنین، می‌توان آزمایش‌های آزمایشگاهی مربوط به زمینه ژنتیکی بیماران یا سایر داده‌های بیولوژیکی (مانند اندازه‌گیری پروتئین) را جمع‌آوری کرد. با توجه به نتیجه آنها، یک مدل پیش‌بینی می‌تواند برای پیش‌بینی پاسخ به درمان جایگزین بر اساس این داده‌ها ایجاد شود. سوال مهم برای متخصص و بیمار، پیش‌بینی *چگونگی* واکنش بیمار به تغییر درمان است. مهمتر از همه، این پیش‌بینی باید دقیق باشد. اگر مدلی برای انجام این پیش‌بینی ایجاد شود، نباید با الزام تفسیرپذیری محدود شود. می‌توان یک استدلال قوی ارائه داد که این غیراخلاقی است. تا زمانی که مدل می‌تواند به درستی اعتبار سنجی شود، مهم نیست که یک جعبه سیاه باشد یا یک مدل ساده و قابل تفسیر.

در حالی که علاقه اولیه مدل‌سازی پیش‌بینی تولید پیش‌بینی‌کننده‌های دقیق است، علاقه ثانویه ممکن است تفسیر مدل و ­درک دلیل کارکرد آن باشد. واقعیت تاسف بار این است که هرچه به سمت دقت بالاتر پیش می‌رویم، مدل‌ها پیچیده‌تر می‌شوند و تفسیرپذیری آنها دشوارتر می‌شود. زمانی که دقت پیش‌بینی هدف اصلی است، تقریباً همیشه این معامله‌ای است که انجام می‌دهیم.

اجزای کلیدی مدل‌های پیش‌بینی

مثال‌های محاوره‌ای تاکنون نشان داده‌اند که داده‌ها، در واقع مجموعه داده‌های بسیار بزرگ، اکنون می‌توانند به راحتی در تلاش برای پاسخگویی به تقریباً هر نوع سؤال تحقیقی تولید شوند. علاوه بر این، نرم‌افزار ساخت مدل رایگان یا نسبتا ارزان مانند JMP، WEKA و بسیاری از بسته‌ها در R و همچنین PCهای قدرتمند، شروع به توسعه مدل‌های پیش‌بینی را برای هر کسی که دانش محاسباتی دارد نسبتاً آسان می‌کند. اما همانطور که [رودریگز](#bookmark1024) [( 2011](#bookmark1024) ) به درستی اشاره می‌کند، اعتبار ساخت مدل ضعیف شده است، به ویژه با گسترش پنجره دسترسی به داده‌ها و ابزارهای تحلیل.

همانطور که در سراسر این متن خواهیم دید، اگر یک سیگنال پیش‌بینی در مجموعه‌ای از داده‌ها وجود داشته باشد، بسیاری از مدل‌ها بدون توجه به تکنیک یا مراقبت‌های انجام شده در توسعه مدل، درجاتی از آن سیگنال را پیدا می‌کنند. بنابراین، بکارگیری مدل ساده (نایو) می‌تواند تا حدی موثر باشد. به قول معروف "حتی یک سنجاب کور هم یک مهره پیدا می‌کند. " اما بهترین و پیش‌بینی‌ترین مدل‌ها اساساً تحت تأثیر یک مدل‌ساز با دانش متخصص و زمینه ­مسأله هستند. این دانش تخصصی ابتدا باید در به دست آوردن داده‌های *مرتبط* برای اهداف تحقیق مورد نظر به کار گرفته شود. در حالی که پایگاه‌های اطلاعاتی وسیعی از اطلاعات را می‌توان به‌عنوان بستری برای ساخت پیش‌بینی‌کننده‌ها استفاده کرد، اطلاعات نامربوط می‌تواند عملکرد پیش‌بینی بسیاری از مدل‌ها را کاهش دهد. لبه دانش موضوعی خاص می‌تواند به جداسازی اطلاعات بالقوه معنی دار از اطلاعات نامربوط، حذف نویزهای مضر و تقویت سیگنال اصلی کمک کند. سیگنال مخدوش کننده نامطلوب نیز ممکن است در داده‌ها وجود داشته باشد و بدون دانش متخصص قابل شناسایی نباشد. به‌عنوان یک مثال افراطی از سیگنال گمراه کننده و نیاز به درک متخصص از مسأله، پایگاه داده سامانه گزارش رویدادهای نامطلوب سازمان غذا و داروی ایالات متحده را در نظر بگیرید که اطلاعاتی در مورد میلیون‌ها مورد گزارش شده ­از داروها و عوارض جانبی گزارش شده آنها ارائه می‌دهد. بایاس‌های آشکار در این مجموعه داده‌ها فراوان است. به‌عنوان مثال، جستجو در یک دارو برای درمان حالت تهوع ممکن است ­نشان دهد که بخش بزرگی از بیمارانی که از این درمان استفاده می‌کنند مبتلا به سرطان خون هستند. تحلیل ناآگاهانه ممکن است لوسمی را به‌عنوان یک عارضه جانبی بالقوه دارو شناسایی کند. توضیح محتمل‌تر این است که آزمودنی‌ها داروی تهوع را برای کاهش عوارض جانبی درمان سرطان مصرف می‌کردند. این ممکن است به‌طور مستقیم واضح باشد، اما واضح است که در دسترس بودن مقادیر زیادی از سوابق، محافظتی در برابر استفاده ناآگاهانه از داده‌ها نیست.

آیرس ( 2007 ) به‌طور گسترده تعامل بین نظر متخصص و مدل‌های تجربی و داده‌محور را مورد مطالعه قرار می‌دهد، دو مشاهدات مهم را نشان می‌دهد که نیاز به دانش ویژه مسئله را تقویت می‌کند. اولا،

"در پایان، [مدل‌سازی پیش‌بینی] جایگزینی برای شهود نیست، بلکه یک مکمل است. "

به بیان ساده، نه مدل‌های مبتنی بر داده و نه متخصصی که صرفاً به ­شهود تکیه می‌کند، بهتر از ترکیبی از این دو نیست. ثانیاً

متخصص سنتی زمانی تصمیمات بهتری می‌گیرند که نتایج پیش‌بینی آماری در اختیار آنها قرار گیرد. کسانی که به اقتدار متخصصین سنتی می‌چسبند، تمایل دارند ایده ترکیب دو شکل "دانش" را با ارائه "پشتیبانی آماری" به متخصصان بپذیرند. *. .* انسان‌ها معمولاً ­زمانی پیش‌بینی‌کننده‌های بهتری انجام می‌دهند که نتایج پیش‌بینی آماری در اختیار آنها قرار گیرد. "

در برخی موارد، مانند تشخیص هرزنامه، ممکن است اجازه داده شود که رایانه‌ها بیشتر تفکر را انجام دهند، قابل قبول باشد. هنگامی که عواقب جدی تر است، مانند پیش‌بینی پاسخ بیمار، یک رویکرد ترکیبی اغلب به نتایج بهتری منجر می‌شود.

به‌طور خلاصه، شالوده یک مدل پیش‌بینی مؤثر با *شهود* و *دانش عمیق از زمینه مسأله که* برای تصمیم‌گیری در مورد توسعه مدل کاملاً حیاتی هستند، گذاشته شده است. ­این فرآیند با داده‌های *مرتبط،* یکی دیگر از عناصر کلیدی، آغاز می‌شود. سومین عنصر یک جعبه ابزار محاسباتی ­*همه کاره* است که شامل تکنیک‌هایی برای پیش پردازش داده‌ها و ­شبیه‌سازی و همچنین مجموعه‌ای از ابزارهای مدل‌سازی برای مدیریت طیف وسیعی از سناریوهای ممکن مانند مواردی است که در جدول توضیح داده شده است. 1. 1.

واژه شناسی

همانطور که قبلا ذکر شد، "مدل‌سازی پیش‌بینی" یکی از نام‌هایی است که به فرآیند کشف روابط در داده‌ها برای پیش‌بینی برخی از نتایج مطلوب اشاره دارد. از آنجایی که بسیاری از حوزه‌های علمی در این زمینه مشارکت داشته اند، مترادف‌هایی برای موجودیت‌های مختلف وجود دارد:

* اصطلاحات *نمونه (سمپل)،* *نقطه داده،* *مشاهده،* یا *نمونه (اینستانس)* به یک واحد مستقل داده مانند مشتری، بیمار یا ترکیب اشاره دارد. اصطلاح *نمونه (سمپل)*همچنین می‌تواند به زیر مجموعه‌ای از نقاط داده مانند نمونه مجموعه آموزشی اشاره داشته باشد. هنگام استفاده از این اصطلاح، متن زمینه مناسب را روشن می‌کند.
* مجموعه *آموزشی* شامل داده‌هایی است که برای توسعه مدل‌ها استفاده می‌شود، در حالی که مجموعه‌های *آزمون* یا *اعتبارسنجی* صرفاً برای ارزیابی عملکرد مجموعه نهایی مدل‌های کاندید استفاده می‌شوند.
* پیش‌بینی*‌*کننده‌ها، *متغیرهای مستقل،* *ویژگی‌ها* یا *توصیفگرها* داده‌هایی هستند که به‌عنوان ورودی معادله پیش‌بینی استفاده می‌شوند.
* *نتیجه،* *متغیر وابسته،* *هدف،* *کلاس* یا *پاسخ* به رویداد یا کمیتی نتیجه‌ای که پیش‌بینی می‌شود اشاره دارد.
* *دیتای پیوسته* دارای مقیاس‌های طبیعی و عددی هستند. فشار خون، هزینه یک آیتم یا تعداد حمام‌ها همگی پیوسته هستند. در آخرین مورد، شمارش‌ها نمی‌توانند یک عدد کسری باشند، اما همچنان به‌عنوان داده‌های پیوسته در نظر گرفته می‌شوند.
* *دیتای دسته‌بندی که* به‌عنوان داده‌های *اسمی،* *ویژگی* یا *گسسته شناخته می‌شوند،* مقادیر خاصی را به خود می‌گیرند که مقیاس ندارند. وضعیت اعتباری ("خوب" یا "بد") یا رنگ ("قرمز"، "آبی" و غیره) نمونه‌هایی از این داده‌ها هستند.
* *ساخت مدل،* *آموزش مدل* و *تخمین پارامتر* همگی به فرآیند استفاده از داده‌ها برای تعیین مقادیر معادلات مدل اشاره دارند.

**1-4- مجموعه داده‌های نمونه و سناریوهای داده‌های معمولی**

در فصل‌های بعدی، از مطالعات موردی برای نشان دادن تکنیک‌ها استفاده می‌شود. قبل از ­ادامه، ممکن است آموزنده باشد که به‌طور خلاصه چند نمونه از مسائل مدلسازی پیش‌بینی و انواع داده‌های مورد استفاده برای حل آنها را بررسی کنیم. تمرکز در اینجا بر تغییرات مسائل و همچنین ویژگی‌های داده‌های جمع‌آوری شده است. چندین مجموعه داده نمونه از رقابت‌های یادگیری ماشین سرچشمه می‌گیرند که مسائل دنیای واقعی را با انگیزه (اغلب پولی) برای ارائه بهترین راه‌حل فراهم می‌کنند. چنین مسابقاتی سابقه طولانی در مدل‌سازی پیش‌بینی دارند و این رشته را به شدت تحریک کرده اند.

**نوع موسیقی**

این مجموعه داده به‌عنوان یک مجموعه داده مسابقه در وب سایت [TunedIT ( http://tunedit. org/challenge/music-retrieval/genres )](http://tunedit.org/challenge/music-retrieval/genres) منتشر شد. در این مسابقه، هدف توسعه یک مدل پیش‌بینی برای دسته‌بندی موسیقی به شش دسته بود. در مجموع 12495 نمونه موسیقی وجود داشت که 191 ویژگی برای آنها تعیین شد. مقوله‌های پاسخ متعادل نبودند (شکل 2).  [1. 1 )](#bookmark77) ، با کوچکترین بخش متعلق به دسته فلزات سنگین (7٪) و بزرگترین از دسته کلاسیک (28٪). همه ­عوامل پیش‌بینی پیوسته بودند. بسیاری از آنها بسیار همبسته بودند و پیش‌بینی‌کننده‌ها مقیاس‌های مختلف اندازه‌گیری را در بر می‌گرفتند. این مجموعه داده‌ها با استفاده از 60 قطعه در هر سازنده ایجاد شد که ­از بین آنها 15-20 قطعه موسیقی برای هر اجراکننده انتخاب شد. سپس 20 بخش از هر قطعه به منظور ایجاد مجموعه داده نهایی پارامتری شد. از این رو، نمونه‌ها ذاتاً مستقل از یکدیگر نیستند.



شکل 1. 1: توزیع فراوانی ژانرها در داده‌های موسیقی

**درخواست‌های گرنت**

این مجموعه داده همچنین برای یک مسابقه در وب سایت [Kaggle ( http://www. kaggle. com )](http://www.kaggle.com) منتشر شد. برای این رقابت، هدف توسعه یک مدل پیش‌بینی برای احتمال موفقیت یک درخواست گرنت بود. پایگاه داده تاریخی شامل 8707 درخواست گرنت دانشگاه ملبورن ­از سال 2009 و 2010 با 249 پیش‌بینی بود. وضعیت اعطا (اعم از "عدم ­موفقیت" یا "موفق") پاسخ بود و نسبتاً متعادل بود (46٪ موفقیت آمیز). این وب سایت خاطرنشان می‌کند که نرخ موفقیت فعلی گرنت استرالیا کمتر از 25 درصد است. از این رو نرخ پایگاه داده تاریخی نماینده نرخ استرالیا نیست. پیش‌بینی‌کننده‌ها شامل اندازه‌گیری‌ها و مقوله‌هایی مانند شناسه حامی، دسته گرانت، محدوده ارزش اعطایی، میدان تحقیقاتی و ­بخش تقسیم می‌شوند و پیوسته، شمارش و دسته‌بندی بودند. یکی دیگر از ویژگی‌های قابل‌توجه ­این مجموعه داده این است که بسیاری از مقادیر پیش‌بینی (83%) وجود نداشتند. علاوه بر این، نمونه‌ها مستقل نبودند زیرا همان نویسندگان گرنت چندین بار در طول داده‌ها رخ داده‌اند. این داده‌ها در سراسر متن برای نشان دادن تکنیک‌های مختلف مدل‌سازی طبقه‌بندی استفاده می‌شوند.

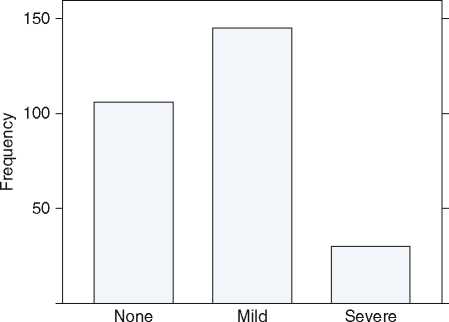
ما از این داده‌ها به‌طور گسترده در سرتاسر Chaps استفاده خواهیم کرد.  [12](#bookmark550) تا 15 و توضیح دقیق تر و خلاصه‌ای از داده‌ها را می‌توان در بخش یافت.  [12. 1 .](#bookmark550)

**آسیب کبدی**

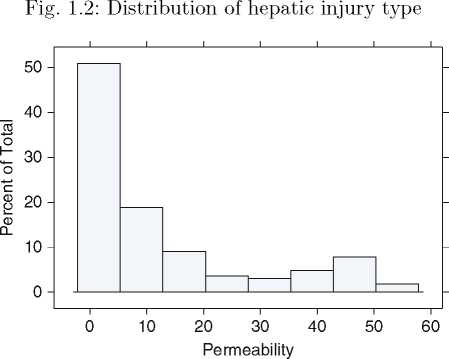
یک مجموعه داده از صنعت داروسازی برای توسعه مدلی برای پیش‌بینی احتمال آسیب کبدی (یعنی آسیب کبدی) ترکیبات مورد استفاده قرار گرفت. این مجموعه داده شامل 281 ترکیب منحصر به فرد بود. 376 پیش‌بینی برای هر کدام اندازه‌گیری یا محاسبه شد. پاسخ قطعی بود (یا "آسیب ایجاد نمی‌کند"، "آسیب خفیف" یا "آسیب شدید") و بسیار نامتعادل بود (شکل.  [1. 2 )](#bookmark84) . این تغییرات پاسخ اغلب در داده‌های دارویی رخ می‌دهد زیرا شرکت‌ها از ایجاد مولکول‌هایی که دارای ­ویژگی‌های ایمنی نامطلوب هستند دوری می‌کنند. بنابراین، تعداد مولکول‌های با رفتار خوب اغلب بسیار بیشتر از مولکول‌های نامطلوب است. پیش‌بینی‌کننده‌ها شامل اندازه‌گیری‌هایی از 184 صفحه بیولوژیکی و 192 پیش‌بینی ویژگی‌های شیمیایی بودند. پیش‌بینی‌کننده‌های بیولوژیکی ­فعالیت را برای هر صفحه نمایش نشان می‌دهند و مقادیری بین 0 تا 10 را با حالت 4 دریافت می‌کنند. پیش‌بینی‌کننده‌های ویژگی‌های شیمیایی تعداد زیرساخت‌های مهم و همچنین معیارهایی از خواص فیزیکی را نشان می‌دهند که تصور می‌شود با آسیب کبدی مرتبط هستند. شرح گسترده‌تری از این نوع پیش‌بینی‌کننده‌ها در فصل آورده شده است. 5.

**نفوذپذیری**

این مجموعه داده‌های دارویی برای توسعه مدلی برای پیش‌بینی ­نفوذپذیری ترکیبات استفاده شد. به‌طور خلاصه، نفوذپذیری معیار توانایی یک مولکول برای عبور از یک غشا است. به‌عنوان مثال، بدن دارای غشای قابل‌توجهی بین بدن و مغز است که به‌عنوان سد خونی-مغزی شناخته می‌شود.



شدت



شکل 1. 3: توزیع مقادیر نفوذپذیری

روده و بدن در روده ها. این غشاها به بدن کمک می‌کنند تا از نواحی حیاتی در مقابل دریافت مواد نامطلوب یا مضر محافظت کند. برای اینکه یک داروی خوراکی مصرف شده در مغز موثر باشد، ابتدا باید از دیواره روده عبور کند و سپس از سد خونی مغزی عبور کند تا برای هدف عصبی مورد نظر وجود داشته باشد. بنابراین، توانایی یک ترکیب برای نفوذ به غشاهای بیولوژیکی مربوطه برای درک در مراحل اولیه کشف دارو بسیار مهم است. ترکیباتی که به نظر می‌رسد برای یک بیماری خاص در آزمایش‌های غربالگری تحقیقاتی موثر هستند، اما به نظر می‌رسد نفوذپذیری ضعیفی دارند، ممکن است نیاز به تغییر داشته باشند تا نفوذپذیری و در نتیجه توانایی ترکیب برای رسیدن به هدف مورد نظر بهبود یابد. شناسایی مسائل نفوذپذیری می‌تواند به هدایت شیمیدانان به سمت مولکول‌های بهتر کمک کند.

سنجش‌های نفوذپذیری مانند PAMPA و Caco-2 برای کمک به اندازه‌گیری نفوذپذیری ترکیبات ایجاد شده‌اند [( Kansy et al. 1998](#bookmark1019) ). این صفحه‌ها در تعیین کمیت نفوذپذیری ترکیب مؤثر هستند، اما این روش ­کار پرهزینه‌ای دارد. با توجه به تعداد کافی ترکیباتی که غربال شده‌اند، می‌توانیم یک مدل پیش‌بینی برای نفوذپذیری در تلاش برای کاهش بالقوه نیاز به سنجش ایجاد کنیم. در این پروژه 165 ترکیب منحصر به فرد وجود داشت. برای هر کدام 1107 اثر انگشت مولکولی تعیین شد. اثر انگشت مولکولی یک دنباله دوتایی از اعداد است که وجود یا عدم وجود یک زیرساخت مولکولی خاص را نشان می‌دهد. پاسخ به شدت کج است (شکل 1).  [1. 3 )](#bookmark84) ، پیش‌بینی‌کننده‌ها پراکنده هستند (15. 5 ٪ *وجود* دارند) و بسیاری از پیش‌بینی‌کننده‌ها به شدت مرتبط هستند.

**فرآیند تولید مواد شیمیایی**

این مجموعه داده حاوی اطلاعاتی در مورد یک فرآیند تولید شیمیایی است که در آن هدف درک رابطه بین فرآیند و بازده محصول نهایی است. مواد اولیه در این فرآیند طی 27 مرحله طی می‌شود تا محصول نهایی دارویی ساخته شود. ماده ­اولیه از یک واحد بیولوژیکی تولید می‌شود و دارای طیف وسیعی از کیفیت و ویژگی است. هدف در این پروژه ایجاد مدلی برای پیش‌بینی درصد بازده فرآیند تولید بود. مجموعه داده‌ها شامل 177 نمونه از مواد بیولوژیکی بود که 57 ویژگی برای آنها اندازه‌گیری شد. از 57 ویژگی، 12 اندازه‌گیری از مواد اولیه بیولوژیکی ­و 45 اندازه‌گیری از فرآیند تولید وجود دارد. متغیرهای فرآیند شامل اندازه گیری‌هایی مانند دما، زمان خشک شدن، زمان شستشو و غلظت محصولات جانبی در مراحل مختلف بود. برخی از ­اندازه گیری‌های فرآیند را می‌توان کنترل کرد، در حالی که برخی دیگر مشاهده می‌شوند. پیش‌بینی‌کننده‌ها پیوسته، شمارش، مقوله‌ای هستند. برخی با هم مرتبط هستند و برخی حاوی مقادیر گمشده هستند. نمونه‌ها مستقل نیستند زیرا مجموعه‌هایی از نمونه‌ها از همان دسته مواد اولیه بیولوژیکی می‌آیند.

**صورتهای مالی تقلبی**

فانینگ و کوگر [( 1998](#bookmark1015) ) مجموعه داده‌ای را توصیف می‌کنند که برای پیش‌بینی تقلب مدیریت برای شرکت‌های سهامی عام استفاده می‌شود. با استفاده از منابع داده‌های عمومی، مانند اسناد کمیسیون بورس و اوراق بهادار ایالات متحده، نویسندگان توانستند 102 صورت مالی جعلی را شناسایی کنند. با توجه به اینکه درصد کمی از اظهارات تقلبی هستند، آنها نمونه‌ای معادل را انتخاب کردند. [[1]](#footnote-1) شرکت‌های غیر متقلب که برای کنترل عوامل مهم (مثلاً اندازه شرکت و نوع صنعت) نمونه‌برداری شدند. از این داده‌ها، 150 نقطه داده برای آموزش مدل‌ها و 54 نقطه مابقی برای ارزیابی آنها استفاده شد.

نویسندگان تحلیل را با تعداد نامشخصی از پیش‌بینی‌کننده‌های مشتق شده از حوزه‌های کلیدی، مانند نرخ گردش مالی، دعوی قضایی و ساختار بدهی آغاز کردند. در نهایت از 20 پیش‌بینی در مدل‌های خود استفاده کردند. به‌عنوان مثال می‌توان به نسبت حساب‌های دریافتنی به فروش، نسبت موجودی کالا به فروش و تغییرات در حاشیه ناخالص بین سال‌ها اشاره کرد. بسیاری از متغیرهای پیش‌بینی نسبت‌ها دارای مخرج مشترک هستند (مثلاً نسبت حساب‌های ­دریافتنی به فروش و نسبت موجودی کالا به فروش). اگرچه داده‌های واقعی منتشر نشدند، اما احتمالاً همبستگی قوی بین پیش‌بینی‌کننده‌ها وجود دارد.

از منظر مدل‌سازی، این مثال به چند دلیل جالب است. اول، به دلیل عدم تعادل کلاس بزرگ، فرکانس‌های دو کلاس در مجموعه داده‌ها با جمعیتی که با عدم تعادل شدید پیش‌بینی می‌شود بسیار متفاوت بود. این یک استراتژی رایج برای به حداقل رساندن عواقب ­چنین عدم تعادلی است و گاهی اوقات به‌عنوان "نمونه برداری پایین" از داده‌ها نامیده می‌شود. دوم، تعداد پیش‌بینی‌کننده‌های ممکن در مقایسه با تعداد نمونه‌ها زیاد بود. در این شرایط، انتخاب پیش‌بینی‌کننده‌ها برای مدل‌ها ظریف است، زیرا تنها تعداد کمی از نمونه‌ها برای انتخاب پیش‌بینی‌کننده‌ها، ساخت مدل‌ها و ارزیابی عملکرد آنها وجود دارد. فصل‌های بعدی مسأله بیش برازش را مورد بحث قرار می‌دهند، جایی که روند در داده‌های آموزشی در سایر نمونه‌های جامعه یافت نمی‌شود. با تعداد زیاد پیش‌بینی‌کننده‌ها و تعداد کمی از نقاط داده، این خطر وجود دارد که پیش‌بینی مربوطه موجود در این مجموعه داده قابل تکرار نباشد.

**مقایسه بین مجموعه داده ها**

این مثال‌ها ویژگی‌هایی را نشان می‌دهند که در اکثر مجموعه‌های داده مشترک است. اول، پاسخ ممکن است پیوسته یا مقوله‌ای باشد و برای پاسخ‌های طبقه‌بندی ­ممکن است بیش از دو دسته وجود داشته باشد. برای داده‌های پاسخ پیوسته، توزیع پاسخ ممکن است متقارن (مثلاً تولید مواد شیمیایی ­) یا اریب (مثلاً نفوذپذیری) باشد. برای داده‌های پاسخ طبقه‌ای، توزیع ممکن است متعادل باشد (مثلاً درخواست‌های گرنت) یا نامتعادل (مانند سبک موسیقی، آسیب کبدی). همانطور که در فصل نشان خواهیم داد.  [4 ،](#bookmark192) درک توزیع پاسخ برای یکی از اولین گام‌ها در فرآیند مدل‌سازی پیش‌بینی ضروری است: تقسیم داده‌ها به مجموعه‌های آموزشی و آزمایشی. درک توزیع پاسخ، مدل ساز را به سمت راه‌های بهتر پارتیشن‌بندی داده‌ها هدایت می‌کند. درک نکردن ویژگی پاسخ می‌تواند منجر به مسائل محاسباتی برای انواع خاصی از مدل‌ها شود و به تبع آن توانایی پیش‌بینی کمتر از حد مطلوب بوجود آید.

مجموعه داده‌های خلاصه‌شده در جدول 1. 1 همچنین ویژگی‌های پیش‌بینی‌کننده‌هایی را که برای اکثر مجموعه‌های داده همه گیر هستند، برجسته می‌کنند. به‌طور خاص، مقادیر پیش‌بینی‌کننده‌ها ممکن است پیوسته، شمارش و/یا مقوله‌ای باشند. ممکن است ­مقادیری از دست رفته داشته باشند و می‌توانند در مقیاس‌های مختلف اندازه‌گیری باشند. علاوه بر این، پیش‌بینی‌کننده‌های درون یک مجموعه داده ممکن است همبستگی یا ارتباط بالایی داشته باشند، بنابراین نشان می‌دهد که مجموعه پیش‌بینی‌کننده‌ها حاوی اطلاعات عددی اضافی است.

علاوه بر این، پیش‌بینی‌کننده‌ها ممکن است پراکنده باشند، به این معنی که اکثر نمونه‌ها حاوی اطلاعات یکسان هستند در حالی که فقط تعداد کمی حاوی اطلاعات منحصر به فرد هستند. مانند پاسخ، پیش‌بینی‌کننده‌ها می‌توانند از توزیع متقارن یا اریب (برای پیش‌بینی‌کننده‌های پیوسته) پیروی کنند یا متعادل یا نامتعادل باشند (برای پیش‌بینی‌کننده‌های طبقه‌بندی ­). در نهایت، پیش‌بینی‌کننده‌های درون یک مجموعه داده ممکن است با پاسخ ­رابطه داشته باشند یا نداشته باشند.

انواع مختلف مدل‌ها این نوع ویژگی‌های پیش‌بینی‌کننده را به روش‌های مختلفی مدیریت می‌کنند. به‌عنوان مثال، حداقل مربعات جزئی به‌طور طبیعی ­پیش‌بینی‌کننده‌های مرتبط را مدیریت می‌کند، اما اگر پیش‌بینی‌کننده‌ها در مقیاس‌های مشابه باشند، از نظر عددی پایدارتر است. از سوی دیگر، پارتیشن‌بندی بازگشتی تحت تأثیر پیش‌بینی‌کننده‌های مقیاس‌های مختلف قرار نمی‌گیرد، اما زمانی که پیش‌بینی‌کننده‌ها همبستگی دارند، ساختار پارتیشن‌بندی پایدار کمتری دارد. به‌عنوان مثال دیگری از تأثیر ویژگی‌های پیش‌بینی‌کننده بر مدل‌ها، رگرسیون خطی چندگانه نمی‌تواند اطلاعات پیش‌بینی از دست رفته را مدیریت کند، اما پارتیشن‌بندی بازگشتی می‌تواند زمانی که پیش‌بینی‌کننده‌ها حاوی مقادیر متوسطی از اطلاعات گمشده هستند استفاده شود. در هر یک از این سناریوها، عدم تنظیم مناسب پیش‌بینی‌کننده‌ها قبل از مدل‌سازی (معروف به پیش پردازش)، مدل‌هایی تولید می‌کند که ­عملکرد پیش‌بینی کمتر از حد مطلوب را دارند. ارزیابی ویژگی‌های پیش‌بینی‌کننده و پرداختن به آن‌ها از طریق پیش پردازش در فصل پوشش داده شده است.  [3 .](#bookmark131)

در نهایت، هر یک از این مجموعه داده‌ها یک ویژگی اساسی دیگر را نشان می‌دهد ­که باید هنگام ساخت یک مدل پیش‌بینی در نظر گرفته شود: رابطه ­بین تعداد نمونه ( *n* ) و تعداد پیش‌بینی‌کننده‌ها ( *P* ). در مورد مجموعه داده‌های ژانر موسیقی، تعداد نمونه‌ها ( *n* = 12496) بسیار بیشتر از تعداد پیش‌بینی‌کننده‌ها است ( *P* = 191). همه مدل‌های پیش‌بینی این سناریو را مدیریت می‌کنند، اما زمان محاسباتی در بین مدل‌ها متفاوت است و احتمالاً با افزایش تعداد نمونه‌ها و پیش‌بینی‌کننده‌ها افزایش می‌یابد. متناوبا، مجموعه داده‌های نفوذپذیری نمونه‌های کمتری ( 165 = ­*n ) نسبت به پیش‌بینییان (* *P* = 1,107) دارد. هنگامی که این اتفاق می‌افتد، مدل‌های پیش‌بینی مانند رگرسیون خطی چندگانه یا تحلیل متمایز خطی نمی‌توانند به‌طور مستقیم استفاده شوند. با این حال، برخی از مدل‌ها [به‌عنوان مثال، پارتیشن‌بندی بازگشتی و *K-* نزدیک‌ترین همسایگان ( *K* NN)] می‌توانند مستقیماً تحت این شرایط استفاده شوند. همانطور که در فصل‌های بعدی درباره هر روش بحث می‌کنیم، توانایی روش را برای مدیریت مجموعه‌های داده در جایی که *n<P.* برای کسانی که نمی‌توانند تحت این شرایط کار کنند، روش‌های مدل‌سازی جایگزین یا مراحل پیش‌پردازش را پیشنهاد می‌کنیم که به‌طور موثری بعد فضای پیش‌بینی‌کننده را کاهش می‌دهد.

به‌طور خلاصه، ما باید قبل از تلاش برای ساخت یک مدل، درک دقیقی از پیش‌بینی‌کننده‌ها و پاسخ برای هر مجموعه داده داشته باشیم. عدم درک می‌تواند منجر به مسائل محاسباتی و عملکرد کمتر از مدل بهینه شود. علاوه بر این، بیشتر مجموعه‌های داده به درجاتی از پیش پردازش نیاز دارند تا بتوان جهان مدل‌های پیش‌بینی ممکن را گسترش داد و عملکرد پیش‌بینی هر مدل را بهینه کرد.

| # Samples | 12,495 | 8,707 | 281 | 204 | 165 | 177 |
| --- | --- | --- | --- | --- | --- | --- |
| # Predictors | 191 | 249 | 376 | 20 | 1,107 | 57 |

Dimensions

1.4 Example Data Sets and Typical Data Scenarios

جدول 1. 1: مقایسه چندین ویژگی مجموعه داده‌های نمونه

مجموعه داده ها

Data Music Grant Hepatic Fraud Chemical

کاربردهای ژانر مشخصه تشخیص آسیب تولید نفوذپذیری

ویژگی‌های پاسخگویی

طبقه‌بندی یا پیوسته

متعادل / متقارن x x x

نامتعادل/کج x x x

مستقل x x

ویژگی‌های پیش‌بینی‌کننده

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| مداوم | ایکس | ایکس | ایکس | ایکس | ایکس |
| شمردن | ایکس | ایکس | ایکس |  | ایکس |
| دسته بندی |  | ایکس | ایکس | X X | ایکس |
| همبسته / مرتبط | ایکس | ایکس | ایکس | X X | ایکس |
| مقیاس‌های مختلف | ایکس | ایکس | ایکس | ایکس | ایکس |
| ارزش از دست رفته |  | ایکس |  |  | ایکس |
| پراکنده |  |  |  | ایکس |  |

بررسی اجمالی

این کتاب شامل چهار بخش است که خواننده را در فرآیند ساخت و ارزیابی انتقادی مدل‌های پیش‌بینی راهنمایی می‌کند. از آنجایی که اکثر خوانندگان به احتمال زیاد مایل به پیاده‌سازی این مفاهیم هستند، ما در پایان هر فصل یک بخش محاسباتی ارائه می‌دهیم که حاوی کدهایی برای پیاده‌سازی موضوعات پوشش داده ­شده در آن فصل است. در سراسر متن، ما بر زبان برنامه نویسی R تمرکز می‌کنیم [( تیم هسته توسعه R 2010](#bookmark1024) ). برای آن دسته از خوانندگانی که با R آشنایی ندارند، مقدمه‌ای از این نرم‌افزار را در پیوست B ارائه می‌کنیم. هدف ضمیمه B ارائه یک نمای کلی و مرجع سریع برای ساختارهای برنامه نویسی R اولیه است. با این حال، این متن ممکن است حاوی اطلاعات کافی برای کسانی که تازه با R هستند نباشد. در این مورد، بسیاری از آموزش‌های دیگر به صورت چاپی و تحت وب وجود دارد. برای آموزش اضافی، ما توصیه می‌­کنیم [ورزانی](#bookmark1027) [( 2002](#bookmark1027) )؛ [Venables و همکاران ( 2003](#bookmark1027) )؛ [Maindonald و Braun](#bookmark1021) [( 2007](#bookmark1021) )؛ [موئنشن](#bookmark1022) [( 2009](#bookmark1022) )، و [اسپکتور ( 2008](#bookmark1025) ).

در بخشاول[،](#bookmark99) روش‌هایی را برای ایجاد پایه‌های قوی توضیح می‌دهیم که می‌توان روی آن‌ها مدل‌ها را ساخت. مفاهیم سنگ بنای پیش پردازش داده‌ها (فصل.  [3 )](#bookmark131) و نمونه‌گیری مجدد (فصل.  [4 )](#bookmark192) باید قبل از مدل‌سازی هر داده‌ای به خوبی درک شود. فصل [3](#bookmark131) روش‌های متداول پیش ­پردازش مانند تبدیل داده‌ها، افزودن و/یا حذف متغیر و دسته کردن (باینینگ) متغیرهای پیوسته را توضیح می‌دهد. این فصل همچنین توضیح می‌دهد که چرا بیشتر مدل‌ها نیاز به پیش پردازش داده‌ها قبل از مدل‌سازی دارند. فصل [4](#bookmark192) ایده مخارج داده و روش‌های صرف داده را به منظور تنظیم مناسب یک مدل و ارزیابی عملکرد آن معرفی می‌کند. علاوه بر این، این فصل نشان می‌دهد که متخصص همیشه باید مجموعه‌ای متنوع از مدل‌ها را برای هر مسأله‌ای امتحان کند.

پس از پایه‌گذاری مدل‌سازی پیش‌بینی، تکنیک‌های رگرسیون سنتی و مدرن را در بخش بررسی می‌کنیم [II .](#bookmark259) این بخش از کتاب ­شامل راه‌هایی برای سنجش عملکرد هنگام مدل‌سازی یک نتیجه پیوسته است (فصل 5 ). فصل 6 درک و شهودی برای ­مدل‌های رگرسیون ارائه می‌کند که ساختار زیربنایی داده‌ها را با استفاده از ترکیب‌های خطی پیش‌بینی‌کننده‌ها جستجو می‌کنند. این مدل‌ها شامل رگرسیون خطی، حداقل مربعات جزئی و منظم‌سازی *L* 1 می‌باشد. متعاقبا (فصل [7 )](#bookmark342) ، ما توضیحی از مدل‌های رگرسیون ارائه می‌کنیم که مبتنی بر ترکیب‌های خطی ساده ­پیش‌بینی‌کننده‌ها نیستند که شامل شبکه‌های عصبی، خطوط رگرسیون تطبیقی چند متغیره (MARS)، ماشین‌های بردار پشتیبان (SVMs) و *K* NNs است. مدل‌های مبتنی بر درخت نیز به ترکیبات خطی پیش‌بینی‌کننده‌ها متکی نیستند. به دلیل محبوبیت آنها و به دلیل استفاده از آنها در **روش‌های گروهی**، ما فصل جداگانه‌ای را به این تکنیک‌ها اختصاص داده ایم (فصل 2).  [8 )](#bookmark388) . در این فصل مروری بر درختان رگرسیون، درختان کیسه‌ای، جنگل‌های تصادفی، تقویت (بوستینگ) و کوبیست ارائه می‌کنیم. قسمت را تمام می‌کنیم [II](#bookmark259) با یک مطالعه موردی (فصل 10 ) که تمام تکنیک‌های فوق را در مورد یک مسئله خاص مقایسه و مقایسه می‌کند: **مدل‌سازی مقاومت فشاری بتن برای به دست آوردن فرمولاسیون با خواص بهتر.**

مدل‌های طبقه‌بندی پیش‌بینی به‌طور طبیعی از توضیح مدل‌های رگرسیون پیروی می‌کنند و در قسمت پوشش داده شده اند [III](#bookmark489) . معیارهای عملکرد برای مسائل طبقه‌بندی متفاوت از رگرسیون است. ما این معیارها را در فصل تعریف می‌کنیم.  [11 .](#bookmark497) پیروی از ساختار مشابه در قسمت [دوم ،](#bookmark259) سپس یک ­درک و شهود کاربردی برای مدل‌های طبقه‌بندی ارائه می‌کنیم (فصل.  [12 )](#bookmark550) که بر اساس ترکیبات خطی پیش‌بینی‌کننده‌ها مانند ­تحلیل تفکیک (دیسکریمیننت) خطی، درجه دوم، منظم و حداقل مربعات جزئی است. ما همچنین روش‌های جریمه شده را برای طبقه‌بندی بررسی می‌کنیم. در فصل [13 ،](#bookmark612) روش‌های طبقه‌بندی را بررسی می‌کنیم که توابع بسیار غیرخطی پیش‌بینی‌کننده‌ها هستند. اینها شامل تحلیل تفکیک انعطاف پذیر، شبکه‌های عصبی، SVM‌ها، *K* NNs، Naive Bayes و نزدیکترین مرکزهای کوچک شده می‌باشد. موازی با قسمت [II ،](#bookmark259) روش‌های مبتنی بر درخت برای طبقه‌بندی در فصل مورد بررسی قرار گرفته‌اند.  [14 ،](#bookmark683) و یک مطالعه موردی طبقه‌بندی و مقایسه روش در فصل ارائه شده است.  [17](#bookmark795) .

این کار را در قسمت به پایان می‌بریم [IV](#bookmark817) با پرداختن به سایر ملاحظات مهم ­هنگام ساخت یک مدل یا ارزیابی عملکرد آن. در تلاش برای یافتن تنها مرتبط‌ترین پیش‌بینی‌کننده‌ها در یک مسئله، انواع مختلفی از روش‌های انتخاب ویژگی پیشنهاد شده‌اند. در حالی که این روش‌ها پتانسیل کشف اطلاعات معنی دار را دارند، اغلب به کاربر کمک می‌کنند تا نویز داده‌ها را به جای ساختار آن درک کند. فصل ­سوم [18](#bookmark822) روش‌های مختلفی را برای کمی کردن اهمیت پیش‌بینی‌کننده‌ها نشان می‌دهد در حالی که فصل.  [19](#bookmark864) مقدمه و راهنمای استفاده صحیح از ­تکنیک‌های انتخاب ویژگی را ارائه می‌دهد. همچنین، مجموعه‌ای از عناصر می‌توانند بر عملکرد مدل تأثیر بگذارند و کارورز را گمراه کنند که یا مدل عملکرد پیش‌بینی ضعیفی دارد (زمانی که واقعاً عملکرد خوبی دارد) یا مدل عملکرد پیش‌بینی خوبی دارد (زمانی که برعکس آن صادق است). برخی از ­عوامل متداول موثر بر عملکرد مدل عبارتند از نویز اضافی در مجموعه پیش‌بینی و/یا پاسخ و برون‌یابی پیش‌بینی. این موضوعات در فصل ارائه شده است.  [20 .](#bookmark918)

نشانه گذاری

یکی از اهداف این متن ارائه توضیحات شهودی از بسیاری از تکنیک‌ها است. در صورت امکان از کلمات به جای معادلات استفاده می‌شود. بسیاری از مدل‌ها را می‌توان با عبارات الگوریتمی توصیف کرد، اما برخی دیگر به توضیحات ریاضی بیشتری نیاز دارند. به‌طور کلی، کاراکترهای *x* و *y* در فونت‌ها و انواع حروف مختلف به ترتیب نشان‌دهنده پیش‌بینی‌کننده‌ها و پاسخ هستند. در اینجا اشکال خاص آنها در این متن آمده است:

*n* = تعداد نقاط داده

*P* = تعداد پیش‌بینی ها

*y i* = *i* امین مقدار مشاهده شده نتیجه، *i* =1. *. . n*

*y i* = برآیند پیش‌بینی‌شده از *i* مین نقطه داده، *i* = 1. *. . n y =* میانگین یا میانگین نمونه از *n* مقدار مشاهده‌شده نتیجه y = بردار تمام *n* مقدار نتیجه

*x ij* = مقدار *j* امین پیش‌بینی برای *i* امین نقطه داده، *i* = 1. *. . n* و

*j* = 1. *. . P*

*x j* = میانگین یا میانگین نمونه از *n* نقطه داده برای پیش‌بینی *j ام،*

*j* = 1. *. . P*

x *i* = مجموعه‌ای (یعنی بردار) از پیش‌بینی‌کننده‌های *P برای i* مین نقطه داده،

*i* = 1. *. . n*

X =amatrixof *P* پیش‌بینی برای تمام نقاط داده. این ماتریس دارای *n* ردیف و *P* ستون است

X *'* = جابجایی X ; این ماتریس دارای *P* ردیف و *n* ستون است

سایر رهنمودهای نمادگذاری مورد استفاده در معادلات در سراسر متن عبارتند از:

*C* = تعداد کلاس‌ها در یک نتیجه طبقه‌بندی شده

*C* = مقدار سطح کلاس *اول*

*p* = احتمال یک رویداد

*p e* = احتمال رخداد *اول*

*P r* [. ] = احتمال وقوع

*n*

= عملگر جمع بر روی شاخص *i*

*i* = 1

X = ماتریس کوواریانس نظری

*E* [ *•* ] = مقدار مورد انتظار *•*

*f (* *•* ) = تابعی از. ; *g (* *•* ) و *h (* *•* ) نیز توابع را در سراسر متن نشان می‌دهند

*P* = یک ضریب مدل ناشناخته یا نظری

*b* = یک ضریب مدل برآورد شده بر اساس نمونه‌ای از نقاط داده

خواننده باید توجه داشته باشد که ما از نمادهای دیگری در سراسر متن استفاده می‌کنیم که ممکن است منحصر به یک موضوع یا مدل خاص باشد. در این موارد ما آن نماد را به صورت محلی تعریف می‌کنیم.

قسمت اول

استراتژی‌های عمومی

فصل 2

تور کوتاهی از مدلسازی پیش‌بینی

روند

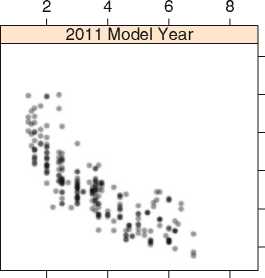
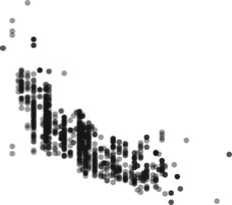
قبل از پرداختن به اجزای رسمی ساخت مدل، یک مثال ساده ارائه می‌کنیم که مفاهیم گسترده مدل‌سازی را نشان می‌دهد. به‌طور خاص، مثال زیر مفاهیم «خرج کردن» داده‌ها، ساخت مدل‌های کاندید و انتخاب مدل بهینه را نشان می‌دهد.

2. 1 مطالعه موردی: پیش‌بینی مصرف سوخت

تارنمای fueleconomy. gov که توسط اداره ­بهره وری انرژی و انرژی‌های تجدیدپذیر وزارت انرژی ایالات متحده و آژانس حفاظت از محیط زیست ایالات متحده اداره می‌شود، تخمین‌های مختلف مصرف سوخت را برای خودروهای سواری و کامیون‌ها فهرست می‌کند. برای هر وسیله نقلیه، مشخصات مختلفی مانند جابجایی موتور یا تعداد سیلندرها ثبت می‌شود. همراه با این مقادیر، ­اندازه گیری‌های آزمایشگاهی برای مقدار مایل طی شده در معابر (شهر و بزرگراه) در هر گالن (MPG) ماشین انجام می‌شود.

در عمل، ما مدلی را بر اساس مشخصات خودرو تا حد امکان می‌سازیم تا بتوانیم پیش‌بینی‌ترین مدل را پیدا کنیم. با این حال، این تصویر مقدماتی مفاهیم سطح بالای ساخت مدل را با استفاده از یک پیش‌بینی‌کننده، جابجایی موتور (حجم داخل سیلندرهای موتور) و یک پاسخ واحد MPG بزرگراه تنظیم‌نشده برای خودروهای مدل سال 2010-2011 متمرکز خواهد کرد.

اولین گام در هر فرآیند ساخت مدل، درک داده‌ها است که به راحتی می‌توان از طریق یک نمودار انجام داد. از آنجایی که ما فقط یک پیش‌بینی و یک پاسخ داریم، این داده‌ها را می‌توان با نمودار پراکندگی مشاهده کرد (شکل 1).  [2. 1 )](#bookmark108) . این شکل رابطه بین جابجایی موتور و مصرف سوخت را نشان می‌دهد. پانل "سال مدل 2010" شامل تمام داده‌های 2010 است در حالی که پانل دیگر داده‌ها را فقط برای خودروهای جدید 2011 نشان می‌دهد. واضح است که با افزایش جابجایی موتور، راندمان سوخت صرف نظر از سال کاهش می‌یابد. این رابطه تا حدودی خطی است اما مقداری انحنا به سمت انتهای محور جابجایی نشان می‌دهد.



**0 CL**

**s**

**o**

**<D**

**LU**

70

60

50

40

30

20

2010 Model Year

2468

جابجایی موتور

شکل 2. 1: رابطه بین جابجایی موتور و بازده سوخت تمام خودروهای مدل سال 2010 و خطوط خودروی جدید 2011

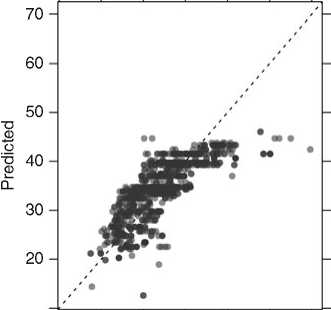
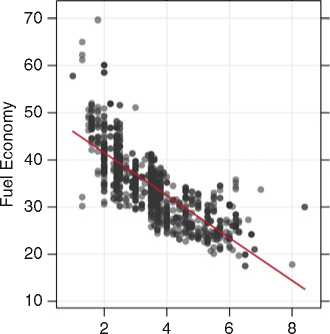
اگر بیش از یک پیش‌بینی‌کننده داشتیم، باید ویژگی‌های پیش‌بینی‌کننده‌ها و روابط بین پیش‌بینی‌کننده‌ها را بیشتر درک کنیم. این ویژگی‌ها ممکن است نشان دهنده مراحل مهم و ضروری پیش پردازشی باشد که باید قبل از ساخت یک مدل انجام شود (فصل [3 )](#bookmark131) .

پس از درک اولیه داده‌ها، مرحله بعدی ساخت و ارزیابی یک مدل بر روی داده‌ها است. یک رویکرد استاندارد این است که نمونه‌ای تصادفی از داده‌ها را برای ساخت مدل گرفته و از بقیه برای درک ­عملکرد مدل استفاده کنید. با این حال، فرض کنید می‌خواهیم MPG را برای یک خط خودروی *جدید پیش‌بینی کنیم.* در این شرایط می‌توان مدل‌هایی را با استفاده از داده‌های سال 2010 (شامل 1107 وسیله نقلیه) ایجاد کرد و روی 245 خودروی جدید 2011 آزمایش کرد. اصطلاح رایج ­این است که داده‌های 2010 به‌عنوان مدل "مجموعه آموزشی" استفاده می‌شود و مقادیر 2011 مجموعه "آزمون" یا "اعتبارسنجی" هستند.

اکنون که داده‌های مورد استفاده برای ساخت و ارزیابی مدل را تعریف کرده ایم، باید تصمیم بگیریم که چگونه عملکرد مدل را اندازه‌گیری کنیم. برای مسائل رگرسیونی که در آن سعی می‌کنیم یک مقدار عددی را پیش‌بینی کنیم، باقیمانده‌ها منابع مهم اطلاعات هستند. باقیمانده‌ها به‌عنوان مقدار مشاهده شده منهای مقدار پیش‌بینی شده (یعنی *y i - y i* ) محاسبه می‌شوند. هنگام پیش‌بینی مقادیر عددی، ریشه میانگین مربعات خطا (RMSE) معمولاً برای ارزیابی مدل‌ها استفاده می‌شود. در فصل با جزئیات بیشتر توضیح داده شده است.  [7 ،](#bookmark342) RMSE به این صورت تفسیر می‌شود که به‌طور متوسط، باقیمانده‌ها چقدر از صفر فاصله دارند.

در این مرحله، مدل ساز تکنیک‌های مختلفی را برای تعریف ریاضی رابطه بین پیش‌بینی و نتیجه امتحان می‌کند. برای این کار از مجموعه آموزشی برای تخمین مقادیر مختلف مورد نیاز معادلات مدل استفاده می‌شود­. مجموعه آزمایشی تنها زمانی استفاده می‌شود که چند مدل کاندید قوی نهایی شده باشند (استفاده مکرر از مجموعه آزمایشی در فرآیند ساخت مدل، کاربرد آن را به‌عنوان داور نهایی مدل‌ها نفی می‌کند).

فرض کنید یک مدل رگرسیون خطی ایجاد شده است که در آن MPG پیش‌بینی شده یک مدل شیب و عرض از مبدأ پایه است. با استفاده از داده‌های آموزشی، ما تخمین می‌زنیم



Engine Displacement

Fig. 2.2: Quality of fit diagnostics for the linear regression model. The training set data and its associated predictions are used to understand how well the model works

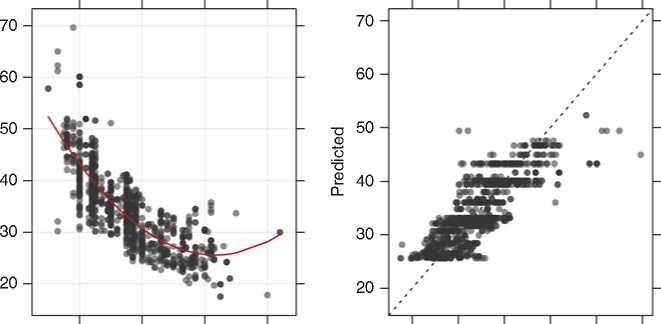
20 30 40 50 60 70

Observed

عرض از مبدأ به 50. 6 و شیب به *-* 4. 5 MPG / لیتر با استفاده از روش حداقل مربعات (بخش.  [6. 2 )](#bookmark286) . برازش مدل در شکل نشان داده شده است.  [2. 2](#bookmark0) برای داده‌های مجموعه آموزشی. [[2]](#footnote-2) پانل سمت چپ داده‌های مجموعه آموزشی را با یک مدل خطی برازش تعریف شده توسط شیب و عرض از مبدأ تخمینی نشان می‌دهد. پانل سمت راست MPG مشاهده شده و پیش‌بینی شده را ترسیم می‌کند. این نمودارها نشان می‌دهند که این مدل برخی از الگوهای موجود در داده‌ها را از دست می‌دهد، مانند پیش‌بینی کم بازده سوخت زمانی که جابجایی کمتر از 2 لیتر یا بالاتر از 6 لیتر است.

هنگام کار با مجموعه آموزشی، باید مراقب بود که به سادگی عملکرد مدل را با استفاده از داده‌های مشابه مورد استفاده برای ساخت مدل ارزیابی نکنید. اگر به سادگی داده‌های مجموعه آموزشی را مجدداً پیش‌بینی کنیم، پتانسیل ­تولید تخمین‌های بیش از حد خوش بینانه از نحوه عملکرد مدل وجود دارد، به خصوص اگر مدل بسیار سازگار باشد. یک رویکرد جایگزین برای تعیین کمیت عملکرد مدل، استفاده از *نمونه‌گیری مجدد است که* در آن از زیرنسخه مختلف مجموعه داده‌های آموزشی برای برازش با مدل استفاده می‌شود. تکنیک‌های نمونه‌گیری مجدد در فصل بحث شده است.  [4 .](#bookmark192) برای این داده‌ها، ما از شکلی از نمونه‌گیری مجدد به نام اعتبارسنجی متقاطع 10 برابری استفاده کردیم تا مدل RMSE را 4. 6 MPG تخمین بزنیم.

با نگاه کردن به شکل [2. 2](#bookmark0) ، می‌توان تصور کرد که مسأله ممکن است با وارد کردن مقداری غیرخطی در مدل حل شود. خیلی راه‌ها برای انجام دادن این وجود دارد. اساسی‌ترین رویکرد تکمیل ­مدل رگرسیون خطی قبلی با پیچیدگی بیشتر است. افزودن یک عبارت مربع برای ­جابجایی موتور به معنای تخمین پارامتر شیب اضافی مرتبط با مربع پیش‌بینی است. در انجام این کار، معادله مدل به



2468 20 30 40 50 60 70

جابجایی موتور مشاهده شده است

شکل 2. 3: کیفیت تشخیص برازش برای مدل رگرسیون درجه دوم (با استفاده از مجموعه آموزشی)

بازده = 63. 2-11. \_ *\_* 9 *x* جابجایی + 0. *94 x* جابجایی 2

این به‌عنوان یک *مدل درجه دوم نامیده می‌شود* زیرا شامل یک عبارت مربع است. برازش مدل در شکل نشان داده شده است.  [2. 3 .](#bookmark109) بدون شک، افزودن عبارت درجه دوم برازش مدل را بهبود می‌بخشد. در حال حاضر RMSE با استفاده از اعتبارسنجی متقاطع 4. 2 MPG برآورد شده است. یکی از مسائل مربوط به مدل‌های درجه دوم این است که آنها می‌توانند در ­حد فاصل پیش‌بینی عملکرد ضعیفی داشته باشند. در شکل [2. 3 ،](#bookmark109) ممکن است اشاره‌ای به این موضوع برای وسایل‌نقلیه با مقادیر جابجایی بسیار بالا وجود داشته باشد. به نظر می‌رسد مدل به‌طور غیر واقعی به سمت بالا خم می‌شود. پیش‌بینی وسایل‌نقلیه جدید با مقادیر جابجایی زیاد ممکن است نتایج قابل‌توجهی نادرست ایجاد کند.

فصل 6 [تا 8](#bookmark388) بسیاری از تکنیک‌های دیگر را برای ایجاد روابط پیچیده بین پیش‌بینی‌کننده‌ها و نتیجه مورد بحث قرار می‌دهد. یکی از این رویکردها، مدل اسپلاین رگرسیون تطبیقی چند متغیره (MARS) است [( فریدمن 1991](#bookmark1016) ). هنگامی که با یک پیش‌بینی منفرد استفاده می‌شود، MARS می‌تواند خطوط رگرسیون خطی جداگانه را برای محدوده‌های مختلف جابجایی موتور قرار دهد. شیب‌ها و عرض از مبدأ‌ها برای این مدل و همچنین تعداد و اندازه مناطق مجزا برای مدل‌های خطی تخمین زده شده است. برخلاف مدل‌های رگرسیون خطی، این ­تکنیک دارای یک *پارامتر تنظیم است* که نمی‌توان مستقیماً از داده‌ها تخمین زد. هیچ معادله تحلیلی وجود ندارد که بتوان از آن برای تعیین اینکه چند بخش برای مدل‌سازی داده‌ها استفاده شود، استفاده کرد. در حالی که مدل MARS دارای ­الگوریتم‌های داخلی برای انجام این تعیین است، کاربر می‌تواند مقادیر مختلف را امتحان کند و از نمونه‌گیری مجدد برای تعیین مقدار مناسب استفاده کند. هنگامی که مقدار پیدا شد، یک مدل نهایی MARS با استفاده از تمام داده‌های مجموعه آموزشی برازش می‌شود و برای پیش‌بینی استفاده می‌شود.

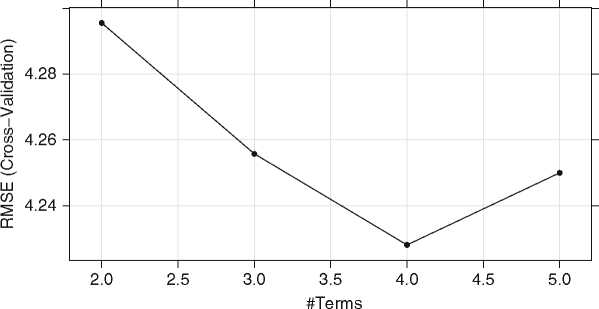
70

Fig. 2.4: The cross-validation profile for the MARS tuning parameter

60

50

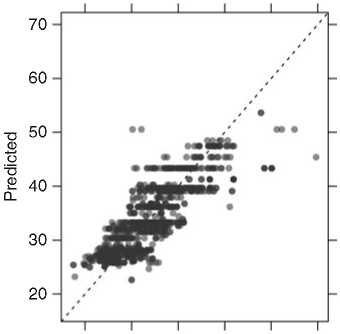
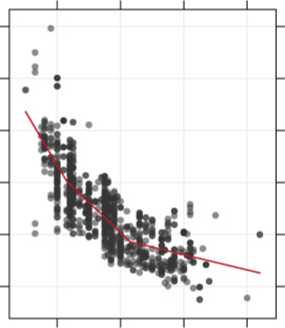
اوه

40

<D

30

20

کیفیت تشخیص مناسب برای مدل MARS (با استفاده از مجموعه آموزشی). مدل MARS چندین برازش رگرسیون خطی با نقاط تغییر در 2. 3، 3. 5 و 4. 3 L ایجاد می‌کند.

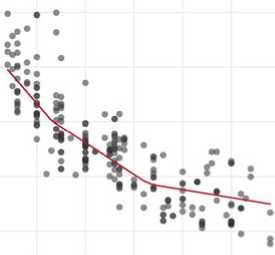
2468

Engine Displacement

20 30 40 50 60 70

Observed

برای یک پیش‌بینی، MARS می‌تواند تا پنج عبارت مدل (مشابه شیب‌ها و عرض از مبدأ‌های قبلی) را مجاز کند. با استفاده از اعتبارسنجی متقاطع، ما چهار مقدار کاندید را برای این پارامتر تنظیم ارزیابی کردیم تا نمایه نمونه‌گیری مجدد را ایجاد کنیم که در شکل نشان داده شده است.  [2. 4 .](#bookmark110) کمترین مقدار RMSE با چهار عبارت مرتبط است، اگرچه مقیاس تغییر در مقادیر RMSE نشان می‌دهد که مقداری حساسیت نسبت به این پارامتر تنظیم وجود دارد. RMSE مرتبط با مدل بهینه 4. 2 MPG بود. پس از برازش مدل نهایی MARS با چهار عبارت، برازش مجموعه آموزشی در شکل نشان داده شده است.  [2. 5](#bookmark110) که در آن چندین بخش خطی پیش‌بینی شده بود.



o  
LU

■ffl

60­

50­

40­

30­

20-

234567

234567

**J I I I I L**

Quadratic Regression

جابجایی موتور

داده‌های مجموعه تست و با مدل برای دو مدل مناسب است

بر اساس این سه مدل، مدل‌های رگرسیون درجه دوم و MARS بر روی مجموعه آزمون مورد ارزیابی قرار گرفتند. شکل [2. 6](#bookmark115) این نتایج را نشان می‌دهد. هر دو مدل بسیار شبیه هم هستند. مجموعه آزمون مقادیر RMSE برای مدل درجه دوم 4. 72 MPG و مدل MARS 4. 69 MPG بود. بر این اساس، هر یک از مدل‌ها برای پیش‌بینی خطوط جدید خودرو مناسب خواهند بود.

تم ها

چندین جنبه از فرآیند ساخت مدل وجود دارد که ارزش ­بحث بیشتر را دارد، به ویژه برای کسانی که تازه وارد مدل‌سازی پیش‌بینی شده اند.

تقسیم داده ها

اگرچه در فصل بعدی بحث شد، نحوه تخصیص داده‌ها به وظایف خاص (مثلاً ساخت مدل، ارزیابی عملکرد) جنبه مهم مدل‌سازی است. برای این مثال، علاقه اولیه پیش‌بینی مصرف سوخت وسایل‌نقلیه *جدید* است که جمعیت مشابهی با داده‌های مورد استفاده برای ساخت مدل نیست. این به این معنی است که تا حدودی، ما در حال آزمایش این هستیم که چگونه مدل به یک جمعیت متفاوت *تعمیم می‌یابد.* اگر ما علاقه مند به پیش‌بینی از روی همان جمعیت وسایل‌نقلیه (یعنی *درونیابی* ) بودیم، گرفتن یک نمونه تصادفی ساده از داده‌ها مناسب تر خواهد بود. نحوه تعیین مجموعه‌های آموزشی و آزمایشی باید نشان دهنده نحوه اعمال یا هدف مدل باشد.

چه مقدار داده باید به مجموعه‌های آموزشی و آزمایشی اختصاص داده شود؟ به‌طور کلی به موقعیت بستگی دارد. اگر مجموعه داده‌ها کوچک باشد، تصمیمات تقسیم داده‌ها می‌تواند حیاتی باشد. یک آزمون کوچک به‌عنوان قضاوت عملکرد، کاربرد محدودی دارد. در این مورد، تنها تکیه بر تکنیک‌های نمونه‌گیری مجدد (یعنی بدون مجموعه آزمایشی) ممکن است مؤثرتر باشد. مجموعه داده‌های بزرگ، بحرانی بودن این تصمیمات را کاهش می‌دهد.

داده‌های پیش‌بینی‌کننده

این مثال حول یکی از بسیار پیش‌بینی‌کننده‌ها می‌چرخد: ­جابجایی موتور. داده‌های اصلی شامل بسیاری از عوامل دیگر مانند تعداد سیلندرها، نوع انتقال و سازنده است. یک وسوسه جدی ­برای پیش‌بینی مصرف سوخت می‌تواند تا حد امکان پیش‌بینی‌کننده‌ها را برای بهبود عملکرد بررسی کند. با استفاده از پیش‌بینی‌کننده‌های بیشتر، این احتمال وجود دارد که RMSE برای خودروهای مدل جدید بیشتر کاهش یابد. برخی از تحقیقات ­در مورد داده‌ها نیز می‌تواند کمک کند. به‌عنوان مثال، هیچ یک از مدل‌ها در پیش‌بینی مصرف سوخت در زمانی که جابجایی موتور کم بود، مؤثر نبودند. گنجاندن پیش‌بینی‌کننده‌هایی که این نوع وسایل‌نقلیه را هدف قرار می‌دهند به بهبود عملکرد کمک می‌کند.

جنبه‌ای از مدل‌سازی که در اینجا مورد بحث قرار نگرفت، **انتخاب ویژگی** بود: فرآیند تعیین حداقل مجموعه پیش‌بینی‌کننده‌های مرتبطِ مورد نیاز مدل. این وظیفه رایج در فصل بحث شده است.  [19 .](#bookmark864)

**برآورد عملکرد**

قبل از استفاده از مجموعه آزمون، از دو تکنیک برای تعیین اثربخشی مدل استفاده شد. اول، ارزیابی‌های کمی آمار (یعنی RMSE) با استفاده از نمونه‌گیری مجدد به کاربر کمک می‌کند تا بفهمد هر تکنیک در داده‌های جدید چگونه عمل می‌کند. ابزار دیگر ایجاد تجسم‌های ساده از یک مدل، مانند ترسیم مقادیر مشاهده‌شده و پیش‌بینی‌شده، برای کشف مناطقی از داده‌ها بود که مدل به‌خصوص خوب یا بد است. این نوع اطلاعات کیفی برای بهبود مدل‌ها حیاتی است و زمانی که مدل فقط بر اساس آمار خلاصه سنجیده شود، فرصت این بهبودهای بالقوه از بین می‌رود.

ارزیابی چندین مدل

برای این داده‌ها، سه مدل مختلف مورد ارزیابی قرار گرفت. تجربه ما این است که برخی از متخصصان مدل‌سازی مدل مورد علاقه‌ای دارند که بی‌تفاوت بر روی آن تکیه می‌کنند. قضیه "بدون ناهار رایگان" [( Wolpert 1996](#bookmark1028) ) استدلال می‌کند که بدون داشتن اطلاعات اساسی در مورد مسأله مدل‌سازی، هیچ مدل واحدی وجود ندارد که همیشه بهتر از هر مدل دیگری عمل کند. به همین دلیل، می‌توان یک مورد قوی برای آزمایش طیف گسترده‌ای از تکنیک‌ها ایجاد کرد، سپس تعیین کرد که روی کدام مدل تمرکز کنید. در مثال ما، یک نمودار ساده از داده‌ها نشان می‌دهد که یک رابطه غیرخطی بین نتیجه و پیش‌بینی وجود دارد. با توجه به این دانش، ممکن است مدل‌های خطی را از بررسی حذف کنیم، اما هنوز تکنیک‌های متنوعی برای ارزیابی وجود دارد. شاید بتوان گفت که "مدل *X* همیشه بهترین مدل با عملکرد است" اما برای این داده‌ها، یک مدل درجه دوم ساده بسیار رقابتی است.

انتخاب مدل

در مرحله‌ای از فرآیند، یک مدل خاص باید انتخاب شود. این مثال دو نوع انتخاب مدل را نشان داد. ابتدا برخی از مدل‌ها را نسبت به مدل‌های دیگر انتخاب کردیم: مدل رگرسیون خطی مناسب نبود و حذف شد. در این مورد، ما *بین مدل‌ها را انتخاب کردیم.* نوع دوم انتخاب مدل نیز نشان داده شد. برای MARS، پارامتر تنظیم با استفاده از اعتبارسنجی متقابل انتخاب شد. این نیز انتخاب مدل بود که در آن ما در مورد *نوع مدل MARS* برای استفاده تصمیم گرفتیم. در این مورد، ما انتخاب را *در* مدل‌های مختلف MARS انجام دادیم.

در هر صورت، ما به اعتبار سنجی متقاطع و مجموعه تست برای تولید ارزیابی‌های کمی از مدل‌ها برای کمک به ما در انتخاب تکیه کردیم. از آنجایی که ما بر روی یک پیش‌بینی واحد تمرکز کردیم که اغلب این‌طور نیست، ما همچنین تجسم‌هایی از مدل مناسب برای کمک به اطلاع‌رسانی به ما انجام دادیم. در پایان فرآیند، به نظر می‌رسد که مدل‌های MARS و درجه دوم عملکردی معادل دارند. با این حال، با دانستن اینکه مدل درجه دوم ممکن است برای وسایل‌نقلیه با جابجایی بسیار زیاد خوب عمل نکند، شهود ما ممکن است به ما بگوید که مدل MARS را ترجیح دهیم. یکی از اهداف این کتاب کمک به کاربر برای به دست آوردن شهود در مورد نقاط قوت و ضعف مدل‌های مختلف برای تصمیم‌گیری آگاهانه است.

خلاصه

در ارزش اسمی، ساخت مدل ساده به نظر می‌رسد: یک ­تکنیک مدل‌سازی را انتخاب کنید، داده‌ها را متصل کنید و یک پیش‌بینی ایجاد کنید. در حالی که این رویکرد ­یک مدل پیش‌بینی ایجاد می‌کند، به احتمال زیاد *یک* مدل قابل اعتماد و قابل اعتماد برای پیش‌بینی نمونه‌های جدید ایجاد نخواهد کرد. برای بدست آوردن این نوع مدل، ابتدا باید داده‌ها *و* هدف مدل‌سازی را درک کنیم. پس از درک ­داده‌ها و اهداف، داده‌ها را پیش پردازش و تقسیم می‌کنیم. تنها پس از این مراحل، در نهایت به ساخت، ارزیابی و انتخاب مدل‌ها می‌پردازیم.

فصل 3

پیش پردازش داده ها

تکنیک‌های پیش پردازش داده‌ها عموماً به افزودن، حذف یا تبدیل داده‌های مجموعه آموزشی اشاره دارد. اگرچه این متن در درجه اول به تکنیک‌های مدل‌سازی مربوط می‌شود، آماده‌سازی داده‌ها می‌تواند توانایی پیش‌بینی یک مدل را ایجاد یا از بین ببرد. مدل‌های مختلف حساسیت‌های متفاوتی نسبت به نوع پیش‌بینی‌کننده‌های مدل دارند. *نحوه* ورود پیش‌بینی‌کننده‌ها به مدل نیز مهم است. تغییر شکل داده‌ها برای کاهش تأثیر چولگی یا پرت داده‌ها می‌تواند منجر به بهبود قابل‌توجهی در عملکرد شود. استخراج ویژگی، مورد ­بحث در بخش.  [3. 3](#bookmark150) ، یکی از تکنیک‌های تجربی برای ایجاد متغیرهای جایگزین است که ترکیبی از چندین پیش‌بینی کننده هستند. علاوه بر این، استراتژی‌های ساده‌تری مانند حذف پیش‌بینی‌کننده‌ها بر اساس کمبود محتوای اطلاعاتی نیز می‌تواند مؤثر باشد.

نیاز به پیش پردازش داده‌ها بر اساس نوع مدل مورد استفاده تعیین می‌شود. برخی از رویه‌ها، مانند مدل‌های مبتنی بر درخت، به ویژه نسبت به ویژگی‌های داده‌های پیش‌بینی حساس نیستند. دیگران، مانند رگرسیون خطی، حساس هستند. در این فصل، طیف گسترده‌ای از روش‌های *ممکن* مورد بحث قرار می‌گیرد. برای تکنیک‌های مدل‌سازی که در فصل‌های بعدی توضیح داده می‌شوند، همچنین بحث خواهیم کرد که در صورت وجود، تکنیک‌های پیش پردازش می‌توانند مفید باشند.

پردازش داده‌های ­*بدون نظارت* را تشریح می‌کند : متغیر خروجی توسط تکنیک‌های پیش پردازش در نظر گرفته نمی‌شود. در فصل‌های دیگر، روش‌های *نظارت* شده که در آن از نتیجه برای پیش پردازش داده‌ها استفاده می‌شود، نیز بحث شده است. برای مثال، مدل‌های حداقل مربعات جزئی (PLS) اساساً نسخه‌های نظارت شده تحلیل مؤلفه اصلی (PCA) هستند. ما همچنین استراتژی‌هایی را برای حذف پیش‌بینی‌کننده‌ها بدون در نظر گرفتن اینکه چگونه آن متغیرها ممکن است با نتیجه مرتبط باشند، توصیف می‌کنیم. فصل [19،](#bookmark864) تکنیک‌هایی را برای یافتن زیرمجموعه‌های پیش‌بینی‌کننده‌هایی که توانایی مدل را برای پیش‌بینی پاسخ بهینه می‌کنند، مورد بحث قرار می‌دهد. ­

*مهندسی ویژگی* نامیده می‌شود، می‌تواند تأثیر قابل‌توجهی بر عملکرد مدل داشته باشد. به‌عنوان مثال، گاهی اوقات استفاده از ترکیبی از پیش‌بینی‌کننده‌ها می‌تواند مؤثرتر از استفاده از مقادیر فردی باشد: نسبت دو پیش‌بینی ممکن است مؤثرتر از استفاده از دو پیش‌بینی مستقل باشد.

\_3،

© predictors. اغلب مؤثرترین رمزگذاری داده‌ها با درک مدل ساز از مسئله مشخص می‌شود و بنابراین از هیچ تکنیک ریاضی مشتق نمی‌شود.

معمولا چندین روش مختلف برای رمزگذاری داده‌های پیش‌بینی وجود دارد. به‌عنوان مثال، در Chaps.  [در 12](#bookmark550) تا 15، یک مجموعه داده گویا برای پیش‌بینی موفقیت گرنت تحصیلی توضیح داده شده است. یکی از اطلاعات موجود در داده‌ها، تاریخ ارسال گرنت است. این تاریخ را می‌توان به روش‌های بی شماری نشان داد:

تعداد روزهای پس از تاریخ مرجع

جداسازی ماه، سال و روز هفته به‌عنوان پیش‌بینی‌کننده‌های جداگانه

روز عددی سال (بدون توجه به سال تقویمی)

این که آیا تاریخ در سال تحصیلی بوده است (برخلاف جلسات تعطیلات یا تابستان)

مهندسی ویژگی "صحیح" به عوامل مختلفی بستگی دارد. اول، برخی از ­کدگذاری‌های en ممکن است برای برخی مدل‌ها بهینه و برای برخی دیگر ضعیف باشند. به‌عنوان مثال، مدل‌های مبتنی بر درخت داده‌ها را به دو یا چند bin تقسیم می‌کنند. از نظر تئوری، اگر ماه مهم بود، درخت روز عددی سال را بر این اساس تقسیم می‌کرد. همچنین، در برخی از مدل‌ها، رمزگذاری‌های متعدد از یک داده ممکن است مسائلی ایجاد کند. همانطور که چندین بار در فصل‌های بعدی نشان داده خواهد شد، برخی از مدل‌ها شامل *انتخاب ویژگی داخلی* هستند، به این معنی که مدل فقط شامل پیش‌بینی‌کننده‌هایی است که به حداکثر رساندن دقت کمک می‌کنند. در این موارد، مدل می‌تواند انتخاب کند که کدام نمایش داده‌ها بهترین است.

رابطه بین پیش‌بینی کننده و نتیجه عامل دوم است. به‌عنوان مثال، اگر یک جزء فصلی در این داده‌ها وجود داشته باشد و به نظر می‌رسد که وجود دارد، روز عددی سال بهترین خواهد بود. همچنین، اگر برخی از ماه‌ها درصد موفقیت بالاتری نسبت به ماه‌های دیگر نشان دادند، کدگذاری بر اساس ماه ترجیح داده می‌شود.

مانند بسیاری از سؤالات آماری، پاسخ به این سؤال که "کدام روش‌های مهندسی ویژگی بهترین هستند؟" این است که *بستگی دارد.* به‌طور خاص، به مدل مورد استفاده و رابطه واقعی با نتیجه بستگی دارد. بحث گسترده‌ای در مورد نحوه کدگذاری داده‌ها برای تحلیل‌های ما در بخش ارائه شده است.  [12. 1 .](#bookmark550)

قبل از پرداختن به تکنیک‌های خاص، یک مجموعه داده گویا که در سرتاسر فصل استفاده می‌شود، معرفی شده است.

مطالعه موردی: تقسیم‌بندی سلولی در غربالگری با محتوای بالا

محققان متخصصی اغلب به دنبال درک اثرات داروها یا بیماری‌ها بر اندازه، شکل، وضعیت رشد و تعداد سلول‌های موجود زنده یا گیاه هستند. برای انجام این کار، متخصصان می‌توانند سرم یا بافت مورد نظر را ­زیر میکروسکوپ بررسی کرده و به صورت دستی ویژگی‌های سلول مورد نظر را ارزیابی کنند.

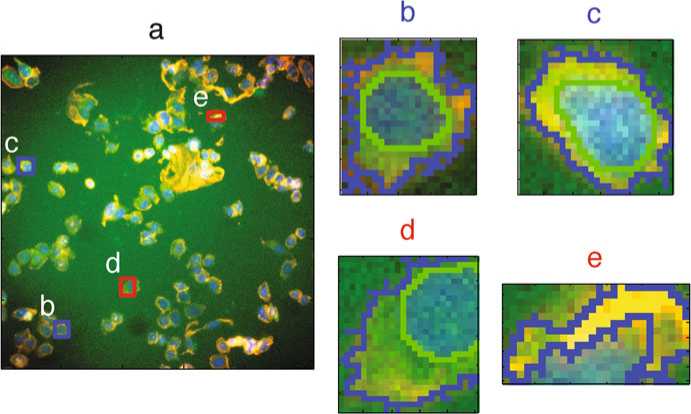
این کار خسته کننده است و نیاز به دانش تخصصی از نوع و ویژگی‌های سلول دارد.

راه دیگر برای اندازه‌گیری ویژگی‌های سلولی از این نوع نمونه‌ها استفاده از غربالگری با محتوای بالا است [( Guliano et al. 1997](#bookmark1017) ). به‌طور خلاصه، ابتدا یک نمونه با ماده‌ای رنگ می‌شود که به ویژگی مورد نظر سلول‌ها متصل می‌شود. برای مثال، اگر محققی بخواهد اندازه یا شکل هسته‌های سلولی را اندازه‌گیری کند، می‌توان روی نمونه‌ای که به DNA سلول می‌چسبد، لکه‌ای اعمال کرد. سلول‌ها را می‌توان در ماده‌ای ثابت کرد که حالت طبیعی سلول را حفظ کند. سپس نمونه توسط ابزاری (مانند میکروسکوپ کانفوکال) که در آن رنگ نور را منحرف می‌کند و آشکارسازها درجه پراکندگی را برای آن طول موج مشخص کمیت می‌کنند، مورد بازجویی قرار می‌گیرد. اگر چندین ویژگی سلول مورد نظر باشد، می‌توان از چندین رنگ و فرکانس‌های نور متعدد به‌طور همزمان استفاده کرد. سپس اندازه‌گیری‌های پراکندگی نور از طریق نرم‌افزار تصویربرداری پردازش می‌شوند تا ویژگی‌های سلولی مورد نظر را تعیین کنند.

استفاده از یک رویکرد خودکار و با توان بالا برای ارزیابی ویژگی‌های سلولی نمونه‌ها گاهی اوقات می‌تواند نتایج گمراه کننده‌ای ایجاد کند.  [هیل و همکاران ( 2007](#bookmark1018) ) یک پروژه تحقیقاتی را توصیف می‌کند که از غربالگری با محتوای بالا برای اندازه‌گیری ­چندین جنبه از سلول‌ها استفاده می‌کرد. آنها مشاهده کردند که نرم‌افزار تصویربرداری که برای ­تعیین مکان و شکل سلول استفاده می‌شود، در تقسیم‌بندی سلول‌ها (یعنی تعیین مرزهای سلول ها) مسأله دارد. شکل را در نظر بگیرید.  [3. 1 ،](#bookmark139) که چندین سلول نمونه از این مطالعه را به تصویر می‌کشد. در این تصاویر، مرزهای سبز روشن هسته سلول را مشخص می‌کنند، در حالی که مرزهای آبی محیط سلول را مشخص می‌کنند. واضح است که برخی از سلول‌ها به *خوبی تقسیم‌بندی شده اند،* در حالی که برخی دیگر اینگونه نیستند. به نظر می‌رسد سلول‌هایی که تقسیم‌بندی ضعیفی دارند آسیب دیده اند، در حالی که در واقعیت آسیب ندیده اند. اگر اندازه، شکل، و/یا کمیت سلول نقطه پایانی مورد علاقه در یک مطالعه باشد، مهم است که ابزار و نرم‌افزار تصویربرداری بتواند سلول‌ها را به درستی تقسیم‌بندی کند.

برای این تحقیق، [هیل و همکاران ( 2007](#bookmark1018) ) مجموعه داده‌ای متشکل از 2019 سلول را جمع‌آوری کرد. از این سلول‌ها، 1300 سلول ضعیف تقسیم‌بندی شدند (PS) و 719 به خوبی تقسیم شدند (WS). 1009 سلول برای مجموعه آموزشی رزرو شده بود. [[3]](#footnote-3)

برای نوع خاصی از سلول، محققان از لکه‌های مختلفی استفاده کردند که برای کانال‌های نوری مختلف قابل مشاهده بود. کانال یک با بدنه سلولی مرتبط بود و می‌توان از آن برای تعیین محیط سلول، مساحت و سایر کیفیت‌ها استفاده کرد. کانال دو با رنگ آمیزی DNA هسته ای، هسته سلول را مورد بازجویی قرار داد (در شکل 2 با سایه آبی نشان داده شده است.  [3. 1 )](#bookmark139) . کانال‌های سه و چهار به ترتیب برای تشخیص اکتین و توبولین رنگ آمیزی شدند. این دو نوع رشته هستند که سلول‌ها را در داربست‌ها عرضی می‌کنند و بخشی از اسکلت سلولی سلول هستند. برای همه سلول‌ها، 116 ویژگی (به‌عنوان مثال، مساحت سلول، تعداد فیبر نقطه ای) اندازه‌گیری شد و برای پیش‌بینی کیفیت تقسیم‌بندی سلول‌ها استفاده شد. [[4]](#footnote-4)



شکل 3. 1: تصویری که بخش‌بندی سلول را نشان می‌دهد [هیل و همکاران ( 2007](#bookmark1018) ). *کادرهای قرمز* [ پانل‌های ( d ) و ( e )] سلول‌های تقسیم‌بندی ضعیفی را نشان می‌دهند در حالی که سلول‌های *جعبه‌های آبی* نمونه‌هایی از تقسیم‌بندی مناسب هستند.

این فصل از نمونه‌های مجموعه آموزشی شناسایی شده توسط نویسندگان اصلی برای نشان دادن تکنیک‌های پیش پردازش داده‌ها استفاده می‌کند.

تبدیل داده‌ها برای پیش‌بینی‌کننده‌های فردی

تبدیل متغیرهای پیش‌بینی ممکن است به دلایل مختلفی مورد نیاز باشد. برخی از تکنیک‌های مدل‌سازی ممکن است الزامات سخت‌گیری داشته باشند، مانند پیش‌بینی‌کننده‌ها ­که مقیاس مشترکی دارند. در موارد دیگر، ایجاد یک مدل خوب ممکن است به دلیل ویژگی‌های خاص داده‌ها (به‌عنوان مثال، پرت) دشوار باشد. در اینجا ما در مورد تبدیل مرکزی، مقیاس‌بندی و چولگی بحث می‌کنیم.

مرکز و مقیاس بندی

ساده‌ترین و رایج‌ترین تبدیل داده، مقیاس مرکزی متغیرهای پیش‌بینی است. برای قرار دادن یک متغیر پیش‌بینی، میانگین مقدار پیش‌بینی از همه مقادیر کم می‌شود. در نتیجه مرکز سازی، پیش‌بینی میانگین صفر دارد. به‌طور مشابه، برای مقیاس‌بندی داده‌ها، هر مقدار از متغیر پیش‌بینی بر انحراف استاندارد آن تقسیم می‌شود. مقیاس‌گذاری داده‌ها، مقادیر را مجبور می‌کند تا انحراف استاندارد مشترک یک را داشته باشند. این دستکاری‌ها هستند

به‌طور کلی برای بهبود ثبات عددی برخی از محاسبات استفاده می‌شود. برخی از مدل‌ها مانند PLS (Sects.  [6. 3](#bookmark8) و [12. 4 )](#bookmark572) ، از اینکه پیش‌بینی‌کننده‌ها در مقیاس مشترک قرار دارند، بهره مند شوید. تنها نقطه ضعف واقعی این تبدیل‌ها از دست دادن قابلیت تفسیر مقادیر فردی است زیرا داده‌ها دیگر در واحدهای اصلی نیستند.

تبدیل متغیریی برای رفع کجی

یکی دیگر از دلایل رایج برای تبدیل، حذف چولگی توزیعی ­است. توزیع غیر اریب توزیعی است که تقریباً متقارن است. این بدان معنی است که احتمال سقوط در هر دو طرف میانگین توزیع تقریباً برابر است. یک توزیع دارای انحراف راست تعداد زیادی نقطه در سمت چپ توزیع (مقادیر کوچکتر) نسبت به سمت راست (مقادیر بزرگتر) دارد. برای مثال، داده‌های تقسیم‌بندی سلولی حاوی یک پیش‌بینی است که انحراف استاندارد شدت پیکسل‌ها را در رشته‌های اکتین اندازه‌گیری می‌کند. در واحدهای طبیعی، داده‌ها چولگی سمت راست قوی را نشان می‌دهند. غلظت بیشتری از نقاط داده در مقادیر نسبتاً کوچک و تعداد کمی از مقادیر بزرگ وجود دارد. شکل [3. 2](#bookmark2) هیستوگرام داده‌ها را در واحدهای طبیعی (پانل سمت چپ) نشان می‌دهد.

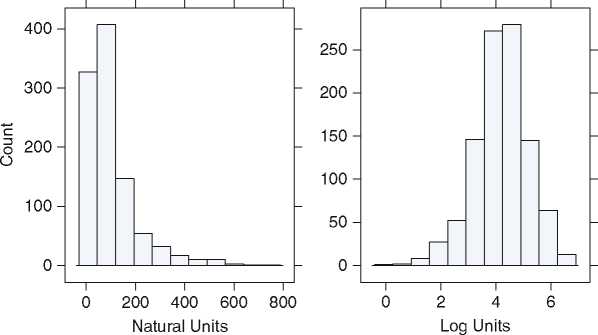
یک قانون کلی که باید در نظر گرفته شود این است که داده‌های اریب که نسبت بالاترین مقدار به کمترین مقدار بیشتر از 20 است دارای چولگی قابل‌توجهی هستند. همچنین می‌توان از آمار چولگی به‌عنوان یک تشخیص استفاده کرد. اگر توزیع پیش‌بینی تقریباً متقارن باشد، مقادیر چولگی نزدیک به صفر خواهد بود. همانطور که توزیع به سمت راست تر می‌شود، آمار چولگی بزرگتر می‌شود. به‌طور مشابه، هرچه توزیع به سمت چپ بیشتر شود، مقدار منفی می‌شود. فرمول آمار چولگی نمونه به این صورت است که *x* متغیر پیش‌بینی، *n* تعداد مقادیر و *x* میانگین نمونه پیش‌بینی است. برای داده‌های رشته اکتین نشان داده شده در شکل.  [3. 2 ،](#bookmark2) آمار چولگی 2. 39 محاسبه شد در حالی که نسبت به بزرگترین و کوچکترین مقدار 870 بود.

skewness =

where *v* =

E(*Xi - x*)3(*n -* 1) *v 3 / 3*E( *Xi — x* )2*(n —* 1)

جایگزین کردن داده‌ها با log، ریشه مربع یا معکوس ممکن است به حذف انحراف کمک کند. برای داده‌های شکل.  [3. 2 ،](#bookmark2) پانل سمت راست توزیع داده‌ها را پس از اعمال تبدیل log نشان می‌دهد. پس از تبدیل، توزیع کاملاً متقارن نیست، اما این داده‌ها بهتر از زمانی که در واحدهای طبیعی بودند رفتار می‌کنند.



شکل 3. 2: *سمت چپ* : هیستوگرام انحراف معیار شدت پیکسل‌ها در رشته‌های اکتین. این پیش‌بینی دارای چولگی سمت راست قوی با غلظت نقاط با مقادیر کم است. برای این متغیر، نسبت کوچکترین به بزرگترین مقدار 870 و مقدار چولگی 2. 39 است. *راست* : همان داده‌ها پس از تبدیل گزارش. مقدار چولگی برای داده‌های ثبت شده *-* 0 *بود.* 4

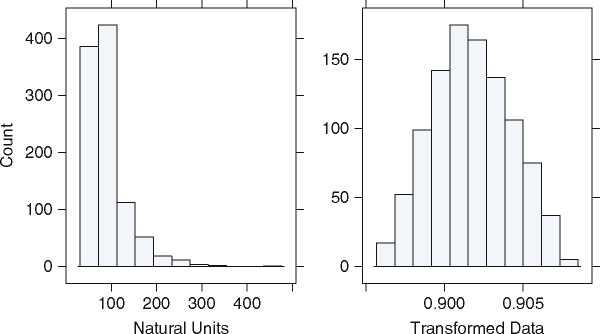
روش دیگر، روش‌های آماری را می‌توان برای شناسایی تجربی یک تبدیل مناسب مورد استفاده قرار داد.  [باکس و کاکس](#bookmark1012) [( 1964](#bookmark1012) ) *خانواده‌ای* از تبدیل متغیر را پیشنهاد می‌کند[[5]](#footnote-5) که با یک پارامتر نمایه می‌شوند که به صورت *A مشخص می‌شود* :

*x* = / ^1 اگر *A* = 0

log( *x* ) اگر *A* =0 باشد

علاوه بر تبدیل log، این خانواده می‌تواند ­شکل ترانس مربع ( *A* = 2)، ریشه مربع ( *A* = 0. 5 *)،* معکوس ( *A* = *-* 1) و موارد دیگر را در بین آنها شناسایی کند. با استفاده از داده‌های آموزشی می‌توان *A* را تخمین زد.  [باکس و کاکس](#bookmark1012) [( 1964](#bookmark1012) ) نشان می‌دهد که چگونه می‌توان از تخمین حداکثر درستنمایی برای تعیین ­پارامتر تبدیل استفاده کرد. این روش به‌طور مستقل برای هر داده پیش‌بینی که حاوی مقادیر بزرگتر از صفر است اعمال می‌شود.

برای داده‌های تقسیم‌بندی، 69 پیش‌بینی به دلیل مقادیر صفر یا منفی تبدیل نشدند و 3 پیش‌بینی تخمین *A در* 0 *± 1 داشتند.* 02، بنابراین هیچ تغییری اعمال نشد. 44 پیش‌بینی باقی‌مانده دارای مقادیری بودند ­که بین -2 و 2 تخمین زده شده‌اند. برای مثال، داده‌های پیش‌بینی نشان‌داده‌شده در شکل.  [3. 2](#bookmark2) دارای ارزش تبدیل تخمینی 0. 1 هستند که نشان دهنده ورود به سامانه است



شکل 3. 3: *سمت چپ* : هیستوگرام پیش‌بینی محیط سلول. *راست* : همان داده‌ها پس از تبدیل Box-Cox با *A* تخمین زده می‌شود *-* 1. 1

تحول منطقی است پیش‌بینی دیگر، محیط تخمینی سلول، *تخمینی* برابر با *1* داشت. 1. برای این داده‌ها، مقادیر اصلی و تبدیل شده در شکل نشان داده شده است.  [3. 3](#bookmark147) .

3. 3 تبدیل داده‌ها برای پیش‌بینی‌کننده‌های چندگانه

این تبدیل متغیر بر روی گروه‌هایی از پیش‌بینی‌کننده‌ها، معمولاً کل مجموعه مورد بررسی، عمل می‌کنند. روش‌هایی برای حل و فصل موارد پرت و کاهش ابعاد داده‌ها از اهمیت اولیه برخوردار است.

تبدیل متغیر برای حل موارد پرت

ما به‌طور کلی مقادیر پرت را به‌عنوان نمونه‌هایی تعریف می‌کنیم که به‌طور استثنایی از جریان اصلی داده‌ها فاصله دارند. بر اساس مفروضات خاص، ­تعاریف آماری رسمی از موارد پرت وجود دارد. حتی با درک کامل داده‌ها، تعریف نقاط پرت ممکن است دشوار باشد. با این حال، ما اغلب می‌توانیم یک ­مقدار غیر معمول را با نگاه کردن به یک شکل شناسایی کنیم. هنگامی که یک یا چند نمونه مشکوک به پرت بودن نمونه‌ها هستند، اولین قدم این است که مطمئن شوید که مقادیر از نظر علمی معتبر هستند (مثلاً فشار خون مثبت) و هیچ اشتباهی در ثبت داده‌ها رخ نداده است. باید بسیار مراقب بود که مقادیر، عجولانه حذف یا تغییر داده ­نشود، به خصوص اگر حجم نمونه کوچک باشد. با اندازه‌های کوچک نمونه، نقاط پرت ظاهری ممکن است نتیجه یک توزیع اریب باشد که در آن هنوز داده‌های کافی برای مشاهده چولگی وجود ندارد. همچنین، داده‌های بیرونی ممکن است نشان‌دهنده ­بخش خاصی از جامعه مورد مطالعه باشد که تازه نمونه‌گیری را آغاز کرده‌اند. بسته به نحوه جمع‌آوری داده‌ها، «خوشه‌ای» از نقاط معتبری که خارج از جریان اصلی داده‌ها قرار دارند، ممکن است به جمعیت متفاوتی نسبت به نمونه‌های دیگر تعلق داشته باشند. [[6]](#footnote-6)

چندین مدل پیش‌بینی وجود دارد که در برابر عوامل پرت مقاوم هستند. مدل‌های طبقه‌بندی مبتنی بر درخت، تقسیم‌هایی از داده‌های آموزشی ایجاد می‌کنند و ­معادله پیش‌بینی مجموعه‌ای از گزاره‌های منطقی است مانند «اگر پیش‌بینی *A* بزرگ‌تر از *X* است، کلاس را پیش‌بینی کنید که *Y باشد* »، بنابراین معمولاً نقطه پرت تأثیر استثنایی ندارد. روی مدل همچنین، ماشین‌های بردار پشتیبان برای طبقه‌بندی معمولاً بخشی از نمونه‌های مجموعه آموزشی را هنگام ایجاد یک معادله پیش‌بینی نادیده می‌گیرند. نمونه‌های حذف شده ممکن است دور از مرز تصمیم‌گیری و خارج از جریان اصلی داده باشند.

اگر مدلی نسبت به نقاط پرت حساس در نظر گرفته شود، یکی از تبدیل داده‌ها که می‌تواند مسأله را به حداقل برساند، *علامت فضایی است.* [( سرنیلز و همکاران 2006](#bookmark1025) ). این رویه مقادیر پیش‌بینی را روی یک کره چند بعدی نشان می‌دهد. این باعث می‌شود که همه نمونه‌ها از مرکز کره فاصله یکسانی داشته باشند. از نظر ریاضی، هر نمونه بر مجذور هنجار آن تقسیم می‌شود:

*\**

*ij* =

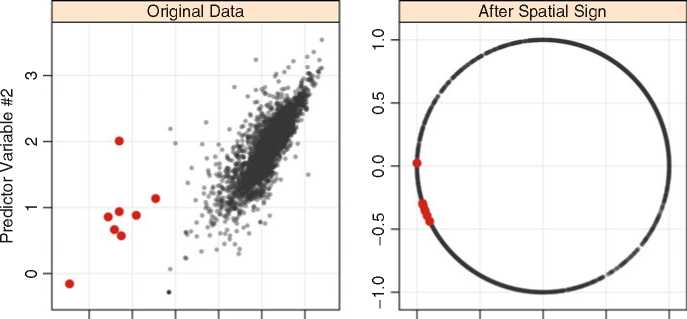
*xij*

E *j*=i *x23*

از آنجایی که مخرج برای اندازه‌گیری فاصله مجذور مرکز توزیع پیش‌بینی در نظر گرفته شده است، مهم است که داده‌های پیش‌بینی را قبل از استفاده از این تبدیل مرکز و مقیاس‌بندی کنیم. توجه داشته باشید که بر خلاف مرکز یا مقیاس بندی، این دستکاری پیش‌بینی‌کننده‌ها آنها را به‌عنوان یک گروه تبدیل می‌کند. حذف متغیرهای پیش‌بینی پس از اعمال تبدیل علامت فضایی ممکن است مسأله ساز باشد.

شکل [3. 4](#bookmark150) مجموعه داده دیگری را با دو پیش‌بینی همبسته نشان می‌دهد. در این داده‌ها، حداقل هشت نمونه از اکثریت داده‌های دیگر دور می‌شوند. این نقاط داده احتمالاً یک زیرجمعیت داده معتبر، اما نمونه ضعیفی هستند. مدلساز بررسی می‌کند که چرا این نقاط متفاوت هستند. شاید آنها نماینده گروهی از علاقه مندی‌ها، مانند مشتریان بسیار سودآور باشند. تبدیل علامت فضایی در پانل سمت راست نشان داده شده است، جایی که تمام نقاط داده در یک فاصله مشترک از مبدا پیش‌بینی می‌شوند. نقاط پرت هنوز در بخش شمال غربی توزیع قرار دارند اما به سمت داخل منقبض می‌شوند. این امر تأثیر نمونه‌ها را بر آموزش مدل کاهش می‌دهد.

جریان اصلی داده • "غیره" •



-6 -4 -2 0 2 4 -1. 0 -0. 5 0. 0 0. 5 1. 0

متغیر پیش‌بینی شماره 1

شکل 3. 4: *سمت چپ* : یک مثال گویا با گروهی از نقاط داده دور افتاده. *راست* : هنگامی که داده‌های اصلی تبدیل می‌شوند، نتایج نتایج را به سمت اکثریت داده‌ها می‌رساند

کاهش داده و استخراج ویژگی

تکنیک‌های کاهش داده دسته دیگری از تبدیل‌های پیش‌بینی هستند. این روش‌ها با تولید مجموعه کوچک‌تری از پیش‌بینی‌کننده‌ها که به دنبال جذب اکثریت اطلاعات در متغیرهای اصلی هستند، داده‌ها را کاهش می‌دهند. به این ترتیب می‌توان از متغیرهای کمتری استفاده کرد که وفاداری معقولی را به داده‌های اصلی ارائه دهد. برای اکثر تکنیک‌های کاهش داده، پیش‌بینی‌کننده‌های جدید توابع پیش‌بینی‌کننده‌های اصلی هستند. بنابراین، همه پیش‌بینی‌کننده‌های اصلی هنوز برای ایجاد متغیرهای جایگزین مورد نیاز هستند. این دسته از روش‌ها اغلب ***استخراج سیگنال***یا تکنیک‌های ***استخراج ویژگی*** *نامیده می‌شوند.*

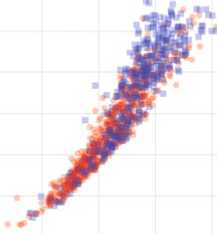
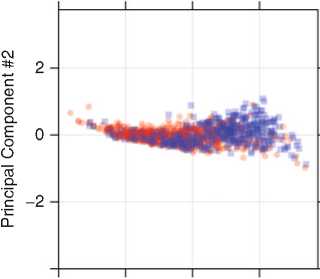
PCA یک تکنیک کاهش داده رایج است ( عبدی و ویلیامز 2010 ). این روش به دنبال یافتن ترکیب‌های خطی از پیش‌بینی‌کننده‌ها، معروف به مؤلفه‌های اصلی (PC) است که بیشترین واریانس ممکن را ثبت می‌کند. اولین PC به‌عنوان ترکیب خطی پیش‌بینی‌کننده‌ها تعریف می‌شود که بیشترین تغییرات را در بین تمام ترکیب‌های خطی ممکن ثبت می‌کند. سپس، PCهای بعدی به گونه‌ای مشتق می‌شوند که این ترکیب‌های خطی بیشترین تغییرات باقی‌مانده را به دست می‌آورند در حالی که با تمام PCهای قبلی همبستگی ندارند. به‌طور ریاضی، *j* th PC را می‌توان به صورت زیر نوشت:

PC *j = ( a j* 1 *x* Predictor 1) + ( *a j 2 x* Predictor 2) + *• • •* + ( *a jp x* Predictor *P* ).

*P* تعداد پیش‌بینی‌کننده‌ها است. ضرایب *a j* 1 , *a j* 2 ,. . . , *a jP وزن مؤلفه* نامیده می‌شوند ­و به ما کمک می‌کنند بفهمیم کدام پیش‌بینی برای هر PC مهمتر است.

**داده‌های اصلی**

PS WS

عرض فیبر کانال 1

**Transformed**

-2 0 2

Principal Component #1

c §

O

S’ l5 2- w

<D

2.8­

2.6 1

2.4­

2.2­

2.0-

1.3 1.4 1.5 1.6

شکل 3. 5: نمونه‌ای از تبدیل مولفه اصلی برای داده‌های بخش‌بندی سلول. *شکل‌ها* و *رنگ‌ها* نشان می‌دهند که کدام سلول‌ها به خوبی تقسیم‌بندی نشده‌اند یا به خوبی تقسیم‌بندی شده‌اند

برای نشان دادن PCA، داده‌های شکل 1 را در نظر بگیرید.  [3. 5](#bookmark151) . این مجموعه شامل زیرمجموعه‌ای از دو پیش‌بینی همبسته، میانگین شدت پیکسل کانال 1 و آنتروپی مقادیر شدت در سلول (معیار شکل سلول) و یک پاسخ طبقه‌بندی است. با توجه به همبستگی زیاد بین پیش‌بینی‌کننده‌ها (0. 93)، می‌توانیم استنباط کنیم که میانگین شدت پیکسل و مقادیر شدت آنتروپی، اطلاعات اضافی در مورد سلول‌ها را اندازه‌گیری می‌کند و می‌توان از پیش‌بینی یا ترکیب خطی این پیش‌بینی‌کننده‌ها به جای پیش‌بینی‌کننده‌های اصلی استفاده کرد. . در این مثال، دو PC را می‌توان استخراج کرد (نمودار سمت راست در شکل 1).  [3. 5 )](#bookmark151) ؛ این تبدیل نشان دهنده چرخش داده‌ها حول محور بیشترین تغییرات است. کامپیوتر اول 97 درصد از تغییرات اولیه را خلاصه می‌کند، در حالی که دومین رایانه 3 درصد را خلاصه می‌کند. از این رو، منطقی است که فقط از اولین PC برای مدل‌سازی استفاده شود زیرا اکثریت اطلاعات در داده‌ها را به خود اختصاص می‌دهد.

مزیت اصلی PCA و دلیل اینکه محبوبیت خود را به‌عنوان روش کاهش داده حفظ کرده است، این است که اجزایی را ایجاد می‌کند که همبستگی ندارند. همانطور که قبلاً در این فصل ذکر شد، برخی از مدل‌های پیش‌بینی ترجیح می‌دهند پیش‌بینی‌کننده‌ها غیرهمبسته (یا حداقل همبستگی کم) باشند تا راه‌حل‌هایی پیدا کنند و پایداری عددی مدل را بهبود بخشند. پیش پردازش PCA پیش‌بینی‌کننده‌های جدیدی با ویژگی‌های مطلوب برای این نوع مدل‌ها ایجاد می‌کند.

در حالی که PCA پیش‌بینی‌کننده‌های جدیدی با ویژگی‌های مطلوب ارائه می‌دهد، باید با درک و دقت مورد استفاده قرار گیرد. شایان ذکر است، متخصص باید ­درک کنند که PCA بدون توجه به درک بیشتر از پیش‌بینی‌کننده‌ها (یعنی مقیاس‌ها یا توزیع‌ها) یا دانش اهداف مدل‌سازی (یعنی متغیر پاسخ) به دنبال تغییرات مجموعه پیش‌بینی است. از این رو، بدون راهنمایی مناسب، PCA می‌تواند مؤلفه‌هایی تولید کند که ویژگی‌های داده‌ها را خلاصه می‌کند که به ساختار زیربنایی داده‌ها و همچنین به هدف نهایی مدل‌سازی بی‌ربط هستند.

از آنجایی که PCA به دنبال ترکیب‌های خطی پیش‌بینی‌کننده‌هایی است که تغییرات را به حداکثر می‌رساند، طبیعتاً ابتدا به جمع‌بندی پیش‌بینی‌کننده‌هایی کشیده می‌شود که تغییرات بیشتری دارند. اگر پیش‌بینی‌کننده‌های اصلی در مقیاس‌های اندازه‌گیری باشند که از نظر بزرگی متفاوت باشند [پیش‌بینی‌کننده‌های جمعیت‌شناختی مانند سطح درآمد (به دلار) و قد (بر حسب فوت) را در نظر بگیرید]، آنگاه چند مؤلفه اول ­بر خلاصه کردن پیش‌بینی‌کننده‌های بزرگ‌تر تمرکز خواهند کرد (مثلاً، درآمد)، در حالی که مولفه‌های دوم پیش‌بینی‌کننده‌های واریانس پایین تر (مثلاً قد) را خلاصه می‌کنند. این به این معنی است که وزن PC برای پیش‌بینی‌کننده‌های تغییرات بالاتر ­در چند مؤلفه اول بزرگ‌تر خواهد بود. علاوه بر این، این بدان معناست که PCA تلاش‌های خود را بر شناسایی ساختار داده بر اساس مقیاس‌های اندازه‌گیری متمرکز خواهد کرد تا بر اساس روابط مهم درون داده‌ها برای مسأله فعلی.

برای اکثر مجموعه داده‌ها، پیش‌بینی‌کننده‌ها در مقیاس‌های مختلف هستند. علاوه بر این، ­عوامل پیش‌بینی ممکن است توزیع‌های منحرفی داشته باشند. از این رو، برای کمک به PCA از خلاصه کردن تفاوت‌های توزیعی و اطلاعات مقیاس پیش‌بینی، بهتر است ابتدا پیش‌بینی‌کننده‌های اریب را تغییر دهیم (بخش.  [3. 2 ) و سپس پیش‌بینی‌کننده‌ها](#bookmark139) را قبل از انجام PCA مرکز و مقیاس کنید. ­مرکز کردن و مقیاس‌بندی PCA را قادر می‌سازد تا روابط زیربنایی را در داده‌ها بیابد بدون اینکه تحت تأثیر مقیاس‌های اندازه‌گیری اصلی قرار گیرد.

دومین هشدار PCA این است که هدف مدلسازی یا متغیر پاسخ را هنگام خلاصه کردن تغییرات در نظر نمی‌گیرد. از آنجایی که PCA نسبت به پاسخ کور است، این یک *تکنیک بدون نظارت است.* اگر رابطه پیش‌بینی بین پیش‌بینی‌کننده‌ها و پاسخ به ­توانایی متغیر پیش‌بینی‌کننده‌ها مرتبط نباشد، PCهای مشتق‌شده رابطه مناسبی با پاسخ ارائه نمی‌کنند. در این مورد، یک *تکنیک نظارت شده،* مانند PLS (Sects.  [6. 3](#bookmark8) و [12. 4 )](#bookmark572) ، مولفه‌ها را در حالی که پاسخ مربوطه را در نظر می‌گیرد، استخراج می‌کند.

هنگامی که ما در مورد تبدیل مناسب متغیرهای پیش‌بینی تصمیم گرفتیم، می‌توانیم PCA را اعمال کنیم. برای مجموعه‌های داده‌ای با متغیرهای پیش‌بینی زیاد، باید تصمیم بگیریم که چه تعداد مؤلفه را حفظ کنیم. یک رویکرد اکتشافی برای تعیین تعداد مؤلفه‌هایی که باید حفظ شوند، ایجاد یک نمودار اسکری (scree plot) است که شامل شماره مؤلفه مرتب شده ( محور *x* ) و مقدار تغییرپذیری خلاصه شده ( محور *y* ) است (شکل.  [3. 6 )](#bookmark152) . برای اکثر مجموعه‌های داده، چند کامپیوتر اول اکثریت تغییرات را خلاصه می‌کنند و نمودار یک نزول تند را نشان می‌دهد. سپس تغییرات برای اجزای باقیمانده کاهش می‌یابد. به‌طور کلی، تعداد مؤلفه قبل از کاهش تغییرات، حداکثر مؤلفه‌ای است که حفظ می‌شود. در شکل [3. 6](#bookmark152) ، تغییرات در جزء 5 کاهش می‌یابد. با استفاده از این قانون کلی، چهار PC باقی می‌مانند. در یک فرآیند ساخت مدل خودکار، تعداد بهینه مولفه‌ها را می‌توان با اعتبارسنجی متقاطع تعیین کرد (به بخش مراجعه کنید.  [4. 4 )](#bookmark210) .

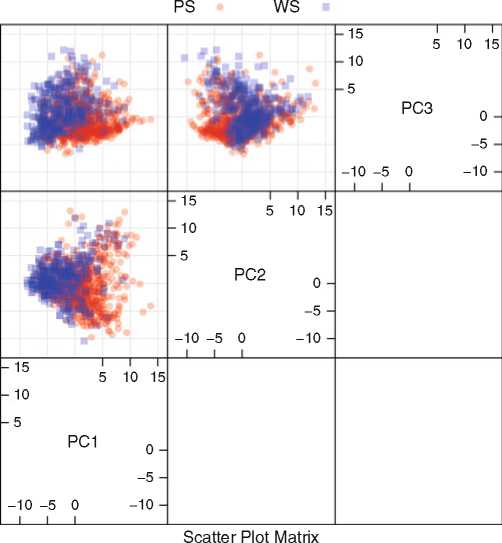


شکل 3. 6: یک "نقشه صفحه" که در آن درصد کل واریانس توضیح داده شده توسط هر جزء نشان داده شده است.

بررسی بصری اجزایِ اصلی، یک گام مهم برای ارزیابی کیفیت داده‌ها و به دست آوردن شهود برای مسأله است. برای انجام این کار، چند مؤلفه اصلی را می‌توان در برابر یکدیگر ترسیم کرد و نمادهای نمودار را می‌توان با ویژگی‌های مربوطه، مانند برچسب‌های کلاس، رنگ آمیزی کرد. اگر PCA مقدار کافی از اطلاعات را در داده‌ها جمع‌آوری کرده باشد، این نوع نمودار می‌تواند خوشه‌هایی از نمونه‌ها یا نقاط پرت را نشان دهد که ممکن است باعث بررسی دقیق‌تر ­نقاط داده‌ای شود. برای مسائل طبقه بندی، نمودار PCA می‌تواند جداسازی بالقوه کلاس‌ها را نشان دهد (در صورت وجود جدایی). این می‌تواند انتظارات اولیه مدل ساز را تعیین کند. اگر خوشه‌بندی کمی از کلاس‌ها وجود داشته باشد، نمودار مقادیر مؤلفه اصلی همپوشانی قابل‌توجهی از نقاط را برای هر کلاس نشان می‌دهد. هنگام ترسیم قطعات باید دقت شود. مقیاس مولفه‌ها کوچکتر می‌شود زیرا تغییرات کمتر و کمتری در داده‌ها ایجاد می‌کنند. به‌عنوان مثال، در شکل.  [3. 5](#bookmark151) ، مقادیر جزء یک از 7/3- تا 4/3 متغیر *است.* در حالی که مولفه دو از *1- تا* 1/1 متغیر است. اگر محورها در مقیاس‌های جداگانه نمایش داده شوند، پتانسیل تفسیر بیش از حد الگوهایی که ممکن است برای ­مؤلفه‌هایی که تغییرات کمی دارند، وجود داشته باشد. برای نمونه‌های دیگر از این موضوع به Geladi, Manley, and Lestander [( 2003 ) مراجعه کنید.](#bookmark1016)

PCA برای کل مجموعه پیش‌بینی‌کننده‌های داده‌های تقسیم‌بندی اعمال شد. همانطور که قبلا نشان داده شد، برخی از پیش‌بینی‌کننده‌ها با چولگی قابل‌توجهی وجود دارد. از آنجایی که پیش‌بینی‌کننده‌های اریب می‌توانند بر PCA تأثیر بگذارند، 44 متغیر وجود داشت که با استفاده از روش Box-Cox که قبلاً توضیح داده شد، تبدیل شدند. پس از تبدیل‌ها، پیش‌بینی‌کننده‌ها قبل از انجام PCA متمرکز و مقیاس‌بندی شدند.

شکل [3. 6](#bookmark152) درصد تغییرات کل در داده‌ها را نشان می‌دهد که توسط هر جزء محاسبه شده است. توجه داشته باشید که با اضافه شدن اجزای بیشتر، درصدها کاهش می‌یابد. سه مؤلفه اول به ترتیب 14 درصد، 6/12 درصد و 4/9 درصد از کل واریانس را به خود اختصاص دادند. پس از چهار جزء،

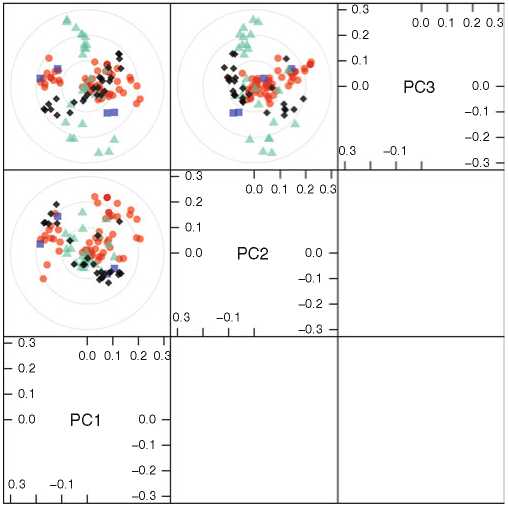


شکل 3. 7: نموداری از سه جزء اصلی اول برای ­داده‌های بخش‌بندی سلول، رنگ آمیزی بر اساس نوع سلول

کاهش شدیدی در درصد تغییرات توضیح داده شده وجود دارد، ­اگرچه این چهار مؤلفه تنها 42. 4 درصد از اطلاعات مجموعه داده را توصیف می‌کنند.

شکل [3. 7](#bookmark153) ماتریس نمودار پراکندگی را برای سه جزء اصلی اول نشان می‌دهد. نقاط بر اساس کلاس رنگ می‌شوند (کیفیت بخش بندی). از آنجایی که درصد ­تغییرات توضیح داده شده برای سه جزء اول زیاد نیست، مهم است که تصویر حاصل را بیش از حد تفسیر نکنید. از این نمودار، به نظر می‌رسد که هنگام ترسیم مؤلفه‌های اول و دوم، بین کلاس‌ها جدایی وجود دارد. با این حال، توزیع سلول‌های تقسیم‌بندی شده به خوبی در توزیع سلول‌هایی که ضعیف شناسایی شده‌اند، وجود دارد. یک نتیجه‌گیری از این تصویر این است که انواع سلول‌ها به راحتی ­*از هم جدا نمی‌شوند.* با این حال، این بدان معنا نیست که مدل‌های دیگر، به‌ویژه مدل‌هایی که می‌توانند روابط بسیار غیرخطی را در خود جای دهند، به همین نتیجه می‌رسند. همچنین، در حالی که سلول‌هایی در داده‌ها وجود دارد که کاملاً در جریان اصلی داده نیستند، هیچ نقطه پرت آشکاری وجود ندارد.

یکی دیگر از کاربردهای اکتشافی PCA مشخص کردن این است که کدام پیش‌بینی‌کننده‌ها ­با هر جزء مرتبط هستند. به یاد داشته باشید که هر جزء یک ترکیب خطی ­از پیش‌بینی‌کننده‌ها است و ضریب هر پیش‌بینی، بارگذاری یا وزن نامیده می‌شود. بارگذاری‌های نزدیک به صفر نشان می‌دهد که متغیر پیش‌بینی کمک زیادی به آن جزء نکرده است. شکل [3. 8](#bookmark154) بارگذاری سه جزء اول را در داده‌های بخش‌بندی سلول نشان می‌دهد. هر نقطه مربوط به یک متغیر پیش‌بینی است و توسط کانال نوری مورد استفاده در آزمایش قبلی رنگ می‌شود. برای اولین جزء اصلی، بارگذاری برای اولین کانال (مرتبط با بدنه سلولی) در نهایت قرار دارد. این نشان می‌دهد که ویژگی‌های بدن سلولی بیشترین تأثیر را بر اولین مؤلفه اصلی و با گسترش مقادیر پیش‌بینی دارند. همچنین توجه داشته باشید که اکثر بارهای ­کانال سوم (اندازه‌گیری اکتین و توبولین) برای جزء اول به صفر نزدیکتر است. برعکس، سومین مؤلفه اصلی بیشتر با کانال سوم مرتبط است در حالی که کانال بدن سلولی در اینجا نقش کوچکی ایفا می‌کند. حتی اگر اندازه‌گیری‌های بدن سلولی تغییرات بیشتری ­را در داده‌ها به حساب می‌آورند، اما این بدان معنا نیست که این متغیرها با پیش‌بینی کیفیت تقسیم‌بندی مرتبط هستند

3. 4 مقابله با مقادیر گمشده

Channel 1 •

Channel 2 ■

Channel 3

Channel 4

Scatter Plot Matrix

Fig. 3.8: A plot of the loadings of the first three principal components for the cell segmentation data, colored by optical channel. Recall that channel one was associated with the cell body, channel two with the cell nucleus, channel three with actin, and channel four with tubulin

در بسیاری از موارد، برخی از پیش‌بینی‌کننده‌ها هیچ مقداری برای یک نمونه معین ندارند. این داده‌های گمشده ممکن است از نظر *ساختاری ناپدید شوند،* مانند تعداد فرزندانی که یک مرد به دنیا آورده است. در موارد دیگر، مقدار نمی‌تواند یا در زمان ساخت مدل تعیین نشده است.

مهم است که بفهمیم *چرا* مقادیر گم شده اند. اول ­از همه، مهم است که بدانیم آیا الگوی داده‌های از دست رفته با نتیجه مرتبط است یا خیر. به این «فقدان اطلاعاتی» می‌گویند، زیرا الگوی داده‌های از دست رفته به خودی خود آموزشی است. فقدان اطلاعاتی می‌تواند ­بایاس قابل‌توجهی را در مدل القا کند. در فصل مقدماتی، یک مثال کوتاه در رابطه با پیش‌بینی پاسخ بیمار به دارو بیان شد. فرض کنید دارو به شدت بی اثر بود یا عوارض جانبی قابل‌توجهی داشت. ممکن است بیمار ملاقات‌های دکتر را از دست بدهد یا از مطالعه خارج شود. در این مورد، به وضوح بین احتمال از دست رفتن مقادیر و درمان رابطه وجود دارد. رتبه‌بندی‌های مشتریان اغلب می‌تواند اطلاعاتی را از دست بدهد. مردم زمانی که نظرات قوی (خوب یا بد) دارند، مجبور هستند به محصولات امتیاز دهند. در این حالت، داده‌ها با داشتن مقادیر کمی در وسط مقیاس رتبه بندی، به احتمال زیاد قطبی می‌شوند. در رقابت یادگیری ماشینی جایزه نتفلیکس برای پیش‌بینی فیلم‌هایی که مردم بر اساس رتبه‌بندی قبلی‌شان دوست دارند، «اثر دینامیت ناپلئون» بسیاری از شرکت‌کنندگان را گیج کرد زیرا افرادی که به فیلم *ناپلئون دینامیت امتیاز دادند،* آن را دوست داشتند یا از آن متنفر بودند.

داده‌های از دست رفته را نباید با داده‌های *سانسور شده اشتباه* گرفت، جایی که در خصوص داده‌های *سانسور شده* مقدار دقیق آن وجود ندارد اما چیزی در مورد ارزش آن مشخص است. به‌عنوان مثال، شرکتی که دیسک‌های فیلم را از طریق پست اجاره می‌دهد، ممکن است از مدت زمانی که مشتری یک فیلم را در مدل‌های خود نگه داشته است استفاده کند. اگر مشتری هنوز فیلمی را پس نداده باشد، ما از مدت زمان واقعی اطلاع نداریم، فقط به اندازه مدت زمان فعلی است. هنگام استفاده از اندازه‌گیری‌های آزمایشگاهی، داده‌های سانسور شده نیز می‌توانند رایج باشند. برخی از سنجش‌ها نمی‌توانند زیر حد تشخیص خود اندازه‌گیری شوند. در چنین مواردی، می‌دانیم که مقدار از حد کوچکتر است اما دقیقاً اندازه‌گیری نشده است.

اغلب میدانیم داده‌های *سانسور شده* از مقدار مشخصی بیشتر یا از مقدار مشخصی کمتر است.

آیا با داده‌های سانسور شده متفاوت از داده‌های از دست رفته رفتار می‌شود؟ هنگام ساخت مدل‌های آماری سنتی با تمرکز بر تفسیر یا استنباط، سانسور ­معمولاً به روشی رسمی با ایجاد مفروضاتی ­در مورد مکانیسم سانسور در نظر گرفته می‌شود. برای مدل‌های پیش‌بینی، ­معمول‌تر است که این داده‌ها را به‌عنوان داده‌های گمشده ساده تلقی کنیم یا از مقدار سانسور شده به‌عنوان مقدار مشاهده‌شده استفاده کنیم. به‌عنوان مثال، هنگامی که یک نمونه دارای مقدار کمتر از حد تشخیص است، می‌توان از مقدار حد به جای مقدار واقعی استفاده کرد. برای این وضعیت، استفاده از یک عدد تصادفی بین صفر و حد تشخیص نیز رایج است.

در تجربه ما، مقادیر از دست رفته بیشتر ­از نمونه به متغیرهای پیش‌بینی مربوط می‌شوند. به همین دلیل، مقدار داده‌های از دست رفته ممکن است ­در زیرمجموعه‌ای از پیش‌بینی‌کننده‌ها متمرکز شود نه اینکه به‌طور تصادفی در همه پیش‌بینی‌کننده‌ها رخ دهد. در برخی موارد، درصد داده‌های از دست رفته به اندازه کافی قابل‌توجه است تا این پیش‌بینی را از فعالیت‌های مدل‌سازی بعدی حذف کند.

مواردی وجود دارد که مقادیر از دست رفته ممکن است در نمونه‌های خاصی متمرکز شوند. برای مجموعه داده‌های بزرگ، حذف نمونه‌ها بر اساس مقادیر از دست رفته مشکلی نیست، با این فرض که کمبود اطلاعاتی نیست. در مجموعه داده‌های کوچک‌تر، حذف نمونه‌ها قیمت بالایی دارد. برخی از رویکردهای جایگزین شرح داده شده در زیر ممکن است مناسب تر باشند.

اگر داده‌های از دست رفته را حذف نکنیم، دو رویکرد کلی وجود دارد. اول، چند مدل پیش‌بینی، به‌ویژه تکنیک‌های مبتنی بر درخت، می‌توانند به‌طور ­مشخص داده‌های از دست رفته را حساب کنند. اینها در فصل بیشتر مورد بحث قرار می‌گیرند.  [8 .](#bookmark388)

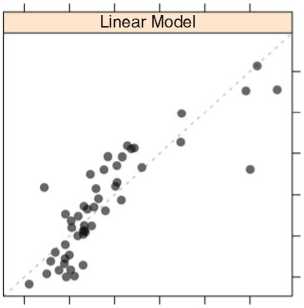
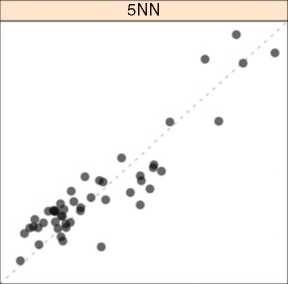
از طرف دیگر، داده‌های از دست رفته را می‌توان منتسبت [به سایر متغیرها] کرد (ایمپیوت کرد). در این مورد، می‌توانیم در ­شکل‌گیری در مجموعه آموزشی از پیش‌بینی‌کننده‌ها استفاده کنیم تا در اصل، مقادیر پیش‌بینی‌کننده‌های دیگر را تخمین بزنیم. این به یک مدل پیش‌بینی در یک مدل پیش‌بینی می‌شود.

انتساب به‌طور گسترده در ادبیات آماری مورد مطالعه قرار گرفته است، اما در زمینه ایجاد روش‌های آزمون فرضیه صحیح در ­حضور داده‌های از دست رفته، این یک مسأله جداگانه است. برای مدل‌های پیش‌بینی، ما نگران دقت پیش‌بینی‌کننده‌ها هستیم تا استنتاج‌های معتبر. ادبیات کوچکی در مورد انتساب برای مدل‌های پیش‌بینی وجود دارد. [Saar-Tsechansky و Provost ( 2007b](#bookmark1025) ) مسئله مقادیر از دست رفته را بررسی می‌کنند و در مورد چگونگی برخورد مدل‌های خاص با این موضوع تحقیق می‌کنند.  [جرز و همکاران ( 2010](#bookmark1019) ) همچنین به طیف گسترده‌ای از روش‌های انتساب برای یک مجموعه داده خاص نگاه می‌کند.

همانطور که قبلا ذکر شد، انتساب تنها لایه دیگری از مدل‌سازی است که در آن سعی می‌کنیم مقادیر متغیرهای پیش‌بینی را بر اساس سایر متغیرهای پیش‌بینی تخمین بزنیم. مرتبط‌ترین طرح برای انجام این کار، استفاده از مجموعه آموزشی برای ساخت یک مدل انتساب برای هر پیش‌بینی در مجموعه داده است. قبل از آموزش مدل یا پیش‌بینی نمونه‌های جدید، مقادیر گمشده با استفاده از انتساب پر می‌شوند. توجه داشته باشید که این لایه اضافی از مدل‌ها عدم قطعیت را اضافه می‌کند. اگر از نمونه‌گیری مجدد برای انتخاب مقادیر پارامتر تنظیم یا تخمین عملکرد استفاده می‌کنیم، انتساب باید در نمونه‌گیری مجدد گنجانده شود. این باعث افزایش زمان محاسباتی برای ساخت مدل‌ها می‌شود، اما تخمین‌های صادقانه‌ای از عملکرد مدل نیز ارائه می‌دهد.

اگر تعداد پیش‌بینی‌کننده‌های تحت‌تاثیر مقادیر از دست رفته کم باشد، یک ­تحلیل آزمایشی از روابط بین پیش‌بینی‌کننده‌ها ایده خوبی است. برای مثال، تجسم‌سازی‌ها یا روش‌هایی مانند PCA را می‌توان برای تعیین اینکه آیا روابط قوی بین پیش‌بینی‌کننده‌ها وجود دارد یا خیر، استفاده کرد. اگر یک متغیر با ­مقادیر از دست رفته با پیش‌بینی دیگری که مقادیر کمی از دست رفته دارد، همبستگی بالایی داشته باشد، یک مدل متمرکز اغلب می‌تواند برای انتساب مؤثر باشد (نمونه زیر را ببینید).

یکی از تکنیک‌های رایج برای انتساب، مدل *K-* nearest همسایه است. یک نمونه جدید با یافتن نمونه‌ها در مجموعه آموزشی «نزدیک‌ترین» به آن نسبت داده می‌شود و این نقاط نزدیک را برای پر کردن مقدار میانگین می‌گیرد.  [ترویانسکایا و همکاران](#bookmark1026) [( 2001](#bookmark1026) ) این رویکرد را برای داده‌های با ابعاد بالا با حجم نمونه کوچک بررسی می‌کند. یکی از مزیت‌های این رویکرد این است که داده‌های منتسب به محدوده مقادیر مجموعه آموزشی محدود می‌شوند. یکی از معایب این است که کل مجموعه آموزشی هر بار که نیاز به انتساب یک مقدار از دست رفته باشد، مورد نیاز است. همچنین، تعداد همسایگان یک پارامتر تنظیم است. با این حال، [ترویانسکایا و همکاران ( 2001](#bookmark1026) ) دریافتند که رویکرد نزدیکترین همسایه نسبت به پارامترهای تنظیم و همچنین مقدار داده‌های از دست رفته نسبتاً مستحکم (روباست) است.



<15

1.5-

1.0-

0.5-

0.0-

•0.5-

— I

1.0-

1.0

0.5

1.0

1.5

-0.5 0.0

0.5

1.0

1.5

-0.5 0.0

-1.0

مقدار اصلی (مرکز و مقیاس شده)

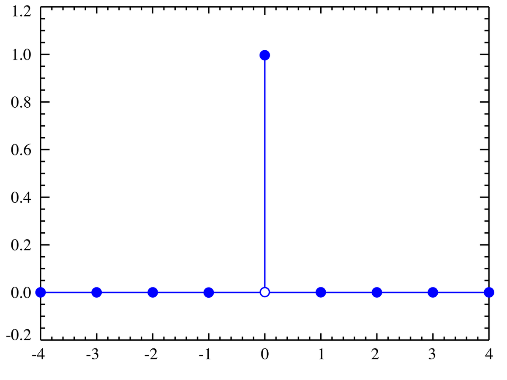
شکل 3. 9: پس از شبیه‌سازی 50 مقدار مجموعه آزمون گمشده به صورت تصادفی برای داده‌های محیطی سلول، دو مدل انتساب متفاوت با مجموعه آموزشی ساخته شد و بر روی مقادیر مجموعه آزمون از دست رفته اعمال شد. این نمودار مقادیر متمرکز و مقیاس شده را قبل و بعد از انتساب نشان می‌دهد که "نزدیک" دو نقطه را خاتمه می‌دهد.

در بخش [3. 2 ،](#bookmark139) یک پیش‌بینی که محیط سلول را اندازه‌گیری می‌کند برای نشان دادن ­چولگی استفاده شد (شکل 2 را ببینید).  [3. 3 )](#bookmark147) . به‌عنوان مثال، یک مدل 5 نزدیکترین همسایه با استفاده از مقادیر مجموعه آموزشی ایجاد شد. در مجموعه آزمون، مقادیر گمشده به‌طور تصادفی در 50 مقدار محیط سلولی مجموعه آزمایشی القا شد و سپس به مدل نسبت داده ­شد. شکل [3. 9](#bookmark161) نمودار پراکندگی نمونه‌ها را نشان می‌دهد که روی از دست رفته تنظیم شده‌اند. پانل سمت چپ نتایج رویکرد 5 نزدیکترین همسایه را نشان می‌دهد. این مدل انتساب کار خوبی را در پیش‌بینی نمونه‌های موجود انجام می‌دهد. همبستگی بین مقادیر واقعی و نسبت داده شده 0. 91 است.

متناوبا، می‌توان از یک رویکرد ساده تر برای نسبت دادن محیط سلول استفاده کرد. طول فیبر سلولی، یکی دیگر از پیش‌بینی‌کننده‌های مرتبط با اندازه سلول، همبستگی بسیار بالایی (0. 99) با داده‌های محیطی سلول دارد. ما می‌توانیم یک مدل رگرسیون خطی ساده با استفاده از این داده‌ها برای پیش‌بینی مقادیر از دست رفته ایجاد کنیم. این نتایج در پانل سمت راست شکل.  [3. 9 .](#bookmark161) برای این رویکرد، همبستگی بین مقادیر واقعی و منتسب 0. 85 است.

3. 5 حذف پیش‌بینی کننده‌ها

حذف پیش‌بینی‌کننده‌ها قبل از مدل‌سازی مزایای بالقوه‌ای دارد. اول، پیش‌بینی‌کننده‌های کمتر به معنای کاهش زمان و پیچیدگی محاسباتی است. دوم، اگر دو پیش‌بینی‌کننده همبستگی بالایی داشته باشند، این نشان می‌دهد که آنها اطلاعات زیربنایی یکسانی را اندازه‌گیری می‌کنند. حذف یک پیش‌بینی‌کننده نباید ­عملکرد مدل را به خطر بیندازد و ممکن است منجر به مدلی مقرون‌به‌صرفه‌تر و قابل تفسیر شود. سوم، برخی از مدل‌ها می‌توانند توسط پیش‌بینی‌کننده‌ها با توزیع‌های منحط (دی‌جِنِرِیت) فلج شوند. در این موارد، بهبود قابل‌توجهی ­در عملکرد و/یا پایداری مدل بدون متغیرهای مسأله ساز وجود دارد.



degenerate distribution

توزیع منحط (توزیع تک مقداری با واریانس صفر)

یک متغیر پیش‌بینی را در نظر بگیرید که دارای یک مقدار منحصربه‌فرد است. ما به این نوع داده‌ها به‌عنوان پیش‌بینی واریانس صفر اشاره می‌کنیم. برای برخی از مدل‌ها، چنین ­متغیر غیر اطلاعاتی ممکن است تأثیر کمی بر محاسبات داشته باشد. یک مدل مبتنی بر درخت (Sects.  [8. 1](#bookmark392) و [14. 1 )](#bookmark686) نسبت به این نوع پیش‌بینی نفوذ ناپذیر است زیرا هرگز در یک تقسیم استفاده نمی‌شود. با این حال، مدلی مانند رگرسیون خطی این داده‌ها را مسأله‌ساز می‌داند و احتمالاً باعث ایجاد خطا در محاسبات می‌شود­. در هر صورت، این داده‌ها هیچ اطلاعاتی ندارند و به راحتی می‌توان آنها را دور انداخت. به‌طور مشابه، برخی از پیش‌بینی‌کننده‌ها ممکن است تنها تعداد انگشت شماری از مقادیر منحصر به فرد داشته باشند که با فرکانس‌های بسیار پایین رخ می‌دهند. این " پیش‌بینی‌کننده با واریانس نزدیک به صفر ­" ممکن است یک مقدار واحد برای اکثریت قریب به اتفاق نمونه‌ها داشته باشند.

یک برنامه متن کاوی را در نظر بگیرید که در آن تعداد کلمات کلیدی برای مجموعه بزرگی از اسناد جمع‌آوری می‌شود. پس از فیلتر کردن «کلمات توقف» که معمولاً استفاده می‌شوند، مانند « *و* »، می‌توان متغیرهای پیش‌بینی را برای کلمات کلیدی جالب ایجاد کرد. فرض کنید یک کلمه کلیدی در گروه کوچکی از اسناد وجود دارد اما در غیر این صورت استفاده نشده است. توزیع فرضی چنین توزیع تعداد کلمات در جدول آورده شده است [3. 1](#bookmark165) . از 531 سندی که جستجو شد، تنها چهار عدد منحصر به فرد وجود داشت. اکثریت اسناد (523) کلید ­واژه را ندارند. در حالی که شش سند دارای دو وقوع، یک سند دارای سه و سند دیگر دارای شش وقوع است. از آنجایی که 98 درصد داده‌ها دارای مقادیر صفر هستند، تعداد کمی از اسناد ممکن است تأثیر نامناسبی بر مدل داشته باشند. همچنین، در صورت استفاده از نمونه‌گیری مجدد (بخش [4. 4 )](#bookmark210) ، احتمال قوی وجود دارد که یکی از مجموعه داده‌های نمونه برداری مجدد (بخش.  [4. 4 )](#bookmark210) فقط حاوی اسناد بدون کلمه کلیدی خواهد بود، بنابراین این پیش‌بینی فقط یک مقدار منحصر به فرد خواهد داشت.

چگونه کاربر می‌تواند این حالت از داده‌های مسأله ساز را تشخیص دهد؟ اول، تعداد ­نقاط منحصر به فرد در داده‌ها باید نسبت به تعداد نمونه‌ها کم باشد. در مثال سند، 531 سند در مجموعه داده وجود دارد، اما تنها چهار مقدار منحصر به فرد، بنابراین درصد مقادیر منحصر به فرد 0. 8٪ است. درصد کمی از مقادیر منحصر به فرد به خودی خود دلیلی برای نگرانی به‌عنوان بسیاری از "متغیرهای ساختگی" نیست (بخش.  [3. 6](#bookmark169) زیر) که از پیش‌بینی‌کننده طبقه‌بندی شده تولید شده است، با این توصیف مطابقت دارد. مسأله زمانی رخ می‌دهد که فراوانی این مقادیر منحصر به فرد به شدت نامتناسب باشد. نسبت رایج‌ترین فرکانس به دومین فرکانس رایج نشان دهنده عدم تعادل در فرکانس‌ها است. اکثر اسناد موجود در مجموعه داده‌ها ( *n* = 523) کلمه کلیدی ندارند. پس از این، متداول‌ترین مورد اسناد با دو رخداد ( *n* = 6) است. نسبت این فرکانس‌ها، 523 */* 6 = 87، نسبتاً زیاد است و نشان دهنده عدم تعادل قوی است.

جدول 3. 1: پیش‌بینی‌ای که تعداد اسنادی را که یک کلمه کلیدی در آنها رخ داده است، توصیف می‌کند

#اسناد

|  |  |
| --- | --- |
| موارد وقوع: 0 | 523 |
| موارد وقوع: 2 | 6 |
| موارد وقوع: 3 | 1 |
| موارد وقوع: 6 | 1 |

با توجه به این، یک قانون کلی برای تشخیص پیش‌بینی‌کننده‌های واریانس نزدیک به صفر این است:

* کسری از مقادیر منحصر به فرد در اندازه نمونه کم است (مثلاً 10٪).
* نسبت فراوانی رایج‌ترین مقدار به فراوانی دومین مقدار رایج بزرگ است (مثلاً حدود 20).

اگر هر دوی این معیارها درست باشند و مدل مورد نظر مستعد این نوع پیش‌بینی باشد، حذف متغیر از مدل ممکن است سودمند باشد.

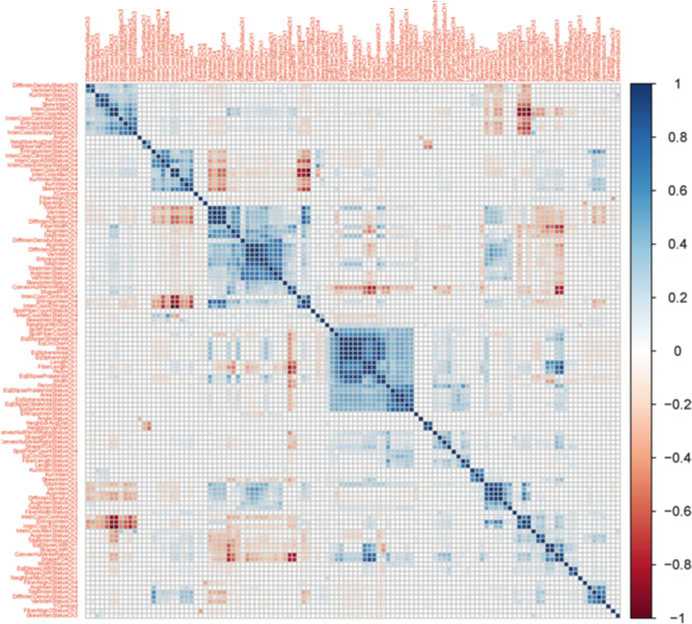
همبستگی‌های بین پیش‌بینی

Collinearity اصطلاح فنی برای وضعیتی است که در آن یک جفت ­*متغیر* پیش‌بینی همبستگی قابل‌توجهی با یکدیگر دارند. همچنین ممکن است روابط بین چندین پیش‌بینی به‌طور همزمان وجود داشته باشد (که multicollinearity یا *چند خطی نامیده می‌شود* ).

به‌عنوان مثال، داده‌های بخش‌بندی سلول دارای تعدادی پیش‌بینی است که اندازه سلول را منعکس می‌کند. اندازه‌گیری‌های محیط، عرض و طول سلول و همچنین محاسبات پیچیده‌تر دیگری وجود دارد. همچنین ویژگی‌هایی وجود دارد که مورفولوژی سلول (یعنی شکل) را اندازه‌گیری می‌کند، مانند زبری سلول.

شکل [3. 10](#bookmark166) ماتریس همبستگی مجموعه آموزشی را نشان می‌دهد. هر همبستگی زوجی از داده‌های آموزشی محاسبه شده و با توجه به بزرگی آن رنگ می‌شود. این تجسم متقارن است: مورب‌های بالا و پایین اطلاعات یکسانی را نشان می‌دهند. رنگ‌های آبی تیره نشان‌دهنده همبستگی‌های مثبت قوی هستند، قرمز تیره برای همبستگی‌های منفی قوی استفاده می‌شود و سفید دلالت بر هیچ رابطه تجربی بین پیش‌بینی‌کننده‌ها ندارد. **در این شکل، متغیرهای پیش‌بین با استفاده از تکنیک خوشه‌بندی گروه‌بندی شده‌اند** [**( Everitt et al. 2011**](#bookmark1015) **) به‌طوری که گروه‌های خطی پیش‌بینی‌کننده‌ها در مجاورت یکدیگر باشند**. با نگاهی به مورب، بلوک‌هایی از همبستگی‌های مثبت قوی وجود دارد که ­«خوشه‌های» همخطی را نشان می‌دهند. نزدیک مرکز مورب بلوک بزرگی از پیش‌بینی‌کننده‌ها از کانال اول قرار دارد. این پیش‌بینی‌کننده‌ها به اندازه سلول مانند عرض و طول سلول مربوط می‌شوند.

هنگامی که مجموعه داده‌ها از پیش‌بینی‌کننده‌های زیادی تشکیل می‌شود که نمی‌توان آن‌ها را به صورت بصری بررسی کرد، تکنیک‌هایی مانند PCA می‌توانند برای توصیف بزرگی مسأله استفاده شوند. به‌عنوان مثال، اگر اولین جزء اصلی یک درصد واریانس بزرگ را تشکیل دهد ، این نشان می‌دهد که حداقل یک گروه از پیش‌بینی‌کننده‌ها وجود دارند که اطلاعات یکسانی را نشان می‌دهند. به‌عنوان مثال، شکل.  [3. 6](#bookmark152) نشان می‌دهد که 3-4 جزء اول سهم نسبی در ­واریانس کل دارند. این نشان می‌دهد که حداقل 3-4 رابطه معنادار بین پیش‌بینی‌کننده‌ها وجود دارد. بارگذاری‌های PCA می‌توانند برای درک اینکه کدام پیش‌بینی‌کننده‌ها با هر مؤلفه مرتبط هستند تا این روابط را از بین ببرند، استفاده شوند.



شکل 3. 10: تصویری از ماتریس همبستگی تقسیم‌بندی سلولی. ترتیب متغیرها بر اساس یک الگوریتم خوشه‌بندی است

به‌طور کلی، دلایل خوبی برای اجتناب از داده‌هایی با پیش‌بینی‌کننده‌های بسیار همبسته وجود دارد. اول، پیش‌بینی‌کننده‌های اضافی اغلب پیچیدگی بیشتری به مدل اضافه می‌کنند تا اطلاعاتی که به مدل ارائه می‌کنند. در شرایطی که ­به دست آوردن داده‌های پیش‌بینی پرهزینه است (چه از نظر زمانی یا پولی)، بدیهی است که متغیرهای کمتر بهتر است. در حالی که این استدلال عمدتاً فلسفی است، وجود داده‌های پیش‌بینی همبسته دارای معایب ریاضی است. استفاده از پیش‌بینی‌کننده‌های بسیار همبسته در تکنیک‌هایی مانند رگرسیون خطی می‌تواند منجر به مدل‌های بسیار ناپایدار، خطاهای عددی و عملکرد پیش‌بینی بدتر شود.

تحلیل رگرسیون کلاسیک دارای چندین ابزار برای تشخیص چند خطی برای رگرسیون خطی است. از آنجایی که پیش‌بینی‌کننده‌های Collinear، می‌توانند بر واریانسِ *تخمین‌های پارامتر* در این مدل تأثیر بگذارند، آماری به نام **ضریب تورم واریانس** [**( VIF**](#bookmark1022) **)** می‌تواند برای شناسایی پیش‌بینی‌کننده‌هایی که تحت تأثیر قرار می‌گیرند استفاده شود. [1994](#bookmark1022) ). فراتر از رگرسیون خطی، این روش ممکن است **به چند دلیل ناکافی باشد**: 1- این روش برای مدل‌های خطی توسعه یافته است، 2- به نمونه‌های بیشتری نسبت به متغیرهای پیش‌بینی نیاز دارد و 3- در حالی که پیش‌بینی‌کننده‌های خطی را شناسایی می‌کند، تعیین نمی‌کند که کدام یک باید حذف شود تا مسأله حل شود.

**یک رویکرد کمتر نظری و بیشتر اکتشافی برای مقابله با این موضوع، حذف حداقل تعداد پیش‌بینی‌کننده‌ها است تا اطمینان حاصل شود که همه همبستگی‌های زوجی زیر یک آستانه مشخص هستند. در حالی که این روش تنها خطوط متقابل را در دو بعد شناسایی می‌کند، اما می‌تواند تأثیر مثبت قابل‌توجهی بر عملکرد برخی از مدل‌ها داشته باشد.**

الگوریتم به شرح زیر است:

1. ماتریس همبستگی پیش‌بینی‌کننده‌ها را محاسبه کنید.
2. دو متغیر پیش‌بینی مرتبط با بزرگترین همبستگی زوجی مطلق را تعیین کنید (آنها را متغیر پیش‌بینی *A* و *B بنامید* ).
3. میانگین همبستگی بین *A* و سایر متغیرها را تعیین کنید. همین کار را برای متغیر پیش‌بینی *B انجام دهید.*
4. اگر *A* دارای میانگین همبستگی بزرگتری است، آن را حذف کنید. در غیر این صورت، متغیر پیش‌بینی *B را حذف کنید.*
5. مراحل 2-4 را تکرار کنید تا زمانی که هیچ همبستگی مطلق بالاتر از آستانه نباشد.

ایده این است که ابتدا پیش‌بینی‌کننده‌هایی که بیشترین همبستگی را دارند حذف کنیم.

فرض کنید می‌خواهیم از مدلی استفاده کنیم که به ویژه به همبستگی‌های بین پیش‌بینی حساس است، ممکن است آستانه 0. 75 را اعمال کنیم. این بدان معنی است که ما می‌خواهیم حداقل تعداد پیش‌بینی‌کننده‌ها را حذف کنیم تا به همه همبستگی‌های زوجی کمتر از 0. 75 دست یابیم. برای داده‌های تقسیم بندی، این الگوریتم حذف 43 پیش‌بینی را پیشنهاد می‌کند.

همانطور که قبلا ذکر شد، روش‌های استخراج ویژگی (به‌عنوان مثال، اجزای اصلی ­) تکنیک دیگری برای کاهش اثر همبستگی قوی بین پیش‌بینی‌کننده‌ها هستند. با این حال، این تکنیک‌ها ارتباط بین پیش‌بینی‌کننده‌ها و نتیجه را پیچیده‌تر می‌کنند. علاوه بر این، از آنجایی که روش‌های استخراج سیگنال ­معمولاً بدون نظارت هستند، هیچ تضمینی وجود ندارد که پیش‌بینی‌کننده‌های جایگزین منتج رابطه‌ای با نتیجه داشته باشند.

اضافه کردن پیش‌بینی ها

هنگامی که یک پیش‌بینی مقوله‌ای است، مانند جنسیت یا نژاد، تجزیه پیش‌بین به مجموعه‌ای از متغیرهای خاص‌تر معمول است. به‌عنوان مثال، داده‌های امتیازدهی اعتباری که در بخش بحث شده است.  [4. 5](#bookmark222) حاوی یک پیش‌بینی بر اساس مقدار پول در حساب پس انداز متقاضی است. این داده‌ها بودند

جدول 3. 2: یک پیش‌بینی طبقه‌بندی شده با پنج گروه مجزا از مطالعه موردی امتیازدهی اعتبار. مقادیر، مقدار موجود در حساب پس انداز (به دویچه مارک) است.

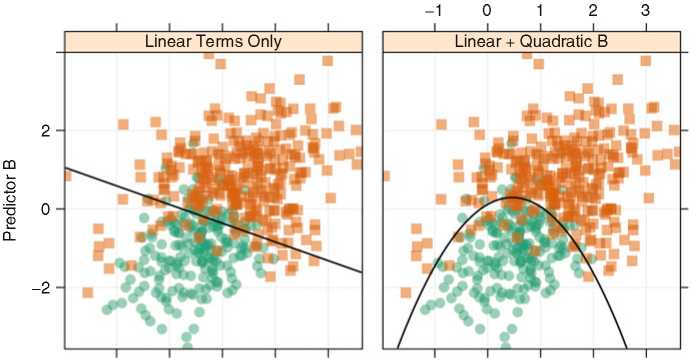
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| متغیرهای ساختگی | | | | | | |
| مقدار | *n* | *<* 100 | 100-500 500- | 1000 *>* 1000 *\_* \_ | | ناشناس |
| *<* 100 DM | 103 | 1 | 0 | 0 | 0 | 0 |
| 100-500 DM | 603 | 0 | 1 | 0 | 0 | 0 |
| 500-1000 DM | 48 | 0 | 0 | 1 | 0 | 0 |
| *>* 1000 DM | 63 | 0 | 0 | 0 | 1 | 0 |
| ناشناس | 183 | 0 | 0 | 0 | 0 | 1 |

به چندین گروه رمزگذاری شده است، از جمله گروهی برای "ناشناخته". جدول [3. 2](#bookmark170) مقادیر این پیش‌بینی و تعداد متقاضیانی که در هر سطل قرار می‌گیرند را نشان می‌دهد.

برای استفاده از این داده‌ها در مدل‌ها، دسته‌ها دوباره در بیت‌های کوچک‌تری از اطلاعات به نام «متغیرهای ساختگی» کدگذاری می‌شوند. معمولاً هر دسته متغیر ساختگی خود را دارد که یک شاخص صفر/یک برای آن گروه است. جدول [3. 2](#bookmark170) متغیرهای ساختگی احتمالی را برای این داده‌ها نشان می‌دهد. فقط چهار متغیر ساختگی در اینجا مورد نیاز است. هنگامی که مقدار چهار متغیر ساختگی را بدانید، می‌توان پنجمی را استنباط کرد. **با این حال، تصمیم برای گنجاندن همه متغیرهای ساختگی می‌تواند به انتخاب مدل بستگی داشته باشد.** مدل‌هایی که شامل یک عبارت عرض از مبدأ هستند، مانند رگرسیون خطی ساده (بخش.  [6. 2 )](#bookmark286) ، اگر هر متغیر ساختگی در مدل گنجانده شود، مسائل عددی خواهد داشت. دلیل آن این است که برای هر نمونه، همه این متغیرها به یک عدد می‌رسند و این اطلاعات همان عرض از مبدأ را ارائه می‌دهد. **اگر مدل نسبت به این نوع مسائل حساس نباشد، استفاده از مجموعه کامل متغیرهای ساختگی به بهبود تفسیر مدل کمک می‌کند.**

بسیاری از مدل‌های توصیف شده در این متن به‌طور خودکار روابط بسیار پیچیده و غیرخطی بین متغیرهای پیش‌بینی و نتیجه ایجاد می‌کنند. مدل‌های ساده‌تر این کار را نمی‌کنند مگر اینکه کاربر به‌طور دستی مشخص کند که کدام متغیرهای پیش‌بینی باید غیرخطی و به چه شکلی باشند. به‌عنوان مثال، رگرسیون لجستیک یک مدل طبقه‌بندی شناخته شده است که به‌طور پیش فرض، ­مرزهای طبقه‌بندی خطی را ایجاد می‌کند. **شکل** [**3. 11**](#bookmark173) **مثال گویا دیگری را با دو پیش‌بینی و دو کلاس نشان می‌دهد. پانل سمت چپ ­مرزهای طبقه‌بندی رگرسیون لجستیک پایه را هنگامی که پیش‌بینی‌کننده‌ها به روش معمول (خطی) اضافه می‌شوند نشان می‌دهد. پانل سمت راست یک مدل لجستیک با عبارت‌های خطی پایه و یک جمله اضافی با مربع پیش‌بینی *B را* نشان می‌دهد.** از آنجایی که رگرسیون لجستیک یک مدل به خوبی مشخص و پایدار است، استفاده از این مدل با برخی اصطلاحات غیرخطی اضافی ممکن است به تکنیک‌های بسیار پیچیده (که ممکن است برازش بیش از حد ایجاد کند) ترجیح داده شود.

**علاوه بر این،** [**فورینا و همکاران ( 2009**](#bookmark1016) **) یک تکنیک را برای تقویت ­داده‌های پیش‌بینی با افزودن ترکیب‌های پیچیده داده‌ها نشان می‌دهد. برای مدل‌های طبقه‌بندی، «مرکزهای کلاس» را محاسبه می‌کنند که مراکز داده‌های پیش‌بینی برای هر کلاس هستند. سپس برای هر متغیر پیش‌بینی، می‌توان ­فاصله هر مرکز کلاس را محاسبه کرد و این فاصله‌ها را به مدل اضافه کرد.**



Class 1

Class 2

-1

01

23

Predictor A

Fig. 3.11: Classification boundaries from two logistic regression models. The *left panel* has linear terms for the two predictors while the *right panel* has an additional quadratic term for predictor *B* . This model is discussed in more detail in Chap. [12](#bookmark550)

Binning Predictors

در حالی که تکنیک‌های توصیه‌شده برای پیش‌پردازش (شاید منظورش پیچیده تر کردن دیتا) داده‌ها وجود دارد، روش‌هایی نیز برای *جلوگیری از آن*(شاید منظورش ساده تر کردن دیتا)  *وجود دارد.* یکی از روش‌های رایج برای ساده‌سازی مجموعه داده، استفاده از یک پیش‌بینی عددی و دسته‌بندی یا «بین یا دسته بندی » کردن آن به دو یا چند گروه قبل از تحلیل داده‌ها است. مثلا، [استخوان و همکاران ( 1992](#bookmark1011) ) مجموعه‌ای از علائم بالینی را برای تشخیص سندرم پاسخ التهابی سامانه‌یک (SIRS) تعریف می‌کند. SIRS ممکن است پس از اینکه فرد تحت نوعی ضربه فیزیکی قرار می‌گیرد (مثلاً تصادف اتومبیل) رخ می‌دهد. نسخه ساده شده معیارهای بالینی SIRS عبارتند از:

* دمای کمتر از 36 *◦* C یا بیشتر از 38 *◦* C.
* ضربان قلب بیش از 90 ضربه در دقیقه.
* تعداد تنفس بیش از 20 تنفس در دقیقه.
* تعداد گلبول‌های سفید کمتر از 4000 سلول در میلی متر 3 یا بیشتر از 12000 سلول در میلی متر 3.

فردی که دو یا چند مورد از این معیارها را نشان دهد، مبتلا به SIRS تشخیص داده می‌شود.

مزایای درک شده برای این رویکرد عبارتند از:

* توانایی بیان جملات به ظاهر ساده، یا به خاطر داشتن یک قانون تصمیم‌گیری ساده (مانند مثال SIRS) و یا اعتقاد به اینکه یک تفسیر ساده از مدل وجود خواهد داشت.
* مدل ساز مجبور نیست رابطه دقیق بین پیش‌بینی‌کننده‌ها و نتیجه را بداند.
* نرخ پاسخ بالاتر برای سؤالات نظرسنجی که در آن گزینه‌ها درج شده است. به‌عنوان مثال، پرسیدن تاریخ آخرین واکسن کزاز از یک فرد احتمالاً پاسخ‌های کمتری نسبت به درخواست دامنه دارد (مثلاً در 2 سال گذشته، در 4 سال گذشته).

موضوعات زیادی در باینینگ دستی داده‌های پیوسته وجود دارد. اول، ممکن است کاهش قابل‌توجهی در عملکرد در مدل وجود داشته باشد. بسیاری از ­تکنیک‌های مدل‌سازی مورد بحث در این متن در تعیین روابط پیچیده بین پیش‌بینی‌کننده‌ها و نتایج بسیار خوب هستند. باینینگ دستی پیش‌بینی‌کننده‌ها این پتانسیل را محدود می‌کند. دوم، وقتی پیش‌بینی‌کننده‌ها دسته‌بندی می‌شوند، دقت در پیش‌بینی‌کننده‌ها از دست می‌رود. ­به‌عنوان مثال، اگر دو متغیر پیش‌بینی binned وجود داشته باشد، تنها چهار ترکیب در مجموعه داده وجود دارد، بنابراین فقط ­پیش‌بینی‌های ساده را می‌توان انجام داد. سوم، تحقیقات نشان داده است ( Austin and Brunner 2004 ) که طبقه‌بندی پیش‌بینی‌کننده‌ها می‌تواند منجر به نرخ بالای مثبت کاذب شود (یعنی پیش‌بینی‌کننده‌های نویز که حاوی اطلاعات هستند مشخص می‌شود).

**متأسفانه، مدل‌های پیش‌بینی که قوی‌ترین هستند، معمولاً کمتر قابل تفسیر هستند. نتیجه نهایی این است که بهبود درک شده در تفسیرپذیری که با دسته‌بندی دستی به دست می‌آید معمولاً با ­کاهش قابل‌توجهی در عملکرد جبران می‌شود**. **از آنجایی که این کتاب به مدل‌های پیش‌بینی می‌پردازد (که در آن تفسیر هدف اصلی نیست)، باید از از دست دادن عملکرد اجتناب شود. در واقع، در برخی موارد ممکن است دسته‌بندی خودسرانه پیش‌بینی‌کننده‌ها غیراخلاقی (آن‌اِتیکال) باشد. برای مثال، تحقیقات زیادی در مورد پیش‌بینی جنبه‌های بیماری (به‌عنوان مثال، پاسخ به درمان، غربالگری بیماران) وجود دارد. اگر ­برای چنین تشخیص‌های مهمی از یک تشخیص متخصصی استفاده شود، بیماران به دنبال دقیق‌ترین پیش‌بینی ممکن هستند. تا زمانی که مدل‌های پیچیده به درستی اعتبار سنجی شده باشند، ممکن است استفاده از مدلی که برای تفسیر به جای عملکرد پیش‌بینی ساخته شده است، نامناسب باشد.**

**توجه داشته باشید که استدلال در اینجا به دسته‌بندی *دستی* پیش‌بینی‌کننده‌ها قبل از ساخت مدل مربوط می‌شود. چندین مدل مانند درختان طبقه‌بندی/رگرسیون و خطوط رگرسیون تطبیقی چند متغیره وجود دارد که نقاط برش را در فرآیند ساخت مدل تخمین می‌زنند. تفاوت بین این روش‌شناسی و باینینگ دستی در این است که مدل‌ها از همه پیش‌بینیکننده‌ها ­برای استخراج bin‌ها بر اساس یک هدف واحد (مانند حداکثر کردن دقت) استفاده می‌کنند. آنها بسیاری از متغیرها را به‌طور همزمان ارزیابی می‌کنند و معمولاً بر اساس روش‌های آماری معتبر هستند.**

محاسبه

این بخش از داده‌های بسته AppliedPredictiveModeling و توابع از بسته‌های caret، corrplot، e1071 و lattice استفاده می‌کند.

دو مکان وجود دارد که کد R مربوطه را می‌توان یافت:

فهرست فصل‌های بسته AppliedPredictiveModeling حاوی کد خاصی برای بازتولید مدل‌های خاص مورد استفاده در فصل است. این در نظر گرفته شده است تا به خواننده اجازه دهد تا ببیند مدل‌های مورد استفاده در اینجا دقیقا چگونه ایجاد شده اند.

بسیاری از فصل‌های این کتاب شامل بخش‌هایی در انتهای فصل است که نحوه انجام محاسبات را به‌طور کلی در R توضیح می‌دهد. به‌عنوان مثال، توابع جداگانه‌ای وجود دارد که با روش‌های پیش ­پردازش داده نشان داده شده در این فصل مطابقت دارد. در حالی که بخش محاسبات این جزئیات را ارائه می‌دهد، توابع فردی ممکن است به‌طور مستقیم در عمل مورد استفاده قرار نگیرند. به‌عنوان مثال، هنگام استفاده از تابع آموزش، مراحل پیش پردازش در یک آرگومان مشخص می‌شود و از توابع جداگانه استفاده نمی‌شود. این بخش‌ها به مدل‌های ایجاد شده در هر فصل مربوط می‌شوند، اما به‌عنوان نکات بحث برای توابع.

به این ترتیب، بخش‌های Computing در هر فصل نحوه انجام محاسبات را به‌طور کلی توضیح می‌دهد در حالی که کد موجود در فهرست فصل‌های بسته AppliedPredictiveModeling بهترین منبع برای محاسبات مدل‌های خاص در هر فصل است.

همانطور که در ضمیمه B بحث شد، چند تابع مفید R وجود دارد که می‌توان از آنها برای یافتن توابع یا کلاس‌های مورد علاقه موجود استفاده کرد. تابع apropos هر بسته R بارگذاری شده را برای یک عبارت مشخص جستجو می‌کند. به‌عنوان مثال، برای یافتن توابعی برای ایجاد یک ماتریس سردرگمی در بسته‌های بارگذاری شده فعلی:

*> apropos ("گیج")*

[1] "confusionMatrix" "confusionMatrix. train"

برای یافتن چنین تابعی در هر بسته، تابع RSiteSearch می‌تواند کمک کند. اجرای دستور:

*RSiteSearch ("سرگردانی"، محدود = "توابع")*

برای یافتن مسابقات به صورت آنلاین جستجو می‌کند و یک مرورگر وب را برای نمایش نتایج باز می‌کند.

مجموعه داده‌های تقسیم‌بندی خام در بسته AppliedPredictiveModeling موجود است. [[7]](#footnote-7) برای بارگذاری مجموعه داده در R :

*کتابخانه (AppliedPredictiveModeling)*

*داده (SegmentationOriginal)*

فیلدهایی وجود داشت که هر سلول را شناسایی می‌کرد (به نام سلول ) و یک بردار فاکتور که نشان می‌داد کدام سلول‌ها به خوبی تقسیم شده اند ( کلاس ). متغیر Case نشان می‌دهد که کدام سلول‌ها در ابتدا برای مجموعه‌های تمرین و آزمایش استفاده شده‌اند. تحلیل در این فصل بر روی نمونه‌های مجموعه آموزشی متمرکز شده است، بنابراین داده‌ها برای این سلول‌ها فیلتر می‌شوند:

*> segData <- زیر مجموعه (segmentationOriginal, Case == "Train")*

Class و Cell در بردارهای جداگانه ذخیره می‌شوند و سپس از شی اصلی حذف می‌شوند:

*cellID <- segData$Cell*

*کلاس <- segData$Class*

*مورد <- segData$Case*

*# حالا ستون‌ها را حذف کنید*

*segData <- segData[, -(1:3)]*

داده‌های اصلی حاوی چندین ستون "وضعیت" بودند که نسخه‌های باینری ­پیش‌بینی‌کننده‌ها بودند. برای حذف این موارد، نام ستون‌های حاوی "وضعیت" را پیدا کرده و آنها را حذف می‌کنیم:

*statusColNum <- grep("وضعیت"، نامها(segData))*

*statusConum*

[1] 2 4 9 10 11 12 14 16 20 21 22 26 27 28 30 32 34

36 38 40 43 44 46 48 51 52 55 56 59 60 63 64 68 69

70 72 73 74 76 78 80 82 84 86 88 92 93 94 97 98 103

104 105 106 110 111 112 114

*> segData <- segData[, -statusColNum]*

تحولات

همانطور که قبلاً بحث شد، برخی از ویژگی‌ها چولگی قابل‌توجهی را نشان دادند. تابع چولگی در بسته e1071 آمار چولگی نمونه را ­برای هر پیش‌بینی محاسبه می‌کند:

*کتابخانه (e1071)*

*# برای یک پیش‌بینی:*

*چولگی (segData$AngleCh1)*

[1] -0. 0243

*# از آنجایی که همه پیش‌بینی‌کننده‌ها ستون‌های عددی هستند، تابع application می‌تواند*

*# برای محاسبه چولگی ستون‌ها استفاده شود.*

*skewValues <- اعمال (segData، 2، چولگی)*

*سر (ارزشهای کج)*

AngleCh1 AreaCh1 AvgIntenCh1 AvgIntenCh2 AvgIntenCh3 AvgIntenCh4

-0. 0243 3. 5251 2. 9592 0. 8482 2. 2023 1. 9005

با استفاده از این مقادیر به‌عنوان راهنما، می‌توان متغیرها را برای تجسم توزیع اولویت‌بندی کرد. برای ارزیابی شکل توزیع، می‌توان از تابع اصلی R Hist یا تابع هیستوگرام در شبکه استفاده کرد.

برای تعیین اینکه کدام نوع تبدیل باید استفاده شود، بسته MASS حاوی تابع boxcox است. اگرچه این تابع *A را تخمین می‌زند،* اما متغیر(های) تبدیل شده را ایجاد نمی‌کند. یک تابع caret، BoxCoxTrans، می‌تواند تبدیل مناسب را پیدا کرده و آنها را در داده‌های جدید اعمال کند:

*کتابخانه (کارت)*

*Ch1AreaTrans <- BoxCoxTrans(segData$AreaCh1)*

*Ch1AreaTrans*

تبدیل جعبه-کاکس

1009 نقطه داده برای تخمین لامبدا استفاده شد

خلاصه داده‌های ورودی:

حداقل 1 ق. میانگین میانه 3 ق. حداکثر

150 194 256 325 376 2190

بزرگترین/کوچکترین: 14. 6

چولگی نمونه: 3. 53

لامبدا تخمینی: -0. 9

*# داده‌های اصلی*

*head(segData$AreaCh1)*

[1] 819 431 298 256 258 358

*# پس از تحول*

*پیش‌بینی (Ch1AreaTrans، head(segData$AreaCh1))*

[1] 1. 1085 1. 1064 1. 1045 1. 1036 1. 1036 1. 1055

*> (819~(-. 9) - 1)/(-. 9)*

[1] 1. 1085

تابع Caret دیگر، preProcess، این تبدیل را برای مجموعه‌ای از پیش‌بینی‌کننده‌ها اعمال می‌کند. این تابع در زیر مورد بحث قرار گرفته است. تابع R پایه prcomp را می‌توان برای PCA استفاده کرد. در کد زیر، داده‌ها قبل از PCA مرکز و مقیاس‌بندی شده اند.

*pcaObject <- prcomp(segData,*

*+ مرکز = درست، مقیاس. = درست)*

*# درصد تجمعی واریانسی را که هر جزء > # محاسبه می‌کند، محاسبه کنید.*

*درصد واریانس <- pca0bject$sd~2/sum(pca0bject$sd~2)\*100*

*درصد واریانس [1:3]*

20. 9 17. 0 11. 9

مقادیر تبدیل شده در pcaObject به‌عنوان یک شی فرعی به نام x ذخیره می‌شوند :

*> head(pcaObject$x[, 1:5])*

PC1 PC2 PC3 PC4 PC5

2 5. 099 4. 551 -0. 0335 -2. 64 1. 278

3 -0. 255 1. 198 -1. 0206 -3. 73 0. 999

4 1. 293 -1. 864 -1. 2511 -2. 41 -1. 491

12 -1. 465 -1. 566 0. 4696 -3. 39 -0. 330

15 -0. 876 -1. 279 -1. 3379 -3. 52 0. 394

16 -0. 862 -0. 329 -0. 1555 -2. 21 1. 473

شیء فرعی دیگری به نام چرخش بارهای متغیر را ذخیره می‌کند، جایی که ردیف‌ها با متغیرهای پیش‌بینی مطابقت دارند و ستون‌ها با مؤلفه‌ها مرتبط هستند:

*head(pcaObject$rotation[, 1:3])*

PC1 PC2 PC3

AngleCh1 0. 00121 -0. 0128 0. 00682

AreaCh1 0. 22917 0. 1606 0. 08981

AvgIntenCh1 -0. 10271 0. 1797 0. 06770

AvgIntenCh2 -0. 15483 0. 1638 0. 07353

AvgIntenCh3 -0. 05804 0. 1120 -0. 18547

AvgIntenCh4 -0. 11734 0. 2104 -0. 10506

کلاس بسته caret spatialSign دارای قابلیتی برای تبدیل علامت فضایی است. اگر چه ما این تکنیک را برای این داده‌ها اعمال نمی‌کنیم، نحو اصلی آن spatialSign(segData) خواهد بود.

همچنین، این داده‌ها مقادیر گمشده‌ای برای انتساب ندارند. برای تعیین مقادیر گمشده، بسته impute دارای تابع impute. knn است که از *K* - نزدیکترین همسایه‌ها برای تخمین داده‌های از دست رفته استفاده می‌کند. تابع preProcess که قبلاً ذکر شد، روش‌های انتساب را براساس *K* -نزدیک‌ترین همسایه‌ها ­یا درختان کیسه‌دار اعمال می‌کند.

برای اجرای یک سری تبدیل به مجموعه داده‌های چندگانه، کلاس caret preProcess توانایی تبدیل، مرکز، مقیاس کردن، یا مقادیر را دارد و همچنین تبدیل علامت فضایی و استخراج ویژگی را اعمال می‌کند. تابع مقادیر مورد نیاز برای تبدیل را محاسبه می‌کند. پس از ­فراخوانی تابع preProcess، روش پیش‌بینی نتایج را روی مجموعه‌ای از داده‌ها اعمال می‌کند. به‌عنوان مثال، برای تبدیل Box-Cox، مرکز و مقیاس‌بندی داده‌ها، سپس اجرای PCA برای استخراج سیگنال، نحو به صورت زیر خواهد بود:

*trans <- preProcess(segData,*

*+ روش = c("BoxCox"، "center"، "scale"، "pca"))*

*ترانس*

صدا زدن:

preProcess. default(x = segData، روش = c("BoxCox"، "center"، "scale"، "pca"))

ایجاد شده از 1009 نمونه و 58 متغیر

پیش پردازش: تبدیل Box-Cox، مرکز، مقیاس شده، استخراج سیگنال جزء اصلی

تخمین لامبدا برای تبدیل باکس-کاکس:

حداقل 1 ق. میانگین میانه 3 ق. حداکثر NA

-2. 00 -0. 50 -0. 10 0. 05 0. 30 2. 00 11

PCA برای ثبت 95 درصد واریانس به 19 جزء نیاز داشت

*# تبدیل‌ها را اعمال کنید:*

*تبدیل شده <- پیش‌بینی (trans, segData)*

*# این مقادیر متفاوت از اجزای قبلی PCA هستند*

*# آنها قبل از PCA > head (تبدیل شده[, 1:5]) تبدیل شدند.*

PC1 PC2 PC3 PC4 PC5 2 1. 568 6. 291 -0. 333 -3. 06 -1. 342 3 -0. 666 2. 046 -1. 442 -4. 70 -1. 742 -10 -16 -0. 392 -0. 669 -4. 02 1. 793 12 0. 377 -2. 190 1. 438 -190 1. 438 -0. 865 16 -0. 380 0. 217 0. 439 -2. 07 -1. 936

ترتیبی که در آن تبدیل ممکن اعمال می‌شود تبدیل، مرکزیت، مقیاس بندی، انتساب، استخراج ویژگی و سپس علامت فضایی است.

بسیاری از توابع مدل‌سازی دارای گزینه‌هایی برای مرکز و مقیاس‌بندی قبل از مدل‌سازی هستند. به‌عنوان مثال، هنگام استفاده از تابع آموزش (که در فصل‌های بعدی بحث شد)، گزینه‌ای برای استفاده از preProcess قبل از مدل‌سازی در تکرارهای نمونه‌برداری مجدد وجود دارد.

فیلتر کردن

برای فیلتر کردن پیش‌بینی‌کننده‌های واریانس نزدیک به صفر، تابع بسته caret در نزدیکی Zero Var، شماره ستون‌های هر پیش‌بینی‌ای را که شرایط ذکر شده در بخش را برآورده می‌کنند، برمی‌گرداند.  [3. 5](#bookmark161) . برای داده‌های تقسیم‌بندی سلولی، هیچ پیش‌بینی مسأله‌داری وجود ندارد:

*nearZeroVar(segData)*

عدد صحیح (0)

*# وقتی پیش‌بینی‌کننده‌ها باید حذف شوند، بردار اعداد صحیح > # برگردانده می‌شود که نشان می‌دهد کدام ستون‌ها باید حذف شوند.*

به‌طور مشابه، برای فیلتر کردن همبستگی‌های بین پیش‌بینی، تابع cor می‌تواند همبستگی‌های بین متغیرهای پیش‌بین را محاسبه کند:

*> همبستگی‌ها <- cor(segData)*

*> کم نور (همبستگی ها)*

[1] 58 58

*> همبستگی‌ها [1:4، 1:4]*

|  |  |  |  |
| --- | --- | --- | --- |
|  | AngleCh1 AreaCh1 AvgIntenCh1 AvgIntenCh2 | | |
| AngleCh1 | 1. 00000 -0. 00263 | -0. 0430 | -0. 0194 |
| منطقه Ch1 | -0. 00263 1. 00000 | -0. 0253 | -0. 1533 |
| AvgIntenCh1 | -0. 04301 -0. 02530 | 1. 0000 | 0. 5252 |
| AvgIntenCh2 | -0. 01945 -0. 15330 | 0. 5252 | 1. 0000 |

برای بررسی بصری ساختار همبستگی داده‌ها، بسته ­corrplot دارای یک تابع عالی به همین نام است. این تابع گزینه‌های زیادی دارد، از جمله یکی که متغیرها را به گونه‌ای مرتب می‌کند که خوشه‌هایی از پیش‌بینی‌کننده‌های بسیار همبسته را نشان می‌دهد. برای تولید شکل زیر از دستور زیر استفاده شده است.  [3. 10 :](#bookmark166)

*کتابخانه (corrplot)*

*corrplot (همبستگی، نظم = "hclust")*

اندازه و رنگ نقاط با قدرت همبستگی بین دو متغیر پیش‌بینی مرتبط است.

برای فیلتر بر اساس همبستگی‌ها، تابع findCorrelation الگوریتم را در بخش اعمال می‌کند.  [3. 5 .](#bookmark165) برای یک آستانه معین از همبستگی‌های زوجی، تابع ­اعداد ستون را نشان می‌دهد که نشان دهنده پیش‌بینی‌کننده‌هایی است که برای حذف توصیه می‌شود:

*highCorr <- findCorrelation (همبستگی، برش = 0. 75)*

*طول (میزان زیاد)*

[1] 33

*> سر (HighCorr)*

[1] 23 40 43 36 7 15

*filteredSegData <- segData[, -highCorr]*

همچنین چندین تابع در بسته فرعی وجود دارد که می‌تواند یک هدف را انجام دهد.

**ایجاد متغیرهای ساختگی**

چندین روش برای ایجاد متغیرهای ساختگی بر اساس یک مدل خاص وجود دارد. بخش [4. 9](#bookmark245) روش‌های مختلفی را برای تعیین نحوه ­ورود عوامل پیش‌بینی به مدل مورد بحث قرار می‌دهد. یک رویکرد، روش فرمول، به انعطاف پذیری زیادی برای ایجاد تابع مدل اجازه می‌دهد. استفاده از فرمول‌ها در توابع مدل، پیش‌بینی‌کننده‌ها را به گونه‌ای اندازه‌گیری می‌کند که همه دسته‌ها دارای متغیرهای ساختگی نیستند. این رویکرد برای رگرسیون خطی با جزئیات بیشتری نشان داده خواهد شد.

همانطور که قبلا ذکر شد، مواردی وجود دارد که مجموعه کاملی از متغیرهای ساختگی مفید است. به‌عنوان مثال، تقسیمات در یک مدل مبتنی بر درخت زمانی قابل تفسیر هستند که متغیرهای ساختگی تمام اطلاعات آن پیش‌بینی را رمزگذاری می‌کنند. توصیه می‌کنیم هنگام کار با مدل‌های مبتنی بر درخت، از متغیرهای ساختگی if استفاده کنید.

برای نشان دادن کد، زیرمجموعه‌ای از مجموعه داده‌های خودروها را در بسته کارت می‌گیریم. برای سال 2005، اطلاعات فروش مجدد کتاب Kelly Blue Book برای 804 اتومبیل GM جمع‌آوری شد [( کویپر 2008](#bookmark1020) ). هدف این مدل پیش‌بینی قیمت خودرو بر اساس ویژگی‌های شناخته شده بود. این نمایش بر روی قیمت، مسافت پیموده شده و نوع خودرو (به‌عنوان مثال، سدان) برای زیر مجموعه‌ای از وسایل‌نقلیه تمرکز خواهد کرد:

*سر (زیر مجموعه ماشین)*

قیمت مسافت پیموده شده نوع 214 19981 24323 سدان 299 21757 1853 سدان 460 15047 12305 سدان

728 15327 4318 سدان

162 20628 20770 سدان

718 16714 26328 سدان

*> سطوح (carSubset$Type)*

[1] "convertible" "coupe"

"hatchback" "sedan"

"wagon"

برای مدل‌سازی قیمت به‌عنوان تابعی از مسافت پیموده شده و نوع خودرو، می‌توانیم از تابع dummyVars برای تعیین کدهای پیش‌بینی استفاده کنیم. فرض کنید مدل اول ما فرض می‌کند که قیمت را می‌توان به‌عنوان یک تابع افزودنی ساده از مسافت پیموده شده و نوع مدل‌سازی کرد:

*> simpleMod <- dummyVars (~ مسافت پیموده شده + نوع،*

|  |  |
| --- | --- |
| *+*  *+* | *داده = ماشین زیر مجموعه،*  *## نام متغیر را از قسمت حذف کنید* |
| *+*  *+*  *> simpleMod*  شی متغیر ساختگی | *## فقط سطوح نام ستون = TRUE)* |

فرمول: ~ مسافت پیموده شده + متغیرهای نوع 2، 1 عامل

نام متغیرهای فاکتور حذف خواهد شد

برای تولید متغیرهای ساختگی برای مجموعه آموزشی یا هر نمونه جدید، از روش پیش‌بینی همراه با موضوع dummyVars استفاده می‌شود :

*> پیش‌بینی (simpleMod، head(carSubset))*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | مسافت پیموده شده | قابل تبدیل | کوپه | هاچ بک | سدان | واگن |
| 214 | 24323 | 0 | 0 | 0 | 1 | 0 |
| 299 | 1853 | 0 | 0 | 0 | 1 | 0 |
| 460 | 12305 | 0 | 0 | 0 | 1 | 0 |
| 728 | 4318 | 0 | 0 | 0 | 1 | 0 |
| 162 | 20770 | 0 | 0 | 0 | 1 | 0 |
| 718 | 26328 | 0 | 0 | 0 | 1 | 0 |

فیلد نوع به پنج متغیر برای پنج سطح عامل گسترش یافت. مدل ساده است زیرا فرض می‌کند که اثر مسافت پیموده شده برای هر نوع خودرو یکسان است. برای تطبیق یک مدل پیشرفته تر، می‌توانیم فرض کنیم که اثر *مشترک* مسافت پیموده شده و نوع خودرو وجود دارد. این نوع اثر به‌عنوان یک تعامل شناخته می‌شود. در فرمول مدل، یک کولون بین فاکتورها نشان می‌دهد که باید یک تعامل ایجاد شود. برای این داده‌ها، این پنج پیش‌بینی دیگر به چارچوب داده اضافه می‌کند:

*withInteraction <- dummyVars(~ مسافت پیموده شده + نوع + مسافت پیموده شده: نوع،*

*+ داده = زیر مجموعه خودرو،*

*+ سطوح فقط = درست)*

*با تعامل*

شی متغیر ساختگی

فرمول: ~ مسافت پیموده شده + نوع + مسافت پیموده شده: متغیرهای نوع 2، 1 فاکتور نام متغیرهای فاکتور حذف خواهد شد

*پیش‌بینی (با تعامل، سر (مجموعه ماشین))*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | مسافت پیموده شده | قابل تبدیل | کوپه | هاچ بک | سدان | مسافت پیموده شده واگن: | قابل تبدیل |
| 214 | 24323 | 0 | 0 | 0 | 1 | 0 | 0 |
| 299 | 1853 | 0 | 0 | 0 | 1 | 0 | 0 |
| 460 | 12305 | 0 | 0 | 0 | 1 | 0 | 0 |
| 728 | 4318 | 0 | 0 | 0 | 1 | 0 | 0 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 162 | 20770 | 0 | 0 | 010 | 0 |
| 718 | 26328 | 0 | 0 | 010 | 0 |
|  | مسافت پیموده شده:کوپه مسافت پیموده شده:هاچ بک مسافت پیموده شده:سدان مسافت پیموده شده:واگن | | | | |
| 214 |  | 0 |  | 0 24323 | 0 |
| 299 |  | 0 |  | 0 1853 | 0 |
| 460 |  | 0 |  | 0 12305 | 0 |
| 728 |  | 0 |  | 0 4318 | 0 |
| 162 |  | 0 |  | 0 20770 | 0 |
| 718 |  | 0 |  | 0 26328 | 0 |

تمرینات

مخزن یادگیری ماشین UC Irvine[[8]](#footnote-8) شامل مجموعه‌ای از داده‌های مربوط به شناسایی شیشه است. داده‌ها شامل 214 نمونه شیشه‌ای است که به‌عنوان یکی از هفت دسته طبقه‌بندی شده است. 9 پیش‌بینی شامل ضریب شکست و درصد هشت عنصر Na، Mg، Al، Si، K، Ca، Ba و Fe وجود دارد.

دسترسی به داده‌ها از طریق:

*> کتابخانه (mlbench)*

*> داده (شیشه ای)*

*> خیابان (شیشه ای)*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| data. frame': | | | 214 obs. از 10 متغیر: | | | |
| $ | RI | : تعداد | 1. 52 | 1. 52 | 1. 52 1. 52 1. 52. . . |  |
| $ | Na | : تعداد | 13. 6 | 13. 9 | 13. 5 13. 2 13. 3. . . |  |
| $ | Mg | : تعداد | 4. 49 | 3. 6 3 | . 55 3. 69 3. 62 3. 61 3. 6 3. 61 3 | . 58 3. 6. . . |
| $ | ال | : تعداد | 1. 1 1 | . 36 1 | 0. 54 1. 29 1. 24 1. 62 1. 14 1. 05 | 1. 37 1. 36. . . |
| $ | سی | : تعداد | 71. 8 | 72. 7 | 73 72. 6 73. 1. . . |  |
| $ | ک | : تعداد | 0. 06 | 0. 48 | 0. 39 0. 57 0. 55 0. 64 0. 58 0. 57 | 0. 56 0. 57. . . |
| $ | حدود | : تعداد | 8. 75 | 7. 83 | 7. 78 8. 22 8. 07 8. 07 8. 17 8. 24 | 8. 3 8. 4. . . |
| $ | با | : تعداد | 000 | 00 | 00000. . . |  |
| $ | Fe | : تعداد | 000 | 00 | 0. 26 0 0 0 0. 11. . . |  |
| $ | تایپ کنید | فاکتور w/ | | 6 سطح "1"، "2"، "3"، "5"،. . . : 1 1 1 | | 1111111 |

با استفاده از تجسم‌ها، متغیرهای پیش‌بینی را برای درک توزیع آنها و همچنین روابط بین پیش‌بینی‌کننده‌ها بررسی کنید.

آیا به نظر می‌رسد که در داده‌ها موارد پرت وجود دارد؟ آیا هیچ پیش‌بینی‌ای منحرف است؟

آیا تغییرات مرتبطی در یک یا چند پیش‌بینی وجود دارد که ممکن است مدل طبقه‌بندی را بهبود بخشد؟

داده‌های دانه سویا را می‌توان در مخزن یادگیری ماشین UC Irvine نیز یافت. داده‌ها برای پیش‌بینی بیماری در 683 دانه سویا جمع‌آوری شد. 35 پیش‌بینی عمدتاً طبقه‌بندی هستند و شامل اطلاعاتی در مورد ­شرایط محیطی (به‌عنوان مثال، دما، بارش) و شرایط گیاهی (به‌عنوان مثال، لکه‌های سمت چپ، رشد کپک‌ها) هستند. برچسب‌های نتیجه از 19 کلاس مجزا تشکیل شده‌اند.

داده‌ها را می‌توان از طریق:

*> کتابخانه (mlbench)*

*> داده (سویا)*

*> ## برای جزئیات بیشتر به سویا مراجعه کنید*

توزیع فرکانس را برای پیش‌بینی‌کننده‌های طبقه‌بندی بررسی کنید. آیا هیچ یک از توزیع‌ها به روشی که قبلاً در این فصل مورد بحث قرار گرفت، منحط شده است؟

تقریباً 18 درصد از داده‌ها از دست رفته است. آیا پیش‌بینی‌کننده‌های خاصی وجود دارد که احتمال گم شدن آنها بیشتر است؟ آیا الگوی داده‌های از دست رفته مربوط به کلاس‌ها است؟

یک استراتژی برای مدیریت داده‌های از دست رفته، با حذف پیش‌بینی‌کننده‌ها یا انتساب ایجاد کنید.

فصل 5 مدل‌سازی کمی ساختار-فعالیت (QSAR) را معرفی می‌کند که در آن ویژگی‌های یک ترکیب شیمیایی برای پیش‌بینی سایر خواص شیمیایی استفاده می‌شود. بسته caret شامل یک مجموعه داده QSAR از [منته و لومباردو](#bookmark1022) [( 2005](#bookmark1022) ). در اینجا، توانایی یک ماده شیمیایی برای نفوذ به سد خونی مغزی به‌طور تجربی برای 208 ترکیب تعیین شد. برای هر ترکیب 134 توصیفگر اندازه‌گیری شد.

R را راه اندازی کنید و از این دستورات برای بارگیری داده‌ها استفاده کنید:

*> کتابخانه (کارت)*

*> داده (BloodBrain)*

*> # برای مشاهده جزئیات بیشتر از ?BloodBrain استفاده کنید*

برآیند عددی در بردار logBBB وجود دارد در حالی که پیش‌بینی‌کننده‌ها در ­قاب داده bbbDescr هستند.

آیا هیچ یک از پیش‌بینی‌کننده‌های فردی توزیع‌های منحط دارند؟

به‌طور کلی، آیا روابط قوی بین ­داده‌های پیش‌بینی وجود دارد؟ اگر چنین است، چگونه می‌توان همبستگی‌ها را در مجموعه پیش‌بینی کاهش داد؟ آیا این تأثیر چشمگیری بر تعداد پیش‌بینی‌کننده‌های موجود برای مدل‌سازی دارد؟

فصل 4

بیش برازش و تنظیم مدل

بسیاری از مدل‌های طبقه‌بندی و رگرسیون مدرن بسیار سازگار هستند. آنها قادر به مدل‌سازی روابط پیچیده هستند. با این حال، آنها می‌توانند به راحتی بر الگوهایی که تکرارپذیر نیستند تأکید کنند. بدون رویکرد روش‌شناختی برای ارزیابی مدل‌ها، مدل‌ساز تا زمانی که مجموعه بعدی نمونه‌ها پیش‌بینی نشود، از مسأله مطلع نخواهد شد.

بیش برازشی در زمینه‌های پیش‌بینی [( کلارک 2004](#bookmark1013) )، تحقیقات متخصصی [( سایمون و همکاران 2003 ) مورد بحث قرار گرفته است](#bookmark1025) .  [Steyerberg 2010](#bookmark1025) [)،](#bookmark1017) شیمی سنجی ( Gowen et al. [2010](#bookmark1017) ; [هاوکینز 2004](#bookmark1018) ; [دفرنز و کمسلی 1997](#bookmark1014) )، هواشناسی [(](#bookmark1019) هسیه و تانگ [1998](#bookmark1019) [)،](#bookmark1018) امور مالی [( دوایر 2005](#bookmark1015) ) و تحقیقات زناشویی ( هیمن و اسلپ [2001](#bookmark1018) ) به چند مورد اشاره می‌کنند. این منابع نشان می‌دهد که برازش بیش از حد یک نگرانی برای هر مدل پیش‌بینی بدون در نظر گرفتن زمینه تحقیق است. هدف این فصل تشریح و تشریح اصول کلیدی ایجاد پایه‌ای است که می‌توان بر اساس آن مدل‌های قابل اعتماد ساخته و متعاقباً برای پیش‌بینی استفاده کرد. به‌طور خاص، استراتژی‌هایی را توصیف می‌کنیم که ما را قادر می‌سازد اطمینان داشته باشیم که مدلی که می‌سازیم، نمونه‌های جدید را با درجه دقت مشابهی بر روی مجموعه داده‌هایی که مدل برای آن ارزیابی شده است، پیش‌بینی می‌کند. بدون این اطمینان، پیش‌بینی‌های مدل *بی فایده* است.

در یک نکته عملی، تمام تلاش‌های ساخت مدل توسط ­داده‌های موجود محدود می‌شوند. برای بسیاری از مسائل، داده‌ها ممکن است تعداد محدودی نمونه داشته باشند، ممکن است کیفیت کمتر از مطلوبی داشته باشند، و/یا ممکن است نماینده ­نمونه‌های آینده نباشند. در حالی که راه‌هایی برای ساخت مدل‌های پیش‌بینی بر روی مجموعه‌های داده کوچک وجود دارد که در این فصل توضیح خواهیم داد، اما فرض می‌کنیم که کیفیت داده‌ها کافی است و نماینده کل جامعه نمونه است.

با کار بر اساس این مفروضات، باید از داده‌های موجود برای یافتن بهترین مدل پیش‌بینی استفاده کنیم. تقریباً تمام تکنیک‌های مدل‌سازی پیش‌بینی دارای پارامترهای تنظیمی هستند که مدل را قادر می‌سازد تا ساختار را در داده‌ها پیدا کند. از این رو، ما باید از داده‌های موجود برای شناسایی تنظیمات پارامترهای مدل استفاده کنیم که بهترین و واقعی‌ترین عملکرد پیش‌بینی (معروف به تنظیم مدل) را ارائه می‌دهند. به‌طور سنتی، این با تقسیم داده‌های موجود به مجموعه‌های آموزشی و آزمایشی به دست می‌آید. مجموعه آموزشی برای ساخت و تنظیم مدل و مجموعه تست برای تخمین عملکرد پیش‌بینی مدل استفاده می‌شود. رویکردهای مدرن برای ساخت مدل، داده‌ها را به مجموعه‌های آموزشی و آزمایشی متعدد تقسیم می‌کند که نشان داده شده است که اغلب پارامترهای تنظیم بهینه‌تری را پیدا می‌کنند و نمایش دقیق‌تری از عملکرد پیش‌بینی مدل ارائه می‌دهند.

برای شروع این فصل، مفهوم برازش بیش از حد را از طریق مثالی که به راحتی قابل تجسم است، توضیح خواهیم داد. برای جلوگیری از برازش بیش از حد، ما یک رویکرد ساخت مدل کلی را پیشنهاد می‌کنیم که شامل تنظیم مدل و ارزیابی مدل ­با هدفِ نهاییِ یافتن ساختار قابل تکرار در داده‌ها است. این رویکرد مستلزم تقسیم داده‌های موجود به مجموعه‌های مجزا برای اهداف تنظیم پارامترهای مدل و ارزیابی عملکرد مدل است. انتخاب روش تقسیم داده‌ها به ویژگی‌های داده‌های موجود مانند اندازه و ساختار آن بستگی دارد. در بخش [4. 4 ، ما برعکس‌ترین](#bookmark210) تکنیک‌های تقسیم داده‌ها را تعریف و توضیح می‌دهیم و مزایا و معایب هر کدام را بررسی می‌کنیم. ­در نهایت، فصل را با بخش محاسباتی که کدی را برای اجرای استراتژی ساخت مدل کلی ارائه می‌کند، پایان می‌دهیم.

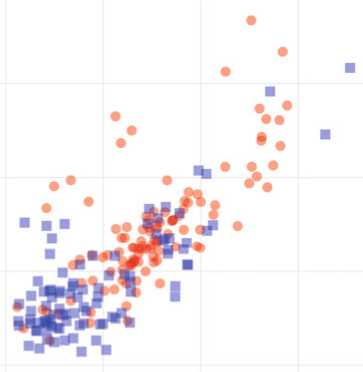
4. 1 مسأله بیش برازش

در حال حاضر تکنیک‌های زیادی وجود دارد که می‌توانند ساختار مجموعه‌ای از داده‌ها را آنقدر خوب یاد بگیرند که وقتی مدل بر روی داده‌هایی که مدل بر اساس آن ساخته شده است اعمال می‌شود، هر نمونه را به درستی پیش‌بینی می‌کند. مدل علاوه بر یادگیری الگوهای کلی در داده‌ها، ویژگی‌های نویز منحصر به فرد هر نمونه را نیز یاد گرفته است. گفته می‌شود که این نوع مدل بیش از حد مناسب است و معمولاً هنگام پیش‌بینی نمونه جدید دقت ضعیفی دارد. برای نشان دادن بیش برازش و مفاهیم دیگر در این فصل، به مثال طبقه‌بندی ساده در شکل 1 توجه کنید.  [4. 1](#bookmark196) که دارای دو متغیر پیش‌بینی (یعنی متغیرهای مستقل) است. این داده‌ها شامل 208 نمونه است که به‌عنوان "کلاس 1" یا "کلاس 2" تعیین شده اند. کلاس‌ها نسبتا متعادل هستند. 111 نمونه در کلاس اول و 97 نمونه در کلاس دوم وجود دارد. علاوه بر این، همپوشانی قابل‌توجهی بین کلاس‌ها وجود دارد که اغلب برای اکثر مسائل مدل‌سازی کاربردی وجود دارد.

یک هدف برای مجموعه داده‌ای مانند این، ایجاد مدلی برای طبقه‌بندی نمونه‌های جدید است. در این مثال دو بعدی، مدل‌ها یا قوانین طبقه‌بندی را می‌توان با خطوط مرزی نشان داد. ­شکل [4. 2](#bookmark196) نمونه مرزهای کلاس را از دو مدل طبقه‌بندی مجزا نشان می‌دهد. خطوط، ناحیه‌ای را می‌پوشانند که هر مدل داده‌ها را به‌عنوان کلاس دوم پیش‌بینی می‌کند (مربع‌های آبی). پانل سمت چپ ("مدل شماره 1") مرزی را نشان می‌دهد که پیچیده است و **در ­وسوسه محصور کردن هر نقطه داده ممکن** است. الگوی این پانل به احتمال زیاد به داده‌های جدید تعمیم نمی‌یابد. پانل سمت راست **یک مدل جایگزین مناسب** را نشان می‌دهد که در آن مرز نسبتاً صاف است و برای طبقه‌بندی صحیح هر نقطه داده در مجموعه آموزشی خود را بیش از حد گسترش نمی‌دهد.

برای سنجش میزان خوبی که مدل نمونه‌ها را طبقه‌بندی می‌کند، یک شخص می‌تواند از مجموعه آموزشی استفاده کند. در انجام این کار، میزان خطای تخمین زده شده برای مدل در ­پانل سمت چپ بیش از حد خوش بینانه خواهد بود. تخمین سودمندی یک مدل با **پیش‌بینی مجدد** مجموعه آموزشی به ***عملکرد ظاهری***مدل اشاره می‌شود (مثلاً میزان خطای ظاهری). **در دو بعد، تجسم اینکه یک مدل بیش برازش دارد، دشوار نیست، اما اکثر مسائل مدل‌سازی در ابعاد بسیار بالاتری هستند. در این مواقع، داشتن ابزاری برای مشخص کردن اینکه یک مدل تا چه اندازه با داده‌های آموزشی مطابقت بیش از حد دارد بسیار مهم است**.

کلاس 1 کلاس 2



0.6-

0.4-

0.2-

0.0-

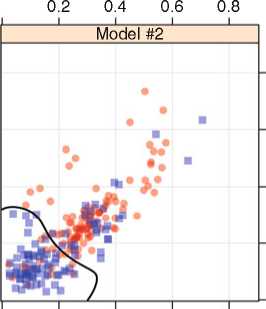
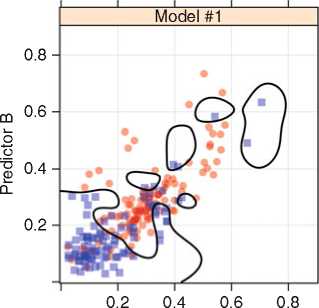
0.0 0.2

0.4 0.6

پیش‌بینی A

نمونه‌ای از داده‌های طبقه‌بندی که در سرتاسر فصل استفاده می‌شود

کلاس 1 • کلاس 2



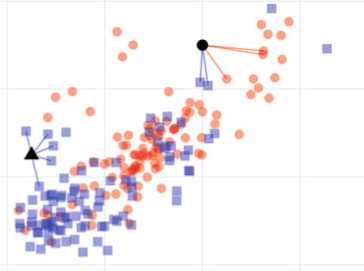
پیش‌بینی A

نمونه‌ای از مجموعه آموزشی با دو کلاس و دو پیش‌بینی. پانل‌ها دو مدل طبقه‌بندی مختلف و مرزهای کلاس مرتبط با آنها را نشان می‌دهند

**4. 2 تنظیم مدل**

بسیاری از مدل‌ها دارای پارامترهای مهمی هستند که نمی‌توان مستقیماً از روی داده‌ها تخمین زد. به‌عنوان مثال، در مدل طبقه‌بندی *K- نزدیک‌ترین* همسایه، یک نمونه جدید بر اساس *K-* نزدیک‌ترین نقاط داده در مجموعه آموزشی پیش‌بینی می‌شود. تصویری از مدل 5 نزدیکترین همسایه در شکل 1 نشان داده شده است.  [4. 3 .](#bookmark199) در اینجا، دو نمونه جدید (که با نقطه دایره توپر و مثلث پر شده مشخص می‌شوند) پیش‌بینی می‌شوند. یک نمونه ( • ) نزدیک به مخلوطی از دو کلاس است. سه تا از پنج همسایه نشان می‌دهند که نمونه باید به‌عنوان کلاس اول پیش‌بینی شود. نمونه دیگر (*k)* دارای هر پنج نقطه است که نشان می‌دهد کلاس دوم باید پیش‌بینی شود. این سوال باقی می‌ماند که **چه تعداد همسایه باید استفاده شود**. انتخاب همسایه‌های خیلی کم ممکن است با تک تک نقاط مجموعه آموزش بیش برازش داشته باشد، در حالی که همسایه‌های بیش از حد ممکن است آنقدر حساس نباشند که عملکرد معقول را ارائه دهد. این نوع پارامتر مدل به‌عنوان *پارامتر تنظیم نامیده می‌شود* زیرا هیچ فرمول تحلیلی برای محاسبه مقدار مناسب وجود ندارد.

کلاس 1 • کلاس 2 ■



0.6

0.4

0.2

0.0

0.0 0.2 0.4 0.6

پیش‌بینی A

مدل طبقه‌بندی *K* نزدیکترین همسایه. دو نقطه جدید که ­نماد آن *مثلث پر* و *نقطه جامد است،* با استفاده از مجموعه آموزشی پیش‌بینی می‌شوند.

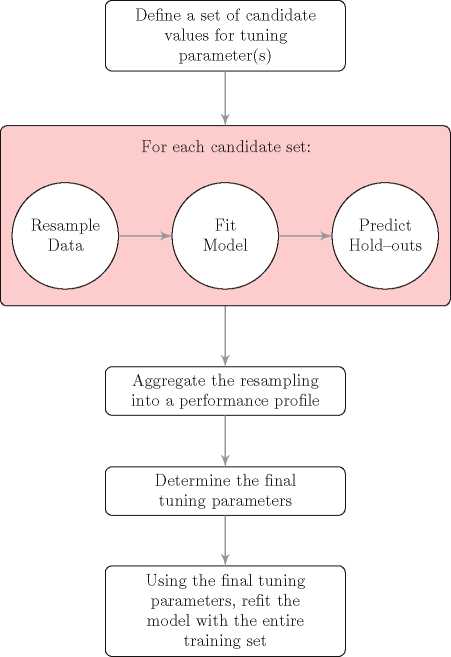
چندین مدل مورد بحث در این متن حداقل یک پارامتر تنظیم دارند. از آنجایی که بسیاری از این پارامترها پیچیدگی مدل را کنترل می‌کنند، **انتخاب ضعیف** برای مقادیر می‌تواند منجر به **بیش برازش** شود. شکل [4. 2](#bookmark196) این نکته را نشان می‌دهد. یک **ماشین بردار پشتیبانی** (بخش [13. 4 )](#bookmark32) برای ایجاد مرزهای کلاس در هر پانل استفاده شد. یکی از پارامترهای تنظیم برای این مدل، قیمت نمونه‌های طبقه‌بندی نادرست در مجموعه آموزشی را تعیین می‌کند و به‌طور کلی به‌عنوان **پارامتر "هزینه**" نامیده می‌شود. وقتی هزینه زیاد باشد، مدل تمام تلاش خود را می‌کند تا هر نقطه را به درستی برچسب‌گذاری کند (مانند پانل سمت چپ) در حالی که وقتی هزینه کم باشد مدل‌هایی را تولید می‌کنند که چندان تهاجمی نیستند. مرز کلاس در پانل سمت چپ با تنظیم دستی پارامتر هزینه روی یک عدد بسیار بالا ایجاد شد. **در پانل سمت راست، ارزش هزینه با استفاده از ­اعتبارسنجی متقاطع تعیین شد** (بخش.  [4. 4 )](#bookmark210) .

روش‌های مختلفی برای جستجوی بهترین پارامترها وجود دارد. یک ­رویکرد عمومی که می‌تواند تقریباً برای هر مدلی اعمال شود، تعریف مجموعه‌ای از مقادیر کاندید، تولید تخمین‌های قابل اعتماد از مطلوبیت مدل در میان ­مقادیر کاندید، سپس انتخاب تنظیمات بهینه است. نمودار جریان این فرآیند در شکل 1 نشان داده شده است.  [4. 4 .](#bookmark201)

هنگامی که مجموعه‌ای از مقادیر پارامتر انتخاب شد، باید برآوردهای قابل اعتمادی از عملکرد مدل به دست آوریم. سپس عملکرد نمونه‌های نگهدارنده (hold-out samples) در یک پروفایل عملکرد همفزون می‌شود که سپس برای تعیین پارامترهای تنظیم نهایی استفاده می‌شود. سپس یک مدل نهایی با تمام داده‌های آموزشی با استفاده از پارامترهای تنظیم انتخاب شده می‌سازیم. با استفاده از مثال *K-* نزدیکترین همسایه برای نشان دادن رویه شکل.  [4. 4](#bookmark201) ، مجموعه کاندید ممکن است شامل **تمام مقادیر فرد *K* بین 1 و 9 باشد** (مقادیر فرد در موقعیت دو کلاسه برای جلوگیری از تساوی استفاده می‌شود). سپس داده‌های آموزشی بارها برای هر مقدار پارامتر تنظیم مجدد نمونه‌گیری و ارزیابی می‌شوند. سپس این نتایج برای یافتن مقدار بهینه *K جمع می‌*شوند.

رویه تعریف شده در شکل.  [4. 4](#bookmark201) از مجموعه‌ای از مدل‌های کاندید استفاده می‌کند که توسط پارامترهای تنظیم تعریف شده اند. سایر رویکردها مانند الگوریتم‌های ژنتیک ( [میچل 1998](#bookmark1022) ) یا روش‌های جستجوی سیمپلکس [( اولسون و نلسون 1975](#bookmark1023) ) نیز می‌توانند پارامترهای تنظیم بهینه را پیدا کنند. این روش‌ها به‌طور الگوریتمی مقادیر مناسبی را برای پارامترهای تنظیم تعیین می‌کنند و تا زمانی که ­به تنظیمات پارامتر با عملکرد بهینه برسند، تکرار می‌شوند. این تکنیک‌ها تمایل دارند تعداد زیادی از مدل‌های کاندید را ارزیابی کنند و زمانی که عملکرد مدل را بتوان به‌طور موثر محاسبه کرد، می‌تواند بر مجموعه‌ای از پارامترهای تنظیم تعریف شده برتر باشد.  [کوهن و همکاران ( 2005](#bookmark1013) ) مقایسه‌ای از روال‌های جستجو را برای تنظیم مدل ماشین بردار پشتیبان ارائه می‌دهد.

مسأله دشوارتر بدست آوردن تخمین‌های قابل اعتماد از مدل در هر عملکرد ­برای این مدل‌های کاندید است. همانطور که قبلاً بحث شد، نرخ خطای ظاهری می‌تواند برآوردهای عملکرد بسیار خوش بینانه‌ای را ایجاد کند. یک ­رویکرد بهتر این است که مدل را بر روی نمونه‌هایی که برای آموزش استفاده نشده اند، آزمایش کنیم.



شکل 4. 4: شماتیکی از فرآیند تنظیم پارامتر. نمونه‌ای از مجموعه نامزدی از مقادیر پارامتر تنظیم برای *K* -نزدیک‌ترین همسایه‌ها ممکن است اعداد فرد بین 1 و 9 باشد. برای هر یک از این مقادیر، داده‌ها چندین بار نمونه‌برداری می‌شوند تا عملکرد مدل برای هر مقدار ارزیابی شود.

ارزیابی مدل در یک مجموعه آزمایشی انتخاب واضحی است، اما برای به دست آوردن دقت معقول از مقادیر عملکرد، ممکن است نیاز باشد اندازه مجموعه آزمایشی بزرگ باشد.

یک رویکرد جایگزین برای ارزیابی یک مدل در یک مجموعه آزمایشی، ***نمونه‌برداری مجدد***از مجموعه آموزشی است. این فرآیند از چندین نسخه اصلاح شده مجموعه آموزشی برای ساخت مدل‌های متعدد استفاده می‌کند و سپس از روش‌های آماری برای ارائه برآوردهای صادقانه از عملکرد مدل استفاده می‌کند (یعنی نه خیلی خوش‌بینانه). بخش [4. 4](#bookmark210) چندین تکنیک نمونه‌گیری مجدد را نشان می‌دهد و بخش.  [4. 6](#bookmark6) رویکردهایی را برای انتخاب پارامترهای نهایی با استفاده از نتایج نمونه‌گیری مجدد مورد بحث قرار می‌دهد.

**تقسیم داده ها**

اکنون که روش کلی برای یافتن پارامترهای تنظیم بهینه را بیان کردیم، به بحث در مورد قلب فرآیند می‌پردازیم: تقسیم داده ها. برخی از مراحل رایج در ساخت مدل عبارتند از:

* پیش پردازش داده‌های متغیر پیش‌بینی
* تخمین پارامترهای مدل
* انتخاب پیش‌بینی‌کننده‌ها برای مدل
* ارزیابی عملکرد مدل
* قوانین پیش‌بینی کلاس تنظیم دقیق (از طریق منحنی‌های ROC و غیره)

با توجه به مقدار ثابتی از داده، مدل ساز باید تصمیم بگیرد که چگونه نقاط داده خود را برای تطبیق با این فعالیت‌ها «خرج» کند.

یکی از اولین تصمیماتی که هنگام مدل‌سازی باید گرفته شود، تصمیم‌گیری است که از کدام نمونه‌ها برای ارزیابی عملکرد استفاده شود. در حالت ایده‌آل، مدل باید بر روی نمونه‌هایی ارزیابی شود که برای ساخت یا تنظیم دقیق مدل مورد استفاده قرار نگرفته‌اند، به‌طوری که حس بدون بایاس از اثربخشی مدل ارائه کنند. هنگامی که حجم زیادی از داده‌ها در دسترس است، مجموعه‌ای از نمونه‌ها را می‌توان برای ارزیابی مدل نهایی کنار گذاشت. مجموعه داده‌های "آموزش" اصطلاح کلی برای نمونه‌هایی است که برای ایجاد مدل استفاده می‌شود، در حالی که مجموعه داده‌های "تست" یا "اعتبارسنجی" برای واجد شرایط بودن عملکرد استفاده می‌شود.

با این حال، زمانی که تعداد نمونه‌ها زیاد نباشد، می‌توان یک مورد قوی ایجاد کرد که باید از یک مجموعه آزمایشی اجتناب شود زیرا ممکن است هر نمونه برای ساخت مدل مورد نیاز باشد. علاوه بر این، اندازه مجموعه تست ممکن است قدرت یا دقت کافی برای قضاوت منطقی نداشته باشد. چندین محقق [( Molinaro 2005](#bookmark1022) ; [مارتین و هیرشبرگ 1996](#bookmark1021) ; [هاوکینز و همکاران 2003](#bookmark1018) ) نشان می‌دهد که اعتبارسنجی با استفاده از [g](#bookmark1018) یک مجموعه آزمایشی می‌تواند انتخاب ضعیفی باشد. هاوکینز و همکاران [( 2003](#bookmark1018) ) به‌طور خلاصه این نکته را خلاصه می‌کند: "**نمونه‌های نگهدارنده با اندازه قابل تحمل [. . . ] با خود اعتبارسنجی متقابل برای قابلیت اطمینان در ارزیابی برازش مدل مطابقت ندارند و انگیزه دادن به آنها سخت است**. " **روش‌های نمونه‌گیری مجدد، مانند اعتبارسنجی متقابل، می‌تواند برای تولید تخمین‌های مناسب از عملکرد مدل با استفاده از مجموعه آموزشی استفاده شود**. اینها به‌طور مفصل در بخش بحث شده است.  [4. 4 .](#bookmark210) اگرچه تکنیک‌های نمونه‌گیری مجدد را می‌توان به اشتباه به کار برد، مانند مثال نشان‌داده‌شده در Ambroise و McLachlan ( 2002 )، آنها اغلب تخمین‌های عملکردی را بالاتر از یک مجموعه آزمایشی واحد تولید می‌کنند، زیرا بسیاری از نسخه‌های جایگزین داده‌ها را ارزیابی می‌کنند.

اگر یک مجموعه آزمایش ضروری تشخیص داده شود، روش‌های مختلفی برای تقسیم نمونه‌ها وجود دارد. گاهی اوقات رویکردهای غیر تصادفی برای تقسیم داده‌ها مناسب است. مثلا،

* اگر از مدلی برای پیش‌بینی نتایج بیمار استفاده می‌شد، مدل ممکن است با استفاده از مجموعه‌های بیمار خاص (مثلاً از همان محل بالینی یا مرحله بیماری) ایجاد شود و سپس بر روی یک جمعیت نمونه متفاوت آزمایش شود تا بفهمد که چقدر مدل به خوبی تعمیم می‌یابد.
* در مدل‌سازی شیمیایی برای کشف دارو، "فضای شیمیایی" جدید دائما در حال کاوش است. ما بیشتر به پیش‌بینی‌کننده‌های دقیق در ­فضای شیمیایی که در حال حاضر در حال بررسی است، علاقه مند هستیم تا فضایی که سال‌ها قبل ارزیابی شده بود. همین امر را می‌توان در مورد فیلتر هرزنامه نیز گفت. برای مدل مهم‌تر است که تکنیک‌های ارسال هرزنامه جدید را به جای طرح‌های ارسال هرزنامه قبلی درک کند.

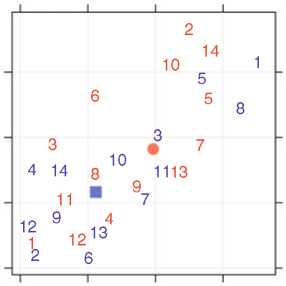
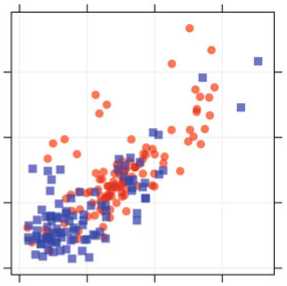
با این حال، در بیشتر موارد، تمایل به ایجاد یکنواختی تا حد امکان مجموعه‌های آموزشی و تست وجود دارد. برای ایجاد مجموعه داده‌های مشابه می‌توان از روش‌های نمونه‌گیری تصادفی استفاده کرد.

ساده‌ترین راه برای تقسیم داده‌ها به یک مجموعه آموزشی و آزمایشی، گرفتن یک نمونه تصادفی ساده است. این هیچ یک از ویژگی‌های داده، مانند درصد داده‌ها در کلاس‌ها را کنترل نمی‌کند. هنگامی که یک کلاس دارای ­فرکانس نامتناسب نسبتاً کمی در مقایسه با سایرین است، این احتمال وجود دارد که توزیع نتایج بین مجموعه آموزشی و آزمون تفاوت اساسی داشته باشد.

برای در نظر گرفتن نتیجه هنگام تقسیم داده‌ها، **نمونه‌گیری تصادفی طبقه‌ای** از نمونه‌گیری تصادفی در زیر گروه‌ها (مانند کلاس‌ها) استفاده می‌کند. به این ترتیب، احتمال بیشتری وجود دارد که توزیع‌های نتیجه مطابقت داشته باشند. **هنگامی که نتیجه یک عدد است، می‌توان از یک استراتژی مشابه استفاده کرد. مقادیر عددی به گروه‌های مشابه (مثلاً کم، متوسط و زیاد) تقسیم می‌شوند و تصادفی‌سازی در این گروه‌ها انجام می‌شود.**

. [ویلت ( 1999](#bookmark1027) ) و [کلارک](#bookmark1013) [( 1997](#bookmark1013) ) تقسیم داده‌ها را بر اساس ***نمونه‌گیری حداکثر تفاوت*** *(maximum dissimilarity sampling) پیشنهاد می‌کند.* عدم تشابه بین دو نمونه را می‌توان به روش‌های مختلفی اندازه‌گیری کرد. ساده‌ترین روش استفاده از فاصله بین مقادیر پیش‌بینی برای دو نمونه است. اگر فاصله کم باشد، نقاط در مجاورت نزدیک هستند. فواصل بزرگتر بین نقاط نشان دهنده ناهمگونی ­است. برای استفاده از عدم تشابه به‌عنوان ابزاری برای تقسیم داده‌ها، فرض کنید مجموعه تست با یک نمونه واحد (یک ردیف یا یک نقطه داده) مقداردهی اولیه شده است. تفاوت بین این نمونه اولیه ­و نمونه‌های تخصیص نیافته قابل محاسبه است. سپس نمونه تخصیص نیافته که بسیار متفاوت است به مجموعه آزمایشی اضافه می‌شود. برای تخصیص نمونه‌های بیشتر به مجموعه آزمون، روشی برای تعیین تفاوت‌های بین *گروه‌های* نقاط (یعنی دو نقطه در مجموعه آزمایشی و نقاط تخصیص نشده) مورد نیاز است. یکی از رویکردها استفاده از میانگین یا حداقل عدم تشابه است. به‌عنوان مثال، برای اندازه‌گیری تفاوت‌های بین دو نمونه در مجموعه آزمایشی و یک نقطه تخصیص‌نخورده، می‌توانیم دو عدم تشابه را تعیین کرده ­و از آنها میانگین‌گیری کنیم. سومین نقطه اضافه شده به مجموعه آزمایشی به‌عنوان دارای حداکثر میانگین تفاوت با مجموعه موجود انتخاب می‌شود. این فرآیند تا رسیدن به اندازه مجموعه آزمایشی هدفمند ادامه خواهد داشت.

شکل [4. 5](#bookmark210) این فرآیند را برای داده‌های طبقه‌بندی مثال نشان می‌دهد. نمونه‌گیری ­شباهت به‌طور جداگانه در هر کلاس انجام شد. ابتدا، یک نمونه ­در هر کلاس برای شروع فرآیند انتخاب شد (در شکل به صورت ■ و • مشخص شده است). عدم تشابه نمونه اولیه به نمونه‌های تخصیص نیافته در کلاس محاسبه شد و متفاوت‌ترین نقطه به مجموعه آزمون اضافه شد. برای کلاس اول، متفاوت‌ترین نقطه در منتهی الیه جنوب غربی نمونه اولیه بود. در دور دوم، ­تفاوت‌های غیرمشابه با استفاده از حداقل (برخلاف میانگین) همفزون شدند. از نو، برای کلاس اول، نقطه انتخاب شده بسیار دور در شمال شرقی فضای متغیر پیش‌بینی بود. با ادامه نمونه‌گیری، نمونه‌ها در حاشیه داده‌ها انتخاب شدند و سپس به سمت داخل کار کردند.



Class 1 • Class 2 ■

0.6

0.4

0.2

0.0

0.0 0.2 0.4 0.6

Predictor A

Fig. 4.5: An example of maximum dissimilarity sampling to create a test set. After choosing an initial sample within a class, 14 more samples were added

Class 1 • Class 2 ■

0.6

0.4

0.2

0.0

0.0 0.2 0.4 0.6

Predictor A

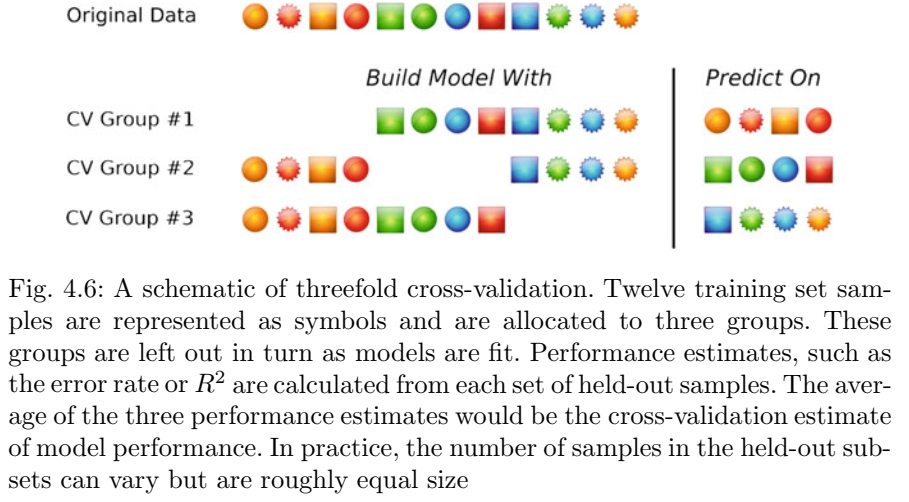
مارتین و همکاران [( 2012](#bookmark1022) ) روش‌های مختلف تقسیم داده‌ها از جمله نمونه‌گیری تصادفی، نمونه‌گیری ناهمسانی و روش‌های دیگر را با هم مقایسه می‌کند.

تکنیک‌های نمونه‌گیری مجدد

به‌طور کلی، تکنیک‌های نمونه‌گیری مجدد برای تخمین عملکرد مدل به‌طور مشابه عمل می‌کنند: زیر مجموعه‌ای از نمونه‌ها برای برازش یک مدل و نمونه‌های باقی‌مانده ­برای تخمین کارایی مدل استفاده می‌شوند. این فرآیند چندین بار تکرار می‌شود و نتایج جمع‌آوری و خلاصه می‌شوند. تفاوت ­در تکنیک‌ها معمولا حول روشی است که در آن نمونه‌های فرعی انتخاب می‌شوند. ما طعم‌های اصلی نمونه برداری مجدد را در چند بخش بعدی در نظر خواهیم گرفت.

k-Fold Cross-Validation

نمونه‌ها به‌طور تصادفی به *k* مجموعه با اندازه تقریبا مساوی تقسیم می‌شوند. یک مدل با استفاده از همه نمونه‌ها به جز زیر مجموعه اول (**به نام اولین *فولد***) برازش می‌شود. **نمونه‌های نگه‌داشته‌شده** (**احتمالاً همان اولین *فولد***) توسط این مدل پیش‌بینی می‌شوند و برای برآورد معیارهای عملکرد استفاده می‌شوند. زیرمجموعه اول به مجموعه آموزشی برگردانده می‌شود و با نگه داشتن (held out) زیر مجموعه دوم، روند تکرار می‌شود و به همین ترتیب. برآوردهای عملکرد *k* ­ نمونه گیری مجدد خلاصه می‌شود (معمولاً با میانگین و خطای استاندارد) و برای درک رابطه بین پارامتر(های) تنظیم و کاربرد مدل استفاده می‌شود. فرآیند اعتبار سنجی متقابل با *k* =3 در شکل 1 نشان داده شده است.  [4. 6](#bookmark216) .



شماتیکی از اعتبارسنجی متقاطع سه گانه. 12 نمونه مجموعه آموزشی ­به صورت نماد نمایش داده می‌شوند و به سه گروه تخصیص داده می‌شوند. این گروه‌ها به نوبه خود به دلیل مناسب بودن مدل‌ها کنار گذاشته می‌شوند. برآوردهای عملکرد، مانند میزان خطا یا *R2* از هر مجموعه‌ای از نمونه‌های نگهداری شده محاسبه می‌شود. میانگین ­سنی سه برآورد عملکرد، تخمین اعتبار متقابل عملکرد مدل خواهد بود. در عمل، تعداد نمونه‌ها در زیر ­مجموعه‌های نگه‌داشته‌شده می‌تواند متفاوت باشد، اما اندازه آنها تقریباً برابر است

یک نوع کوچک از این روش، انتخاب پارتیشن‌های *k* به‌گونه‌ای است که دسته‌ها نسبت به نتیجه متعادل شوند [( کوهاوی 1995](#bookmark1020) ). ­نمونه‌گیری طبقه‌ای تصادفی را انجام داد که قبلاً در بخش مورد بحث قرار گرفت.  [4. 3](#bookmark204) ، تعادل را با توجه به نتیجه ایجاد می‌کند.

نسخه دیگر، اعتبارسنجی متقاطع ترک یک خروجی (LOOCV)، مورد خاصی است که *k* تعداد نمونه‌ها است. **leave-one-out cross-validation (LOOCV)**

در این مورد، از آنجایی که تنها یک نمونه ­در هر زمان نگه داشته می‌شود، عملکرد نهایی از *k* در ­پیش‌بینی‌کننده‌های منقسم‌شده محاسبه می‌شود. بعلاوه، اعتبارسنجی متقاطع *k -fold مکرر رویه شکل 1 را تکرار می‌کند.*  [4. 6](#bookmark216) چندین بار. **برای مثال، اگر اعتبارسنجی متقاطع 10 برابری پنج بار تکرار شود، 50 مجموعه نگه‌داشته‌شده مختلف برای تخمین اثربخشی مدل استفاده می‌شود.**

انتخاب *k* معمولاً 5 یا 10 است، اما هیچ قانون رسمی وجود ندارد. با بزرگتر شدن *k،* تفاوت در اندازه بین مجموعه آموزشی و زیر مجموعه‌های نمونه برداری مجدد کوچکتر می‌شود. با کاهش این تفاوت، *بایاس* تکنیک کوچکتر می‌شود (یعنی بایاس برای *k* = 10 از *k* = 5 کوچکتر است). در این زمینه، بایاس تفاوت بین مقادیر برآورد شده و واقعی عملکرد است.

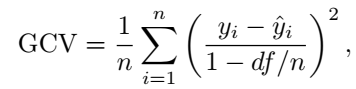
یکی دیگر از جنبه‌های مهم تکنیک نمونه‌گیری مجدد عدم قطعیت (یعنی واریانس یا نویز) است. یک روش بی طرفانه ممکن است ارزش صحیح را تخمین بزند (مثلاً عملکرد نظری واقعی) اما ممکن است در عدم قطعیت هزینه بالایی را بپردازد. این بدان معناست که تکرار روش نمونه‌گیری مجدد ممکن است ­مقدار بسیار متفاوتی تولید کند (اما اگر بارها انجام شود، مقدار واقعی را تخمین می‌زند). **اعتبار سنجی متقاطع *k* -fold به‌طور کلی در مقایسه با سایر روش‌ها واریانس بالایی دارد و به همین دلیل ممکن است جذاب نباشد**. باید گفت که برای مجموعه‌های آموزشی بزرگ، مسائل احتمالی با واریانس و بایاس ناچیز می‌شود.

از نقطه نظر عملی، مقادیر بزرگتر *k* از نظر محاسباتی سنگین تر هستند. در نهایت، LOOCV از نظر محاسباتی بیشترین مالیات را دارد زیرا به تعداد نقاط داده به برازش مدل نیاز دارد و هر برازش مدل از زیر مجموعه‌ای استفاده می‌کند که تقریباً به اندازه مجموعه آموزشی است.  [مولینارو ( 2005](#bookmark1022) ) دریافت که اعتبار سنجی متقاطع 10 برابری و *k* = 10 برابر نتایج مشابهی را به همراه دارد که نشان می‌دهد *k* = 10 از منظر کارایی محاسباتی جذاب تر است. همچنین، مقادیر کوچک *k،* مثلاً 2 یا 3، بایاس بالایی دارند اما از نظر محاسباتی بسیار کارآمد هستند. با این حال، بایاس که با مقادیر کوچک *k همراه* است، تقریباً مشابه بایاس تولید شده توسط بوت استرپ است (به پایین مراجعه کنید)، اما با واریانس بسیار بیشتر.

تحقیق [( Molinaro 2005](#bookmark1022) ; [Kim 2009](#bookmark1020) نشان می‌دهد که تکرار اعتبارسنجی متقاطع ­*k -fold* می‌تواند برای افزایش مؤثر دقت تخمین‌ها و در عین حال حفظ یک بایاس کوچک استفاده شود.

اعتبار سنجی متقابل تعمیم یافته

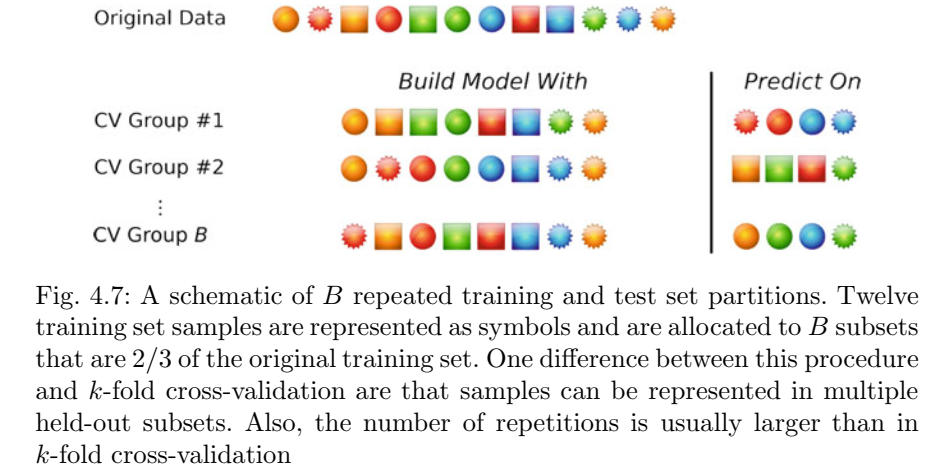
برای مدل‌های رگرسیون خطی، فرمولی برای تقریب نرخ خطای ترک یک خروجی وجود دارد. [آماره](#bookmark1017) اعتبارسنجی متقابل عمومی شده (GCV) ( Golub et al. [1979](#bookmark1017) ) نیازی به تطبیق مجدد مدل در زیر مجموعه داده‌های مختلف ندارد. فرمول این آمار، اولین *نتیجه* مجموعه آموزشی *است*



جایی که *y i* در نتیجه مجموعه آموزشی، i ام است، *y i پیش‌بینی مدل* آن نتیجه و *df* درجه آزادی مدل است. درجات آزادی حسابداری از تعداد پارامترهایی است که توسط مدل تخمین زده می‌شود و با بسط، معیاری از پیچیدگی برای مدل‌های رگرسیون خطی است. بر اساس این معادله، دو مدل با مجموع مربعات خطاهای یکسان (شمارگر) در صورتی که پیچیدگی‌های مدل‌ها متفاوت باشد، مقادیر GCV متفاوتی خواهند داشت.

پارتیشن بندی آموزش و آزمون مکرر

**اعتبارسنجی متقاطع ترک گروهی**» یا «**اعتبار متقابل مونت کارلو**» نیز شناخته می‌شود. ­این تکنیک به سادگی چندین تقسیم از داده‌ها را به مجموعه‌های مدل‌سازی و پیش‌بینی ایجاد می‌کند (شکل 2 را ببینید).  [4. 7 )](#bookmark219) . نسبت داده‌هایی که به هر زیر مجموعه می‌روند، به‌عنوان تعداد تکرارها، توسط متخصص کنترل می‌شود. همانطور که قبلا بحث شد، بایاس تکنیک نمونه‌گیری مجدد با نزدیک شدن مقدار داده در زیر مجموعه به مقدار مجموعه مدل‌سازی کاهش می‌یابد. یک قانون کلی خوب حدود 75-80٪ است. اگر تعداد تکرارها زیاد باشد نسبت‌های بالاتر ایده خوبی است.

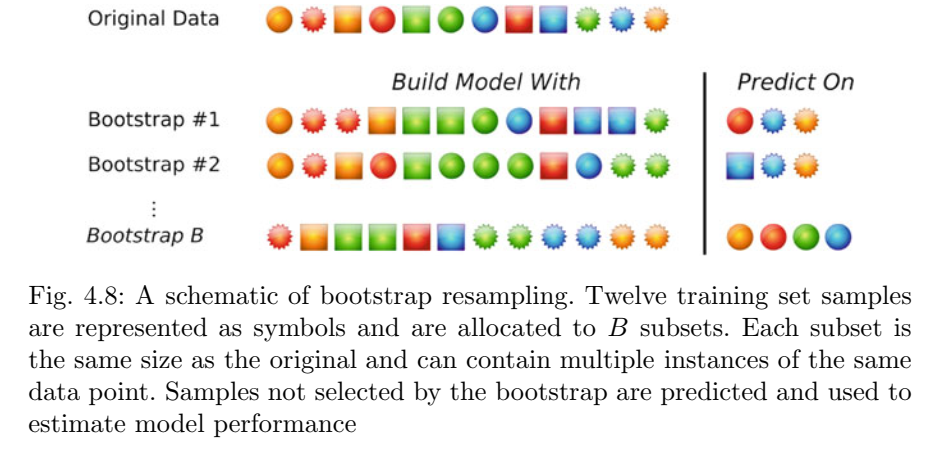


شماتیک *B* آموزش مکرر و پارتیشن‌های مجموعه تست. دوازده نمونه مجموعه آموزشی به‌عنوان نماد نمایش داده می‌شوند و به زیرمجموعه‌های *B* که 2/3 مجموعه آموزشی اصلی هستند اختصاص داده می‌شوند. یکی از تفاوت‌های بین این روش و اعتبارسنجی متقاطع *k* برابر این است که نمونه‌ها را می‌توان در زیر مجموعه‌های متعدد نگه‌داشته‌شده نشان داد. همچنین، تعداد تکرارها معمولاً بزرگتر از اعتبارسنجی متقاطع *k* برابر

تعداد تکرارها مهم است. افزایش تعداد زیر مجموعه‌ها باعث کاهش عدم قطعیت برآوردهای عملکرد می‌شود. به‌عنوان مثال، برای به دست آوردن یک برآورد ناخالص از عملکرد مدل، اگر کاربر مایل به پذیرش مقداری بی ثباتی در مقادیر حاصل باشد، 25 تکرار کافی خواهد بود. با این حال، برای به دست آوردن تخمین‌های پایدار از عملکرد، پیشنهاد می‌شود تعداد بیشتری از تکرارها را انتخاب کنید (مثلاً 50-200). این نیز تابعی از نسبت نمونه‌هایی است که به‌طور تصادفی به مجموعه پیش‌بینی تخصیص داده می‌شوند. هرچه این درصد بیشتر باشد، تکرارهای بیشتری برای کاهش عدم قطعیت در برآوردهای عملکرد مورد نیاز است.

**بوت استرپ**

نمونه بوت استرپ یک نمونه تصادفی از داده‌های گرفته شده *با جایگزین ­است* [( افرون و تبشیرانی 1986](#bookmark1015) ). این بدان معناست که پس از انتخاب یک نقطه داده برای زیر مجموعه، همچنان برای انتخاب دوباره در دسترس است. **نمونه بوت استرپ به اندازه مجموعه داده اصلی است**. در نتیجه، برخی از نمونه‌ها چندین بار در نمونه بوت استرپ نمایش داده می‌شوند در حالی که برخی دیگر اصلا انتخاب نمی‌شوند. نمونه‌هایی که انتخاب نشده اند معمولاً به‌عنوان نمونه‌های "خارج از کیسه" (out-of-bag) نامیده می‌شوند. **برای یک تکرار معین از نمونه‌برداری مجدد بوت استرپ، یک مدل بر روی نمونه‌های انتخاب شده ساخته می‌شود و برای پیش‌بینی نمونه‌های خارج از کیسه استفاده می‌شود (شکل 2).**  [**4. 8 )**](#bookmark222) **.**



شماتیکی از نمونه برداری مجدد بوت استرپ. دوازده نمونه مجموعه آموزشی به‌عنوان نماد نمایش داده می‌شوند و به زیر مجموعه‌های *B اختصاص داده می‌شوند.* هر زیرمجموعه همان اندازه اصلی است و می‌تواند چندین نمونه از یک نقطه داده را شامل شود. نمونه‌هایی که بوت استرپ انتخاب نمی‌کنند، پیش‌بینی می‌شوند و برای تخمین عملکرد مدل مشابه اعتبارسنجی متقاطع *k برابر در زمانی که k «* 2 است، استفاده می‌شوند. اگر اندازه مجموعه آموزشی کوچک باشد، این بایاس ممکن است مسأله‌ساز باشد، اما با حجم نمونه مجموعه آموزشی کاهش می‌یابد. بزرگتر می‌شود.

به طور کلی، نرخ خطای بوت استرپ نسبت به اعتبارسنجی متقاطع k-fold عدم قطعیت کمتری دارد (Efron 1983). با این حال، به طور متوسط، 63.2 درصد از نقاط داده نمونه بوت استرپ حداقل یک بار نشان داده می‌شود، بنابراین این تکنیک دارای بایاس مشابه اعتبارسنجی متقاطع k-fold در زمانی که k ≈ 2 است. اگر اندازه مجموعه آموزشی کوچک باشد، این بایاس ممکن است مشکل ساز باشد، اما با بزرگتر شدن حجم نمونه مجموعه آموزشی کاهش می‌یابد.

برای از بین بردن این بایاس، چند اصلاح روی روش ساده بوت استرپ ابداع شده است. "**روش 632**" (Efron 1983) با ایجاد یک تخمین عملکرد که ترکیبی از تخمین ساده بوت استرپ و تخمین حاصل از پیش‌بینی مجدد مجموعه آموزشی (به عنوان مثال، میزان خطای ظاهری) است، به این موضوع می‌پردازد. به عنوان مثال، اگر یک مدل طبقه‌بندی با میزان خطای آن مشخص شود، از روش 632 استفاده می شود:

(0. 632 *x* برآورد راه‌انداز ساده) + ( *0.* 368 *x* نرخ خطای ظاهری).



تخمین بوت استرپ اصلاح شده بایاس را کاهش می‌دهد، اما می‌تواند با اندازه نمونه‌های کوچک ناپایدار باشد. این تخمین همچنین می‌تواند منجر به نتایج بی‌رویه خوش‌بینانه شود، زمانی که مدل به شدت با داده‌ها برازش داشته باشد، زیرا نرخ خطای ظاهری نزدیک به صفر خواهد بود.  [افرون و تبشیرانی](#bookmark1015) [( 1997](#bookmark1015) ) تکنیک دیگری به نام "روش 632+" را برای تنظیم تخمین‌های بوت استرپ مورد بحث قرار می‌دهد.

مطالعه موردی: امتیازدهی اعتباری

یک کاربرد ساده از مدل‌های پیش‌بینی، امتیازدهی اعتباری است. از داده‌های موجود می‌توان برای ایجاد مدلی برای پیش‌بینی احتمال اعتبار مناسب متقاضیان استفاده کرد. از این اطلاعات می‌توان برای تعیین کمیت ریسک برای وام دهنده استفاده کرد.

مجموعه داده اعتباری آلمانی ابزار محبوبی برای محک زدن الگوریتم‌های یادگیری ماشینی است. این شامل 1000 نمونه است که به آنها برچسب اعتبار خوب و بد داده شده است. در مجموعه داده‌ها، 70 درصد به‌عنوان دارای اعتبار خوب رتبه‌بندی شدند. همانطور که در بخش بحث شد.  [11. 2](#bookmark512) ، هنگام ارزیابی دقت یک مدل، **نرخ دقت پایه باید 70٪ باشد** (که ما می‌توانیم به سادگی با پیش‌بینی تمام نمونه‌ها برای داشتن اعتبار خوب به آن دست پیدا کنیم).

همراه با این نتایج، داده‌های مربوط به سابقه اعتباری، اشتغال، وضعیت حساب و غیره جمع‌آوری شد. برخی از پیش‌بینی‌کننده‌ها عددی هستند، مانند مبلغ وام. با این حال، بیشتر پیش‌بینی‌کننده‌ها ماهیت طبقه‌بندی دارند، مانند هدف وام، جنسیت یا وضعیت تأهل. پیش‌بینی‌کننده‌های طبقه‌بندی به «متغیرهای ساختگی» تبدیل شدند که به یک دسته منفرد مربوط ­می‌شوند. برای مثال، اطلاعات محل اقامت متقاضی به‌عنوان «اجاره»، «مالک» یا «مسکن رایگان» طبقه‌بندی می‌شود. این پیش‌بینی برای هر دسته به سه بیت اطلاعات بله/خیر تبدیل می‌شود. به‌عنوان مثال، ­اگر متقاضی اجاره داده شود، یک پیش‌بینی ارزش یک خواهد داشت و در غیر این صورت صفر است. ایجاد متغیرهای ساختگی به‌طور طولانی در بخش بحث شده است.  [3. 6](#bookmark169) . در مجموع، 41 پیش‌بینی برای مدل‌سازی وضعیت اعتبار یک فرد مورد استفاده قرار گرفت.

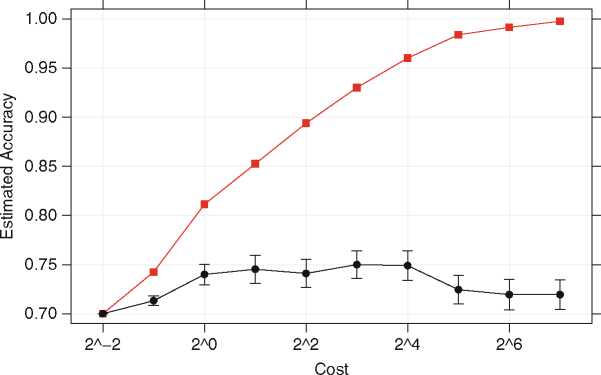
ما از این داده‌ها برای نشان دادن فرآیند تنظیم مدل‌ها ­با نمونه‌گیری مجدد، همانطور که در شکل تعریف شده است استفاده خواهیم کرد.  [4. 4](#bookmark201) . برای مثال، ما یک نمونه تصادفی طبقه‌بندی شده از 800 مشتری را برای استفاده از مدل‌های آموزشی انتخاب کردیم. نمونه‌های باقی‌مانده به‌عنوان مجموعه‌ای آزمایشی برای تأیید عملکرد زمانی که مدل نهایی تعیین شد، استفاده می‌شوند. بخش [11. 2](#bookmark512) نتایج مجموعه تست را با جزئیات بیشتری مورد بحث قرار خواهد داد.

انتخاب پارامترهای تنظیم نهایی

هنگامی که عملکرد مدل در مجموعه‌ای از پارامترهای تنظیم کمّی، محاسبه ­شد، چندین فلسفه در مورد نحوه انتخاب تنظیمات نهایی وجود دارد. ساده‌ترین روش این است که تنظیمات مربوط به تخمین‌های عددی بهترین عملکرد را انتخاب کنید.

برای مثال امتیازدهی اعتباری، یک مدل ماشین بردار پشتیبان غیرخطی[[9]](#footnote-9) [[10]](#footnote-10) [[11]](#footnote-11) بر روی مقادیر هزینه از 2 به توان منفی دو تا 2 به توان هفت ­ ارزیابی شد. هر مدل با استفاده از پنج تکرار 10 برابر اعتبار متقاطع ارزیابی شد. شکل [4. 9](#bookmark225) و جدول [4. 1](#bookmark226) نمایه دقت را در بین مقادیر نامزد پارامتر هزینه نشان می‌دهد. برای هر مدل، اعتبارسنجی متقابل 50 تخمین مختلف از دقت را ایجاد کرد. نقاط جامد در شکل [4. 9](#bookmark225) میانگین این برآوردها است. میله‌ها میانگین خطاهای دو استاندارد مثبت/منفی میانگین را منعکس می‌کنند. فایل حرفه‌ای ­افزایش دقت را نشان می‌دهد تا زمانی که ارزش هزینه یک باشد. مدل‌هایی با ارزش هزینه بین 1 و 16 نسبتا ثابت هستند. پس از آن، دقت ­کاهش می‌یابد (**احتمالاً به دلیل برازش بیش از حد**). مقدار بهینه عددی پارامتر هزینه 8 است، با نرخ دقت مربوطه 75٪. توجه داشته باشید که میزان دقت ظاهری که با پیش‌بینی مجدد نمونه‌های مجموعه آموزشی تعیین می‌شود، نشان می‌دهد که با افزایش هزینه، مدل بهبود می‌یابد، اگرچه مدل‌های پیچیده‌تر یا کاملاً متناسب با مجموعه آموزشی هستند.

ظاهری ■ اعتبار متقابل •



شکل 4. 9: نمایه عملکرد یک ماشین بردار پشتیبان تابع پایه شعاعی ­برای مثال امتیازدهی اعتبار بر روی مقادیر مختلف پارامتر هزینه. خطوط *عمودی* نشان دهنده *±* خطاهای دو استاندارد دقت است

به‌طور کلی، ممکن است ایده خوبی باشد که مدل‌های ساده‌تر را به مدل‌های پیچیده‌تر ترجیح دهیم و انتخاب پارامترهای تنظیم بر اساس مقدار بهینه عددی ممکن است منجر به مدل‌هایی شود که بیش از حد پیچیده هستند. طرح‌های دیگر برای انتخاب مدل‌های کمتر پیچیده باید بررسی شوند، زیرا ممکن است به مدل‌های ساده‌تری منجر شوند که عملکرد قابل قبولی (نسبت به ­تنظیمات عددی بهینه) ارائه می‌دهند.

روش «**خطای یک استاندارد**» برای انتخاب مدل‌های ساده‌تر، ­مقدار بهینه عددی و خطای استاندارد متناظر آن را پیدا می‌کند و سپس ساده‌ترین مدلی را جستجو می‌کند که عملکرد آن در یک خطای استاندارد واحد از بهترین مقدار عددی باشد. این روش از درختان طبقه‌بندی و رگرسیون [( Breiman et al. ( 1984](#bookmark1012) ) و Sects نشات گرفت.  [8. 1](#bookmark392) و [14. 1 )](#bookmark686) . در شکل [4. 10 ،](#bookmark229) خطای استاندارد مقادیر دقت زمانی که هزینه 8 است، حدود 0. 7٪ است. این تکنیک ساده‌ترین تنظیمات پارامتر تنظیم مرتبط با دقت کمتر از 74. 3٪ (75٪ -0. 7٪) را پیدا می‌کند. این روش مقدار 2 را برای پارامتر هزینه انتخاب می‌کند.

رویکرد دیگر این است که مدل ساده‌تری را انتخاب کنید که در محدوده تلورانس مشخصی از بهترین مقدار عددی باشد. درصد کاهش در عملکرد را می‌توان با ( *X - O* ) */O* که در آن *X* مقدار عملکرد و *O* مقدار عددی بهینه است، کمی‌سازی کرد. به‌عنوان مثال، در شکل.  [4. 9 ،](#bookmark225) بهترین مقدار دقت در سراسر نمایه 75٪ بود. اگر 4 درصد از دست دادن دقت به‌عنوان یک مبادله برای یک مدل ساده‌تر قابل قبول باشد، مقادیر دقت بیشتر از 71. 2 درصد خواهد بود.

جدول 4. 1: نتایج دقت تأیید متقابل مکرر برای مدل ماشین بردار پشتیبان

|  |  |  |  |
| --- | --- | --- | --- |
| دقت نمونه‌گیری مجدد (%) | | | |
| هزینه | منظور داشتن | Std. خطا | ٪ تحمل |
| 0. 25 | 70. 0 | 0. 0 | *-* 6. 67 |
| 0. 50 | 71. 3 | 0. 2 | *-* 4. 90 |
| 1. 00 | 74. 0 | 0. 5 | *-* 1. 33 |
| 2. 00 | 74. 5 | 0. 7 | *-* 0. 63 |
| 4. 00 | 74. 1 | 0. 7 | *-* 1. 20 |
| ساعت 8. 00 | 75. 0 | 0. 7 | 0. 00 |
| ساعت 16. 00 | 74. 9 | 0. 8 | *-* 0. 13 |
| 32. 00 | 72. 5 | 0. 7 | *-* 3. 40 |
| 64. 00 | 72. 0 | 0. 8 | *-* 4. 07 |
| 128. 00 | 72. 0 | 0. 8 | *-* 4. 07 |

قانون خطای یک استاندارد ساده‌ترین مدل را با دقت کمتر از 74. 3٪ (75٪ -0. 7٪) انتخاب می‌کند. این مربوط به ارزش هزینه 2 است. راه حل "انتخاب بهترین" به صورت پررنگ نشان داده شده است

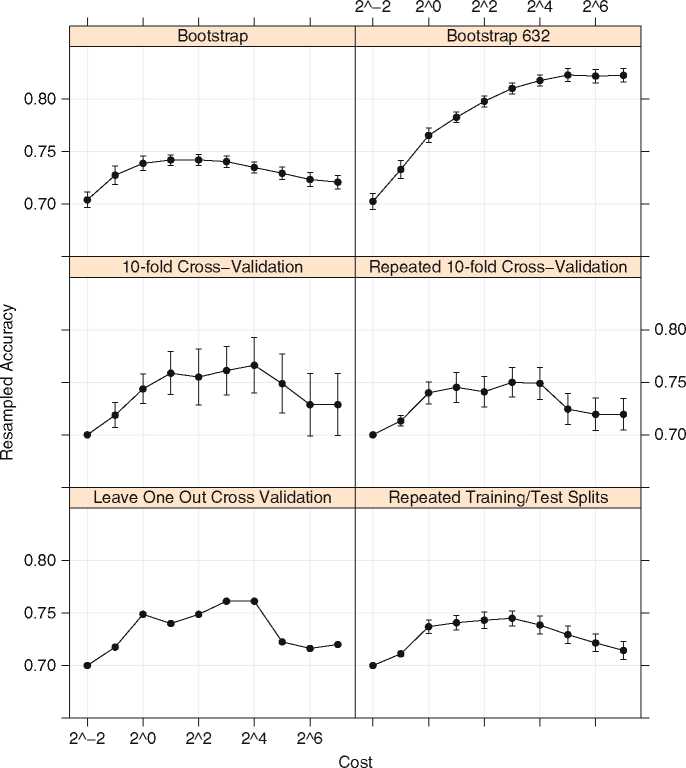
قابل قبول باشد برای نمایه در شکل [4. 9 ،](#bookmark225) ارزش هزینه 1 با استفاده از این رویکرد انتخاب می‌شود.

به‌عنوان مثال، روش‌های نمونه‌گیری مجدد اضافی برای همان داده‌ها اعمال شد: اعتبارسنجی متقاطع 10 برابری، LOOCV، راه‌انداز (با و بدون تنظیم 632) و تقسیم‌های آموزشی/آزمون مکرر (با 20 درصد باز ماندن). دو روش آخر از 50 نمونه مجدد برای تخمین عملکرد استفاده کردند.

نتایج در شکل نمایش داده شده اند.  [4. 10 .](#bookmark229) یک الگوی رایج در روش‌های اعتبارسنجی متقاطع دیده می‌شود که در آن دقت در مقادیر هزینه بین 4 تا 16 به اوج می‌رسد و تقریباً در این پنجره ثابت می‌ماند.

در هر مورد، عملکرد به سرعت با ارزش هزینه افزایش می‌یابد و سپس، پس از پیک، با سرعت کمتری کاهش می‌یابد زیرا بیش برازش شروع می‌شود. تکنیک‌های اعتبارسنجی متقاطع دقت را بین 74. 5٪ و 76. 6٪ تخمین می‌زنند. در مقایسه با روش‌های دیگر، بوت استرپ ساده کمی بدبینانه است و دقت را 74. 2% تخمین می‌زند در حالی که به نظر می‌رسد قانون 632 بیش از حد بایاس را جبران می‌کند و دقت را 82. 3% تخمین می‌زند. توجه داشته باشید که باندهای خطای استاندارد روش اعتبارسنجی متقاطع 10 برابری ساده‌تر از روش‌های دیگر است، بیشتر به این دلیل که خطای استاندارد تابعی از تعداد نمونه‌های مجدد استفاده شده است (10 در مقابل 50 مورد استفاده شده توسط بوت استرپ یا تقسیم مکرر).

زمان‌های محاسباتی به‌طور قابل‌توجهی متفاوت بود. سریع‌ترین آنها اعتبارسنجی متقاطع 10 برابری بود که در 0. 82 دقیقه انجام شد. اعتبارسنجی متقاطع مکرر، بوت استرپ و تقسیم‌های مکرر تمرین-تست با تعداد یکسانی از مدل‌ها مطابقت دارند و به‌طور متوسط حدود 5 برابر زمان بیشتری برای اتمام طول می‌کشد. LOOCV که به تعداد نمونه‌های موجود در مجموعه آموزشی مناسب است، 86 برابر بیشتر طول کشید و تنها زمانی باید در نظر گرفته شود که تعداد نمونه‌ها بسیار کم باشد.



شکل 4. 10: مشخصات عملکرد ماشین بردار پشتیبان غیرخطی بر مقادیر مختلف پارامتر هزینه برای مثال امتیازدهی اعتباری با استفاده از چندین روش نمونه‌گیری مجدد متفاوت. خطوط *عمودی* نشان دهنده *±* دو خطای استاندارد دقت است

توصیه‌های تقسیم داده ها

همانطور که قبلاً مورد بحث قرار گرفت، یک مورد فنی قوی باید در برابر یک مجموعه آزمایشی مستقل ایجاد شود:

* مجموعه آزمون یک ارزیابی واحد از مدل است و توانایی محدودی برای توصیف عدم قطعیت در نتایج دارد.
* مجموعه‌های آزمایشی نسبتاً بزرگ، داده‌ها را به گونه‌ای تقسیم می‌کنند که بایاس در برآوردهای عملکرد را افزایش می‌دهد.
* با حجم نمونه کوچک:
  + مدل ممکن است به هر نقطه داده ممکن نیاز داشته باشد تا مقادیر مدل را به اندازه کافی تعیین کند.
  + عدم قطعیت مجموعه آزمون می‌تواند به‌طور قابل‌توجهی زیاد باشد تا جایی که مجموعه‌های مختلف آزمایش ممکن است نتایج بسیار متفاوتی را تولید کنند.
* روش‌های نمونه‌گیری مجدد می‌تواند پیش‌بینی‌کننده‌های معقولی از عملکرد مدل در نمونه‌های آینده ایجاد کند.

هیچ روش نمونه‌گیری مجدد به‌طور یکنواخت بهتر از روش دیگر نیست. انتخاب باید با در نظر گرفتن چندین عامل انجام شود. اگر حجم نمونه‌ها کوچک است، به چند دلیل، اعتبارسنجی متقاطع 10 برابری را توصیه می‌کنیم: ویژگی‌های بایاس و واریانس خوب هستند و با توجه به حجم نمونه، هزینه‌های محاسباتی زیاد نیستند. اگر هدف انتخاب بین مدل‌ها باشد، به‌جای دریافت بهترین شاخص عملکرد، می‌توان یک مورد قوی برای استفاده از یکی از رویه‌های بوت استرپ ایجاد کرد، زیرا این روش‌ها واریانس بسیار کمی دارند. برای اندازه‌های نمونه بزرگ، تفاوت بین روش‌های نمونه‌گیری مجدد کمتر مشخص می‌شود و بازده محاسباتی اهمیت بیشتری پیدا می‌کند. در اینجا، اعتبارسنجی متقاطع 10 برابری ساده باید واریانس قابل قبول، بایاس کم و محاسبه نسبتاً سریع را ارائه دهد.

وارما و سایمون [( 2006](#bookmark1026) ) و [Boulesteix و Strobl ( 2009](#bookmark1011) ) خاطرنشان می‌کنند که یک بایاس بالقوه وجود دارد که می‌تواند هنگام تخمین عملکرد مدل در طول تنظیم پارامتر رخ دهد. فرض کنید که مدل نهایی مطابق با مقدار پارامتر تنظیم مرتبط با کمترین میزان خطا انتخاب شده است. این نرخ خطا پتانسیل خوش بینانه بودن را دارد زیرا یک کمیت تصادفی است که از مجموعه بالقوه بزرگی از پارامترهای تنظیم انتخاب می‌شود. تحقیقات آنها بر روی سناریوهایی با تعداد کم نمونه و تعداد زیادی پیش‌بینی متمرکز است که این مسأله را تشدید می‌کند. با این حال، برای مجموعه‌های آموزشی نسبتاً بزرگ، تجربه ما این است که این بایاس کوچک است. در بخش‌های بعدی، مقایسه‌هایی بین تخمین‌های نمونه‌برداری مجدد از عملکرد و برآوردهای حاصل از یک مجموعه آزمایشی انجام می‌شود. برای این مجموعه داده‌های خاص، *بایاس بهینه‌سازی اساسی* نیست.

4. 8 انتخاب بین مدل ها

هنگامی که تنظیمات پارامترهای تنظیم برای هر مدل مشخص شد، این سوال باقی می‌ماند: چگونه بین چندین مدل انتخاب کنیم؟ باز هم، این تا حد زیادی به ویژگی‌های داده‌ها و نوع سؤالاتی که به آنها پاسخ داده می‌شود بستگی دارد. با این حال، پیش‌بینی اینکه کدام مدل برای هدف مناسب‌تر است می‌تواند دشوار باشد. با توجه به این موضوع، ما طرح زیر را برای نهایی کردن نوع مدل پیشنهاد می‌کنیم:

1. با چندین مدل شروع کنید که کمترین تفسیر و انعطاف پذیری را دارند، مانند درختان تقویت شده (boosted trees) یا ماشین‌های بردار پشتیبان. در بسیاری از حوزه‌های مسأله، این مدل‌ها احتمال بالایی برای تولید نتایج بهینه تجربی (یعنی دقیق‌ترین) دارند.
2. مدل‌های ساده‌تری را که کمتر مات هستند (مثلاً جعبه‌های سیاه کامل نیستند)، مانند خطوط رگرسیون تطبیقی چند متغیره (MARS)، حداقل مربعات جزئی، مدل‌های افزایشی تعمیم‌یافته، یا مدل‌های ساده بیز بررسی کنید.
3. استفاده از ساده‌ترین مدل را در نظر بگیرید که به‌طور منطقی عملکرد روش‌های پیچیده تر را تقریب می‌کند.

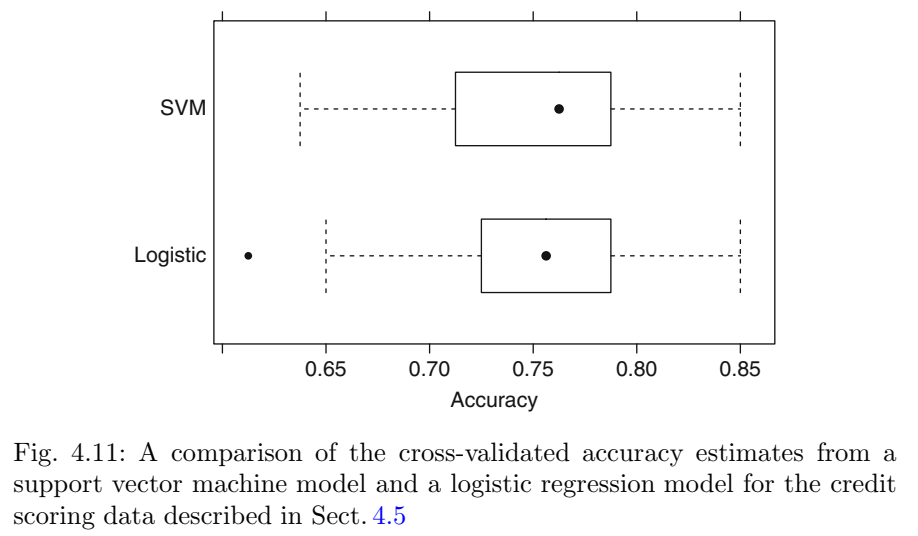
با استفاده از این روش، مدل‌ساز می‌تواند «**سقف عملکرد**» مجموعه داده‌ها را قبل از استقرار در یک مدل کشف کند. در بسیاری از موارد، طیف وسیعی از مدل‌ها از نظر عملکرد معادل خواهند بود، بنابراین متخصص می‌تواند مزایای روش‌های مختلف (مانند پیچیدگی محاسباتی، پیش‌بینی آسان، تفسیرپذیری) را ارزیابی کند. به‌عنوان مثال، یک ماشین بردار پشتیبان غیرخطی یا مدل جنگل تصادفی ممکن است دقت بالاتری داشته باشد، اما پیچیدگی و دامنه معادله پیش‌بینی ممکن است صادرات معادله پیش‌بینی به یک سامانه تولید را ممنوع کند. با این حال، اگر یک مدل قابل تفسیرتر، مانند مدل MARS، دقت مشابهی را به همراه داشت، اجرای معادله پیش‌بینی بی‌اهمیت خواهد بود و همچنین زمان اجرای بهتری خواهد داشت.

مدل طبقه‌بندی ماشین بردار پشتیبان امتیازدهی اعتباری را در نظر بگیرید که با استفاده از نمونه‌گیری مجدد در بخش مشخص شد.  [4. 6](#bookmark6) . با استفاده از اعتبارسنجی متقابل 10 برابری مکرر، دقت برای این مدل 75٪ با اکثر نتایج نمونه‌گیری مجدد بین 66٪ و 82٪ برآورد شد.

رگرسیون لجستیک (بخش.  [12. 2 )](#bookmark557) یک تکنیک ساده تر از ­مدل ماشین بردار پشتیبان غیر خطی برای تخمین مرز طبقه‌بندی است. هیچ پارامتر تنظیمی ندارد و معادله پیش‌بینی آن با استفاده از اکثر نرم‌افزارها ساده و آسان است. با استفاده از همان طرح اعتبار سنجی متقاطع، دقت تخمین زده شده برای این مدل 74. 9٪ بود که اکثر نتایج نمونه‌گیری مجدد بین 66٪ و 82٪ بود.

برای ارزیابی هر مدل از همان 50 نمونه مجدد استفاده شد. شکل [4. 11](#bookmark236) از نمودارهای جعبه برای نشان دادن توزیع تخمین‌های دقت نمونه برداری مجدد استفاده می‌کند. واضح است که با استفاده از یک مدل ساده تر برای این داده‌ها هیچ افت عملکردی وجود ندارد.

هاتورن و همکاران [( 2005](#bookmark1018) ) و [اوگستر و همکاران ( 2008](#bookmark1015) ) روش‌های آماری را برای مقایسه روش شناسی بر اساس نتایج نمونه‌گیری مجدد توصیف می‌کند. از آنجایی که دقت‌ها ­با استفاده از مجموعه داده‌های نمونه‌گیری مجدد یکسان اندازه‌گیری شدند، می‌توان از روش‌های آماری برای *مقایسه‌های زوجی* برای تعیین اینکه آیا تفاوت‌های بین مدل‌ها از نظر آماری معنادار است یا خیر، استفاده کرد. از آزمون *t* زوجی می‌توان برای ارزیابی این فرضیه استفاده کرد که مدل‌ها دارای دقت معادل (به‌طور متوسط) هستند یا به‌طور مشابه، میانگین تفاوت در دقت برای مجموعه داده‌های نمونه‌گیری مجدد صفر است. برای این دو مدل، میانگین تفاوت در دقت مدل 0. 1٪ بود که رگرسیون لجستیک نتایج بهتری را ارائه می‌کرد. فاصله اطمینان 95٪ ­برای این تفاوت ( *1.* 2 ٪،1٪) بود که نشان می‌دهد وجود دارد



شکل 4. 11: مقایسه تخمین‌های دقت تایید شده متقاطع از یک مدل ماشین بردار پشتیبان و یک مدل رگرسیون لجستیک برای داده‌های امتیازدهی اعتباری شرح داده شده در بخش.  [4. 5](#bookmark222)

هیچ مدرکی برای حمایت از این ایده وجود ندارد که دقت برای هر یک از مدل‌ها به‌طور قابل‌توجهی بهتر است. این به معنای شهودی است. دقت‌های نمونه برداری مجدد در شکل.  [محدوده 4. 11](#bookmark236) از 61. 3 % تا 85 %؛ با توجه به این مقدار تغییر در نتایج، بهبود 0. 1 درصدی دقت معنی‌دار نیست.

هنگامی که یک مدل به روش‌های مختلف مشخص می‌شود، این احتمال وجود دارد که مقایسه بین مدل‌ها منجر به نتایج متفاوتی شود. به‌عنوان مثال، اگر یک مدل برای پیش‌بینی دو کلاس ایجاد شود، **حساسیت و ویژگی** ممکن است برای مشخص کردن کارایی مدل‌ها مورد استفاده قرار گیرد (به فصل 13 مراجعه کنید).  [11 )](#bookmark497) . اگر مجموعه داده شامل رویدادهای بیشتری نسبت به غیر رویدادها باشد، حساسیت را می‌توان با دقت بیشتری نسبت به ویژگی تخمین زد. با افزایش دقت، احتمال بیشتری وجود دارد که مدل‌ها از نظر حساسیت نسبت به ویژگی متمایز شوند.

4. 9 محاسبات

زبان R برای نشان دادن تکنیک‌های مدل‌سازی استفاده می‌شود. بررسی مختصر R و کاربرد اساسی آن در ضمیمه B یافت می‌شود. کسانی که به R تازه وارد هستند باید این مطالب را قبل از ادامه بررسی کنند. بخش‌های زیر به ­توابع از بسته‌های AppliedPredictiveModeling، caret، Design، e1071، ipred و MASS اشاره می‌کنند. نحو با استفاده از مثال ساده دو کلاسه نشان داده شده در شکل نشان داده می‌شود.  [4. 2](#bookmark196) و [4. 3](#bookmark199) و داده‌های حاصل از مطالعه موردی امتیازدهی اعتباری.

تقسیم داده ها

داده‌های دو کلاسه نشان داده شده در شکل.  [4. 1](#bookmark196) در بسته AppliedPredictive- Modeling موجود است و می‌توان با استفاده از آن به دست آورد

*کتابخانه (AppliedPredictiveModeling)*

*داده (twoClassData)*

پیش‌بینی‌کننده‌های داده‌های نمونه در یک چارچوب داده به نام پیش‌بینی ذخیره می‌شوند. دو ستون برای پیش‌بینی‌کننده‌ها و 208 نمونه در ردیف وجود دارد. کلاس‌های نتیجه در یک بردار عاملی به نام کلاس‌ها قرار دارند.

*str (پیش‌بینی‌کننده‌ها)*

'data. frame': 208 obs. از 2 متغیر:

$ PredictorA: num 0. 158 0. 655 0. 706 0. 199 0. 395. . .

$ PredictorB: num 0. 1609 0. 4918 0. 6333 0. 0881 0. 4152. . .

*خیابان (کلاس ها)*

فاکتور w/ 2 سطح "Class1"، "Class2": 2 2 2 2 2 2 2 2 2 2. . .

نمونه تابع پایه R می‌تواند تقسیمات تصادفی ساده‌ای از داده‌ها ایجاد کند. برای ایجاد تقسیمات تصادفی طبقه‌بندی شده از داده‌ها (بر اساس کلاس ها)، می‌توان از تابع createDataPartition در بسته caret استفاده کرد. درصد داده‌هایی که به مجموعه آموزشی تخصیص داده می‌شود باید مشخص شود.

*# دانه اعداد تصادفی را تنظیم کنید تا بتوانیم نتایج را بازتولید کنیم*

*set. seed (1)*

*# به‌طور پیش فرض، اعداد به‌عنوان یک لیست برگردانده می‌شوند. استفاده كردن*

*# list = FALSE، ماتریسی از اعداد ردیف ایجاد می‌شود.*

*# این نمونه‌ها به مجموعه آموزشی اختصاص می‌یابد.*

*trainingRows <- createDataPartition(کلاس‌ها،*

*+ p = 0. 80،*

*+ لیست = نادرست)*

*سر (TrainingRows)*

نمونه 1

[1،] 99

[2،] 100

[3،] 101

[4،] 102

[5،] 103

[6،] 104

*# داده‌ها را در اشیاء برای آموزش استفاده کنید*

*# زیر تنظیم عدد صحیح.*

*trainPredictors <- predictors[trainingRows, ]*

*trainClasses <- classes[trainingRows]*

*# همین کار را برای مجموعه تست با استفاده از اعداد صحیح منفی انجام دهید.*

*testPredictors <- predictors[-trainingRows, ]*

*testClasses <- classes[-trainingRows]* *> str(trainPredictors)*

'data. frame': 167 obs. از 2 متغیر:

$ PredictorA: num 0. 226 0. 262 0. 52 0. 577 0. 426. . .

$ PredictorB: تعداد 0. 291 0. 225 0. 547 0. 553 0. 321. . .

*> str(testPredictors)*

'data. frame': 41 obs. از 2 متغیر:

$ PredictorA: num 0. 0658 0. 1056 0. 2909 0. 4129 0. 0472. . .

$ PredictorB: num 0. 1786 0. 0801 0. 3021 0. 2869 0. 0414. . .

برای تولید یک مجموعه آزمایشی با استفاده از نمونه‌گیری حداکثر تفاوت، می‌توان از تابع ­caret maxdissim برای نمونه‌برداری متوالی داده‌ها استفاده کرد.

نمونه‌گیری مجدد

بسته caret عملکردهای مختلفی برای تقسیم داده‌ها دارد. به‌عنوان مثال، برای استفاده از تقسیم‌های آموزشی/تست مکرر، تابع createDataPartition می‌تواند دوباره با یک آرگومان اضافی به نام times برای ایجاد چند تقسیم استفاده شود.

|  |
| --- |
| *> set. seed(1)* |
| *> # برای مثال، اطلاعات مورد نیاز برای سه را ایجاد کنید* |
| *> # نسخه نمونه برداری مجدد از مجموعه آموزشی.* |
| *> repeatedSplits <- createDataPartition(trainClasses, p = 0. 80,* |
| *+ بار = 3)* |
| *> str (تکرار تقسیمات)* |
| لیست 3 |
| $ Resample1: int [1:135] 1 2 3 4 5 6 7 9 11 12. . . |
| $ Resample2: int [1:135] 4 6 7 8 9 10 11 12 13 14. . . |
| $ Resample3: int [1:135] 2 3 4 6 7 8 9 10 11 12. . . |

به‌طور مشابه، بسته caret دارای توابع createResamples (برای بوت استرپ)، createFolds (برای اعتبارسنجی متقاطع *k* -old) و createMultiFolds (برای اعتبارسنجی متقابل مکرر) است. برای ایجاد شاخص‌هایی برای اعتبارسنجی متقاطع 10 برابری،

*> set. seed(1)*

|  |  |
| --- | --- |
| *> cvSplits <- createFolds(trainClasses,* | *k = 10،* |
| *+ بازگشت آموزش =* | *درست است، واقعی)* |
| *> str(cvSplits)* |  |
| لیست 10 |  |
| $ Fold01: int [1:151] 1 2 3 4 5 6 7 8 | 9 11. . . |
| $ Fold02: int [1:150] 1 2 3 4 5 6 8 9 | 10 12. . . |
| $ Fold03: int [1:150] 1 2 3 4 6 7 8 10 | 11 13. . . |
| $ Fold04: int [1:151] 1 2 3 4 5 6 7 8 | 9 10. . . |
| $ Fold05: int [1:150] 1 2 3 4 5 7 8 9 | 10 11. . . |
| $ Fold06: int [1:150] 2 4 5 6 7 8 9 10 | 11 12. . . |
| $ Fold07: int [1:150] 1 2 3 4 5 6 7 8 | 9 10. . . |
| $ Fold08: int [1:151] 1 2 3 4 5 6 7 8 | 9 10. . . |
| $ Fold09: int [1:150] 1 3 4 5 6 7 9 10 | 11 12. . . |
| $ Fold10: int [1:150] 1 2 3 5 6 7 8 9 | 10 11. . . |
| *> # اولین مجموعه اعداد ردیف را از لیست دریافت کنید* | |
| *> fold1 <- cvSplits[[1]]* |  |

برای دریافت 90 درصد اول داده‌ها (نخستین):

*cvPredictors1 <- trainPredictors[fold1,]*

*cvClasses1 <- trainClasses[fold1]*

*nrow (trainPredictors)*

[1] 167

*> nrow(cvPredictors1)*

[1] 151

در عمل، توابع مورد بحث در بخش بعدی را می‌توان برای ­ایجاد خودکار مجموعه داده‌های نمونه برداری مجدد، متناسب با مدل‌ها و ارزیابی عملکرد استفاده کرد.

**ساختمان مدل پایه در R**

اکنون که مجموعه‌های آموزشی و آزمایشی داریم، می‌توانیم مدل طبقه‌بندی 5 نزدیک‌ترین همسایه (شکل 2) را برازش کنیم.  [4. 3 )](#bookmark199) به داده‌های آموزشی و استفاده از آن برای پیش‌بینی مجموعه آزمون. چندین تابع R برای ساخت این مدل وجود دارد: تابع knn در بسته MASS، تابع ipredknn در بسته ipred و تابع knn3 در caret. تابع knn3 می‌تواند پیش‌بینی‌کننده‌های کلاس و همچنین نسبت همسایگان را برای هر کلاس تولید کند.

دو قرارداد اصلی برای مشخص کردن مدل‌ها در R وجود دارد : رابط فرمول و رابط غیر فرمول (یا ماتریس). برای اولی، پیش‌بینی‌کننده‌ها به صراحت فهرست شده اند. یک فرمول اصلی R دو طرف دارد: سمت چپ نشان دهنده نتیجه است و سمت راست نحوه استفاده از پیش‌بینی‌کننده‌ها را توصیف می‌کند. اینها با یک تار ( *~* ) از هم جدا می‌شوند. به‌عنوان مثال، فرمول

*ModelFunction(قیمت ~ تعداد اتاق خوابها + تعداد حمام + هکتار،*

*+ داده = مسکن داده)*

می تواند قیمت پایانی یک خانه را با استفاده از سه شاخص کمی پیش‌بینی کند. فرمول y *~.* می‌تواند برای نشان دادن اینکه همه ستون‌های مجموعه داده (به جز y ) باید به‌عنوان پیش‌بینی استفاده شوند استفاده شود. رابط فرمول دارای امکانات بسیاری است. برای مثال، تبدیل‌هایی مانند log(acre) را می‌توان در خط مشخص کرد. متأسفانه، R به‌طور مؤثر اطلاعات مربوط به فرمول را ذخیره نمی‌کند. استفاده از این رابط با مجموعه داده‌هایی که حاوی تعداد زیادی پیش‌بینی هستند ممکن است محاسبات را به‌طور غیر ضروری کُند نماید.

رابط غیر فرمول پیش‌بینی‌کننده‌های مدل را با استفاده از یک ماتریس یا قاب داده مشخص می‌کند (همه پیش‌بینی‌کننده‌های موجود در شی در مدل استفاده می‌شوند). داده‌های نتیجه معمولاً به‌عنوان یک شی برداری به مدل منتقل می‌شوند. مثلا،

*modelFunction (x = خانه پیش‌بینی، y = قیمت)*

توجه داشته باشید که همه توابع R دارای هر دو رابط نیستند.

برای knn3، می‌توانیم مدل 5 نزدیک‌ترین همسایه را با تخمین بزنیم

*trainPredictors <- as. matrix(trainPredictors)*

*knnFit <- knn3(x = trainPredictors، y = trainClasses، k = 5)*

*knnFit*

مدل طبقه‌بندی 5-نزدیک‌ترین همسایه

صدا زدن:

knn3. matrix(x = trainPredictors، y = trainClasses، k = 5)

توزیع کلاس مجموعه آموزشی:

کلاس 1 کلاس 2 89 78

در این مرحله، شی knn3 آماده پیش‌بینی نمونه‌های جدید است. برای تخصیص نمونه‌های جدید به کلاس‌ها، از روش پیش‌بینی با مدل Object استفاده می‌شود. کنوانسیون استاندارد است

*testPredictions <- predict(knnFit، newdata = testPredictors،*

*+ نوع = "کلاس")*

*سر (تست پیش‌بینی ها)*

[1] Class2 Class2 Class1 Class1 Class2 Class2

سطوح: کلاس 1 کلاس 2

*> str(testPredictions)*

فاکتور w/ 2 سطح "Class1"، "Class2": 2 2 1 1 2 2 2 2 2 2. . .

مقدار آرگومان نوع در توابع مدلسازی مختلف متفاوت است.

تعیین پارامترهای تنظیم

برای انتخاب پارامترهای تنظیم با استفاده از نمونه‌گیری مجدد، مجموعه‌ای از مقادیر نامزد با استفاده از نمونه‌های مجدد متفاوت داده‌ها ارزیابی می‌شوند. برای درک رابطه بین عملکرد و مقادیر پارامتر می‌توان یک پروفایل ایجاد کرد. R دارای چندین تابع و بسته برای این کار است. بسته e1071 حاوی تابع تنظیم است که می‌تواند چهار نوع مدل را در طیف وسیعی از پارامترها ارزیابی کند. به‌طور مشابه، تابع خطا در بسته ipred می‌تواند مدل‌های تک را مجدداً نمونه برداری کند. تابع آموزش در پکیج caret دارای ماژول‌های داخلی برای 144 مدل است و شامل قابلیت‌هایی برای روش‌های مختلف نمونه‌گیری مجدد، معیارهای عملکرد و الگوریتم‌هایی برای انتخاب بهترین مدل از پروفایل است. این تابع همچنین دارای قابلیت‌هایی برای پردازش موازی است به‌طوری که مدل‌های نمونه‌برداری مجدد می‌توانند در چندین کامپیوتر یا پردازنده اجرا شوند. تمرکز ما بر عملکرد آموزش خواهد بود.

بخش [4. 6](#bookmark6) تنظیم پارامتر مصور برای ماشین بردار پشتیبان با استفاده از داده‌های امتیازدهی اعتباری. با استفاده از نمونه‌گیری مجدد، مقداری از پارامتر هزینه برآورد شد. همانطور که در فصل‌های بعدی بحث شد، مدل SVM با نوع *عملکرد هسته* که مدل استفاده می‌کند مشخص می‌شود. به‌عنوان مثال، تابع هسته خطی یک رابطه خطی بین پیش‌بینی‌کننده‌ها و نتیجه را مشخص می‌کند. برای داده‌های امتیازدهی اعتباری، تابع هسته تابع پایه شعاعی (RBF) استفاده شد. این تابع هسته دارای یک پارامتر تنظیم اضافی مرتبط با آن است که به‌عنوان *یک نشان داده شده است که* بر صاف بودن مرز تصمیم تأثیر می‌گذارد. به‌طور معمول، چندین ترکیب از هر دو پارامتر تنظیم با استفاده از نمونه‌گیری مجدد ارزیابی می‌شوند. با این حال، [کاپوتو و همکاران ( 2002](#bookmark1012) ) یک فرمول تحلیلی را توصیف می‌کند که می‌تواند برای بدست آوردن تخمین‌های منطقی از *یک مورد استفاده قرار گیرد.* آموزش تابع caret از این روش برای تخمین پارامتر هسته استفاده می‌کند و تنها پارامتر هزینه را برای تنظیم باقی می‌گذارد.

برای تنظیم یک مدل SVM با استفاده از نمونه‌های مجموعه آموزشی امتیازدهی اعتباری، می‌توان از تابع آموزش استفاده کرد. هم پیش‌بینی‌کننده‌های مجموعه آموزشی و هم نتیجه در یک چارچوب داده R به نام GermanCreditTrain قرار دارند.

*> کتابخانه (کارت)*

*> داده (اعتبار آلمانی)*

فهرست فصل‌های بسته AppliedPredictiveModeling حاوی کد ایجاد مجموعه‌های آموزشی و آزمایشی است. این مجموعه داده‌ها به ترتیب در فریم‌های داده GermanCreditTrain و GermanCreditTest قرار دارند.

ما از تمام پیش‌بینی‌کننده‌ها برای مدل‌سازی نتیجه استفاده خواهیم کرد. برای این کار از رابط فرمول با فرمول Class *~ استفاده می‌کنیم.* کلاس‌ها در ستون قاب داده به نام کلاس ذخیره می‌شوند. ابتدایی‌ترین فراخوانی تابع خواهد بود *> set. seed(1056)*

*> svmFit <- train(Class ~. ,*

*> داده = GermanCreditTrain،*

*> # آرگومان "روش" نوع مدل را نشان می‌دهد.*

*> # برای لیست مدل‌های موجود به ?train مراجعه کنید.*

*> روش = "svmRadial")*

با این حال، ما می‌خواهیم محاسبات را با نادیده گرفتن چندین مقدار پیش فرض تنظیم کنیم. ابتدا، می‌خواهیم داده‌های پیش‌بینی را با مرکزیت و مقیاس‌بندی مقادیر آن‌ها، از قبل پردازش کنیم. برای این کار می‌توان از آرگومان preProc استفاده کرد:

*> set. seed(1056)*

*> svmFit <- train(Class ~. ,*

*> داده = GermanCreditTrain،*

*> روش = "svmRadial"،*

*> preProc = c("مرکز"، "مقیاس") )*

همچنین برای این تابع، کاربر می‌تواند مقادیر دقیق هزینه را برای بررسی مشخص کند. علاوه بر این، تابع دارای الگوریتم‌هایی برای تعیین مقادیر معقول برای بسیاری از مدل‌ها است. با استفاده از گزینه tuneLength = 10، مقادیر هزینه 2 *-* 2، 2 *-* 2. *. .* 2 7 ارزیابی می‌شوند.

*> set. seed(1056)*

*> svmFit <- train(Class ~. ,*

*> داده = GermanCreditTrain،*

*> روش = "svmRadial"،*

*> preProc = c("مرکز"، "مقیاس")،*

*طول کوک = 10 )*

به‌طور پیش فرض، بوت استرپ اولیه برای محاسبه معیارهای عملکرد استفاده می‌شود. اعتبارسنجی متقابل 10 برابری مکرر را می‌توان با تابع trainControl مشخص کرد. سپس دستور نهایی *> set. seed(1056) است.*

*svmFit <- train(Class ~. ,*

|  |  |
| --- | --- |
| *>*  *>*  *>*  *>*  *>*  *>* | *داده = GermanCreditTrain،*  *روش = "svmRadial"،*  *preProc = c("مرکز"، "مقیاس")،*  *طول کوک = 10،*  *trControl = trainControl (روش = "repeatedcv"، تکرار = 5) )* |

*svmFit*

800 نمونه

41 پیش‌بینی

2 کلاس: "بد"، "خوب"

پیش پردازش: متمرکز، مقیاس شده

نمونه‌گیری مجدد: اعتبارسنجی متقاطع (10 برابر، 5 بار تکرار)

خلاصه حجم نمونه: 720، 720، 720، 720، 720، 720،.

نمونه برداری مجدد از نتایج در پارامترهای تنظیم:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| سی | دقت | کاپا | دقت | SD Kappa SD |
| 0. 25 | 0. 7 | 0 | 0 | 0 |
| 0. 5 | 0. 724 | 0. 141 | 0. 0218 | 0. 0752 |
| 1 | 0. 75 | 0. 326 | 0. 0385 | 0. 106 |
| 2 | 0. 75 | 0. 363 | 0. 0404 | 0. 0984 |
| 4 | 0. 754 | 0. 39 | 0. 0359 | 0. 0857 |
| 8 | 0. 738 | 0. 361 | 0. 0404 | 0. 0887 |
| 16 | 0. 738 | 0. 361 | 0. 0458 | 0. 1 |
| 32 | 0. 732 | 0. 35 | 0. 043 | 0. 0928 |
| 64 | 0. 732 | 0. 352 | 0. 0453 | 0. 0961 |
| 128 | 0. 731 | 0. 349 | 0. 0451 | 0. 0936 |
| تنظیم | پارامتر | "سیگما" | برگزار شد | ثابت در مقدار 0. 0202 |

برای انتخاب مدل بهینه با استفاده از بیشترین مقدار از دقت استفاده شد. مقادیر نهایی مورد استفاده برای مدل C = 4 و سیگما = 0. 0202 بود.

در تحلیل اصلی از یک عدد تصادفی متفاوت و مجموعه‌ای از مقادیر هزینه استفاده شد، بنابراین نتایج دقیقاً مشابه نتایج نشان داده شده در بخش نیست.  [4. 6 .](#bookmark6) با استفاده از رویکرد "انتخاب بهترین"، یک مدل نهایی برای همه 800 نمونه مجموعه آموزشی با *مقدار* 0. 0202 و ارزش هزینه 4 مناسب بود. روش نمودار می‌تواند برای تجسم نمایه عملکرد استفاده شود. شکل [4. 12](#bookmark252) یک مثال تجسم ایجاد شده از نحو را نشان می‌دهد

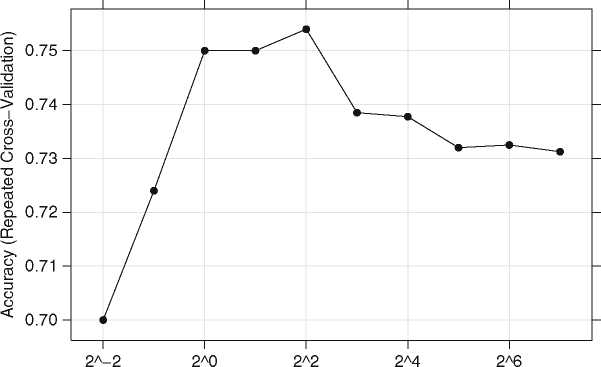
*# نمودار خطی از عملکرد متوسط*

*نمودار (svmFit، مقیاس = لیست (x = لیست (log = 2)))*

برای پیش‌بینی نمونه‌های جدید با این مدل، روش پیش‌بینی نامیده می‌شود

*predictedClasses <- predict(svmFit, GermanCreditTest)*

*str (کلاس‌های پیش‌بینی شده)*



Cost

Fig. 4.12: A visualization of the average performance profile of an SVM clas­sification model produced from the plot method for the train class

فاکتور با 2 سطح "بد"، "خوب": 1 1 2 2 1 2 2 2 1 1. . .

*# از گزینه "type" برای بدست آوردن احتمالات کلاس استفاده کنید*

*predictedProbs <- predict(svmFit, newdata = GermanCreditTest,*

*+ نوع = "مسأله")*

*سر (پیش‌بینی شده)*

بد 1 0. 5351870 2 0. 5084049 3 0. 3377344 4 0. 1092243 5 0. 6024404 6 0. 1339467

Good 0.4648130 0.4915951 0.6622656 0.8907757 0.3975596 0.8660533

R دیگری نیز وجود دارند که می‌توانند عملکرد را از طریق نمونه‌گیری مجدد تخمین بزنند. تابع اعتبارسنجی در بسته Design و تابع خطا در بسته ipred را می‌توان برای تخمین عملکرد یک مدل با یک مجموعه نامزد واحد از پارامترهای تنظیم استفاده کرد. عملکرد تنظیم بسته e1071 همچنین می‌تواند تنظیمات پارامتر را با استفاده از نمونه برداری مجدد تعیین کند.

**مقایسه بین مدل ها**

در بخش [4. 6 ،](#bookmark6) مدل SVM با مدل رگرسیون لجستیک مقایسه شد. در حالی که رگرسیون لجستیک پایه هیچ پارامتر تنظیمی ندارد، هنوز هم می‌توان از نمونه برداری مجدد برای مشخص کردن عملکرد مدل استفاده کرد. تابع آموزش بار دیگر با آرگومان روش متفاوت " glm " ( برای مدل‌های خطی تعمیم یافته) استفاده می‌شود. از همان مشخصات نمونه‌گیری مجدد استفاده می‌شود و از آنجایی که ­دانه‌های عدد تصادفی قبل از مدل‌سازی تنظیم شده است، نمونه‌های مجدد دقیقاً مشابه نمونه‌های مدل SVM هستند.

*> set. seed(1056)*

*> logisticReg <- train(Class ~. ,*

|  |  |
| --- | --- |
| *+*  *+*  *+*  *+* | *داده = GermanCreditTrain،*  *روش = "glm"،*  *trControl = trainControl (روش = "repeatedcv"، تکرار = 5))* |

*logisticReg*

800 نمونه

41 پیش‌بینی

2 کلاس: "بد"، "خوب"

بدون پیش پردازش

نمونه‌گیری مجدد: اعتبارسنجی متقاطع (10 برابر، 5 بار تکرار)

خلاصه حجم نمونه: 720، 720، 720، 720، 720، 720،. . .

نتایج نمونه‌گیری مجدد

Accuracy Kappa Accuracy SD Kappa SD 0. 749 0. 365 0. 0516 0. 122

برای مقایسه این دو مدل بر اساس آمار اعتبارسنجی متقاطع آنها، تابع نمونه‌های مجدد را می‌توان با مدل‌هایی استفاده کرد که مجموعه مشترکی از مجموعه داده‌های نمونه‌گیری مجدد را به اشتراک می‌گذارند. از آنجایی که دانه اعداد تصادفی قبل از اجرای مدل‌های SVM و لجستیک مقداردهی اولیه شد، اندازه گیری‌های دقت جفتی برای هر مجموعه داده وجود دارد. ابتدا یک شی نمونه مجدد از مدل‌ها ایجاد می‌کنیم :

*resamp <- resamples(list(SVM = svmFit، Logistic = logisticReg))*

*خلاصه (resamp)*

صدا زدن:

summary. resamples(object = resamp)

مدل ها: SVM، لجستیک

تعداد نمونه مجدد: 50

دقت

حداقل 1 ق. میانگین میانه 3 ق. حداکثر NA

SVM 0. 6500 0. 7375 0. 7500 0. 754 0. 7625 0. 85 0

لجستیک 0. 6125 0. 7250 0. 7562 0. 749 0. 7844 0. 85 0

کاپا

حداقل 1 ق. میانگین میانه 3 ق. حداکثر NA

SVM 0. 18920 0. 3519 0. 3902 0. 3897 0. 4252 0. 5946 0

لجستیک 0. 07534 0. 2831 0. 3750 0. 3648 0. 4504 0. 6250 0

خلاصه نشان می‌دهد که توزیع عملکرد بسیار مشابه است. ستون NA مربوط به مواردی است که مدل‌های نمونه‌گیری مجدد شکست خوردند (معمولاً ­به دلیل مسائل عددی). کلاس resamples چندین روش برای تجسم مقادیر جفت شده دارد ( برای لیستی از انواع نمودار به ?xyplot. resamples مراجعه کنید). برای ارزیابی تفاوت‌های احتمالی بین مدل‌ها، از روش diff استفاده می‌شود:

*> modelDifferences <- diff(resamp)*

*> خلاصه (تفاوتهای مدل)*

صدا زدن:

summary. diff. resamples(object = modelDifferences)

تنظیم p-value: بونفرونی

مورب بالا: تخمین تفاوت

مورب پایین: p-value برای H0: تفاوت = 0

دقت

SVM لجستیک

SVM 0. 005

لجستیک 0. 5921

کاپا

SVM لجستیک

SVM 0. 02498

لجستیک 0. 2687

*p برای مقایسه* مدل بزرگ است (0. 592 برای دقت و 0. 269 برای Kappa) که نشان می‌دهد مدل‌ها هیچ تفاوتی در عملکرد نشان نمی‌دهند.

تمرینات

مجموعه داده ژانر موسیقی که در بخش توضیح داده شده است را در نظر بگیرید.  [1. 4 .](#bookmark77) هدف این داده‌ها استفاده از پیش‌بینی‌کننده‌ها برای طبقه‌بندی نمونه‌های موسیقی در ژانر موسیقی مناسب است.

از چه روش(های) تقسیم داده برای این داده‌ها استفاده می‌کنید؟ توضیح.

با استفاده از ابزارهای شرح داده شده در این فصل، کدی را برای پیاده‌سازی رویکرد(های) خود ارائه دهید.

مجموعه داده‌های نفوذپذیری شرح داده شده در بخش را در نظر بگیرید.  [1. 4](#bookmark77) . هدف این داده‌ها استفاده از پیش‌بینی‌کننده‌ها برای مدل‌سازی نفوذپذیری ترکیبات است.

از چه روش(های) تقسیم داده برای این داده‌ها استفاده می‌کنید؟ توضیح.

با استفاده از ابزارهای شرح داده شده در این فصل، کدی را برای پیاده‌سازی رویکرد(های) خود ارائه دهید.

حداقل مربعات جزئی (بخش [6. 3 )](#bookmark8) برای مدلسازی بازده یک فرآیند تولید شیمیایی استفاده شد (بخش.  [1. 4 )](#bookmark77) . داده‌ها را می‌توان در بسته AppliedPredictiveModeling پیدا کرد و با استفاده از آن می‌توان آن را بارگیری کرد

نمونه‌گیری مجدد از *R* 2

|  |  |  |
| --- | --- | --- |
| اجزاء | Mean Std | . خطا |
| 1 | 0. 444 | 0. 0272 |
| 2 | 0. 500 | 0. 0298 |
| 3 | 0. 533 | 0. 0302 |
| 4 | 0. 545 | 0. 0308 |
| 5 | 0. 542 | 0. 0322 |
| 6 | 0. 537 | 0. 0327 |
| 7 | 0. 534 | 0. 0333 |
| 8 | 0. 534 | 0. 0330 |
| 9 | 0. 520 | 0. 0326 |
| 10 | 0. 507 | 0. 0324 |

*> کتابخانه (AppliedPredictiveModeling)*

*> داده‌ها (فرایند تولید شیمیایی)*

هدف از این تحلیل یافتن تعداد مؤلفه‌های PLS است که *R بهینه را به دست می‌دهد*[[12]](#footnote-12) مقدار (بخش 5. 1 ). مدل‌های PLS با 1 تا 10 جزء هر کدام با استفاده از پنج تکرار اعتبارسنجی متقاطع 10 برابری ارزیابی شدند و نتایج در جدول زیر ارائه شده‌اند:

با استفاده از روش «خطای یک استاندارد»، چه تعداد مؤلفه PLS مقرون‌به‌صرفه‌ترین مدل را ارائه می‌کند؟

مقادیر تحمل را برای این مثال محاسبه کنید. اگر 10 درصد ضرر در *R2* قابل قبول است، پس تعداد بهینه اجزای PLS چقدر است؟

چندین مدل دیگر (در قسمت بحث شده است [II )](#bookmark259) با درجات مختلف ­پیچیدگی آموزش و تنظیم شدند و نتایج در شکل 1 ارائه شده است.  [4. 13 .](#bookmark256) اگر هدف انتخاب مدلی است که *R* 2 را بهینه می‌کند، پس کدام مدل(های) را انتخاب می‌کنید و چرا؟

زمان پیش‌بینی و همچنین پیچیدگی مدل (بخش.  [4. 8 )](#bookmark232) عوامل دیگری هستند که هنگام انتخاب مدل(های) بهینه باید در نظر گرفته شوند. با توجه به ­زمان پیش‌بینی هر مدل، پیچیدگی مدل و *R2* تخمین زده می‌شود، کدام مدل(های) را انتخاب می‌کنید و چرا؟

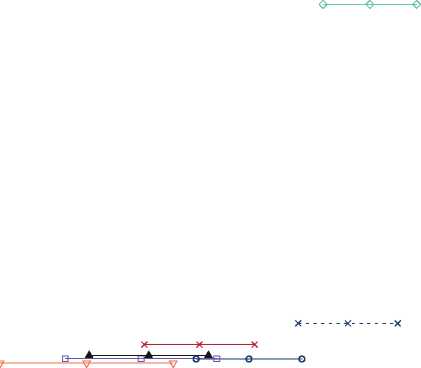
Brodnjak-Vonina و همکاران. [( 2005](#bookmark1012) ) یک روش برای آزمایشگاه‌های مواد غذایی ­برای تعیین نوع روغن از یک نمونه توسعه دادند. در روش خود، آنها از یک گاز کروماتوگراف (ابزاری که مواد شیمیایی را در یک نمونه جدا می‌کند) برای اندازه‌گیری هفت اسید چرب مختلف در یک روغن استفاده کردند. سپس از این اندازه‌گیری‌ها برای پیش‌بینی نوع روغن در نمونه‌های غذا استفاده می‌شود. برای ایجاد مدل خود از 96 نمونه 2 استفاده کردند از هفت نوع روغن

این داده‌ها را می‌توان در بسته caret با استفاده از داده (روغن) پیدا کرد. انواع روغن در یک متغیر فاکتور به نام oilType قرار دارند. انواع آن کدو حلوایی است

[رگرسیون خطی تقویت شده](#bookmark319)  [o](#bookmark319)

Knn x

رگرسیون خطی v

PLS □

Random Forests Regression Tree SVM

w

200

150

**<D**

**E**

100

o

**o**

TS

**2**

**CL**

50

0.4

0.5 0.6 0.7

R مربعی مجدد با 95٪ CI

شکل 4. 13: نموداری از عملکرد مدل تخمینی در برابر زمان برای پیش‌بینی 500000 نمونه جدید با استفاده از داده‌های تولید شیمیایی (با کد A )، آفتابگردان ( B )، بادام زمینی ( C )، زیتون ( D )، سویا ( E )، کلزا ( F ) و ذرت ( G ). در R،

*داده (روغن)*

*str (نوع روغن)*

فاکتور با 7 سطح "A"، "B"، "C"، "D"،. . . : 1 1 1 1 1 1 1 1 1 1. . .

*جدول (نوع روغن)*

نوع روغن

ABCDEFG

37 26 3 7 11 10 2

از تابع نمونه در پایه R برای ایجاد یک نمونه کاملا تصادفی از 60 روغن استفاده کنید. فرکانس‌های نمونه تصادفی چقدر با نمونه‌های اصلی مطابقت دارد؟ برای درک تغییرات در فرآیند نمونه برداری، این روش را چندین بار تکرار کنید.

از تابع بسته caret createDataPartition برای ایجاد یک نمونه تصادفی طبقه‌بندی شده استفاده کنید. این در مقایسه با نمونه‌های کاملاً تصادفی چگونه است ­؟

با چنین اندازه نمونه کوچک، چه گزینه‌هایی برای تعیین عملکرد مدل وجود دارد؟ آیا باید از مجموعه تست استفاده شود؟

یک روش برای درک عدم قطعیت یک مجموعه تست استفاده از فاصله اطمینان است. برای به دست آوردن فاصله اطمینان برای دقت کلی، می‌توان از ­تابع R مبتنی بر binom. test استفاده کرد. برای محاسبه فاصله، کاربر باید تعداد نمونه‌ها و تعداد طبقه‌بندی شده را به درستی وارد کند. به‌عنوان مثال، فرض کنید یک مجموعه آزمایشی شامل 20 نمونه روغن کنار گذاشته شد و 76 نمونه برای آموزش مدل استفاده شد. برای اندازه مجموعه تست و مدلی که حدود 80 درصد دقیق است (16 از 20 صحیح)، فاصله اطمینان با استفاده از

*> binom. test(16، 20)*

تست دو جمله‌ای دقیق

داده ها: 16 و 20

تعداد موفقیت = 16، تعداد آزمایش‌ها = 20، p-value = 0. 01182 فرضیه جایگزین: احتمال موفقیت واقعی برابر با 0. 5 نیست 95 درصد فاصله اطمینان:

0. 563386 0. 942666

برآوردهای نمونه: احتمال موفقیت 0. 8

در این حالت، عرض فاصله اطمینان 95 درصد 37. 9 درصد است. برای درک مبادله بین عدم قطعیت در نتایج، عملکرد مدل و اندازه مجموعه تست، اندازه‌های نمونه و نرخ‌های دقت مختلف را امتحان کنید.

قسمت دوم

مدل‌های رگرسیون

فصل 5

سنجش عملکرد در رگرسیون

مدل ها

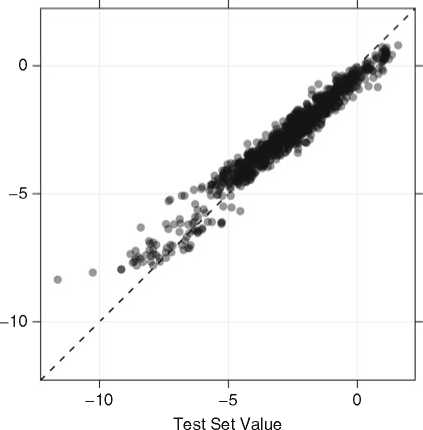
برای مدل‌هایی که یک پیامد عددی را پیش‌بینی می‌کنند، معمولاً از برخی معیارهای دقت برای ارزیابی اثربخشی مدل استفاده می‌شود. با این حال، روش‌های مختلفی برای اندازه‌گیری دقت وجود دارد که هر کدام تفاوت‌های ظریف خاص خود را دارند. برای درک نقاط قوت و ضعف یک مدل خاص، اتکای صرف بر تنها یک معیار مسأله ساز است. تجسم برازش مدل، به ویژه نمودارهای باقیمانده، برای درک اینکه آیا مدل برای هدف مناسب است، حیاتی است. این تکنیک‌ها در این فصل مورد بحث قرار می‌گیرند.

معیارهای کمّی عملکرد

هنگامی که نتیجه یک عدد است، رایج‌ترین روش برای توصیف ­قابلیت‌های پیش‌بینی یک مدل استفاده از ریشه میانگین مربعات خطا (RMSE) است. این معیار تابعی از باقیمانده‌های مدل است که ­مقادیر مشاهده شده منهای پیش‌بینی شده مدل هستند. میانگین مربعات خطا (MSE) با مجذور کردن باقیمانده‌ها و جمع آنها محاسبه می‌شود. سپس RMSE با گرفتن جذر MSE محاسبه می‌شود تا در واحدهای مشابه داده‌های اصلی قرار گیرد. مقدار معمولاً به صورت فاصله (به‌طور متوسط) باقیمانده‌ها از صفر یا به‌عنوان فاصله متوسط بین مقادیر مشاهده شده و پیش‌بینی‌کننده‌های مدل تفسیر می‌شود.

یکی دیگر از معیارهای رایج، ضریب تعیین است که معمولاً به صورت *R نوشته می‌*شود. این مقدار را می‌توان به‌عنوان نسبت ­شکل‌گیری در داده‌هایی که توسط مدل توضیح داده می‌شود تفسیر کرد. بنابراین، یک *R2* مقدار 0. 75 نشان می‌دهد که مدل می‌تواند سه چهارم تغییرات در نتیجه را توضیح دهد. چندین فرمول برای محاسبه این کمیت وجود دارد [( Kvalseth](#bookmark1020) [1985](#bookmark1020) )، اگرچه ساده‌ترین نسخه ضریب همبستگی بین مقادیر مشاهده شده و پیش‌بینی شده (معمولاً با *R نشان داده می‌شود* ) را پیدا کرده و آن را مربع می‌کند.

در حالی که این یک آمار به راحتی قابل تفسیر است، متخصص باید توجه داشته باشد ­*R2* معیاری برای همبستگی است، نه دقت. شکل [5. 1](#bookmark267) نشان می‌دهد



شکل 5. 1: نموداری از نتایج مشاهده شده و پیش‌بینی شده که در آن *R* 2 متوسط (51%) است، اما پیش‌بینی‌کننده‌ها به‌طور یکسان دقیق نیستند. خط مرجع *خاکستری مورب* نشان می‌دهد که مقادیر مشاهده شده و پیش‌بینی شده در کجا برابر هستند

مثالی که در آن *R* 2 بین مقادیر مشاهده‌شده و پیش‌بینی‌شده زیاد است (51%)، اما مدل تمایل به پیش‌بینی بیش از حد مقادیر پایین و کمتر پیش‌بینی مقادیر بالا دارد. این پدیده می‌تواند برای برخی از مدل‌های رگرسیون مبتنی بر درخت که در فصل مورد بحث قرار گرفت، مشترک باشد.  [8](#bookmark388) . بسته به زمینه، این ­بایاس موضوعی سامانه‌ی در پیش‌بینی‌کننده‌ها ممکن است قابل قبول باشد اگر مدل به خوبی کار کند.

همچنین مهم است که بدانیم *R* 2 بستگی به تغییرات متغیر خروجی دارد. با استفاده از این تفسیر که این آمار نسبت واریانس توضیح داده شده توسط مدل را اندازه می‌گیرد، باید به خاطر داشت که مخرج آن نسبت با استفاده از واریانس نمونه متغیر نتیجه محاسبه می‌شود. برای مثال، فرض کنید متغیر نتیجه در مجموعه آزمون، دارای واریانس 4. 2 است. اگر RMSE یک مدل پیش‌بینی 1 بود،*R2* تقریباً 76 درصد خواهد بود. اگر مجموعه تست دیگری با همان RMSE داشتیم، اما نتایج آزمون کمتر متغیر بود، نتایج بدتر به نظر می‌رسید. به‌عنوان مثال، اگر واریانس مجموعه آزمون 3 بود،*R2* 67 درصد خواهد بود.

از نظر عملی، این وابستگی به واریانس نتیجه نیز می‌تواند تأثیر شدیدی بر نحوه مشاهده مدل داشته باشد. برای مثال، فرض کنید ما در حال ساخت مدلی برای پیش‌بینی قیمت فروش خانه‌ها با استفاده از پیش‌بینی‌کننده‌هایی مانند ویژگی‌های خانه (مثلاً متراژ مربع، تعداد اتاق‌خواب، تعداد حمام) و همچنین اندازه و مکان قطعه هستیم. اگر محدوده خانه‌های موجود در مجموعه آزمون، مثلاً از 60 هزار دلار تا 2 میلیون دلار، بزرگ باشد، واریانس قیمت فروش نیز بسیار زیاد خواهد بود. ممکن است مدلی با *R2 90% مشاهده* شود مثبت تلقی شود، اما RMSE ممکن است ده‌ها هزار دلار باشد - دقت پیش‌بینی ضعیف برای هر کسی که ملکی با قیمت متوسط می‌فروشد.

در برخی موارد، هدف مدل، رتبه‌بندی ساده نمونه‌های جدید است. همانطور که قبلاً بحث شد، دانشمندان داروسازی ممکن است تعداد زیادی از ترکیبات را از نظر فعالیت آنها در تلاش برای یافتن "تعداد" بررسی کنند. سپس دانشمندان ترکیباتی را که پیش‌بینی می‌شود از نظر بیولوژیکی فعال‌تر هستند، پیگیری کنند. در اینجا، تمرکز بر توانایی رتبه‌بندی مدل به جای دقت پیش‌بینی آن است. در این شرایط، تعیین *همبستگی رتبه* بین ­مقادیر مشاهده شده و پیش‌بینی شده ممکن است معیار مناسب‌تری باشد. همبستگی رتبه رتبه‌بندی مقادیر پیامد مشاهده‌شده (برخلاف ­اعداد واقعی آنها) را می‌گیرد و ارزیابی می‌کند که چقدر به رتبه‌های پیش‌بینی‌کننده‌های مدل نزدیک هستند. برای محاسبه این مقدار، رتبه‌های نتایج مشاهده شده و پیش‌بینی شده به دست آمده و ضریب همبستگی بین این رتبه‌ها محاسبه می‌شود. این معیار معمولاً به‌عنوان همبستگی رتبه اسپیرمن شناخته می‌شود.

مبادله واریانس-بایاس

MSE را می‌توان به قطعات خاص‌تری تجزیه کرد. به‌طور رسمی، MSE یک *مدل* است

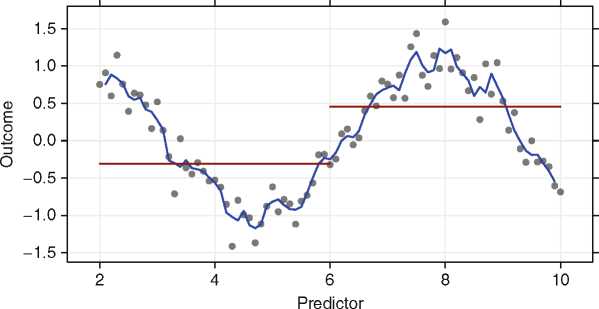
1 *n*

MSE -V *y i - y i* ) 2، *n   
i* =1

که در آن *y i* نتیجه و *y i* مدل پیش‌بینی نتیجه آن نمونه است. اگر فرض کنیم که نقاط داده از نظر آماری مستقل هستند و باقیمانده‌ها دارای میانگین نظری صفر و واریانس ثابت *2 هستند،* آنگاه

*E* [MSE] = *a 2* + (Model Bias) 2 + Model Variance *,*  (5. 1)

که در آن *E* مقدار مورد انتظار است. بخش اول ( *a* 2 ) معمولاً "صدای کاهش ناپذیر" نامیده می‌شود و با مدل‌سازی قابل حذف نیست. عبارت دوم، *بایاس مجذور* مدل است. این نشان می‌دهد که شکل عملکردی مدل چقدر می‌تواند به رابطه واقعی بین پیش‌بینی‌کننده‌ها و نتیجه نزدیک شود. آخرین عبارت واریانس مدل است. شکل [5. 2](#bookmark273) نمونه‌های افراطی از مدل‌هایی را نشان می‌دهد که یا دارای بایاس زیاد یا واریانس بالا هستند. داده‌ها یک موج *سینوسی را* شبیه‌سازی میکنند. برازش مدل نشان داده شده با رنگ قرمز داده‌ها را به نصف تقسیم می‌کند و هر نیمه را با میانگین ساده پیش‌بینی می‌کند. این مدل دارای واریانس پایینی است زیرا اگر مجموعه دیگری از نقاط داده به همین روش تولید شوند، به‌طور اساسی تغییر نمی‌کند. با این حال، در مدل‌سازی داده‌ها بی‌اثر است، زیرا به دلیل سادگی و به همین دلیل، بایاس بالایی دارد. برعکس، خط آبی یک میانگین متحرک سه نقطه است. برای مدل‌سازی موج سینوسی  *(*یعنی بایاس کم) به اندازه کافی انعطاف‌پذیر است، اما اغتشاش‌های کوچک در داده‌ها به‌طور قابل‌توجهی برازش مدل را تغییر می‌دهد. به همین دلیل واریانس بالایی دارد.



شکل 5. 2: دو مدل متناسب با موج سینوسی *.* خط *قرمز* داده‌ها را با استفاده از میانگین‌های ساده نیمه اول و دوم داده‌ها پیش‌بینی می‌کند. خط *آبی* یک میانگین متحرک سه نقطه‌ای است

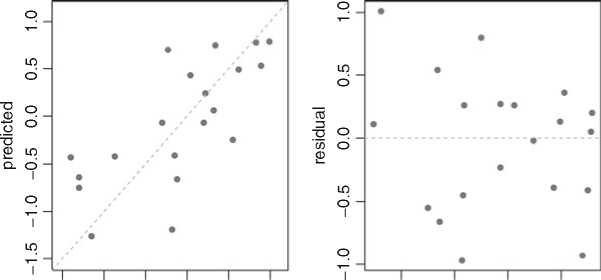
به‌طور کلی درست است که مدل‌های پیچیده‌تر می‌توانند تغییرات بسیار بالایی داشته باشند که منجر به برازش بیش از حد می‌شود. از سوی دیگر، مدل‌های ساده تمایل دارند بیش برازش نکنند، اما اگر به اندازه کافی انعطاف‌پذیر نباشند تا رابطه واقعی را مدل‌سازی کنند (بنابراین بایاس زیاد) بیش از حد مناسب نیستند. همچنین، پیش‌بینی‌کننده‌های بسیار همبسته می‌توانند منجر به مسائل *همخطی* شوند و این می‌تواند واریانس مدل را تا حد زیادی افزایش دهد. در ­فصل‌های بعدی، مدل‌هایی مورد بحث قرار خواهند گرفت که می‌توانند بایاس را در مدل افزایش دهند تا واریانس مدل را به‌عنوان راهی برای کاهش مسأله همخطی کاهش دهند. این به‌عنوان *مبادله بایاس واریانس نامیده می‌شود.*

5. 3 محاسبات

بخش‌های زیر به توابع بسته caret اشاره می‌کنند.

برای محاسبه عملکرد مدل، نتایج مشاهده شده و پیش‌بینی شده باید در بردارها ذخیره شوند. برای رگرسیون، این بردارها باید عددی باشند. در اینجا، دو بردار مثال به صورت دستی ایجاد می‌شوند تا تکنیک‌ها را نشان دهند (در عمل، بردار پیش‌بینی‌کننده‌ها توسط تابع مدل تولید می‌شود ­):

|  |
| --- |
|  |



-1. 5 -1. 0 -0. 5 0. 0 0. 5 1. 0 -1. 0 -0. 5 0. 0 0. 5

مشاهده شده پیش‌بینی شده است

شکل 5. 3: *سمت چپ* : نمودار مقادیر مشاهده شده و پیش‌بینی شده. *راست* : باقیمانده‌ها در مقابل مقادیر پیش‌بینی شده

یک گام مهم در ارزیابی کیفیت مدل، تجسم نتایج است. اول، نموداری از مقادیر مشاهده شده در برابر مقادیر پیش‌بینی شده به فرد کمک می‌کند تا بفهمد مدل چقدر برازش دارد. همچنین، نمودار باقیمانده‌ها در مقابل مقادیر پیش‌بینی شده می‌تواند به کشف الگوهای نظام‌مند در پیش‌بینی‌کننده‌های مدل، مانند روند نشان داده شده در شکل، کمک کند.  [5. 1 .](#bookmark267) از دو دستور زیر برای تولید تصاویر در شکل استفاده شده است.  [5. 3 :](#bookmark274)

*# مقادیر مشاهده شده در مقابل مقادیر پیش‌بینی شده*

*# ایده خوبی است که مقادیر را در یک مقیاس مشترک رسم کنید.*

*axisRange <- extendrange(c(مشاهده شده، پیش‌بینی شده))*

*طرح (مشاهده، پیش‌بینی،*

*+ ylim = محدوده محوری،*

*+ xlim = محدوده محور)*

*# یک خط مرجع 45 درجه اضافه کنید*

*abline(0، 1، col = "خاکستری تیره"، lty = 2)*

*# مقادیر پیش‌بینی شده در مقابل مقادیر باقیمانده*

*نمودار (پیش‌بینی شده، مقادیر باقیمانده، ylab = "باقیمانده")*

*abline(h = 0، col = "خاکستری تیره"، lty = 2)*

بسته caret شامل توابعی برای محاسبه RMSE و *R* 2 است مقدار:

*R2 (پیش‌بینی شده، مشاهده شده)*

[1] 0. 5170123

*> RMSE (پیش‌بینی شده، مشاهده شده)*

[1] 0. 5234883

فرمول‌های مختلفی برای *R* 2 وجود دارد.  [کوالست](#bookmark1020) [( 1985](#bookmark1020) ) بررسی این موارد را ارائه می‌دهد. به‌طور پیش فرض، تابع R2 از مربع ضریب همبستگی استفاده می‌کند. پایه R شامل یک تابع برای محاسبه همبستگی، از جمله همبستگی رتبه اسپیرمن است.

*# همبستگی ساده*

*کور (پیش‌بینی شده، مشاهده شده)*

[1] 0. 7190357

*# همبستگی رتبه*

*کور (پیش‌بینی شده، مشاهده شده، روش = " نیزه دار")*

[1] 0. 7554552

فصل 6

رگرسیون خطی و پسرعموهای آن

در این فصل ما چندین مدل را مورد بحث قرار خواهیم داد که همگی شبیه به رگرسیون خطی هستند که هر کدام را می‌توان به‌طور مستقیم یا غیرمستقیم به شکل نوشتاری نوشت.

*y i = b 0* + *b* 1 *X i* 1 + *b 2 X i 2* + + *b p X iP* + *e i،*  (6. 1)

جایی که *y i* نشان دهنده پاسخ عددی برای نمونه *i* است، *b* 0 نشان دهنده فاصله تخمینی، *b j* نشان دهنده ضریب تخمینی برای پیش‌بینی ­*j،* *X ij* مقدار پیش‌بینی *j را برای نمونه i م و e i* نشان دهنده خطای تصادفی است که توسط مدل قابل توضیح نیست. وقتی می‌توان یک مدل را به شکل معادله نوشت. 6. 1 می‌گوییم که *در پارامترها خطی است.* علاوه بر رگرسیون خطی معمولی، این نوع ­مدل‌ها شامل حداقل مربعات جزئی (PLS) و مدل‌های جریمه‌شده مانند رگرسیون ridge، lasso و شبکه الاستیک می‌شوند.

هر یک از این مدل‌ها به دنبال یافتن تخمینی از پارامترها هستند تا مجموع مجذور خطاها یا تابعی از مجموع مجذور خطاها به حداقل برسد. بخش [5. 2](#bookmark270) نشان می‌دهد که میانگین مربعات خطا (MSE) را می‌توان به مولفه‌های تغییرات غیر قابل کاهش، بایاس مدل و واریانس مدل تقسیم کرد. اهداف روش‌های ارائه‌شده در این فصل، ­تخمین‌های پارامتری را پیدا می‌کنند که در امتداد طیف مبادله بایاس-واریانس قرار می‌گیرند. رگرسیون خطی معمولی، در یک افراط، تخمین پارامترهایی را می‌یابد که دارای حداقل بایاس هستند، در حالی که رگرسیون خطی، ریج و خالص الاستیک تخمین‌هایی را پیدا می‌کنند که واریانس کمتری دارند. تأثیر این مبادله بر توانایی پیش‌بینی این مدل‌ها در بخش‌های بعدی نشان داده خواهد شد.

مزیت متمایز مدل‌هایی که از فرم معادله پیروی می‌کنند. 6. 1 این است که آنها بسیار قابل تفسیر هستند. به‌عنوان مثال، اگر ضریب تخمینی یک پیش‌بینی 2. 5 باشد، آنگاه افزایش 1 واحدی در مقدار آن پیش‌بینی، به‌طور متوسط، پاسخ را 2. 5 واحد افزایش می‌دهد. علاوه بر این، روابط بین پیش‌بینی‌کننده‌ها را می‌توان از طریق ضرایب برآورد شده بیشتر تفسیر کرد.

مزیت دیگر این نوع مدل‌ها این است که ماهیت ریاضی آنها ما را قادر می‌سازد تا خطاهای استاندارد ضرایب را محاسبه کنیم، مشروط بر اینکه مفروضات خاصی در مورد توزیع باقیمانده‌های مدل داشته باشیم.

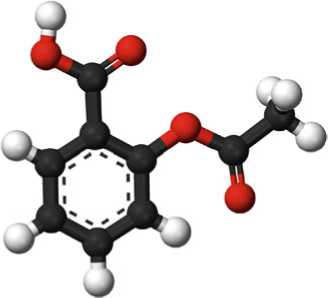
سپس می‌توان از این خطاهای استاندارد برای ارزیابی اهمیت آماری هر پیش‌بینی کننده در مدل استفاده کرد. این دیدگاه استنباطی می‌تواند درجه بیشتری از درک مدل را فراهم کند، تا زمانی که مفروضات توزیعی به اندازه کافی برآورده شوند. از آنجایی که این کار بر پیش‌بینی مدل متمرکز است، زمان زیادی را صرف ماهیت استنتاجی این مدل‌ها نمی‌کنیم.

در حالی که مدل‌های نوع رگرسیون خطی بسیار قابل تفسیر هستند، اما می‌توانند از نظر سودمندی محدود شوند. اول، این مدل‌ها زمانی مناسب هستند که رابطه ­بین پیش‌بینی‌کننده‌ها و پاسخ در امتداد یک ابرصفحه قرار گیرد. به‌عنوان مثال، اگر داده‌ها فقط یک پیش‌بینی داشته باشند، اگر رابطه بین پیش‌بینی و پاسخ در یک خط مستقیم قرار گیرد، تکنیک‌ها مناسب خواهند بود. با پیش‌بینی‌کننده‌های بیشتر، رابطه باید به یک ابر صفحه مسطح نزدیک شود. اگر یک رابطه منحنی بین پیش‌بینی‌کننده‌ها و پاسخ وجود داشته باشد (مثلاً، مانند درجه دوم، مکعب، یا برهمکنش بین پیش‌بینی ها)، مدل‌های رگرسیون خطی را می‌توان با پیش‌بینی‌کننده‌های اضافی که توابعی از پیش‌بینی‌کننده‌های اصلی هستند، در تلاش برای به تصویر ­کشیدن اضافه کرد. این روابط بحث بیشتر در مورد استراتژی‌های تقویت پیش‌بینی‌کننده‌های اصلی در بخش‌های زیر دنبال خواهد شد. با این حال، روابط غیر خطی بین پیش‌بینی‌کننده‌ها و پاسخ ممکن است به اندازه کافی با این مدل‌ها گرفته نشود. اگر این مورد برای داده‌ها باشد، پس روش‌هایی ­که در فصل‌ها توضیح داده شده است.  [7](#bookmark342) و [8](#bookmark388) رابطه پیش‌بینی بین پیش‌بینی‌کننده‌ها و پاسخ را بهتر آشکار می‌کند.

**مطالعه موردی: مدلسازی کمی رابطه ساختار-فعالیت**

مواد شیمیایی از جمله داروها را می‌توان با فرمول‌های شیمیایی نشان داد. به‌عنوان مثال، شکل.  [6. 1](#bookmark282) ساختار آسپرین را نشان می‌دهد که حاوی 9 اتم کربن، هشت هیدروژن و چهار اتم اکسیژن است. از این پیکربندی، ­اندازه‌گیری‌های کمی مانند وزن مولکولی، بار الکتریکی یا مساحت سطح را می‌توان به دست آورد. این مقادیر به‌عنوان *توصیفگرهای شیمیایی شناخته می‌*شوند و انواع بی شماری از توصیف کننده‌ها وجود دارد که می‌توانند از یک معادله شیمیایی مشتق شوند. برخی از آن‌ها مانند تعداد اتم‌های کربن ساده هستند، در حالی که برخی دیگر را می‌توان محرمانه توصیف کرد (به‌عنوان مثال، مجموع ضریب آخرین بردار ویژه از ماتریس Barysz وزن‌شده با حجم واندروالس).

برخی از ویژگی‌های مولکول‌ها را نمی‌توان به صورت تحلیلی از روی ساختار شیمیایی تعیین کرد. به‌عنوان مثال، یکی از راه‌هایی که ممکن است یک ترکیب ­ارزش متخصصی داشته باشد، این است که بتواند تولید یک پروتئین خاص را مهار کند. این معمولاً فعالیت بیولوژیکی یک ترکیب نامیده می‌شود. رابطه بین ساختار شیمیایی و فعالیت آن می‌تواند پیچیده باشد. به این ترتیب، رابطه معمولاً به صورت تجربی با استفاده از آزمایش‌ها تعیین می‌شود. یک راه برای انجام این کار، ایجاد یک سنجش بیولوژیکی برای هدف مورد نظر (یعنی پروتئین) است. سپس مجموعه‌ای از ترکیبات را می‌توان در آزمون قرار داد و فعالیت یا مهار آنها را اندازه‌گیری کرد. این اطلاعات فعالیت داده‌هایی را تولید می‌کند که می‌تواند به‌عنوان استفاده شود



شکل 6. 1: نمایشی از آسپرین که حاوی اتم‌های کربن (به صورت *توپ‌های سیاه* ) و هیدروژن ( *سفید* ) و اتم‌های اکسیژن ( *قرمز* ) است. فرمول شیمیایی این مولکول O=C(Oc1ccccc1C(=O)O)C است که از آن می‌توان توصیف‌های مولکولی مانند وزن مولکولی 180. 2 گرم بر مول را تعیین کرد.

مجموعه آموزشی برای مدل‌سازی پیش‌بینی به‌طوری که ترکیباتی که ممکن است هنوز وجود نداشته باشند، بتوانند از نظر فعالیت غربالگری شوند. این فرآیند به‌عنوان ­مدل‌سازی رابطه ساختار-فعالیت کمی (QSAR) نامیده می‌شود.  [لیچ و ژیلت](#bookmark1020) [( 2003](#bookmark1020) ) مقدمه‌ای در سطح بالا برای مدل‌سازی QSAR و توصیفگرهای مولکولی ارائه می‌کند.

در حالی که فعالیت مهم است، سایر ویژگی‌ها باید ارزیابی شوند تا مشخص شود آیا یک ترکیب "شبیه دارو" است [( Lipinski et al. 1997](#bookmark1021) ). کیفیت‌های فیزیکی، مانند حلالیت یا چربی دوستی (یعنی "چربی بودن") و همچنین خواص دیگر، مانند سمیت، ارزیابی می‌شوند. حلالیت یک ترکیب اگر به صورت خوراکی یا تزریقی تجویز شود بسیار مهم است. ما ­تکنیک‌های مدل‌سازی رگرسیون مختلفی را با پیش‌بینی حلالیت با استفاده از ساختارهای شیمیایی نشان خواهیم داد.

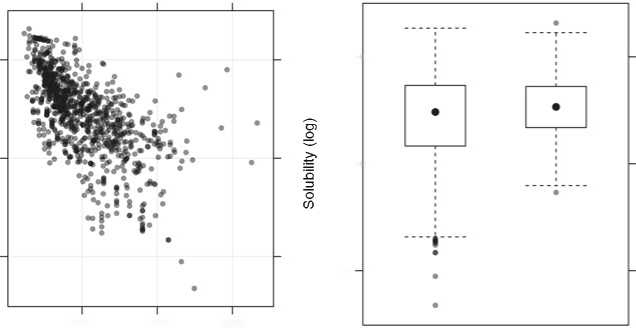
تتکو و همکاران [( 2001](#bookmark1026) ) و [هووسکونن](#bookmark1019) [( 2000](#bookmark1019) ) مجموعه‌ای از ترکیبات را با مقادیر حلالیت تجربی متناظر با استفاده از مجموعه‌های پیچیده‌ای از پیش بینی کننده ها بررسی کردند. آنها از مدل‌های رگرسیون خطی و شبکه عصبی برای تخمین رابطه بین ساختار شیمیایی و حلالیت استفاده کردند. برای تحلیل خود، از 1267 ترکیب و مجموعه‌ای از توصیفگرهای قابل درک تر استفاده خواهیم کرد که در یکی از سه گروه قرار می‌گیرند:

دویست و هشت "اثر انگشت" دوتایی که وجود یا عدم وجود یک زیرساخت شیمیایی خاص را نشان می‌دهد.

تعداد شانزده توصیف کننده، مانند تعداد پیوندها یا تعداد اتم‌های برم.

چهار توصیف کننده پیوسته، مانند وزن مولکولی یا مساحت سطح.

به‌طور متوسط، توصیفگرها همبستگی ندارند. با این حال، جفت‌های زیادی وجود دارد که همبستگی مثبت قوی نشان می‌دهند. 47 جفت همبستگی بیشتر از

شکل 6. 2: رابطه بین حلالیت و دو توصیفگر. *سمت چپ* : با افزایش وزن مولکولی، حلالیت به‌طور کلی کاهش می‌یابد. این رابطه تقریباً لگ خطی است، به جز چندین ترکیب با حلالیت کم و وزن زیاد و حلالیت بین 0 تا *-* 5. *درست* : برای توصیفگر اثر انگشت خاص، زمانی که زیرساخت مورد نظر در مولکول وجود نداشته باشد، حلالیت کمی بالاتر است.

-5

-5 -

10

-10

200

400

600

Molecular Weight

structure absent structure present

0. 90. در برخی موارد، باید انتظار همبستگی بین توصیفگرها را داشته باشیم. برای مثال، در داده‌های حلالیت، سطح یک ترکیب برای مناطق مرتبط با اتم‌های خاص (مثلاً نیتروژن یا اکسیژن) محاسبه می‌شود. یک ­توصیفگر در این داده‌ها مساحت سطح مرتبط با دو عنصر خاص را اندازه‌گیری می‌کند در حالی که دیگری از همان عناصر به اضافه دو عنصر دیگر استفاده می‌کند. با توجه به تعاریف آنها، ما انتظار داریم که دو پیش‌بینی سطح سطح همبستگی داشته باشند. در واقع، توصیفگرها برای 87 درصد از ترکیبات یکسان هستند. تفاوت‌های کوچک بین پیش‌بینی‌کننده‌های سطح سطح ممکن است حاوی ­اطلاعات مهمی برای پیش‌بینی باشد، اما مدل‌ساز باید بداند که پیامدهای افزونگی در مدل وجود دارد. یکی دیگر از ویژگی‌های مرتبط پیش‌بینی‌کننده‌های حلالیت این است که توصیف‌گرهای مبتنی بر شمارش چولگی راست قابل‌توجهی را نشان می‌دهند که ممکن است بر برخی از مدل‌ها تأثیر بگذارد (به فصل 13 مراجعه کنید).  [3](#bookmark131) برای بحث در مورد این مسائل).

داده‌های نتیجه در لاگ 10 اندازه‌گیری شد مقیاس و محدوده از *-* 11. 6 تا 1. 6 با میانگین حلالیت ورود به سامانه 2 *-.* 7. شکل [6. 2](#bookmark283) رابطه بین مقادیر حلالیت مشتق شده تجربی و دو نوع توصیفگر در داده‌های مثال را نشان می‌دهد.

داده‌ها با استفاده از نمونه‌گیری تصادفی به یک مجموعه آموزشی ( *n* = 951) و مجموعه آزمون ( *n* = 316) تقسیم شدند. مجموعه آموزشی برای تنظیم و تخمین مدل‌ها و همچنین برای تعیین تخمین‌های اولیه عملکرد با استفاده از اعتبارسنجی متقاطع 10 برابری استفاده می‌شود. مجموعه آزمایشی برای توصیف نهایی مدل‌های مورد علاقه استفاده خواهد شد.

کاوش مجموعه آموزشی برای درک ویژگی‌های داده‌ها قبل از مدل‌سازی مفید است. به یاد بیاورید که 208 مورد از پیش‌بینی‌کننده‌ها اثر انگشت باینری هستند. از آنجایی که فقط دو مقدار از این متغیرها وجود دارد، پیش پردازش بسیار کمی وجود دارد که انجام شود.

با حرکت به جلو، می‌توانیم پیش‌بینی‌کننده‌های پیوسته را برای چولگی ارزیابی کنیم. میانگین آمار چولگی 1. 6 (با حداقل 0. 7 و حداکثر 3. 8) بود که نشان می‌دهد این پیش‌بینی‌کننده‌ها تمایل به انحراف درست دارند. برای تصحیح این چولگی، یک تبدیل Box-Cox برای همه پیش‌بینی‌کننده‌ها اعمال شد (یعنی، پارامتر تبدیل برای هیچ یک از پیش‌بینی‌کننده‌های پیوسته نزدیک به یک تخمین زده نشد).

با استفاده از این پیش‌بینی‌کننده‌های تبدیل شده، آیا می‌توان فرض کرد که رابطه ­بین پیش‌بینی‌کننده‌ها و نتیجه خطی است؟ شکل 6. 3 نمودارهای پراکنده پیش‌بینی‌کننده‌ها را در برابر نتیجه همراه با یک خط رگرسیون از یک مدل "صاف تر" انعطاف‌پذیر به نام لس نشان می‌دهد [( کلولند 1979](#bookmark1013) ). خطوط رگرسیون هموار نشان می‌دهد که برخی روابط خطی بین پیش‌بینی‌کننده‌ها و نتیجه (مثلاً وزن مولکولی) و برخی روابط غیرخطی (مثلاً تعداد منشاءها یا کلرها) وجود دارد. به همین دلیل، ممکن است مجموعه پیش‌بینی را با عبارات درجه دوم برای برخی متغیرها افزایش دهیم.

آیا همبستگی بین پیش‌بینی معناداری وجود دارد؟ برای پاسخ به این سوال، از تحلیل مؤلفه اصلی (PCA) روی مجموعه کامل ­پیش‌بینی‌کننده‌های تبدیل شده استفاده شد و درصد واریانس محاسبه‌شده توسط هر ­مؤلفه تعیین می‌شود. شکل [6. 4](#bookmark287) معمولاً به‌عنوان نمودار scree شناخته می‌شود و نمایه‌ای از تغییرات محاسبه شده توسط هر جزء را نشان می‌دهد. توجه داشته باشید که مقدار تغییرات خلاصه شده توسط مؤلفه به شدت کاهش می‌یابد، به‌طوری که هیچ جزء بیش از 13 درصد از واریانس را تشکیل نمی‌دهد. این نمایه ­نشان می‌دهد که ساختار داده‌ها در تعداد ابعاد بسیار کمتری نسبت به تعداد ابعاد فضای اصلی قرار دارد. این اغلب به دلیل تعداد زیادی از هم خطی‌ها در بین پیش‌بینی‌کننده‌ها است. شکل [6. 5](#bookmark287) ساختار همبستگی پیش‌بینی‌کننده‌های پیوسته تبدیل شده را نشان می‌دهد. بسیاری از همبستگی‌های مثبت قوی وجود دارد (که با دایره‌های بزرگ و آبی تیره مشخص می‌شود). همانطور که قبلاً بحث شد، این می‌تواند مسائلی را در توسعه برخی ­مدل‌ها (مانند رگرسیون خطی) ایجاد کند و مراحل پیش پردازش مناسب برای در نظر گرفتن این مسأله باید انجام شود.

رگرسیون خطی

هدف از رگرسیون خطی حداقل مربعات معمولی یافتن صفحه‌ای است که مجموع مربعات خطاها (SSE) بین پاسخ مشاهده شده و پیش‌بینی شده را به حداقل می‌رساند:

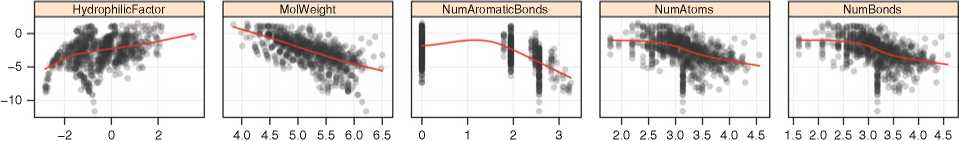
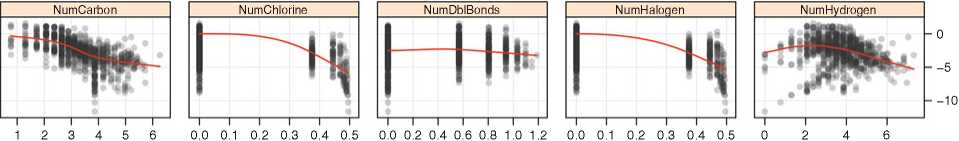
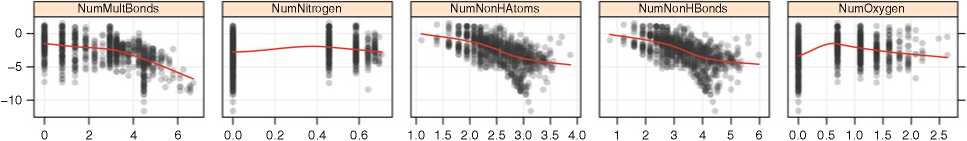
*n*

sse=£( *y i - y i* ) 2 *,*

*i* = 1

شکل 6. 3: نمودارهای پراکندگی پیش‌بینی‌کننده‌های پیوسته تبدیل شده در مجموعه داده حلالیت. خط *قرمز* یک طرح پراکندگی هموارتر است

106 6 رگرسیون خطی و پسرعموهای آن



اجزاء

شکل 6. 4: نمودار اسکری از تحلیل PCA از پیش‌بینی‌کننده‌های حلالیت

w   
"U

NumDblBonds

NumOxygen

سطح 1

سطح 2

NumRotBonds

NumHydrogen

NumAtoms

NumBonds

NumSulfer

NumNitrogen

فاکتور هیدروفیل

NumClorine

NumHalogen NumMultBonds

NumAromaticBonds

::□□□□□□□□□□□□

□□□□□من   
□□□□□

□□□□□□: من د

□■□□□■□a:   
□□□□□□□□□

□□□□□□

□□□□□□

مولی وزن

NumRings •

NumCarbon

NumNonHAtoms NumNonHBonds

□□□□□

□□□□□□□□

■mm]

□□□□□□□□

□□□□□□□□□□□□□□□: m t i

□□□□□□□□DO

□□□□□□mm

mnnhn:

1

0. 8

0. 6

0. 4

0. 2

0

-0. 2

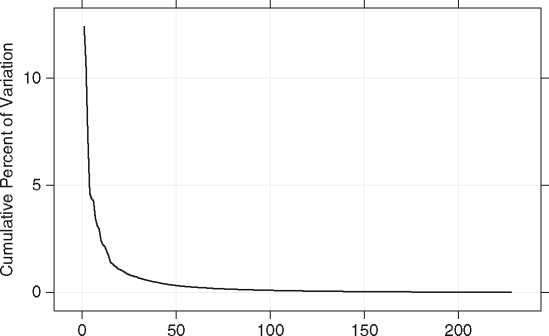
-0. 4

-0. 6

-0. 8

-1

پذیری پیوسته تبدیل شده­



که در آن *y i* نتیجه و *y i* مدل پیش‌بینی نتیجه آن نمونه است. از نظر ریاضی می‌توان صفحه بهینه را نشان داد

( X *T* X ) *“* 1 X *T y,*  (6. 2)

که در آن X ماتریس پیش‌بینی‌کننده‌ها و *y* بردار پاسخ است. معادله [6. 2](#bookmark288) در متون آماری به‌عنوان *P (*"بتا کلاه") نیز شناخته می‌شود و برداری است که ­شامل تخمین پارامترها یا ضرایب برای هر پیش‌بینی است. محاسبه این ­کمیت [( 6. 2 )](#bookmark288) آسان است و ضرایب مستقیماً قابل تفسیر هستند. با ایجاد برخی فرضیات حداقلی در مورد توزیع باقیمانده‌ها، نشان دادن آن ساده است که تخمین‌های پارامتری که SSE را به حداقل می‌رساند، آنهایی هستند که کمترین بایاس را در بین تمام تخمین‌های پارامتر ممکن دارند [( Graybill 1976](#bookmark1017) ). از این رو، این تخمین‌ها مولفه بایاس مبادله بایاس-واریانس را به حداقل می‌رساند.

تفسیرپذیری ضرایب آن را به‌عنوان یک ابزار مدل‌سازی بسیار جذاب می‌کند. در عین حال، ویژگی‌هایی که آن را قابل تفسیر می‌کند، آن را مستعد نقص‌های بالقوه کشنده نیز می‌کند. توجه کنید که در معادله تعبیه شده است. [( 6. 2 )](#bookmark288) عبارت X *T* X *-* 1 است که با ماتریس کوواریانس پیش‌بینی‌کننده‌ها متناسب است. معکوس منحصربه‌فرد این ماتریس زمانی وجود دارد که (1) هیچ پیش‌بینی‌ای را نمی‌توان از ترکیب یک یا چند پیش‌بینی دیگر تعیین کرد و (2) تعداد نمونه‌ها از تعداد پیش‌بینی‌کننده‌ها بیشتر باشد. اگر داده‌ها تحت هر یک از این شرایط قرار گیرند، مجموعه منحصر به فردی از همکارا رگرسیون ­وجود ندارد. با این حال، یک مجموعه منحصر به فرد از مقادیر پیش‌بینی‌شده هنوز هم می‌تواند برای داده‌هایی که تحت شرط (1) قرار می‌گیرند، با جایگزین کردن ( XTX *)* 1 با یک معکوس شرطی [(](#bookmark1017) *Graybill* [1976](#bookmark1017) *)* یا با حذف پیش‌بینی‌کننده‌هایی که هم خط هستند، به دست آورد. به‌طور پیش‌فرض، زمانی که برازش یک مدل خطی با R و همخطی در میان پیش‌بینی‌کننده‌ها وجود دارد، «. *. .* R با حذف متغیرها به ترتیب معکوس شکل ظاهری در فرمول مدل، با بزرگترین مدل قابل شناسایی برازش می‌کند» [( Faraway )](#bookmark1016) [2005](#bookmark1016) ). نتیجه این حقایق این است که رگرسیون خطی هنوز هم می‌تواند برای پیش‌بینی استفاده شود زمانی که همخطی در داده‌ها وجود دارد. اما از آنجایی که ضرایب رگرسیون برای تعیین این پیش‌بینی‌کننده‌ها منحصر به فرد نیستند، ما توانایی خود را برای تفسیر معنی دار ضرایب از دست می‌دهیم.

هنگامی که شرط (2) برای یک مجموعه داده صادق است، متخصص می‌تواند چندین مرحله را برای ایجاد یک مدل رگرسیون انجام دهد. به‌عنوان اولین قدم پیشنهاد می‌کنیم از تکنیک‌های پیش پردازش ارائه شده در بخش استفاده کنید.  [3. 3](#bookmark147) حذف پیش‌بینی‌کننده‌های همبسته زوجی که تعداد پیش‌بینی‌کننده‌های کلی را کاهش می‌دهد. با این حال، این مرحله پیش پردازش ممکن است به‌طور کامل همخطی را حذف نکند، زیرا یک یا چند پیش‌بینی ممکن است توابع *دو* یا چند پیش‌بینی دیگر باشند. برای تشخیص چند خطی بودن در زمینه رگرسیون خطی، می‌توان از *عامل تورم واریانس استفاده کرد* [( Myers 1994](#bookmark1022) ). این آمار برای هر پیش‌بینی و تابعی از همبستگی بین پیش‌بین انتخاب‌شده و همه پیش‌بینی‌کننده‌های دیگر محاسبه می‌شود.

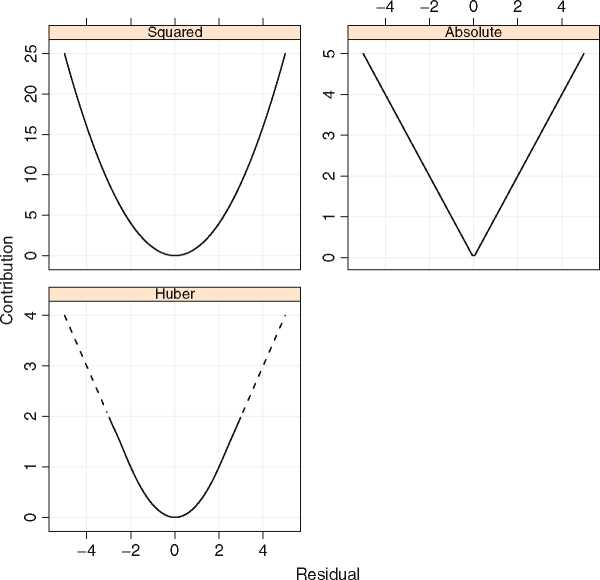
پس از پیش پردازش داده‌ها، اگر تعداد پیش‌بینی‌کننده‌ها همچنان از تعداد مشاهدات بیشتر باشد، باید اقدامات دیگری را برای ­کاهش ابعاد فضای پیش‌بینی انجام دهیم. پیش پردازش PCA (بخش [3. 3 )](#bookmark147) یکی از راه حل‌های ممکن است راه‌حل‌های دیگر عبارتند از کاهش ابعاد و رگرسیون همزمان از طریق PLS یا استفاده از روش‌هایی که ­تخمین‌های پارامتر را کوچک می‌کنند، مانند رگرسیون برآمدگی، ریج، یا شبکه الاستیک.

یکی دیگر از اشکالات رگرسیون خطی چندگانه خطی بودن راه حل آن در پارامترها است. این بدان معنی است که راه حلی که به دست می‌آوریم یک ابر صفحه تخت است. واضح است که اگر داده‌ها دارای ساختار انحنا یا غیرخطی باشند، رگرسیون قادر به شناسایی این ویژگی‌ها نخواهد بود. یکی از سرنخ‌های بصری برای درک ­اینکه آیا رابطه بین پیش‌بینی‌کننده‌ها و پاسخ خطی نیست، بررسی نمودارهای تشخیصی اولیه نشان‌داده‌شده در شکل 1 است.  [5. 3](#bookmark274) . انحنا در نمودار پیش‌بینی‌شده در مقابل باقیمانده یک شاخص اولیه است که رابطه زیربنایی ­خطی نیست. برهمکنش‌های درجه دوم، مکعبی یا بین پیش‌بینی‌کننده‌ها را می‌توان با افزودن برهمکنش‌های درجه دوم، مکعب و برهمکنش‌های پیش‌بینی‌کننده‌های اصلی در رگرسیون تطبیق داد. اما هر چه تعداد پیش‌بینی‌کننده‌های اصلی بیشتر باشد، گنجاندن برخی یا همه این اصطلاحات کاربردی‌تر می‌شود. استفاده از این ­رویکرد برنامه می‌تواند باعث شود که ماتریس داده‌ها پیش‌بینی‌کننده‌های بیشتری نسبت به مشاهدات داشته باشد و ما دوباره نمی‌توانیم ماتریس را معکوس کنیم.

اگر روابط غیرخطی به راحتی قابل شناسایی بین پیش‌بینی‌کننده‌ها و پاسخ وجود داشته باشد، آن پیش‌بینی‌کننده‌های اضافی را می‌توان به ­ماتریس توصیف کننده اضافه کرد. با این حال، اگر امکان شناسایی این روابط وجود نداشته باشد یا روابط بین پیش‌بینی‌کننده‌ها و پاسخ بسیار ­غیرخطی باشد، روش‌های پیچیده‌تری مانند آنچه در فصل بحث شد.  [7](#bookmark342) این ساختار را به‌طور موثرتر و کارآمدتر پیدا می‌کند.

سومین مسأله قابل‌توجه رگرسیون خطی چندگانه این است که مستعد تعقیب مشاهداتی است که از روند کلی اکثریت داده‌ها دور هستند. به یاد بیاورید که رگرسیون خطی به دنبال یافتن تخمین پارامترهایی ­است که SSE را به حداقل می‌رساند. از این رو، مشاهداتی که از روند اکثر داده‌ها دور هستند، باقیمانده‌های بزرگی خواهند داشت. به منظور به حداقل رساندن SSE، رگرسیون خطی تخمین پارامترها را برای تطبیق بهتر این مشاهدات غیر معمول تنظیم می‌کند. مشاهداتی که باعث تغییرات قابل‌توجهی در تخمین پارامترها می‌شوند، *تأثیرگذار نامیده می‌شوند* و زمینه ­رگرسیون ro bust برای رسیدگی به این نوع مسائل ایجاد شده است. یکی از رویکردهای رایج استفاده از یک معیار جایگزین برای SSE است که حساسیت کمتری نسبت به مقادیر پرت بزرگ دارد. برای مثال، یافتن تخمین‌های پارامتری که مجموع خطاهای مطلق را به حداقل می‌رسانند، در برابر نقاط پرت مقاوم‌تر است، همانطور که در شکل 1 مشاهده می‌شود.  [6. 6 .](#bookmark290) همچنین، تابع Huber از مجذور باقیمانده‌ها زمانی که "کوچک" هستند و تفاوت ساده بین مقادیر مشاهده شده و پیش‌بینی شده زمانی که باقیمانده‌ها بالاتر از یک آستانه هستند استفاده می‌کند. این رویکرد می‌تواند تأثیر مشاهداتی را که از روند کلی داده‌ها دور می‌شوند، به حداقل برساند.

هیچ پارامتر تنظیمی برای رگرسیون خطی چندگانه وجود ندارد. با این حال، این واقعیت، متخصص را از استفاده از ابزارهای دقیق اعتبارسنجی مدل، به‌ویژه زمانی که از این مدل برای پیش‌بینی استفاده می‌کند، منع نمی‌کند. در واقع، ما باید از همان تکنیک‌های آموزشی و اعتبار سنجی که در فصل توضیح داده شده است استفاده کنیم.  [4](#bookmark192) برای ­درک توانایی پیش‌بینی این مدل بر روی داده‌هایی که مدل ندیده است.



شکل 6. 6: رابطه بین باقیمانده مدل و سهم آن در تابع هدف برای چندین تکنیک. برای رویکرد هوبر، از آستانه 2 استفاده شد

هنگام استفاده از تکنیک‌های نمونه‌گیری مجدد مانند بوت استرپینگ یا اعتبارسنجی متقابل، متخصص همچنان باید از مسائلی که در بالا توضیح داده شد آگاه باشد. به‌عنوان مثال، مجموعه داده‌ای را در نظر بگیرید که در آن 100 نمونه و 75 پیش‌بینی کننده وجود دارد. اگر از طرح نمونه‌گیری مجدد استفاده کنیم که از دو سوم داده‌ها برای آموزش استفاده می‌کند، نمی‌توانیم مجموعه‌ای از ضرایب رگرسیون منحصربه‌فرد را پیدا کنیم، زیرا تعداد پیش‌بینی‌کننده‌ها در مجموعه آموزشی از تعداد نمونه‌ها بیشتر خواهد بود. بنابراین برای رگرسیون خطی چندگانه، متخصص باید نه تنها هنگام کار با مجموعه داده‌های اصلی، بلکه هنگام کار با زیر مجموعه‌های داده ایجاد شده در طول آموزش و ارزیابی مدل، از مسائل آن آگاه باشد.

برای نشان دادن مسأله پیش‌بینی‌کننده‌های همبسته، مدل‌های خطی با ترکیبی از توصیفگرهای مربوط به تعداد اتم‌های غیرهیدروژن و تعداد پیوندهای هیدروژنی مطابقت داشتند. در مجموعه آموزشی، این پیش‌بینی‌کننده‌ها همبستگی بالایی دارند (همبستگی: 0. 994). شکل 6. 3 رابطه آنها را با نتیجه نشان می‌دهد که تقریباً یکسان است. ابتدا، دو ­مدل رگرسیون جداگانه را با شرایط فردی و سپس مدل سوم را با هر دو برازش می‌دهیم

جدول 6. 1: ضرایب رگرسیون برای دو پیش‌بینی بسیار همبسته در چهار مدل جداگانه

|  |  |  |
| --- | --- | --- |
| مدل | NumNonHAtoms | NumNonHBonds |
| فقط NumNonHAtoms  فقط NumNonHBonds | *-* 1. 2 (0. 1) | *-* 1. 2 (0. 1) |
| هر دو | *-* 0 *\_* 3 (0. 5) | *-* 0 *\_* 9 (0. 5) |
| همه پیش‌بینی ها | 8. 2 (1. 4) | *-* 9. 1 (1. 6) |

مقررات. پیش‌بینی‌کننده‌ها قبل از مدل‌سازی متمرکز و مقیاس‌بندی شدند تا واحدهای آنها یکسان باشد. جدول [6. 1](#bookmark293) ضرایب رگرسیون و خطاهای استاندارد آنها را در پرانتز نشان می‌دهد. برای هر مدل، ضرایب رگرسیون تقریباً یکسان است و خطاهای استاندارد آنها نیز مشابه است. با این حال، هنگام برازش یک مدل با هر دو عبارت، نتایج متفاوت است. شیب مربوط به تعداد اتم‌های غیر هیدروژن به شدت کاهش می‌یابد. همچنین خطاهای استاندارد در مقایسه با مدل‌های فردی پنج برابر افزایش می‌یابد. این نشان دهنده بی ثباتی در خطی رگرسیون ناشی از روابط بین پیش‌بینی ­است و این بی ثباتی مستقیماً به پیش‌بینی‌کننده‌های مدل منتشر می‌شود. جدول [6. 1](#bookmark293) همچنین ضرایب این دو توصیفگر را هنگامی که همه پیش‌بینی‌کننده‌ها در مدل قرار می‌گیرند نشان می‌دهد. یادآوری از شکل [6. 5](#bookmark287) که پیش‌بینی‌کننده‌های خطی زیادی در داده‌ها وجود دارد و ما انتظار داریم که اثر همخطی تشدید شود. در واقع، برای این دو پیش‌بینی، مقادیر به شدت بزرگ می‌شوند و خطاهای استاندارد آنها 14 تا 16 برابر بزرگ‌تر از خطاهای مدل‌های جداگانه است.

در عمل، چنین پیش‌بینی‌کننده‌هایی با همبستگی بالا ممکن است به صورت دستی با حذف یکی از پیش‌بینی‌کننده‌های متخلف مدیریت شوند. با این حال، اگر تعداد پیش‌بینی‌کننده‌ها زیاد باشد، این ممکن است دشوار باشد. همچنین، در بسیاری از موارد، روابط بین پیش‌بینی‌کننده‌ها می‌تواند پیچیده باشد و پیش‌بینی‌کننده‌های زیادی را شامل شود. در این موارد، حذف دستی پیش‌بینی‌کننده‌های خاص ممکن است امکان‌پذیر نباشد و مدل‌هایی که می‌توانند همخطی بودن را تحمل کنند ممکن است مفیدتر باشند.

رگرسیون خطی برای داده‌های حلالیت

به یاد بیاورید که در بخش.  [6. 1](#bookmark281) داده‌های حلالیت را به مجموعه‌های آموزشی و آزمایشی تقسیم کردیم و یک تبدیل Box-Cox را برای پیش‌بینی‌کننده‌های پیوسته اعمال کردیم تا چولگی را حذف کنیم. گام بعدی در فرآیند ساخت مدل برای رگرسیون خطی، شناسایی پیش‌بینی‌کننده‌هایی است که همبستگی‌های زوجی بالایی دارند و حذف پیش‌بینی‌کننده‌ها به‌طوری‌که هیچ همبستگی زوجی مطلقی بیشتر از سطح از پیش تعیین‌شده نباشد. در این مورد، ما تصمیم گرفتیم پیش‌بینی‌کننده‌هایی را حذف کنیم که همبستگی‌های زوجی بیشتر از 0. 9 دارند (به بخش مراجعه کنید.  [3. 3 )](#bookmark147) . در این سطح، 38 پیش‌بینی شناسایی و حذف شدند. با حذف این پیش‌بینی‌ها، الف

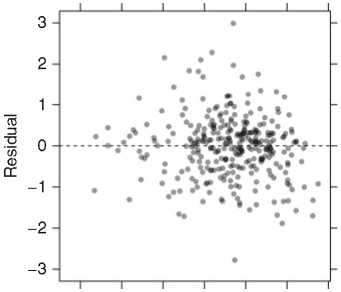


Fig. 6.7: *Left*: Observed versus predicted values for the solubility test set. *Right* : Residuals versus the predicted values. The residuals appear to be ran­domly scattered about 0 with respect to the predicted values

0

Predicted

-6

-2

-8

-4

-10

مدل خطی با داده‌های آموزشی برازش داشت. [[13]](#footnote-13) مدل خطی با استفاده از اعتبارسنجی متقاطع 10 برابری نمونه‌برداری شد و ریشه میانگین مربعات خطا (RMSE) 0. 71 با *R متناظر بود.* [[14]](#footnote-14) مقدار 0. 88.

پیش‌بینی‌کننده‌هایی که از داده‌های آموزشی حذف شده بودند، سپس ­از داده‌های آزمون نیز منتقل شدند و سپس مدل به مجموعه آزمون اعمال شد. *R* 2 \_ مقدار بین مقادیر مشاهده شده و پیش‌بینی شده 0. 87 بود و نمودارهای تشخیصی رگرسیون پایه در شکل 1 نشان داده شده است.  [6. 7](#bookmark8) . به نظر می‌رسد هیچ گونه بایاس در پیش‌بینی وجود نداشته باشد و توزیع بین مقادیر پیش‌بینی شده و باقیمانده‌ها تصادفی در حدود صفر به نظر می‌رسد.

**حداقل مربعات جزئی**

برای بسیاری از مجموعه‌های داده‌های واقعی، پیش‌بینی‌کننده‌ها می‌توانند همبستگی داشته باشند و حاوی اطلاعات پیش‌بینی مشابهی باشند که با داده‌های حلالیت نشان داده شده است. اگر ­همبستگی بین پیش‌بینی‌کننده‌ها زیاد باشد، راه‌حل حداقل مربعات معمولی برای رگرسیون خطی چندگانه دارای تغییرات بالایی بوده و ناپایدار می‌شود.

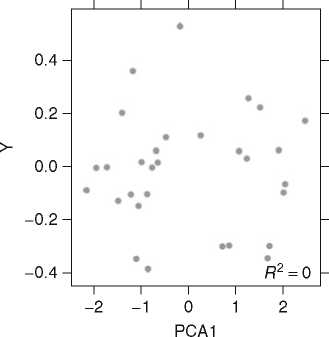
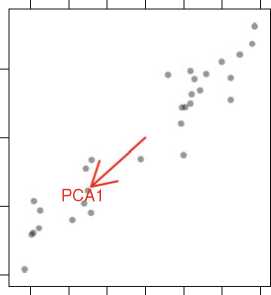
برای سایر مجموعه‌های داده، تعداد پیش‌بینی‌کننده‌ها ممکن است بیشتر از تعداد مشاهدات باشد. در این مورد نیز، حداقل مربعات معمولی در شکل معمول خود قادر به یافتن مجموعه‌ای منحصر به فرد از ضرایب رگرسیونی که SSE را به حداقل می‌رساند، نخواهد بود.

چند راه‌حل متداول برای مسأله رگرسیون تحت این ­شرایط شامل پیش‌پردازش پیش‌بینی‌کننده‌ها با حذف پیش‌بینی‌کننده‌های بسیار همبسته با استفاده از تکنیک‌هایی است که در بخش توضیح داده شد.  [3. 3](#bookmark147) یا (2) انجام PCA بر روی پیش‌بینی‌کننده‌ها همانطور که در بخش توضیح داده شده است.  [3. 3](#bookmark147) . حذف پیش‌بینی‌کننده‌های بسیار همبسته تضمین می‌کند که همبستگی‌های زوجی بین پیش‌بینی‌کننده‌ها زیر یک آستانه از پیش تعیین‌شده است. با این حال، این فرآیند لزوماً تضمین نمی‌کند که ترکیب‌های خطی پیش‌بینی‌کننده‌ها با دیگر پیش‌بینی‌کننده‌ها همبستگی ندارند. اگر اینطور باشد، راه حل حداقل مربعات معمولی همچنان ناپایدار خواهد بود. بنابراین درک این نکته مهم است که حذف پیش‌بینی‌کننده‌های زوجی با همبستگی بالا ممکن است راه‌حل حداقل مربعات پایدار را تضمین نکند. روش دیگر، استفاده از PCA برای پیش پردازش تضمین می‌کند که ­پیش‌بینی‌کننده‌های حاصل، یا ترکیبی از آن‌ها، همبستگی ندارند. مبادله در استفاده از PCA این است که پیش‌بینی‌کننده‌های جدید ترکیب‌های خطی ­پیش‌بینی‌کننده‌های اصلی هستند و بنابراین، درک عملی پیش‌بینی‌کننده‌های جدید می‌تواند مبهم شود.

پیش‌پردازنده‌های پیش‌پردازش از طریق PCA قبل از انجام رگرسیون به‌عنوان رگرسیون مؤلفه اصلی (PCR) شناخته می‌شود [( Masy 1965](#bookmark1022) ). این تکنیک به‌طور گسترده در زمینه مسائل با ­پیش‌بینی‌کننده‌های ذاتاً مرتبط یا مسائلی با پیش‌بینی‌کننده‌های بیشتر از مشاهدات استفاده شده است. در حالی که این رویکرد رگرسیون دو مرحله‌ای (کاهش ابعاد، سپس رگرسیون) با موفقیت برای توسعه مدل‌های پیش‌بینی تحت این شرایط استفاده شده است، به راحتی می‌توان آن را گمراه کرد. به‌طور خاص، کاهش ابعاد از طریق PCA لزوماً پیش‌بینی‌کننده‌های جدیدی تولید نمی‌کند که پاسخ را توضیح دهد. به‌عنوان نمونه‌ای از این سناریو، داده‌های شکل 1 را در نظر بگیرید.  [6. 8](#bookmark297) که شامل دو پیش‌بینی و یک پاسخ است. دو پیش‌بینی همبسته هستند و PCA این رابطه را با استفاده از جهت حداکثر تغییرات خلاصه می‌کند. نمودار سمت راست این شکل، با این حال، نشان می‌دهد که اولین جهت PCA حاوی هیچ اطلاعات پیش‌بینی‌ای در مورد پاسخ نیست.

همانطور که این مثال ساده نشان می‌دهد، PCA هنگام انتخاب اجزای خود هیچ جنبه‌ای از پاسخ را در نظر نمی‌گیرد. در عوض، به سادگی تغییرپذیری موجود در سراسر فضای پیش‌بینی را تعقیب می‌کند. اگر این تغییرات با تغییرپذیری پاسخ مرتبط باشد، PCR شانس خوبی برای شناسایی یک رابطه پیش‌بینی دارد. با این حال، اگر تغییرپذیری در فضای پیش‌بینی به تغییرپذیری پاسخ مرتبط نباشد، PCR می‌تواند در شناسایی یک رابطه پیش‌بینی در زمانی که واقعاً وجود داشته باشد، مسأله داشته باشد. به دلیل این مسأله ذاتی با PCR، زمانی که پیش‌بینی‌کننده‌های همبسته وجود دارد و راه حلی از نوع رگرسیون خطی مورد نظر است، توصیه می‌کنیم از PLS استفاده کنید.

PLS با الگوریتم حداقل مربعات جزئی تکراری غیرخطی (NIPALS) هرمان ولد [( Wold 1966](#bookmark1028) )، [1982](#bookmark1028) ) که مدل‌هایی را که در پارامترها غیرخطی بودند خطی کرد. متعاقبا، [ولد و همکاران ( 1983](#bookmark1028) ) روش NIPALS را برای تنظیم رگرسیون با پیش‌بینی‌کننده‌های همبسته و این اقتباس را "PLS" نامید. به‌طور خلاصه، الگوریتم NIPALS به‌طور مکرر به دنبال یافتن روابط زیربنایی یا نهفته در میان پیش‌بینی‌کننده‌هایی است که به شدت با پاسخ همبستگی دارند. برای یک پاسخ تک متغیره، هر تکرار الگوریتم رابطه بین پیش‌بینی‌کننده‌های ( X ) و پاسخ ( y ) را ارزیابی می‌کند و به صورت عددی این رابطه را با بردار وزن‌ها ( w ) خلاصه می‌کند. این بردار به‌عنوان یک *جهت نیز شناخته می‌شود.* سپس داده‌های پیش‌بینی به صورت متعامد در جهت ایجاد امتیاز ( t ) پیش‌تاب می‌شوند. سپس از امتیازها برای تولید بارگیری ( p ) استفاده می‌شود که همبستگی بردار امتیاز را با پیش‌بینی‌کننده‌های اصلی اندازه‌گیری می‌کند. در پایان هر تکرار، پیش‌بینی‌کننده‌ها و پاسخ به ترتیب با کم کردن تخمین فعلی ­ساختار پیش‌بینی و پاسخ، «کاهش» می‌شوند. سپس اطلاعات پاسخ و پیش‌بینی کاهش‌یافته جدید برای تولید مجموعه بعدی وزن‌ها، امتیازها و بارگذاری‌ها استفاده می‌شود. این مقادیر به ترتیب در ماتریس‌های W، T و P ذخیره می‌شوند و برای پیش‌بینی نمونه‌های جدید و محاسبه اهمیت پیش‌بینی استفاده می‌شوند. یک شماتیک از رابطه PLS بین پیش‌بینی‌کننده‌ها و پاسخ را می‌توان در شکل 1 مشاهده کرد.  [6. 9](#bookmark298)  و توضیح کاملی از الگوریتم را می‌توان در آن یافت [گلادی و کوالسکی](#bookmark1016) [( 1986](#bookmark1016) ).



-2

1

0

*X*

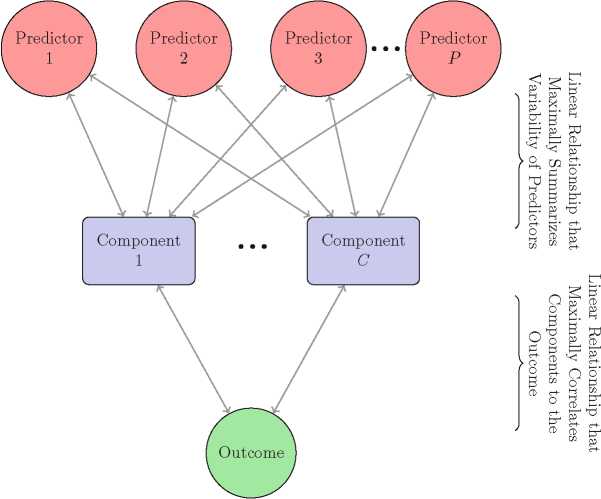
-1

-1.5 -1.0 -0.5 0.0 0.5 1.0 1.5

X1

Fig. 6.8: An example of principal component regression for a simple data set with two predictors and one response. *Left*: A scatter plot of the two predictors shows the direction of the first principal component. *Right*: The first PCA direction contains no predictive information for the response

برای به دست آوردن درک بهتر [g](#bookmark1025) از عملکرد الگوریتم، استون و بروکس [( 1990](#bookmark1025) ) آن را به مفاهیم آماری معروف کوواریانس و رگرسیون مرتبط کردند. به‌طور خاص، استون و بروکس نشان دادند که مانند PCA، PLS ترکیبات خطی پیش‌بینی‌کننده‌ها را پیدا می‌کند. این ترکیبات خطی معمولاً ­*جزء* یا متغیرهای پنهان نامیده می‌شوند. در حالی که ترکیب‌های خطی PCA ­برای خلاصه حداکثری تغییرات فضای پیش‌بینی انتخاب می‌شوند، ترکیب‌های خطی پیش‌بینی PLS برای خلاصه حداکثری کوواریانس با پاسخ انتخاب می‌شوند.



شکل 6. 9: نموداری که ساختار یک مدل PLS را نشان می‌دهد. PLS مؤلفه‌هایی را پیدا ­می‌کند که به‌طور همزمان تغییرات پیش‌بینی‌کننده‌ها را خلاصه می‌کنند در حالی که به‌طور بهینه با نتیجه همبستگی دارند.

این بدان معناست که PLS مؤلفه‌هایی را پیدا می‌کند که حداکثر تغییرات پیش‌بینی‌کننده‌ها را خلاصه می‌کنند در حالی که به‌طور همزمان نیاز دارند که این مؤلفه‌ها حداکثر همبستگی را با پاسخ داشته باشند. بنابراین PLS بین اهداف کاهش بعد فضای پیش‌بینی و یک رابطه پیش‌بینی با پاسخ مصالحه ایجاد می‌کند. به عبارت دیگر، PLS را می‌توان به‌عنوان یک روش کاهش بعد *نظارت شده مشاهده کرد.* PCR یک روش *بدون نظارت* است.

برای درک بهتر نحوه عملکرد PLS و ارتباط آن با PCR، داده‌های ارائه شده در شکل 1 را مجدداً بررسی خواهیم کرد.  [6. 8](#bookmark297) . این بار ما به دنبال اولین جزء PLS هستیم. نمودار پراکندگی سمت چپ در شکل.  [6. 10](#bookmark301) در تضاد اولین جهت PLS با جهت اول PCA است. برای این تصویر، دو جهت تقریباً متعامد هستند که نشان می‌دهد جهت کاهش ابعاد بهینه با حداکثر تغییرات در فضای پیش‌بینی مرتبط نیست. در عوض، PLS کاهش ابعاد فضای پیش‌بینی بهینه را به منظور رگرسیون با پاسخ شناسایی کرد.

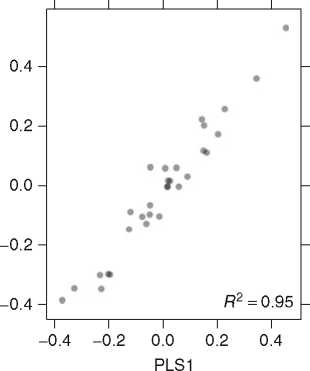
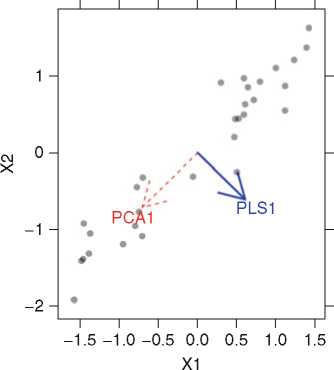
واضح است که این مثال برای نشان دادن یک نقص مهم در PCR طراحی شده است. در عمل، PCR این امر را به شدت شکست نمی‌دهد. بلکه PCR مدل‌هایی با توانایی پیش‌بینی مشابه PLS تولید می‌کند. بر اساس تجربه ما، تعداد اجزای حفظ شده از طریق اعتبارسنجی متقاطع با استفاده از PCR همیشه برابر است با یا بیشتر از تعداد اجزای نگهداشته شده توسط PLS. این به دلیل این واقعیت است که ابعاد حفظ شده توسط PLS به گونه‌ای انتخاب شده اند که به‌طور بهینه با پاسخ مرتبط باشند، در حالی که ابعاد انتخاب شده با PCR چنین نیستند.

Fig. 6.10: An example of partial least squares regression for a simple data set with two predictors and one response. *Left*: The first PLS direction is nearly orthogonal to the first PCA direction. *Right*: Unlike PCA, the PLS direction contains highly predictive information for the response

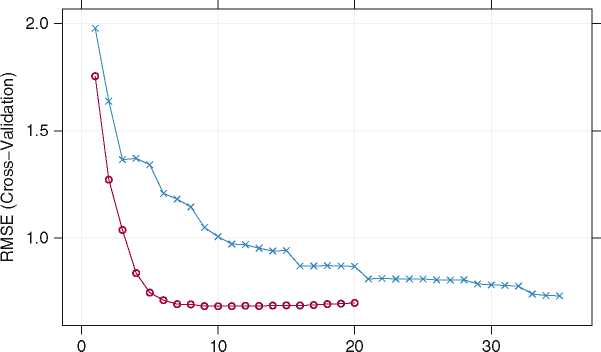
قبل از انجام PLS، پیش‌بینی‌کننده‌ها باید در مرکز و مقیاس‌بندی شوند، به‌ویژه اگر پیش‌بینی‌کننده‌ها در مقیاس‌هایی با اندازه‌های متفاوت باشند. همانطور که در بالا توضیح داده شد، PLS جهت حداکثر تغییرات را جستجو می‌کند در حالی که به‌طور همزمان همبستگی با پاسخ را در نظر می‌گیرد. حتی با وجود محدودیت ­همبستگی با پاسخ، به‌طور طبیعی تر به سمت پیش‌بینی‌کننده‌هایی با تغییرات زیاد کشیده می‌شود. بنابراین، پیش‌بینی‌کننده‌ها باید قبل از انجام PLS به اندازه کافی پیش پردازش شوند.

هنگامی که پیش‌بینی‌کننده‌ها از قبل پردازش شدند، متخصص می‌تواند پاسخ را با PLS مدل‌سازی کند. PLS یک پارامتر تنظیم دارد: تعداد اجزایی که باید حفظ شوند. تکنیک‌های نمونه‌گیری مجدد همانطور که در بخش توضیح داده شده است. برای تعیین تعداد بهینه اجزا می‌توان از [4. 4 استفاده کرد.](#bookmark210)

PCR و PLSR برای داده‌های حلالیت

برای نشان دادن فرآیند ساخت مدل با PLS، اجازه دهید به داده‌های حلالیت از Sect بازگردیم.  [6. 1 .](#bookmark281) اگرچه 228 پیش‌بینی وجود دارد، شکل.  [6. 4](#bookmark287) و [6. 5](#bookmark287) نشان می‌دهد که بسیاری از پیش‌بینی‌کننده‌ها به شدت همبستگی دارند و اطلاعات کلی در فضای پیش‌بینی در تعداد کمتری از ابعاد موجود است. این شرایط پیش‌بینی برای اعمال PLS بسیار مطلوب است.

PLS یا PCR x



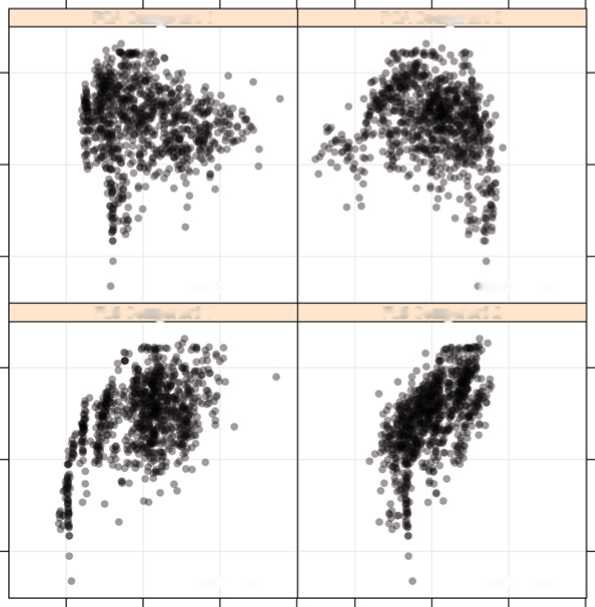
# اجزاء

شکل 6. 11: اعتبار متقابل RMSE توسط جزء برای PLS و PCR. RMSE با ده جزء PLS و 35 جزء PCR به حداقل می‌رسد

اعتبارسنجی متقابل برای تعیین تعداد بهینه اجزای PLS ­برای حفظ آن به حداقل رساندن RMSE استفاده شد. در همان زمان، PCR با استفاده از همان مجموعه‌های اعتبارسنجی متقاطع برای مقایسه عملکرد آن با PLS انجام شد. شکل ­ure [6. 11](#bookmark302) حاوی نتایج است که در آن PLS حداقل RMSE (0. 682) را با ده جزء و PCR حداقل RMSE (0. 731) را با 35 مؤلفه یافت. با این داده‌ها می‌بینیم که کاهش ابعاد نظارت‌شده حداقل RMSE با مؤلفه‌های بسیار کمتری نسبت به کاهش ابعاد بدون نظارت پیدا ­می‌کند. با استفاده از قانون خطای یک استاندارد (بخش.  [4. 6 )](#bookmark6) تعداد مؤلفه‌های PLS مورد نیاز را به 8 کاهش می‌دهد.

شکل [6. 12](#bookmark303) رابطه بین هر یک از دو جزء اول PCR و PLS را با پاسخ مقایسه می‌کند. از آنجا که RMSE برای هر یک از دو جزء اول PLS در مقایسه با دو جزء اول PCR کمتر است، جای تعجب نیست که همبستگی بین این اجزا و پاسخ برای PLS بیشتر از PCR باشد. این شکل نشان می‌دهد که PLS سریعتر به سمت رابطه اساسی با پاسخ هدایت می‌شود.

پیش‌بینی مجموعه تست با استفاده از مدل‌های بهینه PCR و PLS را می‌توان در شکل 1 مشاهده کرد.  [6. 13](#bookmark304) . توانایی پیش‌بینی هر روش خوب است و به نظر می‌رسد که باقیمانده‌ها به‌طور تصادفی در حدود صفر پراکنده شده اند. اگرچه ­توانایی پیش‌بینی این مدل‌ها نزدیک است، PLS مدل ساده‌تری پیدا می‌کند که از اجزای بسیار کمتری نسبت به PCR استفاده می‌کند.



-10

0

10

0

-5

10

PCA Component 1

PCA Component 2

corr: -0.08

corr: -0.22

PLS Component 1

PLS Component 2

corr: 0.52

corr: 0.59

5

10

-10

0

10

Component Score

Fig. 6.12: A contrast of the relationship between each of the first two PCR and PLS components with the solubility response. Because the dimension reduction offered by PLS is supervised by the response, it is more quickly steered towards the underlying relationship between the predictors and the response

ضرایب رگرسیون PLS برای داده‌های حلالیت در جدول ارائه شده است [6. 2](#bookmark316) (صفحه [127 )](#bookmark316)  و بزرگی‌ها مشابه مدل رگرسیون خطی است که فقط آن دو پیش‌بینی را شامل می‌شود.

از آنجایی که متغیرهای پنهان از PLS با استفاده از ترکیب خطی ­پیش‌بینی‌کننده‌های اصلی ساخته می‌شوند، کمی کردن سهم نسبی هر پیش‌بینی در مدل دشوارتر است.  [ولد و همکاران ( 1993](#bookmark1028) ) یک روش اکتشافی را برای ارزیابی اهمیت متغیر در هنگام استفاده از ریتم الگوی NIPALS معرفی کرد ­و این *اهمیت متغیر محاسبه را در طرح ریزی نامید.* در حالت ساده، فرض کنید که رابطه بین پیش‌بینی‌کننده‌ها و پاسخ را می‌توان به‌اندازه کافی با یک مدل PLS یک جزئی خلاصه کرد. اهمیت پیش‌بینی *j* با مقدار آن متناسب است

0

-2

-4

-6

-8

10

0

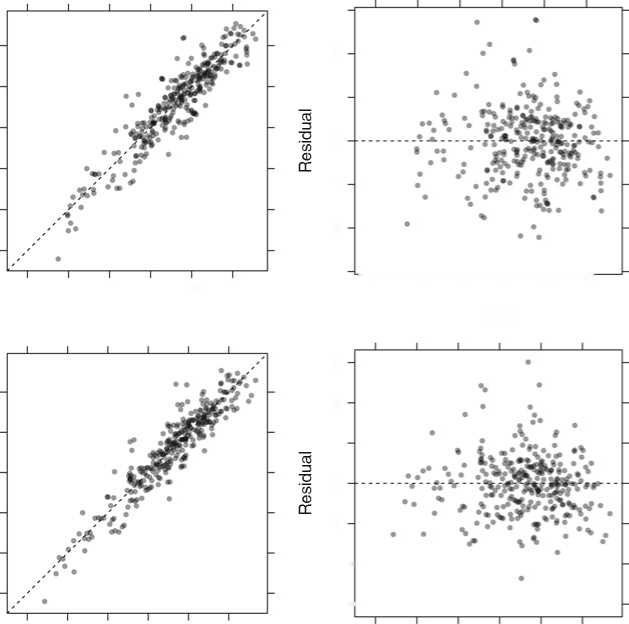
-2

-4

-6

-8

10

شکل 6. 13: *سمت چپ* : مقادیر مشاهده شده در مقابل مقادیر پیش‌بینی شده برای مجموعه تست حلالیت برای PCR ( *بالا* ) و PLS ( *پایین* ) *سمت راست* : باقیمانده‌ها در مقابل مقادیر پیش‌بینی شده برای PCR و PLS. به نظر می‌رسد باقیمانده‌ها به‌طور تصادفی در حدود 0 با توجه به مقادیر پیش‌بینی شده پراکنده شده اند. هر دو روش توانایی پیش‌بینی مشابهی دارند، اما PLS این کار را با مولفه‌های بسیار کمتری بردار وزن نرمال شده، *w* مطابق با پیش‌بینی *j انجام می‌دهد.* هنگامی که رابطه بین پیش‌بینی‌کننده‌ها و پاسخ به بیش از یک جزء نیاز دارد، محاسبه اهمیت متغیر بیشتر درگیر می‌شود. در این حالت، شمارنده اهمیت پیش‌بینی *j،* مجموع وزنی وزن‌های نرمال شده مربوط به پیش‌بینی *j است. j امین* وزن نرمال شده *k* امین مؤلفه، *w kj،* با مقدار تغییر در پاسخ توضیح داده شده توسط مولفه *k، مقیاس می‌*شود. مخرج اهمیت متغیر، مقدار کل تغییرات پاسخ است که توسط تمام اجزای *k توضیح داده شده است.* بنابراین، هر چه وزن نرمال شده و مقدار تغییرات پاسخ توضیح داده شده توسط مؤلفه بزرگتر باشد، پیش ­فاکتور مهمتری در مدل PLS است.

-10 -8 -6 -4 -2 0

Predicted

2

-2

3

2

-1

1

0

-1

1

0

2 -

— J

•3-

— 1

—i 1 1 1 1 r

-10 -8 -6 -4 -2 0

Predicted

-10

-8

-6

-4

-2

0

-10

-8

-6

-4

-2

0

Predicted

Predicted



اهمیت

شکل 6. 14: امتیازهای اهمیت متغیر حداقل مربعات جزئی برای داده‌های حلالیت

برای داده‌های حلالیت، 25 پیش‌بینی مهم در شکل 1 نشان داده شده است.  [6. 14 .](#bookmark307) هر چه مقدار VIP بزرگتر باشد، پیش‌بینی در ارتباط ساختار پیش‌بینی پنهان با پاسخ اهمیت بیشتری دارد. با ساخت آن، مجذور مقادیر VIP مجموع تعداد کل پیش‌بینی‌کننده‌ها می‌شود. به‌عنوان یک قانون سرانگشتی، مقادیر VIP بیش از 1 حاوی اطلاعات پیش‌بینی برای پاسخ در نظر گرفته می‌شود.  [وولد](#bookmark1028) [( 1995](#bookmark1028) ) همچنین پیشنهاد می‌کند که پیش‌بینی‌کننده‌های با ضرایب رگرسیون PLS کوچک و مقادیر VIP کوچک احتمالاً مهم نیستند و باید به‌عنوان کاندیدایی برای حذف از مدل در نظر گرفته شوند.

تغییرات الگوریتمی PLS

الگوریتم NIPALS برای مجموعه داده‌هایی با اندازه کوچک تا متوسط (مثلاً *کمتر* از 2500 نمونه و *کمتر از* 30 پیش‌بینی) نسبتاً کارآمد عمل می‌کند ( Alin 2009 ). اما وقتی تعداد نمونه‌ها ( *n* ) و پیش‌بینی‌کننده‌ها ( *P* ) بالا می‌رود، الگوریتم ناکارآمد می‌شود. این ناکارآمدی به دلیل نحوه انجام عملیات ماتریس بر روی پیش‌بینی‌کننده‌ها و پاسخ است. به‌طور خاص، هم ­ماتریس پیش فاکتور و هم پاسخ باید خالی شوند (یعنی اطلاعات باید از هر ماتریس کم شود، بنابراین نسخه‌های جدیدی از هر ماتریس ایجاد می‌شود) برای هر متغیر پنهان. این نشان می‌دهد که نسخه‌های مختلف ماتریس پیش‌بینی ­و پاسخ باید در هر تکرار الگوریتم حفظ شوند. بنابراین یک ماتریس *nx P و یک بردار nx 1 باید* در هر تکرار دوباره محاسبه، عمل و ذخیره شوند. همانطور که *n* و *P* رشد می‌کنند، نیازهای حافظه نیز افزایش می‌یابد و عملیات روی این ماتریس‌ها باید در طول فرآیند تکرار شونده انجام شود.

در یک گام محاسباتی رو به جلو، [لیندگرن و همکاران ( 1993](#bookmark1021) ) نشان داد که سازه‌های NIPALS را می‌توان با کار با ماتریس "هسته" با بعد *P x P،* ماتریس کوواریانس پیش‌بینی‌کننده‌ها (همچنین از بعد *P xP* ) و ماتریس کوواریانس پیش‌بینی‌کننده‌ها و پاسخ به دست آورد. از ابعاد *P x* 1). این تنظیم سرعت الگوریتم را بهبود بخشید، به خصوص که تعداد مشاهدات بسیار بیشتر از تعداد پیش‌بینی‌کننده‌ها شد.

تقریباً همزمان با توسعه رویکرد هسته، [دی یونگ](#bookmark1014) [( 1993](#bookmark1014) ) الگوریتم NIPALS را با مشاهده مسأله اساسی ­به‌عنوان یافتن متغیرهای متعامد پنهان در فضای پیش‌بینی که ­کوواریانس را با پاسخ به حداکثر می‌رساند، بهبود بخشید. این تغییر دیدگاه منجر به الگوریتم متفاوتی شد که بر کاهش تورم ماتریس کوواریانس بین ­پیش‌دیکاتورها و پاسخ به جای کاهش تورم ماتریس پیش‌بین و پاسخ متمرکز بود.  [دی یونگ](#bookmark1014) [( 1993](#bookmark1014) ) رویکرد جدید را "SIMPLS" نامید، زیرا این یک اصلاح ساده از الگوریتم PLS بود که از طریق آمار چارچوب‌بندی می‌شد. از آنجایی که رویکرد SIMPLS ماتریس کوواریانس را کاهش می‌دهد، ­نیاز به ذخیره‌سازی ماتریس کوواریانس تخلیه شده در هر تکرار دارد که دارای بعد *P x* 1 است - بهبود محاسباتی قابل‌توجهی نسبت به الزامات ذخیره‌سازی NIPALS. اگرچه رویکرد SIMPLS بهینه‌سازی را به ­روشی متفاوت حل می‌کند، [دی یونگ](#bookmark1014) [( 1993](#bookmark1014) ) نشان داد که متغیرهای نهفته SIMPLS با متغیرهای NIPALS زمانی که تنها یک پاسخ وجود دارد یکسان هستند. (هنگام مدلسازی پاسخ چند متغیره در زیر بیشتر مورد بحث قرار خواهد گرفت. )

سایر نویسندگان نیز اصلاحات محاسباتی را برای الگوریتم NIPALS از طریق تنظیمات هسته a [pp](#bookmark1014) roach پیشنهاد کرده اند ( de Jong and Ter Braak [1994](#bookmark1014) ; [دیال و مک گرگور 1997](#bookmark1014) ).  [دیال و مک گرگور](#bookmark1014) [( 1997](#bookmark1014) ) دو اصلاح کارآمد را توسعه دادند، به ویژه زمانی که *n>>P،* و، مشابه SIMPLS، فقط به کاهش تورم ماتریس کوواریانس بین پیش‌بینی‌کننده‌ها و پاسخ در هر مرحله از فرآیند تکراری نیاز دارد. در اولین تغییر آنها در عملکرد داخلی الگوریتم، ماتریس پیش‌بینی اصلی در محاسبات (بدون کاهش قیمت) استفاده می‌شود. در تغییر دوم، از ماتریس کوواریانس پیش‌بینی‌کننده‌ها در محاسبات استفاده می‌شود (همچنین بدون کاهش قیمت).

آلین ( 2009 ) یک مقایسه کارایی محاسباتی جامع NIPALS با سایر اصلاحات الگوریتمی ارائه کرد. در این کار، آلین از ­تعداد متفاوتی از نمونه‌ها (500-10000)، پیش‌بینی‌کننده‌ها (10-30)، پاسخ‌ها (1-15) و تعداد متغیرهای پنهان برای استخراج (3-10) استفاده کرد. تقریباً در هر سناریو، الگوریتم هسته دوم دیال و مک گرگور از نظر محاسباتی کارآمدتر از همه روش‌های دیگر بود و در مواردی که *n>* 2،500 و *P>* 30 بود، عملکرد برتر ارائه کرد. و در مواردی که الگوریتم دوم محاسباتی‌ترین را ارائه نمی‌کرد بازده، الگوریتم اول انجام داد.

رویکردهای بالا برای اجرای PLS مزایای محاسباتی واضحی را نسبت به الگوریتم اصلی ارائه می‌کنند. با این حال، با افزایش تعداد پیش‌بینی‌ها، هر کدام کمتر کارآمد می‌شوند. برای پرداختن به این سناریو زمانی که *P>n،* [Rannar و همکاران. ( 1994](#bookmark1024) ) هسته‌ای را بر اساس ماتریس پیش‌بینی و پاسخی ساخت که دارای بعد *nxn بود.* سپس یک تحلیل معمول PLS را می‌توان با استفاده از این هسته، محصولات بیرونی پیش‌بینی‌کننده‌ها و محصولات بیرونی پاسخ (هر کدام با بعد *nxn* ) انجام داد. . از این رو، این الگوریتم زمانی که پیش‌بینی‌کننده‌های بیشتری نسبت به نمونه‌ها وجود دارد، از نظر محاسباتی کارآمدتر است.

Asnotedinشکل. [6. 9 ،](#bookmark298) مؤلفه‌های PLS داده‌ها را از طریق زیرساخت‌های خطی (یعنی ابرصفحه‌ها) فضای پیش‌بینی اصلی که به پاسخ مرتبط هستند، خلاصه می‌کنند. اما برای بسیاری از مسائل، ساختار زیربنایی در فضای پیش‌بینی که به‌طور بهینه با پاسخ مرتبط است، خطی نیست بلکه منحنی یا غیرخطی است. چندین نویسنده تلاش کرده اند تا به این نقص PLS بپردازند تا این نوع رابطه فضا/پاسخ پیش‌بینی را بیابند. در حالی که روش‌های زیادی وجود دارد، آسان‌ترین رویکردها با استفاده از الگوریتم‌هایی که در بالا توضیح داده شد، ارائه می‌شوند [برگلوند و ولد](#bookmark1011) [( 1997](#bookmark1011) ) و [برگلوند و همکاران ( 2001](#bookmark1011) ). که در [Berglund و Wold ( 1997](#bookmark1011) )، ­مؤلفان نشان می‌دهند که افزودن پیش‌بینی‌کننده‌های مجذور (و مکعب، در صورت لزوم) می‌تواند با پیش‌بینی‌کننده‌های اصلی گنجانده شود. سپس PLS به مجموعه داده‌های تقویت شده اعمال می‌شود. نویسندگان همچنین نشان می‌دهند که نیازی به اضافه کردن اصطلاحات محصول متقابل وجود ندارد، بنابراین تعداد پیش‌بینی‌کننده‌های جدید اضافه‌شده به ­داده‌های اصلی تا حد زیادی کاهش می‌یابد. متعاقبا، [برگلوند و همکاران ( 2001](#bookmark1011) ) از رویکرد GIFI استفاده می‌کند [( Michailidis and de Leeuw 1998](#bookmark1022) ) که هر پیش‌بینی را به دو یا چند سطل تقسیم می‌کند تا پیش‌بینی‌کننده‌هایی که تصور می‌شود رابطه غیرخطی با پاسخ دارند. نقاط برش برای سطل‌ها توسط کاربر انتخاب می‌شود و بر اساس دانش قبلی یا ویژگی‌های داده‌ها است. سپس پیش‌بینی‌کننده‌های اصلی که صید شده بودند، از مجموعه داده‌ای که شامل نسخه‌های صحافی شده پیش‌بینی‌کننده‌ها است، حذف می‌شوند. سپس PLS بر روی مجموعه پیش‌بینی جدید به روش معمول اعمال می‌شود.

هر دوی این رویکردها با موفقیت روابط غیرخطی بین پیش‌بینی‌کننده‌ها و پاسخ پیدا کرده اند. اما ممکن است تلاش قابل‌توجهی در ساخت مجموعه داده‌ها برای ورودی PLS مورد نیاز باشد، به خصوص که تعداد پیش‌بینی‌کننده‌ها زیاد می‌شود. همانطور که در ­بخش‌های بعدی نشان خواهیم داد، سایر تکنیک‌های مدل‌سازی پیش‌بینی می‌توانند به‌طور طبیعی ساختارهای غیرخطی بین پیش‌بینی‌کننده‌ها و پاسخ را بدون نیاز به تغییر فضای پیش‌بینی شناسایی کنند. بنابراین، اگر یک رابطه ­پیچیده‌تر بین پیش‌بینی‌کننده‌ها و پاسخ وجود داشته باشد، پیشنهاد می‌کنیم به جای تلاش برای بهبود عملکرد PLS از طریق این نوع تقویت، از یکی از تکنیک‌های دیگر استفاده کنید.

**مدل‌های جریمه شده**

بر اساس مفروضات استاندارد، ضرایب تولید شده توسط رگرسیون حداقل مربعات معمولی بدون بایاس هستند و از بین تمامی تکنیک‌های خطی بدون بایاس، این مدل کمترین واریانس را نیز دارد. با این حال، با توجه به اینکه MSE یک است

ترکیبی از واریانس و بایاس (بخش.  [5. 2 )](#bookmark270) ، تولید مدل‌هایی با MSEهای کوچکتر با اجازه دادن به تخمین پارامترها برای بایاس بودن بسیار امکان‌پذیر است. معمول است که یک افزایش کوچک در بایاس می‌تواند افت قابل‌توجهی در واریانس ایجاد کند و بنابراین MSE کوچکتری نسبت به ضرایب رگرسیون حداقل مربعات معمولی ایجاد کند. یکی از نتایج همبستگی‌های بزرگ بین واریانس‌های پیش‌بینی ­این است که واریانس می‌تواند بسیار بزرگ شود. مبارزه با هم خطی با استفاده از مدل‌های بایاس ممکن است منجر به مدل‌های رگرسیونی شود که در آن MSE کلی رقابتی است.

یکی از روش‌های ایجاد مدل‌های رگرسیون بایاس، اضافه کردن جریمه به مجموع خطاهای مجذور است. به یاد بیاورید که رگرسیون حداقل مربعات اولیه تخمین پارامترها را برای به حداقل رساندن مجموع مربعات خطاها یافت:

n

SSE = ]T( *y i - y i* ) 2.

i = 1

زمانی که مدل بیش از حد با داده‌ها مطابقت دارد، یا زمانی که مسائلی در ارتباط با هم خطی وجود دارد (مانند جدول [6. 1 )](#bookmark293) ، تخمین پارامترهای رگرسیون خطی ممکن است ­مسطح شود. به این ترتیب، ممکن است بخواهیم بزرگی این برآوردها را برای کاهش SSE کنترل کنیم. کنترل (یا *منظم* کردن) تخمین‌های پارامتر را می‌توان با اضافه کردن یک جریمه به SSE در صورت بزرگ شدن تخمین‌ها انجام داد. *رگرسیون ریج* [( هورل 1970](#bookmark1018) ) جریمه‌ای بر مجموع ­پارامترهای رگرسیون مجذور اضافه می‌کند:

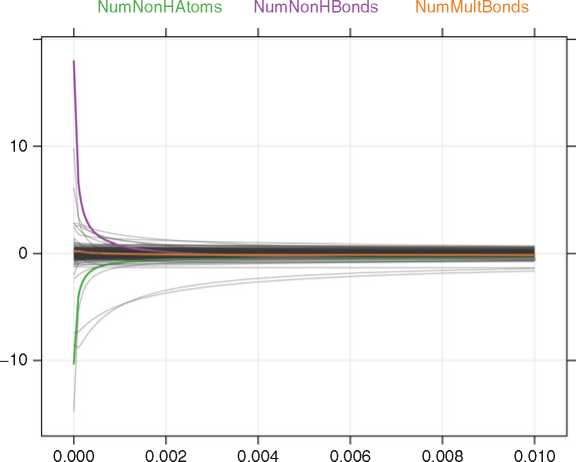
nP

SSE *L 2* = E( *y i - y i* ) 2 + *a* J>J.  *i* = 1 *j* = 1

" *L* 2 " نشان می‌دهد که یک جریمه مرتبه دوم (یعنی مربع) در تخمین پارامترها استفاده می‌شود. اثر این جریمه این است که تخمین پارامترها تنها در صورتی مجاز به بزرگ شدن هستند که کاهش متناسبی در SSE وجود داشته باشد. در واقع، این روش با بزرگ شدن جریمه *A،* تخمین‌ها را تا 0 *کوچک می‌*کند (این تکنیک‌ها گاهی اوقات "روش‌های انقباض" نامیده می‌شوند).

با اضافه کردن جریمه، ما بین واریانس مدل ­و بایاس مبادله‌ای ایجاد می‌کنیم. با قربانی کردن برخی بایاس‌ها، اغلب می‌توانیم واریانس را به اندازه‌ای کاهش دهیم که MSE کلی کمتر از مدل‌های بدون بایاس باشد.

به‌عنوان مثال، شکل.  [6. 15](#bookmark313) مسیر ضرایب رگرسیون *را* برای داده‌های حلالیت در مقادیر مختلف *A* نشان می‌دهد. هر خط مربوط به یک پارامتر مدل است و پیش‌بینی‌کننده‌ها قبل از این تحلیل در مرکز و مقیاس قرار گرفتند تا واحدهای آنها یکسان باشد. وقتی جریمه‌ای وجود ندارد، بسیاری از پارامترها مقادیر معقولی دارند، مانند پیش‌بینی تعداد پیوندهای متعدد (نشان داده شده به رنگ نارنجی). با این حال، برخی تخمین‌های پارامتر به‌طور غیرعادی بزرگ هستند، مانند تعداد اتم‌های غیر هیدروژن (به رنگ سبز) و تعداد پیوندهای غیر هیدروژنی (بنفش) که قبلاً در جدول مشخص شده است.  [6. 1](#bookmark293) . این مقادیر بزرگ نشان دهنده مسائل مربوط به هم خطی هستند. با افزایش جریمه، تخمین پارامترها با نرخ‌های مختلف به 0 نزدیک می‌شوند. تا آن زمان



پنالتی

شکل 6. 15: مسیر ضریب رگرسیون پشته

مقدار جریمه *A* = 0 *است.* 002، این دو پیش‌بینی بسیار بهتر رفتار می‌کنند، اگرچه سایر مقادیر ضرایب هنوز از نظر بزرگی نسبتاً بزرگ هستند.

با استفاده از اعتبارسنجی متقاطع، مقدار جریمه بهینه شد. شکل [6. 16](#bookmark314) نشان می‌دهد که چگونه RMSE با *A تغییر می‌*کند. وقتی پنالتی وجود ندارد، خطا متورم می‌شود. با افزایش جریمه، خطا از 0. 72 به 0. 69 کاهش می‌یابد. با افزایش جریمه بیش از 0. 036، بایاس به بزرگی تبدیل می‌شود و مدل شروع به برازش کمتر می‌کند و در نتیجه MSE افزایش می‌یابد.

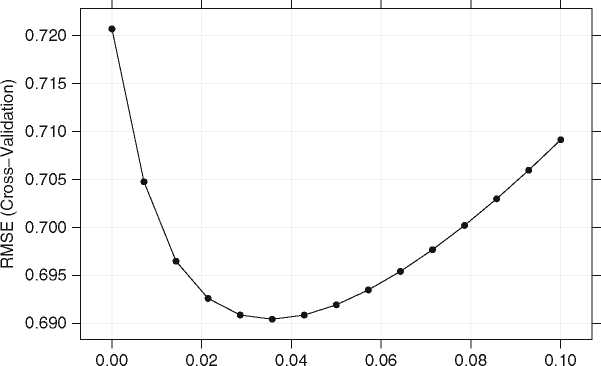
در حالی که رگرسیون خطی تخمین پارامترها را به سمت 0 کوچک می‌کند، مدل مقادیر را برای هیچ مقدار جریمه روی 0 مطلق تنظیم نمی‌کند. حتی اگر برخی تخمین‌های پارامتر بسیار کوچک می‌شوند، این مدل *انتخاب ویژگی را انجام نمی‌دهد.*

یک جایگزین محبوب برای رگرسیون رج، مدل *اپراتور حداقل انقباض و انتخاب است که معمولاً ریج* نامیده می‌شود. [( تیبشیرانی 1375](#bookmark1026) ). این مدل از جریمه مشابهی برای رگرسیون رج استفاده می‌کند:

nP

SSE *L* 1 =£( *y i - y i* ) 2 + *x^ |.   
i* = 1 *j* = 1

اگرچه ممکن است این یک اصلاح کوچک به نظر برسد، پیامدهای عملی آن قابل‌توجه است. در حالی که ضرایب رگرسیون هنوز به سمت 0 کاهش می‌یابد،



پنالتی

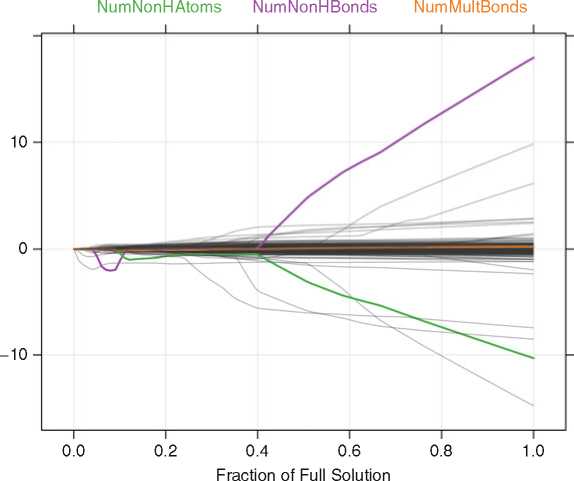
شکل 6. 16: پروفایل‌های اعتبارسنجی متقابل برای مدل رگرسیون پشته

نتیجه جریمه کردن مقادیر مطلق این است که برخی از پارامترها در واقع برای مقداری از *A روی 0 تنظیم می‌شوند.* بنابراین ریج مدل‌هایی را ارائه می‌دهد که ­به‌طور همزمان از منظم‌سازی برای بهبود مدل و انجام انتخاب ویژگی استفاده می‌کنند. در مقایسه، دو نوع مجازات، [فریدمن و همکاران ( 2010](#bookmark1016) ) بیان کرد

رگرسیون ریج ضرایب پیش‌بینی‌کننده‌های همبسته را نسبت به یکدیگر کوچک می‌کند و به آن‌ها اجازه می‌دهد قدرت را از یکدیگر قرض بگیرند. در حالت شدید *k* پیش‌بینی‌کننده‌های یکسان، هر کدام ضرایب یکسانی با 1 */k* ام به اندازه‌ای به دست می‌آورند که هر یک به تنهایی می‌توانست به دست آورد. [. . . ]

از سوی دیگر، ریج نسبت به پیش‌بینی‌کنندگان بسیار همبسته تا حدودی بی‌تفاوت است و تمایل دارد یکی را انتخاب کند و بقیه را نادیده بگیرد. »

شکل [6. 17](#bookmark315) مسیرهای ضرایب ریج را در مقادیر مختلف جریمه نشان می‌دهد. محور *x کسری از* جواب کامل است (یعنی حداقل مربعات معمولی بدون جریمه). مقادیر کوچکتر در محور *x* نشان می‌دهد که جریمه بزرگی استفاده شده است. هنگامی که جریمه بزرگ است، بسیاری از ضرایب رگرسیون روی 0 تنظیم می‌شوند. با کاهش جریمه، بسیاری از آنها دارای هم ­کارایی غیر صفر هستند. با بررسی ردیابی تعداد پیوندهای غیرهیدروژنی (به رنگ بنفش)، ضریب در ابتدا 0 است، کمی افزایش می‌یابد، سپس دوباره به سمت 0 کوچک می‌شود. وقتی کسری در حدود 0. 4 است، این پیش‌بینی با ضریب غیر صفر که به‌طور مداوم افزایش می‌یابد (به احتمال زیاد به دلیل همخطی بودن) دوباره وارد مدل می‌شود. جدول [6. 2](#bookmark316) ضرایب رگرسیون را برای حداقل مربعات معمولی، PLS، رگرسیون پشته و مدل ریج نشان می‌دهد. جریمه رگرسیون رج ­استفاده شده در این جدول 0. 036 و پنالتی ریج 0. 15 بوده است. مدل رگرسیون پشته ضرایب پیش‌بینی‌کننده‌های اتم غیرهیدروژن و پیوند غیرهیدروژنی را به‌طور قابل‌توجهی تا 0 در مقایسه با 0 کوچک می‌کند.



شکل 6. 17: مسیر ضریب ریج برای داده‌های حلالیت. محور *x کسری از* جواب حداقل مربعات کامل است. با افزایش کسر، جریمه ریج ( *A* ) کاهش می‌یابد

حداقل مربعات معمولی مدل می‌کند در حالی که مدل ریج ­پیش‌بینی اتم غیر هیدروژن را از مدل کوچک می‌کند. بین این مدل‌ها، مدل ریج کوچک‌ترین خطای اعتبار متقاطع 0. 67 را داشت که کمی بهتر از مدل PLS (0. 68) و رگرسیون رج (0. 69) بود.

این نوع منظم‌سازی یک حوزه تحقیقاتی بسیار فعال بوده است. مدل ریج به بسیاری از تکنیک‌های دیگر، مانند ­تحلیل تشخیص خطی [( Clemmensen et al. 2011](#bookmark1013) ; [ویتن و تیبشیرانی 2011](#bookmark1028) )، PLS [( چون و کلس 2010](#bookmark1013) ) و PCA [( Jolliffe et al. 2003](#bookmark1019) ; [زو و همکاران 2004](#bookmark1028) ). یک پیشرفت قابل‌توجه برای این مدل بود [افرون و همکاران ( 2004](#bookmark1015) ). مدل آنها، رگرسیون حداقل زاویه (LARS)، یک چارچوب گسترده است که ریج و مدل‌های مشابه را در بر می‌گیرد. مدل LARS را می‌توان برای جا انداختن مدل‌های ریج به‌طور کارآمدتر، به خصوص در مسائل با ابعاد بالا استفاده کرد.  [فریدمن و همکاران](#bookmark1016) [( 2010](#bookmark1016) ) و [هستربرگ و همکاران ( 2008](#bookmark1018) ) بررسی این تکنیک‌ها را ارائه می‌دهد.

یک تعمیم از مدل ریج، *شبکه الاستیک است* [( زو و هستی 2005](#bookmark1028) ). این مدل دو نوع مجازات را ترکیب می‌کند:

n PP

SSE Enet = £( *y i - y i* ) 2 + *A* 1 £ $ + *A* 2 £ *^ |.*

i = 1 j = 1 j = 1

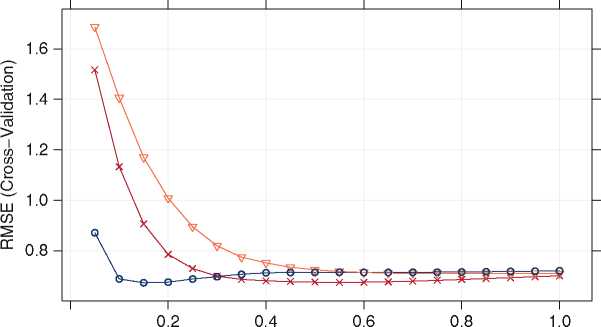
جدول 6. 2: ضرایب رگرسیون برای دو پیش‌بینی بسیار همبسته برای PLS، رگرسیون پشته، شبکه الاستیک و سایر مدل ها

|  |  |  |
| --- | --- | --- |
| مدل | NumNonHAtoms | NumNonHBonds |
| فقط NumNonHAtoms | *-* 1. 2 (0. 1) |  |
| فقط NumNonHBonds |  | *-* 1. 2 (0. 1) |
| هر دو | *-* 0 *\_* 3 (0. 5) | *-* 0 *\_* 9 (0. 5) |
| همه پیش‌بینی ها | 8. 2 (1. 4) | *-* 9. 1 (1. 6) |
| PLS، همه پیش‌بینی ها | *-* 0 *\_* 4 | *-* 0 *\_* 8 |
| ریج، همه پیش‌بینی ها | *-* 0 *\_* 3 | *-* 0 *\_* 3 |
| توری ریج/الاستیک | 0. 0 | *-* 0 *\_* 8 |

پنالتی رج استفاده شده برای این جدول 0. 036 و پنالتی ریج 0. 15 بود. مدل PLS از ده جزء استفاده می‌کند.

کاهش وزن

0 O 0. 01 X 0. 1 v -



کسری از محلول کامل

شکل 6. 18: پروفیل‌های اعتبارسنجی متقاطع برای یک مدل شبکه الاستیک

مزیت این مدل این است که تنظیم موثر از طریق پنالتی نوع رج با کیفیت انتخاب ویژگی پنالتی ریج را امکان‌پذیر می‌کند. [زو و](#bookmark1028) هیستی [( 2005](#bookmark1028) ) نشان می‌دهد که این مدل به‌طور موثرتری با گروه‌های پیش‌بینی‌کننده‌های همبستگی بالا برخورد می‌کند.

هر دو مجازات برای دستیابی به عملکرد مطلوب نیاز به تنظیم دارند. مجدداً، با استفاده از نمونه‌گیری مجدد، این مدل برای داده‌های حلالیت تنظیم شد. شکل [6. 18](#bookmark316) نمایه‌های عملکرد را در سه مقدار جریمه رج و 20 مقدار پنالتی ریج نشان می‌دهد. مدل ریج خالص (با *A* 1 = 0) دارای یک افت اولیه در خطا و سپس افزایش زمانی که کسر بزرگتر از 0. 2 است. دو مدل با مقادیر غیر صفر جریمه پشته دارای حداقل خطا با مدل بزرگتر هستند. در نهایت، عملکرد بهینه با مدل ریج با کسری از 0. 15، مربوط به 130 پیش‌بینی از 228 ممکن همراه بود.

6. 5 محاسبات

بسته‌های R elasticnet، caret، lars، MASS، pls و stats ارجاع خواهند شد.

داده‌های حلالیت را می‌توان از بسته AppliedPredictiveModeling R به دست آورد. پیش‌بینی‌کننده‌های مجموعه‌های آموزشی و آزمایشی به ترتیب در چارچوب‌های داده به نام‌های solTrainX و solTestX قرار دارند. برای بدست آوردن داده‌ها در R

*> کتابخانه (AppliedPredictiveModeling)*

*داده (حلالیت)*

*## اشیاء داده با "sol" شروع می‌شوند:*

*ls (الگو = ""solT")*

[1] "solTestX" "solTestXtrans" "solTestY"

"solTrainX"

[5] "solTrainXtrans" "solTrainY"

هر ستون از داده‌ها به یک پیش‌بینی (یعنی توصیفگر شیمیایی) و ردیف‌ها مربوط به ترکیبات است. 228 ستون در داده‌ها وجود دارد.

یک نمونه تصادفی از نام ستون است

*set. seed (2)*

*نمونه (نام (solTrainX)، 8)*

[1] "FP043" "FP160" "FP130"

"FP038"

"NumBonds"

[6] "NumNonHAtoms" "FP029" "FP185"

« FP » با پیش‌بینی‌کننده‌های اثر انگشت باینری 0/1 مطابقت دارند که با وجود یا عدم وجود یک ساختار شیمیایی خاص مرتبط هستند. نسخه‌های جایگزین این داده‌ها که باکس-کاکس تبدیل شده اند در فریم‌های داده solTrainXtrans و solTestXtrans موجود هستند. این نسخه‌های اصلاح شده در تحلیل‌های این فصل و فصل‌های بعدی مورد استفاده قرار گرفتند.

مقادیر حلالیت برای هر ترکیب در بردارهای عددی به نام‌های solTrainY و solTestY موجود است.

رگرسیون خطی معمولی

تابع اولیه برای ایجاد مدل‌های رگرسیون خطی با استفاده از حداقل مربعات ساده lm است. این تابع یک فرمول و فریم داده را به‌عنوان ورودی می‌گیرد. به همین دلیل، پیش‌بینی‌کننده‌های مجموعه آموزشی و نتیجه باید در چارچوب داده یکسانی قرار گیرند. ما می‌توانیم یک چارچوب داده جدید برای این منظور ایجاد کنیم:

*trainingData <- solTrainXtrans*

*## نتیجه حلالیت را اضافه کنید*

*trainingData$حلالیت <- solTrainY*

برای تطبیق یک مدل خطی با همه پیش‌بینی‌کننده‌های وارد شده در مدل به‌عنوان عبارت‌های خطی ساده و مستقل، میانبر فرمول حلالیت *~.* میتواند مورد استفاده قرار گیرد:

*lmFitAllPredictors <- lm(حلالیت ~. ، داده = آموزش داده)*

یک عبارت عرض از مبدأ به‌طور خودکار به مدل اضافه می‌شود. روش خلاصه آمار خلاصه مدل، تخمین پارامترها، خطاهای استاندارد آنها و مقادیر *p* را برای آزمایش اینکه آیا هر ضریب فردی متفاوت از 0 است را نمایش می‌دهد:

*خلاصه (lmFitAllPredictors)*

صدا زدن:

lm (فرمول = حلالیت ~. ، داده = آموزش داده)

باقیمانده ها:

حداقل 1Q میانه 3Q حداکثر

-1. 75620 -0. 28304 0. 01165 0. 30030 1. 54887

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ضرایب: | تخمین زدن | Std. خطا | مقدار t | Pr(>|t|) |
| (عرض از مبدأ) | 2. 431e+00 | 2. 162e+00 | 1. 124 | 0. 261303 |
| FP001 | 3. 594e-01 | 3. 185e-01 | 1. 128 | 0. 259635 |
| FP002 | 1. 456e-01 | 2. 637e-01 | 0. 552 | 0. 580960 |
| FP003 | -3. 969e-02 | 1. 314e-01 | -0. 302 | 0. 762617 |
| FP004 | -3. 049e-01 | 1. 371e-01 | -2. 223 | 0. 026520 \* |
| FP005 | 2. 837e+00 | 9. 598e-01 | 2. 956 | 0. 003223 \*\* |
| FP006 | -6. 886e-02 | 2. 041e-01 | -0. 337 | 0. 735917 |
| FP007 | 4. 044e-02 | 1. 152e-01 | 0. 351 | 0. 725643 |
| FP008 | 1. 121e-01 | 1. 636e-01 | 0. 685 | 0. 493331 |
| FP009 | -8. 242e-01 | 8. 395e-01 | -0. 982 | 0. 326536 |

|  |  |  |  |
| --- | --- | --- | --- |
| NumAtoms مولی وزن | -1. 232e+00  -1. 478e+01 | 2. 296e-01  3. 473e+00 | -5. 365 1. 09e-07 \*\*\*  -4. 257 2. 35e-05 \*\*\* |
| NumNonHAtoms | 1. 795e+01 | 3. 166e+00 | 5. 670 2. 07e-08 \*\*\* |
| NumBonds | 9. 843e+00 | 2. 681e+00 | 3. 671 0. 000260 \*\*\* |
| NumNonHBonds | -1. 030e+01 | 1. 793e+00 | -5. 746 1. 35e-08 \*\*\* |
| NumMultBonds | 2. 107e-01 | 1. 754e-01 | 1. 201 0. 229990 |
| NumRotBonds | -5. 213e-01 | 1. 334e-01 | -3. 908 0. 000102 \*\*\* |
| NumDblBonds | -7. 492e-01 | 3. 163e-01 | -2. 369 0. 018111 \* |
| NumAromaticBonds | -2. 364e+00 | 6. 232e-01 | -3. 794 0. 000161 \*\*\* |
| NumHydrogen | 8. 347e-01 | 1. 880e-01 | 4. 439 1. 04e-05 \*\*\* |
| NumCarbon | 1. 730e-02 | 3. 763e-01 | 0. 046 0. 963335 |
| NumNitrogen | 6. 125e+00 | 3. 045e+00 | 2. 011 0. 044645 \* |
| NumOxygen | 2. 389e+00 | 4. 523e-01 | 5. 283 1. 69e-07 \*\*\* |
| NumSulfer | -8. 508e+00 | 3. 619e+00 | -2. 351 0. 018994 \* |
| NumClorine | -7. 449e+00 | 1. 989e+00 | -3. 744 0. 000195 \*\*\* |
| NumHalogen | 1. 408e+00 | 2. 109e+00 | 0. 668 0. 504615 |
| NumRings | 1. 276e+00 | 6. 716e-01 | 1. 901 0. 057731. |
| فاکتور هیدروفیل | 1. 099e-02 | 1. 137e-01 | 0. 097 0. 922998 |

SurfaceArea1 8. 825e-02 6. 058e-02 1. 457 0. 145643

SurfaceArea2 9. 555e-02 5. 615e-02 1. 702 0. 089208.

---

Signif. کدها: 0 '\*\*\*' 0. 001 '\*\*' 0. 01 '\*' 0. 05 '. ' 0. 1 "" 1

خطای استاندارد باقیمانده: 0. 5524 در 722 درجه آزادی

R-squared چندگانه: 0. 9446، R-squared تنظیم شده: 0. 9271

آماره F: 54. 03 در 228 و 722 DF، p-value: < 2. 2e-16

(از آنجایی که 229 پیش‌بینی در مدل وجود دارد، خروجی بسیار طولانی است و نتایج کوتاه شده اند. ) بحث جامع تر از مدل‌های خطی در R را می‌توان در [خیلی دور](#bookmark1016) [( 2005](#bookmark1016) ).

برآوردهای ساده RMSE و *R2* به ترتیب 0. 55 و 0. 945 بودند. توجه داشته باشید که این مقادیر احتمالاً بسیار خوش بینانه هستند زیرا با پیش‌بینی مجدد داده‌های مجموعه آموزشی به دست آمده اند.

برای محاسبه مقادیر حلالیت مدل برای نمونه‌های جدید، از روش پیش‌بینی استفاده می‌شود:

*lmPred1 <- پیش‌بینی (lmFitAllPredictors، solTestXtrans)*

*سر (lmPred1)*

20 21 23 25 28 31

0. 99370933 0. 06834627 -0. 69877632 0. 84796356 -0. 16578324 1. 40815083

می‌توانیم مقادیر مشاهده‌شده و پیش‌بینی‌شده را در یک چارچوب داده جمع‌آوری کنیم، سپس از تابع caret defaultSummary برای تخمین عملکرد مجموعه آزمایشی استفاده کنیم:

*lmValues1 <- data. frame(obs = solTestY، pred = lmPred1)*

*خلاصه پیش فرض (lmValues1)*

RMSE Rsquared

0. 7455802 0. 8722236

بر اساس مجموعه آزمون، خلاصه‌های تولید شده توسط تابع خلاصه برای lm خوش بینانه بودند.

اگر ما یک مدل رگرسیون خطی قوی می‌خواهیم، می‌توان از تابع مدل خطی قوی ( rlm ) از بسته MASS استفاده کرد که به‌طور پیش‌فرض ­از رویکرد Huber استفاده می‌کند. مشابه تابع lm، rlm به صورت زیر نامیده می‌شود:

*rlmFitAllPredictors <- rlm(حلالیت ~. ، داده = آموزش داده)*

تابع آموزش یک تخمین نمونه برداری مجدد از عملکرد تولید می‌کند. از آنجایی که اندازه مجموعه آموزشی کوچک نیست، اعتبارسنجی متقاطع 10 برابری باید برآوردهای معقولی از عملکرد مدل ایجاد کند. ­تابع trainControl نوع نمونه‌گیری مجدد را مشخص ­می‌کند:

*ctrl <- trainControl(روش = "cv"، عدد = 10)*

آموزش فرمول مدل یا رابط غیر فرمول را می‌پذیرد (به بخش مراجعه کنید.  [4. 9](#bookmark245) برای خلاصه‌ای از روش‌های مختلف برای تعیین مدل‌های پیش‌بینی). رابط غیر فرمول است

*set. seed (100)*

*lmFit1 <- train(x = solTrainXtrans، y = solTrainY،*

*+ روش = "lm"، trControl = ctrl)*

دانه اعداد تصادفی قبل از مدل‌سازی تنظیم می‌شود تا بتوان نتایج را بازتولید کرد. نتایج عبارتند از:

*lmFit1*

951 نمونه

228 پیش‌بینی

بدون پیش پردازش

نمونه‌گیری مجدد: اعتبارسنجی متقاطع (10 برابر)

خلاصه حجم نمونه: 856، 857، 855، 856، 856، 855،. . .

نتایج نمونه‌گیری مجدد

RMSE Rsquared RMSE SD Rsquared SD 0. 721 0. 877 0. 07 0. 0247

برای مدل‌هایی که برای *توضیح ساخته شده اند،* بررسی مفروضات مدل، مانند توزیع باقیمانده، مهم است. برای مدل‌های پیش‌بینی، برخی از تکنیک‌های تشخیصی مشابه می‌توانند مناطقی را که مدل به خوبی پیش‌بینی نمی‌کند، روشن کند. به‌عنوان مثال، ما می‌توانیم باقیمانده‌ها را در مقابل مقادیر پیش‌بینی شده برای مدل رسم کنیم. اگر نمودار یک ابر تصادفی از نقاط را نشان دهد، ما احساس راحتی می‌کنیم که هیچ عبارت اصلی در مدل وجود ندارد (مانند عبارت درجه دوم و غیره) یا مقادیر پرت قابل‌توجهی. نمودار مهم دیگر مقادیر پیش‌بینی‌شده در مقابل مقادیر مشاهده‌شده برای ارزیابی نزدیکی پیش‌بینی‌کننده‌ها ­به مقادیر واقعی است. دو روش برای انجام این کار (با استفاده از نمونه مجموعه آموزشی می‌باشد

*xyplot(solTrainY ~ predict(lmFit1)*

*+ ## نقاط را رسم کنید (نوع =* ' *p* ' *) و یک شبکه پس زمینه (* ' *g* ' *)*

*+ نوع = c("p"، "g")،*

*+ xlab = "پیش‌بینی"، ylab = "مشاهده")*

*xyplot(resid(lmFit1) ~ predict(lmFit1)*

*+ نوع = c("p"، "g")،*

*+ xlab = "پیش‌بینی شده"، ylab = "باقیمانده ها")*

نتایج در شکل نمایش داده شده اند.  [6. 19](#bookmark324) . توجه داشته باشید که تابع resid باقیمانده‌های مدل را برای مجموعه آموزشی تولید می‌کند و استفاده از تابع ­پیش‌بینی بدون آرگومان داده اضافی، مقادیر پیش‌بینی شده را برای مجموعه آموزشی برمی گرداند. برای این مدل، هیچ علامت هشدار آشکاری در نمودارهای تشخیصی وجود ندارد.

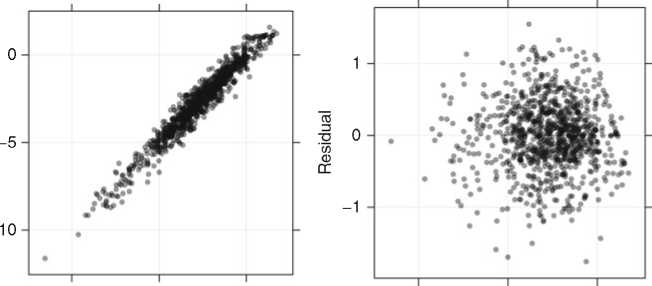
برای ساخت یک مدل کوچکتر بدون پیش‌بینی با همبستگی بسیار بالا، می‌توانیم از روش‌های Sect استفاده کنیم.  [3. 3](#bookmark147) برای کاهش تعداد پیش‌بینی‌کننده‌ها به‌طوری که هیچ همبستگی زوجی مطلق بالای 0. 9 وجود نداشته باشد:

*corThresh <-. 9*

*tooHigh <- findCorrelation(cor(solTrainXtrans)، corThresh)*

*corrPred <- names(solTrainXtrans)[tooHigh]*

*trainXfiltered <- solTrainXtrans[, -tooHigh]*



-10 -5 0 -10 -5 0

پیش‌بینی شده پیش بینی شد

شکل 6. 19: نمودارهای تشخیصی برای مدل خطی با استفاده از مجموعه آموزشی. *سمت چپ* : نموداری از مقادیر مشاهده شده در مقابل مقادیر پیش‌بینی شده. این نمودار می‌تواند نقاط پرت یا مناطقی را که مدل کالیبره نشده است را نشان دهد. *راست* : نموداری از باقیمانده‌ها در مقابل مقادیر پیش‌بینی شده. اگر مدل به خوبی مشخص شده باشد، این نمودار باید یک ابر تصادفی از نقاط بدون نقاط پرت یا الگو باشد (به‌عنوان مثال، یک شکل قیف)

*testXfiltered <- solTestXtrans[, -tooHigh]*

*set. seed (100)*

*lmFiltered <- train(solTrainXtrans, solTrainY, method = "lm",*

*+ trControl = ctrl)*

*lmFiltered*

951 نمونه

228 پیش‌بینی

بدون پیش پردازش

نمونه‌گیری مجدد: اعتبارسنجی متقاطع (10 برابر)

خلاصه حجم نمونه: 856، 857، 855، 856، 856، 855،. . .

نتایج نمونه‌گیری مجدد

RMSE Rsquared RMSE SD Rsquared SD

0. 721 0. 877 0. 07 0. 0247

رگرسیون خطی قوی همچنین می‌تواند با استفاده از تابع آموزش که از تابع rlm استفاده می‌کند، انجام شود. با این حال، توجه به این نکته مهم است که rlm اجازه نمی‌دهد که ماتریس کوواریانس پیش‌بینی‌کننده‌ها منفرد باشد (برخلاف تابع lm ). برای اطمینان از اینکه پیش‌بینی‌کننده‌ها تکی نیستند، پیش‌بینی‌کننده‌ها را با استفاده از PCA از قبل پردازش می‌کنیم. با استفاده از مجموعه فیلتر شده از پیش‌بینی‌ها، عملکرد مدل رگرسیون قوی است

*set. seed (100)*

*rlmPCA <- train(solTrainXtrans, solTrainY,*

*+ روش = "rlm"،*

*+ preProcess = "pca"،*

*+ trControl = ctrl)*

*rlmPCA*

951 نمونه

228 پیش‌بینی

پیش پردازش: استخراج سیگنال جزء اصلی، مقیاس‌بندی شده، متمرکز

نمونه‌گیری مجدد: اعتبارسنجی متقاطع (10 برابر)

خلاصه حجم نمونه: 856، 857، 855، 856، 856، 855،. . .

نتایج نمونه‌گیری مجدد

RMSE Rsquared RMSE SD Rsquared SD

0. 782 0. 854 0. 0372 0. 0169

حداقل مربعات جزئی

بسته pls [( Mevik and Wehrens 2007](#bookmark1022) ) دارای عملکردهایی برای PLS و PCR است. SIMPLS، اولین الگوریتم دیال و مک گرگور و ریتم ­الگوریتم توسعه یافته توسط [رانار و همکاران ( 1994](#bookmark1024) ) هر کدام در دسترس هستند. به‌طور پیش‌فرض، بسته pls از اولین الگوریتم هسته Dayal و MacGregor استفاده می‌کند در حالی که الگوریتم‌های دیگر را می‌توان با استفاده از آرگومان ­متد با استفاده از مقادیر «oscorespls»، «simpls» یا «widekernelpls» مشخص کرد. تابع plsr مانند تابع lm به یک فرمول مدل نیاز دارد:

*plsFit <- plsr (حلالیت ~. ، داده = آموزش داده)*

تعداد کامپوننت‌ها را می‌توان با استفاده از آرگومان ncomp ثابت کرد یا اگر به حالت پیش فرض رها شود، حداکثر تعداد کامپوننت‌ها محاسبه می‌شود. پیش‌بینی‌کننده‌های نمونه‌های جدید را می‌توان با استفاده از تابع پیش‌بینی محاسبه کرد. پیش‌بینی‌کننده‌ها را می‌توان برای تعداد مشخصی از مؤلفه‌ها یا برای چندین مقدار در یک زمان انجام داد. مثلا

*> پیش‌بینی (plsFit، solTestXtrans[1:5،]، ncomp = 1:2)*

, , 1 comps

انحلال پذیری

20 -1. 789335

21 -1. 427551

23 -2. 268798

25 -2. 269782

28 -1. 867960

, , 2 کمپ

انحلال پذیری

20 0. 2520469

21 0. 3555028

23 -1. 8795338

25 -0. 6848584

28 -1. 5531552

تابع plsr دارای گزینه‌هایی برای اعتبار سنجی متقاطع *K -fold* یا leave-one-out (از طریق آرگومان اعتبارسنجی ) یا الگوریتم PLS برای استفاده است، مانند SIMPLS (با استفاده از آرگومان متد ).

چندین تابع کمکی برای استخراج مؤلفه‌های PLS (در بارگذاری‌های تابع )، امتیازات PLS ( نمرات ) و مقادیر دیگر وجود دارد. تابع طرح دارای تجسم برای بسیاری از جنبه‌های مدل است.

آموزش همچنین می‌تواند با مقادیر متد pls مانند "oscorespls"، "simpls" یا "widekernelpls" استفاده شود. مثلا

*> set. seed(100)*

*> plsTune <- train(solTrainXtrans, solTrainY,*

|  |  |
| --- | --- |
| *+*  *+* | *روش = "pls"*  *## شبکه تنظیم پیش فرض ارزیابی می‌شود* |
| *+*  *+* | *## کامپوننت 1. . . تنظیم طول تنظیم طول = 20،* |
| *+*  *+* | *trControl = ctrl، preProc = c("مرکز"، "مقیاس"))* |

این کد مدل PLS نمایش داده شده در شکل 1 را بازتولید می‌کند.  [6. 11](#bookmark302) .

مدل‌های رگرسیون مجازات شده

مدل‌های رگرسیون ریج را می‌توان با استفاده از تابع lm. ridge در بسته MASS یا تابع enet در بسته الاستیک نت ایجاد کرد. هنگام فراخوانی تابع enet، آرگومان لامبدا جریمه رگرسیون ridge را مشخص می‌کند:

*ridgeModel <- enet(x = as. matrix(solTrainXtrans)، y = solTrainY، + lambda = 0. 001)*

به یاد بیاورید که مدل شبکه الاستیک دارای هر دو پنالتی برآمدگی و ریج است و در این مرحله، شیء RridgeModel فقط مقدار جریمه رج را ثابت کرده است. جریمه ریج را می‌توان به‌طور موثر برای بسیاری از مقادیر پنالتی محاسبه کرد. تابع پیش‌بینی برای اشیاء enet پیش‌بینی‌کننده‌هایی را برای یک یا چند مقدار جریمه ریج به‌طور همزمان با استفاده از آرگومان‌های ­s و mode ایجاد می‌کند. برای رگرسیون ریج، ما فقط یک پنالتی ریج 0 را می‌خواهیم، بنابراین راه‌حل کامل را می‌خواهیم. برای تولید یک جواب رگرسیون پشته، s=1 را با mode = "کسری" تعریف می‌کنیم. این گزینه آخر نحوه تعیین میزان جریمه ­را مشخص می‌کند. در این مورد، مقدار 1 مربوط به یک جناح 1 است، یعنی جواب کامل:

*ridgePred <- predict(ridgeModel, newx = as. matrix(solTestXtrans)*

*+ s = 1، حالت = "کسری"،*

*+ نوع = "مناسب")*

*سر (ridgePred$fit)*

20 21 23 25 28 31

0. 96795590 0. 06918538 -0. 54365077 0. 96072014 -0. 03594693 1. 59284535

برای تنظیم پنالتی، آموزش می‌تواند با روش متفاوتی استفاده شود:

*## مجموعه مقادیر نامزد را تعریف کنید*

*ridgeGrid <- data. frame(. lambda = seq(0، 0. 1، طول = 15))*

*set. seed (100)*

*> ridgeRegFit <- train(solTrainXtrans, solTrainY,*

*+ روش = "خط الراس"،*

*+ ## مدل را روی بسیاری از مقادیر جریمه قرار دهید*

*+ tuneGrid = ridgeGrid،*

*+ trControl = ctrl،*

*+ ## پیش‌بینی‌کننده‌ها را در یک مقیاس قرار دهید*

*+ preProc = c ("مرکز"، "مقیاس"))*

*> ridgeRegFit*

951 نمونه

228 پیش‌بینی

پیش پردازش: متمرکز، مقیاس شده

نمونه‌گیری مجدد: اعتبارسنجی متقاطع (10 برابر)

خلاصه حجم نمونه: 856، 857، 855، 856، 856، 855،. . .

نمونه برداری مجدد از نتایج در پارامترهای تنظیم:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| لامبدا | RMSE | Rsquared | RMSE SD | Rsquared SD |
| 0 | 0. 721 | 0. 877 | 0. 0699 | 0. 0245 |
| 0. 00714 | 0. 705 | 0. 882 | 0. 045 | 0. 0199 |
| 0. 0143 | 0. 696 | 0. 885 | 0. 0405 | 0. 0187 |
| 0. 0214 | 0. 693 | 0. 886 | 0. 0378 | 0. 018 |
| 0. 0286 | 0. 691 | 0. 887 | 0. 0359 | 0. 0175 |
| 0. 0357 | 0. 69 | 0. 887 | 0. 0346 | 0. 0171 |
| 0. 0429 | 0. 691 | 0. 888 | 0. 0336 | 0. 0168 |
| 0. 05 | 0. 692 | 0. 888 | 0. 0329 | 0. 0166 |
| 0. 0571 | 0. 693 | 0. 887 | 0. 0323 | 0. 0164 |
| 0. 0643 | 0. 695 | 0. 887 | 0. 032 | 0. 0162 |
| 0. 0714 | 0. 698 | 0. 887 | 0. 0319 | 0. 016 |
| 0. 0786 | 0. 7 | 0. 887 | 0. 0318 | 0. 0159 |
| 0. 0857 | 0. 703 | 0. 886 | 0. 0318 | 0. 0158 |
| 0. 0929 | 0. 706 | 0. 886 | 0. 032 | 0. 0157 |
| 0. 1 | 0. 709 | 0. 885 | 0. 0321 | 0. 0156 |

برای انتخاب مدل بهینه با استفاده از کوچکترین مقدار از RMSE استفاده شد. مقدار نهایی استفاده شده برای مدل لامبدا = 0. 0357 بود.

مدل ریج را می‌توان با استفاده از تعدادی توابع مختلف تخمین زد. بسته lars شامل تابع lars، بسته elasticnet دارای enet و بسته glmnet دارای تابعی به همین نام است. نحو این توابع بسیار شبیه است. برای تابع enet، استفاده خواهد بود

*enetModel <- enet(x = as. matrix(solTrainXtrans)، y = solTrainY,*

*+ لامبدا = 0. 01، عادی‌سازی = درست)*

داده‌های پیش‌بینی باید یک شی ماتریسی باشند، بنابراین چارچوب داده solTrainXtrans باید برای تابع enet تبدیل شود. پیش‌بینی‌کننده‌ها باید ­قبل از مدل‌سازی متمرکز و مقیاس شوند. آرگومان ­normalize این استانداردسازی را به صورت خودکار انجام می‌دهد. پارامتر لامبدا جریمه رگرسیون پشته را کنترل می‌کند و با تنظیم این مقدار روی 0، متناسب با مدل ریج است. پنالتی ریج تا زمان پیش‌بینی نیازی به مشخص شدن ندارد:

*enetPred <- predict(enetModel, newx = as. matrix(solTestXtrans)*

*+ s = 0. 1، حالت = "کسری"،*

*+ نوع = "مناسب")*

*## یک لیست با چندین مورد برگردانده می‌شود:*

*نام‌ها (enetPred)*

"s" "کسری" "حالت" "مناسب"

*> ## مؤلفه* ' *fit* ' *مقادیر پیش‌بینی شده را دارد:*

*سر (enetPred$fit)*

20 21 23 25 28 31

-0. 60186178 -0. 42226814 -1. 20465564 -1. 23652963 -1. 25023517 -0. 05587631

برای تعیین اینکه کدام پیش‌بینی در مدل استفاده می‌شود، از روش پیش‌بینی با نوع = "ضرایب" استفاده می‌شود :

*enetCoef<- پیش‌بینی(enetModel، newx = as. matrix(solTestXtrans)،*

*+ s = 0. 1، حالت = "کسری"،*

*+ نوع = "ضرایب")*

*دم (enetCoef$ضرایب)*

|  |  |  |  |
| --- | --- | --- | --- |
| NumClorine | NumHalogen | NumRings | فاکتور هیدروفیل |
| 0. 00000000 | 0. 00000000 | 0. 00000000 | 0. 12678967 |
| سطح 1 | سطح 2 |  |  |
| 0. 09035596 | 0. 00000000 |  |  |

بیش از یک مقدار s را می‌توان با تابع پیش‌بینی برای تولید پیش‌بینی از بیش از یک مدل به‌طور همزمان استفاده کرد.

بسته‌های دیگر متناسب با مدل ریج یا برخی از نسخه‌های جایگزین مدل، biglars (برای مجموعه داده‌های بزرگ)، FLLat (برای ریج ذوب‌شده)، grplasso (کند گروهی)، جریمه‌شده، ریلاکسو (کند آرام) و موارد دیگر هستند. برای تنظیم مدل شبکه الاستیک با استفاده از train، روش = "enet" را مشخص می‌کنیم. در اینجا، ما مدل را روی مجموعه‌ای سفارشی از جریمه‌ها تنظیم می‌کنیم:

*enetGrid <- expand. grid(. lambda = c(0, 0. 01,. 1)*

*+ . fraction = seq(. 05، 1، طول = 20))*

*set. seed (100)*

*enetTune <- train(solTrainXtrans, solTrainY,*

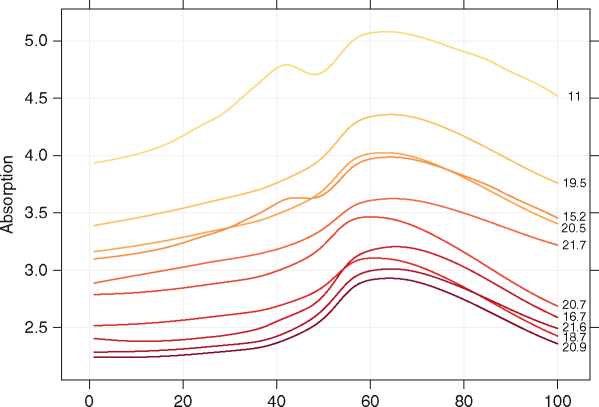
*+ روش = "enet"،*

*+ tuneGrid = enetGrid،*

*+ trControl = ctrl،*

*+ preProc = c ("مرکز"، "مقیاس"))*

شکل [6. 18](#bookmark316) را می‌توان از این شی با استفاده از plot(enetTune) ایجاد کرد.



شکل 6. 20: نمونه‌ای از ده طیف از داده‌های Tecator. رنگ منحنی‌ها مقادیر جذب را منعکس می‌کنند، جایی که *زرد* نشان دهنده جذب کم و *قرمز* نشان دهنده جذب بالا است.

تمرینات

فناوری طیف‌سنجی فروسرخ (IR) برای تعیین ترکیب شیمیایی ­یک ماده استفاده می‌شود. تئوری طیف‌سنجی IR معتقد است که ساختارهای مولکولی منحصربه‌فرد فرکانس‌های IR را متفاوت جذب می‌کنند. در عمل یک طیف ­اتر یک سری فرکانس IR را به یک ماده نمونه شلیک می‌کند و دستگاه جذب نمونه را در هر فرکانس جداگانه اندازه‌گیری می‌کند. این سری از اندازه‌گیری‌ها یک پروفایل طیفی ایجاد می‌کند که می‌توان از آن برای تعیین ترکیب شیمیایی ماده نمونه استفاده کرد.

ابزار تحلیل غذا و خوراک Tecator Infratec برای تحلیل 215 نمونه گوشت در 100 فرکانس استفاده شد. نمونه‌ای از این ­فایل‌های فرکانس حرفه‌ای در شکل 1 نمایش داده شده است.  [6. 20](#bookmark335) . علاوه بر نمایه IR، شیمی تجزیه درصد محتوای آب، چربی و پروتئین را برای هر نمونه تعیین کرد. اگر بتوانیم یک رابطه پیش‌بینی بین طیف IR و محتوای چربی برقرار کنیم، آن‌گاه دانشمندان مواد غذایی می‌توانند به جای استفاده از شیمی تحلیلی، محتوای چربی نمونه را با IR پیش‌بینی کنند. این امر باعث صرفه جویی در هزینه‌ها می‌شود، زیرا شیمی تجزیه فرآیندی گران تر و وقت گیر است:

R را راه اندازی کنید و از این دستورات برای بارگیری داده‌ها استفاده کنید:

*کتابخانه (کارت)*

*داده (تکاتور)*

*# برای مشاهده جزئیات بیشتر از ?tecator استفاده کنید*

جذب ماتریس حاوی 100 مقدار جذب برای 215 نمونه است، در حالی که نقاط پایانی ماتریس حاوی درصد رطوبت، چربی و پروتئین در ستون‌های 1-3 است.

در این مثال، پیش‌بینی‌کننده‌ها اندازه‌گیری در فرکانس‌های فردی هستند. از آنجایی که فرکانس‌ها در یک نظم نظام‌مند قرار دارند (850-1050 نانومتر)، پیش‌بینی‌کننده‌ها دارای درجه بالایی از همبستگی هستند. بنابراین، داده‌ها در ابعاد کوچکتری نسبت به تعداد کل پیش‌بینی‌کننده‌ها قرار دارند (215). از PCA برای تعیین ابعاد مؤثر این داده‌ها استفاده کنید. بعد موثر چیست؟

داده‌ها را به یک مجموعه آموزشی و آزمایشی تقسیم کنید، داده‌ها را از قبل پردازش کنید و هر مدل از مدل‌های توصیف شده در این فصل را بسازید. برای آن ­مدل‌هایی که پارامترهای تنظیم دارند، مقادیر بهینه پارامتر(های) تنظیم چیست؟

کدام مدل بهترین توانایی پیش‌بینی را دارد؟ آیا هر مدلی به‌طور قابل‌توجهی بهتر یا بدتر از بقیه است؟

توضیح دهید که از چه مدلی برای پیش‌بینی میزان چربی یک نمونه استفاده می‌کنید.

توسعه مدلی برای پیش‌بینی نفوذپذیری (به بخش [1. 4 )](#bookmark77) می تواند ­منابع قابل‌توجهی را برای یک شرکت داروسازی ذخیره کند و در عین حال مولکول‌هایی را که دارای نفوذپذیری کافی برای تبدیل شدن به دارو هستند، با سرعت بیشتری شناسایی کند:

R را راه اندازی کنید و از این دستورات برای بارگیری داده‌ها استفاده کنید:

*> کتابخانه (AppliedPredictiveModeling)*

*> داده (نفوذپذیری)*

اثر انگشت ماتریس حاوی 1107 پیش‌بینیی مولکولی دوتایی ­برای 165 ترکیب است، در حالی که نفوذپذیری حاوی پاسخ نفوذپذیری است.

پیش‌بینی‌کننده‌های اثر انگشت وجود یا عدم وجود ­زیرساخت‌های یک مولکول را نشان می‌دهند و اغلب پراکنده هستند به این معنی که تعداد نسبتاً کمی از مولکول‌ها حاوی هر زیرساخت هستند. با استفاده از تابع nearZeroVar از بسته caret، پیش‌بینی‌کننده‌هایی را که فرکانس پایین دارند، فیلتر کنید. چند پیش‌بینی برای مدل‌سازی باقی مانده است؟

داده‌ها را به یک مجموعه آموزشی و آزمایشی تقسیم کنید، داده‌ها را از قبل پردازش کنید و یک مدل PLS را تنظیم کنید. چه تعداد از متغیرهای پنهان بهینه هستند و تخمین نمونه‌گیری مجدد مربوطه از *R2* چیست ؟

پاسخ مجموعه تست را پیش‌بینی کنید. برآورد مجموعه آزمایشی *R* 2 چیست؟

سعی کنید مدل‌های دیگری را بسازید که در این فصل بحث شده است. آیا هیچ کدام عملکرد پیش‌بینی بهتری دارند؟

آیا هر یک از مدل‌های خود را برای جایگزینی آزمایش آزمایشگاهی نفوذپذیری توصیه می‌کنید؟

یک فرآیند تولید شیمیایی برای یک محصول دارویی در بخش مورد بحث قرار گرفت.  [1. 4](#bookmark77) . در این مسئله، هدف درک رابطه ­بین اندازه‌گیری‌های بیولوژیکی مواد خام (پیش‌بینی‌کننده‌ها)، اندازه‌گیری‌های فرآیند تولید (پیش‌بینی‌کننده‌ها) و پاسخ عملکرد محصول است. پیش‌بینی‌کننده‌های بیولوژیکی را نمی‌توان تغییر داد، اما می‌توان از آنها برای ارزیابی کیفیت مواد خام قبل از پردازش استفاده کرد. از سوی دیگر، پیش‌بینی‌کننده‌های فرآیند تولید را می‌توان در فرآیند تولید تغییر داد. بهبود بازده محصول به میزان 1 درصد باعث افزایش درآمد تقریباً صد هزار دلاری در هر دسته می‌شود:

R را راه اندازی کنید و از این دستورات برای بارگیری داده‌ها استفاده کنید:

*> کتابخانه (AppliedPredictiveModeling)*

*> داده‌ها (تولید شیمیایی)*

ماتریس processPredictors شامل 57 پیش‌بینی (12 پیش‌بینی مواد بیولوژیکی ورودی و 45 پیش‌بینی فرآیند توصیف کننده) برای 176 دوره تولید است. بازده شامل درصد بازدهی برای هر اجرا است.

درصد کمی از سلول‌ها در مجموعه پیش‌بینی حاوی مقادیر گمشده هستند. برای پر کردن این مقادیر از دست رفته از یک تابع imputation استفاده کنید (مثلاً به بخش مراجعه کنید.  [3. 8 )](#bookmark4) .

داده‌ها را به یک مجموعه آموزشی و آزمایشی تقسیم کنید، داده‌ها را از قبل پردازش کنید و مدلی را که انتخاب می‌کنید از این فصل تنظیم کنید. مقدار بهینه معیار عملکرد چقدر است؟

پاسخ مجموعه تست را پیش‌بینی کنید. ارزش معیار عملکرد چیست و چگونه این معیار با معیار عملکرد نمونه‌گیری مجدد در مجموعه آموزشی مقایسه می‌شود؟

کدام پیش‌بینی‌کننده‌ها در مدلی که آموزش داده‌اید مهم‌تر هستند؟ آیا پیش‌بینی‌کننده‌های بیولوژیکی یا فرآیندی بر فهرست غالب هستند؟

روابط بین هر یک از پیش‌بینی‌کننده‌های برتر و پاسخ را بررسی ­کنید. چگونه این اطلاعات می‌تواند در بهبود عملکرد در مراحل آتی فرآیند تولید مفید باشد؟

فصل 7

مدل‌های رگرسیون غیرخطی

فصل قبل مدل‌های رگرسیونی را که ذاتاً خطی بودند مورد بحث قرار داد. بسیاری از این مدل‌ها را می‌توان با افزودن دستی عبارات مدل (مثلاً عبارت‌های مربع) با روندهای غیرخطی در داده‌ها سازگار کرد. با این حال، برای انجام این کار، باید ماهیت خاص غیرخطی بودن در داده‌ها را دانست.

مدل‌های رگرسیونی متعددی وجود دارد که ذاتاً غیرخطی هستند. هنگام استفاده از این مدل‌ها، نیازی نیست که شکل دقیق غیرخطی بودن به‌طور صریح یا مشخص قبل از آموزش مدل مشخص شود. این فصل به چندین مدل نگاه می‌کند: شبکه‌های عصبی، خطوط رگرسیون تطبیقی چند متغیره (MARS)، ماشین‌های بردار پشتیبان (SVM) و نزدیک‌ترین همسایگان K ( *K NN* ). مدل‌های مبتنی بر درخت نیز غیرخطی هستند. با توجه به محبوبیت و کاربرد آنها در مدل‌های مجموعه، فصل بعدی را به آن روش‌ها اختصاص داده ایم.

شبکه‌های عصبی

شبکه‌های عصبی [( بیشاپ 1995](#bookmark1011) ; [ریپلی 1996](#bookmark1024) ; [Titterington 2010](#bookmark1026) ) ­تکنیک‌های رگرسیون غیرخطی قدرتمندی هستند که از تئوری‌هایی در مورد نحوه عملکرد مغز الهام گرفته شده اند. مانند حداقل مربعات جزئی، نتیجه توسط یک ­مجموعه واسطه‌ای از متغیرهای مشاهده نشده (که در اینجا *متغیرهای پنهان* یا *واحدهای پنهان نامیده می‌*شوند) مدل می‌شود.

این واحدهای پنهان ترکیبی خطی از پیش‌بینی‌کننده‌های اصلی هستند، اما بر خلاف مدل‌های PLS، به صورت سلسله مراتبی تخمین زده نمی‌شوند (شکل 2).  [7. 1 )](#bookmark343) .

همانطور که قبلاً گفته شد، هر واحد پنهان ترکیبی خطی از برخی یا همه متغیرهای پیش‌بینی است. با این حال، این ترکیب خطی معمولاً توسط یک تابع غیرخطی *g (* *•* )، مانند تابع لجستیک (یعنی سیگموئیدی) تبدیل می‌شود:

*h k (* x ) = *g*  0 *k* + ^ *x j/ S jk^*

where

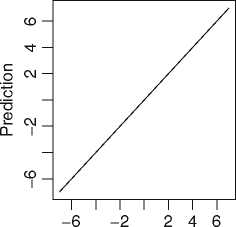
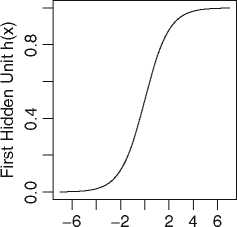
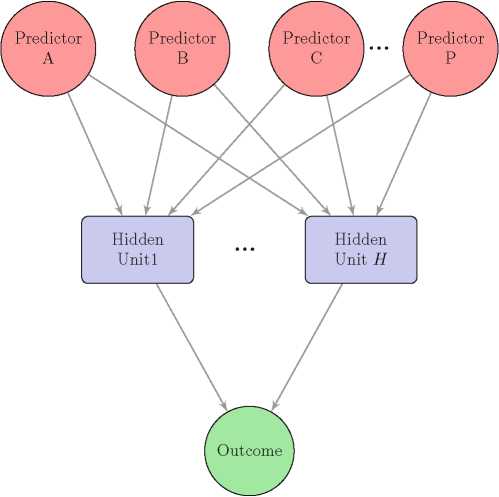
*1 1*  1

*g (* *u* ) = *.*

1+ *e - u*

\_7،

©



Pi + Pllxl + P21x2

Fig. 7.1: A diagram of a neural network with a single hidden layer. The hidden units are linear combinations of the predictors that have been transformed by a sigmoidal function. The output is modeled by a linear combination of the hidden units

Yo + Yihi(x) + Y2h2(x) + Y3h3(x)

*ft* شبیه به ضرایب رگرسیون هستند. ضریب *ft jk* اثر پیش‌بینی *j بر k* امین واحد پنهان است. یک مدل شبکه عصبی معمولاً شامل چندین واحد پنهان برای مدل‌سازی نتیجه است. توجه داشته باشید که برخلاف ترکیب‌های خطی در PLS، هیچ محدودیتی وجود ندارد که به تعریف این ترکیب‌های خطی کمک کند. به همین دلیل، احتمال کمی وجود دارد که ضرایب در هر واحد، بخشی از اطلاعات منسجم را نشان دهد.

هنگامی که تعداد واحدهای پنهان تعریف شد، هر واحد باید با نتیجه مرتبط باشد. ترکیب خطی دیگری واحدهای پنهان را به نتیجه متصل می‌کند:

*H   
f ( x ) = Y 0* + *^Y k h k.   
k* = 1

برای این نوع مدل شبکه و پیش‌بینی‌کننده‌های P، مجموع پارامترهای کل H ( P + 1) + H + 1 برآورد می‌شود که با افزایش P به سرعت بزرگ می‌شود. برای داده‌های حلالیت، به یاد بیاورید که 228 پیش‌بینی وجود دارد. یک مدل شبکه عصبی با سه واحد پنهان 691 پارامتر را برآورد می‌کند در حالی که مدلی با پنج واحد پنهان دارای 1151 ضریب است.

با در نظر گرفتن این مدل به‌عنوان یک مدل رگرسیون غیرخطی، پارامترها معمولاً برای به حداقل رساندن مجموع مجذور باقیمانده بهینه می‌شوند. این می‌تواند یک مسئله بهینه‌سازی عددی چالش برانگیز باشد (به یاد داشته باشید که هیچ محدودیتی ­در پارامترهای این مدل پیچیده غیرخطی وجود ندارد). پارامترها معمولاً به مقادیر تصادفی مقداردهی اولیه می‌شوند و سپس از الگوریتم‌های تخصصی برای حل [معادلات](#bookmark1025) استفاده می‌شود. الگوریتم پس انتشار ( Rumelhart et al. [1986](#bookmark1025) ) یک روش بسیار کارآمد است که با مشتقات برای یافتن پارامترهای بهینه کار می‌کند. با این حال، معمول است که یک راه حل برای این معادله یک راه حل کلی نیست، به این معنی که ما نمی‌توانیم تضمین کنیم که مجموعه پارامترهای حاصل به‌طور یکنواخت بهتر از هر مجموعه دیگری است.

همچنین، شبکه‌های عصبی به دلیل تعداد زیاد ­ضرایب رگرسیون، تمایل زیادی به برازش بیش از حد رابطه بین پیش‌بینی‌کننده‌ها و پاسخ دارند. برای مبارزه با این موضوع، چندین رویکرد مختلف پیشنهاد شده است. اول، الگوریتم‌های تکراری برای حل معادلات رگرسیون را می‌توان پیش از موعد متوقف کرد [( Wang and Venkatesh 1984](#bookmark1027) ). این رویکرد به‌عنوان توقف زودهنگام نامیده می‌شود و هنگامی که برخی از برآوردهای نرخ خطا شروع به افزایش می‌کند (به جای مقداری تحمل عددی برای نشان دادن پایدار بودن تخمین پارامترها یا نرخ خطا) روند بهینه‌سازی را متوقف می‌کند. ­با این حال، مسائل آشکاری در این روش وجود دارد. ابتدا، چگونه خطای مدل را برآورد کنیم؟ نرخ خطای ظاهری می‌تواند بسیار خوش بینانه باشد (همانطور که در بخش بحث شد.  [4. 1 )](#bookmark195) و تقسیم بیشتر مجموعه آموزشی می‌تواند مسأله ساز باشد. همچنین، از آنجایی که نرخ خطای اندازه‌گیری شده مقداری عدم قطعیت مرتبط با آن دارد، چگونه می‌توانیم بگوییم که آیا واقعاً در حال افزایش است؟

رویکرد دیگر برای تعدیل برازش بیش از حد، استفاده از کاهش وزن است، ­روشی برای تنظیم کردن مدل مشابه با رگرسیون برآمدگی که در فصل گذشته مورد بحث قرار گرفت. در اینجا، ما یک جریمه برای ضرایب رگرسیون بزرگ اضافه می‌کنیم، به‌طوری که هر مقدار بزرگ باید تأثیر قابل‌توجهی بر خطاهای مدل داشته باشد. به‌طور رسمی، بهینه‌سازی تولید شده سعی می‌کند یک نسخه جایگزین از مجموع مربعات خطاها را به حداقل برساند:

n HP H

y i - f i ( x + XY^ j + ^Y k

i = 1 k = 1 j = 0 k = 0 برای مقدار معین A. با افزایش مقدار منظم‌سازی، مدل برازش شده صاف‌تر می‌شود و کمتر احتمال دارد که بیش از حد با مجموعه آموزشی برازش کند. البته مقدار این پارامتر باید مشخص باشد و به همراه تعداد واحدهای پنهان، پارامتر تنظیمی برای مدل باشد. مقادیر معقول A بین 0 تا 0. 1 است. همچنین توجه داشته باشید که از آنجایی که ضرایب رگرسیون در حال جمع شدن هستند، باید در یک مقیاس باشند. بنابراین پیش‌بینی‌کننده‌ها باید قبل از مدل‌سازی متمرکز و مقیاس شوند.

ساختار مدلی که در اینجا توضیح داده شده است، ساده‌ترین معماری شبکه عصبی است: یک شبکه پیشخور تک لایه. انواع دیگری نیز وجود دارد، مانند مدل‌هایی که بیش از یک لایه واحد پنهان وجود دارد (یعنی لایه‌ای از واحدهای پنهان وجود دارد که واحدهای پنهان دیگر را مدل می‌کند). همچنین، دیگر معماری‌های مدل دارای حلقه‌هایی هستند که در هر دو جهت بین لایه‌ها حرکت می‌کنند. دست اندرکاران این مدل‌ها همچنین ممکن است برای بهینه‌سازی بیشتر مدل، اتصالات خاص بین اشیاء را حذف کنند. همچنین چندین رویکرد بیزی برای شبکه‌های عصبی وجود داشته است [( نیل 1996](#bookmark1022) ). چارچوب بیزی که در [نیل ( 1996](#bookmark1022) ) برای این مدل‌ها به‌طور خودکار منظم‌سازی و انتخاب خودکار ویژگی را در بر می‌گیرد. این رویکرد به شبکه‌های عصبی بسیار قدرتمند است، اما جنبه‌های محاسباتی مدل حتی قدرتمندتر می‌شوند. مدلی که بسیار شبیه به شبکه‌های عصبی است، نقشه‌های خودسازماندهی هستند [( Kohonen 1995](#bookmark1020) ). این مدل را می‌توان به‌عنوان یک تکنیک اکتشافی بدون نظارت یا به روشی تحت نظارت برای پیش‌بینی استفاده کرد [( ملسسن و همکاران 2006](#bookmark1022) ).

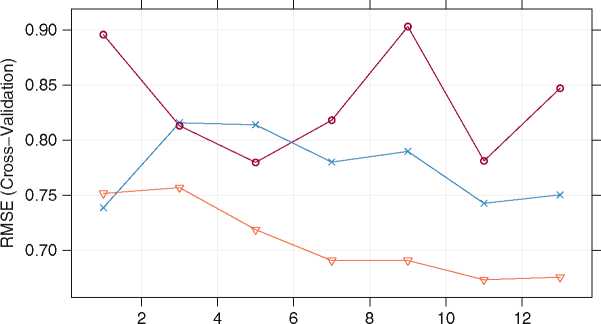
با توجه به چالش تخمین تعداد زیادی از پارامترها، ­مدل برازش داده تخمین‌های پارامتری را پیدا می‌کند که به صورت محلی بهینه هستند. یعنی الگوریتم همگرا می‌شود، اما تخمین پارامترهای حاصل بعید است که برآوردهای بهینه جهانی باشند. اغلب، راه‌حل‌های بهینه محلی مختلف ­می‌توانند مدل‌هایی تولید کنند که بسیار متفاوت هستند اما عملکرد تقریباً برابری دارند. این بی ثباتی مدل گاهی اوقات می‌تواند مانع این مدل شود. به‌عنوان یک جایگزین، چندین مدل را می‌توان با استفاده از مقادیر شروع مختلف و میانگین‌گیری نتایج این مدل برای تولید پیش‌بینی پایدارتر ایجاد کرد [( پرون و کوپر 1993](#bookmark1023) ; [ریپلی 1995](#bookmark1024) ; [Tumer and Ghosh 1996](#bookmark1026) ). چنین *میانگین‌گیری مدل* اغلب تأثیر مثبت قابل‌توجهی بر شبکه‌های عصبی دارد.

این مدل‌ها اغلب تحت تأثیر همبستگی بالا بین متغیرهای پیش‌بینی قرار می‌گیرند (زیرا از گرادیان‌ها برای بهینه‌سازی پارامترهای مدل استفاده می‌کنند ­). دو رویکرد برای کاهش این موضوع، فیلتر کردن پیش‌بینی‌کننده‌ها برای حذف پیش‌بینی‌کننده‌هایی است که با همبستگی‌های بالا مرتبط هستند. روش ­دیگر استخراج ویژگی، مانند تحلیل مؤلفه‌های اصلی، می‌تواند قبل از مدل‌سازی برای حذف همبستگی‌ها مورد استفاده قرار گیرد. یک اثر جانبی مثبت ­هر دو این رویکرد این است که اصطلاحات مدل کمتری باید بهینه شوند، بنابراین زمان محاسبات بهبود می‌یابد.

برای داده‌های حلالیت، از شبکه‌های عصبی میانگین مدل استفاده شد. سه ارزش پوسیدگی وزنی مختلف ( *A* = 0. 00،0. 01،0. 10 *) همراه با* یک لایه مخفی منفرد با اندازه‌های بین 1 تا 13 واحد پنهان مورد ارزیابی قرار گرفت. پیش‌بینی‌کننده‌های نهایی، میانگین‌های پنج شبکه عصبی مختلف است که با استفاده از مقادیر پارامترهای اولیه متفاوت ایجاد شده‌اند. پروفایل‌های RMSE تایید شده متقابل این مدل‌ها در شکل 1 نمایش داده شده است.  [7. 2](#bookmark348) . افزایش میزان پوسیدگی وزن

کاهش وزن

0. 01 x -

عملکرد مدل به وضوح بهبود یافته است، در حالی که واحدهای پنهان بیشتر نیز خطای مدل را کاهش می‌دهند. مدل بهینه از 11 واحد پنهان با مجموع 2531 ضریب استفاده کرد. عملکرد مدل برای درجه بالایی از منظم‌سازی نسبتاً پایدار است (یعنی *A* = 0. 1 *)،* بنابراین مدل‌های کوچکتر نیز می‌توانند برای این داده‌ها مؤثر باشند.

0 o

0.1 v -

#Hidden Units

Fig. 7.2: RMSE profiles for the neural network model. The optimal model used *A* = 0*.*1 and 11 hidden units

**Splines رگرسیون تطبیقی چند متغیره**

مانند شبکه‌های عصبی و حداقل مربعات جزئی، MARS [( فریدمن 1991](#bookmark1016) ) از ویژگی‌های جایگزین به جای پیش‌بینی‌کننده‌های اصلی استفاده می‌کند. با این حال، در حالی که PLS و شبکه‌های عصبی مبتنی بر ترکیب‌های خطی پیش‌بینی‌کننده‌ها هستند، MARS دو نسخه متضاد یک پیش‌بینی را برای ورود به مدل ایجاد می‌کند. همچنین، ویژگی‌های جایگزین در MARS معمولاً تابعی از یک یا دو پیش‌بینی در هر زمان هستند. ماهیت ویژگی‌های MARS، پیش‌بینی را به دو گروه تقسیم می‌کند و روابط خطی بین پیش‌بینی و نتیجه را در هر گروه مدل می‌کند. به‌طور خاص، با توجه به یک نقطه برش برای یک پیش‌بینی، دو ویژگی جدید عملکردهای "لولا" یا "چوب هاکی" اصلی هستند (شکل 2 را ببینید.  [7. 3 )](#bookmark350) . ویژگی "چپ دست" دارای مقادیر صفر بزرگتر از نقطه برش است، در حالی که ویژگی دوم صفر کمتر از نقطه برش است. ویژگی‌های جدید به یک مدل رگرسیون خطی پایه برای تخمین شیب‌ها و بریدگی‌ها اضافه می‌شوند. در واقع، این طرح یک *مدل خطی تکه‌ای ایجاد* می‌کند که در آن هر ویژگی جدید، بخش جدا شده‌ای از داده‌های اصلی را مدل می‌کند.

نقطه برش چگونه تعیین شد؟ هر نقطه داده برای هر پیش‌بینی با ایجاد یک مدل رگرسیون خطی با ویژگی‌های کاندید به‌عنوان یک نقطه برش نامزد ارزیابی می‌شود و خطای مدل مربوطه محاسبه می‌شود. سپس ترکیب پیش‌بینی/برش نقطه‌ای که به کوچک‌ترین خطا می‌رسد برای مدل استفاده می‌شود. ماهیت تبدیل پیش‌بینی، چنین تعداد زیادی از رگرسیون‌های خطی را از نظر محاسباتی امکان‌پذیر می‌سازد. در برخی از پیاده‌سازی‌های MARS، از جمله مورد استفاده شده در اینجا، کاربرد اصطلاحات خطی ساده برای هر پیش‌بینی (یعنی بدون تابع لولا) نیز ارزیابی می‌شود.

پس از ایجاد مدل اولیه با دو ویژگی اول، مدل جستجوی جامع دیگری را برای یافتن مجموعه بعدی از ویژگی‌ها انجام می‌دهد که با توجه به مجموعه اولیه، بهترین برازش مدل را به دست می‌دهد. این روند تا رسیدن به یک نقطه توقف (که توسط کاربر قابل تنظیم است) ادامه می‌یابد.

در جستجوی اولیه برای ویژگی‌های داده‌های حلالیت، نقطه برش 5. 9 برای وزن مولکولی کمترین میزان خطا را داشت. پیش‌بینیی‌های مصنوعی حاصل ­در دو پانل بالای شکل نشان داده شده‌اند.  [7. 3](#bookmark350) . یک پیش‌بینی همه مقادیر کمتر از نقطه برش را روی صفر دارد و مقادیر بزرگتر از نقطه برش بدون تغییر باقی می‌ماند. ویژگی دوم، تصویر آینه اولی است. به جای داده‌های اصلی، از این دو پیش‌بینی جدید برای پیش‌بینی نتیجه در مدل رگرسیون خطی استفاده می‌شود. پانل پایین شکل.  [7. 3](#bookmark350) نتیجه رگرسیون خطی با دو ویژگی جدید و ماهیت تکه‌ای رابطه را نشان می‌دهد. هنگامی که وزن مولکولی کمتر از 5. 9 باشد ویژگی "چپ دست" با شیب منفی همراه است در حالی که ویژگی "راست" شیب مثبت را برای مقادیر بزرگتر پیش‌بینی تخمین می‌زند.

از نظر ریاضی، تابع لولا برای ویژگی‌های جدید می‌تواند به صورت نوشته شود

*xx>* 0

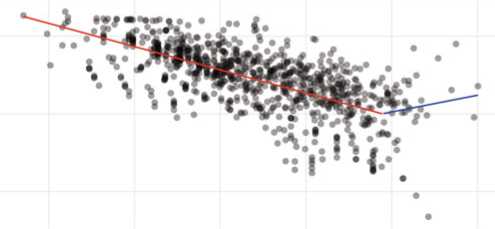
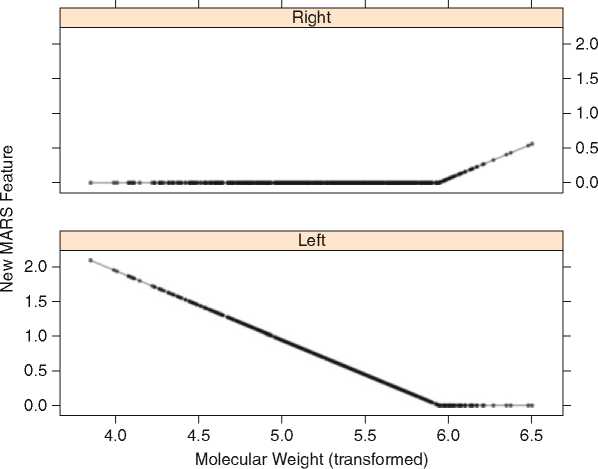
0 *x < 0 (7J)*

یک جفت تابع لولا معمولاً به صورت *h (* *x-a* ) و *h (* *a-x* ) نوشته می‌شود. اولی وقتی *0> a* است غیر صفر است، در حالی که وقتی *0<a است دومی غیر صفر است.* توجه داشته باشید که وقتی این درست است مقدار تابع در واقع *-0* است. برای مدل MARS نشان داده شده در شکل.  [7. 3](#bookmark350) ، معادله مدل واقعی خواهد بود

*-* 5 + 2. 1 *xh (* *MolWeight - 5. 94516 )* + 3 *xh (5. 94516* - *MolWeight* ).

جمله دوم در این معادله با ویژگی سمت راست نشان داده شده در شکل 1 مرتبط است.  [7. 3](#bookmark350) در حالی که آخرین جزء معادله ویژگی سمت چپ است. خط رگرسیون زیر برش با وجود ضریب مثبت برای آخرین ویژگی در حال کاهش است.

جدول [7. 1](#bookmark351) چند مرحله اول مرحله تولید ویژگی (قبل از هرس) را نشان می‌دهد. ویژگی‌ها از بالا به پایین وارد مدل رگرسیون خطی شدند. در اینجا توصیفگر اثر انگشت باینری مدل را به‌عنوان یک عبارت خطی ساده وارد می‌کند (تقسیم یک متغیر باینری بی معنی خواهد بود). ستون اعتبار متقاطع تعمیم یافته (GCV) RMSE تخمین زده شده را نشان می‌دهد



co

0-

•5-

I

■10-

4.0

5.0

6.0

6.5

4.5

5.5

وزن مولکولی (تبدیل شده)

شکل 7. 3: نمونه‌ای از ویژگی‌های استفاده شده توسط MARS برای داده‌های حلالیت. پس از یافتن نقطه برش 5. 9 برای وزن مولکولی، دو ویژگی جدید ایجاد شده و در مدل رگرسیون خطی استفاده می‌شود. دو پانل *بالا* رابطه بین پیش‌بینی اصلی و دو ویژگی حاصل را نشان می‌دهد. پانل *پایینی* رابطه پیش‌بینی شده را هنگام استفاده از این دو ویژگی در مدل رگرسیون خطی نشان می‌دهد. خط *قرمز* سهم تابع لولای "چپ دست" را نشان می‌دهد در حالی که *خط آبی* با ویژگی دیگر مرتبط است.

جدول 7. 1: نتایج چندین تکرار الگوریتم MARS قبل از هرس

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| پیش‌بینی | تایپ کنید | قطع کردن | ضریب RMSE | |
| عرض از مبدأ |  |  | 4. 193 | *-* 9. 33 |
| مولی وزن | درست | 5. 95 | 2. 351 | *-* 3. 23 |
| مولی وزن | ترک کرد | 5. 95 | 1. 148 | 0. 66 |
| سطح 1 | درست | 1. 96 | 0. 935 | 0. 19 |
| سطح 1 | ترک کرد | 1. 96 | 0. 861 | *-* 0 *\_* 66 |
| NumNonHAtoms | درست | 3. 00 | 0. 803 | *-* 7. 51 |
| NumNonHAtoms | ترک کرد | 3. 00 | 0. 761 | 8. 53 |
| FP137 | خطی |  | 0. 727 | 1. 24 |
| NumOxygen | درست | 1. 39 | 0. 701 | 2. 22 |
| NumOxygen | ترک کرد | 1. 39 | 0. 683 | *-* 0 *\_* 43 |
| NumNonHBonds | درست | 2. 58 | 0. 670 | 2. 21 |
| NumNonHBonds | ترک کرد | 2. 58 | 0. 662 | *-* 3. 29 |

ریشه میانگین مربعات خطا با استفاده از آماره GCV که مدل حاوی عبارت‌های ردیف فعلی و تمام ردیف‌های بالا است، برآورد شد. قبل از هرس، هر جفت عملکرد لولا علیرغم کاهش جزئی در RMSE تخمین زده شده در مدل نگهداری می‌شود.

هنگامی که مجموعه کاملی از ویژگی‌ها ایجاد شد، الگوریتم به‌طور متوالی ویژگی‌های فردی را که به‌طور قابل‌توجهی به معادله مدل کمک نمی‌کنند، حذف می‌کند. این روش «هرس» هر متغیر پیش‌بینی را ارزیابی می‌کند و ­با گنجاندن آن در مدل میزان خطا را تخمین می‌زند. این فرآیند در مسیری که ویژگی‌ها اضافه شده‌اند به عقب پیش نمی‌رود. برخی از ویژگی‌هایی که در ابتدای فرآیند مهم تلقی می‌شوند ممکن است حذف شوند در حالی که ویژگی‌های اضافه شده در پایان ممکن است حفظ شوند. برای تعیین سهم هر ویژگی در مدل، از آماره *GCV استفاده می‌شود.* این مقدار یک میانبر محاسباتی برای مدل‌های رگرسیون خطی است که یک مقدار خطایی تولید می‌کند که اعتبارسنجی متقاطع ترک یک خروجی را تقریبی می‌کند [( Golub et al. 1979](#bookmark1017) ). GCV برآوردهای بهتری نسبت به میزان خطای ظاهری برای تعیین اهمیت هر ویژگی در مدل ایجاد می‌کند. تعداد ­عبارت‌هایی که باید حذف شوند را می‌توان به صورت دستی تنظیم کرد یا به‌عنوان یک پارامتر تنظیم در نظر گرفت و با استفاده از شکل دیگری از نمونه‌گیری مجدد تعیین کرد.

فرآیند بالا توصیفی از یک مدل MARS افزایشی است که در آن هر ویژگی جایگزین شامل یک پیش‌بینی است. با این حال، MARS می‌تواند مدل‌هایی بسازد که در آن ویژگی‌ها شامل چندین پیش‌بینی به‌طور همزمان باشد. با یک مدل *درجه دوم* MARS، الگوریتم جستجوی یکسانی را برای یک عبارت انجام می‌دهد که مدل را بهبود می‌بخشد و پس از ایجاد جفت ویژگی‌های اولیه، جستجوی دیگری را برای ایجاد برش‌های جدید برای جفت شدن با هر یک از ویژگی‌های اصلی آغاز می‌کند. فرض کنید جفت تابع لولا به صورت *A* و *B نشان داده شده* است.

روش جستجو تلاش می‌کند تا توابع لولای *C* و *D را پیدا کند که وقتی در A* ضرب می‌شوند، منجر به بهبود مدل می‌شود. به عبارت دیگر، مدل دارای عباراتی برای *A , A x B* و *A x C خواهد بود.* همین رویه برای ویژگی *B نیز رخ خواهد* داد. توجه داشته باشید که اگر مدل با جمع آنها بهبود نیابد، الگوریتم اصطلاحات اضافی را اضافه نخواهد کرد. همچنین، روش هرس ممکن است شرایط اضافی را حذف کند. برای مدل‌های MARS که می‌توانند شامل دو یا چند عبارت در یک زمان باشند، ناپایداری‌های گاه به گاه را در پیش‌بینی‌کننده‌های مدل مشاهده کرده‌ایم که در آن چند پیش‌بینی نمونه بسیار نادرست هستند (شاید یک مرتبه بزرگی از مقدار واقعی). این مسأله در مدل‌های افزودنی MARS مشاهده نشده است.

به‌طور خلاصه، دو پارامتر تنظیم مرتبط با مدل MARS وجود دارد: درجه ویژگی‌هایی که به مدل اضافه می‌شوند و تعداد عبارت‌های حفظ شده. پارامتر دوم را می‌توان به‌طور خودکار ­با روش هرس پیش فرض (با استفاده از GCV)، تعیین شده توسط کاربر یا با استفاده از روش نمونه‌گیری مجدد خارجی تعیین کرد. برای تحلیل داده‌های حلالیت، از اعتبارسنجی متقاطع 10 برابری برای توصیف عملکرد مدل نسبت ­به مدل‌های مرتبه اول و دوم و 37 مقدار برای تعداد اصطلاحات مدل استفاده کردیم که ­از 2 تا 38 متغیر است. نمایه عملکرد حاصل نشان داده شده است. در شکل [7. 4](#bookmark353) . به نظر می‌رسد تفاوت بسیار کمی در مدل‌های درجه اول و دوم از نظر RMSE وجود دارد.

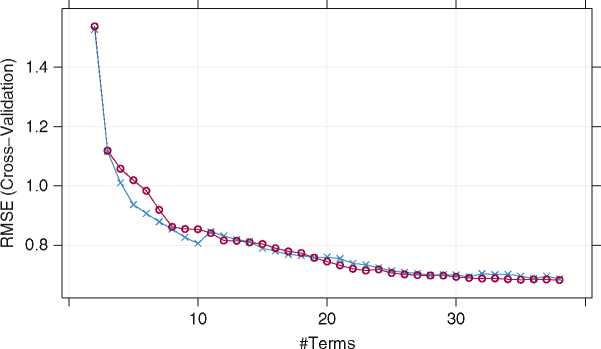
روش اعتبارسنجی متقاطع یک مدل درجه دوم با 38 عبارت را انتخاب کرد. با این حال، از آنجایی که پروفیل‌های مدل مرتبه اول و دوم تقریباً یکسان هستند، مدل مرتبه اول خردمندتر به‌عنوان مدل نهایی انتخاب شد. این مدل از 38 عبارت استفاده می‌کرد اما تابعی از 30 پیش‌بینی (از 228 مورد ممکن) بود.

اعتبارسنجی متقاطع RMSE را 0. 7 واحد ورود و R2 تخمین *زده* است 0. 887 باشد. به یاد بیاورید که روش MARS به‌طور داخلی از GCV برای تخمین عملکرد مدل استفاده می‌کند. با استفاده از GCV، RMSE 0. 4 واحد ورود و *R2* برآورد شد. از 0. 908. با استفاده از مجموعه آزمایشی 316 نمونه، RMSE 0. 7 با *R2 مربوطه* تعیین شد. از 0. 879. واضح است که تخمین‌های GCV دلگرم‌کننده‌تر از برآوردهای به‌دست‌آمده از روش اعتبارسنجی متقابل یا مجموعه آزمایشی هستند. با این حال، توجه داشته باشید که تخمین GCV داخلی که MARS از آن استفاده می‌کند، یک مدل فردی را ارزیابی می‌کند در حالی که ­روش اعتبارسنجی متقاطع خارجی در معرض تغییرات در کل فرآیند ساخت مدل، از جمله انتخاب ویژگی است. از آنجایی که تخمین GCV عدم قطعیت انتخاب ویژگی را منعکس نمی‌کند، از *بایاس انتخاب* رنج می‌برد ( Ambroise and McLachlan 2002 ). این پدیده در فصل بیشتر مورد بحث قرار خواهد گرفت.  [19](#bookmark864) .

استفاده از MARS چندین مزیت دارد. اول، مدل به‌طور خودکار انتخاب ویژگی را انجام می‌دهد. معادله مدل مستقل از ­متغیرهای پیش تعیین کننده است که با هیچ یک از ویژگی‌های مدل نهایی درگیر نیستند. این نکته را نمی‌توان دست کم گرفت. با توجه به تعداد زیادی پیش‌بینی که در بسیاری از حوزه‌های مسأله دیده می‌شود، MARS به‌طور بالقوه مجموعه پیش‌بینی را با استفاده از همان الگوریتمی که مدل را می‌سازد، نازک می‌کند. به این ترتیب، روال انتخاب ویژگی ­ارتباط مستقیمی با عملکرد عملکردی دارد. مزیت دوم تفسیرپذیری است. هر ویژگی لولا مسئول مدل‌سازی خاصی است

درجه محصول

1 o 2 X



شکل 7. 4: پروفایل‌های RMSE برای مدل MARS. روش اعتبارسنجی متقاطع یک مدل درجه دوم با 38 عبارت انتخاب کرد، اگرچه تفاوت کمی بین مدل‌های درجه اول و دوم وجود دارد. با توجه به این هم ارزی، مدل مرتبه اول ساده تر به‌عنوان مدل نهایی انتخاب شد

منطقه در فضای پیش‌بینی با استفاده از یک مدل خطی (تکه ای). وقتی مدل MARS افزودنی است، سهم هر پیش‌بینی را می‌توان بدون نیاز به در نظر گرفتن دیگران جدا کرد. این می‌تواند برای ارائه تفسیرهای واضح از نحوه ارتباط هر پیش‌بینی با نتیجه استفاده شود. برای مدل‌های غیرافزودنی، قدرت تفسیری مدل کاهش نمی‌یابد. یک ویژگی درجه دوم را در نظر بگیرید که شامل دو پیش‌بینی است. از آنجایی که هر تابع لولا به دو ناحیه تقسیم می‌شود، سه ناحیه از چهار ناحیه ممکن صفر خواهند بود و هیچ کمکی به مدل ارائه نمی‌دهند. به همین دلیل، تأثیر این دو عامل را می‌توان بیشتر جدا کرد و تفسیر را به سادگی مدل افزودنی می‌کند. در نهایت، مدل MARS به پیش پردازش بسیار کمی از داده‌ها نیاز دارد. تبدیل داده‌ها و فیلتر کردن پیش‌بینی‌کننده‌ها مورد نیاز نیست. به‌عنوان مثال، یک پیش‌بینی واریانس صفر هرگز برای یک تقسیم انتخاب نمی‌شود، زیرا هیچ اطلاعات پیش‌بینی ممکنی ارائه نمی‌دهد. پیش‌بینی‌کننده‌های مرتبط به شدت بر عملکرد مدل تأثیر نمی‌گذارند، اما می‌توانند تفسیر مدل را پیچیده کنند. فرض کنید مجموعه آموزشی حاوی دو پیش‌بینی بود که تقریباً کاملاً همبسته بودند. از آنجایی که MARS می‌تواند یک پیش‌بینی را بیش از یک بار در طول تکرارها انتخاب کند، انتخاب پیش‌بینی مورد استفاده در ویژگی اساساً تصادفی است. در این مورد، تفسیر مدل توسط دو ­قطعه اطلاعات اضافی که در قسمت‌های مختلف مدل با نام‌های مختلف نشان داده می‌شوند، مختل می‌شود.

روش دیگری برای کمک به درک ماهیت چگونگی تأثیر پیش‌بینی‌کننده‌ها بر مدل، کمی کردن *اهمیت آنها* برای مدل است. برای مریخ، یک فناوری­

نکته اصلی برای انجام این کار، ردیابی کاهش در ریشه میانگین مربعات خطا (که با استفاده از آماره GCV اندازه‌گیری می‌شود) است که هنگام افزودن یک ویژگی خاص به مدل رخ می‌دهد. این کاهش به پیش‌بینی(های) اصلی مرتبط با ویژگی نسبت داده می‌شود. این پیشرفت‌ها در مدل را می‌توان ­برای هر پیش‌بینی به‌عنوان معیاری نسبی از تأثیر روی مدل جمع‌آوری کرد. همانطور که در جدول مشاهده می‌شود [7. 1 ،](#bookmark351) پس از اضافه شدن دو ویژگی وزن مولکولی به مدل، RMSE از 4. 19 به 1. 15 (کاهش 3. 04) کاهش یافته است. پس از این، افزودن عبارت برای اولین پیش‌بینی مساحت سطح، er ­rr را 0. 29 کاهش می‌دهد. با توجه به این اعداد، به نظر می‌رسد که پیش‌بینی وزن مولکولی برای مدل مهمتر از اولین پیش‌بینی سطح سطح است. این فرآیند برای هر پیش‌بینی استفاده شده در مدل تکرار می‌شود. پیش‌بینی‌کننده‌هایی که در هیچ ویژگی استفاده نشده اند دارای اهمیت صفر هستند. برای مدل حلالیت، به نظر می‌رسد پیش‌بینی‌کننده‌های MolWeight، NumNonHAtoms و SurfaceArea2 بیشترین تأثیر را بر مدل MARS دارند (برای جزئیات بیشتر به بخش محاسبات در پایان فصل مراجعه کنید).

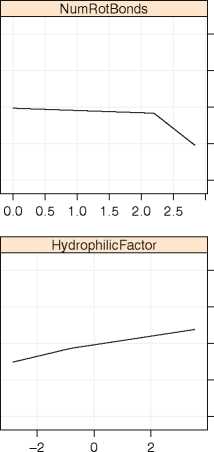
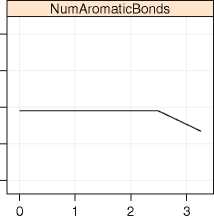
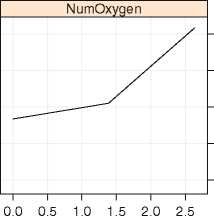
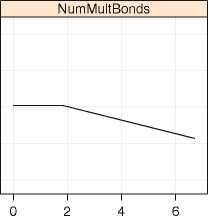
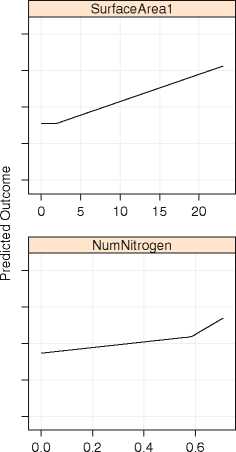
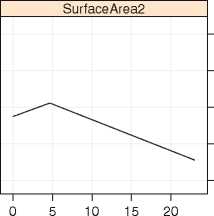
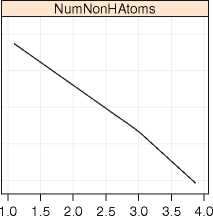
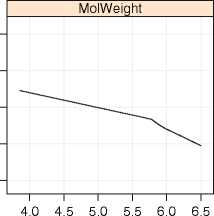
شکل [7. 5](#bookmark357) تفسیرپذیری مدل MARS افزایشی را با پیش‌بینی‌کننده‌های پیوسته نشان می‌دهد. برای هر پانل، خط نمایه پیش‌بینی آن متغیر را زمانی نشان می‌دهد که بقیه در سطح میانگین ثابت نگه داشته شوند. ماهیت افزایشی مدل اجازه می‌دهد تا هر پیش‌بینی به صورت مجزا مشاهده شود. تغییر مقادیر سایر متغیرهای پیش‌بینی، شکل نمایه را تغییر نمی‌دهد، فقط مکان روی محور *y* که نمایه شروع می‌شود، تغییر نمی‌دهد.

7. 3 ماشین‌های بردار پشتیبانی

SVM‌ها دسته‌ای از تکنیک‌های مدل‌سازی قدرتمند و بسیار انعطاف‌پذیر هستند. تئوری پشت SVM‌ها در اصل در چارچوب مدل‌های طبقه‌بندی توسعه داده شد. بعداً در فصل [13 ،](#bookmark612) انگیزه این تکنیک در شکل طبیعی تر آن مورد بحث قرار گرفته است. برای رگرسیون دنبال می‌کنیم [اسمولا](#bookmark1025) [( 1996](#bookmark1025) ) و [دراکر و همکاران ( 1997](#bookmark1015) ) و انگیزه این تکنیک را در چارچوب *رگرسیون قوی* که در آن ما به دنبال به حداقل رساندن اثر پرت بر معادلات رگرسیون هستیم. همچنین، چندین نوع رگرسیون بردار پشتیبان وجود دارد و ما بر روی یک تکنیک خاص به نام *رگرسیون حساس به الکترونیک تمرکز می‌کنیم.*

به یاد بیاورید که رگرسیون خطی به دنبال یافتن تخمین‌های پارامتری است که ­SSE را کوچک می‌کند (بخش.  [6. 2 )](#bookmark286) . یکی از اشکالات به حداقل رساندن SSE این است که تخمین پارامترها را می‌توان تنها تحت تأثیر یک مشاهده قرار داد که از روند کلی داده‌ها فاصله زیادی دارد. وقتی داده‌ها ممکن است حاوی مشاهدات تأثیرگذار باشند، می‌توان از یک معیار کمینه‌سازی جایگزین که حساسیت کمتری دارد، مانند تابع Huber، برای یافتن بهترین تخمین‌های پارامتر استفاده کرد. این تابع از باقیمانده‌های مجذور زمانی که "کوچک" هستند استفاده می‌کند و از باقیمانده‌های مطلق زمانی که باقی مانده‌ها بزرگ هستند استفاده می‌کند. شکل [6. 6](#bookmark290) در صفحه [110](#bookmark290) برای مثال.

SVM‌ها برای رگرسیون از تابعی مشابه تابع Huber استفاده می‌کنند که یک تفاوت مهم دارد. با توجه به آستانه تعیین شده توسط کاربر (که با *e نشان داده می‌شود* )،



NumNonHBonds

NumChlorine

0.0 0.1 0.2 0.3 0.4 0.5

1 23456

شکل 7. 5: رابطه پیش‌بینی‌شده بین نتیجه و ­پیش‌بینی‌کننده‌های پیوسته با استفاده از مدل MARS (همه پیش‌بینی‌کننده‌های دیگر را در مقدار میانگین نگه می‌دارد). ماهیت افزایشی مدل اجازه می‌دهد تا هر پیش‌بینی به صورت مجزا مشاهده شود. توجه داشته باشید که مقادیر پیش‌بینی‌شده نهایی، مجموع هر پروفایل فردی است. پانل‌ها از بالا به پایین بر اساس اهمیت آنها برای مدل مرتب شده اند

نقاط داده با باقیمانده‌ها در آستانه به برازش رگرسیون کمک نمی‌کنند در حالی که نقاط داده با اختلاف مطلق بیشتر از آستانه مقداری در مقیاس خطی دارند. این رویکرد پیامدهای متعددی دارد. اول، از آنجایی که از مجذور باقیمانده استفاده نمی‌شود، مقادیر پرت بزرگ تأثیر محدودی بر معادله رگرسیون دارند. دوم، نمونه‌هایی که مدل به خوبی برازش می‌کند (یعنی باقیمانده‌ها کوچک هستند) *هیچ* تاثیری بر معادله رگرسیون ندارند. در واقع، اگر آستانه روی مقدار نسبتاً بزرگی تنظیم شود، نقاط پرت تنها نقاطی هستند که خط رگرسیون را تعریف می‌کنند! این تا حدودی خلاف واقع است: نقاط پیش‌بینی‌شده ضعیف خط را مشخص می‌کنند. با این حال، نشان داده شده است که این رویکرد در تعریف مدل بسیار موثر است.

برای تخمین پارامترهای مدل، SVM از تابع ضرر *e که در شکل 1 نشان داده شده است استفاده می‌کند.*  [7. 6](#bookmark358) اما یک پنالتی نیز اضافه می‌کند. ضرایب رگرسیون SVM به حداقل می‌رسد

nP

*Cost^L (* *y i - y i* ) + £ *ftft j ,   
i* = 1 *j* = 1

که در آن *L e (* *•* ) تابع حساس به *e* است. پارامتر *Cost جریمه هزینه‌ای* است که توسط کاربر تعیین می‌شود و باقیمانده‌های بزرگ را جریمه می‌کند. [[15]](#footnote-15)

به یاد بیاورید که مدل رگرسیون خطی ساده نمونه‌های جدید را با استفاده از ترکیب خطی داده‌ها و پارامترها پیش‌بینی می‌کرد. برای نمونه جدید، *u* معادله پیش‌بینی است

*y = ft* 0 + *ft* 1 *u* 1 +. *. .* + *ft p u p*

پ

= 0 *$* + *$ j u j*

j = 1

تابع پیش‌بینی ماشین بردار پشتیبان خطی بسیار مشابه است. تخمین ­پارامترها را می‌توان به‌عنوان توابعی از مجموعه‌ای از پارامترهای ناشناخته ( *a i* ) و نقاط داده مجموعه آموزشی نوشت به‌طوری که

*y* = *ft* o + *ft* 1 *u* 1 +. *. .* + *ft p u p*

پ

= 0 *$* + *$ j u j*

j = 1

pn

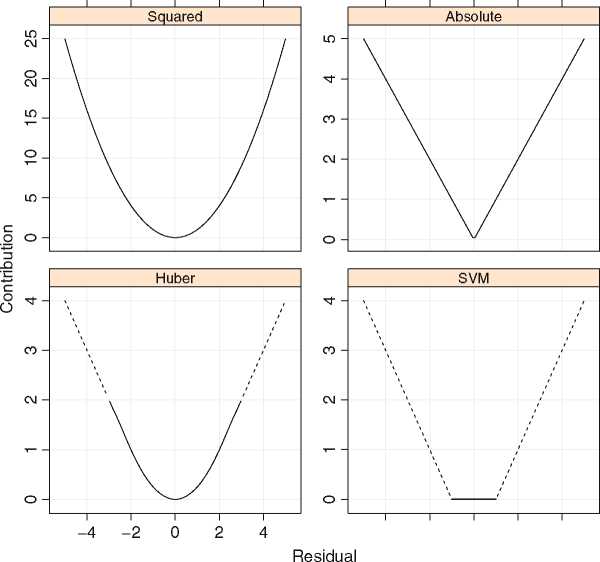
= $ 0 + a i x ij u j

np

= *$* 0 + *a i x ij u j.*  (7. 2)

i = 1 j = 1

-4 -2 0 2 4



شکل 7. 6: رابطه بین باقیمانده مدل و سهم آن در خط رگرسیون برای چندین تکنیک. برای رویکرد هوبر، از آستانه 2 استفاده شد در حالی که برای ماشین بردار پشتیبان، مقدار *e* = 1 استفاده شد. توجه داشته باشید که مقیاس‌های محور *y* برای آسان‌تر خواندن شکل‌ها متفاوت است

چندین جنبه از این معادله وجود دارد که قابل ذکر است. اول، به تعداد نقاط داده پارامتر *a وجود دارد.* از ­نقطه نظر مدل‌سازی رگرسیون کلاسیک، این مدل *بیش از حد ­پارامتری در نظر گرفته می‌شود.* به‌طور معمول، بهتر است پارامترهای کمتری نسبت به نقاط داده تخمین زده شود. با این حال، استفاده از ارزش هزینه به‌طور موثر مدل را منظم می‌کند تا به کاهش این مسأله کمک کند.

دوم، نقاط داده مجموعه آموزشی فردی (یعنی *x ij* ) برای پیش‌بینی‌کننده‌های جدید مورد نیاز است. وقتی مجموعه آموزشی بزرگ باشد، این باعث می‌شود که معادلات پیش‌بینی فشرده‌تر از سایر تکنیک‌ها نباشد. با این حال، برای درصدی از نمونه‌های مجموعه آموزشی، پارامترهای *a i* دقیقاً صفر خواهند بود که نشان ­می‌دهد هیچ تاثیری بر معادله پیش‌بینی ندارند. نقاط داده مرتبط با *پارامتر i* صفر، نمونه‌های مجموعه آموزشی هستند که در *± e* خط رگرسیون قرار دارند (یعنی در داخل "قیف" یا "لوله" در اطراف خط رگرسیون قرار دارند). در نتیجه، تنها زیر مجموعه‌ای از نقاط داده مجموعه آموزشی که در آن *a* = 0، برای پیش‌بینی مورد نیاز است. از آنجایی که خط رگرسیون با استفاده از این نمونه‌ها تعیین می‌شود، آنها *را بردارهای پشتیبان می‌*گویند زیرا از خط رگرسیون پشتیبانی می‌کنند.

شکل [7. 7](#bookmark360) استحکام این مدل را نشان می‌دهد. یک مدل خطی ساده با شیب 4 و فاصله 1 شبیه‌سازی شد. یک عدد پرت شدید به داده‌ها اضافه شد. پانل بالایی مدل مناسب را برای مدل رگرسیون خطی (خط توپر سیاه) و مدل رگرسیون ماشین بردار پشتیبان (خط چین آبی) با *e* = 0 *نشان می‌دهد.* 01. خط رگرسیون خطی به سمت این نقطه کشیده می‌شود که در نتیجه شیب و قطع به ترتیب 3. 5 و 1. 2 برآورد می‌شود. برازش رگرسیون بردار پشتیبان به رنگ آبی نشان داده شده است و به خط رگرسیون واقعی با شیب 3. 9 و فاصله 0. 9 بسیار نزدیکتر است. پانل میانی دوباره مدل SVM را نشان می‌دهد، اما بردارهای پشتیبانی دایره‌های سیاه و سفید و سایر نقاط به رنگ قرمز نشان داده شده اند. خطوط مرجع خاکستری افقی صفر *± e را نشان می‌دهد.* از 100 نقطه داده، 70 مورد از آنها بردار پشتیبان بودند.

در نهایت توجه داشته باشید که در آخرین شکل معادله [7. 2 ،](#bookmark9) نمونه‌های جدید به‌عنوان مجموع محصولات متقاطع با مقادیر نمونه جدید وارد تابع پیش‌بینی می‌شوند. در جبر ماتریسی، این مربوط به یک *محصول نقطه‌ای (*یعنی x *'* u ) است. این مهم است زیرا این معادله رگرسیون را می‌توان به‌طور کلی به صورت *n بازنویسی کرد*

*f (* u ) = *p* o + ^ 2 *a i K (* x *i ,* u ) *,   
i* =1

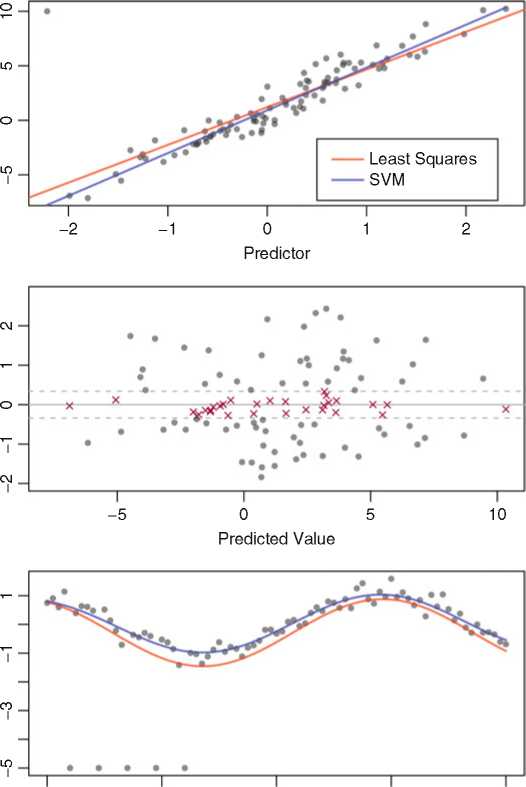
که در آن *K (* *•* ) *تابع هسته نامیده می‌شود.* هنگامی که پیش‌بینی‌کننده‌ها به صورت خطی وارد مدل می‌شوند، تابع هسته به یک مجموع ساده از محصولات متقاطع نشان داده شده در بالا کاهش می‌یابد *:*

K ( x i , u ) = '^^xij U j = x i u.

j = 1

با این حال، انواع دیگری از توابع هسته وجود دارند که می‌توانند برای ­کلی‌سازی مدل رگرسیون استفاده شوند و توابع *غیرخطی پیش‌بینی‌کننده‌ها را* در بر گیرند: چند جمله‌ای = ( *p (* x'u *)* + 1) تابع پایه شعاعی *درجه* = exp( *—a|* | x *—* u *|* | 2) مماس هذلولی = tanh ( *p (* x'u *) +* 1)،

که در آن *p* و *a* پارامترهای مقیاس هستند. از آنجایی که این توابع پیش‌بینی‌کننده‌ها منجر به مدل‌های غیرخطی می‌شوند، این تعمیم اغلب «ترفند هسته» نامیده می‌شود. برای نشان دادن توانایی این مدل برای انطباق با روابط غیرخطی، داده‌هایی را شبیه‌سازی کردیم که از یک موج سینوسی در پایین شکل 1 پیروی می‌کنند.  [7. 7](#bookmark360) . موارد پرت نیز به این داده‌ها اضافه شد. یک مدل رگرسیون خطی با یک وقفه و یک عبارت برای *sin (* *x* ) با مدل (خط سیاه یکدست) مطابقت داشت. دوباره، خط رگرسیون به سمت نقاط بیرون کشیده می‌شود. یک مدل SVM با تابع هسته پایه شعاعی با خط چین آبی (بدون مشخص کردن شکل تابعی *sin ) نشان داده می‌شود.* این خط ساختار کلی داده‌ها را بهتر توصیف می‌کند.



46

10

Predictor

Fig. 7.7: The robustness qualities of SVM models. *Top*: a small simulated data set with a single large outlier is used to show the difference between an ordinary regression line (*red*) and the linear SVM model (*blue*). *Middle*: the SVM residuals versus the predicted values (the upper end of the *y*-axis scale was reduced to make the plot more readable). The plot symbols indicate the support vectors (shown as *grey colored circles*) and the other samples (*red crosses*). The *horizontal lines* are *±e* = 0*.*01. *Bottom*: A simulated sin wave with several outliers. The *red line* is an ordinary regression line (intercept and a term for *sin*(*x*)) and the *blue line* is a radial basis function SVM model

کدام تابع هسته باید استفاده شود؟ این بستگی به مسأله دارد. تابع پایه شعاعی بسیار موثر نشان داده شده است. با این حال، زمانی که خط رگرسیون واقعاً خطی باشد، تابع هسته خطی انتخاب بهتری خواهد بود.

توجه داشته باشید که برخی از توابع هسته دارای پارامترهای اضافی هستند. به‌عنوان مثال، درجه چند جمله‌ای در هسته چند جمله‌ای باید مشخص شود. به‌طور مشابه، تابع پایه شعاعی دارای یک پارامتر ( *a* ) است که مقیاس را کنترل می‌کند. این پارامترها، همراه با ارزش هزینه، پارامترهای تنظیم مدل را تشکیل می‌دهند. در مورد تابع پایه شعاعی، یک ­میانبر محاسباتی ممکن برای تخمین پارامتر هسته وجود دارد.  [کاپوتو و همکاران ( 2002](#bookmark1012) ) پیشنهاد کرد که پارامتر را می‌توان با استفاده از ترکیبی از نقاط مجموعه آموزشی برای محاسبه توزیع *||x — x ' || تخمین زد.* 2، سپس از صدک 10 و 90 به‌عنوان یک محدوده برای *یک* استفاده کنید. به جای تنظیم این پارامتر ­بر روی شبکه‌ای از مقادیر کاندید، می‌توانیم از نقطه میانی این دو صدک استفاده کنیم.

پارامتر هزینه ابزار اصلی برای تنظیم پیچیدگی مدل است. وقتی هزینه زیاد باشد، مدل بسیار انعطاف‌پذیر می‌شود زیرا اثر خطاها تقویت می‌شود. هنگامی که هزینه کوچک است، مدل «سفت‌تر» می‌شود و احتمال کمتر شدن بیش‌ازحد (اما به احتمال زیاد کمتر) می‌شود، زیرا سهم پارامترهای مربع به نسبت در تابع خطای اصلاح‌شده بزرگ است. همچنین می‌توان مدل را روی اندازه قیف تنظیم کرد (مثلاً روی *e* ). با این حال، بین *e* و پارامتر هزینه رابطه وجود دارد. در تجربه ما متوجه شده ایم که پارامتر هزینه انعطاف بیشتری را برای تنظیم مدل فراهم می‌کند. بنابراین پیشنهاد می‌کنیم مقدار *e را ثابت کنید* و روی سایر پارامترهای هسته تنظیم کنید.

از آنجایی که پیش‌بینی‌کننده‌ها به‌عنوان مجموع محصولات متقاطع وارد مدل می‌شوند، تفاوت در مقیاس‌های پیش‌بینی می‌تواند بر مدل تأثیر بگذارد. بنابراین، ما توصیه می‌کنیم ­که پیش‌بینی‌کننده‌ها را قبل از ساخت یک مدل SVM متمرکز و مقیاس‌بندی کنیم.

SVM‌ها برای داده‌های حلالیت اعمال شدند. ابتدا از یک هسته تابع پایه شعاعی استفاده شد. پارامتر هسته به صورت تحلیلی *a* = 0 برآورد شد. 0039 و مدل بر روی 14 مقدار هزینه بین 0. 25 و 2048 در مقیاس log 2 تنظیم شد (شکل 2).  [7. 8 )](#bookmark362) . هنگامی که مقادیر هزینه کوچک هستند، مدل *با* داده‌ها مطابقت دارد، اما، با شروع به افزایش خطا با نزدیک شدن هزینه به ­2 10، برازش بیش از حد شروع می‌شود. ارزش هزینه مرتبط با کوچکترین RMSE 128 بود. یک مدل چند جمله‌ای نیز ارزیابی شد. در اینجا، ما هزینه، درجه چند جمله‌ای و یک ضریب مقیاس را تنظیم کردیم. به‌طور کلی، مدل‌های درجه دوم ­نسبت به مدل‌های خطی نرخ خطای کمتری دارند. همچنین مدل‌های مرتبط با فاکتورهای مقیاس بزرگتر عملکرد بهتری دارند. مدل بهینه درجه دوم با ضریب مقیاس 0. 01 و ارزش هزینه 2 بود (شکل 1).  [7. 9 )](#bookmark362) .

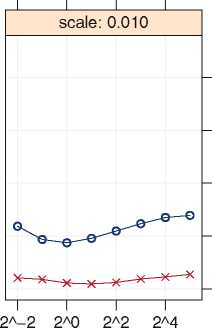
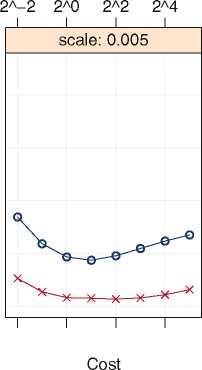
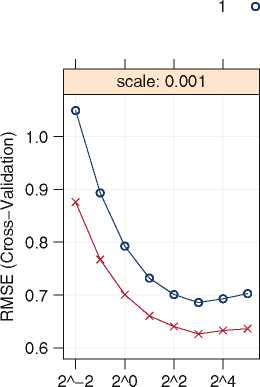
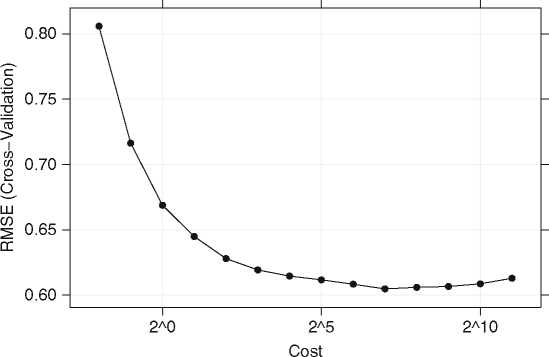
به‌عنوان مقایسه، هر دو مدل پایه شعاعی بهینه و مدل SVM چند جمله‌ای از تعداد مشابهی بردار پشتیبان، به ترتیب 623 و 627 (از 951 نمونه آموزشی) استفاده می‌کنند. همچنین ذکر این نکته ضروری است که تنظیم پارامتر هسته تابع پایه شعاعی آسانتر از تنظیم مدل چند جمله‌ای (که دارای سه پارامتر تنظیم است) بود.

ادبیات مربوط به مدل‌های SVM و سایر روش‌های هسته ­پررنگ بوده و بسیاری از روش‌های جایگزین پیشنهاد شده‌اند. یک روش،

شکل 7. 8: پروفایل‌های اعتبارسنجی متقاطع برای مدل SVM تابع پایه شعاعی به داده‌های حلالیت اعمال می‌شود. پارامتر هسته به‌طور تحلیلی 0 = *a برآورد شد.* 0039

درجه چند جمله ای

شکل 7. 9: نتایج اعتبارسنجی متقابل برای مدل چند جمله‌ای SVM برای ­داده‌های قابلیت حل پذیری. مدل نهایی با استفاده از مدل درجه دوم با ضریب مقیاس 0. 01 و ارزش هزینه 2 برازش شد.



بردار *ربط* [( Tipping 2001](#bookmark1026) )، یک آنالوگ بیزی مدل SVM است. در این مورد، پارامترهای *a* که در بالا توضیح داده شد، دارای توزیع‌های قبلی هستند و انتخاب *بردارهای مرتبط* با استفاده از توزیع پسین آنها تعیین می‌شود. اگر توزیع خلفی بسیار ­حول محور صفر متمرکز باشد، نمونه در معادله پیش‌بینی استفاده نمی‌شود. معمولاً بردارهای مرتبط کمتری در این مدل نسبت به بردارهای پشتیبانی در یک مدل SVM وجود دارد.

7. 4 *K* -نزدیکترین همسایه ها

رویکرد *K* NN به سادگی یک نمونه جدید را با استفاده از *K-* نزدیکترین نمونه‌ها ­از مجموعه آموزشی پیش‌بینی می‌کند (مشابه شکل 1).  [4. 3 )](#bookmark199) . برخلاف روش‌های دیگر در این فصل، *K* NN را نمی‌توان با مدلی مانند مدل ­ارائه شده در معادله خلاصه کرد.  [7. 2 .](#bookmark9) در عوض، ساخت آن صرفاً بر اساس نمونه‌های فردی از داده‌های آموزشی است. برای پیش‌بینی یک نمونه جدید برای رگرسیون، *K NN K* NN آن نمونه را در فضای پیش‌بینی شناسایی می‌کند. پاسخ پیش‌بینی‌شده برای نمونه جدید میانگین پاسخ همسایگان *K است.* سایر آمارهای خلاصه مانند میانه نیز می‌توانند به جای میانگین برای پیش‌بینی نمونه جدید استفاده شوند.

روش پایه *K* NN همانطور که در بالا توضیح داده شد به نحوه تعریف کاربر فاصله بین نمونه‌ها بستگی دارد. فاصله اقلیدسی (یعنی فاصله خط مستقیم بین دو نمونه) رایج‌ترین معیار مورد استفاده است و به صورت زیر تعریف می‌شود:

1

*P* \ 2

^ ' (*x aj - x bj* ) *,*

j = 1

که در آن x a و x b دو نمونه جداگانه هستند. فاصله مینکوفسکی یک ­کلی‌سازی فاصله اقلیدسی است و به این صورت تعریف می‌شود

1

*P \ q*

^' l x aj - x bj | 9  ،

j = 1

جایی که *q>* 0 ( [لیو 2007](#bookmark1021) ). به راحتی می‌توان فهمید که وقتی *q* = 2 باشد، فاصله مینکوفسکی همان فاصله اقلیدسی است. وقتی *q* = 1 باشد، فاصله مینکوفسکی ­معادل فاصله منهتن (یا بلوک شهر) است که یک معیار معمولی است که برای نمونه‌هایی با پیش‌بینی‌کننده‌های باینری استفاده می‌شود. بسیاری دیگر از معیارهای فاصله وجود دارند، مانند تانیموتو، همینگ و کسینوس و برای انواع خاصی از پیش‌بینی‌کننده‌ها و در زمینه‌های علمی خاص مناسب‌تر هستند. برای مثال، زمانی که مولکول‌ها با استفاده از اثر انگشت باینری توصیف می‌شوند، از فاصله تانیموتو به‌طور منظم در مسائل شیمی محاسباتی استفاده می‌شود ( ­[مک کارن و همکاران.](#bookmark1022)  [2011](#bookmark1022) ).

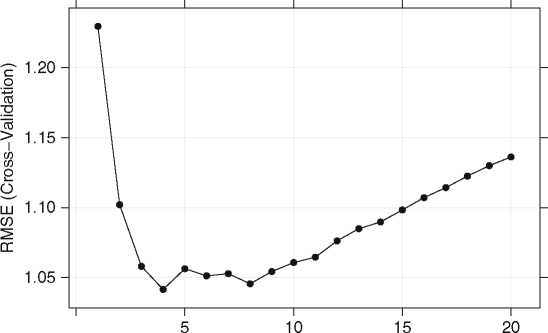
از آنجایی که روش *K* NN اساساً به فاصله بین نمونه‌ها بستگی دارد، مقیاس پیش‌بینی‌کننده‌ها می‌تواند تأثیر چشمگیری بر فواصل بین نمونه‌ها داشته باشد. داده‌هایی با پیش‌بینی‌کننده‌هایی که در مقیاس‌های بسیار متفاوت هستند، فواصلی را ایجاد می‌کنند که نسبت به پیش‌بینی‌کننده‌هایی که بزرگترین مقیاس‌ها را دارند، وزن می‌شوند. یعنی پیش‌بینی‌کننده‌هایی با بزرگترین مقیاس‌ها بیشترین سهم را در فاصله بین نمونه‌ها دارند. برای جلوگیری از این بایاس بالقوه و فعال کردن هر پیش‌بینی برای مشارکت یکسان در محاسبه فاصله، توصیه می‌کنیم که همه پیش‌بینی‌کننده‌ها قبل از انجام *K* NN در مرکز و مقیاس قرار گیرند.

علاوه بر مسئله مقیاس‌بندی، استفاده از فواصل بین نمونه‌ها در صورتی که یک یا چند مقدار پیش‌بینی برای یک نمونه وجود نداشته باشد، می‌تواند مسأله‌ساز باشد، زیرا محاسبه فاصله بین نمونه‌ها ممکن نیست. اگر اینطور باشد، تحلیلگر چند گزینه دارد. ابتدا، نمونه‌ها یا پیش‌بینی‌کننده‌ها را می‌توان از تحلیل حذف کرد. این نامطلوب‌ترین گزینه است. با این حال، اگر نمونه(ها) یا پیش‌بینی‌کننده‌ها پراکنده باشند، ممکن است تنها انتخاب عملی باشد. اگر یک پیش‌بینی حاوی مقدار کافی اطلاعات در سراسر نمونه‌ها باشد، یک رویکرد جایگزین این است که داده‌های گمشده را با استفاده از یک برآوردگر ساده‌لوح مانند میانگین پیش‌بینی، یا یک رویکرد نزدیک‌ترین همسایه که فقط از پیش‌بینی‌کننده‌ها با اطلاعات کامل استفاده می‌کند، نسبت داده شود. فرقه [3. 4 )](#bookmark157) .

پس از پیش پردازش داده‌ها و انتخاب معیار فاصله، گام بعدی یافتن تعداد بهینه همسایگان است. مانند پارامترهای تنظیم از مدل‌های دیگر، *K* را می‌توان با نمونه‌گیری مجدد تعیین کرد. برای داده‌های حلالیت، 20 مقدار *K* در محدوده 1 تا 20 مورد ارزیابی قرار گرفت. همانطور که در شکل نشان داده شده است.  [7. 10 ،](#bookmark367) نمایه RMSE به سرعت در چهار مقدار اول *K کاهش می‌یابد،* سپس از طریق *K = 8 کاهش می‌یابد و به دنبال آن با افزایش K،* RMSE افزایش ثابتی پیدا می‌کند. این نمایه عملکرد برای *K* NN معمولی است، زیرا مقادیر کوچک *K* معمولاً بیش برازش دارند و مقادیر بزرگ *K کمتر* از داده‌ها هستند. RMSE از 1. 041 تا 1. 23 در سراسر مقادیر نامزد، با حداقل در *K* = 4 رخ می‌دهد. تایید متقابل *R2* در *K* بهینه 0. 747 است.

نسخه ابتدایی *K* NN بصری و ساده است و می‌تواند پیش‌بینی‌کننده‌های مناسبی را ایجاد کند، به خصوص زمانی که پاسخ به ساختار پیش‌بینی محلی وابسته باشد. با این حال، این نسخه دارای مسائل قابل‌توجهی است که محققان به دنبال راه حلی برای آنها بوده اند. دو مسأله متداول ذکر شده زمان محاسباتی و قطع ارتباط بین ساختار محلی و توانایی پیش‌بینی *K* NN است.

ابتدا برای پیش‌بینی یک نمونه، فاصله بین نمونه و سایر نمونه‌ها باید محاسبه شود. بنابراین زمان محاسبات با *n افزایش می‌یابد* زیرا داده‌های آموزشی باید در حافظه بارگذاری شوند و به این دلیل که فاصله بین نمونه جدید و همه نمونه‌های آموزشی باید محاسبه شود. برای کاهش این مسأله، می‌توان داده‌های اصلی را با نمایشی با حافظه کم‌تر از داده‌ها جایگزین کرد ­که مکان‌های داده اصلی را توصیف می‌کند. یکی از نمونه‌های خاص این نمایش، یک درخت با ابعاد *k (*یا درخت *k* -d) است [( بنتلی 1975](#bookmark1011) ). یک درخت *k -* d به‌طور متعامد فضای پیش‌بینی را با استفاده از رویکرد درختی تقسیم می‌کند اما با قوانینی متفاوت از انواع درخت‌های توصیف شده



#همسایه ها

شکل 7. 10: نمایه اعتبارسنجی متقابل RMSE برای یک مدل *K* NN به داده‌های حلالیت اعمال می‌شود. تعداد بهینه همسایه‌ها 4 است

در فصل [8 .](#bookmark388) پس از رشد درخت، یک نمونه جدید در ساختار قرار می‌گیرد. فاصله‌ها فقط برای مشاهدات آموزشی در درخت که نزدیک به نمونه جدید هستند محاسبه می‌شود. این رویکرد پیشرفت‌های محاسباتی قابل‌توجهی را ارائه می‌دهد، به خصوص زمانی که تعداد نمونه‌های آموزشی بسیار بیشتر از تعداد پیش‌بینی‌کننده‌ها باشد.

روش *K* NN می‌تواند عملکرد پیش‌بینی ضعیفی داشته باشد زمانی که ­ساختار پیش‌بینی محلی به پاسخ مربوط نباشد. پیش‌بینی‌کننده‌های نامربوط یا پر سر و صدا یکی از مقصران هستند، زیرا می‌توانند باعث دور شدن نمونه‌های مشابه از یکدیگر در فضای پیش‌بینی شوند. از این رو، حذف پیش‌دیکتورهای نامربوط و مملو از نویز ­یک مرحله پیش پردازش کلیدی برای *K* NN است. رویکرد دیگر برای افزایش پیش‌بینی *K* NN این است که سهم همسایگان در پیش‌بینی یک نمونه جدید بر اساس فاصله آنها تا نمونه جدید وزن شود. در این تغییر، نمونه‌های آموزشی که به نمونه جدید نزدیک‌تر هستند، بیشتر به پاسخ پیش‌بینی‌شده کمک می‌کنند، در حالی که نمونه‌هایی که دورتر هستند، کمتر به پاسخ پیش‌بینی‌شده کمک می‌کنند.

7. 5 محاسبات

این بخش به توابع بسته‌های caret، earth، kernlab و nnet اشاره می‌کند.

R تعدادی بسته و توابع برای ایجاد شبکه‌های عصبی دارد. بسته‌های مربوطه شامل nnet، عصبی و RSNNS است. بسته nnet در اینجا تمرکز دارد زیرا از مدل‌های شبکه عصبی پایه که در این فصل توضیح داده شده است (یعنی یک لایه واحد از واحدهای پنهان) و کاهش وزن پشتیبانی می‌کند و دارای ­نحو ساده است. RSNNS su [pp](#bookmark1011) آرایه وسیعی از شبکه‌های عصبی را ارائه می‌دهد. برگمیر و بنیتز [( 2012](#bookmark1011) ) بسته‌های مختلف شبکه عصبی را در R تشریح می‌کنند و حاوی آموزش RSNNS هستند.

شبکه‌های عصبی

برای برازش یک مدل رگرسیون، تابع nnet هر دو رابط فرمول و غیر فرمول را می‌گیرد. ­برای رگرسیون، رابطه خطی بین واحدهای پنهان و پیش‌بینی را می‌توان با گزینه lineout = TRUE استفاده کرد. یک فراخوانی تابع اصلی شبکه عصبی خواهد بود

*nnetFit <- nnet(پیش‌بینی‌ها، نتیجه،*

*+ سایز = 5،*

*+ فروپاشی = 0. 01،*

*+ خط خطی = درست،*

*+ ## مقدار خروجی چاپ شده را کاهش دهید*

*+ ردیابی = نادرست،*

*+ ## تعداد تکرارها را برای پیدا کردن گسترش دهید*

*+ ## تخمین پارامتر. .*

*+ حداکثر = 500،*

*+ ## و تعداد پارامترهای استفاده شده توسط مدل*

*+ MaxNWts = 5 \* (ncol(پیش‌بینی ها) + 1)+5+1)*

این یک مدل واحد با 5 واحد مخفی ایجاد می‌کند. توجه داشته باشید، این فرض را بر این می‌گذارد که داده‌ها در پیش‌بینی‌کننده‌ها استاندارد شده‌اند تا در یک مقیاس باشند.

برای استفاده از میانگین‌گیری مدل، تابع avNNet در بسته caret دارای نحو تقریباً یکسانی است:

*nnetAvg <- avNNet(پیش‌بینی‌ها، نتیجه،*

*+ سایز = 5،*

*+ فروپاشی = 0. 01،*

*+ ## مشخص کنید که چند مدل به‌طور میانگین*

*+ تکرار = 5،*

*+ خط خطی = درست،*

*+ ## مقدار خروجی چاپ شده را کاهش دهید*

*+ ردیابی = نادرست،*

*+ ## تعداد تکرارها را برای پیدا کردن گسترش دهید*

*+ ## تخمین پارامتر. .*

*+ حداکثر = 500،*

*+ ## و تعداد پارامترهای استفاده شده توسط مدل*

*+ MaxNWts = 5 \* (ncol(پیش‌بینی ها) + 1) + 5 + 1)*

دوباره، نمونه‌های جدید با استفاده از پردازش می‌شوند

*پیش‌بینی (nnetFit، newData)*

*>##یا*

*پیش‌بینی (nnetAvg، newData)*

برای تقلید از رویکرد قبلی انتخاب تعداد واحدهای پنهان و میزان کاهش وزن از طریق نمونه‌گیری مجدد، تابع آموزش را می‌توان با استفاده از روش = "nnet" یا روش = "avNNet" اعمال کرد. ابتدا، پیش‌بینی‌کننده‌ها را حذف می‌کنیم ­تا اطمینان حاصل کنیم که حداکثر همبستگی زوجی مطلق بین پیش‌بینی‌کننده‌ها کمتر از 0. 75 است.

*## findCorrelation یک ماتریس همبستگی می‌گیرد و آن را تعیین می‌کند*

*## اعداد ستون که باید حذف شوند تا همه به صورت جفتی حفظ شوند*

*## همبستگی زیر یک آستانه*

*tooHigh <- findCorrelation(cor(solTrainXtrans)، برش = 0. 75)*

*trainXnnet <- solTrainXtrans[, -tooHigh]*

*testXnnet <- solTestXtrans[, -tooHigh]*

*## یک مجموعه کاندیدای خاص از مدل‌ها برای ارزیابی ایجاد کنید:*

*nnetGrid <- expand. grid(. decay = c(0, 0. 01,. 1)*

*+ . size = c(1:10)،*

*+ ## گزینه بعدی استفاده از بسته‌بندی است (به قسمت مراجعه کنید*

*+ ## فصل بعدی) به جای تصادفی متفاوت*

*+ ## دانه.*

*+ . bag = FALSE)*

*set. seed (100)*

*nnetTune <- train(solTrainXtrans, solTrainY,*

*+ روش = "avNNet"،*

*+ tuneGrid = nnetGrid،*

*+ trControl = ctrl،*

*+ ## به‌طور خودکار داده‌ها را قبل از مدل‌سازی استاندارد کنید*

*+ ## و پیش‌بینی*

*+ preProc = c ("مرکز"، "مقیاس")،*

*+ خط خطی = درست،*

*+ ردیابی = نادرست،*

*+ MaxNWts = 10 \* (ncol(trainXnnet) + 1) + 10 + 1،*

*+ حداکثر = 500)*

Splines رگرسیون تطبیقی چند متغیره

مدل‌های MARS در چندین بسته هستند، اما گسترده‌ترین پیاده‌سازی در بسته زمین است. مدل MARS با استفاده از گذر اسمی رو به جلو و گام هرس را می‌توان به سادگی نامید

*marsFit <- earth (solTrainXtrans، solTrainY)*

*marsFit*

38 مورد از 47 عبارت و 30 مورد از 228 پیش‌بینی انتخاب شد

اهمیت: NumNonHAtoms، MolWeight، SurfaceArea2، SurfaceArea1، FP142،. . .

تعداد اصطلاحات در هر درجه تعامل: 1 37 (مدل افزودنی)

GCV 0. 3877448 RSS 312. 877 GRSq 0. 907529 RSq 0. 9213739

توجه داشته باشید که از آنجایی که این مدل از تکنیک GCV داخلی برای انتخاب مدل استفاده ­می‌کند، جزئیات این مدل با آنچه قبلاً در این فصل استفاده شده بود متفاوت است. روش خلاصه خروجی گسترده‌تری تولید می‌کند:

*خلاصه (marsFit)*

فراخوانی: earth(x=solTrainXtrans، y=solTrainY)

|  |  |
| --- | --- |
|  | ضرایب |
| (عرض از مبدأ) | -3. 223749 |
| FP002 | 0. 517848 |
| FP003 | -0. 228759 |
| FP059 | -0. 582140 |
| FP065 | -0. 273844 |
| FP075 | 0. 285520 |
| FP083 | -0. 629746 |
| FP085 | -0. 235622 |
| FP099 | 0. 325018 |
| FP111 | -0. 403920 |
| FP135 | 0. 394901 |
| FP142 | 0. 407264 |
| FP154 | -0. 620757 |
| FP172 | -0. 514016 |
| FP176 | 0. 308482 |
| FP188 | 0. 425123 |
| FP202 | 0. 302688 |
| FP204 | -0. 311739 |
| FP207 | 0. 457080 |
| h (MolWeight-5. 77508) | -1. 801853 |
| h (5. 94516-مول وزن) | 0. 813322 |
| h(NumNonHAtoms-2. 99573) | -3. 247622 |
| h(2. 99573-NumNonHAtoms) | 2. 520305 |
| h(2. 57858-NumNonHBonds) | -0. 564690 |
| h(NumMultBonds-1. 85275) | -0. 370480 |
| h (NumRotBonds-2. 19722) | -2. 753687 |
| h(2. 19722-NumRotBonds) | 0. 123978 |
| h(NumAromaticBonds-2. 48491) | -1. 453716 |
| h(NumNitrogen-0. 584815) | 8. 239716 |
| h(0. 584815-NumNitrogen) | -1. 542868 |
| h(NumOxygen-1. 38629) | 3. 304643 |
| h (1. 38629-NumOxygen) | -0. 620413 |
| h (NumChlorine-0. 46875) | -50. 431489 |
| h (HydrophilicFactor- -0. 816625) | 0. 237565 |
| h(-0. 816625-HydrophilicFactor) | -0. 370998 |
| h (SurfaceArea1-1. 9554) | 0. 149166 |
| h (SurfaceArea2-4. 66178) | -0. 169960 |
| h(4. 66178-SurfaceArea2) | -0. 157970 |
| 38 مورد از 47 عبارت و 30 مورد انتخاب شد | از 228 پیش‌بینی |

اهمیت: NumNonHAtoms، MolWeight، SurfaceArea2، SurfaceArea1، FP142، تعداد اصطلاحات در هر درجه تعامل: 1 37 (مدل افزودنی)

GCV 0. 3877448 RSS 312. 877 GRSq 0. 907529 RSq 0. 9213739

در این خروجی، *h (* *•* ) تابع لولا است. در خروجی بالا، زمانی که وزن مولکولی کمتر از 5. 77508 باشد، عبارت h(MolWeight-5. 77508) صفر است (یعنی مشابه پانل بالایی شکل 1).  [7. 3 )](#bookmark350) . تابع لولا منعکس شده به صورت h (5. 77508 - MolWeight) نشان داده می‌شود.

تابع plotmo در بسته ارت را می‌توان برای تولید نمودارهایی مشابه شکل 1 استفاده کرد.  [7. 5 .](#bookmark357) برای تنظیم مدل با استفاده از نمونه برداری مجدد خارجی، می‌توان از تابع آموزش استفاده کرد. کد زیر نتایج را در شکل 1 بازتولید می‌کند.  [7. 4](#bookmark353) :

*# مدل‌های کاندید را برای آزمایش تعریف کنید*

*marsGrid <- expand. grid(. degree = 1:2،. nprune = 2:38)*

*# دانه را ثابت کنید تا نتایج دوباره تولید شود*

*set. seed (100)*

*marsTuned <- train(solTrainXtrans, solTrainY,*

*+ روش = "زمین"،*

*+ # صراحتاً مدل‌های کاندید را برای آزمایش اعلام کنید*

*+ tuneGrid = marsGrid،*

*+ trControl = trainControl (روش = "cv"))*

*marsTuned*

951 نمونه

228 پیش‌بینی

بدون پیش پردازش

نمونه‌گیری مجدد: اعتبارسنجی متقاطع (10 برابر)

خلاصه حجم نمونه: 856، 857، 855، 856، 856، 855،

نمونه برداری مجدد از نتایج در پارامترهای تنظیم:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| درجه | nprune | RMSE | Rsquared | RMSE SD | Rsquared SD |
| 1 | 2 | 1. 54 | 0. 438 | 0. 128 | 0. 0802 |
| 1 | 3 | 1. 12 | 0. 7 | 0. 0968 | 0. 0647 |
| 1 | 4 | 1. 06 | 0. 73 | 0. 0849 | 0. 0594 |
| 1 | 5 | 1. 02 | 0. 75 | 0. 102 | 0. 0551 |
| 1 | 6 | 0. 984 | 0. 768 | 0. 0733 | 0. 042 |
| 1 | 7 | 0. 919 | 0. 796 | 0. 0657 | 0. 0432 |
| 1 | 8 | 0. 862 | 0. 821 | 0. 0418 | 0. 0237 |
| 2 | 33 | 0. 701 | 0. 883 | 0. 068 | 0. 0307 |
| 2 | 34 | 0. 702 | 0. 883 | 0. 0699 | 0. 0307 |
| 2 | 35 | 0. 696 | 0. 885 | 0. 0746 | 0. 0315 |
| 2 | 36 | 0. 687 | 0. 887 | 0. 0604 | 0. 0281 |
| 2 | 37 | 0. 696 | 0. 885 | 0. 0689 | 0. 0291 |
| 2 | 38 | 0. 686 | 0. 887 | 0. 0626 | 0. 029 |

برای انتخاب مدل بهینه با استفاده از کوچکترین مقدار از RMSE استفاده شد. مقادیر نهایی مورد استفاده برای مدل درجه = 1 و nprune = 38 بود.

*head(predict(marsTuned، solTestXtrans))*

[1] 0. 3677522 -0. 1503220 -0. 5051844 0. 5398116 -0. 4792718 0. 7377222

دو تابع وجود دارد که اهمیت هر پیش‌بینی را در مدل MARS تخمین می‌زند: evimp در بسته زمین و varImp در بسته caret (اگرچه دومی اولی را فراخوانی می‌کند):

*> varImp (marsTuned)*

اهمیت متغیر زمین

تنها 20 متغیر مهم نشان داده شده است (از 228)

|  |  |
| --- | --- |
|  | به‌طور کلی |
| مولی وزن | 100. 00 |
| NumNonHAtoms | 89. 96 |
| سطح 2 | 89. 51 |
| سطح 1 | 57. 34 |
| FP142 | 44. 31 |
| FP002 | 39. 23 |
| NumMultBonds | 39. 23 |
| FP204 | 37. 10 |
| FP172 | 34. 96 |
| NumOxygen | 30. 70 |
| NumNitrogen | 29. 12 |
| FP083 | 28. 21 |
| NumNonHBonds | 26. 58 |
| FP059 | 24. 76 |
| FP135 | 23. 51 |
| FP154 | 21. 20 |
| FP207 | 19. 05 |
| FP202 | 17. 92 |
| NumRotBonds | 16. 94 |
| FP085 | 16. 02 |

این نتایج بین 0 تا 100 مقیاس‌بندی شده و با نتایج نشان داده شده در جدول متفاوت است [7. 1](#bookmark351) (از مدل موجود در جدول [7. 1](#bookmark351) تحت فرآیند رشد و هرس مدل کامل قرار نگرفت). توجه داشته باشید که بعد از چند متغیر اول، بقیه اهمیت بسیار کمتری برای مدل دارند.

ماشین‌های بردار پشتیبانی

تعدادی بسته R با پیاده‌سازی مدل‌های ماشین بردار پشتیبانی وجود دارد. تابع svm در بسته e1071 یک رابط با کتابخانه LIBSVM [( چانگ و لین 2011](#bookmark1013) ) برای رگرسیون دارد. یک ­پیاده‌سازی جامع تر از مدل‌های SVM برای رگرسیون، بسته kernlab است [( Karatzoglou et al. 2004](#bookmark1019) ). در آن بسته، تابع ksvm برای مدل‌های رگرسیون و تعداد زیادی از توابع هسته موجود است. تابع پایه شعاعی تابع هسته پیش فرض است. اگر مقادیر مناسب پارامترهای هزینه و هسته شناخته شده باشد، این مدل می‌تواند به‌عنوان مناسب باشد

*svmFit <- ksvm(x = solTrainXtrans، y = solTrainY، + هسته = "rbfdot"، kpar = "خودکار"،*

*+ C = 1، اپسیلون = 0. 1)*

تابع به‌طور خودکار از رویکرد تحلیلی برای تخمین *یک.* از آنجایی که y یک بردار عددی است، تابع می‌داند که یک مدل رگرسیون (به جای یک مدل طبقه بندی) برازش می‌کند. توابع دیگر هسته را می‌توان استفاده کرد، از جمله چند جمله‌ای (با استفاده از هسته = "polydot" ) و خطی ( kernel = "vanilladot" ).

اگر مقادیر ناشناخته باشند، می‌توان آنها را از طریق نمونه‌گیری مجدد تخمین زد. در train، مقادیر متد «svmRadial»، «svmLinear» یا «svmPoly» با هسته‌های مختلف مطابقت دارند:

*svmRTuned <- train(solTrainXtrans, solTrainY,*

*+ روش = "svmRadial"،*

*+ preProc = c ("مرکز"، "مقیاس")،*

*+ آهنگ طول = 14،*

*+ trControl = trainControl (روش = "cv"))*

tuneLength از جستجوی شبکه پیش‌فرض 14 مقدار هزینه بین 2 *\_* 2،2 *\_* 1،*. . . ،* 2 11 استفاده می‌کند. مجدداً، *a* به‌طور پیش فرض به صورت تحلیلی تخمین زده می‌شود.

*svmRTuned*

951 نمونه

228 پیش‌بینی

پیش پردازش: متمرکز، مقیاس شده

نمونه‌گیری مجدد: اعتبارسنجی متقاطع (10 برابر)

خلاصه حجم نمونه: 855، 858، 856، 855، 855، 856،. . .

نمونه برداری مجدد از نتایج در پارامترهای تنظیم:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| سی | RMSE | Rsquared | RMSE SD | Rsquared SD |
| 0. 25 | 0. 793 | 0. 87 | 0. 105 | 0. 0396 |
| 0. 5 | 0. 708 | 0. 889 | 0. 0936 | 0. 0345 |
| 1 | 0. 664 | 0. 898 | 0. 0834 | 0. 0306 |
| 2 | 0. 642 | 0. 903 | 0. 0725 | 0. 0277 |
| 4 | 0. 629 | 0. 906 | 0. 067 | 0. 0253 |
| 8 | 0. 621 | 0. 908 | 0. 0634 | 0. 0238 |
| 16 | 0. 617 | 0. 909 | 0. 0602 | 0. 0232 |
| 32 | 0. 613 | 0. 91 | 0. 06 | 0. 0234 |
| 64 | 0. 611 | 0. 911 | 0. 0586 | 0. 0231 |
| 128 | 0. 609 | 0. 911 | 0. 0561 | 0. 0223 |
| 256 | 0. 609 | 0. 911 | 0. 056 | 0. 0224 |
| 512 | 0. 61 | 0. 911 | 0. 0563 | 0. 0226 |
| 1020 | 0. 613 | 0. 91 | 0. 0563 | 0. 023 |
| 2050 | 0. 618 | 0. 909 | 0. 0541 | 0. 023 |

پارامتر تنظیم سیگما در مقدار 0. 00387 RMSE برای انتخاب مدل بهینه با استفاده از کوچکترین مقدار استفاده شد. مقادیر نهایی مورد استفاده برای مدل C = 256 و سیگما = 0. 00387 بود.

فرعی با نام finalModel شامل مدل ایجاد شده توسط تابع ksvm است:

*svmRTuned$finalModel*

شیء ماشین بردار پشتیبانی از کلاس "ksvm"

نوع SV: پارامتر eps-svr (رگرسیون): اپسیلون = 0. 1 هزینه C = 256

تابع هسته پایه شعاعی گاوسی.

هایپرپارامتر: سیگما = 0. 00387037424967707

تعداد بردارهای پشتیبانی : 625

مقدار تابع هدف: -1020. 558

خطای آموزش: 0. 009163

در اینجا، می‌بینیم که مدل از 625 نقطه داده مجموعه آموزشی به‌عنوان بردار پشتیبان (66 درصد از مجموعه آموزشی) استفاده کرده است.

kernlab پیاده‌سازی مدل RVM برای رگرسیون در تابع rvm دارد. نحو بسیار شبیه به مثال نشان داده شده برای ksvm است.

K -نزدیکترین همسایه ها

تابع knnreg در بسته caret با مدل رگرسیون *K* NN مطابقت دارد. آموزش مدل را روی *K کوک می‌کند* :

*> # ابتدا چند اثر انگشت پراکنده و نامتعادل را بردارید*

*> knnDescr <- solTrainXtrans[, -nearZeroVar(solTrainXtrans)]*

*> set. seed(100)*

*> knnTune <- train(knnDescr,*

|  |  |
| --- | --- |
| *+*  *+* | *solTrainY،*  *روش = "knn"،* |
| *+*  *+* | *# مرکز و مقیاس‌بندی برای پیش‌بینی‌کننده‌های جدید نیز اتفاق می‌افتد preProc = c("center", "scale"),* |
| *+*  *+* | *tuneGrid = data. frame (. k = 1:20)، trControl = trainControl (روش = "cv"))* |

هنگام پیش‌بینی نمونه‌های جدید با استفاده از این شی، نمونه‌های جدید به صورت خودکار ­در مرکز قرار می‌گیرند و با استفاده از مقادیر تعیین‌شده توسط مجموعه آموزشی، مقیاس‌بندی می‌شوند.

تمرینات

شبیه‌سازی یک پیش‌بینی منفرد و یک رابطه غیرخطی، مانند موج سینوسی که در شکل 1 نشان داده شده است.  [7. 7 ،](#bookmark360) و رابطه بین پارامترهای هزینه، *e و* هسته را برای مدل ماشین بردار پشتیبان بررسی کنید:

*set. seed (100)*

*x <- runif (100، حداقل = 2، حداکثر = 10)*

*y <- sin(x) + rnorm(length(x)) \* 0. 25*

*sinData <- data. frame(x = x,y=y)*

*نمودار (x, y)*

*## یک شبکه از مقادیر x ایجاد کنید تا از آن برای پیش‌بینی استفاده کنید*

*dataGrid <- data. frame(x = seq(2, 10, length = 100))*

(الف) مدل‌های مختلف را با استفاده از تابع پایه شعاعی و مقادیر مختلف هزینه ( پارامتر C ) و *e.* منحنی متناسب را رسم کنید. مثلا:

*> کتابخانه (kernlab)*

*rbfSVM <- ksvm(x = x,y=y, data = sinData,*

*+ هسته = "rbfdot"، kpar = "automatic"،*

*+ C = 1، اپسیلون = 0. 1)*

*modelPrediction <- پیش‌بینی (rbfSVM، newdata = dataGrid)*

*## این یک ماتریس با یک ستون است. ما می‌توانیم طرح ریزی کنیم*

*پیش‌بینی مدل ## با اضافه کردن امتیاز به نمودار قبلی*

*نقاط (x = dataGrid$x، y = modelPrediction[,1]، + type = "l"، col = "آبی")*

*## پارامترهای دیگر را امتحان کنید*

(ب) پارامتر *a* را می‌توان با استفاده از آرگومان kpar تنظیم کرد، مانند kpar = list (sigma = 1). مقادیر مختلف *a را امتحان کنید* تا بفهمید چگونه این پارامتر برازش مدل را تغییر می‌دهد. هزینه، *e* و مقادیر *a چگونه* بر مدل تأثیر می‌گذارند؟

فریدمن [( 1991](#bookmark1016) ) چندین مجموعه داده معیار ایجاد شده توسط شبیه‌سازی را معرفی کرد. یکی از این شبیه سازی‌ها از معادله غیرخطی زیر برای ایجاد داده استفاده می‌کند:

*y* = 10 sin( *nx* 1 *x* 2 ) + 20 ( *x* 3 *—* 0. 5) *2* + 10 *x 4* + 5 *x* 5 + *N (*0 *, a* 2 )

که در آن مقادیر *x* متغیرهای تصادفی هستند که به‌طور یکنواخت بین [0، 1] توزیع شده اند (همچنین 5 متغیر غیر اطلاعاتی دیگر نیز در شبیه‌سازی ایجاد شده اند ­). بسته mlbench حاوی تابعی به نام mlbench. friedman1 است که این داده‌ها را شبیه‌سازی می‌کند:

*> کتابخانه (mlbench)*

*> set. seed(200)*

*trainingData <- mlbench. friedman1 (200، sd = 1)*

*## داده‌های* ' *x* ' *را از یک ماتریس به یک قاب داده تبدیل می‌کنیم*

*## یک دلیل این است که نام ستون‌ها را می‌دهد.*

*trainingData$x <- data. frame(trainingData$x)*

*> ## با استفاده از داده‌ها نگاه کنید*

*> featurePlot (trainingData$x، trainingData$y)*

*> ## یا روش‌های دیگر.*

*>*

*> ## این یک لیست با بردار* ' *y* ' *و یک ماتریس ایجاد می‌کند*

*> ## پیش‌بینی* ' *x* '. *همچنین یک مجموعه تست بزرگ را شبیه‌سازی کنید*

*> ## نرخ خطای واقعی را با دقت خوب تخمین بزنید:*

*> testData <- mlbench. friedman1 (5000، sd = 1)*

*> testData$x <- data. frame(testData$x)*

*>*

چندین مدل را روی این داده‌ها تنظیم کنید. مثلا:

*> کتابخانه (کارت)*

*> knnModel <- train(x = trainingData$x,*

*+ y = trainingData$y،*

*+ روش = "knn"،*

*+ preProc = c ("مرکز"، "مقیاس")،*

*+ آهنگ طول = 10)*

*> knnModel*

200 نمونه

10 پیش‌بینی

پیش پردازش: متمرکز، مقیاس شده

نمونه برداری مجدد: بوت استرپ (25 تکرار)

خلاصه حجم نمونه: 200، 200، 200، 200، 200، 200،

نمونه برداری مجدد از نتایج در پارامترهای تنظیم:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ک | RMSE | Rsquared | RMSE SD | Rsquared SD |
| 5 | 3. 51 | 0. 496 | 0. 238 | 0. 0641 |
| 7 | 3. 36 | 0. 536 | 0. 24 | 0. 0617 |
| 9 | 3. 3 | 0. 559 | 0. 251 | 0. 0546 |
| 11 | 3. 24 | 0. 586 | 0. 252 | 0. 0501 |
| 13 | 3. 2 | 0. 61 | 0. 234 | 0. 0465 |
| 15 | 3. 19 | 0. 623 | 0. 264 | 0. 0496 |
| 17 | 3. 19 | 0. 63 | 0. 286 | 0. 0528 |
| 19 | 3. 18 | 0. 643 | 0. 274 | 0. 048 |
| 21 | 3. 2 | 0. 646 | 0. 269 | 0. 0464 |
| 23 | 3. 2 | 0. 652 | 0. 267 | 0. 0465 |

برای انتخاب مدل بهینه با استفاده از کوچکترین مقدار از RMSE استفاده شد. مقدار نهایی استفاده شده برای مدل k = 19 بود.

*> knnPred <- پیش‌بینی (knnModel، newdata = testData$x)*

*> ## تابع* " *postResample* " *را می‌توان برای دریافت مجموعه تست استفاده کرد*

*> ## مقادیر عملکرد*

*> postResample (pred = knnPred، obs = testData$y)*

RMSE Rsquared

3. 2286834 0. 6871735

به نظر می‌رسد کدام مدل‌ها بهترین عملکرد را دارند؟ آیا MARS پیش‌بینی‌کننده‌های آموزنده (آنهایی که X1 - X5 نام دارند ) را انتخاب می‌کند؟

برای داده‌های Tecator که در فصل آخر توضیح داده شد، مدل‌های SVM، شبکه عصبی، MARS و *K* NN بسازید. از آنجایی که شبکه‌های عصبی به ویژه ­به پیش‌بینی‌کننده‌های همبسته بسیار حساس هستند، آیا پیش پردازش با استفاده از PCA به مدل کمک می‌کند؟

به مسأله نفوذپذیری که در تمرین مشخص شده است بازگردید [6. 2](#bookmark336) . چندین مدل رگرسیون غیرخطی را آموزش دهید ­و نمونه‌برداری مجدد و عملکرد مجموعه تست را ارزیابی کنید.

کدام مدل رگرسیون غیرخطی، نمونه‌گیری مجدد و عملکرد مجموعه آزمون را بهینه می‌دهد؟

آیا هر یک از مدل‌های غیرخطی بهتر از مدل خطی بهینه‌ای است که قبلاً در Exercise ایجاد کرده اید [6. 2 ؟](#bookmark336) اگر چنین است، چه چیزی می‌تواند در مورد رابطه اساسی بین پیش‌بینی‌کننده‌ها و پاسخ به شما بگوید؟

آیا هر یک از مدل‌هایی را که ایجاد کرده اید برای جایگزینی آزمایش آزمایشگاهی نفوذپذیری توصیه می‌کنید؟

ورزش [6. 3](#bookmark336) داده‌های یک فرآیند تولید شیمیایی را توصیف می‌کند. از همان مراحل انباشت داده، تقسیم داده‌ها و مراحل پیش پردازش مانند قبل استفاده کنید و چندین مدل رگرسیون غیرخطی را آموزش دهید.

کدام مدل رگرسیون غیرخطی، نمونه‌گیری مجدد و عملکرد مجموعه آزمون را بهینه می‌دهد؟

کدام پیش‌بینی‌کننده‌ها در مدل رگرسیون غیرخطی بهینه مهم هستند ­؟ آیا متغیرهای بیولوژیکی یا فرآیندی بر فهرست غالب هستند؟ چگونه ده پیش‌بینی مهم برتر با ده پیش‌بینی برتر مدل خطی بهینه مقایسه می‌شود؟

روابط بین پیش‌بینی‌کننده‌های برتر و پاسخ پیش‌بینی‌کننده‌هایی را که منحصر به مدل رگرسیون غیرخطی بهینه هستند، بررسی کنید. آیا این نمودارها شهودی را درباره پیش‌بینی‌کننده‌های بیولوژیکی یا فرآیندی و رابطه آن‌ها با عملکرد نشان می‌دهند؟

فصل 8

درختان رگرسیون و مدل‌های مبتنی بر قانون

مدل‌های مبتنی بر درخت شامل یک یا چند عبارت if-then تو در تو برای پیش‌بینی‌کننده‌هایی هستند که داده‌ها را تقسیم‌بندی می‌کنند. در این پارتیشن‌ها، یک مدل برای پیش‌بینی نتیجه استفاده می‌شود. به‌عنوان مثال، یک درخت بسیار ساده را می‌توان به صورت تعریف کرد

اگر پیش‌بینی A >= 1. 7 باشد

| اگر پیش‌بینی B>= 202. 1، نتیجه = 1. 3 است

| دیگری نتیجه = 5. 6

دیگری نتیجه = 2. 5

در این حالت، فضای پیش‌بینی دوبعدی به سه ناحیه تقسیم می‌شود و در هر ناحیه، نتیجه با یک عدد واحد (یا 1. 3، 2. 5 یا 5. 6) پیش‌بینی می‌شود. شکل [8. 1](#bookmark389) این قوانین را در فضای پیش‌بینی ارائه می‌دهد.

در اصطلاح مدل‌های درختی، دو *تقسیم* از داده‌ها به سه *گره پایانی* یا *برگ* درخت وجود دارد. برای به دست آوردن یک پیش‌بینی برای یک نمونه جدید، عبارات if-then تعریف شده توسط درخت را با استفاده از مقادیر پیش‌بینی‌کننده‌های آن نمونه دنبال می‌کنیم تا زمانی که به یک گره پایانی برسیم. سپس فرمول مدل در گره پایانه برای تولید پیش‌بینی استفاده می‌شود. در تصویر بالا، مدل یک مقدار عددی ساده است. در موارد دیگر، گره پایانه ممکن است توسط یک تابع پیچیده تر از پیش‌بینی‌کننده‌ها تعریف شود. درختان برای رگرسیون در فرقه‌ها مورد بحث قرار خواهند گرفت.  [8. 1](#bookmark392) و [8. 2](#bookmark402) .

توجه داشته باشید که عبارات if-then تولید شده توسط یک درخت یک مسیر منحصر به فرد به یک گره ترمینال را برای هر نمونه تعریف می‌کند. یک *قاعده* مجموعه‌ای از شرایط if-then ( ­احتمالاً توسط یک درخت ایجاد شده) است که در شرایط مستقل فرو ریخته شده است. برای مثال بالا، سه قانون وجود دارد:

اگر پیش‌بینی A >= 1. 7 و پیش‌بینی B >= 202. 1، نتیجه = 1. 3

اگر پیش‌بینی A >= 1. 7 و پیش‌بینی B <202. 1، نتیجه = 5. 6

اگر پیش‌بینی A < 1. 7، نتیجه = 2. 5 است

قوانین را می‌توان به گونه‌ای ساده یا هرس کرد که نمونه‌ها توسط قوانین متعدد پوشش داده شوند. این رویکرد می‌تواند مزایایی نسبت به ­مدل‌های ساده مبتنی بر درخت داشته باشد. مدل‌های مبتنی بر قانون در بخش‌ها مورد بحث قرار خواهند گرفت.  [8. 3](#bookmark408) و [8. 7](#bookmark432) .

\_8،

©

250

200

150

100

50

2.5

1.3

5.6

0. 5 1. 0 1. 5 2. 0 2. 5

پیش‌بینی A

شکل 8. 1: نمونه‌ای از مقادیر پیش‌بینی شده در مناطق تعریف شده توسط یک مدل مبتنی بر درخت

مدل‌های مبتنی بر درخت و مبتنی بر قانون به دلایل متعددی از ابزارهای مدل‌سازی محبوب هستند. اول، آنها مجموعه‌ای از شرایط را ایجاد می‌کنند که بسیار قابل ­تفسیر هستند و به راحتی قابل اجرا هستند. به دلیل منطق ساخت آنها، آنها می‌توانند به‌طور موثر انواع بسیاری از پیش‌بینی‌کننده‌ها (پراکنده، اریب، ­پیوسته، طبقه‌بندی و غیره) را بدون نیاز به پیش پردازش آنها مدیریت کنند. علاوه بر این، این مدل‌ها نیازی ندارند که کاربر شکل رابطه پیش‌بینی‌کننده‌ها را ­با پاسخ مشخص کند، مثلاً یک مدل رگرسیون خطی نیاز دارد. علاوه بر این، این مدل‌ها می‌توانند به‌طور موثر داده‌های از دست رفته را مدیریت کنند و به‌طور ضمنی انتخاب ویژگی را انجام دهند، ویژگی‌هایی که برای بسیاری از مسائل مدل‌سازی واقعی مطلوب هستند.

با این حال، مدل‌های مبتنی بر تک درخت یا قوانین، دارای نقاط ضعف خاصی هستند. دو نقطه ضعف شناخته شده عبارتند از (1) بی ثباتی مدل (یعنی تغییرات جزئی در داده‌ها می‌تواند ساختار درخت یا قوانین و در نتیجه تفسیر را به شدت تغییر دهد) و (2) عملکرد پیش‌بینی کمتر از بهینه. مورد دوم به این دلیل است که این مدل‌ها مناطق مستطیلی را تعریف می‌کنند که حاوی مقادیر نتیجه همگن بیشتری هستند. اگر رابطه بین پیش‌بینی‌کننده‌ها و پاسخ را نتوان به‌اندازه کافی توسط فضاهای فرعی مستطیلی پیش‌بینی‌کننده‌ها تعریف ­کرد، آن‌گاه مدل‌های مبتنی بر درخت یا مبتنی بر قانون نسبت به سایر مدل‌ها خطای پیش‌بینی بزرگ‌تری خواهند داشت.

برای مبارزه با این مسائل، محققان روش‌های مجموعه‌ای را توسعه دادند که بسیاری از درختان (یا مدل‌های مبتنی بر قانون) را در یک مدل ترکیب می‌کند. گروه‌ها نسبت به تک درختان عملکرد پیش‌بینی بسیار بهتری دارند (و این به‌طور کلی برای مدل‌های مبتنی بر قانون نیز صادق است). گروه‌ها در فرقه‌ها مورد بحث قرار خواهند گرفت.  [8. 4](#bookmark411) [- 8. 7](#bookmark432) .

درختان رگرسیون پایه

درختان رگرسیون پایه داده‌ها را به گروه‌های کوچکتر تقسیم می‌کنند که از نظر پاسخ همگن تر هستند. برای دستیابی به همگنی نتیجه، درختان رگرسیون تعیین می‌کنند:

* پیش‌بینی تقسیم بر و ارزش تقسیم
* عمق یا پیچیدگی درخت
* معادله پیش‌بینی در گره‌های پایانی

در این بخش، ما بر روی تکنیک‌هایی تمرکز می‌کنیم که در آن مدل در گره‌های ترمینال ثابت ساده است.

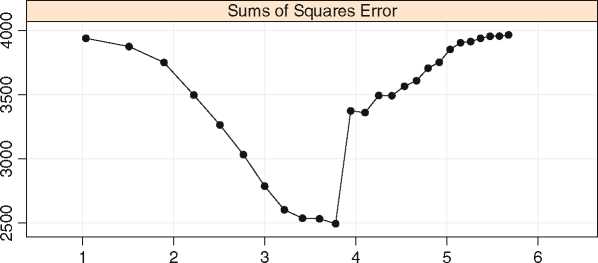
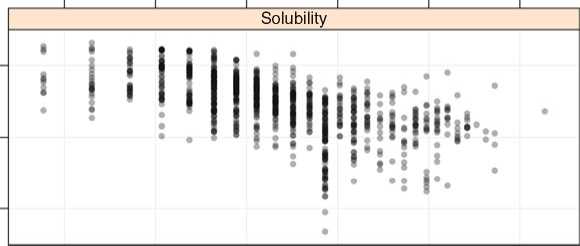
تکنیک‌های زیادی برای ساخت درختان رگرسیون وجود دارد. یکی از قدیمی‌ترین و پرکاربردترین روش‌های طبقه‌بندی و درخت رگرسیون (CART) است [بریمن و همکاران ( 1984](#bookmark1012) ). برای رگرسیون، مدل با کل مجموعه داده، *S شروع* می‌شود و هر مقدار متمایز از هر پیش‌بینی را جستجو می‌کند تا پیش‌بینی و مقدار تقسیمی را پیدا کند که داده‌ها را به دو گروه ( *S* 1 و *S* 2 ) تقسیم می‌کند، به‌طوری که مجموع مجذورات کلی خطا به حداقل می‌رسد:

SSE = £ ( *y i - y i ) 2* + £ ( *y i - y 2 ) 2 ,*  (8. 1)

i E S 1 i E S 2

که در آن *y* y *1* و *y* y *2* به ترتیب میانگین نتایج مجموعه آموزشی در گروه *S 1* و *S 2* هستند. سپس در هر یک از گروه‌های *S 1* و *S 2* , این روش برای پیش‌بینی و مقدار تقسیم که به بهترین وجه *SSE را کاهش می‌دهد جستجو می‌*کند. به دلیل ماهیت تقسیم بازگشتی درختان رگرسیون، این روش به‌عنوان پارتیشن‌بندی بازگشتی نیز شناخته می‌شود.

با بازگشت به داده‌های حلالیت، شکل.  [8. 2](#bookmark393) *SSE* را برای زنجیره شکاف‌ها برای تعداد اتم‌های کربن (در مقیاس تبدیل شده) نشان می‌دهد. با استفاده از رویکرد درخت رگرسیون، نقطه تقسیم بهینه برای این متغیر 3. 78 است. کاهش *SSE* مرتبط با این تقسیم با مقادیر بهینه برای همه پیش‌بینی‌کننده‌های دیگر مقایسه می‌شود و تقسیم مربوط به ­حداقل خطای مطلق برای تشکیل زیرمجموعه‌های *S 1* و *S 2* استفاده می‌شود. پس از در نظر گرفتن همه متغیرهای دیگر، این متغیر به‌عنوان بهترین انتخاب شد (شکل 2 را ببینید).  [8. 3 )](#bookmark394) . اگر فرآیند در این نقطه متوقف شود، تمام نمونه‌هایی که مقادیر این پیش‌بینی کمتر از 3. 78 دارند *-* 1 پیش‌بینی می‌شوند. 84 (میانگین نتایج حلالیت برای این نمونه ها) و نمونه‌های بالای شکاف‌ها همگی مقدار پیش‌بینی شده *4-* دارند. 49:



IO

I

o

o

V

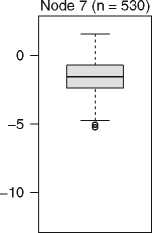
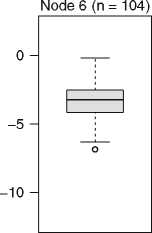
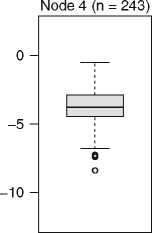
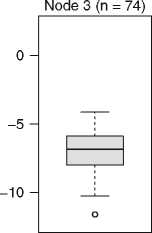
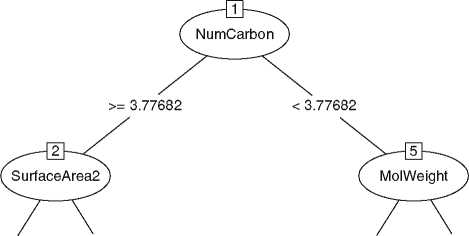
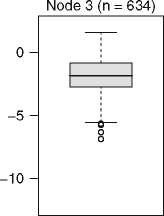
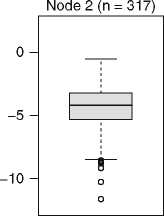
Number of Carbon Atoms (transformed)

Fig. 8.2: *Top*: A scatter plot of the solubility values (*y*-axis) versus the number of carbon atoms (on a transformed scale). *Bottom*: The SSE profile across all possible splits for this predictor. The splits used here are the midpoints between two distinct data points

اگر تعداد اتم‌های کربن >= 3. 78 باشد، حلالیت = 4. 49- و غیره حلالیت = 1. 84-

در عمل، این فرآیند در مجموعه‌های *S* 1 و *S* 2 ادامه می‌یابد تا زمانی که تعداد نمونه‌ها در تقسیم‌بندی‌ها به زیر آستانه (مانند 20 نمونه) برسد. این *مرحله رشد درخت را به پایان می‌رساند.* شکل 8. 4 مجموعه دوم تقسیمات را برای داده‌های مثال نشان می‌دهد.

هنگامی که پیش‌بینی پیوسته است، فرآیند یافتن ­نقطه تقسیم بهینه ساده است زیرا داده‌ها را می‌توان به روشی طبیعی مرتب کرد. پیش‌بینی‌کننده‌های باینری نیز به راحتی تقسیم می‌شوند، زیرا تنها یک نقطه تقسیم ممکن وجود دارد. با این حال، زمانی که یک پیش‌بینی بیش از دو دسته داشته باشد، فرآیند یافتن نقطه تقسیم بهینه می‌تواند چند مسیر قابل توجیه را طی کند. برای بحث مفصل در مورد این موضوع، بخش.  [14. 1](#bookmark686) .



NumCarbon

>= 3.77682

< 3.77682

>= 0.9777

< 0.9777

>= 5.37282

< 5.37282

شکل 8. 3: *بالا* : تقسیم اولیه داده‌های حلالیت. *پایین* : پس از تقسیم اول، دو گروه بیشتر به چهار پارتیشن تقسیم می‌شوند

هنگامی که درخت کامل رشد کرد، درخت ممکن است بسیار بزرگ باشد و احتمالاً بیش از حد به مجموعه آموزشی برازش دارد. سپس درخت به عمق بالقوه کمتری *هرس می‌شود.* پردازش شده توسط [بریمن و همکاران ( 1984](#bookmark1012) ) *تنظیم هزینه-پیچیدگی است.*

هدف این فرآیند یافتن درختی با اندازه مناسب است که کمترین میزان خطا را داشته باشد. برای انجام این کار، نرخ خطا را با استفاده از اندازه درخت جریمه می‌کنیم:

SSE *c p =* SSE + *c p x (*# گره ترمینال)،

که در آن *c p پارامتر پیچیدگی* نامیده می‌شود. برای مقدار خاصی از پارامتر پیچیدگی، کوچکترین درخت هرس شده را پیدا می‌کنیم که کمترین میزان خطای جریمه را دارد.  [بریمن و همکاران ( 1984](#bookmark1012) ) تئوری و الگوریتم‌هایی را برای یافتن بهترین درخت برای مقدار خاصی از *c p* نشان می‌دهد. مانند سایر روش‌های منظم‌سازی ­که قبلاً مورد بحث قرار گرفت، جریمه‌های کوچک‌تر تمایل به تولید مدل‌های پیچیده‌تری دارند که در این مورد، درختان بزرگ‌تر را به همراه دارند. مقادیر بزرگتر پارامتر پیچیدگی ممکن است منجر به درختی با یک شکاف (یعنی یک کنده) یا حتی درختی بدون شکاف شود. نتیجه اخیر نشان می‌دهد که هیچ پیش‌بینی‌ای به اندازه کافی تغییرات در نتیجه را در مقدار انتخابی پارامتر پیچیدگی توضیح نمی‌دهد.

برای یافتن بهترین درخت هرس شده، داده‌ها را در یک دنباله از مقادیر cp ارزیابی می‌*کنیم.* این فرآیند برای هر مقدار cp انتخابی یک *SSE* تولید می‌*کند.* اما می‌دانیم که اگر نمونه متفاوتی از مشاهدات را انتخاب کنیم، این مقادیر *SSE متفاوت خواهند بود.* برای درک تغییرات *SSE* s در هر *c* [*p*](#bookmark1012) ارزش، بریمن و همکاران. [( 1984](#bookmark1012) ) استفاده از رویکرد اعتبارسنجی متقابل مشابه روش مورد بحث در فصل را پیشنهاد می‌کند.  [4](#bookmark192) . آنها همچنین استفاده از *قانون خطای یک استاندارد را* در معیارهای بهینه‌سازی برای شناسایی ساده‌ترین درخت پیشنهاد می‌کنند: کوچک‌ترین درختی را پیدا کنید که در یک خطای استاندارد درخت با کوچک‌ترین خطای مطلق قرار دارد (به بخش مراجعه کنید.  [4. 6 ،](#bookmark6) صفحه [74 )](#bookmark6) . رویکرد دیگر اندازه درخت مرتبط با کوچکترین خطای عددی را انتخاب می‌کند [( Hastie et al. 2008](#bookmark1018) ).

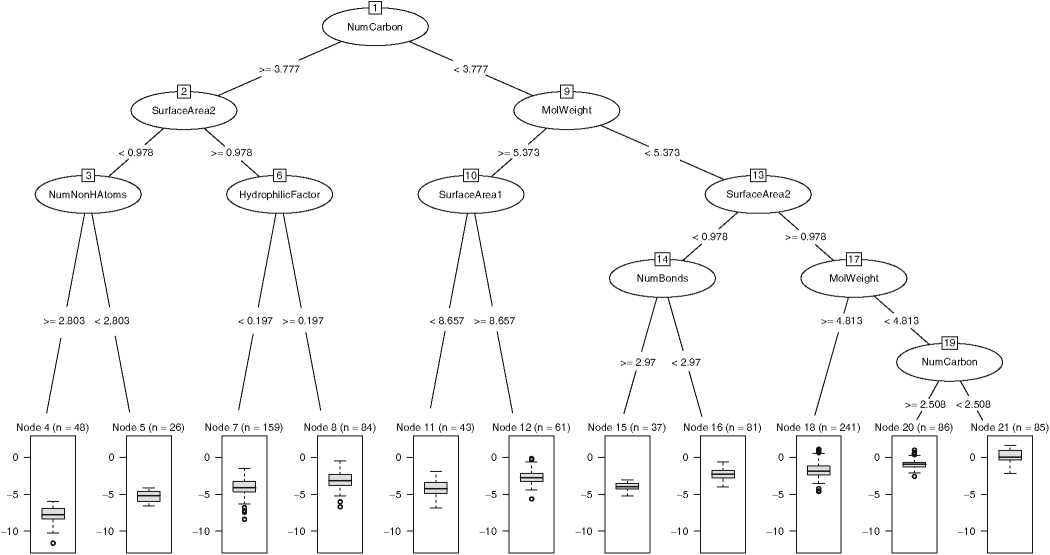
با استفاده از قانون یک خطای استاندارد، درخت رگرسیون ساخته شده بر روی داده‌های حلالیت دارای 11 گره پایانی ( *cp* = *0. 01 )* و برآورد اعتبار متقابل RMSE 1. 05 بود. شکل 8. 4 درخت رگرسیون مدل را نشان می‌دهد. تمام شکاف‌های حفظ‌شده در مدل شامل پیش‌بینی‌کننده‌های پیوسته یا شمارش می‌شوند و چندین مسیر از طریق درخت از برخی از پیش‌بینی‌کننده‌های یکسان بیش از یک بار استفاده می‌کنند.

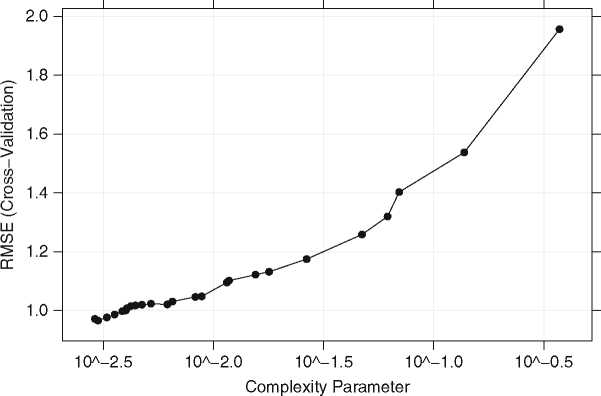
در ظاهر، درخت در شکل 8. 4 کاملاً قابل تفسیر به نظر می‌رسد. به‌عنوان مثال، می‌توان گفت که اگر ترکیبی دارای تعداد متوسط تعداد اتم‌های کربن باشد، سطح بسیار کمتری داشته باشد و تعداد اتم‌های غیر هیدروژن زیادی داشته باشد، پس کمترین انحلال پذیری را دارد. با این حال، پارتیشن‌های زیادی در داده‌ها وجود دارد که همپوشانی دارند. به‌عنوان مثال، گره‌های 12 و 16 تقریباً توزیع مقادیر حلالیت یکسانی دارند، اگرچه یکی از این مسیرها دارای مساحت سطح کم و دیگری دارای مساحت سطح بالایی است.

متناوبا، مدل را می‌توان با انتخاب مقدار ­پارامتر پیچیدگی مرتبط با کوچکترین مقدار ممکن RMSE تنظیم کرد. نمایه اعتبارسنجی متقاطع در شکل نشان داده شده است.  [8. 5 .](#bookmark396) در این مورد، فرآیند تنظیم یک درخت بزرگتر با مقدار *cp 0. 003* و 25 گره پایانه را انتخاب کرد. RMSE برآورد شده از این مدل 0. 97 بود. ­اگرچه این مدل با داده‌ها مطابقت بیشتری دارد، اما بسیار عمیق تر از درخت نشان داده شده در شکل است. 8. 4. تفسیر این مدل بسیار دشوارتر خواهد بود.

شکل 8. 4: مدل نهایی CART برای داده‌های حلالیت. نمودارهای *جعبه* در گره‌های پایانی توزیع مقادیر حلالیت مجموعه آموزشی را نشان می‌دهد. پیش‌بینی نهایی بر اساس میانگین نمونه‌ها در نمودارهای جعبه است

8. 1 درختان رگرسیون پایه 179



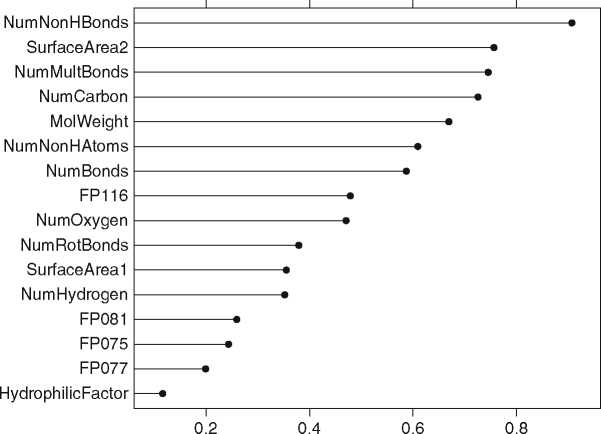


شکل 8. 5: نمایه RMSE معتبر متقابل برای درخت رگرسیون

این روش درختی خاص همچنین می‌تواند داده‌های از دست رفته را مدیریت کند. هنگام ساخت درخت، داده‌های از دست رفته نادیده گرفته می‌شوند. برای هر تقسیم، انواع ­بومی‌های جایگزین (به نام *تقسیم‌های جایگزین* ) ارزیابی می‌شوند. تقسیم جایگزین، تقسیمی است که نتایج آن شبیه به تقسیم اصلی است که در واقع در درخت استفاده می‌شود. اگر یک تقسیم جایگزین تقریبی به تقسیم اصلی باشد، می‌توان از آن در زمانی استفاده کرد که داده‌های پیش‌بینی مرتبط با تقسیم اصلی در دسترس نباشد. در عمل، چندین شکاف جایگزین ممکن است برای هر شکاف خاصی در درخت ذخیره شود.

هنگامی که درخت نهایی شد، شروع به ارزیابی اهمیت نسبی پیش‌بینی‌کننده‌ها برای نتیجه می‌کنیم. یکی از راه‌های محاسبه یک مقیاس کلی اهمیت، پیگیری کاهش کلی در معیارهای بهینه‌سازی برای هر پیش‌بینی است [( Breiman et al. 1984](#bookmark1012) ). اگر *SSE* معیار بهینه‌سازی باشد، کاهش *SSE* برای مجموعه آموزشی برای هر پیش‌بینی جمع می‌شود. به‌طور شهودی، پیش‌بینی‌کننده‌هایی که در درخت بالاتر ظاهر می‌شوند (یعنی تقسیم‌های زودتر) یا آن‌هایی که چندین بار در درخت ظاهر می‌شوند، مهم‌تر از پیش‌بینی‌کننده‌هایی هستند که در پایین‌تر درخت یا اصلاً رخ نمی‌دهند. شکل [8. 6](#bookmark397) مقادیر اهمیت 16 پیش‌بینی را در مدل حلالیت نهایی پیچیده تر نشان می‌دهد.

مزیت مدل‌های مبتنی بر درخت این است که وقتی درخت بزرگ نیست، مدل ساده و قابل تفسیر است. همچنین، این نوع درخت را می‌توان به سرعت محاسبه کرد (علی رغم استفاده از جستجوهای جامع متعدد). مدل‌های درختی ذاتاً انتخاب ویژگی را انجام می‌دهند. اگر یک پیش‌بینی هرگز در یک تقسیم استفاده نشود، معادله پیش‌بینی مستقل از این داده‌ها است. این مزیت زمانی تضعیف می‌شود که پیش‌بینی‌کننده‌های همبستگی بالایی وجود داشته باشد. اگر دو پیش‌بینی بسیار همبسته باشند، انتخاب کدام یک برای استفاده در یک تقسیم تا حدودی تصادفی است. به‌عنوان مثال، دو پیش‌بینی سطح دارای همبستگی بسیار بالایی هستند (0. 96) و هر کدام



اهمیت

شکل 8. 6: امتیازهای اهمیت متغیر برای 16 پیش‌بینی مورد استفاده در ­مدل درخت رگرسیون برای حلالیت در درخت نشان داده شده در شکل 8. 4 استفاده شده است. ممکن است تفاوت اندک بین این پیش‌بینی‌کننده‌ها به شدت باعث انتخاب بین این دو شود، اما به احتمال زیاد به دلیل تفاوت‌های کوچک و تصادفی در متغیرها باشد. به همین دلیل، ممکن است پیش‌بینی‌کننده‌های بیشتری از آنچه واقعاً مورد نیاز است انتخاب شوند. علاوه بر این، مقادیر اهمیت متغیر تحت تأثیر قرار می‌گیرند. اگر داده‌های حلالیت فقط حاوی یکی از پیش‌بینی‌کننده‌های سطح سطح باشد، احتمالاً این پیش‌بینی دو بار در درخت استفاده می‌شود، بنابراین ارزش اهمیت آن افزایش می‌یابد. در عوض، گنجاندن هر دو پیش‌بینی سطح در داده‌ها باعث می‌شود اهمیت آنها فقط مقادیر متوسطی داشته باشد.

در حالی که درختان بسیار قابل تفسیر هستند و محاسبه آنها آسان است، اما دارای معایب قابل‌توجهی هستند. اول، درختان رگرسیون منفرد در مقایسه با سایر روش‌های مدل‌سازی به احتمال زیاد عملکرد پیش‌بینی کمتر از بهینه دارند. این تا حدودی به دلیل سادگی مدل است. با ساخت، مدل‌های درختی داده‌ها را به مناطق مستطیلی فضای پیش‌بینی تقسیم می‌کنند. اگر رابطه بین پیش‌بینی‌کننده‌ها و نتیجه به اندازه کافی توسط این مستطیل‌ها توصیف نشود، عملکرد پیش‌بینی یک درخت بهینه نخواهد بود. همچنین، تعداد نتایج پیش‌بینی‌شده ممکن از یک درخت محدود است و با تعداد گره‌های پایانی تعیین می‌شود. برای ­داده‌های حلالیت، درخت بهینه دارای 11 گره پایانی است و در نتیجه تنها می‌تواند 11 مقدار پیش‌بینی شده ممکن را تولید کند. بعید است که این محدودیت بتواند تمام تفاوت‌های ظریف داده‌ها را در بر بگیرد. به‌عنوان مثال، در شکل. 8. 4، گره 21 مربوط به بالاترین پیش‌بینی حلالیت است. با این حال، توجه داشته باشید که داده‌های مجموعه آموزشی که در این مسیر درخت قرار می‌گیرند در چندین واحد ورود به سامانه داده متفاوت است. اگر نقاط داده جدید با داده‌های آموزشی سازگار باشد، بسیاری از نمونه‌های جدید که در این مسیر قرار می‌گیرند با درجه بالایی از دقت پیش‌بینی نمی‌شوند. دو مدل درخت رگرسیون نشان داده شده تا کنون دارای مقادیر RMSE هستند که به‌طور قابل ملاحظه‌ای بزرگتر از RMSE تولید شده توسط مدل رگرسیون خطی ساده نشان داده شده در فصل هستند. 6.

یک عیب اضافی این است که یک درخت منفرد تمایل به *ناپایداری دارد* [نگاه کنید به [بریمن](#bookmark1012) [( 1996b](#bookmark1012) ) و [هستی و همکاران ( 2008](#bookmark1018) ، فصل 8)]. اگر داده‌ها کمی تغییر کنند، ممکن است مجموعه‌ای کاملاً متفاوت از تقسیم‌بندی پیدا شود (یعنی واریانس مدل بالا است). در حالی که این یک نقطه ضعف است، روش‌های مجموعه (که بعداً در این فصل مورد بحث قرار خواهد گرفت) از این ویژگی برای ایجاد مدل‌هایی استفاده می‌کنند که عملکرد بسیار خوبی دارند.

*بایاس انتخاب* رنج می‌برند : پیش‌بینی‌کننده‌هایی با تعداد مقادیر متمایز بالاتری نسبت به پیش‌بینی گرهای رگه‌ای بیشتر ترجیح داده می‌شوند ( [Loh](#bookmark1021) and Shih [1997](#bookmark1021) ; [کارولین و همکاران 2007](#bookmark1013) ; [لوه 2010](#bookmark1021) ).  [لوه و شیه](#bookmark1021) [( 1997](#bookmark1021) ) اشاره کرد که

خطر زمانی رخ می‌دهد که یک مجموعه داده از ترکیبی از متغیرهای اطلاعاتی و نویز تشکیل شده باشد و متغیرهای نویز نسبت به متغیرهای اطلاعاتی تقسیمات بسیار بیشتری دارند. سپس احتمال زیادی وجود دارد که متغیرهای نویز برای تقسیم گره‌های بالای درخت انتخاب شوند. هرس یا درختی با ساختار گمراه کننده ایجاد می‌کند یا اصلاً درختی ندارد.

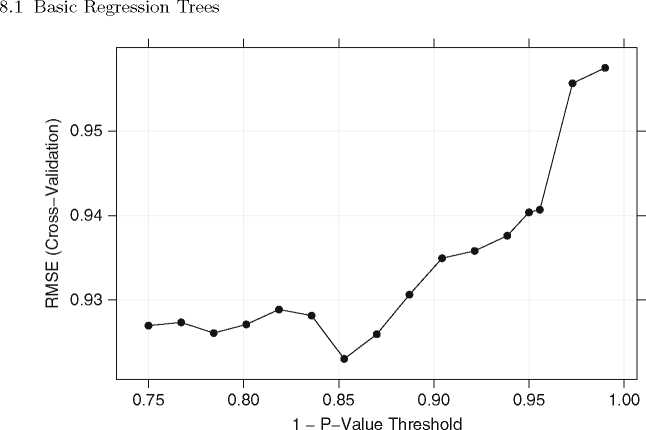
همچنین، با افزایش تعداد مقادیر از دست رفته، انتخاب پیش‌بینی‌کننده‌ها بایاس تر می‌شود [( Carolin et al. 2007](#bookmark1013) ).

شایان ذکر است که امتیازهای اهمیت متغیر برای درخت رگرسیون حلالیت (شکل 2).  [8. 6 )](#bookmark397) نشان می‌دهد که مدل تمایل بیشتری به ­پیش‌بینی‌کننده‌های پیوسته (یعنی دانه‌بندی کمتر) دارد تا اثر انگشت دودویی. این می‌تواند به دلیل بایاس انتخاب یا محتوای متغیرها باشد.

چندین تکنیک درخت رگرسیون *بی طرفانه وجود دارد.* مثلا، [لوه](#bookmark1021) [( 2002](#bookmark1021) ) الگوریتم تشخیص و تخمین تعامل تعمیم یافته، بی طرفانه (GUIDE) را پیشنهاد کرد که با جدا کردن فرآیند انتخاب متغیر تقسیم و مقدار تقسیم، مسأله را حل می‌کند. این الگوریتم با استفاده از آزمون فرضیه‌های آماری، پیش‌بینی‌کننده‌ها را رتبه‌بندی می‌کند و سپس مقدار تقسیم مناسب مرتبط با مهمترین عامل را پیدا می‌کند.

رویکرد دیگر *درختان استنتاج مشروط* است [هاتورن و همکاران ( 2006](#bookmark1018) ). آنها چارچوب یکپارچه‌ای را برای مدل‌های مبتنی بر درخت بی طرفانه برای رگرسیون، طبقه‌بندی و سناریوهای دیگر توصیف می‌کنند. در این مدل، از آزمون‌های فرضیه‌های آماری برای انجام یک جستجوی جامع در میان پیش‌بینی‌کننده‌ها و ­نقاط تقسیم احتمالی آنها استفاده می‌شود. برای یک تقسیم کاندید، یک آزمون آماری برای ارزیابی تفاوت بین میانگین دو گروه ایجاد شده توسط تقسیم استفاده می‌شود و یک *p-value* می‌تواند برای آزمون محاسبه شود.

استفاده از آماره آزمون *p-value* چندین مزیت دارد. ابتدا، پیش‌بینی‌کننده‌هایی که در مقیاس‌های متفاوت هستند را می‌توان با هم مقایسه کرد زیرا مقادیر *p* در یک مقیاس هستند. دوم، اصلاحات مقایسه چندگانه [( وست فال و یانگ](#bookmark1027) [1993](#bookmark1027) ) را می‌توان برای مقادیر *p خام* در یک پیش‌بینی اعمال کرد تا بایاس حاصل از تعداد زیادی کاندیدهای تقسیم را کاهش دهد. این اصلاحات سعی در کاهش تعداد نتایج آزمایش مثبت کاذب دارند



شکل 8. 7: نمایه RMSE معتبر متقابل برای درختان رگرسیون استنتاج شرطی

انجام تعداد زیادی آزمون فرضیه‌های آماری. بنابراین، پیش‌بینی‌کننده‌ها به‌طور فزاینده‌ای توسط روش‌های مقایسه چندگانه جریمه می‌شوند، زیرا تعداد تقسیم‌ها (و مقادیر *p- مرتبط* ) افزایش می‌یابد. به همین دلیل، بایاس برای داده‌های بسیار دانه‌ای کاهش می‌یابد. یک آستانه برای اهمیت آماری برای تعیین اینکه آیا شکاف‌های اضافی باید ایجاد شوند استفاده می‌شود [[ Hothorn et al. ( 2006](#bookmark1018) ) از یک منهای *p-value* ] استفاده کنید.

به‌طور پیش فرض، این الگوریتم از هرس استفاده نمی‌کند. از آنجایی که مجموعه داده‌ها بیشتر از ­هم جدا می‌شوند، کاهش تعداد نمونه‌ها قدرت آزمون‌های فرضیه را کاهش می‌دهد. این منجر به مقادیر *p بالاتر* و احتمال کمتری برای تقسیم جدید (و بیش برازش) می‌شود. با این حال، آزمون‌های فرضیه‌های آماری مستقیماً با عملکرد پیش‌بینی مرتبط نیستند و به همین دلیل، ­انتخاب پیچیدگی درخت بر اساس عملکرد (از طریق نمونه‌گیری مجدد یا روش‌های دیگر) هنوز قابل مشاهده است.

با آستانه معنی داری 0. 05 (یعنی 5 درصد نرخ مثبت کاذب برای اهمیت آماری)، یک درخت استنتاج مشروط برای داده‌های حلالیت دارای 32 گره پایانی بود. این درخت بسیار بزرگتر از درخت رگرسیون پایه نشان داده شده در شکل 8. 4 است. ما همچنین آستانه اهمیت را به‌عنوان یک پارامتر تنظیم در نظر گرفتیم و 16 مقدار را بین 0. 75 و 0. 99 ارزیابی کردیم (شکل 2 را ببینید).  [8. 7](#bookmark399) برای نمایه اعتبارسنجی متقابل). اندازه درخت مرتبط با کوچکترین خطا دارای 36 گره پایانی بود (با استفاده از آستانه 0. 853). تنظیم آستانه، RMSE برآورد شده را به مقدار 0. 92 در مقایسه با RMSE 0. 94 مرتبط با آستانه معناداری 0. 05 بهبود داد.

8. 2 درختان مدل رگرسیون

یکی از محدودیت‌های درختان رگرسیون ساده این است که هر گره پایانی از میانگین نتایج مجموعه آموزشی در آن گره برای پیش‌بینی استفاده می‌کند. در نتیجه، این مدل‌ها ممکن است در پیش‌بینی نمونه‌هایی که نتایج واقعی آن‌ها بسیار زیاد یا پایین است، کار خوبی انجام ندهند. در فصل 5، شکل.  [5. 1](#bookmark267) نمودار نمونه‌ای از نتایج مشاهده شده و پیش‌بینی شده را برای یک مجموعه داده نشان داد. در این شکل، مدل تمایل دارد نمونه‌ها را در هر یک از موارد افراطی پیش‌بینی کند. پیش‌بینی‌کننده‌های ­مورد استفاده در این شکل با استفاده از تکنیک مجموعه درختی رگرسیون به نام جنگل‌های تصادفی (که در ادامه این فصل توضیح داده شد) تولید شده‌اند که همچنین از میانگین داده‌های آموزشی در گره‌های پایانه استفاده می‌کند و از همان مسأله رنج می‌برد، اگرچه نه به شدت با یک درخت

یک رویکرد برای مقابله با این مسئله استفاده از تخمینگر متفاوت در گره‌های ترمینال است. در اینجا ما بر روی رویکرد *درخت مدل توضیح داده شده در* [Quinlan تمرکز می‌کنیم](#bookmark1023) [( 1992](#bookmark1023) ) M5 نامیده می‌شود که شبیه درختان رگرسیون است به جز:

معیار تقسیم متفاوت است.

گره‌های پایانه با استفاده از یک مدل خطی (برخلاف میانگین ساده) نتیجه را پیش‌بینی می‌کنند.

هنگامی که یک نمونه پیش‌بینی می‌شود، اغلب ترکیبی از پیش‌بینی‌کننده‌های مدل‌های مختلف در طول یک مسیر از طریق درخت است.

اجرای اصلی این تکنیک "بازسازی منطقی" این مدل به نام M5 است که توسط [وانگ و ویتن](#bookmark1027) [( 1997](#bookmark1027) ) و در بسته نرم‌افزاری Weka گنجانده شده است. روش‌های دیگری نیز برای درختان با مدل‌هایی در برگ‌ها وجود دارد، مانند [لوه](#bookmark1021) [( 2002](#bookmark1021) ) و [زایلیس و همکاران ( 2008](#bookmark1028) ).

مانند درختان رگرسیون ساده، تقسیم اولیه با استفاده از جستجوی جامع بر روی پیش‌بینی‌کننده‌ها و نمونه‌های مجموعه آموزشی یافت می‌شود، اما برخلاف آن مدل‌ها، از کاهش مورد انتظار در نرخ خطای گره استفاده می‌شود. اجازه دهید *S* کل مجموعه داده‌ها را نشان دهد و اجازه دهید *S* 1 *,. . . ,S P* نشان دهنده زیر مجموعه‌های P داده‌ها پس از تقسیم باشد. معیار تقسیم خواهد بود

پ

کاهش = SD ( *S ) -* ^ *n i x* SD ( *S i* )، (8. 2)

i = 1

که در آن SD انحراف معیار و *n i* تعداد نمونه‌ها در بخش ­*i* است. این معیار تعیین می‌کند که آیا کل تغییرات تقسیم‌بندی‌ها، وزن‌دهی شده بر اساس اندازه نمونه، کمتر از داده‌های از پیش تقسیم شده است. این طرح مشابه روش شناسی درختان طبقه‌بندی است که در آن بحث شده است [کوینلان](#bookmark1024) [( 1993b](#bookmark1024) ). تقسیمی که با بیشترین کاهش خطا همراه است انتخاب می‌شود و یک مدل خطی در داخل پارتیشن‌ها با استفاده از متغیر تقسیم در مدل ایجاد می‌شود. برای تکرارهای تقسیم بعدی، این فرآیند تکرار می‌شود: یک تقسیم اولیه تعیین می‌شود و یک مدل خطی برای پارتیشن با استفاده از متغیر تقسیم فعلی و سایر متغیرهای قبل از آن ایجاد می‌شود. خطای مرتبط با هر مدل خطی به جای SD( *S* )inEq استفاده می‌شود. [8. 2](#bookmark402) برای تعیین کاهش مورد انتظار در نرخ خطا برای تقسیم بعدی. روند رشد درخت در ­امتداد شاخه‌های درخت ادامه می‌یابد تا زمانی که بهبود بیشتری در میزان خطا وجود نداشته باشد یا نمونه‌های کافی برای ادامه روند وجود نداشته باشد. هنگامی که درخت به‌طور کامل رشد کرد، یک مدل خطی برای هر گره در درخت وجود دارد.

شکل [8. 8](#bookmark404) نمونه‌ای از یک درخت مدل را با چهار شکاف و هشت مدل رگرسیون خطی نشان می‌دهد. به‌عنوان مثال، مدل 5 با استفاده از همه پیش‌بینی‌کننده‌هایی که در تقسیم‌بندی‌های 1-3 قرار داشتند و با نقاط داده مجموعه آموزشی که شرایط 1 *a،* *2b و 3b را برآورده می‌کنند، ایجاد می‌شود.*

هنگامی که مجموعه کاملی از مدل‌های خطی ایجاد شد، هر کدام تحت یک روش ساده‌سازی قرار می‌گیرند تا به‌طور بالقوه برخی از اصطلاحات حذف شوند. برای یک مدل معین، یک نرخ خطای تنظیم شده محاسبه می‌شود. ابتدا، تفاوت مطلق بین داده‌های مشاهده شده و پیش‌بینی‌شده محاسبه می‌شود و سپس در عبارتی ضرب می‌شود که مدل‌هایی با تعداد زیادی پارامتر را جریمه می‌کند:

Adjusted Error Rate =

*n\** + *p*

*n\**

(8.3)

i=1

که در آن *n \** تعداد نقاط داده مجموعه آموزشی است که برای ساخت مدل استفاده شده است و *p* تعداد پارامترها است. هر عبارت مدل حذف می‌شود و میزان خطای تنظیم شده محاسبه می‌شود. تا زمانی که نرخ خطای تعدیل شده کاهش یابد، اصطلاحات از مدل حذف می‌شوند. در برخی موارد، مدل خطی ممکن است به داشتن یک عرض از مبدأ ساده شود. این روش به‌طور مستقل برای هر مدل خطی اعمال می‌شود.

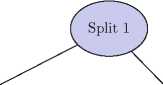
درختان مدل همچنین نوعی *صاف کردن* را برای کاهش پتانسیل بیش برازش دارند. این تکنیک بر اساس روش "کوچک شدن بازگشتی" ­است [هاستی و پرگیبون](#bookmark1018) [( 1990](#bookmark1018) ). هنگام پیش‌بینی، نمونه جدید مسیر مناسب درخت را پایین می‌آورد و با حرکت از پایین به بالا، مدل‌های خطی در طول آن مسیر با هم ترکیب می‌شوند. با استفاده از شکل [8. 8](#bookmark404) به‌عنوان یک مرجع، فرض کنید یک نمونه جدید مسیر مرتبط با مدل 5 را پایین می‌آورد. درخت با استفاده از مدل 5 و همچنین مدل خطی در گره والد (مدل 3 در این مورد) یک پیش‌بینی برای این نمونه ایجاد می‌کند. این دو پیش‌بینی با استفاده از ترکیب می‌شوند

n ( k ) y ( k ) + cy ( p )   
n ( k ) + c

*y( p)* =

که در آن *y (* *k )* پیش‌بینی گره فرزند (مدل 5)، *n (* *k )* تعداد نقاط داده مجموعه آموزشی در گره فرزند، *y ( p )* پیش‌بینی از گره والد است و *c* است ثابت با مقدار پیش فرض 15. هنگامی که این پیش‌بینی ترکیبی محاسبه شد، به‌طور مشابه با مدل بعدی در امتداد درخت (مدل 1) و غیره ترکیب می‌شود. برای مثال ما، نمونه جدیدی که تحت شرایط 1 *a،* 2 *b* و 3 *b* قرار می‌گیرد، از ترکیبی از سه مدل خطی استفاده می‌کند. توجه داشته باشید که معادله هموارسازی یک ترکیب خطی نسبتا ساده از مدل‌ها است.

این نوع هموارسازی می‌تواند تأثیر مثبت قابل‌توجهی بر درخت مدل داشته باشد، زمانی که مدل‌های خطی در سراسر گره‌ها بسیار متفاوت هستند. چندین دلیل احتمالی وجود دارد که مدل‌های خطی ممکن است پیش‌بینی‌کننده‌های بسیار متفاوتی تولید کنند.



**شرط 1** *الف*  **شرط 1** *ب*

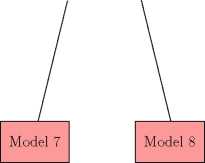
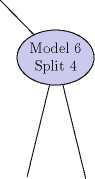


Fig. 8.8: An example of a regression model tree

**Condition 4***a* **Condition 4***b*

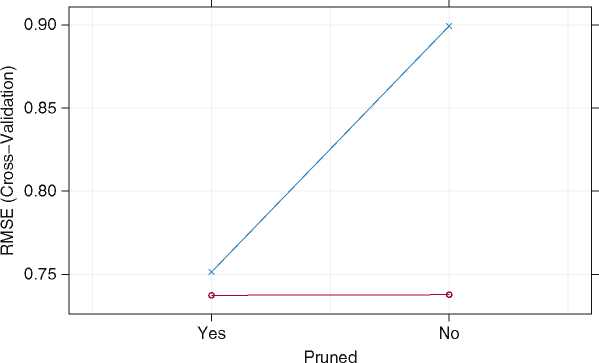
یون‌ها اولاً، تعداد نمونه‌های مجموعه آموزشی که در یک گره در دسترس هستند، با اضافه شدن تقسیم‌های جدید کاهش می‌یابد. این می‌تواند منجر به گره‌هایی شود که مناطق بسیار متفاوتی از مجموعه آموزشی را مدل می‌کنند و بنابراین، مدل‌های خطی بسیار متفاوتی را تولید می‌کنند. این به ویژه برای مجموعه‌های آموزشی کوچک صادق است. ثانیا، مدل‌های خطی به دست آمده توسط فرآیند تقسیم ممکن است از همخطی قابل‌توجهی رنج ببرند. فرض کنید دو پیش‌بینی در مجموعه آموزشی همبستگی بسیار بالایی با یکدیگر دارند. در این مورد، الگوریتم ممکن است بین دو پیش‌بینی به‌طور تصادفی انتخاب کند. اگر هر دو پیش‌بینی در نهایت در تقسیم‌بندی‌ها استفاده شوند و کاندیدای مدل‌های خطی شوند، دو عبارت در مدل خطی برای یک قطعه اطلاعات وجود دارد. همانطور که در فصل‌های قبلی بحث شد، این می‌تواند منجر به بی‌ثباتی قابل‌توجهی در ­کارایی مدل شود. هموارسازی با استفاده از چندین مدل می‌تواند به کاهش تاثیر هر مدل خطی ناپایدار کمک کند.

درختان فرعی ناکافی و حذف آنها دوباره هرس می‌شود. ­با شروع از گره‌های ترمینال، نرخ خطای تنظیم شده با و بدون درخت فرعی محاسبه می‌شود. اگر درخت فرعی نرخ خطای تنظیم شده را کاهش ندهد، از مدل هرس می‌شود. این روند تا زمانی ادامه می‌یابد که دیگر درختان فرعی حذف نشوند.

درختان مدل بر روی داده‌های حلالیت در شرایط با و بدون هرس و با و بدون هموارسازی ساخته شدند. شکل [8. 9](#bookmark10) نموداری از پروفایل‌های اعتبارسنجی متقابل برای این داده‌ها را نشان می‌دهد. درخت هرس نشده دارای 159 مسیر از طریق درخت است که ممکن است بیش از حد با داده‌های آموزشی مطابقت داشته باشد. وقتی که

صاف شده

بله o نه x -



شکل 8. 9: پروفایل‌های RMSE تایید شده متقابل برای درخت مدل

درخت هرس نمی‌شود، صاف کردن مدل به‌طور قابل‌توجهی میزان خطا را بهبود می‌بخشد. برای این داده‌ها، تأثیر هرس بر روی مدل نیز قابل‌توجه بود: تعداد مسیرهای درخت از 159 به 18 کاهش یافت. از هرس و صاف کردن استفاده کرد.

درخت مدل به دست آمده (شکل 8. 10 ) نشان می‌دهد که بسیاری از شکاف‌ها شامل پیش‌بینی‌کننده‌های یکسانی مانند تعداد کربن هستند. همچنین، برای این داده‌ها، تقسیم‌بندی‌ها به جای اثر انگشت، به نفع پیش‌بینی‌کننده‌های پیوسته هستند. [[16]](#footnote-16) برای این داده‌ها، تقسیم بر اساس *SSE* و کاهش نرخ خطا نتایج تقریباً یکسانی ایجاد می‌کند. جزئیات مدل‌های خطی در شکل نشان داده شده است.  [8. 11](#bookmark405) (ضرایب مدل در یک مقیاس نرمال شده است). از این شکل می‌توان فهمید که اکثر مدل‌ها از پیش‌بینی‌کننده‌های زیادی از جمله تعداد زیادی اثر انگشت استفاده می‌کنند. با این حال، ضرایب اثر انگشت نسبت به پیش‌بینی‌کننده‌های پیوسته کوچک است.

علاوه بر این، این مدل می‌تواند برای نشان دادن مسائل مربوط به هم خطی ­استفاده شود. در شکل، مدل خطی 5 (در سمت چپ پایین درخت) با شرایط زیر همراه است:

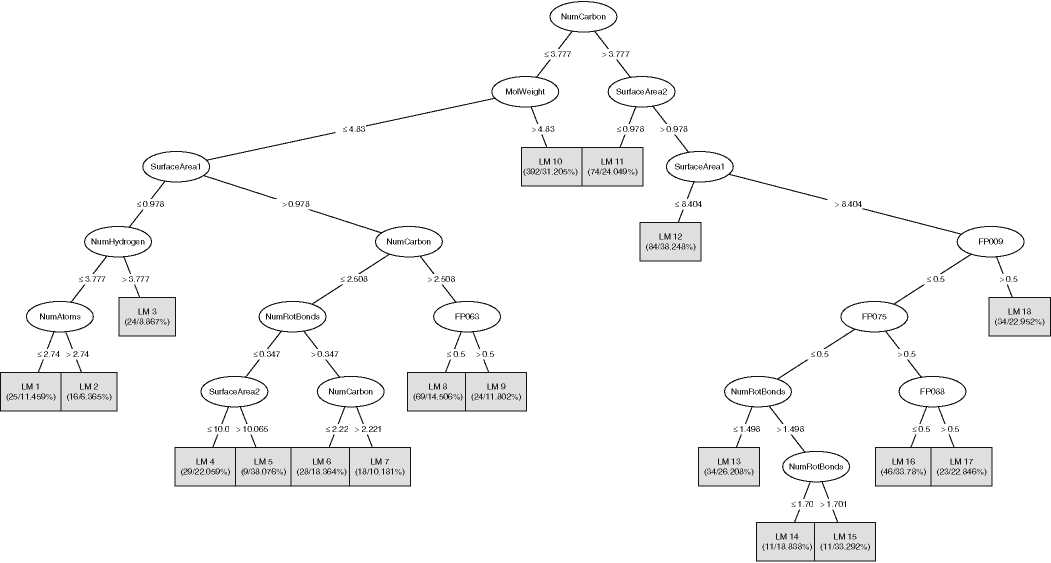
NumCarbon <= 3. 777 &

MolWeight <= 4. 83 & Surface Area1 > 0. 978 &

NumCarbon <= 2. 508 &

شکل 8. 10: درخت مدل نهایی برای داده‌های حلالیت. اعداد در پایین هر گره پایانه تعداد نمونه‌ها و درصد پوشش گره را نشان می‌دهد

188 8 درختان رگرسیون و مدل‌های مبتنی بر قانون



FP003 FP004 FP006 FP007 FP008 FP009 FP010 FP011 FP012 FP013 FP017 FP028 FP029 FP030 FP031

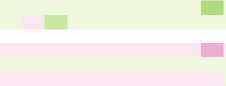
FP049 FP053 FP057 FP061 FP063 FP069 FP070 FP071 FP073 FP074 FP075 FP077 FP078 FP082 FP085 FP088 FP090 FP094 FP095 FP103 FP106 FP107 FP108 FP115 FP116 FP117 FP124 FP125 FP134 FP137 FP138 FP140 FP142 FP145 FP148 FP150 FP168 FP169 FP178

HydrophilicFactor Molweight NumAtoms NumBonds NumCarbon NumDblBonds NumHalogen NumHydrogen NumMultBonds NumNonHAtoms NumNonHBonds NumOxygen NumRotBonds Surface Area1 Surface Area2

01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18

1

2



مدل #

شکل 8. 11: ضرایب مدل خطی برای درخت مدل که در شکل 8. 10 مشاهده می‌شود. ضرایب نرمال شده اند تا در یک مقیاس باشند. بلوک‌های *سفید* نشان می‌دهد که پیش‌بینی در مدل خطی دخالت نداشته است

NumRotBonds <= 0. 347 & SurfaceArea2 > 10. 065

پس از کاهش و هموارسازی مدل، 57 ضریب در ­مدل خطی sponding مربوطه، شامل هر دو پیش‌بینی سطح سطح وجود داشت. در مجموعه آموزشی، این دو پیش‌بینی همبستگی بالایی دارند (0. 96). در نتیجه انتظار داریم که هم خطی شدید باشد. دو ضریب مقیاس‌بندی شده برای این پیش‌بینی‌کننده‌ها ­کاملاً متضاد هستند: 0. 9 برای SurfaceArea1 و *-* 0. 8 برای SurfaceArea2. از آنجایی که دو پیش‌بینی تقریباً یکسان هستند، یک تناقض وجود دارد: در ­افزایش سطح به همان اندازه حلالیت افزایش و کاهش می‌یابد. بسیاری از مدل‌هایی که در شکل نشان داده شده اند.  [8. 11](#bookmark405) دارای علائم متضاد برای این دو متغیر است. با وجود این، عملکرد این مدل نسبتاً رقابتی است. هموارسازی مدل‌ها باعث به حداقل رساندن مسائل خطی می‌شود. حذف پیش‌بینی‌کننده‌های همبسته، مدلی تولید می‌کند که ­سازگاری کمتری دارد و قابل تفسیرتر است. با این حال، با استفاده از استراتژی، افت عملکرد قابل اندازه‌گیری وجود دارد.

مدل‌های مبتنی بر قانون

یک قانون به‌عنوان یک مسیر متمایز از طریق یک درخت تعریف می‌شود. درخت مدل نشان داده شده در بخش آخر و مسیر رسیدن به مدل خطی 15 را در سمت راست پایین شکل 8. 10 در نظر بگیرید :

NumCarbon > 3. 777 &

SurfaceArea2 > 0. 978 &

SurfaceArea1 > 8. 404 &

FP009 <= 0. 5 و

FP075 <= 0. 5 و

NumRotBonds > 1. 498 &

NumRotBonds > 1. 701

برای درخت مدل نشان داده شده در شکل 8. 10، در مجموع 18 قانون وجود دارد. برای درخت، یک نمونه جدید فقط می‌تواند در یک مسیر واحد از درخت تعریف شده توسط این قوانین حرکت کند. تعداد نمونه‌هایی که تحت تأثیر یک قانون قرار می‌گیرند، *پوشش* آن نامیده می‌شود.

علاوه بر الگوریتم‌های هرس توضیح داده شده در بخش آخر، پیچیدگی درخت مدل را می‌توان با حذف کل قوانین یا حذف برخی از شرایطی که قانون را تعریف می‌کنند، کاهش داد. در قانون قبلی توجه داشته باشید که تعداد پیوندهای قابل چرخش دو بار استفاده می‌شود. این به این دلیل رخ داد که مسیر دیگری از طریق درخت مشخص کرد که مدل‌سازی زیرمجموعه داده‌ها که در آن تعداد پیوندهای قابل چرخش بین 1. 498 و 1. 701 است مهم است. با این حال، هنگامی که به صورت مجزا مشاهده می‌شود، قانون فوق به دلیل این افزونگی غیر ضروری پیچیده است. همچنین، حذف سایر شرایط در قانون ممکن است سودمند باشد زیرا آنها کمک زیادی به مدل نمی‌کنند.

Quinlan [( 1993b](#bookmark1024) ) روش‌هایی را برای ساده‌سازی قوانین تولید شده ­از درختان طبقه‌بندی توصیف می‌کند. تکنیک‌های مشابهی را می‌توان برای درختان مدل به کار برد تا مجموعه‌ای ساده تر از قوانین را از درخت مدل اولیه ایجاد کند. این رویکرد خاص بعداً در این فصل در زمینه مدل‌های کوبیسم توضیح داده می‌شود (بخش.  [8. 7 )](#bookmark432) .

روش دیگری برای ایجاد قوانین از درختان مدل در هلمز و همکاران بیان [شده](#bookmark1018) است. [( 1993](#bookmark1018) ) که از استراتژی "جدا کن و تسخیر" استفاده می‌کند. این رویه به جای اینکه از یک درخت منفرد قوانین را از بسیاری از درختان مدل‌های مختلف استخراج کند. ابتدا یک درخت مدل اولیه ایجاد می‌شود (آنها توصیه می‌کنند از درختان مدل صاف نشده استفاده کنید). با این حال، تنها قانون با بیشترین پوشش از این مدل نجات یافته است. نمونه‌های تحت پوشش قانون از مجموعه آموزشی حذف می‌شوند و درخت مدل دیگری با داده‌های باقی مانده ایجاد می‌شود. باز هم، فقط قانون با حداکثر پوشش حفظ می‌شود. این فرآیند تا زمانی تکرار می‌شود که تمام داده‌های مجموعه آموزشی حداقل توسط یک قانون پوشش داده شوند. یک نمونه جدید با تعیین اینکه تحت کدام قانون (ها) قرار می‌گیرد، پیش‌بینی می‌شود و سپس مدل خطی مرتبط با بیشترین پوشش را اعمال می‌کند.

برای داده‌های حلالیت، یک مدل مبتنی بر قانون مورد ارزیابی قرار گرفت. مشابه فرآیند تنظیم درخت مدل، چهار مدل با استفاده از تمام ترکیبات برای هرس و صاف کردن مناسب بودند. از همان مجموعه داده‌های نمونه‌گیری مجدد در تحلیل درخت مدل استفاده شد، بنابراین می‌توان مقایسه مستقیم انجام داد. شکل [8. 12](#bookmark411) نتایج این فرآیند را نشان می‌دهد. پنل سمت راست مانند شکل [8. 9](#bookmark10) در حالی که پانل سمت چپ نتایج را هنگامی که درختان مدل به قوانین تبدیل می‌شوند نشان می‌دهد. برای این داده‌ها، زمانی که هموارسازی و هرس استفاده می‌شود، نسخه درخت مدل و قانون ­مبتنی بر نرخ خطای معادل داشت. همانند درختان مدل، هرس تأثیر زیادی بر روی مدل و صاف کردن تأثیر بیشتری بر روی مدل‌های هرس نشده داشت.

بهترین درخت مدل برازش با RMSE اعتبارسنجی متقابل 0. 737 همراه بود. بهترین مدل مبتنی بر قانون به مقدار RMSE 0. 741 منجر شد.

بر اساس این به تنهایی، درخت مدل برای پیش‌بینی استفاده می‌شود. با این حال، برای مثال، مدل مبتنی بر قانون با جزئیات بیشتری مورد بررسی قرار خواهد گرفت.

در مجموع، نه قانون برای مدل‌سازی این داده‌ها استفاده شد، اگرچه قانون نهایی هیچ شرایط مرتبطی ندارد. شرایط قوانین است

قانون 1: NumCarbon <= 3. 777 و MolWeight > 4. 83

قانون 2: NumCarbon > 2. 999

قانون 3: SurfaceArea1 > 0. 978 و NumCarbon > 2. 508 و NumRotBonds > 0. 896

قانون 4: Surface Area1 > 0. 978 & MolWeight <= 4. 612 & FP063 <= 0. 5

قانون 5: سطح 1 > 0. 978 و وزن مول <= 4. 612

قانون 6: Surface Area1 <= 4. 159 & NumHydrogen <= 3. 414

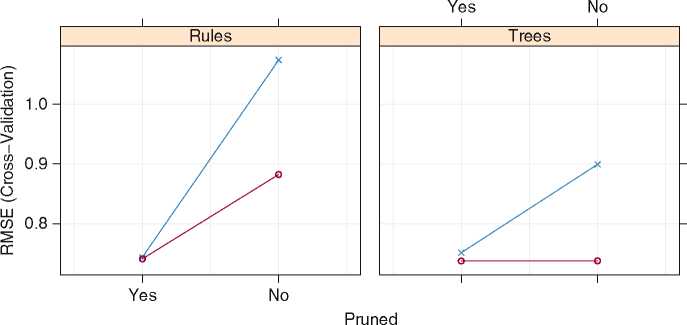
قانون 7: SurfaceArea1 > 2. 241 & FP046 <= 0. 5 & NumBonds > 2. 74

قانون 8: NumHydrogen <= 3. 414

با نگاهی به درخت مدل کامل در شکل 8. 10، قانون مربوط به مدل 10 بیشترین پوشش را با استفاده از شرایط NumCarbon *> 3. 77* و MolWeight *>* 4. 83 دارد. این قانون به‌عنوان اولین قانون در مدل جدید حفظ شد. درخت مدل بعدی با استفاده از نمونه‌های باقی مانده ایجاد شد. در اینجا، قانون با بیشترین پوشش دارای شرایطی مشابه قانون قبلی است: NumCarbon *>* 2. 99. در این مورد، نمونه‌ای با NumCarbon *>* 2. 99 خواهد بود

صاف شده

بله o نه x -



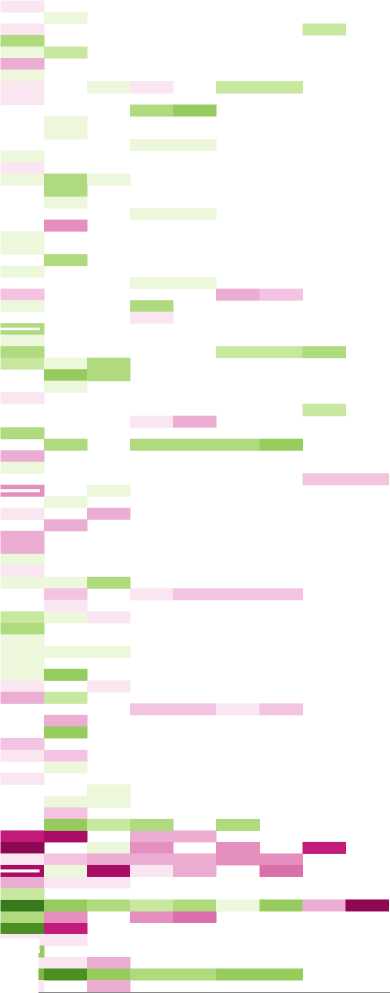
شکل 8. 12: نمایه‌های RMSE معتبر متقابل برای درختان مدل قبل و بعد از تبدیل به قوانین

تحت پوشش حداقل دو قانون. قوانین دیگر از بسیاری از پیش‌دیگرهای مشابه استفاده می‌کردند ­: SurfaceArea1 (پنج بار)، MolWeight (سه بار) و NumCarbon (همچنین سه بار). شکل [8. 13](#bookmark412) ضرایب مدل‌های خطی را برای هر قانون نشان می‌دهد (مشابه شکل 1).  [8. 11](#bookmark405) برای درخت مدل کامل). در اینجا، مدل‌های خطی پراکنده تر هستند. با ایجاد قوانین بیشتر، تعداد عبارت‌ها در مدل‌های خطی کاهش می‌یابد. این منطقی است زیرا نقاط داده کمتری برای ساخت درختان عمیق وجود دارد.

درختان کیسه ای

در دهه 1990، تکنیک‌های گروهی (روش‌هایی که پیش‌بینی‌کننده‌های بسیاری از مدل‌ها را ترکیب می‌کنند ­) شروع به ظهور کردند. بگینگ، مخفف *b* ootstrap *agg regation،* در اصل ­توسط لئو بریمن پیشنهاد شد و یکی از اولین ­تکنیک‌های یکپارچه توسعه‌یافته بود [( Breiman 1996a](#bookmark1012) ). Bagging یک رویکرد کلی است که از bootstrapping استفاده می‌کند (بخش.  [4. 4 )](#bookmark219) در ارتباط با هر گونه رگرسیون (یا طبقه بندی؛ بخش.  [14. 3 )](#bookmark699) مدل ساختن یک مجموعه. این روش در ساختار نسبتاً ساده است و از مراحل الگوریتم تشکیل شده است [8. 1 .](#bookmark413) سپس هر مدل در مجموعه برای ایجاد یک پیش‌بینی برای یک نمونه جدید استفاده می‌شود و این پیش‌بینی‌کننده‌های *m* میانگین می‌شوند تا پیش‌بینی مدل بسته‌شده را ارائه دهند.

مدل‌های کیسه دار چندین مزیت را نسبت به مدل‌هایی که کیسه‌ای ندارند ارائه می‌کنند. اولاً، بسته‌بندی به‌طور مؤثر واریانس یک پیش‌بینی را از طریق فرآیند تجمیع آن کاهش می‌دهد (به بخش مراجعه کنید).  [5. 2](#bookmark270) برای بحث در مورد مبادله بایاس-واریانس). برای مدل‌هایی که یک پیش‌بینی ناپایدار تولید می‌کنند، مانند درخت‌های رگرسیون، تجمیع در بسیاری از نسخه‌های داده‌های آموزشی در واقع کاهش می‌دهد.



FP004 - FP005 - FP006 - FP007 - FP008 -

FP010 - FP011 - FP012 - FP013 - FP019 -

FP023 - FP026 - FP027 - FP028 - FP029 -

FP031 - FP037 - FP040 -

FP046 - FP048 - FP049 - FP053 - FP054 -

FP057 - FP060 - FP061 - FP063 - FP069 -

FP070 - FP071 - FP073 - FP074 - FP075 -

FP077 - FP078 - FP079 - FP080 - FP082 -

FP084 - FP085 - FP088 - FP090 - FP094 -

FP099 - FP103 - FP105 - FP106 - FP107 -

FP108 - FP115 - FP116 - FP117 - FP123 -

FP124 - FP125 - FP137 - FP138 - FP140 -

FP142 - FP145 - FP148 - FP150 - FP158 -

FP162 - FP168 - FP169 - FP171 - FP178 -

FP198 - FP202 - FP205 -

HydrophilicFactor - MolWeight - NumAtoms - NumBonds -

NumCarbon -

NumDblBonds -

NumHalogen - NumHydrogen -

NumMultBonds - NumNonHAtoms - NumNonHBonds -

NumOxygen - NumRotBonds -

SurfaceArea1 - SurfaceArea2 -

-0

-1

-2

-3

1 23456789

قانون #

شکل 8. 13: ضرایب مدل خطی برای نسخه مبتنی بر قانون M5. همکارایی‌ها ­نرمال شده‌اند تا در مقیاس مشابه شکل 1 باشند.  [8. 11](#bookmark405) . *بلوک‌های سفید* نشان می‌دهند که پیش‌بینی در معادله رگرسیون خطی ­برای آن قانون درگیر نبوده است

برای *من = 1* تا انجام

یک نمونه بوت استرپ از داده‌های اصلی ایجاد کنید

یک مدل درخت هرس نشده را روی این نمونه آموزش دهید

پایان

الگوریتم 8. 1: چمدان

واریانس در پیش‌بینی و در نتیجه، پیش‌بینی را پایدارتر می‌کند. تصویر درختان را در شکل در نظر بگیرید.  [8. 14 .](#bookmark414) در این مثال، شش نمونه بوت استرپ از داده‌های حلالیت تولید شد و یک درخت با حداکثر عمق برای هر نمونه ساخته شد. این درختان در ساختار متفاوت هستند (شکل 1 را مقایسه کنید.  [8. 14 b](#bookmark414) , d که ساختارهای متفاوتی در سمت راست و چپ هر درخت دارند و از این رو پیش‌بینی نمونه‌ها از درختی به درخت دیگر متفاوت خواهد بود. وقتی پیش‌بینی‌کننده‌های یک نمونه در تمام درخت‌های منفرد میانگین می‌شوند، پیش‌بینی میانگین واریانس کمتری نسبت به واریانس بین پیش‌بینی‌کننده‌های فردی دارد. این بدان معناست که اگر بخواهیم دنباله‌ای متفاوت از نمونه‌های بوت استرپ تولید کنیم، یک مدل بر روی هر یک از نمونه‌های راه‌انداز بسازیم و میانگین ­سنی پیش‌بینی‌کننده‌ها در بین مدل‌ها را داشته باشیم، احتمالاً یک ­مقدار پیش‌بینی‌شده بسیار مشابه برای نمونه انتخاب‌شده دریافت می‌کنیم. مدل کیف قبلی این مشخصه همچنین عملکرد پیش‌بینی یک مدل کیسه‌دار را نسبت به مدلی که کیسه نشده است بهبود می‌بخشد. اگر هدف تلاش مدل‌سازی یافتن بهترین پیش‌بینی باشد، در این صورت کیسه‌سازی مزیت مشخصی دارد.

از سوی دیگر، مدل‌های واریانس پایین‌تر مانند رگرسیون خطی و MARS بهبود کمتری در عملکرد پیش‌بینی ارائه می‌دهند. شکل ­\_ [8. 15 ،](#bookmark415) که در آن کیسه‌بندی روی درختان، مدل‌های خطی و MARS برای داده‌های حلالیت و همچنین برای داده‌های حاصل از مطالعه مخلوط‌های بتن اعمال شده است (به فصل 13 مراجعه کنید). 10 ). برای هر مجموعه داده، عملکرد مجموعه آزمون بر اساس *RMSE* با تعداد تکرارهای بسته‌بندی رسم می‌شود. برای داده‌های حلالیت، کاهش RMSE در سراسر تکرار برای درختان، رگرسیون خطی و MARS مشابه است که یک نتیجه معمولی نیست. این نشان می‌دهد که یا پیش‌بینی‌کننده‌های مدل از رگرسیون خطی و MARS دارای ناپایداری ذاتی ­برای این داده‌ها هستند که می‌توان با استفاده از یک مجموعه کیسه‌ای بهبود یافت یا اینکه درختان در مدل‌سازی داده‌ها کمتر مؤثر هستند. نتایج بسته‌بندی برای داده‌های بتن معمولی تر است که در آن رگرسیون خطی و MARS کمترین ­را از طریق مجموعه ثابت می‌کنند، در حالی که پیش‌بینی‌کننده‌ها برای درختان رگرسیون به‌طور چشمگیری بهبود یافته است.

به‌عنوان یک نمایش بیشتر از توانایی کیسه‌بندی برای کاهش واریانس پیش‌بینی یک مدل، موج سینوسی  *شبیه‌سازی شده را در شکل 1 در نظر بگیرید.*  [5. 2 .](#bookmark273) برای این تصویر، 20 موج سینوسی شبیه‌سازی شد و برای هر مجموعه داده، درخت‌های رگرسیون و مدل‌های MARS محاسبه شد. خطوط قرمز در پانل‌ها روند واقعی را نشان می‌دهد در حالی که خطوط مشکی متعدد پیش‌بینی‌کننده‌های هر مدل را نشان می‌دهد. توجه داشته باشید که پانل CART نویز بیشتری در اطراف منحنی *گناه واقعی* نسبت به مدل MARS دارد که فقط تغییرات را در نقاط تغییر الگو نشان می‌دهد. این واریانس بالا در درخت رگرسیون ناشی از مدل را نشان می‌دهد

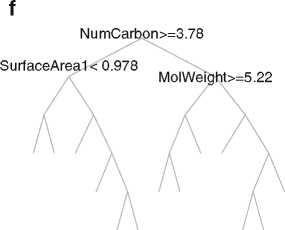
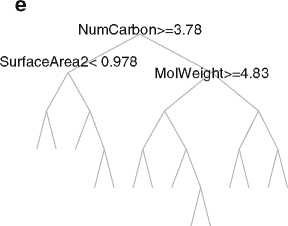
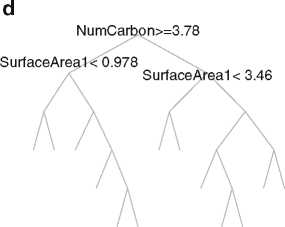
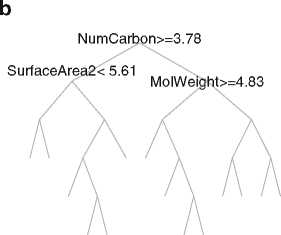
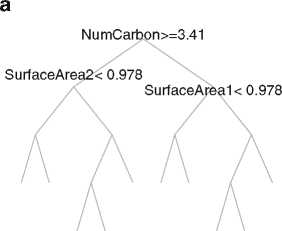
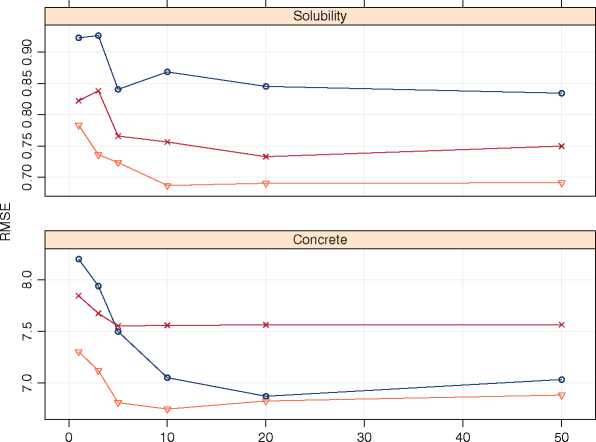


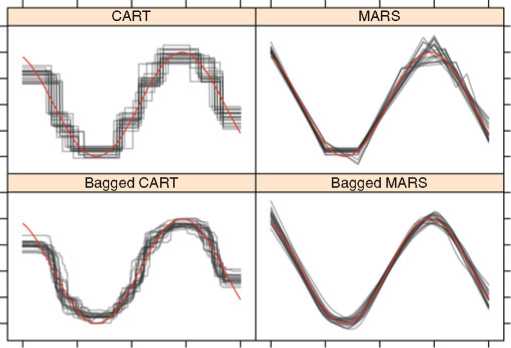
Fig. 8.14: Example of trees of maximum depth from bagging for the solubility data. Notice that the trees vary in structure, and hence the predictions will vary from tree to tree. The prediction variance for the ensemble of trees will be less than the variance of predictions from individual trees. (a) Sample 1. (b) Sample 2. (c) Sample 3. (d) Sample 4. (e) Sample 5. (f) Sample 6

بی ثباتی. پانل‌های پایین شکل، نتایج را برای 20 درخت رگرسیون کیسه‌ای و مدل‌های MARS (هر کدام با 50 تکرار مدل) نشان می‌دهد. تغییرات حول منحنی واقعی برای درختان رگرسیون تا حد زیادی کاهش می‌یابد و برای MARS، تغییرات فقط در اطراف بخش‌های منحنی روی الگو کاهش می‌یابد. با استفاده از یک مجموعه آزمایشی شبیه‌سازی شده برای هر مدل، میانگین کاهش RMSE با کیسه‌بندی درخت 8. 6 درصد بود در حالی که مدل پایدارتر MARS کاهش متناظری 2 درصدی داشت (شکل 2).  [8. 16 )](#bookmark415) .



تعداد تکرارهای بسته بندی

شکل 8. 15: مجموعه تست پروفایل‌های عملکرد RMSE برای درختان بسته بندی، مدل‌های خطی ­و MARS برای داده‌های حلالیت ( *نمودار بالا* ) و داده‌های بتن ( *نمودار پایین* ؛ برای جزئیات داده‌ها به فصل 10 مراجعه کنید) بر اساس تعداد نمونه‌های راه انداز. بسته‌بندی تقریباً همان میزان بهبود را در RMSE برای هر سه روش برای داده‌های حلالیت فراهم می‌کند که غیر معمول است. یک الگوی معمولی تر از بهبود از بسته‌بندی برای داده‌های بتن نشان داده شده است. در این صورت درختان بیشترین سود را دارند



1.0

0.5

0.0

-0.5

-1.0

246810

1.0

0.5

0.0

-0.5

-1.0

246810

Predictor

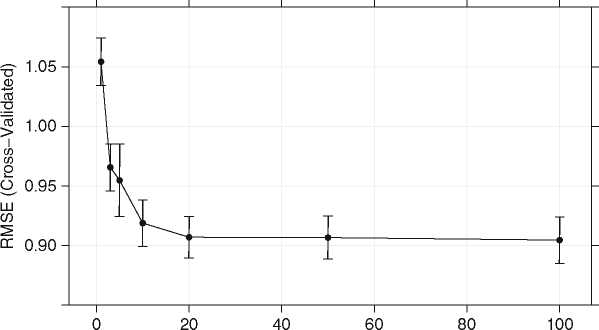
Fig. 8.16: The effect of bagging regression trees and MARS models

مزیت دیگر مدل‌های بسته‌بندی این است که می‌توانند تخمین داخلی خود را از عملکرد پیش‌بینی ارائه دهند که به خوبی با تخمین‌های اعتبارسنجی متقاطع یا برآوردهای مجموعه آزمایشی همبستگی دارد. به همین دلیل است: هنگام ساختن ­یک نمونه بوت استرپ برای هر مدل در مجموعه، نمونه‌های خاصی حذف می‌شوند. این نمونه‌ها *خارج از کیسه نامیده می‌شوند* و می‌توان از آن‌ها برای ارزیابی عملکرد پیش‌بینی آن مدل خاص استفاده کرد، زیرا برای ساخت مدل استفاده نشده است. از این رو، هر مدل در مجموعه، معیاری از عملکرد پیش‌بینی را از نمونه‌های خارج از کیسه تولید می‌کند. سپس می‌توان از میانگین معیارهای عملکرد خارج از کیسه برای سنجش ­عملکرد پیش‌بینی کل مجموعه استفاده کرد و این مقدار معمولاً با ارزیابی عملکرد پیش‌بینی که می‌توانیم با اعتبارسنجی متقاطع یا از آزمایش به دست آوریم، همبستگی خوبی دارد. تنظیم. این تخمین خطا معمولاً به‌عنوان برآورد *خارج از کیسه نامیده می‌*شود.

در شکل اولیه خود، کاربر یک انتخاب برای بسته‌بندی دارد: تعداد نمونه‌های بوت استرپ برای جمع آوری، *m.* اغلب ما شاهد کاهش تصاعدی در بهبود پیش‌بینی با افزایش تعداد تکرارها هستیم. بیشترین پیشرفت در عملکرد پیش‌بینی با تعداد کمی درخت ( *m <* 10) به دست می‌آید. برای نشان دادن این نکته، شکل 1 را در نظر بگیرید.  [8. 17](#bookmark419) که ­عملکرد پیش‌بینی ( *RM SE* ) را برای تعداد متفاوتی از نمونه‌های راه‌اندازی شده برای درختان CART نشان می‌دهد. توجه داشته باشید که عملکرد پیش‌بینی از طریق ده درخت بهبود می‌یابد و سپس با بهبود بسیار محدود فراتر از آن نقطه به پایان می‌رسد. در تجربه ما، هنوز هم می‌توان با استفاده از مجموعه‌های کیسه‌زنی تا اندازه 50، پیشرفت‌های کوچکی انجام داد. اگر عملکرد پس از 50 بار تکرار در بسته‌بندی در سطح قابل قبولی قرار نگرفت، پیشنهاد می‌کنیم سایر روش‌های پیش‌بینی قوی‌تر مانند جنگل‌های تصادفی و تقویت را امتحان کنید. بخش‌های زیر را شرح داد.

برای داده‌های حلالیت، درختان CART بدون کیسه‌بندی یک RMSE تایید متقابل بهینه 0. 97 با خطای استاندارد 0. 021 تولید می‌کنند. با گرم شدن کیسه، عملکرد بهبود می‌یابد و در RMSE 0. 9، با خطای استاندارد 0. 019 پایین می‌آید. درختان استنتاج شرطی، مانند درختان CART، نیز می‌توانند کیسه شوند. به‌عنوان مقایسه، درختان استنتاج شرطی بدون کیسه‌بندی دارای RMSE بهینه و خطای استاندارد به ترتیب 0. 93 و 0. 034 هستند. درختان استنتاج شرطی کیسه‌ای RMSE بهینه را با خطای استاندارد 0. 018 به 0. 8 کاهش می‌دهند. برای هر دو نوع مدل، بسته‌بندی عملکرد را بهبود می‌بخشد ­و واریانس برآورد را کاهش می‌دهد. در این مثال خاص، به نظر می‌رسد که درخت‌های استنتاج شرطی بسته‌بندی دارای یک لبه جزئی نسبت به درختان CART در عملکرد پیش‌بینی هستند که توسط RMSE اندازه‌گیری می‌شود. مجموعه تست *R* 2 مقادیر موازی با عملکرد RMSE تایید شده متقابل با درختان استنتاج شرطی که کمی بهتر از درختان CART (0. 85) عمل می‌کنند (0. 87).

اگرچه بسته‌بندی معمولاً عملکرد پیش‌بینی مدل‌های ناپایدار را بهبود می‌بخشد، اما چند اخطار وجود دارد. اول، هزینه‌های محاسباتی و حافظه مورد نیاز با افزایش تعداد نمونه‌های بوت استرپ افزایش می‌یابد. اگر مدل‌ساز به محاسبات موازی دسترسی داشته باشد، می‌توان این عیب را عمدتاً کاهش داد زیرا فرآیند بسته‌بندی را می‌توان به راحتی موازی کرد. به یاد داشته باشید که هر نمونه بوت استرپ و مدل مربوطه مستقل از هر مدل دیگری است



تعداد تکرارهای بسته بندی

شکل 8. 17: نمایه عملکرد تایید شده متقاطع برای بسته‌بندی درختان CART برای داده‌های حلالیت بر اساس تعداد نمونه‌های بوت استرپ. خطوط عمودی نشان دهنده *±* خطای یک استاندارد *RMSE* است. بیشترین پیشرفت در عملکرد پیش‌بینی با تجمیع ده تکرار بوت استرپ به دست می‌آید

نمونه و مدل این به این معنی است که هر مدل می‌تواند به‌طور جداگانه ساخته شود و تمام مدل‌ها می‌توانند در انتها برای تولید پیش‌بینی گرد هم آیند.

یکی دیگر از معایب این رویکرد این است که یک مدل کیسه‌ای بسیار کمتر از مدلی که کیسه نشده است قابل تفسیر است. قوانین مناسبی که می‌توانیم از یک درخت رگرسیون منفرد مانند آنچه در شکل 8. 4 نشان داده شده است بدست آوریم، قابل دستیابی نیستند. با این حال، معیارهای اهمیت متغیر را می‌توان با ترکیب معیارهای اهمیت از مدل‌های فردی در سراسر مجموعه ایجاد کرد. هنگامی که جنگل‌های تصادفی را بررسی می‌کنیم، بیشتر در مورد اهمیت متغیرها در بخش بعدی مورد بحث قرار خواهد گرفت.

8. 5 جنگل‌های تصادفی

همانطور که با داده‌های حلالیت نشان داده شده است، بسته‌بندی درختان (یا هر تکنیک واریانس بالا، بایاس کم) عملکرد پیش‌بینی را در یک درخت با ­کاهش واریانس پیش‌بینی بهبود می‌بخشد. تولید نمونه‌های راه‌انداز، یک جزء تصادفی را به فرآیند درخت‌سازی معرفی می‌کند که باعث ­توزیع درختان و بنابراین توزیع مقادیر پیش‌بینی‌شده برای هر نمونه می‌شود. با این حال، درختان کیسه‌بندی کاملاً مستقل از یکدیگر نیستند، زیرا همه پیش‌بینی‌کننده‌های اصلی در هر شکاف هر درخت در نظر گرفته می‌شوند. می‌توان تصور کرد که اگر با تعداد کافی نمونه‌های اصلی و رابطه بین پیش‌بینی‌کننده‌ها و پاسخ که می‌تواند به‌اندازه کافی توسط یک درخت مدل‌سازی شود، شروع کنیم، آن‌گاه درخت‌هایی از نمونه‌های راه‌انداز مختلف ممکن است ساختارهای مشابهی با یکدیگر داشته باشند (به ویژه در بالای درختان) به دلیل رابطه زیرین. این مشخصه به‌عنوان همبستگی درختی شناخته می‌شود و از کاهش بهینه واریانس مقادیر پیش‌بینی‌شده توسط بسته‌بندی جلوگیری می‌کند. شکل [8. 14](#bookmark414) یک تصویر مستقیم از این نام ارائه می‌دهد. علی‌رغم گرفتن نمونه‌های بوت استرپ، هر درخت بر روی تعداد اتم‌های کربن با مقدار تقریباً 3. 5 تقسیم می‌شود. شکاف‌های سطح دوم کمی بیشتر متفاوت هستند اما به هر دو پیش‌بینی سطح و وزن مولکولی محدود می‌شوند. در حالی که هر درخت در نهایت منحصر به فرد است - هیچ دو درختی دقیقاً یکسان نیستند - همه آنها با یک ساختار مشابه شروع می‌شوند و در نتیجه به یکدیگر مرتبط هستند. بنابراین، کاهش واریانس ­ارائه شده توسط بسته‌بندی می‌تواند بهبود یابد. برای توضیح ریاضی پدیده همبستگی درختی، نگاه کنید به [هستی و همکاران ( 2008](#bookmark1018) ). کاهش همبستگی بین درختان که به‌عنوان درختان همبستگی شناخته می‌شود، گام منطقی بعدی برای بهبود عملکرد کیسه‌بندی است.

از منظر آماری، کاهش همبستگی بین پیش‌بینی‌کننده‌ها را می‌توان با افزودن تصادفی به فرآیند ساخت درخت انجام داد. پس از پرده برداری از کیسه بریمن، چندین نویسنده الگوریتم را با اضافه کردن تصادفی به فرآیند یادگیری تغییر دادند. از آنجا که درختان یادگیرنده محبوبی برای بسته‌بندی بودند، [دیتریش](#bookmark1014) [( 2000](#bookmark1014) ) ایده انتخاب تقسیم تصادفی را توسعه داد که در آن درختان با استفاده از یک زیر مجموعه تصادفی از پیش‌بینی‌کننده‌های *k بالا* در هر تقسیم در درخت ساخته می‌شوند. رویکرد دیگر ساخت درختان کامل بر اساس زیرمجموعه‌های تصادفی توصیفگرها بود [( هو 1998](#bookmark1018) ؛ آمیت و جمن 1997 ).  [بریمن](#bookmark1012) [( 2000](#bookmark1012) ) همچنین سعی کرد نویز را به پاسخ اضافه کند تا ساختار درخت را مختل کند. پس از ارزیابی دقیق این تعمیم‌ها به الگوریتم کیسه‌بندی اصلی، [بریمن](#bookmark1012) [( 2001](#bookmark1012) ) یک الگوریتم یکپارچه به نام *جنگل‌های تصادفی ساخت.* یک ­الگوریتم جنگل‌های تصادفی عمومی برای یک مدل مبتنی بر درخت می‌تواند همانطور که در الگوریتم نشان داده شده است پیاده‌سازی شود.  [8. 2 .](#bookmark421)

سپس هر مدل در مجموعه برای ایجاد یک پیش‌بینی برای یک نمونه جدید استفاده می‌شود و این پیش‌بینی‌کننده‌های *m* میانگین می‌شوند تا پیش‌بینی جنگل را ارائه دهند. از آنجایی که الگوریتم به‌طور تصادفی پیش‌بینی‌کننده‌ها را در هر تقسیم انتخاب می‌کند، لزوماً همبستگی درختی کاهش می‌یابد. به‌عنوان مثال، اولین تقسیمات برای شش درخت اول در جنگل تصادفی برای داده‌های حلالیت عبارتند از NumNonHBonds، NumCarbon، NumNonHAtoms، NumCarbon، NumCarbon و NumCarbon که با درخت‌های نشان داده شده در شکل متفاوت هستند.  [8. 14](#bookmark414) .

پارامتر تنظیم جنگل‌های تصادفی تعداد پیش‌بینی‌کننده‌های تصادفی انتخاب شده، *k* است که در هر تقسیم‌بندی انتخاب می‌شود و معمولاً به‌عنوان *m try نامیده می‌شود.* در زمینه رگرسیون، [بریمن](#bookmark1012) [( 2001](#bookmark1012) ) توصیه می‌کند که *m سعی کنید* یک سوم تعداد پیش‌بینی‌کننده‌ها باشد. به منظور تنظیم پارامتر *m try،* از آنجایی که جنگل‌های تصادفی از نظر محاسباتی فشرده هستند، پیشنهاد می‌کنیم با پنج مقدار *k شروع کنید* که تا حدودی به‌طور مساوی در محدوده 2 تا *P* قرار دارند. همچنین متخصص باید تعداد درختان جنگل را مشخص کند.  [بریمن](#bookmark1012) [( 2001](#bookmark1012) ) ثابت کرد که جنگل‌های تصادفی از ­برازش بیش از حد محافظت می‌شوند. بنابراین، اگر تعداد زیادی درخت برای جنگل ساخته شود، مدل تحت تأثیر منفی قرار نخواهد گرفت. از نظر عملی، هر چه جنگل بزرگتر باشد،

1 تعداد مدل‌های ساخت را انتخاب کنید، *m*

2 برای *i = 1 tom* do

یک نمونه بوت استرپ از داده‌های اصلی ایجاد کنید

3

4

5

6

7

8

9

برای *هر تقسیم* یک مدل درختی روی این نمونه آموزش دهید

به‌طور تصادفی *k (* *<P* ) از پیش‌بینی‌کننده‌های اصلی را انتخاب کنید

بهترین پیش‌بینی را از میان *k* پیش‌بینی‌کننده‌ها انتخاب کنید و داده‌ها را تقسیم‌بندی کنید

پایان

از معیارهای توقف مدل درختی معمولی برای تعیین زمان کامل بودن درخت استفاده کنید (اما هرس نکنید)

10 پایان

الگوریتم 8. 2: جنگل‌های تصادفی اساسی

بار محاسباتی بیشتری برای آموزش و ساخت مدل متحمل خواهیم شد. به‌عنوان نقطه شروع، استفاده از حداقل 1000 درخت را پیشنهاد می‌کنیم. اگر نمایه‌های عملکرد اعتبارسنجی متقابل هنوز در 1000 درخت بهبود می‌یابند، درخت‌های بیشتری را تا زمانی که سطح عملکرد کاهش یابد، اضافه کنید.

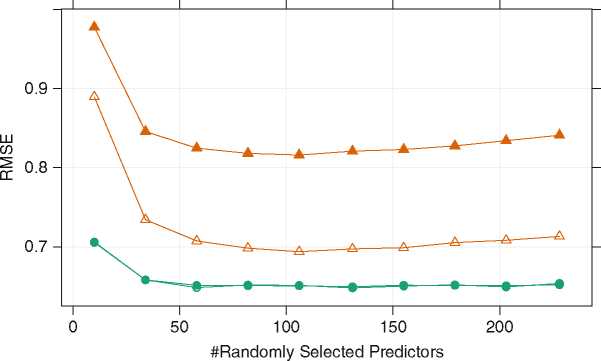
بریمن نشان داد که ترکیب خطی بسیاری از یادگیرندگان مستقل واریانس مجموعه کلی را نسبت به هر یادگیرنده فردی در گروه کاهش می‌دهد. یک مدل جنگل تصادفی این کاهش واریانس را با انتخاب یادگیرندگان قوی و پیچیده‌ای که بایاس کم نشان می‌دهند، به دست می‌آورد. این مجموعه متشکل از بسیاری از یادگیرندگان مستقل و قوی باعث بهبود نرخ خطا می‌شود. از آنجایی که هر یادگیرنده مستقل از همه یادگیرندگان قبلی انتخاب می‌شود، ­جنگل‌های تصادفی نسبت به پاسخ پر سر و صدا مقاوم هستند. ما در این بخش بیشتر توضیح می‌دهیم.  [20. 2](#bookmark923) و تصویری از تأثیر نویز بر روی جنگل‌های تصادفی و همچنین بسیاری از مدل‌های دیگر ارائه می‌دهد. در عین حال، استقلال فراگیران می‌تواند داده‌ها را در زمانی که پاسخ پر سر و صدا نباشد، کم‌رنگ کند.  [5. 1 )](#bookmark267) .

در مقایسه با کیسه‌بندی، جنگل‌های تصادفی از نظر محاسباتی بر اساس درخت به درخت کارآمدتر هستند، زیرا فرآیند درخت‌سازی تنها نیاز به ارزیابی کسری از پیش‌بینی‌کننده‌های اصلی در هر تقسیم دارد، اگرچه معمولاً جنگل‌های تصادفی به درخت‌های بیشتری نیاز دارند. ترکیب این ویژگی با قابلیت ساخت درخت به صورت موازی باعث می‌شود که جنگل‌های تصادفی از نظر محاسباتی کارآمدتر از تقویت شوند (بخش.  [8. 6 )](#bookmark425) .

مانند bagging، CART یا درختان استنتاج شرطی می‌توانند به‌عنوان یادگیرنده پایه در جنگل‌های تصادفی استفاده شوند. از هر دوی این یادگیرندگان پایه و همچنین اعتبارسنجی متقاطع و اعتبار سنجی خارج از کیسه 10 برابر برای آموزش مدل‌ها بر روی داده‌های حلالیت استفاده شد. پارامتر *m try* در ده مقدار از 10 تا 228 ارزیابی شد.  [8. 18 .](#bookmark422) برخلاف بسته بندی، درختان CART عملکرد بهتری نسبت به درختان استنتاج شرطی در تمام مقادیر پارامتر تنظیم دارند. هر کدام از پروفایل ها

سبد خرید (CV) • استنتاج شرطی (CV) ▲

سبد خرید (OOB) o استنتاج شرطی (OOB) △



شکل 8. 18: نمایه RMSE تایید متقابل برای CART و ­رویکردهای استنتاج مشروط به جنگل‌های تصادفی

یک فلات مسطح بین *m try* = 58 و *m try* = 155 را نشان می‌دهد. مدل جنگل تصادفی مبتنی بر CART بدون توجه به روش تخمین RMSE، در *m try = 131 عددی بهینه بود.* تجربه ما این است که پارامتر تنظیم تصادفی جنگل تأثیر شدیدی بر عملکرد ندارد. در این داده‌ها، تنها تفاوت واقعی در RMSE زمانی حاصل می‌شود که از کوچکترین مقدار استفاده شود (در این مورد 10). اغلب اتفاق می‌افتد که چنین مقدار کوچکی با عملکرد بهینه مرتبط نیست. با این حال، نمونه‌های نادری را دیده‌ایم که مقادیر پارامترهای تنظیم کوچک بهترین نتایج را ایجاد می‌کنند. برای ارزیابی سریع عملکرد مدل جنگل تصادفی، مقدار پارامتر تنظیم پیش‌فرض برای رگرسیون ( *m try* = *P/* 3) به خوبی کار می‌کند. اگر تمایل به به حداکثر رساندن عملکرد وجود دارد، تنظیم این مقدار ممکن است منجر به بهبود جزئی شود.

در شکل [8. 18 ،](#bookmark422) همچنین توجه داشته باشید که مدل‌های جنگل تصادفی ساخته شده با درختان CART، نتایج RMSE بسیار مشابهی با تخمین خطای خارج از کیسه و اعتبارسنجی متقاطع داشتند (در مقایسه با پارامترهای تنظیم). مشخص نیست که آیا الگوی مشاهده شده در این داده‌ها تعمیم می‌یابد، به ویژه در ­شرایط مختلف مانند حجم نمونه کوچک. استفاده از نرخ خطای خارج از کیسه زمان محاسباتی برای تنظیم مدل‌های جنگل تصادفی را به شدت کاهش می‌دهد. برای جنگل‌هایی که با استفاده از درخت‌های استنتاج مشروط ایجاد شده‌اند، خطای خارج از کیسه بسیار خوش‌بینانه‌تر از RMSE اعتبار متقابل بود. باز هم، دلیل پشت این الگو نامشخص است.

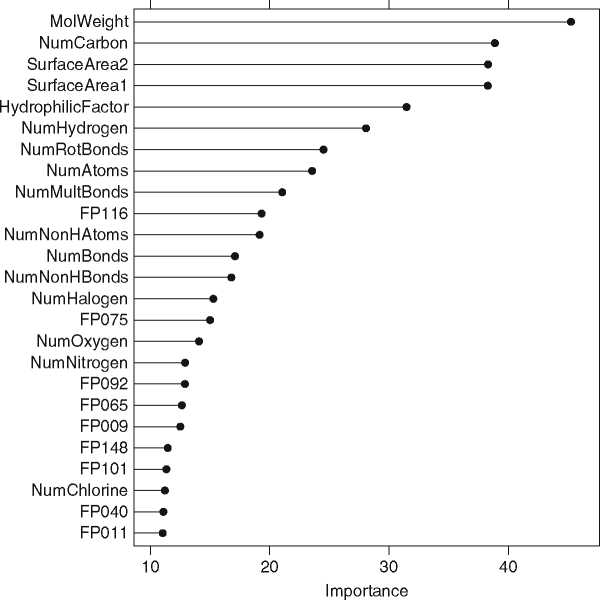
ماهیت مجموعه‌ای جنگل‌های تصادفی، دستیابی به درک رابطه بین پیش‌بینی‌کننده‌ها و پاسخ را غیرممکن می‌کند. با این حال، از آنجایی که درخت‌ها یادگیرنده پایه معمولی برای این روش هستند، می‌توان تأثیر پیش‌بینی‌کننده‌ها را در مجموعه اندازه‌گیری کرد.  [بریمن](#bookmark1012) [( 2000](#bookmark1012) ) در ابتدا پیشنهاد تغییر تصادفی مقادیر هر پیش‌بینی برای نمونه خارج از کیسه یک پیش‌بینی در یک زمان برای هر درخت را پیشنهاد کرد. تفاوت در عملکرد پیش‌بینی بین نمونه تغییر یافته و نمونه جایگزین شده برای هر پیش‌بینی ثبت و در کل جنگل جمع‌آوری می‌شود. رویکرد دیگر اندازه‌گیری بهبود خلوص گره بر اساس معیار عملکرد برای هر پیش‌بینی در هر وقوع آن پیش‌بینی در سراسر جنگل است. این مقادیر بهبود فردی برای هر پیش‌بینی سپس در سراسر جنگل جمع‌آوری می‌شوند تا اهمیت کلی برای پیش‌بینی مشخص شود.

اگرچه این استراتژی برای تعیین تأثیر نسبی یک پیش‌بینی بسیار متفاوت از رویکرد توصیف‌شده در بخش است.  [8](#bookmark388) برای درختان رگرسیون منفرد، از همان محدودیت‌های مربوط به بایاس رنج می‌برد. همچنین، [استروبل و همکاران](#bookmark1026) [( 2007](#bookmark1026) ) نشان داد که همبستگی بین پیش‌بینی‌کننده‌ها می‌تواند تأثیر قابل‌توجهی بر مقادیر اهمیت داشته باشد. برای مثال، پیش‌بینی‌کننده‌های غیر اطلاعاتی با همبستگی بالا با پیش‌بینی‌کننده‌های آموزنده، مقادیر غیرعادی اهمیت زیادی داشتند. در برخی موارد، اهمیت آنها بیشتر یا برابر با متغیرهای ضعیف بود. آنها همچنین نشان دادند که پارامتر تنظیم *m try* تأثیر جدی بر مقادیر اهمیت دارد.

تأثیر دیگر همبستگی‌های بین پیش‌بینی، کاهش ­اهمیت پیش‌بینی‌کننده‌های کلیدی است. برای مثال، فرض کنید یک پیش‌بینی بحرانی اهمیت *X* را داشته باشد. اگر پیش‌بینی دیگری به همان اندازه بحرانی باشد، اما تقریباً به‌طور کامل ­مانند اولی همبستگی داشته باشد، اهمیت این دو پیش‌بینی تقریباً *X/* 2 خواهد بود. اگر سه پیش‌بینی در مدل وجود داشته باشند، مقادیر بیشتر به *X/* 3 کاهش می‌یابد و بنابراین بر. این می‌تواند پیامدهای عمیقی برای برخی مسائل داشته باشد. به‌عنوان مثال، داده‌های پروفایل بیان RNA تمایل به اندازه‌گیری یک ژن در بسیاری از مکان‌ها دارند و در نتیجه، متغیرهای درون ژنی تمایل به همبستگی بسیار بالایی دارند. اگر این ژن برای پیش‌بینی برخی نتایج مهم بود، افزودن همه متغیرها به یک مدل جنگل تصادفی باعث می‌شد که ژن کمتر از آنچه که هست اهمیت داشته باشد.

استروبل و همکاران [( 2007](#bookmark1026) ) یک رویکرد جایگزین برای محاسبه ­اهمیت در مدل‌های جنگل تصادفی ایجاد کرد که همبستگی‌های بین پیش‌بینی را در نظر می‌گیرد. روش شناسی آنها اثر افزونگی بین پیش‌بینی را کاهش می‌دهد. برای اثر رقیق‌سازی فوق الذکر تنظیم نمی‌شود.

مقادیر اهمیت متغیر تصادفی جنگل برای 25 پیش‌بینی برتر داده‌های حلالیت در شکل 1 ارائه شده است.  [8. 19](#bookmark425) . برای این مدل، MolWeight، NumCarbon، SurfaceArea2 و SurfaceArea1 در بالای معیار اهمیت قرار می‌گیرند و مقادیر اهمیت با اثر انگشت کاهش می‌یابد. مقادیر اهمیت برای اثرانگشت 116 و 75 از نظر اهمیت، بهترین عملکردهای اثر انگشت هستند که ممکن است نشان‌دهنده این باشد که ساختارهای نشان‌داده‌شده توسط این اثرانگشت بر حلالیت یک ترکیب تأثیر دارند.



شکل 8. 19: امتیازهای اهمیت متغیر برای 25 پیش‌بینی برتر استفاده شده در مدل درختی CART جنگل تصادفی برای حلالیت

متضاد اهمیت تصادفی جنگل منجر به یک درخت سبد خرید می‌شود (شکل 2).  [8. 6 )](#bookmark397) می بینیم که 2 مورد از 4 پیش‌بینی برتر یکسان هستند ( SurfaceArea2 و NumCarbon ) و 14 مورد از 16 پیش‌بینی برتر یکسان هستند. با این حال، ترتیب اهمیت بسیار متفاوت است. به‌عنوان مثال NumNonHBonds پیش‌بینی برتر برای درخت CART است، اما در نهایت در رتبه چهاردهم برای جنگل‌های تصادفی قرار می‌گیرد. جنگل‌های تصادفی MolWeight را به‌عنوان پیش‌بینی برتر شناسایی می‌کنند، در حالی که درخت CART آن را در رتبه پنجم قرار می‌دهد. این تفاوت‌ها نباید نگران کننده باشد. بلکه تأکید می‌کنند که حریص بودن یک درخت، پیش‌بینی‌کننده‌ها را متفاوت از جنگل تصادفی اولویت می‌دهد.

8. 6 تقویت

مدل‌های تقویتی در ابتدا برای مسائل طبقه‌بندی توسعه یافتند و بعداً به تنظیمات رگرسیون گسترش یافتند. خوانندگانی که با تقویت کننده آشنا نیستند ­ممکن است با مطالعه اول در مورد تقویت برای طبقه‌بندی سود ببرند (بخش.  [14. 5 )](#bookmark37) و سپس به این بخش بازگردید. برای کامل شدن این بخش، ما تاریخچه‌ای از تقویت ارائه می‌دهیم تا پلی از توسعه اولیه ­تقویت در طبقه‌بندی به استفاده از آن در زمینه رگرسیون ارائه دهیم. این تاریخچه با الگوریتم AdaBoost شروع می‌شود و به ماشین تقویت گرادیان تصادفی فریدمن تکامل می‌یابد که اکنون به‌عنوان الگوریتم تقویتی انتخابی در بین متخصصان پذیرفته شده است.

در اوایل دهه 1990 الگوریتم‌های تقویت کننده ظاهر شدند [( Schapire 1990](#bookmark1025) ; [فروند](#bookmark1016) [1995](#bookmark1016) ; [Schapire 1999](#bookmark1025) ) که تحت تأثیر نظریه یادگیری قرار گرفتند [( Valiant 1984](#bookmark1026) ؛ [Kearns and Valiant 1989](#bookmark1019) ) که در آن تعدادی از طبقه‌بندی‌کننده‌های ضعیف (طبقه‌بندی‌کننده‌ای که تا حدی بهتر از تصادفی را پیش‌بینی می‌کند) ترکیب می‌شوند (یا تقویت می‌شوند) تا یک طبقه‌بندی گروهی با نرخ خطای طبقه‌بندی غلط تعمیم یافته برتر محققان مدتی برای یافتن یک پیاده‌سازی مؤثر ­از نظریه تقویت تلاش کردند، تا اینکه فروند و شاپیر برای تولید الگوریتم AdaBoost با یکدیگر همکاری کردند [( Schapire 1999](#bookmark1025) ). AdaBoost (به الگوریتم مراجعه کنید [14. 2 )](#bookmark711) اجرای عملی مفهوم Kerns و Valiant را برای تقویت یک یادگیرنده ضعیف به یک یادگیرنده قوی ارائه کرد ­[( Kearns and Valiant 1989](#bookmark1019) ).

تقویت، به ویژه در قالب الگوریتم AdaBoost، یک ابزار پیش‌بینی قدرتمند است که معمولاً از هر مدلی بهتر عمل می‌کند. موفقیت آن توجه جامعه مدل‌سازی را به خود جلب کرد و استفاده از آن با استفاده از [pp](#bookmark1011) در بیان ژن [( دودویت و همکاران 2002](#bookmark1015) ؛ بن‌دور و همکاران [2000](#bookmark1011) )، شیمی‌سنجی [(](#bookmark1026) وارموزا [و همکاران، 2003](#bookmark1026) ) و شناسایی ژانر موسیقی گسترده شد. ­[( Bergstra et al. 2006](#bookmark1011) )، به نام چند.

الگوریتم AdaBoost به وضوح کار کرد و پس از ورود موفقیت آمیز آن، چندین محقق [( فریدمن و همکاران 2000](#bookmark1016) ) الگوریتم AdaBoost را به مفاهیم آماری توابع ضرر، مدل‌سازی افزایشی و رگرسیون لجستیک متصل کردند ­و نشان دادند که تقویت را می‌توان به‌عنوان یک مرحله به جلو تفسیر کرد. مدل افزودنی که تلفات نمایی را به حداقل می‌رساند. این درک اساسی ­از تقویت منجر به دیدگاه جدیدی از تقویت شد که تعمیم الگوریتمی متعددی را برای مسائل طبقه‌بندی تسهیل کرد (بخش.  [14. 5 )](#bookmark37) . علاوه بر این، این دیدگاه جدید همچنین این روش را قادر می‌سازد تا به مسائل رگرسیونی نیز بسط داده شود.

توانایی فریدمن برای دیدن چارچوب آماری تقویت، یک ­الگوریتم ساده، ظریف و بسیار قابل انطباق برای انواع مختلف مسائل به دست آورد [( فریدمن 2001](#bookmark1016) ). او این روش را "ماشین‌های تقویت کننده گرادیان" نامید که هم طبقه‌بندی و هم رگرسیون را در بر می‌گرفت. اصول اولیه ­افزایش ارزش به شرح زیر است: با توجه به یک تابع ضرر (به‌عنوان مثال، مربع خطا برای رگرسیون) و یک یادگیرنده ضعیف (به‌عنوان مثال، درختان رگرسیون)، الگوریتم به دنبال یافتن یک مدل افزایشی است که تابع ضرر را به حداقل برساند. الگوریتم معمولاً با بهترین حدس پاسخ (مثلاً میانگین پاسخ در رگرسیون) مقدار دهی اولیه می‌شود. گرادیان (مثلاً باقیمانده) محاسبه می‌شود و سپس یک مدل برای به حداقل رساندن تابع تلفات، متناسب با باقیمانده‌ها می‌شود. مدل فعلی به مدل قبلی اضافه می‌شود و این روند برای تعداد تکرار مشخص شده توسط کاربر ادامه می‌یابد.

همانطور که در سراسر این متن توضیح داده شد، هر تکنیک مدل‌سازی با پارامترهای تنظیم می‌تواند طیف وسیعی از توانایی پیش‌بینی را ایجاد کند - از ضعیف تا قوی. از آنجایی که تقویت به یک یادگیرنده ضعیف نیاز دارد، تقریباً هر تکنیکی با پارامترهای تنظیم را می‌توان به یک یادگیرنده ضعیف تبدیل کرد. درختان، همانطور که مشخص است، به چند دلیل، یک یادگیرنده پایه عالی برای تقویت هستند. اول، آنها این انعطاف را دارند که به سادگی با محدود کردن عمق خود، یادگیرندگان ضعیفی باشند. دوم، درختان جداگانه را می‌توان به راحتی با هم جمع کرد، درست مانند پیش‌بینی‌کننده‌های فردی که می‌توانند در یک مدل رگرسیونی با هم جمع شوند تا یک پیش‌بینی ایجاد کنند. و سوم اینکه درختان را می‌توان خیلی سریع تولید کرد. از این رو، نتایج حاصل از درختان منفرد را می‌توان مستقیماً جمع کرد، بنابراین آنها را ذاتاً برای یک فرآیند مدل‌سازی افزایشی مناسب می‌کند.

هنگامی که از درخت رگرسیون به‌عنوان یادگیرنده پایه استفاده می‌شود، تقویت گرادیان ساده برای رگرسیون دارای دو پارامتر تنظیم است: عمق درخت و تعداد تکرار. عمق درخت در این زمینه به‌عنوان *عمق تعامل* نیز شناخته می‌شود، زیرا هر تقسیم بعدی را می‌توان به‌عنوان یک اصطلاح تعاملی سطح بالاتر با سایر پیش‌بینی‌کننده‌های تقسیم قبلی در نظر گرفت. اگر از مربع خطا به‌عنوان تابع ضرر استفاده شود، یک الگوریتم تقویت ساده با استفاده از این پارامترهای تنظیم را می‌توان در الگوریتم پیدا کرد.  [8. 3 .](#bookmark427)

1 عمق درخت، *D* و تعداد تکرارها، *K را انتخاب کنید*

2 پاسخ متوسط، y را محاسبه *کنید و* از آن به‌عنوان مقدار اولیه پیش‌بینی شده برای هر نمونه استفاده کنید

3 برای *k =1toK* انجام دهید

باقیمانده، تفاوت بین مقدار مشاهده شده و مقدار *فعلی* پیش‌بینی شده را برای هر نمونه محاسبه کنید

4

5

6

7

درخت رگرسیون عمق، *D* را با استفاده از باقیمانده‌ها به‌عنوان پاسخ برازش کنید

هر نمونه را با استفاده از برازش درخت رگرسیون در مرحله قبل پیش‌بینی کنید

با افزودن مقدار پیش‌بینی‌شده تکرار قبلی به مقدار پیش‌بینی‌شده تولید شده در مرحله قبل، مقدار پیش‌بینی‌شده هر نمونه را به‌روزرسانی کنید.

8 پایان

الگوریتم 8. 3: تقویت گرادیان ساده برای رگرسیون

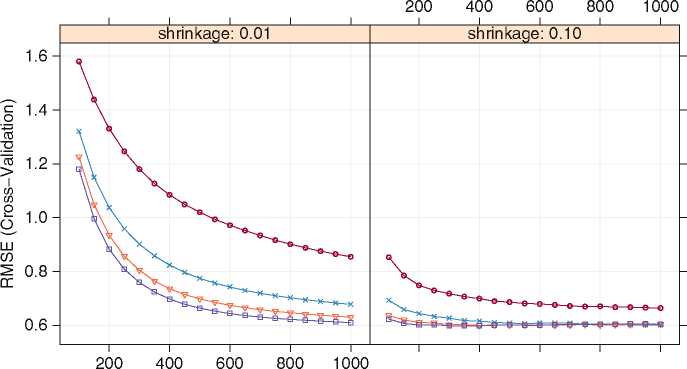
واضح است که نسخه تقویت در الگوریتم ارائه شده است [8. 3](#bookmark427) شباهت‌هایی به جنگل‌های تصادفی دارد: پیش‌بینی نهایی بر اساس مجموعه‌ای از مدل‌ها است و درختان به‌عنوان یادگیرنده پایه استفاده می‌شوند. با این حال، نحوه ساخت مجموعه‌ها به‌طور قابل‌توجهی بین هر روش متفاوت است. در جنگل‌های تصادفی، همه درختان به‌طور مستقل ایجاد می‌شوند، هر درخت به گونه‌ای ایجاد می‌شود که حداکثر عمق را داشته باشد و هر درخت به‌طور مساوی در مدل نهایی مشارکت دارد. درختان در حال تقویت، با این حال، وابسته به درختان گذشته، دارای حداقل عمق هستند و به‌طور نابرابر به مدل نهایی کمک می‌کنند. با وجود این تفاوت‌ها، هم جنگل‌های تصادفی و هم تقویت، عملکرد پیش‌بینی رقابتی را ارائه می‌دهند. زمان محاسبه برای تقویت اغلب بیشتر از جنگل‌های تصادفی است، زیرا جنگل‌های تصادفی را می‌توان به راحتی به صورت موازی پردازش کرد، زیرا درخت‌ها به‌طور مستقل ایجاد می‌شوند.

فریدمن متوجه شد که دستگاه تقویت گرادیان او می‌تواند مستعد ­برازش بیش از حد باشد، زیرا یادگیرنده‌ای که به کار می‌رود - حتی در ظرفیت یادگیری ضعیف آن - وظیفه دارد به‌طور بهینه گرادیان را برازش دهد. این بدان معنی است که تقویت، یادگیرنده بهینه را در هر مرحله از الگوریتم انتخاب می‌کند. علیرغم استفاده از زبان آموزان ضعیف، تقویت همچنان از استراتژی حریصانه انتخاب یادگیرنده ضعیف بهینه در هر مرحله استفاده می‌کند. اگرچه این استراتژی ­یک راه حل بهینه را در مرحله فعلی ایجاد می‌کند، اما دارای اشکالاتی است که مدل جهانی بهینه را پیدا نمی‌کند و همچنین برازش بیش از حد داده‌های آموزشی را دارد. یک راه حل برای حریص این است که فرآیند یادگیری را با استفاده از منظم کردن یا کوچک کردن، به همان روشی که در بخش نشان داده شده است، محدود کنید.  [6. 4 .](#bookmark311) در الگوریتم [8. 3](#bookmark427) ، یک استراتژی منظم‌سازی را می‌توان به خط نهایی حلقه تزریق کرد. به جای افزودن مقدار پیش‌بینی‌شده برای یک نمونه به مقدار پیش‌بینی‌شده تکرار قبلی، تنها کسری از مقدار پیش‌بینی‌شده فعلی به مقدار پیش‌بینی‌شده تکرار قبلی اضافه می‌شود. این کسر معمولاً به‌عنوان *نرخ یادگیری نامیده می‌شود* و با نماد *A پارامترگذاری می‌*شود. این پارامتر می‌تواند مقادیری بین 0 تا 1 بگیرد و به پارامتر تنظیم دیگری برای مدل تبدیل شود.  [ریج وی](#bookmark1024) [( 2007](#bookmark1024) ) پیشنهاد می‌کند که مقادیر کوچک پارامتر یادگیری ( *<* 0. 01) بهترین کار را دارد، اما او همچنین خاطرنشان می‌کند که مقدار پارامتر با زمان محاسبه مورد نیاز برای یافتن یک مدل بهینه نسبت معکوس دارد، زیرا تکرارهای بیشتری لازم است. داشتن تکرارهای بیشتر به این معنی است که حافظه بیشتری برای ذخیره‌سازی مدل مورد نیاز است.

پس از اینکه فریدمن دستگاه تقویت گرادیان خود را منتشر کرد، برخی از ویژگی‌های تکنیک کیسه بریمن را در نظر گرفت. به‌طور خاص، ­ماهیت نمونه‌گیری تصادفی کیسه‌بندی باعث کاهش واریانس پیش‌بینی برای کیسه‌بندی شد. فریدمن الگوریتم ماشین تقویت را با یک طرح نمونه‌گیری تصادفی به روز کرد و روش جدید را *تقویت گرادیان تصادفی نامید.* برای انجام این کار، فریدمن مرحله زیر را قبل از خط در حلقه درج کرد: به‌طور تصادفی کسری از داده‌های آموزشی را انتخاب کنید. باقیمانده‌ها و مدل‌ها در مراحل باقی مانده از تکرار فعلی فقط بر اساس نمونه داده‌ها هستند. کسری از داده‌های آموزشی مورد استفاده که به‌عنوان کسر بسته‌بندی شناخته می‌شود، سپس به پارامتر تنظیم دیگری برای مدل تبدیل می‌شود. به نظر می‌رسد که این اصلاح ساده دقت پیش‌بینی تقویت را بهبود بخشید و در عین حال منابع محاسباتی مورد نیاز را نیز کاهش داد. فریدمن استفاده از کسر کیسه‌ای در حدود 0. 5 را پیشنهاد می‌کند. این مقدار، با این حال، می‌تواند مانند هر پارامتر دیگری تنظیم شود.

شکل [8. 20](#bookmark429) نتایج اعتبار متقابل RMSE را برای درختان تقویت شده در پارامترهای تنظیم عمق درخت (1-7)، تعداد درخت (100-1000) و انقباض (0. 01 یا 0. 1) ارائه می‌دهد. کسر کیسه‌ای در این تصویر روی 0. 5 ثابت شد. هنگام بررسی این رقم، مقدار بزرگتر انقباض (سمت راست

1 o 3 x 5 v 7 □



#درختان

شکل 8. 20: پروفایل‌های RMSE تایید شده متقابل برای مدل درختی تقویت شده

نمودار) بر کاهش RMSE برای تمام انتخاب‌های عمق درخت و تعداد درختان تأثیر دارد. همچنین، RMSE با افزایش عمق درخت زمانی که انقباض 0. 01 است کاهش می‌یابد. زمانی که انقباض 0. 1 و تعداد درختان کمتر از 300 باشد، همین الگو برای RMSE صادق است.

با استفاده از قانون یک خطای استاندارد، درخت تقویت شده بهینه دارای عمق 3 با 400 درخت و انقباض 0. 1 است. این تنظیمات یک RMSE تایید شده متقابل 0. 616 تولید می‌کنند.

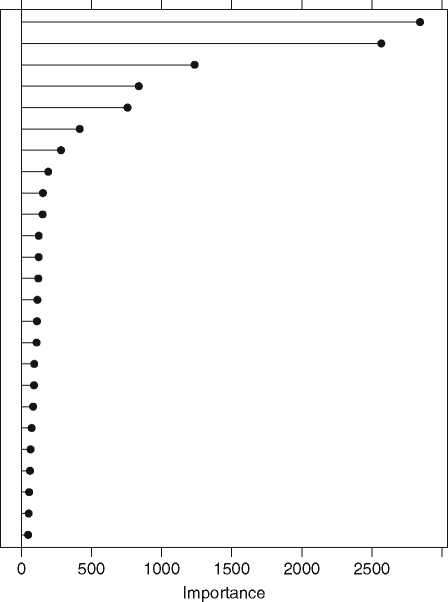
اهمیت متغیر برای تقویت تابعی از کاهش مربع خطا است. به‌طور خاص، بهبود خطای مجذور ناشی از هر پیش‌بینی در هر درخت در مجموعه جمع می‌شود (یعنی هر پیش‌بینی یک ­مقدار بهبود برای هر درخت دریافت می‌کند). سپس مقادیر بهبود برای هر پیش‌بینی در کل مجموعه به‌طور میانگین محاسبه می‌شود تا یک ارزش کلی به دست آید [( فریدمن 2002](#bookmark1016) ; [Ridgeway 2007](#bookmark1024) ). 25 پیش‌بینی برتر برای مدل در شکل 1 ارائه شده است.  [8. 21](#bookmark432) . NumCarbon و MolWeight در این مثال به‌عنوان مهمترین و به دنبال آن SurfaceArea1 و SurfaceArea2 برجسته هستند. مقادیر اهمیت پس از حدود 7 پیش‌بینی کاهش می‌یابد. با مقایسه این نتایج با جنگل‌های تصادفی، می‌بینیم که هر دو روش 4 پیش‌بینی برتر یکسان را، البته به ترتیب متفاوت، شناسایی می‌کنند. مشخصات اهمیت برای تقویت، شیب اهمیت بسیار تندتری ­نسبت به جنگل‌های تصادفی دارد. این به دلیل این واقعیت است که درختان حاصل از تقویت به یکدیگر وابسته هستند و از این رو ساختارهای همبسته‌ای خواهند داشت که روش توسط گرادیان دنبال می‌شود. بنابراین بسیاری از پیش‌بینی‌کننده‌های یکسان در میان درختان انتخاب می‌شوند و سهم آنها ­در معیار اهمیت افزایش می‌یابد. تفاوت بین اهمیت متغیر

NumCarbon MolWeight Surface Area1 Surface Area2

HydrophilicFactor NumAromatic Bonds NumNonHAtoms FP172 NumMultBonds NumHydrogen NumHalogen NumChlorine

NumNonHBonds NumRotBonds NumAtoms FP075 NumOxygen FP072 NumBonds

FP084 FP059 FP043 FP015 FP135 FP028

شکل 8. 21: نمرات اهمیت متغیر برای 25 پیش‌بینی برتر مورد استفاده در مدل تقویت گرادیان تصادفی برای ترتیب حلالیت و بزرگی بین جنگل‌های تصادفی و تقویت نباید نگران کننده باشد. در عوض، باید اینها را به‌عنوان دو دیدگاه متفاوت از داده‌ها در نظر گرفت و از هر دیدگاه برای ارائه درک درستی از روابط ناخالص بین پیش‌بینی‌کننده‌ها و پاسخ استفاده کرد.

8. 7 کوبیست

Cubist یک مدل مبتنی بر قانون است که تلفیقی از چندین روش متدولوژی است ­که مدتی پیش منتشر شد [( کوینلان 1987](#bookmark1023) ، [1992](#bookmark1023) ، [1993a](#bookmark1024) ) اما در این دوره تکامل یافته است. پیش از این، Cubist تنها در یک ظرفیت تجاری در دسترس بود، اما در سال 2011 کد منبع تحت یک مجوز منبع باز منتشر شد. در این زمان، جزئیات کامل نسخه فعلی مدل عمومی شد. توصیف ما از مدل از نسخه منبع باز ناشی می‌شود

از مدل[[17]](#footnote-17) برخی از تفاوت‌های خاص بین کوبیست و رویکردهای توصیف شده قبلی برای درختان مدل و انواع مبتنی بر قانون آنها عبارتند از:

* تکنیک‌های خاص مورد استفاده برای هموارسازی مدل خطی، ایجاد قوانین و هرس متفاوت است
* یک تقویت اختیاری - مانند رویه‌ای که *کمیته نامیده می‌شود*
* پیش‌بینی‌کننده‌های تولید شده توسط قوانین مدل را می‌توان با استفاده از نقاط نزدیک از داده‌های مجموعه آموزشی تنظیم کرد

فرآیند ساخت درخت مدل تقریباً مشابه فرآیند توصیف شده در بخش است.  [8. 2 ،](#bookmark402) اگرچه فرآیند هموارسازی بین مدل‌های خطی پیچیده تر از رویکرد توصیف شده در آن است [کوینلان](#bookmark1023) [( 1992](#bookmark1023) ). در کوبیست، مدل‌ها همچنان با استفاده از ترکیب خطی دو مدل ترکیب می‌شوند:

*y* par = *a X y ( k )* + (1 *- a* ) *X y ( p ) ,*

که در آن *y (* *k )* پیش‌بینی مدل فعلی و *y ( p )* از مدل والد بالای آن در درخت است. در مقایسه با درختان مدل، Cubist نسبت‌های اختلاط را با استفاده از یک معادله متفاوت محاسبه می‌کند. فرض کنید *e (* *k* ) مجموعه‌ای از باقیمانده‌های مدل فرزند باشد (یعنی *y — y ( k* ) ) و *e ( p )* مقادیر مشابهی برای مدل والد باشد. روش هموارسازی ابتدا کوواریانس بین دو مجموعه باقیمانده مدل (که با Cov[ *e (* *p* )، *e (* *k* ) ] مشخص می‌شوند، تعیین می‌کند. این یک معیار کلی از رابطه خطی بین دو مجموعه باقیمانده است. اگر کوواریانس بزرگ باشد، این نشان می‌دهد که باقیمانده‌ها عموماً علامت و مقدار نسبی یکسان دارند، در حالی که مقدار نزدیک به 0 نشان‌دهنده عدم وجود رابطه (خطی) بین خطاهای دو مدل است. Cubist همچنین واریانس تفاوت بین باقیمانده‌ها را محاسبه می‌کند، به‌عنوان مثال، Var[ *e (* *p* ) *- e (* *k* ) ]. ضریب هموارسازی مورد استفاده کوبیست پس از آن است

\_ Var[ *e (* *p* ) ] *-* C ov[ *e (* *k ) ,e (* *p* ) ]

Var[ *e (* *p ) - e (* *k* ) ]

اولین قسمت از شمارشگر متناسب با RMSE مدل والد است. اگر واریانس خطاهای مدل والد بزرگتر از کوواریانس باشد، روش هموارسازی تمایل دارد که وزن کودک را بیشتر از والد کند. برعکس، اگر واریانس مدل والد کم باشد، به آن مدل وزن بیشتری داده می‌شود.

در نهایت مدل با کوچکترین RMSE وزن بیشتری در مدل صاف شده دارد. هنگامی که مدل‌ها دارای RMSE یکسان هستند، در روش هموار (بدون در نظر گرفتن کوواریانس) به همان اندازه وزن می‌شوند.

برخلاف روش‌شناسی «جدا کردن و تسخیر» که قبلاً مورد بحث قرار گرفت، درخت مدل نهایی برای ساخت مجموعه اولیه قوانین استفاده می‌شود. Cubist دنباله‌ای از مدل‌های خطی را در هر گره در یک نمایش واحد و هموار از مدل‌ها جمع‌آوری می‌کند به‌طوری که یک مدل خطی مرتبط با هر قانون وجود دارد. را

نرخ خطای تنظیم شده (معادل [8. 3 )](#bookmark403) معیار هرس و/یا ترکیب قوانین است. با شروع با تقسیم‌هایی که در نزدیکی گره‌های ترمینال ایجاد می‌شوند، هر شرط از قوانین با استفاده از نرخ خطای تنظیم‌شده برای مجموعه آموزشی آزمایش می‌شود. اگر حذف یک شرط در قانون باعث افزایش نرخ خطا نشود، حذف می‌شود. این می‌تواند منجر به حذف کل قوانین از مدل کلی شود. هنگامی که شرایط قانون نهایی شد، یک نمونه جدید با استفاده از میانگین مدل‌های خطی از قوانین مناسب پیش‌بینی می‌شود (برخلاف قانون با بیشترین پوشش).

کمیته‌های مدل را می‌توان با ایجاد دنباله‌ای از مدل‌های مبتنی بر قانون ایجاد کرد. مانند تقویت، هر مدل تحت تأثیر نتیجه مدل‌های قبلی است. به یاد بیاورید که تقویت از وزن‌های جدیدی برای هر نقطه داده بر اساس تناسب‌های قبلی استفاده می‌کند و سپس مدل جدیدی را با استفاده از این وزن‌ها منطبق می‌کند. کامیت تی‌ها متفاوت عمل می‌کنند. نتیجه مجموعه آموزشی بر اساس برازش مدل قبلی تنظیم می‌شود و سپس مجموعه جدیدی از قوانین را با استفاده از این شبه پاسخ ایجاد می‌کند. به‌طور خاص، مدل کمیته *m از یک پاسخ تعدیل شده استفاده می‌کند:*

*y m* ) = *y - (* *y (* *m -* 1) *- y* ).

اساساً، اگر یک نقطه داده کمتر پیش‌بینی شود، مقدار نمونه افزایش می‌یابد به این امید که مدل در تکرار بعدی پیش‌بینی بزرگ‌تری ایجاد کند. به‌طور مشابه، نقاط بیش از حد پیش‌بینی شده به گونه‌ای تنظیم می‌شوند که مدل بعدی پیش‌بینی خود را کاهش دهد. هنگامی که مجموعه کامل مدل‌های کمیته ایجاد شد، نمونه‌های جدید با استفاده از هر مدل پیش‌بینی می‌شوند و پیش‌بینی مبتنی بر قانون نهایی، میانگین ساده پیش‌بینی‌کننده‌های مدل فردی است (به یاد بیاورید که تقویت از وزن‌های مرحله برای میانگین استفاده می‌کند).

هنگامی که مدل مبتنی بر قانون نهایی شد (یا با استفاده از یک مدل واحد یا یک کمیته)، Cubist این توانایی را دارد که پیش‌بینی مدل را با ­نمونه‌هایی از مجموعه آموزشی تنظیم کند [( کوینلان 1993a](#bookmark1024) ). هنگام پیش‌بینی یک نمونه جدید، *K* همسایه‌های مشابه از مجموعه آموزشی تعیین می‌شوند. فرض کنید که مدل نمونه جدید را *y* پیش‌بینی می‌کند و سپس پیش‌بینی نهایی خواهد بود

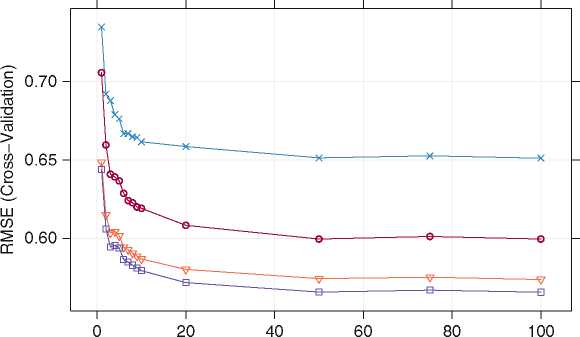
1 K

*ک* ^ *w* [ *t z* + ( *y - t z* )] *,   
z* =1

جایی که *t z* نتیجه مشاهده شده برای یک همسایه مجموعه آموزشی است، *t z* پیش‌بینی مدل آن همسایه است و *w z وزنی* است که با استفاده از فاصله همسایگان مجموعه آموزشی تا نمونه جدید محاسبه می‌شود. با افزایش تفاوت بین پیش‌بینی‌کننده‌های نمونه جدید و نزدیک‌ترین همسایه آن، تعدیل بزرگ‌تر می‌شود.

چندین جزئیات وجود دارد که باید برای اجرای این فرآیند مشخص شود. ابتدا یک معیار فاصله برای تعریف همسایگان مورد نیاز است. اجرای Cubist از فواصل منهتن (معروف به بلوک شهر) برای تعیین نزدیکترین همسایگان استفاده می‌کند. همچنین، همسایگان تنها در صورتی شامل می‌شوند که «به اندازه کافی نزدیک» به نمونه پیش‌بینی شده باشند. برای فیلتر کردن همسایگان، میانگین فاصله زوجی نقاط داده در مجموعه آموزشی به‌عنوان آستانه استفاده می‌شود. اگر فاصله از

0 o 1 X 5 V 9 □



#کمیته ها

شکل 8. 22: پروفایل‌های RMSE معتبر متقابل برای تعداد کمیته‌ها و همسایگان مورد استفاده در Cubist

همسایه بالقوه برای نمونه‌های پیش‌بینی بزرگتر از این فاصله متوسط است، همسایه حذف می‌شود. وزنه‌های *w f* نیز از فاصله همسایگان استفاده می‌کنند. وزن‌های خام به صورت محاسبه می‌شوند

1   
*w f* = *،*

*f  D f* + 0. 5

که در آن *D f* فاصله همسایه تا نمونه پیش‌بینی است. این وزن‌ها به یک مجموع نرمال می‌شوند. وزن دهی تأثیر تأکید بر همسایگانی دارد که شباهت بیشتری به نمونه پیش‌بینی دارند.  [کوینلان](#bookmark1024) [( 1993a](#bookmark1024) ) اطلاعات بیشتر و جزئیات بیشتر را می‌توان در کد منبع [کوبیست در www. RuleQuest. com یافت](http://www.RuleQuest.com) .

برای تنظیم این مدل، تعداد متفاوتی از کمیته‌ها و همسایگان مورد ارزیابی قرار گرفت. شکل [8. 22](#bookmark434) نمایه‌های اعتبارسنجی متقابل را نشان می‌دهد. مستقل از تعداد همسایگان استفاده شده، روندی وجود دارد که در آن با افزایش تعداد کمیته‌ها، خطا به‌طور قابل‌توجهی کاهش می‌یابد و سپس حدود 50 کمیته تثبیت می‌شود. استفاده از مجموعه آموزشی برای تنظیم پیش‌بینی‌کننده‌های مدل جالب است: یک مدل کاملاً مبتنی بر قانون بهتر از تنظیم با یک همسایه منفرد است، اما این خطا زمانی که از 9 همسایه استفاده می‌شود بیشترین کاهش را دارد. در پایان، مدل با کمترین خطا (0. 57 واحد ورود به سامانه‌) ­با 100 کمیته و یک تنظیم با استفاده از نه همسایه مرتبط شد، اگرچه کمیته‌های کمتری نیز می‌توانند بدون از دست دادن عملکرد زیاد مورد استفاده قرار گیرند. برای مدل نهایی کوبیست، میانگین تعداد قوانین در هر کمیته 5. 1 بود اما از 1 تا 15 متغیر بود.

ما می‌توانیم مدل کوبیست را با یک عضو کمیته و بدون تعدیل همسایه با مدل مبتنی بر قانون قبلی مقایسه کنیم. مدل مبتنی بر قانون M5 دارای خطای اعتبارسنجی متقاطع تخمینی 0. 74 بود در حالی که ­مدل کوبیست مربوطه دارای میزان خطای 0. 71 بود. بر اساس تغییرات در نتایج، این تفاوت کمی از نظر آماری معنی‌دار است ( *p -value* : 0. 0485). این ممکن است نشان دهد که تفاوت‌های روش شناختی بین دو روش برای ساخت مدل‌های مبتنی بر قانون، حداقل برای این داده‌ها زیاد نیست.

هیچ تکنیک ثابتی برای اندازه‌گیری اهمیت پیش‌بینی برای مدل‌های کوبیست وجود ندارد. هر مدل خطی یک شیب متناظر برای هر پیش ­فاکتور دارد، اما، همانطور که قبلا نشان داده شد، زمانی که همخطی قابل‌توجهی در داده‌ها وجود داشته باشد، این مقادیر می‌توانند غول پیکر باشند. معیاری که صرفاً بر این مقادیر تکیه می‌کند، پیش‌بینی‌کننده‌هایی که در تقسیم‌بندی‌ها استفاده شده‌اند را نادیده می‌گیرد. با این حال، می‌توان تعداد دفعاتی را که یک متغیر پیش‌بینی در یک مدل خطی یا یک تقسیم استفاده شده است برشمرد و از این جدول‌ها برای دریافت یک ایده تقریبی درباره تأثیر هر پیش‌بین بر روی مدل استفاده کرد. با این حال، این رویکرد اصلاح مبتنی بر همسایه را که گاهی توسط کوبیست استفاده می‌شود نادیده می‌گیرد. مدل‌ساز می‌تواند نحوه وزن کردن شمارش‌ها و مدل‌های خطی را در محاسبه استفاده کلی انتخاب کند.

برای داده‌های حلالیت، مقادیر اهمیت پیش‌بینی برای مدل با 100 کمیته محاسبه شد و پیش‌بینی را با استفاده از 9 نزدیک‌ترین همسایه تصحیح کرد. شکل [8. 23](#bookmark440) تصویری از مقادیر را نشان می‌دهد که در آن محور *x* کل استفاده از پیش‌بینی است (یعنی تعداد دفعاتی که در یک تقسیم یا یک مدل خطی استفاده شده است). مانند بسیاری از مدل‌های دیگر مورد بحث برای این داده‌ها، به نظر می‌رسد پیش‌بینی‌کننده‌های پیوسته تأثیر بیشتری بر مدل نسبت به توصیفگرهای اثر انگشت دارند. برخلاف مدل درختی تقویت‌شده، کاهش تدریجی اهمیت این داده‌ها وجود دارد. زیرمجموعه کوچکی از پیش‌بینی‌کننده‌ها وجود ندارد که بر برازش مدل غالب باشد.

8. 8 محاسبات

**پکیج‌های R مورد استفاده در این بخش عبارتند از caret , Cubist , gbm , ipred , party , partykit , randomForest , rpart , RWeka.**

تک درختان

دو پیاده‌سازی پرکاربرد برای درخت‌های رگرسیون منفرد در R عبارتند از rpart و party. بسته rpart بر اساس متدولوژی CART با استفاده از تابع rpart تقسیم می‌کند، در حالی که طرف بر اساس چارچوب استنتاج شرطی با استفاده از تابع ctree تقسیم می‌کند. هر دو تابع rpart و ctree از روش فرمول استفاده می‌کنند:



شکل 8. 23: امتیازهای اهمیت متغیر برای 25 پیش‌بینی برتر مورد استفاده در مدل کوبیست برای حلالیت

*کتابخانه (rpart)*

*rpartTree <- rpart(y ~. , data = trainData) > # یا،*

*ctreeTree <- ctree(y ~. , data = trainData)*

تابع rpart چندین پارامتر کنترلی دارد که از طریق آرگومان rpart. control قابل دسترسی است. دو مورد که معمولاً در آموزش استفاده می‌شوند و می‌توان از طریق تابع آموزش به آنها دسترسی داشت، پارامتر پیچیده ­ity ( cp ) و حداکثر عمق گره ( maxdepth ) است. برای تنظیم درخت CART بر روی پارامتر پیچیدگی، گزینه روش در تابع آموزش باید روی روش = "rpart" تنظیم شود. برای تنظیم حداکثر عمق، گزینه روش باید روی روش = "rpart2" تنظیم شود :

*set. seed (100)*

*> rpartTune <- train(solTrainXtrans, solTrainY,*

*+ روش = "rpart2"،*

*+ آهنگ طول = 10،*

*+ trControl = trainControl (روش = "cv"))*

به همین ترتیب، بسته حزب دارای چندین پارامتر کنترلی است که از طریق آرگومان ctree\_control قابل دسترسی است. دو مورد از این پارامترها معمولاً در آموزش استفاده می‌شوند: mincriterion و maxdepth. mincriterion معیار آماری را تعریف می‌کند که برای ادامه تقسیم باید رعایت شود. maxdepth حداکثر عمق درخت است. برای تنظیم یک درخت استنتاج شرطی روی mincriterion، گزینه روش در تابع آموزش باید روی روش = "ctree" تنظیم شود. برای تنظیم بیشینه عمق، گزینه روش باید روی method="ctree2" تنظیم شود.

روش نمودار در بسته حزب می‌تواند نمودارهای درختی نشان داده شده در شکل 8. 4 را تولید کند

*طرح (treeObject)*

برای تولید چنین نمودارهایی برای درختان rpart، می‌توان از partykit استفاده کرد تا ابتدا شی rpart را به یک part ob ject تبدیل کرده و سپس از تابع plot استفاده کرد :

*کتابخانه (پارتی کیت)*

*rpartTree2 <- as. party(rpartTree)*

*طرح (rpartTree2)*

درختان مدل

پیاده‌سازی اصلی برای درختان مدل را می‌توان در مجموعه نرم‌افزار Weka یافت، اما مدل را می‌توان در R با استفاده از بسته RWeka مشاهده کرد. دو رابط مختلف وجود دارد: M5P متناسب با درخت مدل است، در حالی که M5Rules از نسخه مبتنی بر قانون استفاده می‌کند. در هر صورت، توابع با روش‌های فرمول کار می‌کنند:

*کتابخانه (RWeka)*

*m5tree <- M5P(y ~. , data = trainData)*

*# یا برای قوانین:*

*m5rules <- M5Rules(y ~. , data = trainData)*

در مثال ما، حداقل تعداد مجموعه‌های آموزشی مورد نیاز برای ایجاد تقسیم‌های اضافی از پیش‌فرض 4-10 افزایش یافت. برای این کار از آرگومان کنترل استفاده می‌شود:

*m5tree <- M5P(y ~. , data = trainData,*

*+ کنترل = Weka\_control (M = 10))*

آرگومان کنترل همچنین گزینه‌هایی برای تغییر استفاده از صاف کردن و هرس کردن دارد. اگر از درخت مدل کامل استفاده شود، تصویرسازی مشابه شکل 8. 10 را می‌توان با تابع نمودار روی خروجی M5P ایجاد کرد.

برای تنظیم این مدل‌ها، تابع آموزش در بسته کارت دو گزینه دارد: با استفاده از روش = "M5" درختان مدل و نسخه‌های مبتنی بر قانون مدل را ارزیابی می‌کند و همچنین استفاده از صاف کردن و هرس کردن را انجام می‌دهد. شکل [8. 12](#bookmark411) نتایج ارزیابی این مدل‌ها را از روی کد نشان می‌دهد:

*> set. seed(100)*

*> m5Tune <- train(solTrainXtrans, solTrainY,*

|  |  |
| --- | --- |
| *+*  *+* | *روش = "M5"،*  *trControl = trainControl (روش = "cv")،* |
| *+*  *+* | *## از گزینه‌ای برای M5() برای تعیین حداقل استفاده کنید*  *تعداد ## نمونه مورد نیاز برای تقسیم بیشتر* |
| *+*  *+* | *داده ## 10 باشد*  *کنترل = Weka\_control (M = 10))* |

به دنبال آن plot(m5Tune). train with method = "M5Rules" فقط نسخه مبتنی بر قانون مدل را ارزیابی می‌کند.

درختان کیسه ای

بسته ipred شامل دو عملکرد برای درختان کیسه‌ای است: bagging از رابط فرمول استفاده می‌کند و ipredbagg دارای رابط غیر فرمول است:

*کتابخانه (ipred)*

*baggedTree <- ipredbagg(solTrainY، solTrainXtrans) >##یا*

*baggedTree <- bagging(y ~. , data = trainData)*

این تابع از تابع rpart استفاده می‌کند و جزئیات مربوط به نوع درخت را می‌توان با ارسال rpart. control به آرگومان کنترل برای bagging و ipredbagg مشخص کرد. به‌طور پیش فرض، بزرگترین درخت ممکن ایجاد می‌شود.

چندین بسته دیگر دارای عملکردهایی برای بسته‌بندی هستند. پکیج RWeka فوق دارای عملکردی به نام Bagging و بسته caret دارای یک چارچوب کلی برای بسته‌بندی انواع مدل‌ها از جمله درختان به نام کیسه است. اگر آرگومان mtry برابر با تعداد پیش‌بینی‌کننده‌ها باشد، درخت‌های استنتاج شرطی را می‌توان با استفاده از تابع cforest در بسته حزبی نیز ذخیره کرد:

*کتابخانه (پارتی)*

*## پارامتر mtry باید تعداد پیش‌بینی‌کننده‌ها باشد (the*

*## تعداد ستون منهای 1 برای نتیجه).*

*bagCtrl <- cforest\_control(mtry = ncol(trainData) - 1)*

*baggedTree <- cforest(y ~. , data = trainData, controls = bagCtrl)*

جنگل تصادفی

پیاده‌سازی اولیه برای جنگل تصادفی از بسته‌ای با همین نام می‌آید:

*کتابخانه (جنگل تصادفی)*

*rfModel <- randomForest(solTrainXtrans، solTrainY)*

*>##یا*

*rfModel <- randomForest (y ~. , data = trainData)*

دو آرگومان اصلی عبارتند از mtry برای تعداد پیش‌بینی‌کننده‌هایی که به‌طور تصادفی به‌عنوان کاندید برای هر تقسیم و ntree برای تعداد نمونه‌های راه‌انداز نمونه‌برداری می‌شوند. پیش‌فرض برای mtry در رگرسیون، تعداد پیش‌بینی‌کننده‌ها ­تقسیم بر 3 است. تعداد درخت‌ها باید به اندازه‌ای بزرگ باشد که نتایج پایدار و قابل تکرار ارائه دهد. اگرچه پیش‌فرض 500 است، حداقل باید از 1000 ­نمونه بوت استرپ استفاده شود (و شاید بیشتر بسته به تعداد پیش‌بینی‌کننده‌ها و مقادیر mtry ). گزینه مهم دیگر اهمیت است. به‌طور پیش‌فرض، امتیازهای اهمیت متغیر محاسبه نمی‌شوند زیرا زمان‌بر هستند. اهمیت = TRUE این مقادیر را ایجاد می‌کند:

*کتابخانه (جنگل تصادفی)*

*rfModel <- randomForest(solTrainXtrans, solTrainY,*

*+ اهمیت = درست،*

*+ nدرخت = 1000)*

برای جنگل‌هایی که با استفاده از درخت‌های استنتاج شرطی ساخته شده‌اند، تابع cforest در بسته حزبی موجود است. گزینه‌های مشابهی دارد، اما استدلال ­کنترل‌ها (به جمع توجه کنید) به کاربر اجازه می‌دهد تا نوع الگوریتم تقسیم را برای استفاده انتخاب کند (مثلاً بایاس یا بی‌طرفدار).

هیچ یک از این توابع را نمی‌توان با داده‌های از دست رفته استفاده کرد.

تابع آموزش حاوی بسته‌بندی‌هایی برای تنظیم هر یک از این مدل‌ها با تعیین روش = "rf" یا روش = "cforest" است. بهینه‌سازی پارامتر mtry ممکن است منجر به افزایش جزئی در عملکرد شود. همچنین، آموزش می‌تواند از روش‌های نمونه‌گیری مجدد استاندارد برای تخمین عملکرد (برخلاف برآورد خارج از کیسه) استفاده کند.

برای مدل‌های randomForest، با استفاده از تابعی به نام اهمیت می‌توان به امتیازهای اهمیت متغیر دسترسی داشت. برای اشیاء cforest، تابع مشابه در بسته party varimp است.

هر بسته تمایل دارد تابع خاص خود را برای محاسبه امتیازهای اهمیت داشته باشد، مشابه وضعیت احتمالات کلاس که در جدول نشان داده شده است.  [B. 1](#bookmark995) از پیوست اول. caret یک تابع یکپارچه به نام varImp دارد که پوششی برای توابع اهمیت متغیر برای اشیاء مدل درختی زیر است: rpart، classbagg (تولید شده توسط توابع بسته‌بندی بسته ipred) randomForest، cforest، gbm و cubist.

درختان تقویت شده

پرکاربردترین بسته برای تقویت درختان رگرسیون از طریق ماشین‌های تقویت گرادیان تصادفی gbm است. مانند رابط جنگل‌های تصادفی، مدل‌ها را می‌توان به دو روش مجزا ساخت:

*کتابخانه (gbm)*

*gbmModel <- gbm. fit(solTrainXtrans، solTrainY، توزیع = "gaussian") >##یا*

*gbmModel <- gbm(y ~. ، داده = trainData، توزیع = "گاوسی")*

توزیع نوع تابع ضرری را که در طول تقویت بهینه می‌شود، تعریف می‌کند. برای پاسخ مداوم، توزیع باید روی "گاوسی" تنظیم شود. تعداد درختان ( n. trees )، عمق درختان ( interaction. depth )، انقباض ( کوچک شدن) و نسبت مشاهداتی که باید ­نمونه برداری شوند ( bag. fraction ) همگی می‌توانند مستقیماً در تماس با gbm تنظیم شوند.

مانند سایر پارامترها، تابع آموزش می‌تواند برای تنظیم روی این پارامترها استفاده شود. به‌عنوان مثال، برای تنظیم عمق تعامل، تعداد درختان و انقباض، ابتدا یک شبکه تنظیم تعریف می‌کنیم. سپس روی این گرید به صورت زیر تمرین می‌کنیم:

*gbmGrid <- expand. grid(. interaction. depth = seq(1, 7, by = 2),*

*+ n. trees = seq(100, 1000, by = 50)*

*+ انقباض = c(0. 01، 0. 1))*

*set. seed (100)*

*> gbmTune <- train(solTrainXtrans, solTrainY,*

*+ روش = "gbm"،*

*+ tuneGrid = gbmGrid،*

*+ ## تابع gbm() مقادیر زیادی تولید می‌کند*

*+ ## از خروجی، پس از گزینه verbose عبور کنید*

*+ ## برای جلوگیری از چاپ زیاد روی صفحه.*

*+ پرمخاطب = FALSE)*

کوبیست

همانطور که قبلا ذکر شد، اجرای این مدل ایجاد شده توسط Rule ­Quest اخیرا با استفاده از یک مجوز منبع باز عمومی شده است. یک بسته R به نام Cubist با استفاده از کد منبع باز ایجاد شد. تابع دارای روش فرمولی نیست زیرا مطلوب است که کد Cubist ایجاد و استفاده از متغیرهای ساختگی را مدیریت کند. برای ایجاد یک مدل مبتنی بر قانون ساده با یک کمیته واحد و بدون تنظیم مبتنی بر نمونه، می‌توانیم از کد ساده استفاده کنیم:

*کتابخانه (کوبیست)*

*cubistMod <- cubist(solTrainXtrans، solTrainY)*

یک استدلال، کمیته‌ها، با مدل‌های متعدد مطابقت دارد. روش پیش‌بینی آشنا برای نمونه‌های جدید استفاده می‌شود:

*پیش‌بینی (cubistMod، solTestXtrans)*

تا زمانی که نمونه‌ها پیش‌بینی نشده اند، نیازی به انتخاب اصلاحات مبتنی بر نمونه نیست. تابع پیش‌بینی یک آرگومان دارد، همسایه ها که می‌تواند یک مقدار صحیح (بین 0 و 9) برای تنظیم پیش‌بینی‌کننده‌های مبتنی بر قانون از مجموعه آموزشی بگیرد.

هنگامی که مدل آموزش داده می‌شود، تابع خلاصه قوانین دقیقی را که استفاده شده است و همچنین مدل خطی هموار نهایی برای هر قانون را تولید می‌کند. همچنین، مانند بسیاری از مدل‌های دیگر، تابع آموزش در بسته کارت می‌تواند مدل را بر روی مقادیر کمیته‌ها و همسایگان از طریق نمونه‌گیری مجدد تنظیم کند:

*> cubistTuned <- train(solTrainXtrans, solTrainY, method = "cubist")*

تمرینات

داده‌های شبیه‌سازی شده از Exercise را دوباره بسازید [7. 2](#bookmark382) :

*> کتابخانه (mlbench)*

*> set. seed(200)*

*> شبیه‌سازی شده <- mlbench. friedman1 (200، sd = 1)*

*> شبیه‌سازی شده <- cbind($x شبیه‌سازی شده،$y شبیه‌سازی شده)*

*> شبیه‌سازی شده <- as. data. frame(شبیه‌سازی شده)*

*> colnames(شبیه‌سازی‌شده)[ncol(شبیه‌سازی‌شده)] <- "y"*

یک مدل جنگل تصادفی را برای همه پیش‌بینی‌کننده‌ها تنظیم کنید، سپس امتیازهای اهمیت متغیر را تخمین بزنید:

*> کتابخانه (جنگل تصادفی)*

*> کتابخانه (کارت)*

*> model1 <- randomForest(y ~. ، داده = شبیه‌سازی شده،*

*+ اهمیت = درست،*

*+ ntree = 1000)*

*> rfImp1 <- varImp(model1, scale = FALSE)*

آیا مدل جنگل تصادفی به‌طور قابل‌توجهی از ­عوامل پیش‌بینیی غیر اطلاعاتی ( V6 - V10 ) استفاده کرد؟

اکنون یک پیش‌بینی اضافه کنید که با یکی از پیش‌بینی‌کننده‌های آموزنده همبستگی زیادی دارد. مثلا:

*> simulated$duplicate1 <- simulated$V1 + rnorm(200) \*. 1*

*> cor (شبیه‌سازی‌$duplicate1، شبیه‌سازی‌شده$V1)*

مدل جنگل تصادفی دیگری را برای این داده‌ها برازش دهید. آیا امتیاز اهمیت برای V1 تغییر کرد؟ چه اتفاقی می‌افتد وقتی پیش‌بینی دیگری اضافه می‌کنید که با V1 همبستگی بالایی دارد ؟

از تابع cforest در بسته حزب برای برازش مدل جنگل تصادفی با استفاده از درختان استنتاج شرطی استفاده کنید. تابع بسته حزب varimp می‌تواند اهمیت پیش‌بینی را محاسبه کند. استدلال شرطی آن تابع ­بین معیار اهمیت سنتی و نسخه اصلاح شده شرح داده شده در تغییر می‌کند [استروبل و همکاران ( 2007](#bookmark1026) ). آیا این اهمیت‌ها همان الگوی مدل جنگل تصادفی سنتی را نشان می‌دهند؟

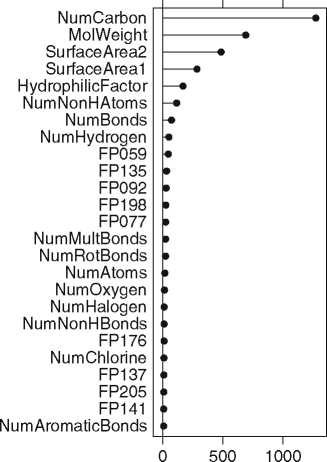
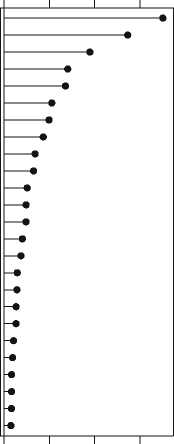
این روند را با مدل‌های مختلف درخت مانند درختان تقویت شده و کوبیست تکرار کنید. آیا همین الگو رخ می‌دهد؟

مولی وزن

HydrophilicFactor NumAromatic Bonds

NumCarbon Surface Area1 NumMultBonds NumAtoms NumHalogen FP172 Surface Area2 NumNonHAtoms NumHydrogen NumBonds NumRotBonds NumNonHBonds FP075

NumRings FP015 FP072 FP074 FP107 FP070 FP065 FP044 FP104

شکل 8. 24: مقایسه مقادیر متغیر اهمیت برای مقادیر متفاوت کسر کیسه و پارامترهای انقباض. هر دو پارامتر تنظیم در شکل *سمت چپ روی 0. 1 تنظیم شده اند.* هر دو در شکل *سمت راست* روی 0. 9 تنظیم شده اند

0 200 400 600

Importance

Importance

از یک شبیه‌سازی برای نشان دادن بایاس درختی با دانه بندی‌های مختلف استفاده کنید.

در گرادیان تصادفی، افزایش کسر کیسه‌بندی و نرخ یادگیری، بر ساختار درختان حاکم است، زیرا آنها توسط گرادیان هدایت می‌شوند. اگرچه مقادیر بهینه این پارامترها باید از طریق فرآیند تنظیم به دست آیند، اما درک اینکه چگونه بزرگی این پارامترها بر بزرگی‌های با اهمیت متغیر تأثیر می‌گذارد مفید است. شکل [8. 24](#bookmark458) نمودارهای اهمیت متغیر را برای تقویت با استفاده از دو مقدار شدید برای کسر بسته‌بندی (0. 1 و 0. 9) و نرخ یادگیری (0. 1 و 0. 9) برای داده‌های حلالیت ارائه می‌دهد. نمودار سمت چپ هر دو پارامتر را روی 0. 1 و نمودار سمت راست هر دو روی 0. 9 تنظیم شده است:

چرا مدل سمت راست اهمیت خود را فقط بر چند پیش‌بینی اول متمرکز می‌کند، در حالی که مدل سمت چپ اهمیت خود را بین پیش‌بینی‌کننده‌های بیشتری پخش می‌کند؟

به نظر شما کدام مدل نسبت به نمونه‌های دیگر بیشتر پیش‌بینی می‌کند؟

چگونه افزایش عمق تعامل بر شیب اهمیت پیش‌بینی ­برای هر یک از مدل‌ها در شکل 1 تأثیر می‌گذارد.  [8. 24 \_](#bookmark458)

از یک پیش‌بینی واحد در داده‌های حلالیت، مانند وزن مولکولی یا تعداد اتم‌های کربن استفاده کنید و چندین مدل را متناسب کنید:

یک درخت رگرسیون ساده

یک مدل جنگل تصادفی

مدل‌های مختلف کوبیست با یک قانون یا چند کمیته (هر کدام با و بدون استفاده از تنظیمات همسایه)

داده‌های پیش‌بینی را در مقابل نتایج حلالیت برای مجموعه آزمایش رسم کنید. پیش‌بینی‌کننده‌های مدل برای مجموعه آزمایشی را پوشش دهید. مدل چگونه متفاوت است؟ آیا تغییر پارامتر(های) تنظیم به‌طور قابل‌توجهی بر برازش مدل تأثیر می‌گذارد؟

مدل‌های مختلف مبتنی بر درخت و قانون را برای داده‌های Tecator که در تمرین مورد بحث قرار گرفت، مناسب کنید [6. 1](#bookmark335) . چگونه آنها را با مدل‌های خطی مقایسه می‌کنند؟ آیا به نظر می‌رسد که همبستگی‌های بین پیش‌بینی بر مدل‌های شما تأثیر می‌گذارد؟ اگر چنین است، چگونه می‌توانید ­داده‌های پیش‌بینی را برای کاهش این مسأله تبدیل یا رمزگذاری مجدد کنید؟

به مسأله نفوذپذیری شرح داده شده در تمرین‌ها برگردید [6. 2](#bookmark336) و [7. 4 .](#bookmark383) چندین مدل مبتنی بر درخت را آموزش دهید و نمونه‌برداری مجدد و عملکرد مجموعه آزمایشی را ارزیابی کنید:

کدام مدل مبتنی بر درخت، نمونه‌گیری مجدد و مجموعه آزمایشی بهینه را در هر ­فرمنس ارائه می‌دهد؟

آیا هیچ یک از این مدل‌ها از مدل‌های رگرسیون مبتنی بر کوواریانس یا غیرکوواریانس که قبلاً برای این داده‌ها ایجاد کرده‌اید، عملکرد بهتری دارند؟ از چه معیارهایی برای مقایسه عملکرد مدل‌ها استفاده کردید؟

از بین تمام مدل‌هایی که تاکنون ایجاد کرده اید، در صورت وجود، کدام یک را جایگزین آزمایش آزمایشگاهی نفوذپذیری می‌کنید؟

رجوع به تمرین شود [6. 3](#bookmark336) و [7. 5](#bookmark384) که یک فرآیند تولید شیمیایی را توصیف می‌کند. از همان مراحل انباشت داده، تقسیم داده‌ها و مراحل پیش پردازش مانند قبل استفاده کنید و چندین مدل مبتنی بر درخت را آموزش دهید:

کدام مدل رگرسیون مبتنی بر درخت، نمونه‌گیری مجدد و عملکرد مجموعه تست را بهینه می‌دهد؟

کدام پیش‌بینی‌کننده‌ها در مدل رگرسیون درختی بهینه مهم هستند؟ آیا متغیرهای بیولوژیکی یا فرآیندی بر فهرست غالب هستند؟ چگونه 10 پیش‌بینی مهم برتر با 10 پیش‌بینی برتر مدل‌های خطی و غیرخطی بهینه مقایسه می‌شود؟

تک درخت بهینه را با توزیع عملکرد در گره‌های انتهایی رسم کنید. آیا این دیدگاه از داده‌ها دانش بیشتری در مورد پیش‌بینی‌کننده‌های بیولوژیکی یا فرآیندی و رابطه آنها با عملکرد ارائه می‌دهد؟

**خلاصه‌ای از مدل‌های حلالیت**

در چند فصل گذشته، مدل‌های مختلفی با مجموعه داده‌های حلالیت مطابقت دارند. چگونه مدل‌ها برای این داده‌ها مقایسه می‌شوند و کدام یک باید برای مدل نهایی انتخاب شود؟ شکل.  [9. 1](#bookmark462) و [9. 2](#bookmark462) نمودارهای پراکندگی معیارهای عملکرد محاسبه شده با استفاده از اعتبارسنجی متقاطع و.

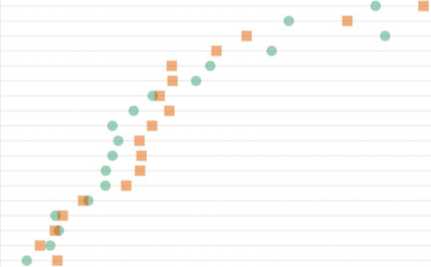
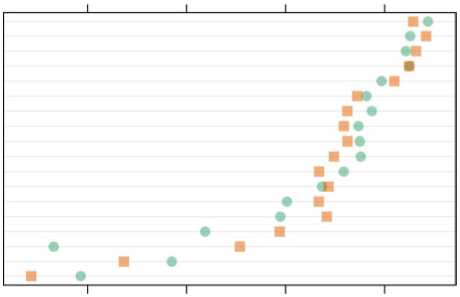
به استثنای مدل‌هایی که عملکرد داده‌های مجموعه آزمون را نشان می‌دهد ضعیفی دارند، همبستگی نسبتاً بالایی بین نتایج حاصل از نمونه‌گیری مجدد و مجموعه آزمایشی وجود دارد (0. 9 برای RMSE و 0. 88 برای R2 *).* در بیشتر موارد، مدل‌ها تمایل دارند که ترتیب را به‌طور مشابه رتبه‌بندی کنند. *K-* نزدیک‌ترین همسایه ضعیف‌ترین عملکرد را داشتند و به دنبال آن دو روش تک درختی قرار گرفتند. در حالی که بسته‌بندی این درختان کمک کرد، اما مدل‌ها را چندان رقابتی نکرد. علاوه بر این، مدل‌های جنگلی تصادفی شرطی نتایج متوسطی داشتند.

مجموعه‌ای از مدل‌ها وجود داشت که نتایج بهتری از جمله درختان مدل، رگرسیون خطی، مدل‌های خطی جریمه‌شده، MARS و شبکه‌های عصبی را نشان دادند. این مدل‌ها ساده‌تر هستند، اما با توجه به تعداد پیش‌بینی‌کننده‌های درگیر در مدل‌های خطی و ­پیچیدگی درخت‌های مدل و MARS، قابل تفسیر نیستند. در بیشتر موارد، اجرای آنها آسان خواهد بود. به یاد داشته باشید که این نوع مدل ممکن است توسط یک ­شرکت داروسازی برای غربالگری *میلیون‌ها* ترکیب بالقوه مورد استفاده قرار گیرد، بنابراین سهولت پیاده‌سازی را نباید ساده انگاشت.

گروه مدل‌های با کارایی بالا شامل ماشین‌های بردار پشتیبانی (SVM)، درختان تقویت‌شده، جنگل‌های تصادفی و کوبیست می‌شوند. هر کدام در اصل یک جعبه سیاه با یک معادله پیش‌بینی بسیار پیچیده است. عملکرد این مدل‌ها سر و گردن بالاتر از بقیه است، بنابراین احتمالاً در یافتن پیاده‌سازی‌های محاسباتی کارآمد که می‌توان برای پیش‌بینی تعداد زیادی از نمونه‌های جدید از آنها استفاده کرد، ارزشی وجود دارد.

آیا تفاوت واقعی بین این مدل‌ها وجود دارد؟ با استفاده از نتایج نمونه‌گیری مجدد، مجموعه‌ای از فواصل اطمینان برای مشخص کردن تفاوت‌های RMSE در مدل‌ها با استفاده از تکنیک‌های نشان‌داده‌شده در بخش ساخته شد.  [4. 8 .](#bookmark232) شکل [9. 3](#bookmark463) فواصل را نشان می‌دهد. تعداد بسیار کمی از نظر آماری معنی دار هستند

اعتبارسنجی متقابل • مجموعه تست i



0.75 0.80 0.85 0.90

0.6 0.7 0.8 0.9

1.0 1.1

Cubist Boosted Tree SVMp SVMr شبکه عصبی الاستیک جنگل تصادفی MARS Ridge PLS Linear Reg.

وضعیت کیسه‌ای M5. وضعیت درخت. تصادفی جنگل کیسه درخت درخت وضعیت. درخت KNN

R-Squared

نموداری از *R* 2 مدل‌های حلالیت با اعتبارسنجی متقاطع 10 ­برابری و مجموعه آزمایشی برآورد شده‌اند

Cross-Validation > Test Set ■

KNN

شرط درخت درخت کیسه دار وضعیت درخت. جنگل تصادفی کیسه‌ای شرایط. درخت

M5

قانون خطی

PLS Ridge Mars

شبکه عصبی

شبکه الاستیک جنگل تصادفی SVMr SVMp درخت تقویت شده

کوبیست

RMSE

نموداری از مدل‌های حلالیت RMSE که با ­اعتبارسنجی متقاطع 10 برابری و مجموعه آزمایشی تخمین زده شده است.

rf - SVMr

rf - gbm

rf - کوبیست

gbm - SVMr gbm - کوبیست کوبیست - SVMr

-0. 05 0. 00 0. 05 0. 10

تفاوت در RMSE

سطح اطمینان 0. 992 (تعدد تنظیم شده)

فواصل اطمینان برای تفاوت‌های RMSE برای تفاوت‌های مدل‌های با کارایی بالا. علاوه بر این، بیشتر تفاوت‌های میانگین تخمین زده شده کمتر از 0. 05 واحد ورود به سامانه هستند که از نظر علمی معنی دار نیستند. با توجه به این، هر یک از این مدل‌ها انتخاب معقولی خواهد بود.

**مطالعه موردی: مقاومت فشاری مخلوط بتن**

تا کنون، تمرکز بر روی مجموعه داده‌های مشاهده‌ای بوده است که در آن مقادیر پیش‌بینی‌کننده‌ها از قبل مشخص نشده بودند. به‌عنوان مثال، داده‌های QSAR مورد استفاده در فصل‌های قبلی شامل مجموعه‌ای از ترکیبات متنوع بود که مقدار کافی از «فضای شیمیایی» را به تصویر می‌کشید. این مجموعه داده خاص با تعیین مقادیر دقیق برای خواص شیمیایی (مانند وزن مولکولی) ایجاد نشده است. در عوض ترکیبات از یک جمعیت مناسب برای استفاده در مدل نمونه برداری شدند.

*آزمایش‌های طراحی‌شده* با برنامه‌ریزی مقادیر دقیق پیش‌بینی‌کننده‌ها (که به‌عنوان عوامل در این زمینه نامیده می‌شوند) با استفاده از نوعی روش‌شناسی استراتژیک ایجاد می‌شوند. پیکربندی تنظیمات پیش‌بینی به گونه‌ای ایجاد می‌شود که از ویژگی‌های ریاضی و تجربی خوبی برخوردار باشند. یکی از این ویژگی‌ها *تعادل است.* طراحی متعادل طرحی است که در آن هیچ یک از عوامل تجربی (یعنی پیش‌بینی‌کننده‌ها) تمرکز بیشتری نسبت به سایرین نداشته باشد. در بیشتر موارد، این بدان معناست که هر پیش‌بینی تعداد سطوح ممکن یکسانی دارد و ­فراوانی سطوح برای هر عامل معادل است. ویژگی‌های مورد استفاده برای انتخاب بهترین طرح آزمایشی بر اساس مرحله آزمایش هدایت می‌شوند.

باکس و همکاران [( 1978](#bookmark1012) ) مفهوم *آزمایش متوالی را رایج کرد* که در آن تعداد زیادی از عوامل تجربی ممکن با وضوح پایین (یعنی "ریختن یک شبکه گسترده") برای تعیین عوامل *فعال* یا مهم مرتبط با نتیجه بررسی می‌شوند. هنگامی که اهمیت پیش‌بینی‌کننده‌ها کمی‌سازی شد، آزمایش‌های متمرکزتری با زیرمجموعه‌ای از ­عوامل مهم ایجاد می‌شوند. در آزمایش‌های بعدی، ماهیت رابطه بین عوامل مهم را می‌توان بیشتر روشن کرد. آخرین مرحله در توالی آزمایش‌ها تنظیم دقیق تعداد کمی از عوامل مهم است. *آزمایشات سطح پاسخ* [( Myers and Montgomery 2009](#bookmark1022) ) از مجموعه کوچکتری از مقادیر پیش‌بینی استفاده می‌کنند. در اینجا، هدف اصلی بهینه‌سازی تنظیمات تجربی بر اساس یک مدل غیر خطی از پیش‌بینی‌کننده‌های تجربی است.

آزمایش‌های طراحی شده و مدل‌های پیش‌بینی چندین تفاوت دارند[[18]](#footnote-18) [[19]](#footnote-19) :

* دنباله‌ای از مطالعات بر یک مجموعه داده واحد و جامع ترجیح داده می‌شود که تلاش می‌کند همه پیش‌بینی‌کننده‌های ممکن (یعنی عوامل تجربی) را با مقادیر زیادی در هر پیش‌بینی شامل شود. پارادایم تکراری برنامه ریزی، طراحی و سپس تحلیل یک آزمایش، در ظاهر، متفاوت از بسیاری از مسائل مدل‌سازی پیش‌بینی است.
* تا مراحل پایانی آزمایش متوالی، تمرکز بر این است ­که بفهمیم کدام پیش‌بینی‌کننده‌ها بر نتیجه و چگونه تأثیر می‌گذارند. هنگامی که آزمایش‌های سطح پاسخ مورد استفاده قرار می‌گیرند، تمرکز فعالیت‌ها صرفاً بر روی پیش‌بینی است.

این مطالعه موردی بر پیش‌بینی فرمول‌بندی بهینه داده‌های مبتنی بر مخلوط بتن از آزمایش‌های طراحی‌شده تمرکز خواهد کرد.

بتن جزء لاینفک اکثر جوامع صنعتی است. تقریباً در تمام سازه‌ها و در بسیاری از جاده‌ها تا حدی از آن استفاده می‌شود. یکی از خواص اصلی مورد توجه (در کنار هزینه) مقاومت فشاری ­بتن سخت است. ترکیب بسیاری از بتن‌ها شامل تعدادی مواد خشک است که با آب مخلوط شده و سپس خشک و سفت می‌شوند. با توجه به فراوانی و نقش حیاتی آن در زیرساخت، ترکیب ­مهم است و به‌طور گسترده مورد مطالعه قرار گرفته است. در این فصل، مدل‌هایی برای کمک به یافتن دستور العمل‌های بالقوه برای به حداکثر رساندن مقاومت فشاری ایجاد می‌شود.

Yeh [( 2006](#bookmark1028) ) یک نوع استاندارد از راه اندازی آزمایشی را برای این سناریو توصیف می‌کند که ­*طراحی مخلوط* نامیده می‌شود. [( Cornell 2002](#bookmark1014) ; [مایرز و مونتگومری 2009](#bookmark1022) ). در اینجا، از مرزهای حد بالایی و پایینی *نسبت مخلوط* برای هر عنصر برای ایجاد مخلوط‌های متعددی استفاده می‌شود که به‌طور روشمند فضای درون مرزها را پر می‌کنند. برای نوع خاصی از طراحی مخلوط، یک مدل رگرسیون خطی مربوطه وجود دارد که معمولاً برای مدل‌سازی رابطه بین مواد تشکیل دهنده و نتیجه استفاده می‌شود. این مدل‌های خطی می‌توانند شامل اثرات متقابل و اصطلاحات درجه بالاتر برای مواد تشکیل دهنده باشند. مواد مورد استفاده در [بله](#bookmark1028) [( 2006](#bookmark1028) ) عبارت بودند از:

* سیمان (کیلوگرم بر متر [[20]](#footnote-20))
* خاکستر بادی (kg/m 3 )، ذرات کوچکی که از سوزاندن زغال سنگ تولید می‌شوند
* سرباره کوره بلند (kg/ m3 )
* آب (kg/m 3 )
* فوق روان کننده (kg/m 3 )، یک افزودنی که تجمع ذرات را کاهش می‌دهد
* سنگدانه درشت (kg/ m3 )
* سنگدانه ریز (کیلوگرم بر مترمربع )

Yeh [( 2006](#bookmark1028) ) همچنین یک عامل غیر مخلوط اضافی مربوط به ­مقاومت فشاری را توصیف می‌کند: سن مخلوط (در زمان آزمایش). از آنجایی که این یک عنصر نیست، معمولاً به‌عنوان یک *عامل فرآیند نامیده می‌شود.* طرح‌های آزمایشی خاص (و فرم‌های مدل خطی) برای آزمایش‌هایی وجود دارد که متغیرهای مخلوط و فرآیند را ترکیب می‌کنند (نگاه کنید به [کرنل](#bookmark1014) [( 2002](#bookmark1014) ) برای جزئیات بیشتر).

Yeh [( 1998](#bookmark1028) ) رویکرد متفاوتی را برای مدل‌سازی آزمایش‌های مخلوط بتن اتخاذ ­می‌کند. در اینجا، آزمایش‌های جداگانه‌ای از 17 منبع با ­عوامل تجربی مشترک در یک «متا آزمایش» ترکیب شدند و نویسنده از شبکه‌های عصبی برای ایجاد مدل‌های پیش‌بینی در کل فضای مخلوط استفاده کرد. سن نیز در مدل گنجانده شده است. اگرچه نسخه عمومی مجموعه داده ­شامل 1030 نقطه داده در آزمایش‌های مختلف است [بله](#bookmark1028) [( 1998](#bookmark1028) ) بیان می‌کند که برخی از مخلوط‌ها به دلیل شرایط غیر استاندارد از تحلیل او حذف شدند. هیچ اطلاعاتی در مورد اینکه دقیقاً کدام مخلوط‌ها حذف شده اند وجود ندارد، بنابراین تحلیل‌های اینجا از تمام نقاط داده موجود استفاده می‌کنند. جدول [10. 1](#bookmark468) خلاصه‌ای از داده‌های پیش‌بینی (به مقدار) و نتیجه را نشان می‌دهد.

شکل [10. 1](#bookmark471) نمودارهای پراکندگی هر پیش‌بینی را در مقابل مقاومت فشاری نشان می‌دهد. سن یک رابطه غیرخطی قوی با پیش‌بینی نشان می‌دهد و مقدار سیمان رابطه خطی دارد. توجه داشته باشید که بسیاری از ­عناصر موجود دارای فرکانس زیادی از یک مقدار هستند، مانند صفر برای فوق روان کننده و مقدار خاکستر بادی. در این موارد، مقاومت فشاری برای آن مقادیر پیش‌بینی‌کننده‌ها بسیار متفاوت است. این ممکن است ­نشان دهد که برخی از روش‌های پارتیشن بندی، مانند درختان یا MARS، ممکن است بتوانند این مخلوط‌ها را در مدل جدا کرده و به‌طور موثرتری ­مقاومت فشاری را پیش‌بینی کنند. به‌عنوان مثال، 53 مخلوط بدون فوق روان کننده یا خاکستر بادی وجود دارد، اما دقیقا 228 کیلوگرم بر متر مکعب آب دارند. این ممکن است نشان‌دهنده زیرجمعیت مهمی از مخلوط‌ها باشد که ممکن است از مدلی استفاده کنند که مخصوص این نوع مخلوط‌ها است. یک مدل مبتنی بر درخت یا قانون توانایی مدل‌سازی چنین زیرگروهی را دارد در حالی که مدل‌های رگرسیون کلاسیک این کار را نمی‌کنند.

اگرچه داده‌های موجود نشان نمی‌دهد که کدام فرمول از هر منبع آمده است، 19 مخلوط متمایز با نقاط داده تکراری وجود دارد. اکثر این مخلوط‌ها فقط دو یا سه شرط تکراری داشتند، ­اگرچه برخی شرایط تا چهار تکرار دارند. هنگام مدل‌سازی این داده‌ها، نتایج تکراری نباید طوری رفتار شود که گویی در ­مشاهدات وابسته هستند. برای مثال، وجود مخلوط‌های تکراری در مجموعه‌های آموزشی و آزمایشی می‌تواند منجر به ارزیابی‌های بیش از حد خوش‌بینانه از نحوه عملکرد مدل شود. یک رویکرد متداول در اینجا این است که نتایج را در هر مخلوط منحصربفرد میانگین کنید. در نتیجه، تعداد مخلوط‌های موجود برای مدل‌سازی از 1030 به 992 کاهش می‌یابد.

9 متغیر 1030 مشاهده

سیمان il. . . i. . iini. iii ii ii. i. . . i. . ،

n گمشده منحصر به فرد میانگین 0. 05 0. 10 0. 25 0. 50 0. 75 0. 90 0. 95

1,030 0 278 281. 2 143. 7 153. 5 192. 4 272. 9 350. 0 425. 0 480. 0

کمترین : 102. 0 108. 3 116. 0 122. 6 132. 0

بالاترین: 522. 0 525. 0 528. 0 531. 3 540. 0

BlastFurnace Slag I

n گمشده منحصر به فرد میانگین 0. 05 0. 10 0. 25 0. 50 0. 75 0. 90 0. 95

1,030 0 185 73. 9 0. 0 0. 0 0. 0 22. 0 142. 9 192. 0 236. 0

کمترین : 0. 0 11. 0 13. 6 15. 0 17. 2

بالاترین: 290. 2 305. 3 316. 1 342. 1 359. 4

فلای اش I. .

n گمشده منحصر به فرد میانگین 0. 05 0. 10 0. 25 0. 50 0. 75 0. 90 0. 95 1,030 0 156 54. 19 0. 0 0. 0 0. 0 0. 0 118. 3 141. 1 167. 0

کمترین : 0. 0 24. 5 59. 0 60. 0 71. 0 بالاترین: 194. 0 194. 9 195. 0 200. 0 200. 1

Water

n missin 1030

منحصر به فرد میانگین 0. 05 0. 10 0. 25 0. 50 0. 75" 0:§0'"0. 95

195 181. 6 146. 1 154. 6 164. 9 185. 0 192. 0 203. 5 228. 0

کمترین : 121. 8 126. 6 127. 0 127. 3 137. 8 بالاترین: 228. 0 236. 7 237. 0 246. 9 247. 0

فوق روان کننده I

n گمشده منحصر به فرد میانگین 0. 05 0. 10 0. 25 0. 50 0. 75 0. 90 0. 95 1,030 0 111 6. 205 0. 00 0. 00 0. 00 6. 40 10. 20 12. 25 16.

کمترین : 0. 0 1. 7 1. 9 2. 0 2. 2،

بالاترین: 22. 0 22. 1 23. 4 28. 2 32. 2

درشت دانه . من. . . . Ilili. J. ii. iil i. . .

n گمشده منحصر به فرد میانگین 0. 05 0. 10 0. 25 0. 50 0. 75 0. 90 0. 95

1,030 0 284 972. 9 842. 0 852. 1 932. 0 968. 0 1029. 4 1076. 5 1104. 0

کمترین : 801. 0 801. 1 801. 4 811. 0 814. 0

بالاترین: 1124. 4 1125. 0 1130. 0 1134. 3 1145. 0

FineAggregate ii. , i. lli. Jii. ilii . . .

n گمشده منحصر به فرد میانگین 0. 05 0. 10 0. 25 . 50 0. 75 . 90 0. 95

1030 0 302 773. 6 613. 0 664. 1 730. 9 779. 5 824. 0 880. 8 898. 1

کمترین : 594. 0 605. 0 611. 8 612. 0 613. 0

بالاترین: 925. 7 942. 0 943. 1 945. 0 992. 6

سن

n گمشده منحصر به فرد میانگین 0. 05 0. 10 0. 25 0. 50 0. 75 0. 90 0. 95

1,030 0 14 45. 66 3 3 7 28 56 100 180

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 3 | 7 | 14 | 28 | 56 | 90 | 91 | 100 | 120 | 180 | 270 | 360 | 365 |
| فرکانس | 2 | 134 | 126 | 62 | 425 | 91 | 54 | 22 | 52 | 3 | 26 | 13 | 6 | 14 |
| % | 0 | 13 | 12 | 6 | 41 | 9 | 5 | 2 | 5 | 0 | 3 | 1 | 1 | 1 |

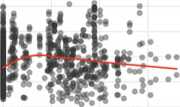
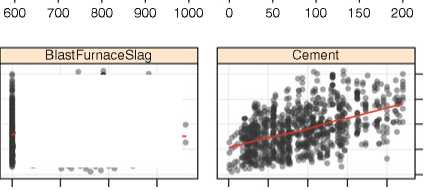
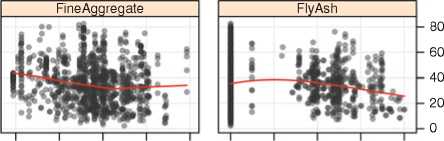
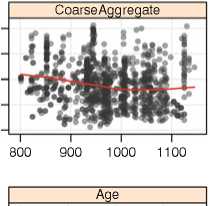
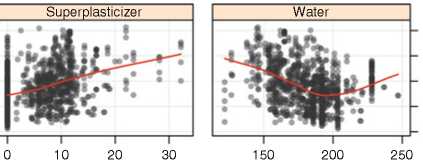
فشاری i. iiiii. ii. iiiilllliiiiillllnillllilldl

n گمشده منحصر به فرد میانگین 0. 05 0. 10 0. 25 0. 50 0. 75 0. 90 0. 95

1,030 0 845 35. 82 10. 96 14. 20 23. 71 34. 45 46. 14 58. 82 66. 80

کمترین : 2. 33 3. 32 4. 57 4. 78 4. 83

بالاترین: 79. 40 79. 99 80. 20 81. 75 82. 60



80 -

60 -

40 -

20 -

0

80 -

60 -

40 -

20 -

0

100 200 300 400 500

0 100 200 300

Feature

jH+H **“1 1 1 1**

0 100 200 300

شکل 10. 1: نمودارهای پراکندگی پیش‌بینی‌کننده‌های بتن در مقابل مقاومت فشاری

10. 1 استراتژی ساخت مدل

مدل‌های شبکه عصبی مورد استفاده در [بله](#bookmark1028) [( 1998](#bookmark1028) ) شبکه‌های تک لایه با هشت واحد پنهان بودند. چندین رویکرد تقسیم داده توسط نویسنده اصلی استفاده شد. چهار مدل با مجموعه‌های آموزشی مختلف برازش داشتند به‌طوری که همه داده‌ها از یک منبع واحد در هر بار نگهداری می‌شدند. این مدل‌ها به مجموعه آزمایشی *R* 2 منجر شدند مقادیر از 0. 814 تا 0. 895. آنها همچنین از یک نمونه تصادفی 25 درصدی داده‌ها برای مجموعه‌های آزمون نگهدارنده استفاده کردند. این کار چهار بار برای تولید مجموعه تست *R* 2 تکرار شد مقادیر بین 0. 908 و 0. 922.

اگرچه نمی‌توان مقایسه سیب به سیب را با تحلیل او انجام داد [بله](#bookmark1028) [( 1998](#bookmark1028) )، یک رویکرد تقسیم داده مشابه برای این مطالعه موردی اتخاذ خواهد شد. یک مجموعه نگهدارنده تصادفی 25٪ ( *n* = 247) به‌عنوان یک مجموعه آزمایشی و پنج تکرار از اعتبارسنجی متقاطع 10 برابری برای تنظیم مدل‌های مختلف استفاده خواهد شد.

در این مطالعه موردی، یک سری مدل ایجاد و ارزیابی خواهد شد. هنگامی که یک مدل نهایی انتخاب شد، مدل برای پیش‌بینی مخلوط‌هایی با مقاومت فشاری بهینه در محدودیت‌های عملی استفاده می‌شود.

چگونه باید از پیش‌بینی‌کننده‌ها برای مدل‌سازی نتیجه استفاده کرد؟ [بله](#bookmark1028) [( 1998](#bookmark1028) ) رویکردهای سنتی، مانند تکیه بر نسبت آب به سیمان را مورد بحث قرار می‌دهد، اما پیشنهاد می‌کند که داده‌های تجربی موجود با استراتژی‌های تاریخی سازگار نیستند. در این فصل، پیش‌بینی‌کننده‌ها مدل‌ها را ­به‌عنوان نسبت کل مقدار وارد می‌کنند. به همین دلیل، یک وابستگی داخلی در مقادیر پیش‌بینی وجود دارد (هر پیش‌بینی را می‌توان به‌طور خودکار ­با دانستن مقادیر هفت مورد دیگر تعیین کرد). علی‌رغم این، همبستگی‌های زوجی زیاد نیستند و بنابراین، انتظار نداریم روش‌هایی که برای مقابله با هم خطی طراحی شده‌اند (به‌عنوان مثال، PLS، رگرسیون پشته) عملکردی بهتر از سایر مدل‌ها داشته باشند.

مجموعه‌ای از مدل‌ها مورد آزمایش قرار گرفتند:

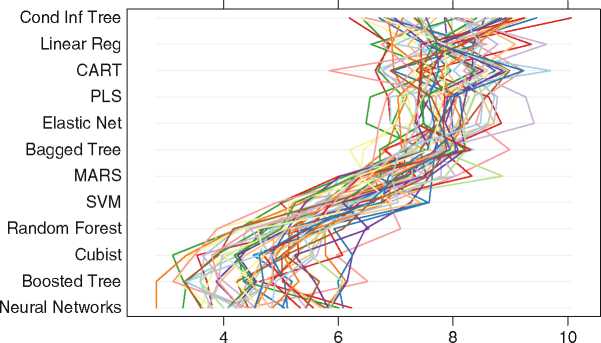
* رگرسیون خطی، حداقل مربعات جزئی و شبکه الاستیک. هر مدل از مجموعه گسترده‌ای از پیش‌بینی‌کننده‌ها استفاده می‌کرد که شامل تمام تعاملات دو عاملی (به‌عنوان مثال، سن *x* آب) و اصطلاحات درجه دوم بود.
* ماشین‌های بردار پشتیبان تابع پایه شعاعی (SVM).
* مدل‌های شبکه عصبی
* مدل‌های MARS
* درختان رگرسیون (هم درختان CART و هم درختان استنتاج شرطی)، درختان مدل (با و بدون قوانین) و کوبیست (با و بدون کمیته‌ها و تنظیمات مبتنی بر همسایه).
* درختان رگرسیون بسته‌بندی شده و تقویت شده، همراه با مدل‌های جنگل تصادفی.

جزئیات نحوه تنظیم مدل‌ها در بخش محاسبات در انتهای فصل آورده شده است.

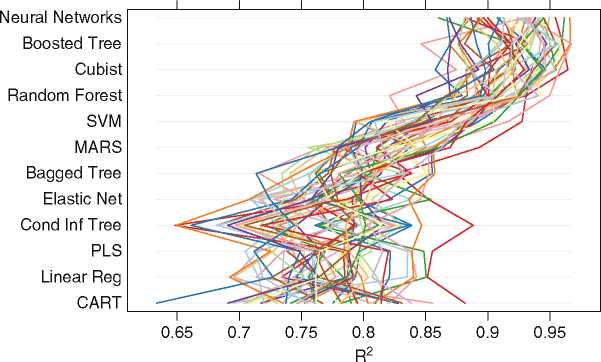
10. 2 عملکرد مدل

برای هر مدل از چین‌های اعتبارسنجی متقابل یکسانی استفاده شد. شکل [10. 2](#bookmark475) *نمودارهای مختصات موازی را* برای نتایج نمونه‌گیری مجدد در سراسر مدل‌ها نشان می‌دهد. هر خط مربوط به یک اعتبار متقابل مشترک است. از این رو، بهترین مدل‌ها شامل مجموعه‌های درختی (جنگل تصادفی و تقویت)، گروه‌های قانون (کوبیست) و شبکه‌های عصبی بودند. مدل‌های خطی و درختان ساده عملکرد خوبی نداشتند. درختان کیسه‌دار، SVMها و MARS نتایج متوسطی را نشان دادند اما به وضوح بدتر از خوشه برتر مدل‌ها هستند. میانگین *R2* \_ محدوده آماری بین 0. 76 تا 0. 92 در بین مدل‌ها بود. سه مدل برتر (بر اساس نمونه‌گیری مجدد رتبه‌بندی شده‌اند) در مجموعه آزمایشی اعمال شدند. مقادیر RMSE تقریباً با رتبه‌بندی اعتبار متقابل مطابقت دارد: 3. 9 (درخت تقویت‌شده)، 4. 2 (شبکه‌های عصبی) و 4. 5 (کوبیست).

شکل [10. 3](#bookmark476) نمودارهای داده‌های خام، پیش‌بینی‌کننده‌ها و باقیمانده‌های سه مدل را نشان می‌دهد. نمودارهای هر مدل تقریباً مشابه است. هر یک تطابق خوبی بین مقادیر مشاهده شده و پیش‌بینی شده با اندکی "فن زدن" در انتهای بالای مقاومت فشاری نشان می‌دهد. اکثریت باقیمانده ها



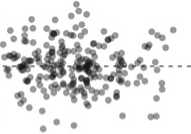
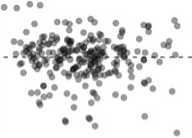
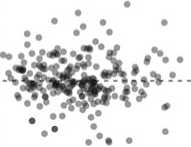
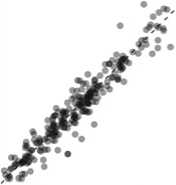
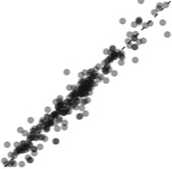
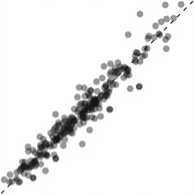
RMSE



شکل 10. 2: نمودارهای مختصات موازی برای اعتبارسنجی متقابل RMSE و R2 *در* مدل‌های مختلف. هر خط نشان دهنده نتایج برای یک ­مجموعه توقف اعتبار متقاطع مشترک است

در محدوده 2 *± هستند.* 8 مگاپاسکال با بزرگترین خطاها کمی بیشتر از 15 مگاپاسکال. هیچ برنده یا بازنده مشخصی در مدل‌های مبتنی بر این طرح‌ها وجود ندارد.

مدل شبکه عصبی از 27 واحد پنهان با ارزش فروپاشی وزن 0. 1 استفاده کرد. نمایه عملکرد برای این مدل (نمایش داده نشده است، اما می‌توان ­با استفاده از نحو ارائه شده در بخش محاسبات زیر دوباره تولید کرد) نشان داد که کاهش وزن تأثیر بسیار کمی بر اثربخشی مدل دارد. مدل نهایی کوبیستی از 100 کمیته استفاده کرد و پیش‌بینی‌کننده‌ها را ­با 3 همسایه نزدیک تنظیم کرد. مشابه نمایه‌های کوبیست نشان داده شده برای



Observed Observed Observed

80­

60­

40­

20-

20 40 60 80

Predicted

80

60

40

20

0

0 20406080

Predicted

80

60

40

20

20 40 60 80

15­

10­

5 -

0­

-5 -

10­

15-

15

10

5

0

-5

10

15

15

10

5

0

-5

10

15

20 40 60 80

Predicted

0 20406080

Predicted

20 40 60 80

پیش‌بینی کرد

Predicted

شکل 10. 3: نمودارهای تشخیصی نتایج مجموعه تست برای سه مدل. ( الف ) شبکه عصبی ( ب ) درختان تقویت شده ( ج ) کوبیست

داده‌های شیمی محاسباتی (به شکل صفحه مراجعه کنید [211 )](#bookmark434) ، زمانی که تعداد همسایگان خیلی کم یا خیلی زیاد بود، عملکرد آسیب دید. درخت تقویت شده سرعت یادگیری سریع و درختان عمیق را ترجیح می‌دهد.

10. 3 بهینه‌سازی مقاومت فشاری

شبکه عصبی و مدل‌های کوبیست برای تعیین مخلوط‌های احتمالی با مقاومت فشاری بهبود یافته استفاده شد. برای انجام این کار، می‌توان از یک روال جستجوی عددی برای یافتن فرمول‌هایی با مقاومت فشاری بالا (همانطور که توسط مدل پیش‌بینی شده است) استفاده کرد. هنگامی که مجموعه‌ای از مخلوط‌ها یافت می‌شود، ­آزمایش‌های اضافی برای این مخلوط‌ها انجام می‌شود تا تأیید شود که استحکام واقعاً بهبود یافته است. برای اهداف توضیحی، سن فرمولاسیون به مقدار 28 روز ثابت شد (تعداد زیادی از نقاط داده در مجموعه آموزشی با این مقدار وجود دارد) و فقط ترکیبات مخلوط بهینه خواهند شد.

جستجو دقیقاً چگونه باید انجام شود؟ روال‌های بهینه‌سازی عددی متعددی وجود دارد که می‌تواند فضای هفت بعدی را جستجو کند. ­بسیاری بر تعیین گرادیان (یعنی اولین مشتق) معادله پیش‌بینی تکیه می‌کنند. چندین مدل دارای معادلات پیش‌بینی صاف هستند (به‌عنوان مثال، شبکه عصبی ­و SVM). با این حال، سایرین ناپیوستگی‌های زیادی دارند (مانند مدل‌های مبتنی بر درخت و قانون و خطوط رگرسیون تطبیقی چند متغیره) که برای روش‌های جستجوی مبتنی بر گرادیان مناسب نیستند.

یک جایگزین استفاده از دسته‌ای از بهینه سازها به نام روش‌ها*ی مستقیم است* که از مشتقات برای یافتن تنظیمات با مقاومت فشاری بهینه و ارزیابی معادله پیش‌بینی بارها بیشتر از ­بهینه سازهای مبتنی بر مشتق استفاده نمی‌کنند. دو روش جستجو از این قبیل روش نلدر-مید سیمپلکس است [( Nelder and Mead 1965](#bookmark1023) ; [اولسون و نلسون 1975](#bookmark1023) ) و بازپخت شبیه‌سازی شده [( بوهاچفسکی و همکاران 1986](#bookmark1011) ). از این میان، روش جستجوی سیمپلکس بهترین نتایج را برای این داده‌ها داشت. [[21]](#footnote-21) روش Nelder-Mead این پتانسیل ­را دارد که در یک منطقه زیر بهینه از فضای جستجو "گیر" کند که می‌تواند مخلوط‌های ضعیفی ایجاد کند. برای مقابله با این موضوع، تکرار جستجو با استفاده از نقاط شروع مختلف و انتخاب جستجوهایی که با بهترین نتایج همراه هستند، معمول است. برای انجام این کار، مخلوط‌های 15-28 روزه ­از مجموعه آموزشی انتخاب شدند. اولین مورد از 15 مورد به صورت تصادفی انتخاب شد و نقاط شروع باقیمانده با استفاده از روش نمونه‌گیری حداکثر عدم تشابه مورد بحث در بخش انتخاب شدند.  [4. 3](#bookmark204) .

قبل از شروع جستجو، از محدودیت‌هایی برای جلوگیری از جستجوی بخش‌هایی از فضای فرمول‌بندی که غیرعملی یا غیرممکن بود، استفاده می‌شد. به‌عنوان مثال، مقدار آب از 5. 1٪ تا 11. 2٪ متغیر بود. روش جستجو فقط به گونه‌ای تنظیم شد که مخلوط‌هایی با حداقل 5 درصد آب در نظر گرفته شود.

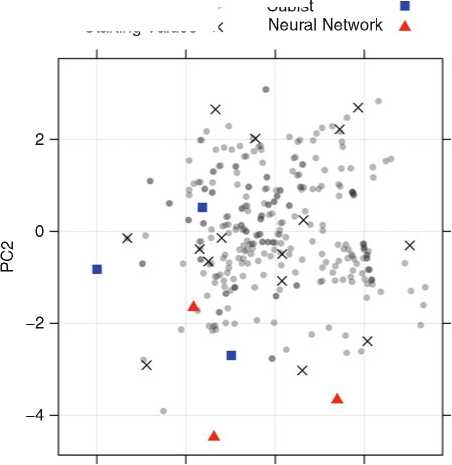
جدول 10. 2: سه مخلوط بهینه برتر پیش‌بینی شده از دو مدل که در آن سن روی مقدار 28 ثابت شده بود. در مجموعه آموزشی، مطابق با سن، قوی‌ترین مخلوط‌ها دارای مقاومت فشاری 81. 75، 79. 99 و 78. 8 بودند.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| مدل | سرباره سیمان | | آش پلاست. سی. | | Agg. اف. | Agg. پیش‌بینی آب | | |
| کوبیست |  | | | | | | | |
| میکس جدید 1 | 12. 7 | 14. 9 | 6. 8 | 0. 5 | 34. 0 | 25. 7 | 5. 4 | 89. 1 |
| میکس جدید 2 | 21. 7 | 3. 4 | 5. 7 | 0. 3 | 33. 7 | 29. 9 | 5. 3 | 88. 4 |
| میکس جدید 3 | 14. 6 | 13. 7 | 0. 4 | 2. 0 | 35. 8 | 27. 5 | 6. 0 | 88. 2 |
| شبکه عصبی |  | | | | | | | |
| میکس جدید 4 | 34. 4 | 7. 9 | 0. 2 | 0. 3 | 31. 1 | 21. 1 | 5. 1 | 88. 7 |
| میکس جدید 5 | 21. 2 | 11. 6 | 0. 1 | 1. 1 | 32. 4 | 27. 8 | 5. 8 | 85. 7 |
| میکس جدید 6 | 40. 8 | 4. 9 | 6. 7 | 0. 7 | 20. 3 | 20. 5 | 6. 1 | 83. 9 |

در مجموعه آموزشی، 416 فرمولاسیون وجود داشت که در 28 روز آزمایش شدند. از این میان، سه مخلوط برتر دارای مقاومت فشاری 81. 75، 79. 99 و 78. 8 بودند. جدول [10. 2](#bookmark479) سه مخلوط پیش‌بینی‌شده برتر را برای یک مدل صاف و غیر هموار نشان می‌دهد (به ترتیب شبکه‌های عصبی و کوبیست). مدل‌ها قادر به یافتن فرمول‌هایی هستند که پیش‌بینی می‌شود قدرت بهتری نسبت به آنچه در داده‌ها دیده می‌شود، داشته باشند.

پیش‌بینی شد که مخلوط‌های کوبیست دارای مقاومت فشاری مشابهی باشند. فرمولاسیون آنها توسط اجزای سیمان، سرباره، خاکستر و نرم کننده متمایز شد. مخلوط‌های شبکه عصبی در یک منطقه نزدیک از ­فضای ترکیبی قرار داشتند و مقادیر پیش‌بینی‌شده‌ای داشتند که کمتر از پیش‌بینی‌کننده‌های مدل کوبیست اما بزرگ‌تر از مخلوط‌های مشاهده‌شده بهترین بودند. در هر شش مورد، مخلوط‌ها نسبت آب بسیار کمی دارند. برای نمایش مخلوط مجموعه آموزشی (در فضای هفت بعدی) با استفاده از دو مولفه از تحلیل مؤلفه اصلی استفاده شد. نمودار PCA از داده‌های 28 روزه در شکل نشان داده شده است.  [10. 4 .](#bookmark480) مقادیر مؤلفه اصلی برای 15 مخلوط مورد استفاده به‌عنوان ­نقطه شروع برای روش جستجو (به‌عنوان نمادهای *x ) و 401 نقطه داده تطبیق زمانی دیگر در مجموعه آموزشی نشان داده شده است (به صورت نقاط خاکستری کوچک نشان داده شده است).* سه پیش‌بینی برتر از دو مدل نیز نشان داده شده است. بسیاری از مخلوط‌های پیش‌بینی‌شده نزدیک به حومه فضای مخلوط هستند و احتمالاً به دلیل برون‌یابی، دچار برخی نادرستی مدل می‌شوند. با توجه به این موضوع، اعتبارسنجی علمی و تجربی این فرمول‌های جدید بسیار مهم است.

همچنین می‌توان از رویکردهای پیچیده‌تری برای یافتن مخلوط‌های بهینه استفاده کرد. به‌عنوان مثال، ممکن است مهم باشد که هزینه مخلوط (یا سایر عوامل) را در جستجو لحاظ کنید. چنین بهینه‌سازی چند ­*متغیره* یا *چند* پارامتری را می‌توان به روش‌های مختلفی اجرا کرد. یک رویکرد ساده *توابع مطلوبیت است* [( درینگر و سوئیچ 1980](#bookmark1014) ; [کاستا و همکاران 2011](#bookmark1014) ). در اینجا، مهم



Cubist

2

Training Set

Starting Values x

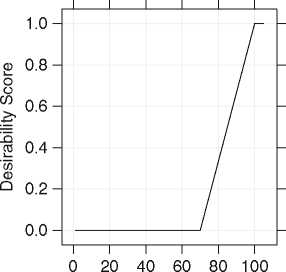
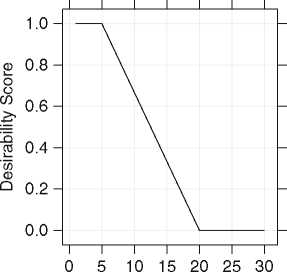
-4

-2 0

PC1

Fig. 10.4: A PCA plot of the training set data where the mixtures were aged 28 days. The search algorithm was executed across 15 different training set mixtures (shown as *x* in the plot). The top three optimal mixtures predicted from two models are also shown

ویژگی‌های یک مخلوط (به‌عنوان مثال، قدرت و هزینه) به یک مقیاس مطلوبیت مشترک بین 0 و 1 ترسیم می‌شود، جایی که یک مطلوب‌ترین و صفر کاملاً نامطلوب است. به‌عنوان مثال، مخلوط‌های بالاتر از هزینه معین ممکن است غیرقابل قبول باشند. مخلوط‌های مرتبط با هزینه‌های بالاتر از این مقدار، مطلوبیت صفر (به معنای واقعی کلمه) دارند. با کاهش هزینه، رابطه بین هزینه و مطلوبیت ممکن است به صورت خطی کاهش یابد. شکل [10. 5](#bookmark483) دو مثال فرضی از تابع مطلوبیت را برای هزینه و قدرت نشان می‌دهد. در اینجا، فرمولاسیون با هزینه‌های بیشتر از 20 و قدرت کمتر از 70 کاملا غیر قابل قبول در نظر گرفته می‌شود. هنگامی که توابع مطلوبیت توسط کاربر برای بهینه‌سازی هر مشخصه ایجاد می‌شود، مطلوبیت کلی با استفاده از میانگین هندسی ترکیب می‌شود. توجه داشته باشید که از آنجایی که میانگین هندسی مقادیر را ضرب می‌کند، اگر هر یک از تابع‌های مطلوبیت دارای امتیاز 0 باشد، همه ویژگی‌های دیگر نامربوط در نظر گرفته می‌شوند (زیرا مقدار بیش از ­همه نیز 0 است). مطلوبیت کلی با یک روش جستجو بهینه می‌شود تا راه حلی پیدا شود که تمام ویژگی‌ها را در نظر بگیرد. [واگر و همکاران ( 2010](#bookmark1027) ) و [کروز-مونته آگودو و همکاران ( 2011](#bookmark1014) ) نمونه‌هایی از این رویکرد را نشان می‌دهد.



مقاومت فشاری

Mixture Cost

شکل 10. 5: نمونه‌هایی از توابع مطلوبیت فردی برای هزینه مخلوط و مقاومت فشاری. میانگین هندسی این امتیازها را می‌توان به منظور یافتن مخلوط‌های قوی و کم هزینه بهینه کرد.

10. 4 محاسبات

این بخش از توابع بسته‌های caret، desirability، Hmisc و plyr استفاده می‌کند.

داده‌های عینی را می‌توان در مخزن یادگیری ماشین UCI یافت. بسته AppliedPredictiveModeling حاوی داده‌های اصلی (به مقدار) و یک نسخه جایگزین است که نسبت‌های مخلوط را دارد:

*> کتابخانه (AppliedPredictiveModeling)*

*داده (بتنی)*

*خ (بتن)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 'data. frame': 1030 obs. از | | | 9 متغیر: | |
| $ | سیمان : | تعداد | 540 540 332 332 199. . . |  |
| $ | سرباره کوره انفجاری: | تعداد | 0 0 142 142 132. . . |  |
| $ | FlyAsh : | تعداد | 0000000000. . . |  |
| $ | آب : | تعداد | 162 162 228 228 192 228 228 | 228 228 228 |
| $ | فوق روان کننده: | تعداد | 2. 5 2. 5 0 0 0 0 0 0 0 0 0. . . |  |
| $ | درشت کل : | تعداد | 1040 1055 932 932 978. . . |  |
| $ | FineAggregate : | تعداد | 676 676 594 594 826. . . |  |
| $ | سن : | بین المللی | 28 28 270 365 360 90 365 28 | 28 28. . . |
| قدرت فشاری $: *> str(مخلوطات)* | | تعداد | 80 61. 9 40. 3 41 44. 3. . . |  |

'data. frame': 1030 obs. از 9 متغیر:

|  |  |  |
| --- | --- | --- |
| سیمان دلار  $ BlastFurnaceSlag : | تعداد  تعداد | 0. 2231 0. 2217 0. 1492 0. 1492 0. 0853  0 0 0. 0639 0. 0639 0. 0569. . . |
| $ FlyAsh  $ آب | تعداد  تعداد | 0000000000. . .  0. 0669 0. 0665 0. 1023 0. 1023 0. 0825 |
| $ فوق روان کننده :  $ درشت کل : | تعداد  تعداد | 0. 00103 0. 00103 0 0 0. . .  0. 43 0. 433 0. 418 0. 418 0. 42. . . |

$ FineAggregate : num 0. 279 0. 278 0. 266 0. 266 0. 355. . .

$ سن : int 28 28 270 365 360 90 365 28 28 28. . .

مقاومت فشاری $: num 80 61. 9 40. 3 41 44. 3. . .

جدول [10. 1](#bookmark468) با استفاده از تابع توصیف در بسته Hmisc ایجاد شد و شکل 1.  [10. 1](#bookmark471) با استفاده از تابع featurePlot در caret ایجاد شد :

*featurePlot(x = بتن[, -9]،*

*+ y = بتن$قدرت فشاری،*

*+ ## مقداری فاصله بین پانل‌ها اضافه کنید*

*+ بین = لیست (x = 1، y = 1)،*

*+ ## یک شبکه پس زمینه (* ' *g* ' *) و یک صاف تر (* ' *صاف* ' *) اضافه کنید*

*+ نوع = c("g"، "p"، "صاف"))*

کد میانگین‌گیری مخلوط‌های تکرار شده و تقسیم داده‌ها به مجموعه‌های آموزشی و آزمایشی است

*میانگین <- ddply(مخلوط‌ها،*

*+ . (سیمان، سرباره کوره انفجاری، فلای اش، آب،*

*+ فوق روان کننده، درشت دانه،*

*+ FineAggregate، Age)،*

*+ تابع(x) c(قدرت فشاری =*

*+ میانگین (x$قدرت فشاری)))*

*set. seed(975)*

*forTraining <- createDataPartition(میانگین$CompressiveStrength،*

*+ p = 3/4)[[1]]*

*TrainingSet <- میانگین[ برای آموزش،]*

*testSet <- میانگین[-forTraining,]*

برای برازش مدل‌های خطی با مجموعه گسترده پیش‌بینی‌کننده‌ها، مانند ­کنش‌های متقابل، یک فرمول مدل خاص ایجاد شد. نقطه در فرمول زیر مختصر همه پیش‌بینی‌کننده‌ها است و (. )"2 به یک مدل با تمام عبارات خطی و همه تعاملات دو عاملی گسترش می‌یابد. عبارت‌های درجه دوم به صورت دستی ایجاد می‌شوند و در داخل تابع I() کپسوله می‌شوند. تابع "as-is" به R می‌گوید که مربع‌سازی پیش‌بینی‌کننده‌ها باید به صورت حسابی (و نه نمادین) انجام شود.

فرمول ابتدا به‌عنوان یک رشته کاراکتر با استفاده از دستور paste ایجاد می‌شود، سپس به فرمول R واقعی تبدیل می‌شود.

*modFormula <- paste("Compressive Strength ~ (. )~2 + I(Cement~2) + ",*

*+ "I(BlastFurnaceSlag~2) + I(FlyAsh~2) + I(Water~2) +"*

*+ "I(Superplasticizer~2) + I(CoarseAggregate"2) + ",*

*+ "I(FineAggregate~2) + I(Age~2)")*

*modFormula <- as. formula(modFormula)*

هر مدل از اعتبارسنجی متقابل 10 برابری مکرر استفاده می‌کند و با تابع trainControl مشخص می‌شود:

*> controlObject <- trainControl(method = "repeatedcv",*

*+ تکرار = 5،*

*+ عدد = 10)*

برای ایجاد چین‌های دقیقاً مشابه، مولد اعداد تصادفی قبل از شروع آموزش به یک دانه مشترک بازنشانی می‌شود. به‌عنوان مثال، برای برازش مدل رگرسیون خطی:

*> set. seed(669)*

*> linearReg <- train(modFormula,*

|  |  |
| --- | --- |
| *+*  *+*  *+*  *> linearReg*  745 نمونه  44 پیش‌بینی | *داده = trainingSet، روش = "lm"، trControl = controlObject)* |

بدون پیش پردازش

نمونه‌گیری مجدد: اعتبارسنجی متقاطع (10 برابر، 5 بار تکرار)

خلاصه حجم نمونه: 671، 671، 672، 670، 669، 669،.

نتایج نمونه‌گیری مجدد

|  |  |
| --- | --- |
| RMSE Rsquared 7. 85 0. 771 | RMSE SD Rsquared SD  0. 647 0. 0398 |
| خروجی فرمول مدل را نشان می‌دهد | که از 44 پیش‌بینی استفاده شد که نشان‌دهنده استفاده گسترده است. |

دو مدل خطی دیگر با:

*> set. seed(669)*

*> plsModel <- train(modForm, data = trainingSet,*

|  |  |
| --- | --- |
| *+*  *+*  *+*  *+* | *روش = "pls"*  *preProc = c ("مرکز"، "مقیاس")، طول = 15،*  *trControl = controlObject)* |

*> enetGrid <- expand. grid(. lambda = c(0,. 001,. 01,. 1)*

|  |  |
| --- | --- |
| *+*  *> set. seed(669)* | *کسر = seq(0. 05، 1، طول = 20))* |

*> enetModel <- train(modForm, data = trainingSet,*

|  |  |
| --- | --- |
| *+*  *+*  *+*  *+* | *روش = "enet"،*  *preProc = c ("مرکز"، "مقیاس")، tuneGrid = enetGrid، trControl = controlObject)* |

MARS، شبکه‌های عصبی و SVM‌ها به شرح زیر ایجاد شدند:

*> set. seed(669)*

*> earthModel <- train(CompressiveStrength ~. , data = trainingSet,*

|  |  |
| --- | --- |
| *+*  *+*  *+*  *+*  *> set. seed(669)* | *روش = "زمین"،*  *tuneGrid = expand. grid(. degree = 1,*  *nprune = 2:25)، trControl = controlObject)* |

*> svmRMmodel <- train(CompressiveStrength ~. , data = trainingSet,*

|  |  |
| --- | --- |
| *+*  *+*  *+* | *متد = "svmRadial"، tuneLength = 15، preProc = c("مرکز"، "مقیاس")،* |

|  |  |
| --- | --- |
| *+*  *> nnetGrid <- e*  *+*  *+*  *> set. seed(669)*  *> nnetModel <­*  *+*  *+*  *+*  *+*  *+*  *+*  *+*  *+* | *trControl = controlObject) xpand. grid(. decay = c(0. 001،. 01،. 1)،. size = seq(1، 27، توسط = 2)،. bag = FALSE)*  *train(CompressiveStrength ~. ، data = trainingSet، روش = "avNNet"، tuneGrid = nnetGrid، preProc = c("center"، "scale")، lineout = TRUE، trace = FALSE، maxit = 1000، trControl = controlObject)* |
| پسرفت | و درختان مدل به‌طور مشابه ایجاد شدند: |
| *> set. seed(669)*  *> rpartModel <­*  *+*  *+*  *+*  *+* | *Train (CompressiveStrength ~. ، data = trainingSet، روش = "rpart"، tuneLength = 30، trControl = controlObject)* |

|  |  |
| --- | --- |
| *> set. seed(669)*  *> ctreeModel <­*  *+*  *+*  *+*  *+* | *train(CompressiveStrength ~. ، data = trainingSet، متد = "ctree"، tuneLength = 10، trControl = controlObject)* |

|  |  |
| --- | --- |
| *> set. seed(669)*  *> mtModel <- tr*  *+*  *+*  *+* | *ain (CompressiveStrength ~. ، داده = trainingSet، متد = "M5"، trControl = controlObject)* |
| موارد زیر | کد g بقیه اشیاء مدل را ایجاد می‌کند: |
| *> set. seed(669)*  *> treebagModel*  *+*  *+*  *+*  *> set. seed(669)*  *rfModel <- tr*  *+*  *+*  *+*  *+*  *+*  *+*  *gbmGrid <- ex +* | *<- train(CompressiveStrength ~. ، data = trainingSet، method = "treebag"، trControl = controlObject)*  *ain(استحکام فشاری ~. ,*  *داده = مجموعه آموزشی،*  *روش = "rf"،*  *طول کوک = 10،*  *nدرخت = 1000،*  *اهمیت = درست،*  *trControl = controlObject) pand. grid(. interaction. depth = seq(1, 7, by = 2),. n. trees = seq(100, 1000, by = 50),* |

*+ انقباض = c(0. 01، 0. 1))*

*> set. seed(669)*

*> gbmModel <- train(CompressiveStrength ~. ,*

*+ داده = مجموعه آموزشی،*

*+ روش = "gbm"،*

*+ tuneGrid = gbmGrid،*

*+ پرمخاطب = نادرست،*

*+ trControl = controlObject)*

*> cubistGrid <- expand. grid(. committees = c(1, 5, 10, 50, 75, 100)*

*+ . neighbors = c(0، 1، 3، 5، 7، 9))*

*> set. seed(669)*

*> cbModel <- train(Compressive Strength ~. ,*

*+ داده = مجموعه آموزشی،*

*+ روش = "کوبیست"،*

*+ tuneGrid = cubistGrid،*

*+ trControl = controlObject)*

نتایج نمونه‌گیری مجدد برای این مدل‌ها با استفاده از تابع نمونه‌گیری مجدد caret در یک شی واحد جمع‌آوری شد. سپس می‌توان از این شی برای تجسم ­یا مقایسه‌های رسمی بین مدل‌ها استفاده کرد.

*> allResamples <- resamples(list("Linear Reg" = lmModel,*

*"PLS" = plsModel،*

*+*

*+*

*+*

*+*

*+*

*+*

*+*

*+*

*+*

*+*

*+*

*"Elastic Net" = enetModel، MARS = earthModel،*

*SVM = svmRMmodel،*

*"شبکه‌های عصبی" = nnetModel، CART = rpartModel،*

*"Cond Inf Tree" = ctreeModel، "Bagged Tree" = treebagModel، "Boosted Tree" = gbmModel، "Random Forest" = rfModel، Cubist = cbModel))*

شکل [10. 2](#bookmark475) از این شی به‌عنوان ایجاد شد

*## مقادیر RMSE را رسم کنید*

*parallelPlot (allResamples)*

*## با استفاده از R-squared:*

*parallellplot (allResamples، معیار = "Rsquared")*

از نتایج نمونه‌گیری مجدد نیز می‌تواند ایجاد شود ( برای سایر گزینه‌ها به ?xyplot. resamples مراجعه کنید).

پیش‌بینی‌کننده‌های مجموعه تست با استفاده از یک کاربرد ساده از تابع پیش‌بینی به دست می‌آیند :

*nnetPredictions <- predict(nnetModel، testData)*

*gbmPredictions <- predict(gbmModel, testData)*

*cbPredictions <- predict(cbModel, testData)*

برای پیش‌بینی مخلوط‌های بهینه، ابتدا از داده‌های ۲۸ روزه برای تولید مجموعه‌ای از نقاط شروع تصادفی از مجموعه آموزشی استفاده می‌کنیم.

از آنجایی که فاصله بین فرمول‌ها به‌عنوان معیاری برای عدم تشابه استفاده می‌شود، داده‌ها از قبل پردازش می‌شوند تا میانگین و واریانس یکسانی برای هر پیش‌بینی داشته باشند. پس از این، یک مخلوط تصادفی انتخاب می‌شود تا فرآیند نمونه‌گیری حداکثر عدم تشابه را اولیه کند:

*age28Data <- زیر مجموعه (trainingData، Age == 28)*

*## ستون‌های سن و مقاومت فشاری را حذف کنید و*

*## سپس ستون‌های پیش‌بینی را مرکز و مقیاس کنید*

*pp1 <- preProcess(age28Data[, -(8:9)]، c("مرکز"، "مقیاس"))*

*scaledTrain <- predict(pp1, age28Data[, 1:7])*

*set. seed(91)*

*startMixture <- نمونه (1:nrow(age28Data)، 1)*

*شروع کننده <- scaledTrain[startMixture, 1:7]*

پس از این، روش نمونه‌گیری حداکثر عدم تشابه از بخش.  [4. 3](#bookmark204) 14 مخلوط دیگر را برای تکمیل مجموعه متنوعی از نقاط شروع برای الگوریتم‌های جستجو انتخاب می‌کند:

*استخر <- scaledTrain*

*index <- maxDissim(شروع کننده، استخر، 14)*

*نقطه شروع <- c(startMixture، index)*

*مبتدیان <- age28Data[startPoints,1:7]*

از آنجایی که هر هفت نسبت مخلوط باید به یک اضافه شود، روش‌های جستجو جستجو را بدون یک عنصر (آب) انجام می‌دهند و نسبت آب ­با مجموع نسبت‌های شش ماده دیگر تعیین می‌شود. بدون این مرحله، روش‌های جستجو مقادیر ترکیبی کاندیدایی را انتخاب می‌کنند که به یک اضافه نمی‌شوند.

*## آب را حذف کنید*

*startingValues <- starters[, -4]*

برای به حداکثر رساندن مقاومت فشاری، تابع R بهینه فضای مخلوط را برای فرمولاسیون بهینه جستجو می‌کند. یک تابع R سفارشی برای ترجمه یک ترکیب کاندید به یک پیش‌بینی مورد نیاز است. این تابع می‌تواند تنظیماتی را برای به *حداقل رساندن* یک تابع پیدا کند، بنابراین منفی مقاومت فشاری را برمی گرداند. تابع زیر بررسی می‌کند که (الف) نسبت‌ها بین 0 و 1 باشد و (ب) نسبت آب کمتر از 5٪ نباشد. اگر این شرایط نقض شوند، تابع یک عدد مثبت بزرگ را برمی‌گرداند که روند جستجو از آن اجتناب می‌کند (همانطور که بهینه برای کمینه‌سازی است).

*## ورودی‌های تابع بردار شش نسبت مخلوط هستند*

*## (در آرگومان* ' *x* ' *) و مدل مورد استفاده برای پیش‌بینی (* ' *mod* ' *)*

*> modelPrediction <- تابع (x، mod)*

*+{*

*+ ## بررسی کنید تا مطمئن شوید که نسبت مخلوط است*

*+ ## در محدوده صحیح*

*+ if(x[1] < 0 I x[1] > 1) return(10~38)*

*+ if(x[2] < 0 I x[2] > 1) return(10~38)*

*+ if(x[3] < 0 I x[3] > 1) return(10338)*

*+ if(x[4] < 0 I x[4] > 1) return(10338)*

*+ if(x[5] < 0 I x[5] > 1) return(10338)*

*+ if(x[6] < 0 I x[6] > 1) return(10338)*

*+*

*+ ## نسبت آب را تعیین کنید*

*+ x <- c(x، 1 - sum (x)) +*

*+ ## محدوده آب را بررسی کنید*

*+ if(x[7] < 0. 05) return(10"38)*

*+*

*+ ## بردار را به یک قاب داده تبدیل کنید، نام‌ها را اختصاص دهید*

*+ ## و تعیین سن در 28 روز*

*+ tmp <- as. data. frame(t(x))*

*+ names(tmp) <- c(* ' *Ciment* '،' *BlastFurnaceSlag* '،' *FlyAsh* ' *,*

*+*  ' *Superplasticizer* ' *,* ' *CarseAggregate* ' *,*

*+*  " *FineAggregate* "،" *Water* " *)*

*+ tmp$سن <- 28*

*+ ## پیش‌بینی مدل را دریافت کنید، آنها را مربع کنید تا به حالت اولیه برگردید*

*+ ## واحد اصلی، سپس منفی نتیجه را برگردانید*

*+ -predict (mod، tmp)*

*+}*

ابتدا از مدل کوبیست استفاده می‌شود:

*cbResults <- startingValues*

*cbResults$Water <- NA*

*cbResults$Prediction <- NA*

*## روی هر نقطه شروع حلقه بزنید و جستجو را انجام دهید*

*for(i در 1:nrow(cbResults))*

*+{*

*+ نتایج <- optim(unlist(cbResults[i,1:6])،*

*+ مدل پیش‌بینی،*

*+ روش = "نلدر مید"،*

*+ ## استفاده از روش =* ' *SANN* ' *برای بازپخت شبیه‌سازی شده*

*+ control=list(maxit=5000)*

*+ ## گزینه بعدی به قسمت ارسال می‌شود*

*+ ## تابع modelPrediction().*

*+ mod = cbModel)*

*+ ## مقاومت فشاری پیش‌بینی‌شده را ذخیره کنید*

*+ cbResults$Prediction[i] <- -results$value*

*+ ## همچنین مقادیر نهایی مخلوط را ذخیره کنید*

*+ cbResults[i,1:6] <- results$par*

*+}*

*## نسبت آب را محاسبه کنید*

*cbResults$Water <- 1 - اعمال (cbResults[,1:6]، 1، مجموع)*

*## سه مخلوط بالا را نگه دارید*

*cbResults <- cbResults[order(-cbResults$Prediction)،][1:3،]*

*cbResults$Model <- "کوبیست"*

سپس از همان فرآیند برای مدل شبکه عصبی استفاده می‌کنیم:

*nnetResults <- startingValues*

*nnetنتایج$آب <- NA*

*nnetResults$Prediction <- NA*

*for(i در 1:nrow(nnetResults))*

*+{*

*+ نتایج <- optim(unlist(nnetResults[i, 1:6,])،*

*+ مدل پیش‌بینی،*

*+ روش = "نلدر مید"،*

*+ control=list(maxit=5000)*

*+ mod = nnetModel)*

*+ nnetResults$Prediction[i] <- -results$value*

*+ nnetResults[i,1:6] <- results$par*

*+}*

*nnetResults$Water <- 1 - اعمال (nnetResults[,1:6]، 1، مجموع)*

*nnetResults <- nnetResults[order(-nnetResults$Prediction)،][1:3،]*

*nnetResults$Model <- "NNet"*

برای ایجاد شکل [10. 4 ،](#bookmark480) PCA روی مخلوط‌های 28 روزه انجام شد و شش مخلوط پیش‌بینی‌شده پیش‌بینی شدند. اجزاء ترکیب شده و رسم می‌شوند:

*## PCA را روی داده در 28 روز اجرا کنید*

*pp2 <- preProcess(age28Data[, 1:7]، "pca")*

*## اجزای این مخلوط‌ها را دریافت کنید*

*pca1 <- پیش‌بینی (pp2، age28Data[, 1:7])*

*pca1$Data <- "Training Set"*

*## برچسب گذاری کنید که کدام نقاط داده برای شروع جستجوها استفاده شده است*

*pca1$Data[startPoints] <- "Starting Values"*

*## مخلوط‌های جدید را به همین ترتیب پخش کنید (مطمئن شوید*

*## ستون‌ها را دوباره مرتب کنید تا با ترتیب شی age28Data مطابقت داشته باشند.*

*pca3 <- predict(pp2، cbResults[، names(age28Data[, 1:7])])*

*pca3$Data <- "کوبیست"*

*pca4 <- predict(pp2، nnetResults[, names(age28Data[, 1:7])])*

*pca4$Data <- "شبکه عصبی"*

*## داده‌ها را ترکیب کنید، محدوده محورها را تعیین کنید و رسم کنید*

*pcaData <- rbind(pca1, pca3, pca4)*

*pcaData$Data <- factor(pcaData$Data,*

*+ سطوح = c("مجموعه آموزشی"، "مقدارهای شروع"،*

*+ "کوبیست"، "شبکه عصبی"))*

*lim <- extension(pcaData[, 1:2])*

*xyplot(PC2 ~ PC1، داده = pcaData، گروه‌ها = داده،*

*+ auto. key = لیست (ستون‌ها = 2)،*

*+ xlim = لیم، یلیم = لیم،*

*+ نوع = c("g"، "p"))*

توابع مطلوبیت را می‌توان با بسته مطلوبیت محاسبه کرد. از توابع dMin و dMax می‌توان برای ایجاد تعاریف منحنی تابع مطلوبیت برای حداقل‌سازی و حداکثر‌سازی استفاده کرد.

قسمت سوم

مدل‌های طبقه بندی

فصل 11

سنجش عملکرد در طبقه بندی

مدل ها

در بخش قبلی این کتاب، ما بر ساخت و ارزیابی مدل‌هایی برای پاسخ پیوسته تمرکز کردیم. اکنون بر ساختن و ارزیابی مدل‌ها برای یک پاسخ طبقه‌بندی تمرکز می‌کنیم. اگرچه بسیاری از تکنیک‌های مدل‌سازی رگرسیون نیز می‌توانند برای طبقه‌بندی استفاده شوند، روشی که ما عملکرد مدل را ارزیابی می‌کنیم لزوماً بسیار متفاوت است زیرا معیارهایی مانند RMSE و *R2* در چارچوب طبقه‌بندی مناسب نیستند. ما این بخش از کتاب را با بحث در مورد معیارهای ارزیابی ­عملکرد مدل طبقه‌بندی آغاز می‌کنیم. در بخش اول این فصل، نگاهی عمیق به جنبه‌های مختلف پیش‌بینی‌کننده‌های مدل طبقه‌بندی و چگونگی ارتباط این پیش‌بینی‌کننده‌ها با سؤال مورد علاقه می‌اندازیم. دو بخش بعدی به بررسی استراتژی‌هایی برای ارزیابی مدل‌های طبقه‌بندی با استفاده از آمار و تجسم می‌پردازد.

11. 1 پیش‌بینی کلاس

مدل‌های طبقه‌بندی معمولاً دو نوع پیش‌بینی ایجاد می‌کنند. مانند مدل‌های رگرسیون، مدل‌های طبقه‌بندی یک پیش‌بینی با مقدار پیوسته تولید می‌کنند که معمولاً به شکل احتمال است (یعنی مقادیر پیش‌بینی‌شده عضویت در کلاس برای هر نمونه فردی بین ۰ و ۱ و مجموع تا ۱ است). علاوه بر پیش‌بینی پیوسته، مدل‌های طبقه‌بندی یک کلاس پیش‌بینی‌شده را تولید می‌کنند که به شکل یک دسته‌بندی گسسته می‌آید. برای اکثر کاربردهای عملی، یک پیش‌بینی دسته‌بندی گسسته برای تصمیم‌گیری مورد نیاز است. به‌عنوان مثال، فیلتر خودکار هرزنامه‌ها نیاز به یک قضاوت قطعی برای هر ایمیل دارد.

اگرچه مدل‌های طبقه‌بندی هر دو نوع پیش‌بینی را تولید می‌کنند، اغلب تمرکز بر پیش‌بینی گسسته به جای پیش‌بینی پیوسته ­است. با این حال، تخمین‌های احتمال برای هر کلاس می‌تواند برای سنجش اطمینان مدل در مورد طبقه‌بندی پیش‌بینی‌شده بسیار مفید باشد. با بازگشت به مثال فیلتر ایمیل هرزنامه، یک پیام ایمیل با احتمال پیش‌بینی شده ­هرزنامه 0. 51 به‌عنوان یک پیام با طبقه‌بندی می‌شود.

-11،

© احتمال پیش‌بینی شده هرزنامه 0. 99 است. در حالی که فیلتر با هر دو پیام یکسان برخورد می‌کند، ما اطمینان بیشتری خواهیم داشت که ­پیام دوم در واقع، واقعاً هرزنامه بوده است. به‌عنوان مثال دوم، ساخت مدلی برای طبقه‌بندی مولکول‌ها بر اساس وضعیت ایمنی *درون تنی (*به‌عنوان مثال، غیر سمی، ضعیف سمی و شدیدا سمی) در نظر بگیرید.  [پیرسما و همکاران 2004](#bookmark1023) ). یک مولکول با احتمالات پیش‌بینی شده در هر دسته سمیت مربوطه 0. 34، 0. 33 و 0. 33، به‌عنوان یک مولکول با احتمالات پیش‌بینی شده مربوطه 0. 98، 0. 01 و 0. 01 طبقه‌بندی می‌شود. با این حال، در این مورد، ما بسیار مطمئن تر هستیم که مولکول دوم در مقایسه با مولکول اول غیر سمی است.

در برخی کاربردها، نتیجه مورد نظر، احتمالات کلاس پیش‌بینی‌شده است ­که سپس به‌عنوان ورودی برای محاسبات دیگر استفاده می‌شود. شرکت بیمه‌ای را در نظر بگیرید که می‌خواهد ادعاهای تقلبی را کشف و تحت پیگرد قانونی قرار دهد. با استفاده از داده‌های ادعاهای تاریخی، می‌توان یک مدل طبقه‌بندی برای پیش‌بینی احتمال تقلب در ادعا ایجاد کرد. سپس این احتمال با هزینه‌های تحقیق شرکت و زیان پولی احتمالی ترکیب می‌شود تا مشخص شود که آیا پیگیری تحقیقات به نفع مالی شرکت بیمه است یا خیر. به‌عنوان مثال دیگری از احتمالات طبقه‌بندی به‌عنوان ورودی‌های ­مدل بعدی، محاسبه ارزش طول عمر مشتری (CLV) را در نظر بگیرید که به‌عنوان مقدار سود مرتبط با مشتری در یک دوره زمانی تعریف می‌شود [( گوپتا و همکاران 2006](#bookmark1017) ). برای تخمین CLV، مقادیر مختلفی مورد نیاز است، از جمله مبلغ پرداخت شده توسط مصرف کننده در یک بازه زمانی معین، هزینه خدمات رسانی به مصرف کننده و احتمال اینکه مصرف کننده در بازه زمانی یک خرید انجام دهد.

همانطور که در بالا ذکر شد، اکثر مدل‌های طبقه‌بندی احتمالات کلاس پیش‌بینی شده را تولید می‌کنند. با این حال، زمانی که برخی از مدل‌ها برای طبقه‌بندی استفاده می‌شوند، مانند ­شبکه‌های عصبی و حداقل مربعات جزئی، پیش‌بینی‌کننده‌های پیوسته‌ای تولید می‌کنند که از تعریف احتمال پیروی نمی‌کنند - مقادیر پیش‌بینی‌شده لزوماً بین ۰ و ۱ نیستند و مجموع ۱ نیستند. به‌عنوان مثال، یک مدل طبقه‌بندی حداقل مربعات جزئی (با جزئیات بیشتر در بخش توضیح داده شده است.  [12. 4 )](#bookmark572) 0/1 متغیرهای ساختگی را برای هر کلاس ایجاد می‌کند و به‌طور همزمان این مقادیر را به‌عنوان تابعی از پیش‌بینی‌کننده‌ها مدل می‌کند. وقتی نمونه‌ها پیش‌بینی می‌شوند، پیش‌بینی‌کننده‌های مدل تضمین نمی‌شوند که در 0 و 1 باشند. برای مدل‌های طبقه‌بندی ­مانند این، باید از یک تبدیل برای وادار کردن پیش‌بینی‌کننده‌ها به مقادیر «احتمال‌مانند» استفاده شود تا بتوان آنها را تفسیر و استفاده کرد. کاتیون طبقه‌بندی ­یکی از این روش‌ها *تبدیل softmax است* [( Bridle 1990](#bookmark1012) ) که به این صورت تعریف شده است

E C =i e y l

که در آن *y £* پیش‌بینی مدل عددی برای کلاس *f t h* و *p \** مقدار تبدیل شده بین 0 و 1 است. فرض کنید که یک نتیجه دارای سه کلاس است و یک مدل PLS مقادیر *y* 1 = 0 را پیش‌بینی می‌کند. 25، *y* 2 = 0. 76 و *y* 3 *=* -0 1. تابع softmax این مقادیر را به *p* l = 0 تبدیل می‌*کند.* 30، *p \** = 0. 49 و *p* > 3 = 0. 21. برای روشن بودن، هیچ گزاره احتمالی با این تبدیل ایجاد نمی‌شود. این فقط تضمین می‌کند که پیش‌بینی‌کننده‌ها همان کیفیت‌های ریاضی احتمالات را دارند.

احتمالات به خوبی کالیبره شده

این که آیا یک مدل طبقه‌بندی برای پیش‌بینی ایمیل‌های هرزنامه، وضعیت سمیت مولکول، یا به‌عنوان ورودی برای تقلب بیمه یا محاسبات ارزش طول عمر مشتری استفاده می‌شود، ما آرزو داریم که احتمالات طبقه‌بندی تخمین زده شده منعکس کننده احتمال واقعی نمونه باشد. یعنی، احتمال کلاس پیش‌بینی‌شده (یا مقدار مشابه احتمال) باید به خوبی کالیبره شود. برای اینکه به خوبی کالیبره شوند، احتمالات باید به‌طور موثر احتمال واقعی رویداد مورد علاقه را منعکس کنند. با بازگشت به تصویر فیلتر هرزنامه، اگر مدلی یک احتمال یا مقدار احتمال مشابه 20 درصد برای احتمال هرزنامه بودن یک ایمیل خاص تولید کند، آنگاه این مقدار به خوبی کالیبره می‌شود اگر نوع پیام‌های مشابه واقعاً باشد. از آن کلاس به‌طور متوسط در 1 نمونه از 5 نمونه.

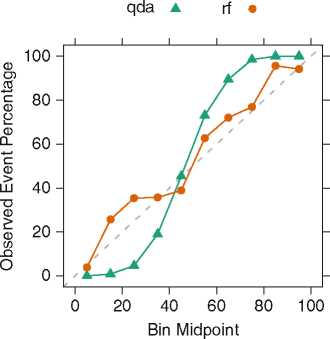
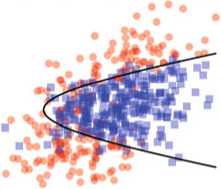
یکی از راه‌های ارزیابی کیفیت احتمالات کلاس استفاده از *نمودار کالیبراسیون است.* برای مجموعه‌ای معین از داده‌ها، این نمودار مقداری اندازه‌گیری از احتمال مشاهده شده یک رویداد در مقابل احتمال کلاس پیش‌بینی شده را نشان می‌دهد. یکی از روش‌های ایجاد این تجسم، امتیازدهی به مجموعه‌ای از نمونه‌ها با نتایج شناخته شده (ترجیحاً یک مجموعه آزمایشی) با استفاده از یک مدل طبقه‌بندی است. گام بعدی این است که داده‌ها را بر اساس احتمالات کلاس آنها در گروه‌ها قرار دهید. به‌عنوان مثال، مجموعه‌ای از سطل‌ها ممکن است [0, 10 %]، (10 %, 20 %]،. *. .* , (90 , 100 %] باشد. برای هر سطل، نرخ رویداد مشاهده شده را تعیین کنید. فرض کنید 50 نمونه برای احتمالات کلاس کمتر از 10 درصد به داخل سطل افتاد و یک رویداد واحد وجود داشت. نقطه میانی سطل 5 درصد و نرخ رویداد مشاهده شده 2 درصد است. نمودار کالیبراسیون نقطه وسط bin را روی *x نشان می‌دهد* - محور و نرخ رویداد مشاهده شده در محور *y.* اگر نقاط در امتداد یک خط 45 *درجه قرار گیرند،* مدل احتمالات به خوبی کالیبره شده را تولید کرده است.

به‌عنوان مثال، یک مجموعه داده به گونه‌ای شبیه‌سازی شد که احتمالات واقعی رویداد مشخص باشد. برای دو کلاس (کلاس 1 و 2) و دو پیش‌بینی ( *A* و *B* )، احتمال واقعی ( *p* ) رویداد از معادله ایجاد می‌شود:

log ( *—— =* = *-* 1 *-* 2 *A -.* 2 *A 2* + 2 *B* 2

شکل [11. 1](#bookmark502) یک مجموعه آزمایش شبیه‌سازی شده را به همراه خط یک خط برای *p* = 0 *نشان می‌دهد.* 50 احتمال رویداد دو مدل برای مجموعه آموزشی مناسب بودند: تحلیل تفکیک درجه دوم (QDA، بخش.  [13. 1 )](#bookmark612) و یک مدل جنگل تصادفی (بخش.  [14. 4 )](#bookmark702) . یک مجموعه آزمایشی از *1000* نمونه برای امتیازدهی به مدل و ایجاد نمودار کالیبراسیون که در شکل نشان داده شده است استفاده شد.  [11. 1](#bookmark502) . هر دو مدل طبقه‌بندی ­دقت مشابهی برای مجموعه آزمایشی دارند (حدود 87. 1٪ برای هر مدل). نمودار کالیبراسیون نشان می‌دهد که احتمالات کلاس QDA در مقایسه با مدل جنگل تصادفی ضعیف عمل می‌کند. به‌عنوان مثال، در سطل زباله با احتمالات کلاس از 20 تا 30 درصد، درصد مشاهده شده رویدادها برای QDA 4. 6 درصد بود، به مراتب کمتر از درصد در مدل جنگل تصادفی (35. 4 درصد).

احتمالات کلاس را می‌توان برای انعکاس دقیق تر احتمال مشابه ­رویداد (یا حداقل احتمال مشاهده شده در داده‌های واقعی) کالیبره کرد.



4-

-2

**J I I I I L**

4­

2­

0-

•

2-

-1

Predictor A

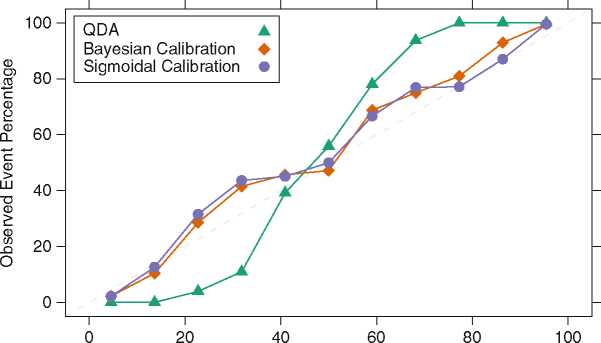
شکل 11. 1: *سمت چپ* : مجموعه داده‌های دو کلاسه شبیه‌سازی شده با دو پیش‌بینی. خط *مشکی* یکپارچه نشان دهنده کانتور احتمال 50٪ است. *راست* : نمودار کالیبراسیون احتمالات مجموعه آزمون برای مدل‌های تحلیل تفکیک‌کننده درجه دوم و جنگل تصادفی

به‌عنوان مثال، شکل.  [11. 1](#bookmark502) یک الگوی سیگموئیدی را نشان می‌دهد به‌طوری که مدل QDA احتمال رویداد را زمانی که احتمال واقعی نسبتاً زیاد یا کم است، کمتر پیش‌بینی می‌کند. می‌توان یک مدل اضافی برای تنظیم این الگو ایجاد کرد. یکی از معادله‌ای که با این الگوی سیگموئیدی سازگار است، مدل رگرسیون لجستیک است (شرح شده در بخش.  [12. 2 )](#bookmark557) . پیش‌بینی‌کننده‌های کلاس و مقادیر نتیجه واقعی از مجموعه آموزشی را می‌توان برای پس پردازش ­تخمین‌های احتمالی با فرمول زیر استفاده کرد [( Platt 2000](#bookmark1023) ):

که در آن پارامترهای ( *با پیش‌بینی کلاس‌های واقعی به‌عنوان تابعی از احتمالات کلاس کالیبره‌نشده تخمین زده می‌شوند. برای* مدل *QDA،* این فرآیند به تخمین‌هایی منجر شد (*3 0* = *-* 5. 7 و ( *1 =* 11. *7.* شکل [11. 2](#bookmark505) نتایج را برای نمونه‌های مجموعه آزمایشی با استفاده از این روش تصحیح نشان می‌دهد. نتایج نشان‌دهنده بهبود کالیبراسیون با داده‌های مجموعه تست است. روش دیگر، یک کاربرد از قانون بیز (مدل توصیف شده بخش است.  [13. 6 )](#bookmark648) می تواند به‌طور مشابه برای ­محاسبه مجدد پیش‌بینی‌کننده‌ها اعمال شود. رویکرد بیزی نیز پیش‌بینی‌کننده‌ها را بهبود می‌بخشد (شکل 1).  [11. 2 )](#bookmark505) . توجه داشته باشید که پس از کالیبراسیون، نمونه‌ها باید مجدداً طبقه‌بندی شوند تا از سازگاری بین احتمالات جدید و طبقات پیش‌بینی شده اطمینان حاصل شود.

1 +exp(*—(*o *- (*i*p)*

(11.1)



بین نقطه

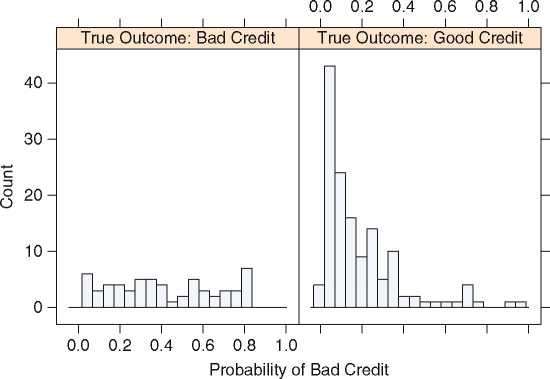
شکل 11. 2: احتمالات کلاس QDA اصلی و نسخه‌های کالیبره شده با استفاده از دو روش مختلف

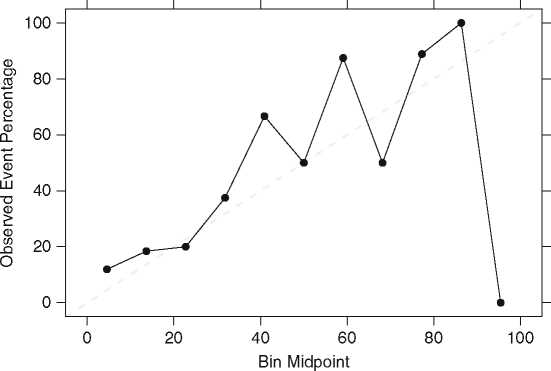
ارائه احتمالات کلاس

تجسم احتمالات کلاس یک روش موثر برای ­ارتباط نتایج مدل است. برای دو کلاس، هیستوگرام کلاس‌های پیش‌بینی شده برای هر یک از نتایج واقعی، نقاط قوت و ضعف یک مدل را نشان می‌دهد. در فصل [4](#bookmark192) مثال امتیازدهی اعتباری را معرفی کردیم. دو مدل طبقه‌بندی ­برای پیش‌بینی کیفیت اعتبار مشتری ایجاد شد: ماشین بردار پشتیبانی (SVM) و رگرسیون لجستیک. از آنجایی که عملکرد دو مدل تقریباً معادل بود، مدل رگرسیون لجستیک به دلیل سادگی مورد علاقه بود. پانل بالای شکل.  [11. 6](#bookmark522) هیستوگرام احتمالات مجموعه آزمون را برای مدل رگرسیون لجستیک نشان می‌دهد (پانل‌ها وضعیت اعتبار واقعی را نشان می‌دهند). احتمال اعتبار بد برای مشتریانی که اعتبار خوب دارند، توزیع کج‌رویی را نشان می‌دهد که در آن ­روابط احتمالی اکثر مشتریان بسیار کم است. در مقابل، احتمالات برای مشتریان دارای اعتبار بد ثابت (یا توزیع یکنواخت) است که نشان دهنده ناتوانی مدل در تشخیص موارد اعتبار بد است.

این شکل همچنین نمودار کالیبراسیون این داده‌ها را نشان می‌دهد. دقت احتمال اعتبار بد با بزرگتر شدن آن کاهش می‌یابد تا جایی که هیچ نمونه‌ای با اعتبار بد با احتمال بالای 82. 7 درصد پیش‌بینی نشد. این الگو نشان دهنده مدلی است که هم کالیبراسیون ضعیف و هم عملکرد ضعیف دارد.

هنگامی که سه کلاس یا بیشتر وجود دارد، یک *نقشه حرارتی* از احتمالات کلاس می‌تواند به سنجش اطمینان در پیش‌بینی‌کننده‌ها کمک کند. شکل [11. 4](#bookmark507) نتایج مجموعه آزمون را با هشت کلاس (نشان دهنده A تا I ) و 48 نمونه نشان می‌دهد. را





شکل 11. 3: *بالا* : هیستوگرام برای مجموعه‌ای از احتمالات مرتبط با اعتبار بد. این دو پنل مشتریان را بر اساس طبقه واقعی خود تقسیم می‌کنند. *پایین* : نمودار کالیبراسیون برای این احتمالات

کلاس‌های true در ردیف‌ها نشان داده شده اند (همراه با شناسه‌های نمونه) و ستون‌ها احتمالات کلاس را منعکس می‌کنند. در برخی موارد، مانند نمونه 20، یک سیگنال واضح مرتبط با کلاس پیش‌بینی شده وجود داشت (احتمال کلاس C 78. 5٪ بود)، در حالی که در موارد دیگر، وضعیت مبهم است. نمونه 7 را در نظر بگیرید. چهار احتمال بزرگ (و طبقات مرتبط) 19. 6٪ ( B )، 19. 3٪ ( C )، 17. 7٪ ( A ) و 15٪ ( E ) بودند. در حالی که مدل بالاترین احتمال فردی را برای این نمونه در کلاس صحیح قرار می‌دهد، مطمئن نیست که می‌تواند از کلاس C، A یا E نیز باشد.

A (نمونه 01)

A (نمونه 02)

A (نمونه 03)

A (نمونه 04)

B (نمونه 05)

B (نمونه 06)

B (نمونه 07)

B (نمونه 08)

B (نمونه 09)

B (نمونه 10)

B (نمونه 11)

C (نمونه 12)

C (نمونه 13)

C (نمونه 14)

C (نمونه 15)

C (نمونه 16)

C (نمونه 17)

C (نمونه 18)

C (نمونه 19)

C (نمونه 20)

C (نمونه 21)

C (نمونه 22)

C (نمونه 23)

C (نمونه24)

C (نمونه25)

C (نمونه26)

E (نمونه27)

E (نمونه28)

E (نمونه29)

E (نمونه30)

E (نمونه 31)

E (نمونه 32)

E (نمونه 33)

E (نمونه 34)

E (نمونه 35)

E (نمونه 36)

E (نمونه 37)

E (نمونه 38)

F (نمونه 39)

F (نمونه40)

F (نمونه41)

F (نمونه42)

G (نمونه43)

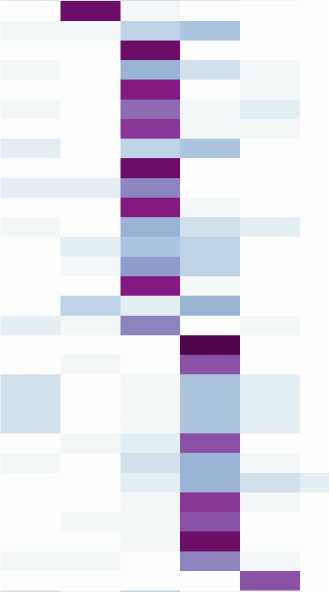
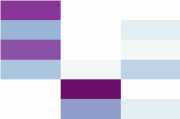
G (نمونه44)

H (نمونه45)

H (نمونه46)

I (نمونه47)

I (نمونه48)

BCEFGH I

A

احتمال کلاس

شکل 11. 4: نقشه حرارتی یک مجموعه آزمایشی با هشت کلاس. کلاس‌های واقعی در برچسب‌های ردیف نشان داده می‌شوند در حالی که ستون‌ها احتمالات هر دسته را کمیت می‌دهند (با برچسب A تا I )

جدول 11. 1: ماتریس سردرگمی برای مسئله دو کلاسه ("رویدادها" و "بدون رویداد. " سلول‌های جدول تعداد مثبت‌های واقعی ( *TP* )، مثبت‌های کاذب ( *FP* )، منفی‌های درست ( *TN* ) و منفی‌های کاذب را نشان می‌دهد. *FN* )

پیش‌بینی مشاهده شد

رویداد بی اتفاق

|  |  |  |
| --- | --- | --- |
| رویداد | *TP* | *FP* |
| بدون رویداد | *FN* | *TN* |

مناطق مبهم

یک رویکرد برای بهبود عملکرد طبقه‌بندی، ایجاد یک منطقه *مبهم* (*Equivocal Zone*) یا *نامعین (indeterminate zone) است* که در آن کلاس به‌طور رسمی پیش‌بینی نشود، زمانی که اطمینان بالا نیست. برای یک مسئله دو کلاسه که تقریباً در پاسخ متعادل است، منطقه مبهم می‌تواند به‌عنوان 0 تعریف شود. 50 *± z.* اگر *z* 0. 10 بود، نمونه‌هایی با احتمال پیش‌بینی بین 0. 40 و 0. 60 «مبهم» نامیده می‌شوند. در این حالت، عملکرد مدل بدون احتساب نمونه‌های موجود در ناحیه نامعین محاسبه می‌شود. **نرخ مبهم نیز باید با عملکرد گزارش شود تا میزان نتایج پیش‌بینی نشده به خوبی درک شود**. برای مجموعه‌های داده با بیش از 2 کلاس ( *C>* 2)، آستانه‌های مشابهی را می‌توان در جایی که بزرگترین احتمال کلاس باید بزرگتر از (1 */C* )+ *z* باشد برای پیش‌بینی قطعی اعمال کرد. برای داده‌های نشان داده شده در شکل.  [11. 4 ،](#bookmark507) اگر (1 */C* )+ *z* روی 30 درصد تنظیم شود، 5 نمونه به‌عنوان مبهم تعیین می‌شود.

11. 2 ارزیابی کلاس‌های پیش‌بینی شده

یک روش رایج برای توصیف عملکرد یک مدل طبقه بندی، ***ماتریس سردرگمی*** *است.* این یک جدول متقابل ساده از کلاس‌های مشاهده شده و پیش‌بینی شده برای داده‌ها است. جدول [11. 1](#bookmark512) مثالی را نشان می‌دهد که نتیجه دارای دو کلاس باشد. سلول‌های مورب مواردی را نشان می‌دهند که در آن کلاس‌ها به درستی پیش‌بینی شده‌اند، در حالی که مورب‌های خارج تعداد خطاها را برای هر مورد ممکن نشان می‌دهند.

ساده‌ترین معیار، میزان دقت کلی (یا برای بدبینان، میزان خطا) است. این نشان دهنده توافق بین طبقات مشاهده شده و پیش‌بینی شده است و صریح‌ترین تفسیر را دارد. با این حال، استفاده از این آمار دارای معایبی است. اولاً، شمارش دقت کلی هیچ تفاوتی در مورد *نوع* خطاهای ایجاد شده ایجاد نمی‌کند. در فیلتر کردن هرزنامه، هزینه حذف اشتباه یک ایمیل مهم احتمالاً بیشتر از عبور نادرست ایمیل‌های اسپم از فیلتر است. در شرایطی که هزینه‌ها متفاوت است،

دقت ممکن است ویژگی‌های مهم مدل را اندازه نگیرد.  [پروست و همکاران](#bookmark1023) [( 1998](#bookmark1023) ) بحث جامعی در مورد این موضوع ارائه می‌دهد که در ادامه بیشتر مورد بررسی قرار می‌گیرد.

دوم اینکه باید فرکانس‌های طبیعی هر کلاس را در نظر گرفت. به‌عنوان مثال، در ایالات متحده، زنان باردار به‌طور معمول برای ­آزمایش آلفا فتوپروتئین خون می‌گیرند که تلاش می‌کند مسائل ژنتیکی مانند سندرم داون را شناسایی کند. میزان این اختلال را فرض کنید[[22]](#footnote-22) در جنین‌ها تقریباً 1 در 800 یا حدود یک دهم درصد است. یک مدل پیش‌بینی می‌تواند با پیش‌بینی منفی بودن همه نمونه‌ها برای سندرم داون، به دقت تقریباً کاملی دست یابد.

برای تعیین اینکه آیا یک مدل به اندازه کافی عمل می‌کند یا خیر، از چه میزان دقت معیاری باید استفاده شود؟ نرخ بدون اطلاعات، میزان دقتی است که بدون مدل قابل دستیابی است. راه‌های مختلفی برای تعریف این نرخ وجود دارد. برای یک مجموعه داده با کلاس‌های *C، ساده‌ترین تعریف، بر اساس تصادفی محض، 1 /C* است. با این حال، این فرکانس‌های نسبی کلاس‌ها در مجموعه آموزشی را در نظر نمی‌گیرد. برای مثال سندرم داون، اگر 1000 نمونه تصادفی از جمعیتی که آزمایش را دریافت می‌کنند جمع‌آوری شود، تعداد نمونه‌های مثبت مورد انتظار کم خواهد بود (شاید 1 یا 2). مدلی که به سادگی همه نمونه‌ها را برای نشانگان داون منفی پیش‌بینی می‌کرد، به راحتی از نرخ بدون اطلاعات بر اساس حدس‌زنی تصادفی (50%) پیشی می‌گیرد. **یک تعریف جایگزین از نرخ بدون اطلاعات، درصد بزرگترین کلاس در مجموعه آموزشی است. مدل‌هایی با دقت بیشتر از این نرخ ممکن است معقول در نظر گرفته شوند.** تأثیر عدم تعادل شدید طبقاتی و برخی راه‌حل‌های ممکن در فصل بحث شده است. 16.

به جای محاسبه دقت کلی و مقایسه آن با ­نرخ بدون اطلاعات، می‌توان از معیارهای دیگری استفاده کرد که توزیع کلاس نمونه‌های مجموعه آموزشی را در نظر می‌گیرد. آمار *کاپا (*همچنین به‌عنوان کاپا کوهن شناخته می‌شود) در ابتدا برای ارزیابی توافق بین دو ارزیاب طراحی شد [( کوهن 1960](#bookmark1013) ). کاپا دقتی را که به‌طور تصادفی ایجاد می‌شود در نظر می‌گیرد. شکل آماری است

*O - E*

*کاپا =* ~~1~~ *~~- E~~*

که در آن *O* دقت مشاهده شده و *E* دقت مورد انتظار بر اساس مجموعات حاشیه‌ای ماتریس سردرگمی است. این آمار می‌تواند ­مقادیری بین *-* 1 و 1 داشته باشد. مقدار 0 به معنای عدم توافق بین کلاس‌های مشاهده شده و پیش‌بینی شده است، در حالی که مقدار 1 نشان دهنده ­تطابق کامل پیش‌بینی مدل و کلاس‌های مشاهده شده است. مقادیر منفی نشان می‌دهد که پیش‌بینی در جهت *مخالف* حقیقت است، اما مقادیر منفی بزرگ به ندرت در هنگام کار با مدل‌های پیش‌بینی رخ می‌دهد. [[23]](#footnote-23) هنگامی که توزیع‌های کلاس معادل هستند، دقت کلی و کاپا متناسب هستند. بسته به زمینه، مقادیر کاپا بین 0. 30 تا 0. 50 نشان دهنده توافق منطقی است. فرض کنید دقت یک مدل بالا باشد (90%) اما دقت مورد انتظار نیز بالا باشد (85%)، آماره کاپا تطابق متوسطی (کاپا = 1/3) را بین کلاس‌های مشاهده شده و پیش‌بینی‌شده نشان می‌دهد.

آمار کاپا همچنین می‌تواند برای ارزیابی تطابق در ­مسائل با بیش از دو کلاس گسترش یابد. هنگامی که یک نظم طبیعی برای کلاس‌ها وجود دارد (به‌عنوان مثال، "کم"، "متوسط" و "بالا")، شکل جایگزینی از آمار به نام *کاپا وزنی می‌*تواند برای اعمال جریمه‌های اساسی تر در مورد ­خطاهای دورتر استفاده شود. از نتیجه واقعی به‌عنوان مثال، یک نمونه «کم» که به اشتباه به‌عنوان «بالا» پیش‌بینی می‌شود، آمار کاپا را بیشتر از خطای «کم» به‌عنوان «متوسط» کاهش می‌دهد. برای جزئیات بیشتر به ( Agresti 2002 ) مراجعه کنید.

*مسائل دو طبقه*

موردی را در نظر بگیرید که دو کلاس وجود دارد. جدول [11. 1](#bookmark512) ماتریس سردرگمی را برای کلاس‌های عمومی "رویداد" و "غیر رویداد" نشان می‌دهد. ردیف بالای جدول مربوط به نمونه‌های پیش‌بینی شده به‌عنوان رویداد است. برخی به درستی پیش‌بینی می‌شوند (مثبت‌های واقعی یا *TP* ) در حالی که برخی دیگر به‌طور نادرست طبقه‌بندی می‌شوند (مثبت کاذب ­یا *FP* ). به‌طور مشابه، ردیف دوم حاوی منفی‌های پیش‌بینی شده با منفی‌های واقعی ( *TN* ) و منفی‌های کاذب ( *FN* ) است.

برای دو کلاس، آمار اضافی وجود دارد که ممکن است زمانی مرتبط باشد که یک کلاس به‌عنوان رویداد مورد علاقه تفسیر شود (مانند سندرم داون در مثال قبلی). ***حساسیت***مدل نرخی است که رویداد مورد علاقه به درستی برای همه نمونه‌های دارای رویداد پیش‌بینی می‌شود، یا

# نمونه با رویداد *و* پیش‌بینی شده که رویداد داشته باشد

*حساسیت = — - -*

# نمونه برگزاری رویداد

حساسیت گاهی اوقات به‌عنوان *نرخ مثبت واقعی در نظر گرفته می‌شود* زیرا دقت را در جمعیت رویداد اندازه‌گیری می‌کند. برعکس، ***ویژگی***به‌عنوان نرخی تعریف می‌شود که نمونه‌های غیر رویدادی به‌عنوان غیرواقعی پیش‌بینی می‌شوند، یا

# نمونه بدون رویداد *و* پیش‌بینی‌شده به‌عنوان غیررویداد

*ویژگی =*

# نمونه بدون رویداد

نرخ *مثبت کاذب* به‌عنوان یک منهای ویژگی تعریف می‌شود. با فرض یک سطح ثابت از دقت برای مدل، معمولاً بین حساسیت و ویژگی یک مبادله وجود دارد. به‌طور شهودی، افزایش حساسیت یک مدل احتمالاً باعث از دست دادن ویژگی می‌شود، زیرا نمونه‌های بیشتری به‌عنوان رویداد پیش‌بینی می‌شوند. معاوضه‌های بالقوه بین حساسیت و ویژگی ممکن است زمانی مناسب باشد که مجازات‌های متفاوتی در ارتباط با هر یک وجود داشته باشد

جدول 11. 2: ماتریس سردرگمی مجموعه تست برای آموزش مدل رگرسیون لجستیک با داده‌های امتیازدهی اعتباری از بخش.  [4. 5](#bookmark222)

پیش‌بینی مشاهده شد

بد خوب

|  |  |  |
| --- | --- | --- |
| بد | 24 | 10 |
| خوب | 36 | 130 |

نوع خطا در فیلتر کردن هرزنامه، معمولاً تمرکز بر ویژگی وجود دارد. اگر ایمیل‌های اعضای خانواده یا همکارانشان حذف نشود، اکثر مردم مایلند که برخی از هرزنامه‌ها را ببینند. منحنی *مشخصه عملکرد گیرنده (ROC)* یکی از تکنیک‌های ارزیابی این مبادله است و در بخش بعدی مورد بحث قرار می‌گیرد.

در فصل [4](#bookmark192) مثال امتیازدهی اعتباری را معرفی کردیم. دو مدل طبقه‌بندی برای پیش‌بینی کیفیت اعتبار مشتری ایجاد شد: SVM و رگرسیون لجستیک. از آنجایی که عملکرد دو مدل تقریباً معادل بود، مدل رگرسیون لجستیک به دلیل سادگی مورد علاقه بود. با استفاده از مجموعه آزمایشی انتخاب شده قبلی از 200 مشتری، جدول [11. 2](#bookmark514) ماتریس سردرگمی مرتبط با مدل رگرسیون لجستیک را نشان می‌دهد. دقت کلی 77٪ بود که کمی بهتر از نرخ بدون اطلاعات 70٪ است. مجموعه تست دارای مقدار کاپا 0. 375 بود که نشان دهنده توافق متوسط است. اگر رویداد مورد علاقه را به‌عنوان مشتری با اعتبار بد انتخاب کنیم، حساسیت این مدل 40 درصد و ویژگی آن 92. 9 درصد برآورد می‌شود. واضح است که این مدل در پیش‌بینی زمانی که مشتریان اعتبار بدی دارند مسأله دارد. این احتمالاً به دلیل عدم تعادل کلاس‌ها و فقدان یک پیش‌بینی قوی برای اعتبار بد است.

اغلب، علاقه به داشتن یک معیار واحد وجود دارد که منعکس کننده ­نرخ‌های مثبت کاذب و منفی کاذب باشد. شاخص *J* Youden [( Youden 1950](#bookmark1028) ) که عبارت است از

*J* = حساسیت *S* + *خاصیت S -* 1

نسبت نمونه‌های پیش‌بینی‌شده درست را برای هر دو گروه رویداد و غیر رویداد اندازه‌گیری می‌کند. در برخی زمینه‌ها، این ممکن است روش مناسبی برای خلاصه کردن بزرگی هر دو نوع خطا باشد. رایج‌ترین روش برای ترکیب حساسیت و ویژگی در یک مقدار واحد از منحنی مشخصه عملکرد گیرنده (ROC) استفاده می‌کند که در زیر مورد بحث قرار گرفته است.

یکی از جنبه‌های حساسیت و ویژگی که اغلب نادیده گرفته می‌شود این است که آنها معیارهای *مشروط هستند.* حساسیت میزان دقت فقط برای جمعیت رویداد (و ویژگی برای غیر رویدادها) است. با استفاده از حساسیت و ویژگی، متخصص زنان و زایمان می‌تواند اظهاراتی مانند "با فرض اینکه جنین سندرم داون نداشته باشد، آزمایش 95٪ دقت دارد. " با این حال، این عبارات ممکن است برای بیمار مفید نباشد، زیرا برای نمونه‌های جدید، تنها چیزی که شناخته شده است، پیش‌بینی است. شخصی که از پیش‌بینی مدل استفاده می‌کند معمولاً به سؤالات *بدون قید و شرط علاقه دارد* مانند "چقدر احتمال دارد جنین به این اختلال ژنتیکی مبتلا شود؟" این به سه مقدار بستگی دارد: حساسیت ­و ویژگی تست تشخیصی و شیوع رویداد در جمعیت. به‌طور شهودی، اگر رویداد نادر است، این باید در پاسخ منعکس شود. با در نظر گرفتن شیوع، شباهت به حساسیت، *مقدار پیش‌بینی شده مثبت است* و شباهت به ویژگی، مقدار پیش‌بینی *شده منفی است.* این مقادیر، داده‌ها را بدون قید و شرط ارزیابی می‌کنند. [[24]](#footnote-24) مقدار پیش‌بینی شده مثبت به این سوال پاسخ می‌دهد که "احتمال اینکه این نمونه یک رویداد باشد چقدر است؟" فرمول‌ها هستند

*حساسیت x شیوع*

*PPV* =

*NPV* =

*( حساسیت x شیوع )* + ((1 *— ویژگی ) x (*1 *— شیوع* ))

*خاصیت x (*1 *- P فراوانی* )

( *فراوانی P x (*1 *- حساسیت* )) + ( *خاصیت x (*1 *- P فراوانی* ))

بدیهی است که مقادیر پیش‌بینی، ترکیبی از عملکرد و نرخ رویدادها هستند. پانل بالایی در شکل [11. 5](#bookmark516) اثر شیوع را بر مقادیر پیش‌بینی نشان می‌دهد زمانی که مدل دارای ویژگی 95٪ و حساسیت 90٪ یا 99٪ باشد. زمانی که شیوع کم باشد، می‌توان مقادیر پیش‌بینی منفی بزرگی را به دست آورد. با این حال، با بالا رفتن نرخ رویداد، مقدار پیش‌بینی منفی بسیار کوچک می‌شود. برعکس برای مقادیر پیش‌بینی مثبت صادق است. این شکل همچنین نشان می‌دهد که تفاوت قابل‌توجهی در حساسیت (90 درصد در مقابل 99 درصد) تأثیر کمی بر مقادیر پیش‌بینی مثبت دارد.

پانل پایینی شکل.  [11. 5](#bookmark516) مقدار اخباری مثبت را به‌عنوان تابعی ­از حساسیت و ویژگی زمانی که نرخ رویداد متعادل است (50٪) نشان می‌دهد. در این حالت، مقدار پیش‌بینی شده مثبت خواهد بود

*حساسیت S*

*PPV* =

*TP*

*TP* + *FP*

*حساسیت x (*1 *- ویژگی )*

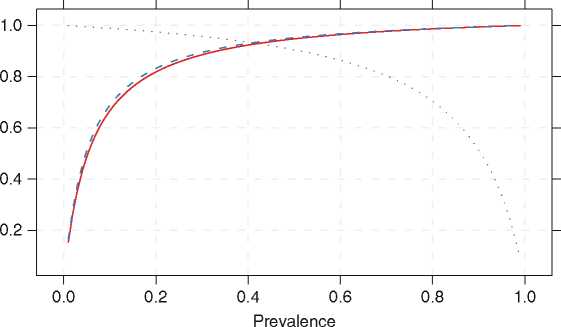
این شکل همچنین نشان می‌دهد که مقدار حساسیت تأثیر کمتری نسبت به ویژگی دارد. به‌عنوان مثال، اگر ویژگی بالا باشد، مثلاً *>* 90 درصد، یک مقدار پیش‌بینی شده مثبت بزرگ را می‌توان در طیف وسیعی از حساسیت‌ها به دست آورد.

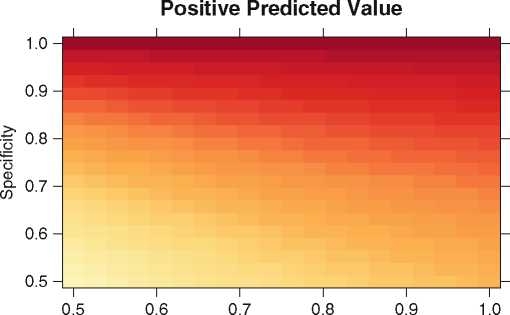
مقادیر پیش‌بینی اغلب برای توصیف مدل استفاده نمی‌شود. دلایل مختلفی وجود دارد که بیشتر آنها به شیوع مربوط می‌شود. اولاً، میزان شیوع مسأله است. تجربه ما این است که تعداد کمی از افراد، حتی متخصصان، مایل به ارائه تخمینی از این مقدار بر اساس دانش قبلی هستند. همچنین، شیوع پویا است. به‌عنوان مثال، نرخ ایمیل‌های هرزنامه ­زمانی که طرح‌های جدید ابداع می‌شوند کاهش می‌یابد اما بعداً به سطح پایه کاهش می‌یابد. برای تشخیص‌های متخصصی، شیوع بیماری‌ها می‌تواند بسیار متفاوت باشد

PPV (حساسیت = 90%)

PPV (حساسیت = 99%)

NPV (ویژگی = 95%)





Sensitivity

**1** 0

0. 9

0. 8

-0. 7

-0. 6

-0. 5

شکل 11. 5: *بالا* : تأثیر شیوع بر مقادیر اخباری مثبت و منفی. PPV با استفاده از ویژگی 95 درصد و دو مقدار حساسیت محاسبه شد. NPV با حساسیت 90 درصد و ویژگی 95 درصد محاسبه شد. *پایین* : برای یک شیوع ثابت 50 درصد، مقادیر اخباری مثبت به‌عنوان تابعی از حساسیت و ویژگی در موقعیت جغرافیایی نشان داده می‌شود (به‌عنوان مثال، شهری در مقابل برون شهری). به‌عنوان مثال، در یک کارآزمایی بالینی چند مرکزی از یک آزمایش تشخیصی *نایسریا گونوره،* شیوع در جمعیت بیمار از 0٪ تا 42. 9٪ در 9 محل بالینی متفاوت بود [( Becton Dickinson and Company 1991](#bookmark1011) ).

جدول 11. 3: ماتریس سردرگمی و هزینه ها/مزایای سود برای ­مثال پست مستقیم [لاروس ( 2006](#bookmark1020) )

|  |  |  |  |
| --- | --- | --- | --- |
| پیش‌بینی کرد | مشاهده شده | | مشاهده شده |
| پاسخ بی پاسخ | | پاسخ بی پاسخ |
| واکنش | *TP* | *FP* | 26. 40 *-* 2. 00 دلار |
| عدم پاسخگویی | *FN* | *TN* | *-* 28. 40 دلار - |

***معیارهای مبتنی بر عدم دقت***

برای بسیاری از کاربردهای تجاری مدل‌های پیش‌بینی، دقت هدف اصلی مدل نیست. اغلب، هدف مدل ممکن است این باشد:

* فرصت‌های سرمایه گذاری را پیش‌بینی کنید که بازده را به حداکثر می‌رساند
* بهبود رضایت مشتری با تقسیم‌بندی بازار
* کاهش هزینه‌های موجودی کالا با بهبود پیش‌بینی‌کننده‌های تقاضای محصول یا
* کاهش هزینه‌های مرتبط با تراکنش‌های تقلبی

در حالی که دقت مهم است، فقط توضیح می‌دهد که مدل چقدر داده‌ها را به خوبی پیش‌بینی می‌کند. اگر مدل *برای هدف بخوبی برازش شده باشد،* سایر معیارهای مستقیم‌تر در عملکرد ­باید در نظر گرفته شوند. این معیارها پیامدهای پیش‌بینی‌کننده‌های صحیح و نادرست (یعنی مزایا و هزینه‌ها) را کمّی می‌کنند. به‌عنوان مثال، در تشخیص تقلب، ممکن است یک مدل برای تعیین کمیت احتمال جعلی بودن یک تراکنش استفاده شود. فرض کنید که تقلب اتفاق مورد علاقه است. هر گونه پیش‌بینی مدل تقلب (صحیح یا غیر صحیح) هزینه‌ای برای بررسی عمیق‌تر پرونده دارد. برای موارد مثبت واقعی، یک مزیت قابل سنجش نیز برای گرفتن تراکنش‌های بد وجود دارد. به همین ترتیب، منفی کاذب منجر به از دست دادن درآمد می‌شود.

برنامه بازاریابی مستقیم را در نظر بگیرید [لاروس](#bookmark1020) [( 2006](#bookmark1020) ، فصل 7) که در آن یک شرکت پوشاک علاقه مند به ارائه تبلیغات از طریق پست به ­مشتریان خود است. با استفاده از داده‌های موجود مشتری در مورد عادات خرید، آنها مایلند پیش‌بینی کنند که چه کسی پاسخ می‌دهد (یعنی دو کلاس و «پاسخ‌دهندگان» و « ­غیرپاسخ‌دهندگان»). جدول 2 *×* 2 نتایج ممکن در جدول نشان داده شده است [11. 3](#bookmark517) که در آن نوع تصمیمات در سمت چپ و درآمد یا هزینه هر تصمیم ­در سمت راست ارائه شده است. به‌عنوان مثال، اگر مدل به‌طور دقیق پاسخ دهنده را پیش‌بینی کند، میانگین سود زمانی که مشتری به تبلیغات پاسخ می‌دهد 28. 40 دلار تخمین زده می‌شود. هزینه کمی 2. 00 دلار برای پست تبلیغاتی ­وجود دارد، بنابراین سود خالص یک تصمیم درست 26. 40 دلار است. اگر به‌طور نادرست پیش‌بینی کنیم که یک مشتری پاسخ می‌دهد (مثبت کاذب)، تنها ضرر هزینه تبلیغات است (2. 00 دلار).

جدول 11. 4: سمت چپ: یک ماتریس سردرگمی آزمون فرضی برای یک مدل پیش‌بینی با حساسیت 75 درصد و ویژگی 94. 4 درصد. راست: ماتریس سردرگمی زمانی که یک پست انبوه برای همه مشتریان استفاده می‌شود

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| پیش‌بینی کرد | مشاهده شده | | مشاهده شده | |
| پاسخ بی پاسخ | | پاسخ بی پاسخ | |
| واکنش | 1،500 | 1000 *\_* \_ | 2000 | 18000 |
| عدم پاسخگویی | 500 | 17,000 *\_* \_ | 0 | 0 |

اگر مدل به‌طور دقیق عدم پاسخگویی را پیش‌بینی کند، هیچ سود یا ضرری وجود ندارد زیرا آنها خریدی انجام نمی‌دادند و نامه‌دهنده ارسال نشد. [[25]](#footnote-25) با این حال، پیش‌بینی نادرست اینکه یک پاسخ‌دهنده واقعی پاسخ نمی‌دهد به این معنی است که 28. 40 دلار بالقوه از دست رفته است، بنابراین این هزینه یک منفی کاذب است. سود کل برای یک مدل خاص پس از آن است

*سود* = 26 *دلار* 40 *TP -* 2 دلار. 00 *FP* - 28 دلار. 40 *FN*  (11. 2)

با این حال، شیوع طبقات باید در نظر گرفته شود. نرخ پاسخ در بازاریابی مستقیم اغلب بسیار پایین است [( لینگ و لی 1998](#bookmark1021) ) بنابراین سود مورد انتظار برای یک برنامه بازاریابی معین ممکن است توسط هزینه‌های منفی کاذب هدایت شود زیرا این مقدار احتمالاً بزرگتر از دو مقدار دیگر در معادله است.  [11. 2 .](#bookmark16)

جدول [11. 4](#bookmark16) ماتریس‌های سردرگمی فرضی را برای 20000 مشتری با نرخ پاسخ دهی 10 درصد نشان می‌دهد. جدول سمت چپ نتیجه یک مدل پیش‌بینی شده با حساسیت 75 درصد و ویژگی 94. 4 درصد است. کل سود به ازای هر مشتری 23400 دلار یا 1. 17 دلار خواهد بود. فرض کنید مدل دیگری حساسیت مشابه اما ویژگی 100 درصدی داشته باشد. در این مورد، سود کل به 25400 دلار افزایش می‌یابد که با توجه به افزایش قابل‌توجه در عملکرد مدل (بیشتر به دلیل هزینه کم پست تبلیغاتی) یک سود نهایی است.

سمت راست جدول [11. 4](#bookmark16) نتایج را زمانی نشان می‌دهد که از یک پست انبوه برای همه مشتریان استفاده می‌شود. این رویکرد دارای حساسیت کامل و بدترین ویژگی ممکن است. در اینجا، به دلیل هزینه‌های کم، سود 16800 دلار یا 0. 84 دلار برای هر مشتری است. این باید عملکرد پایه برای هر مدل پیش‌بینی در نظر گرفته شود. مدل‌ها را می‌توان با استفاده از سود یا *افزایش سود که به‌عنوان سود مدل بالاتر و فراتر از سود حاصل از ارسال انبوه* تخمین زده می‌شود، مشخص کرد.

با دو کلاس، یک طرح کلی برای ترکیب هزینه‌های نابرابر با معیارهای عملکرد ارائه شده است [دراموند و هولته](#bookmark1015) [( 2000](#bookmark1015) ). آنها تابع احتمال-هزینه ( *PCF* ) را به صورت تعریف می‌کنند

*PXC (*+ *1-* )

*PCF* =

*P x C ( -|* +) + (1 *- P* ) *x C (*+ *1-* )

که در آن *P* احتمال (قبلی) رویداد است، *C (* *-|* +) هزینه مربوط به پیش‌بینی نادرست یک رویداد (+) به‌عنوان غیرحوادث و *C (*+ *|-* ) هزینه پیش‌بینی نادرست یک غیرواقعه است. *PCF* نسبت کل هزینه‌های مرتبط با یک نمونه مثبت کاذب است. آنها استفاده از تابع هزینه مورد انتظار نرمال شده ( *NEC* ) را برای توصیف مدل پیشنهاد می‌کنند

*NEC* = *PCF x (*1 *- TP* )+(1 *- PCF* ) *x FP*

برای مجموعه‌ای خاص از هزینه‌ها اساساً، *NEC* شیوع ­رویداد، عملکرد مدل و هزینه‌ها را در نظر می‌گیرد و هزینه کل را بین 0 و 1 مقیاس می‌دهد. توجه داشته باشید که این رویکرد فقط هزینه‌ها را به دو نوع خطا اختصاص می‌دهد و ممکن است مناسب نباشد. برای مسائلی که هزینه یا منافع دیگری وجود دارد (مانند هزینه‌های بازاریابی مستقیم نشان داده شده در جدول).  [11. 3 )](#bookmark517) .

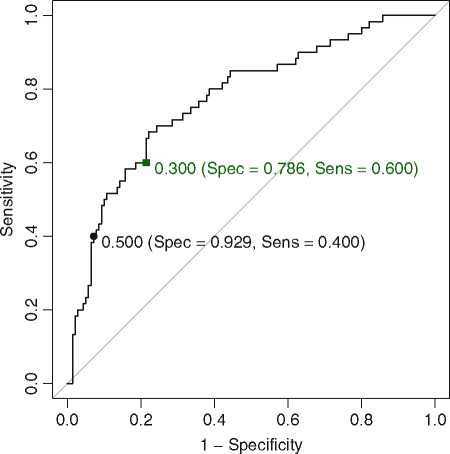
11. 3 ارزیابی احتمالات کلاس

احتمالات کلاس به‌طور بالقوه اطلاعات بیشتری در مورد پیش‌بینی‌کننده‌های مدل نسبت به مقدار کلاس ساده ارائه می‌دهد. در این بخش چندین رویکرد برای استفاده از احتمالات برای مقایسه مدل‌ها مورد بحث قرار گرفت.

**منحنی‌های مشخصه عملیاتی گیرنده (ROC).**

منحنی‌های ROC ( Altman and Bland 1994 ; [براون و دیویس 2006](#bookmark1012) ; [Fawcett 2006](#bookmark1016) ) به‌عنوان یک روش کلی طراحی شد که با توجه به مجموعه‌ای از نقاط داده پیوسته، آستانه موثری را تعیین می‌کند به‌طوری که مقادیر بالاتر از آستانه نشان دهنده یک رویداد خاص باشد. این ابزار در این زمینه در فصل بررسی خواهد شد.  [19 ،](#bookmark864) اما در اینجا، توضیح می‌دهیم که چگونه منحنی ROC می‌تواند برای تعیین برش‌های متناوب برای احتمالات کلاس استفاده شود.

برای مجموعه آزمون مدل اعتباری که قبلاً مورد بحث قرار گرفت، حساسیت برای مدل رگرسیون لجستیک (40٪) ضعیف بود، در حالی که ویژگی نسبتاً بالا بود (92. 9٪). این مقادیر از کلاس‌هایی که با آستانه احتمال پیش فرض 50 درصد تعیین شده بودند، محاسبه شدند. آیا می‌توانیم حساسیت را با کاهش آستانه بهبود دهیم؟[[26]](#footnote-26) برای گرفتن نکات مثبت واقعی بیشتر؟ کاهش آستانه ­قدیمی برای طبقه‌بندی اعتبار بد به 30 درصد منجر به مدلی با حساسیت بهبود یافته می‌شود.



شکل 11. 6: منحنی مشخصه اپراتور گیرنده (ROC) برای مدل رگرسیون لجستیک برای مدل اعتباری به دست می‌آید. نقطه مقدار مربوط به یک برش 50٪ را نشان می‌دهد در حالی که مربع سبز مربوط به برش 30٪ است (یعنی احتمالات بیشتر از 0. 30 رویداد نامیده می‌شوند).

ویژگی (60%) اما کاهش ویژگی (79. 3%). با اشاره به شکل [11. 3 ،](#bookmark506) می بینیم که کاهش آستانه شروع به جذب بیشتر مشتریان با اعتبار بد می‌کند، اما همچنین شروع به تجاوز به عمده مشتریان با اعتبار خوب می‌کند.

منحنی ROC با ارزیابی احتمالات کلاس برای مدل در طول یک زنجیره از آستانه ایجاد می‌شود. برای هر آستانه نامزد، نرخ مثبت واقعی (یعنی حساسیت) و نرخ مثبت کاذب (یک منهای ویژگی) در برابر یکدیگر ترسیم می‌شود. شکل [11. 6](#bookmark522) نتایج این فرآیند را برای داده‌های اعتباری نشان می‌دهد. نقطه سیاه جامد ­آستانه خطای 50٪ است در حالی که مربع سبز با ویژگی‌های عملکرد برای یک آستانه 30٪ مطابقت دارد. در این شکل، اعداد در ­پایان نامه‌های اصلی ( *ویژگی، حساسیت* ) می‌باشد. توجه داشته باشید که مسیر منحنی بین (0، 0) و آستانه 50٪ تند است که نشان می‌دهد حساسیت با سرعت بیشتری نسبت به کاهش ویژگی افزایش می‌یابد. با این حال، زمانی که حساسیت بیشتر از 70٪ باشد، کاهش قابل‌توجهی در ویژگی نسبت به افزایش حساسیت وجود دارد.

این نمودار یک ابزار مفید برای انتخاب آستانه‌ای است که به‌طور مناسبی مبادله بین حساسیت و ویژگی را به حداکثر می‌رساند. با این حال، تغییر آستانه تنها تأثیر مثبتی (یا منفی‌تر) نمونه‌ها را دارد. در ماتریس سردرگمی، نمی‌تواند نمونه‌ها را از *هر دو* سلول جدول خارج از مورب خارج کند. تقریباً همیشه با افزایش 1 حساسیت یا ویژگی کاهش می‌یابد.

منحنی ROC همچنین می‌تواند برای ارزیابی کمی مدل استفاده شود. یک مدل کامل که این دو کلاس را به‌طور کامل از هم جدا کند، 100٪ حساسیت و ویژگی خواهد داشت. از نظر گرافیکی، منحنی ROC یک پله بین (0، 0) و (0، 1) خواهد بود و از (0، 1) تا (1،1) ثابت می‌ماند. مساحت زیر منحنی ROC برای چنین مدلی یک خواهد بود. یک مدل کاملاً ناکارآمد منجر به منحنی ROC می‌شود که از خط مورب 45 *درجه* پیروی می‌کند و مساحت زیر منحنی ROC تقریباً 0. 50 خواهد بود. **برای مقایسه بصری مدل‌های مختلف، منحنی‌های ROC آنها را می‌توان روی یک نمودار قرار داد.** مقایسه منحنی‌های ROC می‌تواند در تضاد دو یا چند مدل با مجموعه‌های پیش‌بینی مختلف (برای یک مدل)، پارامترهای تنظیم متفاوت (یعنی در مقایسه‌های مدل)، یا طبقه‌بندی‌کننده‌های مختلف کامل (یعنی بین مدل‌ها) مفید باشد.

مدل بهینه باید به سمت گوشه سمت چپ بالای طرح منتقل شود. روش دیگر، مدلی با بیشترین مساحت زیر منحنی ROC موثرترین خواهد بود. برای داده‌های اعتباری، مدل لجستیک دارای یک منطقه تخمینی زیر منحنی ROC 0. 78 با فاصله اطمینان 95٪ (0. 7، 0. 85) بود که با استفاده از روش فاصله اطمینان راه انداز تعیین شده بود [( Hall et al. 2004](#bookmark1017) ). تعداد قابل‌توجهی از تحقیقات در مورد روش‌های مقایسه رسمی منحنی‌های ROC متعدد وجود دارد. دیدن [هانلی و مک نیل ( 1982](#bookmark1017) )، [دلانگ و همکاران. ( 1988](#bookmark1014) ) [ونکاترامان](#bookmark1027) [( 2000](#bookmark1027) ) و [پپه و همکاران ( 2009](#bookmark1023) ) برای اطلاعات بیشتر.

یکی از مزیت‌های استفاده از منحنی‌های ROC برای مشخص کردن مدل‌ها این است که، از آنجایی که تابعی از حساسیت و ویژگی است، منحنی به اختلاف در نسبت‌های طبقاتی حساس نیست [( Proost et al. 1998](#bookmark1023) ; [فاوست 2006](#bookmark1016) ). نقطه ضعف استفاده از ناحیه زیر منحنی برای ارزیابی مدل‌ها این است که اطلاعات را مبهم می‌کند. برای مثال، هنگام مقایسه مدل‌ها، معمول است که هیچ منحنی ROC منفردی به‌طور یکنواخت بهتر از منحنی دیگر نیست (یعنی منحنی‌ها متقاطع هستند). با خلاصه کردن این منحنی‌ها، اطلاعات از دست می‌رود، به خصوص اگر یک ناحیه خاص از منحنی مورد توجه باشد. به‌عنوان مثال، یک مدل ممکن است شیب منحنی ROC تند در سمت چپ ایجاد کند اما AUC کمتری نسبت به مدل دیگر داشته باشد. اگر انتهای پایین منحنی ROC مورد توجه اولیه بود، آنگاه AUC بهترین مدل را شناسایی نمی‌کرد. ناحیه جزئی زیر منحنی ROC [( McClish 1989](#bookmark1022) ) جایگزینی است که بر بخش‌های خاصی از منحنی تمرکز دارد.

منحنی ROC فقط برای مسائل دو کلاسه تعریف شده است اما قبلاً ­برای رسیدگی به سه کلاس یا بیشتر تمایل داشته است.  [Hand and Till ( 2001](#bookmark1017) [)،](#bookmark1020) Lachiche and Flach [( 2003](#bookmark1020) ) و [لی و فاین ( 2008](#bookmark1021) ) از رویکردهای مختلفی استفاده می‌کنند که تعریف منحنی ROC را با بیش از دو کلاس گسترش می‌دهد.

نمودارهای بالابر

نمودارهای بالابر [( Ling and Li 1998](#bookmark1021) ) یک ابزار تجسم برای ارزیابی توانایی یک مدل برای تشخیص رویدادها در یک مجموعه داده با دو کلاس است. فرض کنید گروهی از نمونه‌ها با رویدادهای *M* با استفاده از احتمال کلاس رویداد *امتیازدهی می‌شوند.* هنگامی که بر اساس احتمال کلاس ترتیب داده می‌شود، می‌توان امیدوار بود که رویدادها بالاتر از غیر رویدادها قرار گیرند. نمودارهای لیفت دقیقاً این کار را انجام می‌دهند: نمونه‌ها را بر اساس امتیاز آنها رتبه‌بندی کنید و با ارزیابی نمونه‌های بیشتر، نرخ رویداد تجمعی را تعیین کنید. در حالت بهینه، نمونه‌های *M* دارای بالاترین رتبه شامل همه رویدادهای *M* هستند. هنگامی که مدل غیر اطلاعاتی است، بالاترین رتبه *X* ٪ از داده‌ها به‌طور متوسط شامل *X* رویداد است. *بالابر* تعداد نمونه‌هایی است که توسط یک مدل بالاتر از یک انتخاب کاملا تصادفی از نمونه‌ها شناسایی می‌شود.

برای ساخت *نمودار بالابر* مراحل زیر را انجام می‌دهیم:

مجموعه‌ای از نمونه‌ها را پیش‌بینی کنید که در فرآیند ساخت مدل استفاده نشده‌اند اما نتایج شناخته‌شده‌ای دارند.

*پایه،* یعنی درصد رویدادهای واقعی در کل مجموعه داده را تعیین کنید.

داده‌ها را بر اساس احتمال طبقه‌بندی رویداد مورد علاقه مرتب کنید.

برای هر مقدار احتمال کلاس منحصر به فرد، درصد رویدادهای واقعی را در همه نمونه‌ها زیر مقدار احتمال محاسبه کنید.

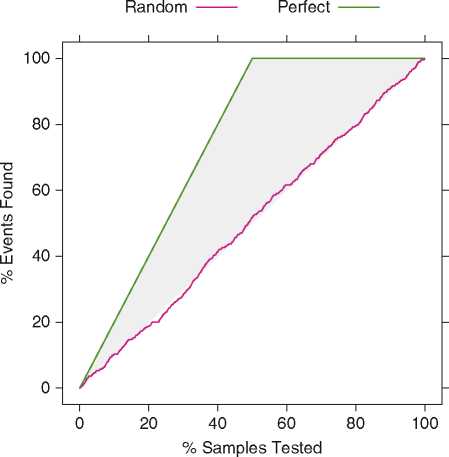
درصد رویدادهای واقعی را برای هر آستانه احتمال بر نرخ رویداد پایه تقسیم کنید.

نمودار بالابر افزایش/افزایش تجمعی را در برابر درصد تجمعی نمونه‌هایی که غربال شده اند ترسیم می‌کند. شکل [11. 7](#bookmark529) بهترین و بدتر منحنی‌های لیفت مورد را برای مجموعه داده‌هایی با نرخ رویداد 50% نشان می‌دهد. مدل غیر اطلاعاتی دارای منحنی نزدیک به خط مرجع 45 *◦* است، به این معنی که مدل هیچ فایده‌ای برای رتبه‌بندی نمونه‌ها ندارد. منحنی دیگر نشان دهنده مدلی است که می‌تواند دو کلاس را کاملاً از هم جدا کند. در نقطه 50 درصد در محور *x،* تمام رویدادها توسط مدل ثبت شده است.

مانند منحنی‌های ROC، منحنی‌های لیفت برای مدل‌های مختلف را می‌توان برای یافتن مناسب‌ترین مدل مقایسه کرد و از ناحیه زیر منحنی به‌عنوان معیار کمی عملکرد استفاده کرد. همچنین مانند منحنی‌های ROC، برخی از قسمت‌های منحنی لیفت بیشتر از سایرین مورد توجه هستند. به‌عنوان مثال، بخش منحنی مرتبط با نمونه‌های دارای بالاترین رتبه باید دارای نرخ مثبت واقعی غنی شده باشد و احتمالاً مهمترین بخش منحنی است.

اپلیکیشن بازاریابی مستقیم را در نظر بگیرید. با استفاده از این منحنی می‌توان یک شبه آستانه برای یک مدل تعیین کرد. مجدداً، فرض کنید نرخ پاسخ دهی وجود دارد و اکثر پاسخ دهندگان در 7 درصد برتر پیش‌بینی‌کننده‌های مدل قرار دارند. ارسال تبلیغات به این زیرمجموعه از مشتریان به‌طور موثر ­آستانه جدیدی را برای پاسخ مشتری تحمیل می‌کند زیرا نمونه‌های زیر آستانه عمل نمی‌کنند.

در این برنامه، به یاد بیاورید که یک مدل پیش‌بینی باید سود بیشتری نسبت به سود پایه مربوط به ارسال تبلیغات ایجاد کند.



شکل 11. 7: نمونه طرح بالابر با دو مدل: یکی که دو کلاس را کاملاً از هم جدا می‌کند و دیگری کاملاً غیر آموزنده است.

به همه مشتریان با استفاده از نمودار بالابر، سود مورد انتظار را می‌توان برای هر نقطه از منحنی محاسبه کرد تا مشخص شود که آیا بالابر برای شکست دادن سود پایه کافی است یا خیر.

11. 4 محاسبات

بسته‌های R AppliedPredictiveModeling، caret، klaR، MASS، pROC و randomForest در این بخش مورد استفاده قرار خواهند گرفت.

برای مثال، مجموعه داده‌های شبیه‌سازی شده نشان داده شده در شکل.  [11. 1](#bookmark502) در این بخش استفاده خواهد شد. برای ایجاد این داده‌ها، تابع quadBoundaryFunc در بسته AppliedPredictiveModeling برای تولید پیش‌بینی‌کننده‌ها و خروجی‌ها استفاده می‌شود ­:

*کتابخانه (AppliedPredictiveModeling)*

*set. seed(975)*

*شبیه‌سازی‌شده آموزش <- quadBoundaryFunc(500)*

*simulatedTest <- quadBoundaryFunc(1000)*

*سر (آموزش شبیه‌سازی شده)*

کلاس پروب X1 X2

1 2. 4685709 2. 28742015 0. 9647251 Class1

2 -0. 1889407 -1. 63949455 0. 9913938 Class1

3 -1. 9101460 -2. 89194964 1. 0000000 Class1

4 0. 3481279 0. 06707434 0. 1529697 Class1

5 0. 1401153 0. 86900555 0. 5563062 Class1

6 0. 7717148 -0. 91504835 0. 2713248 Class2

مدل‌های جنگل تصادفی و تفکیک درجه دوم با داده‌ها برازش خواهند داشت:

*کتابخانه (جنگل تصادفی)*

*rfModel <- randomForest(کلاس ~ X1 + X2،*

*+ داده = آموزش شبیه‌سازی شده،*

*+ ntree = 2000)*

*library(MASS) ## برای تابع qda().*

*qdaModel <- qda (کلاس ~ X1 + X2، داده = شبیه‌سازی شده)*

خروجی تابع پیش‌بینی برای اشیاء qda شامل کلاس‌های پیش‌بینی شده (در اسلات یا شکافی به نام کلاس ) و احتمالات مرتبط در ماتریسی به نام پسین است. برای مدل QDA، پیش‌بینی‌کننده‌هایی برای مجموعه‌های آموزشی و آزمایشی ایجاد خواهد شد. در ادامه این بخش، احتمالات مجموعه آموزشی در یک مدل اضافی برای کالیبره کردن احتمالات کلاس استفاده خواهد شد. سپس کالیبراسیون بر روی احتمالات مجموعه آزمایش اعمال خواهد شد:

*qdaTrainPred <- predict(qdaModel, simulatedTrain)*

*نام‌ها (qdaTrainPred)*

[1] "کلاس" "پسین"

*> head(qdaTrainPred$class)*

[1] Class1 Class1 Class1 Class2 Class1 Class2

سطوح: کلاس 1 کلاس 2

*> head(qdaTrainPred$posterior)* Class1 Class2

0. 7313136 0. 268686374

0. 8083861 0. 191613899

0. 9985019 0. 001498068

0. 3549247 0. 645075330

0. 5264952 0. 473504846

0. 3604055 0. 639594534

*qdaTestPred <- پیش‌بینی (qdaModel، simulatedTest)*

*simulatedTrain$QDAprob <- qdaTrainPred$posterior["Class1"]*

*simulatedTest$QDAprob <- qdaTestPred$posterior["Class1"]*

مدل جنگل تصادفی به دو فراخوانی تابع پیش‌بینی برای بدست آوردن کلاس‌های پیش‌بینی شده و احتمالات کلاس نیاز دارد:

*rfTestPred <- predict(rfModel، simulatedTest، type = "prob") > head(rfTestPred)*

کلاس 1 کلاس 2

1 0. 4300 0. 5700

2 0. 5185 0. 4815

3 0. 9970 0. 0030

4 0. 9395 0. 0605

5 0. 0205 0. 9795

6 0. 2840 0. 7160

*simulatedTest$RFprob <- rfTestPred["Class1"]*

*simulatedTest$RFclass <- predict(rfModel, simulatedTest)*

حساسیت و ویژگی

ci.roc عملکردهایی برای محاسبه حساسیت و ویژگی دارد. این توابع از کاربر می‌خواهند که نقش هر یک از کلاس‌ها را مشخص کند:

*# کلاس 1 به‌عنوان رویداد مورد علاقه استفاده خواهد شد*

*حساسیت (داده = شبیه‌سازی‌شدهTest$RFclass،*

*+ مرجع = شبیه‌سازی‌شدهTest$class،*

*+ مثبت = "Class1")*

0. 8278867

*> ویژگی (داده = شبیه‌سازی‌شدهTest$RFclass،*

*+ مرجع = شبیه‌سازی‌شدهTest$class،*

*+ منفی = "Class2")*

0. 8946396

مقادیر پیش‌بینی را می‌توان با استفاده از شیوع یافت شده در مجموعه داده‌ها (46%) یا با استفاده از قضاوت قبلی محاسبه کرد:

*> posPredValue(داده = شبیه‌سازی‌شدهTest$RFclass،*

*+ مرجع = شبیه‌سازی‌شدهTest$class،*

*+ مثبت = "Class1")*

0. 8695652

*> negPredValue(داده = شبیه‌سازی‌شدهTest$RFclass،*

*+ مرجع = شبیه‌سازی‌شدهTest$class،*

*+ مثبت = "Class2")*

0. 8596803

*# شیوع را به صورت دستی تغییر دهید*

*posPredValue(داده = شبیه‌سازی‌شدهTest$RFclass،*

*+ مرجع = شبیه‌سازی‌شدهTest$class،*

*+ مثبت = "Class1"،*

*+ شیوع = 0. 9)*

0. 9860567

ماتریس سردرگمی

چندین تابع در R برای ایجاد ماتریس سردرگمی وجود دارد. تابع confusionMatrix در بسته caret جدول و ­آمار مرتبط را تولید می‌کند:

*> confusionMatrix(داده = شبیه‌سازی‌شدهTest$RFclass،*

*+ مرجع = شبیه‌سازی‌شدهTest$class،*

*+ مثبت = "Class1")*

ماتریس سردرگمی و آمار

ارجاع

پیش‌بینی کلاس 1 کلاس 2

Class1

Class2

380

79

57

484

دقت: 0. 864

95% فاصله اطمینان (CI): (0. 8412, 0. 8846)

بدون اطلاعات نرخ : 0. 541

P-Value [Acc > NIR]: < 2e-16

کاپا: 0. 7252

P-Value آزمون مک نمار: 0. 07174

حساسیت: 0. 8279

ویژگی: 0. 8946

Pos Pred Value: 0. 8696

ارزش Neg Pred: 0. 8597

شیوع: 0. 4590

میزان تشخیص: 0. 3800

شیوع تشخیص: 0. 4370

کلاس "مثبت": کلاس 1

همچنین در این عملکرد گزینه‌ای برای تنظیم دستی شیوع وجود دارد. اگر بیش از دو کلاس وجود داشته باشد، حساسیت، ویژگی و آمارهای مشابه بر اساس «یک در مقابل همه» محاسبه می‌شود (به‌عنوان مثال، کلاس اول در مقابل مجموعه‌ای از کلاس‌های دو و سه).

منحنی‌های مشخصه عملکرد گیرنده

بسته pROC [( رابین و همکاران 2011](#bookmark1024) ) می‌تواند منحنی ایجاد کند و آمارهای مختلفی را استخراج کند [[27]](#footnote-27). ابتدا، یک شی R باید ایجاد شود که حاوی اطلاعات مربوطه با استفاده از تابع pROC roc باشد. سپس شیء به دست آمده برای تولید منحنی ROC یا محاسبه مساحت زیر منحنی استفاده می‌شود. مثلا،

*> کتابخانه (pROC)*

*> rocCurve <- roc(response = simulatedTest$class,*

*+ پیش‌بینی = شبیه‌سازی شدهTest$RFprob،*

*+ ## این تابع فرض می‌کند که دوم*

*+ ## کلاس رویداد مورد علاقه است، بنابراین ما*

*+ ## برچسب‌ها را معکوس کنید.*

*+ سطوح = دور (سطوح (Test$ کلاس شبیه‌سازی شده)))*

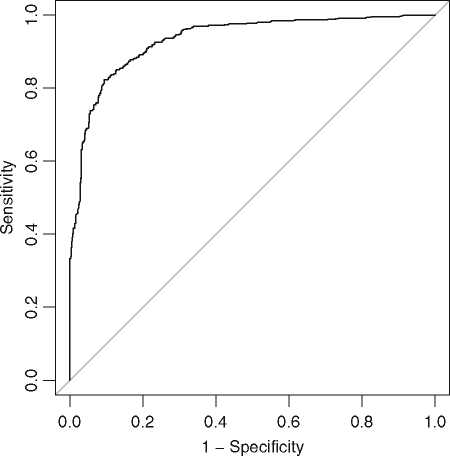
از این شی، ما می‌توانیم آماری (مانند مساحت زیر منحنی ROC و فاصله اطمینان آن) تولید کنیم:

*auc(rocCurve)*

مساحت زیر منحنی: 0. 9328

*ci. roc(rocCurve)*

95% فاصله اطمینان (CI): 0. 9176-0. 948 (DeLong)



شکل 11. 8: نمونه‌ای از منحنی ROC که با استفاده از توابع roc و plot. roc در بسته pROC تولید شده است.

همچنین می‌توانیم از تابع نمودار برای تولید خود منحنی ROC استفاده کنیم:

*plot(rocCurve، legacy. axes = TRUE)*

*## به‌طور پیش‌فرض، محور x به عقب برمی‌گردد، استفاده می‌شود*

*## گزینه legacy. axes = TRUE برای دریافت 1-spec*

*## در محور x در حال حرکت از 0 به 1*

*>*

*## همچنین، منحنی دیگری را می‌توان با استفاده از اضافه کرد*

*دفعه بعد که plot. auc استفاده شد ## افزودن = درست است.*

نتایج این فراخوانی تابع را نشان می‌دهد.

نمودارهای بالابر

منحنی لیفت را می‌توان با استفاده از تابع لیفت در بسته کارت ایجاد کرد. یک فرمول را به‌عنوان ورودی می‌گیرد که در آن کلاس واقعی در سمت چپ فرمول است و یک یا چند ستون برای احتمالات کلاس مدل در سمت راست قرار دارند. به‌عنوان مثال، برای تولید یک نمودار بالابر برای جنگل تصادفی و احتمالات مجموعه آزمون QDA،

*آزمایشگاه‌ها <- c(RFprob = "جنگل تصادفی"،*

*+ QDAprob = "تحلیل متمایز درجه دوم")*

*liftCurve <- lift(class ~ RFprob + QDAprob, data = simulatedTest,*

*+ برچسب‌ها = آزمایشگاه ها)*

*liftCurve*

صدا زدن:

lift. formula (x = کلاس ~ RFprob + QDAprob، داده = شبیه‌سازی شده، برچسب‌ها = آزمایشگاه‌ها)

مدل‌ها: جنگل تصادفی، تحلیل متمایز درجه دوم

رویداد: کلاس 1 (45. 9%)

برای ترسیم دو منحنی بالابر، از تابع xyplot برای ایجاد یک نمودار شبکه استفاده می‌شود:

*## گزینه‌های شبکه را برای تولید افسانه در بالا اضافه کنید*

*xyplot(liftCurve,*

*+ auto. key = لیست (ستون‌ها = 2،*

*+ خطوط = درست،*

*+ امتیاز = FALSE))*

شکل [11. 9 .](#bookmark542)

کالیبره کردن احتمالات

نمودارهای کالیبراسیون همانطور که در بالا توضیح داده شد در تابع calibration. plot در بسته PresenceAbsence و در کالیبراسیون تابع caret موجود است (جزئیات زیر). نحو تابع کالیبراسیون مشابه تابع lift است:

*calCurve <- کالیبراسیون (کلاس ~ RFprob + QDAprob، داده = شبیه‌سازی تست) > calCurve*

صدا زدن:

calibration. formula (x = کلاس ~ RFprob + QDAprob، داده = simulatedTest)

مدل ها: RFprob، QDAprob

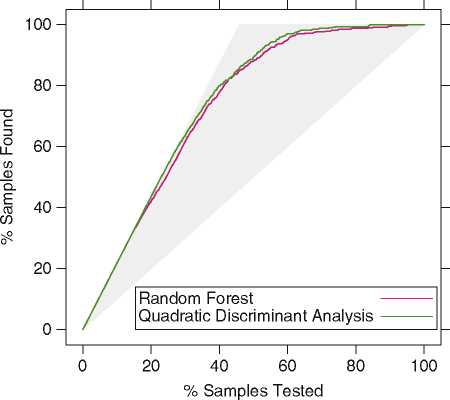
رویداد: کلاس 1

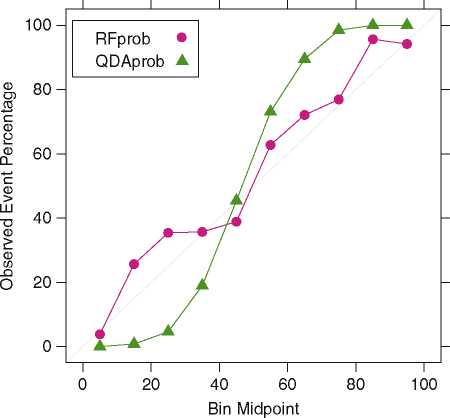
برش: 11

*xyplot (calCurve، auto. key = list (ستون‌ها = 2))*

همچنین این طرح را نشان می‌دهد. یک رویکرد کاملاً متفاوت برای نمودارهای کالیبراسیون که نرخ رویداد مشاهده‌شده را به‌عنوان تابعی از احتمالات کلاس مدل می‌کند را می‌توان در تابع calibrate. plot بسته gbm یافت.

برای تنظیم مجدد احتمالات QDA، یک مدل پس پردازش ایجاد می‌شود که نتیجه واقعی را به‌عنوان تابعی از احتمال کلاس مدل می‌کند. برای برازش یک تابع سیگموئیدی، از مدل رگرسیون لجستیک استفاده می‌شود (به بخش مراجعه کنید.  [12. 2](#bookmark557) برای جزئیات بیشتر) از طریق تابع glm در پایه R. این تابع یک رابط برای مجموعه گسترده‌ای از روش‌ها به نام مدل‌های خطی تعمیم یافته است [( دابسون 2002](#bookmark1015) ) که شامل رگرسیون لجستیک است. برای برازش با مدل، تابع نیاز به





شکل 11. 9: نمونه‌هایی از منحنی‌های لیفت و کالیبراسیون برای مدل‌های جنگل تصادفی و QDA

خانواده برای مشخص کردن نوع داده‌های نتیجه مدل‌سازی شده. از آنجایی که نتیجه ما یک دسته گسسته است، توزیع دوجمله‌ای انتخاب می‌شود:

*## تابع glm() احتمال عامل دوم را مدل می‌کند*

*سطح ##، بنابراین تابع relevel() برای معکوس کردن موقت استفاده می‌شود*

*سطوح فاکتور ##*

*sigmoidalCal <- glm(relevel(class, ref = "Class2") ~ QDAprob,*

*+ داده = آموزش شبیه‌سازی شده،*

*+ خانواده = دوجمله ای)*

*coef(خلاصه(sigmoidalCal))*

برآورد Std. خطای z مقدار Pr(>|z|)

(عرض از مبدأ) -5. 701055 0. 5005652 -11. 38924 4. 731132e-30

QDAprob 11. 717292 1. 0705197 10. 94542 6. 989017e-28

احتمالات تصحیح شده با گرفتن مدل اصلی و اعمال معادله ایجاد می‌شوند.  [11. 1](#bookmark502) با شیب و عرض از مبدأ تخمینی. در R می‌توان از تابع پیش‌بینی استفاده کرد:

*sigmoidProbs <- predict(sigmoidalCal,*

*+ newdata = simulatedTest["QDAprob"، drop = FALSE]،*

*+ نوع = "پاسخ")*

*simulatedTest$QDAsigmoid <- sigmoidProbs*

رویکرد بیزی برای کالیبراسیون، درمان ­توانایی‌های احتمالی کلاس مجموعه آموزشی برای تخمین احتمالات *Pr* [ *X* ] و *Pr* [ *X|K* = *C* ] است (به معادله مراجعه کنید.  [13. 5](#bookmark649) در صفحه [354 )](#bookmark649) . در R می‌توان از تابع مدل ساده بیز NaiveBayes در بسته klaR برای محاسبات استفاده کرد:

*BayesCal <- NaiveBayes(کلاس ~ QDAprob، داده = شبیه‌سازی شده،*

*+ userkernel = TRUE)*

*## مانند qda()، تابع پیش‌بینی برای این مدل ایجاد می‌کند*

*## هم کلاس‌ها و هم احتمالات*

*BayesProbs <- پیش‌بینی (BayesCal,*

*+ newdata = simulatedTest[، "QDAprob"، drop = FALSE])*

*simulatedTest$QDABayes <- BayesProbs$posterior[, "Class1"]*

*## مقادیر احتمال قبل و بعد از کالیبراسیون*

*head(تست شبیه‌سازی شده[, c(5:6، 8، 9)])*

QDAprob RFprob QDAsigmoid QDABayes

0. 3830767 0. 4300 0. 22927068 0. 2515696

0. 5440393 0. 5185 0. 66231139 0. 6383383

0. 9846107 0. 9970 0. 99708776 0. 9995061

0. 5463540 0. 9395 0. 66835048 0. 6430232

0. 2426705 0. 0205 0. 05428903 0. 0566883

0. 4823296 0. 2840 0. 48763794 0. 5109129

گزینه usekernel = TRUE به یک تابع انعطاف‌پذیر اجازه می‌دهد تا ­توزیع احتمال احتمالات کلاس را مدل کند.

این احتمالات جدید با استفاده از نمودار دیگری ارزیابی می‌شوند:

*calCurve2 <- کالیبراسیون(کلاس ~ QDAprob + QDABayes + QDAsigmoid،*

*+ داده = شبیه‌سازی شده)*

*xyplot(calCurve2)*

فصل 12

تحلیل متمایز و خطی دیگر مدل‌های طبقه بندی

به‌طور کلی، تکنیک‌های تمایز یا طبقه‌بندی به دنبال دسته‌بندی نمونه‌ها به گروه‌ها بر اساس ویژگی‌های پیش‌بینی‌کننده هستند و مسیر دستیابی به این کمینه‌سازی برای هر تکنیک متفاوت است. برخی از تکنیک‌ها یک مسیر ریاضی را انتخاب می‌کنند [به‌عنوان مثال، تحلیل تفکیک خطی (LDA)] و برخی دیگر یک مسیر الگوریتمی (به‌عنوان مثال، *k-* نزدیک‌ترین همسایگان).

روش‌های کلاسیک مانند LDA و پسرعموهای ریاضی نزدیک آن (تحلیل تفکیک حداقل مربعات جزئی (PLSDA)، رگرسیون لجستیک ­و غیره) در این فصل مورد بحث قرار خواهند گرفت و بر جداسازی ­نمونه‌ها به گروه‌ها بر اساس ویژگی‌های تغییرات پیش‌بینی تمرکز خواهند کرد.

12. 1 مطالعه موردی: پیش‌بینی برنامه‌های موفق گرنت

این داده‌ها از یک مسابقه Kaggle در سال 2011 است که توسط دانشگاه ملبورن حمایت می‌شد، جایی که علاقه‌ای به پیش‌بینی پذیرش یا عدم پذیرش درخواست گرنت (گرنت اپلیکیشن) وجود داشت. از آنجایی که بودجه عمومی گرنت در طول زمان کاهش یافته بود، تریاژ (اولویت بندی یا ضرورت) درخواست‌های گرنت بر اساس احتمال موفقیت آنها می‌تواند برای تخمین میزان بودجه بالقوه برای دانشگاه مهم باشد. علاوه بر پیش‌بینی موفقیت گرنت، دانشگاه به دنبال درک عواملی بود که در پیش‌بینی موفقیت مهم بودند. همانطور که در ­سرتاسر فصل‌های رگرسیون بحث کردیم، اغلب بین مدل‌هایی که برای درک ایجاد می‌شوند و مدل‌هایی که برای پیش‌بینی توسعه می‌یابند، یک مبادله وجود دارد. همین امر در مورد مدل‌های طبقه‌بندی نیز صادق است. این امر ­در این فصل و فصل‌های بعدی به‌طور مصور توضیح داده خواهد شد.

در این مسابقه، داده‌های مربوط به 8708 گرنت بین سال‌های 2005 و2008 برای ساخت مدل در دسترس بود و مجموعه آزمایشی شامل برنامه‌های کاربردی از سال2009 تا2010 بود. ورودی برنده به منطقه زیر منحنی ROC 0. 968 دست یافت.

© در مجموعه آزمایشی. برندگان رتبه اول و دوم در مورد رویکردهای خود در مورد داده‌ها و مدل‌سازی در وبلاگ Kaggle بحث می‌کنند. [[28]](#footnote-28)

داده‌ها را می‌توان در وب سایتKaggle یافت،[[29]](#footnote-29) اما فقط داده‌های مجموعه آموزشی حاوی نتایج گرنت است. بسیاری از اطلاعات از طریق گرنت جمع‌آوری شد، از جمله اینکه آیا درخواست گرنت موفق بوده است یا خیر. داده‌های اصلی حاوی بسیاری از پیش‌بینی‌کننده‌ها مانند:

* نقش هر فردی که در گرنت ذکر شده است. مقادیر احتمالی عبارتند از: محقق ارشد (در داده‌ها به "CI" کوتاه شده است)، محقق تفویض شده (DR)، سرپرست اصلی (PS)، مشاور خارجی (EA)، رئیس محقق خارجی (ECI)، محقق ارشد دانشجو (SCI)، دانش آموز محقق (SR)، بازدیدکننده افتخاری (HV)، یا ناشناخته (UNK). تعداد کل افراد ­ذکر شده در این گرنت بین 1 تا 14 نفر بود.
* چندین ویژگی هر فرد در این گرنت، مانند تاریخ تولد، زبان اصلی، بالاترین مدرک تحصیلی، ملیت، تعداد ­گرنت موفق (و ناموفق) قبلی، گروه آموزشی، وضعیت هیات علمی، رتبه علمی، طول مدت اشتغال در دانشگاه و تعداد انتشارات در چهار درجه مختلف مجلات.
* یک یا چند کد مربوط به طبقه‌بندی رشته‌ها، دوره‌ها و رشته‌های تحقیقاتی استرالیا (RFCD). با استفاده از این، گرنت را می‌توان به زیر گروه‌هایی مانند اقتصاد کاربردی، میکروبیولوژی و کتابداری طبقه‌بندی کرد. 738 مقدار ممکن از کدهای RFCD در داده‌ها وجود داشت. اگر بیش از یک کد برای گرنت مشخص شده بود، درصد نسبی آنها ثبت می‌شد. کدهای RFCD لیست شده توسط اداره آمار استرالیا[[30]](#footnote-30) محدوده از 210000 تا 449999. گرنت زیادی با کدهای غیرمعنا (مانند 0 یا 999999) وجود داشت که برای این تحلیل‌ها در یک دسته ناشناخته گروه‌بندی شدند.
* یک یا چند کد مربوط به طبقه‌بندی هدف اجتماعی-اقتصادی (SEO). این طبقه بندی، هدف مورد نظر گرنت، مانند توسعه فعالیت‌های ساختمانی یا خدمات بهداشتی را توصیف می‌کند. اگر بیش از یک کد برای گرنت مشخص شده بود، درصد نسبی آنها ثبت می‌شد. مانند کدهای RFCD، مقادیری در داده‌ها وجود داشت که به هیچ یک از کدهای فهرست‌شده توسط دولت استرالیا نگاشت نبودند و در دسته‌ای ناشناخته دسته‌بندی شدند.
* تاریخ ارائه گرنت
* ارزش پولی گرنت به 17 گروه تقسیم شده است
* یک کد دسته گرنت که نوع حامی و همچنین کدی را برای حامی خاص توضیح می‌دهد

یکی از اولین مراحل در فرآیند ساخت مدل تبدیل یا کدگذاری ساختار داده اصلی به شکلی است که برای مدل آموزنده‌ترین باشد (یعنی *مهندسی ویژگی* ). این فرآیند رمزگذاری بسیار مهم است و باید با آینده نگری در تحلیل‌هایی که انجام خواهد شد انجام شود تا بتوان پیش‌بینی‌کننده‌های مناسب را از داده‌های اصلی روشن کرد. عدم موفقیت در ایجاد قالب‌بندی مناسب پیش‌بینی‌کننده‌ها می‌تواند از توسعه مدل‌های پیش‌بینی موثر جلوگیری کند.

شکل اصلی داده‌های گرنت برای مدل‌سازی مناسب نیست. به‌عنوان مثال، بسیاری از زمینه‌ها برای هر فردی که در گرنت مشارکت دارد، تفکیک شده است. به این ترتیب، 15 ستون در داده‌ها برای هر فرد وجود دارد. از آنجایی که ممکن است 14 نفر با گرنت مرتبط باشند، تعداد زیادی ستون برای گرنت وجود دارد که بسیاری از آنها هیچ داده‌ای ندارند.

نحوه رمزگذاری این داده‌ها اولین سوال اولیه است. به‌عنوان مثال، از آنجایی که اغلب افراد متعددی با گرنت مرتبط هستند، این اطلاعات چگونه باید در داده‌ها نمایش داده شود؟ به‌طور مشابه، زمانی که ­کدهایRFCD[[31]](#footnote-31) چندگانه و درصدهای مرتبط وجود دارد، این داده‌ها به چه صورت باید وارد مدل‌ها شوند؟ علاوه بر این، این داده‌ها حاوی مقادیر زیادی از دست رفته هستند که باید قبل از ساخت یک مدل پیش‌بینی، آنها را نیز مدیریت کنیم. ما باید به تمام این سؤالات فکر کنیم و در عین حال هدف از پیش‌بینی موفقیت یک درخواست گرنت را در نظر داشته باشیم.

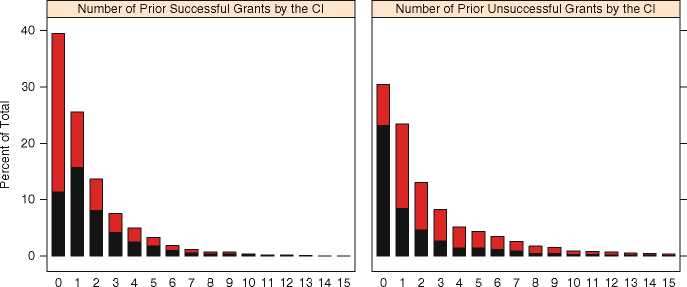
با توجه به این هدف، اقدامات زیر را انجام دادیم. ابتدا، گروهی از پیش‌بینی‌کننده‌ها ایجاد شد که توضیح می‌دهد که چه تعداد محقق که بر اساس نقش تقسیم شده‌اند (مثلاً محقق ارشد) در این گرنت بودند. دوم، گروه‌هایی از ­متغیرهای تعداد نقش خاص نیز برای زبان مادری، ملیت، مدرک تحصیلی، سال تولد، بخش و تاریخچه گرنت ایجاد شد. به‌عنوان مثال، یک متغیر تعداد محققین ارشد از استرالیا را می‌شمارد، در حالی که متغیر دیگر تعداد کل گرنت موفق از همه محققان واگذار شده در این گرنت را محاسبه می‌کند. برای داده‌های انتشار، تعداد کل انتشارات در چهار سطح مجلات ­در همه نقش‌ها جمع‌آوری شد. مدت زمان استخدام به‌طور مشابه در همه نقش‌ها تجمیع شد.

متغیرهای شاخص برای هر کد حامی و دسته گرنت نیز ایجاد شد. برای کدهایRFCD وSEO از تعداد درصدهای غیر صفر برای هر گرنت استفاده شد. در نهایت، شاخص‌هایی برای ماه و روز یا هفته‌ای که گرنت ارائه شد، تولید شد. در مجموع،1784 پیش‌بینی‌کننده ممکن با استفاده از این طرح رمزگذاری ایجاد شد.

در نتیجه، اکثریت وسیع این پیش‌بینی‌کننده‌ها ماهیت گسسته با مقادیر0 زیادی دارند (یعنی یا متغیرهای ساختگی 0/1 و یا متغیر شمارشی). از آنجایی که بسیاری از پیش‌بینی‌کننده‌ها ماهیت طبقه‌بندی دارند، مقادیر گمشده به‌عنوان «ناشناخته» کدگذاری شدند. به‌عنوان مثال، 912 گرنت دارای کدهای دسته‌بندی نشده بودند. یک پیش‌بینی باینری برای دسته‌های گرنت از دست رفته برای دریافت این اطلاعات ایجاد شد.

همانطور که در فصل توضیح داده شد.  [3 ،](#bookmark131) برخی از مدل‌های پیش‌بینی محدودیت‌های متفاوتی در نوع پیش‌بینی‌کننده‌هایی دارند که می‌توانند از آن‌ها استفاده کنند. برای مثال، در این داده‌ها، تعداد قابل‌توجهی از پیش‌بینی‌کننده‌ها همبستگی‌های مطلق زوجی داشتند که بزرگ‌تر از 0. 99 بود. به همین دلیل، یک فیلتر با همبستگی بالا در مجموعه پیش‌بینی برای حذف این پیش‌بینی‌کننده‌های بسیار زائد از داده‌ها استفاده شد. در نهایت56 پیش‌بینی به همین دلیل از داده‌ها حذف شدند. ماهیت دوتایی بسیاری از پیش‌بینی‌کننده‌ها نیز منجر به بسیاری از مواردی شد که داده‌ها بسیار پراکنده و نامتعادل بودند. برای کدهای RFCD،95 درصد از پیش‌بینی‌کننده‌ها کمتر از50 شاخص غیر صفر داشتند. این درجه بالای عدم تعادل طبقاتی نشان می‌دهد که بسیاری از پیش‌بینی‌کننده‌ها را می‌توان به‌عنوان *پیش‌بینی واریانس نزدیک به صفر طبقه‌بندی* کرد که در فصل توضیح داده شد.  [3 ،](#bookmark131) که می‌تواند منجر به مسائل محاسباتی در بسیاری از مدل‌ها شود.

موفق ناموفق |



شکل 12. 1: دو پیش‌بینی پیوسته برتر مرتبط با موفقیت گرنت بر اساس داده‌های قبل از سال 2008. موفقیت قبلی در دریافت گرنت توسط محقق ارشد و همچنین شکست قبلی در دریافت گرنت بیشتر ­با موفقیت یا شکست دریافت گرنت آینده مرتبط است. محور *x به* 15 گرانت کوتاه شده است تا دنباله‌های بلند توزیع تفاوت‌ها را مبهم نکند.

از آنجایی که همه مدل‌ها تحت تأثیر این موضوع قرار نمی‌گیرند، بسته به مدل از دو مجموعه مختلف از پیش‌بینی‌کننده‌ها استفاده شد. «مجموعه کامل» پیش‌بینی‌کننده‌ها شامل همه متغیرها بدون توجه به توزیع آنها می‌شد (1070 پیش‌بینی‌کننده). « ­مجموعه‌ی جدید» برای مدل‌هایی که به پیش‌بینی‌کننده‌های پراکنده و نامتعادل حساس هستند توسعه داده شد ­و شامل ۲۵۲ پیش‌بینی بود. در فصل‌ها و بخش‌های بعدی، متن توضیح خواهد داد که کدام مجموعه پیش‌بینی برای هر مدل استفاده شده است. [[32]](#footnote-32) به‌عنوان یادآوری، فرآیند حذف پیش‌بینی‌کننده‌ها بدون اندازه‌گیری ارتباط آن‌ها با نتیجه، *انتخاب ویژگی بدون نظارت است.* اگرچه چند مدل که از انتخاب ویژگی نظارت شده استفاده می‌کنند در این فصل توضیح داده شده است، بحث گسترده‌تری از انتخاب ویژگی تا فصل ارائه شده است.  [19 .](#bookmark864)

یک بررسی گذرا و تک متغیره از داده‌های رمزگذاری شده جدید چند رابطه جالب را با پاسخ آشکار می‌کند. دو پیش‌بینی‌کننده پیوسته، تعداد درخواست‌های گرنت موفق و ناموفق قبلی توسط محقق ارشد، با موفقیت درخواست گرنت بسیار مرتبط بود. توزیع این ­پیش‌بینی‌کننده‌ها بر اساس موفقیت درخواست گرنت فعلی در شکل 1 نشان داده شده است.  [12. 1](#bookmark20) . جای تعجب نیست که این هیستوگرام‌ها نشان می‌دهد که موفقیت قبلی یا شکست، توزیع مربوطه را به سمت موفقیت یا شکست فعلی تغییر می‌دهد. با توجه به این دانش، انتظار داریم که این پیش‌بینی‌کننده‌ها نقش مهمی را برای اکثر مدل‌های طبقه‌بندی ایفا کنند.

جدول 12. 1: آمار برای سه پیش‌بینی طبقه‌بندی با بالاترین ­ارتباط تک متغیره با تأمین مالی موفقیت آمیز گرنت

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | موفقیت عطا کن | | ن | درصد | شانس | نسبت شانس |
| آره | خیر |
| باند ارزش قرارداد  آ  گروه‌های دیگر | 1 *,* 501  2 *,* 302 | 818  3،569 | 2،319  5،871 | 64. 7  39. 2 | 1. 835  0. 645 | 2. 84 |
| اسپانسر  ناشناس  شناخته شده | 732  3 *,* 071 | 158  4،229 | 890  7 *,* 300 | 82. 2  42. 1 | 4. 633  0. 726 | 6. 38 |
| ماه  ژانویه  ماه‌های دیگر | 480  3،323 | 45  4،342 | 525  7،665 | 91. 4 10. 667  43. 4 0. 765 | | 13. 93 |

سه پیش‌بینی طبقه‌بندی (باند ارزش قرارداد A، ناشناخته حامی و ژانویه) بیشترین ارتباط تک متغیره را با موفقیت درخواست گرنت داشتند. ارتباط برای این سه پیش‌بینی قوی نبود، اما برخی از الگوهای مفید را نشان می‌دهد. جدول [12. 1](#bookmark552) داده‌ها را نشان می‌دهد و پیشنهاد می‌کند که ارسال‌های گرنت با ارزش پولی زیاد، حامی مالی ناشناخته، یا ارسالی در ژانویه با موفقیت بیشتر در تأمین مالی همراه است. با نگاهی متفاوت به مسأله، درخواست‌های گرنت ناموفق احتمالاً ارزش پولی کمتری دارند، یک حامی شناخته شده دارند و در یک ماه غیر از ژانویه ارسال می‌شوند. جدول دارای نرخ موفقیت برای هر گروه و همچنین *شانس* است که نسبت احتمال گرنت به احتمال گرنت ناموفق است. یکی از روش‌های رایج برای ­کمّی کردن توانایی پیش‌بینی یک پیش‌بینی باینری (مانند اینها) *نسبت شانس است.* به‌عنوان مثال، زمانی که گرنت در ژانویه ارائه می‌شود، شانس بسیار بالاتر (10. 7) از ماه‌های دیگر (0. 8) است. نسبت شانس برای این پیش‌بینی نشان می‌دهد که گرنت ارائه شده در ژانویه 13. 9 برابر بیشتر از ماه‌های دیگر موفق هستند. با توجه به نسبت‌های شانس بالا، انتظار ­می‌رود که این پیش‌بینی‌کننده‌ها نیز بر توسعه اکثر مدل‌های طبقه‌بندی تأثیر بگذارند.

در نهایت، ما باید نحوه تقسیم داده‌ها را انتخاب کنیم که مستقیماً مشخص نیست. قبل از تصمیم‌گیری در مورد رویکرد تقسیم، ذکر این نکته مهم است که ­درصد گرنت موفق در طول سال‌ها متفاوت بوده است: 45٪ (2005)، 51. 7٪ (2006)، 47. 2٪ (2007) و 36. 6٪ (2008). اگرچه سال 2008 کمترین درصد را در داده‌ها داشت، اما هنوز اطلاعات کافی برای اعلام روند نزولی وجود ندارد. طرح تقسیم داده‌ها باید *دامنه کاربرد* مدل را در نظر بگیرد: چگونه از آن استفاده می‌شود و معیار برای ارزیابی مناسب بودن آن برای هدف چیست؟ هدف از تمرین مدل ایجاد یک مدل پیش‌بینی برای تعیین کمیت احتمال موفقیت برای گرنت جدید است، به همین دلیل است که رقابت از جدیدترین داده‌ها برای اهداف آزمایشی استفاده می‌کند.

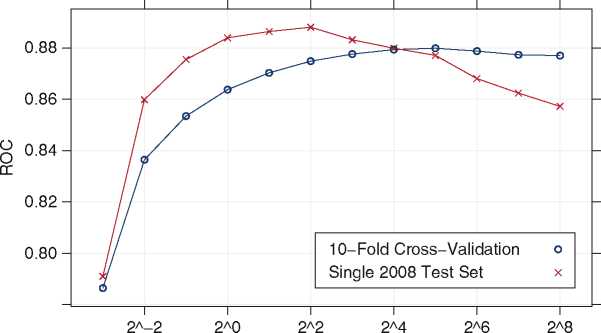
اگر میزان موفقیت گرنت در طول سال‌ها نسبتاً ثابت بود، یک ­استراتژی منطقی تقسیم داده‌ها نسبتاً ساده خواهد بود: تمام داده‌های موجود از سال 2005 تا 2008 را در نظر بگیرید، برخی از داده‌ها را برای یک مجموعه آزمایشی ذخیره کنید و از نمونه‌گیری مجدد با بقیه نمونه‌ها استفاده کنید. برای تنظیم مدل‌های مختلف با این حال، یک نمونه آزمایش تصادفی در تمام سال‌ها احتمالاً منجر به مجموعه‌ای از آزمون کمتر *مرتبط می‌شود.* در واقع، ما در حال ساخت مدل‌هایی هستیم که بر محیط درخواست گرنت *گذشته متمرکز هستند.*

یک استراتژی جایگزین ایجاد مدل‌هایی با استفاده از داده‌های قبل از سال 2008 است، اما آنها را بر اساس میزان برازش آنها با داده‌های سال 2008 تنظیم کنید. اساساً، داده‌های سال 2008 به‌عنوان یک مجموعه آزمایشی واحد عمل می‌کنند که از نظر زمانی با مجموعه آزمایشی داده‌های اولیه از سال 2009 تا 2010 مرتبط‌تر است. با این حال، این یک «نگاه» واحد به داده‌هایی است که هیچ معیار واقعی را ارائه نمی‌کند. عدم قطعیت برای عملکرد مدل مهمتر از آن، این استراتژی ممکن است منجر به بیش برازش قابل‌توجهی با این مجموعه خاص از داده‌های سال 2008 شود و ممکن است به خوبی به سال‌های بعدی تعمیم نیابد. برای مثال، مانند مدل‌های رگرسیونی، تعدادی مدل طبقه‌بندی وجود دارد که به‌طور خودکار انتخاب ویژگی را در حین ساخت مدل انجام می‌دهند. یکی از خطاهای بالقوه روش شناسی که ممکن است با یک مجموعه آزمایشی منفرد که بارها ارزیابی می‌شود رخ دهد این است که ممکن است مجموعه‌ای از پیش‌بینی‌کننده‌ها انتخاب شوند که فقط برای این برنامه‌های گرنت خاص در سال 2008 کار کنند. تا زمانی که مجموعه دیگری از درخواست‌های گرنت اخیر ارزیابی نشود، راهی برای دانستن این موضوع نداریم.

چگونه این دو رویکرد برای این داده‌ها مقایسه می‌شوند؟ شکل [12. 2](#bookmark554) نتایج یک مدل طبقه‌بندی ماشین بردار پشتیبان را نشان می‌دهد که به تفصیل در بخش بحث شده است.  [13. 4](#bookmark32) اما شبیه مدل رگرسیون بردار پشتیبان است که در بخش توضیح داده شده است.  [7. 3 .](#bookmark356) با استفاده از هسته تابع پایه شعاعی که قبلاً ­مورد بحث قرار گرفت، پارامترهای تنظیم عبارتند از پارامتر هسته، *a و* مقدار هزینه، *C که* برای کنترل بیش برازش استفاده می‌شود. چندین مقدار از پارامتر هسته تابع پایه شعاعی و همچنین چندین مقدار تابع هزینه مورد ارزیابی قرار گرفت.

شکل [12. 2](#bookmark554) هر دو رویکرد را برای تنظیم مدل نشان می‌دهد. دو پروفایل پارامتر تنظیم برای ماشین بردار پشتیبانی نشان داده شده است:

* مدل اول بر اساس 8189 گرنت ساخته شده است که شامل تمام داده‌های قبل از 2008 و 25٪ از داده‌های سال 2008 ( *n* = 290) است. برای انتخاب پارامتر(های منظم‌سازی ­) و ker nel، از اعتبارسنجی متقاطع 10 برابری استفاده می‌شود. نمایه عملکرد در سراسر پارامتر هزینه به صورت یک خط آبی در شکل نشان داده شده است.  [12. 2](#bookmark554) (این ­فایل حرفه‌ای از مقدار بهینه پارامتر هسته استفاده می‌کند). مجموعه‌ای از گرنت 2008 ( *n* = 1،785) برای تأیید انتخاب پارامتر تنظیم نهایی (پروفایل آبی) عقب مانده است.
* مدل دوم به‌طور انحصاری بر اساس داده‌های قبل از سال 2008 ساخته شده است و مقدار پارامتر تنظیم برای به حداکثر رساندن سطح زیر منحنی ROC برای گرنت 2008 انتخاب شده است. هیچ نمونه اضافی برای تأیید انتخاب پارامتر (پروفایل قرمز) ذخیره نمی‌شود.



سی

شکل 12. 2: دو مدل برای موفقیت گرنت بر اساس داده‌های قبل از سال 2008 اما با مجموعه داده‌های متفاوتی که برای تنظیم مدل استفاده شده است.

نمایه آبی نشان می‌دهد که مقدار 32 برای پارامتر هزینه، ناحیه زیر منحنی ROC 0. 88 را به دست می‌دهد. نمایه قرمز تنها نتایج ارزیابی داده‌های سال 2008 را نشان می‌دهد (یعنی بدون نمونه‌گیری مجدد). در اینجا، فرآیند تنظیم نشان می‌دهد که برای دستیابی به یک مدل بهینه با مساحت زیر منحنی ROC 0. 89، به ارزش هزینه کمتری نیاز است (4). اولا، با توجه به مقدار داده برای ارزیابی مدل، مسأله ساز است که منحنی‌ها پارامترهای تنظیم متفاوتی را پیشنهاد می‌کنند. ثانیا، زمانی که مدل اعتبار متقابل بر روی داده‌های 2008 ارزیابی می‌شود، سطح زیر منحنی ROC به‌طور قابل‌توجهی کوچک‌تر است (0. 83) از نتایج اعتبارسنجی متقابل.

سازش در اینجا ساختن مدل‌هایی بر روی داده‌های قبل از سال 2008 و تنظیم آن‌ها با ارزیابی یک نمونه تصادفی از 2075 گرنت از سال 2008 است. پس از تعیین پارامترهای بهینه، مدل نهایی با استفاده از این ­پارامترها و کل مجموعه آموزشی ساخته می‌شود. ، داده‌های قبل از سال 2008 و 2075 گرنت اضافی). مجموعه کوچکی از 518 گرنت از سال 2008 استفاده خواهد شد تا اطمینان حاصل شود که هیچ خطای روش شناسی فاحشی از ارزیابی مکرر ­داده‌های 2008 در طول تنظیم مدل رخ نمی‌دهد. در متن، این مجموعه از نمونه‌ها *مجموعه نگهدارنده 2008 نامیده می‌شود.* این مجموعه کوچک از گرنت سال 2008 به‌عنوان *مجموعه آزمایشی نامیده می‌شود* و تا زمانی که مجموعه‌ای از مدل‌های کاندید شناسایی نشود، ارزیابی نخواهد شد (در فصل 15 ). این استراتژی‌ها در جدول خلاصه شده است [12. 2 .](#bookmark557)

برای روشن بودن، هیچ رویکرد واحد و تمیزی برای رسیدگی به این موضوع برای داده‌هایی که به نظر می‌رسد در طول زمان در حال تکامل هستند وجود ندارد. بنابراین، متخصص باید اهداف مدل‌سازی را درک کند و با دقت برنامه‌ای برای آموزش و آزمایش مدل‌ها ایجاد کند. در این مورد، داده‌های گرنت دارای تجملات نسبتاً متوسطی از داده‌های اخیر هستند. داده‌های کافی برای تقسیم یک مجموعه نگهدارنده کوچک وجود دارد

جدول 12. 2: شماتیکی برای استراتژی تقسیم داده‌ها برای داده‌های درخواست گرنت ­مورد استفاده در این فصل و فصل‌های بعدی

|  |  |  |  |
| --- | --- | --- | --- |
|  | تیونینگ مدل | مدل نهایی |  |
|  | هولدات آموزشی | آموزش | Holdout |
| قبل از 2008 ( *n* = 6 *,* 633) | *ایکس* | *ایکس* |  |
| 2008 ( *n* = 1 *,* 557)  2008 ( *n* = 518) | *ایکس* | *ایکس* | *ایکس* |

بدون اینکه به‌طور قابل‌توجهی روند تنظیم را مختل کند. معایب این روش عبارتند از:

1. این فرض وجود دارد که پارامترهای مدل به دست آمده از فرآیند تنظیم برای مدل نهایی که از داده‌های قبل از 2008 و همچنین 2075 گرنت از سال 2008 استفاده می‌کند، مناسب خواهند بود.
2. از آنجایی که مدل نهایی از گرنت 2008 استفاده می‌کند، عملکرد مجموعه آزمایشی احتمالاً بهتر از نتایج تولید شده در فرآیند تنظیم (جایی که پارامترهای مدل در معرض گرنت سال 2008 قرار نگرفته اند) خواهد بود.

در فصل 15، نتایج مجموعه تست با نتایج حاصل از تنظیم مدل مقایسه خواهد شد.

**رگرسیون لجستیک**

رگرسیون خطی (بخش [6. 2 )](#bookmark286) مدلی را تشکیل می‌دهد که در پارامترها خطی است و این پارامترها با به حداقل رساندن مجموع مجذور باقیمانده به دست می‌آیند. معلوم می‌شود که مدلی که مجموع مجذور باقیمانده را به حداقل می‌رساند، زمانی که منطقی باشد فرض کنیم که باقیمانده‌های مدل از توزیع نرمال (یعنی گوسی) پیروی می‌کنند، تخمین‌های حداکثر درستنمایی *پارامترها را نیز تولید می‌کند.*

تخمین پارامتر حداکثر درستنمایی تکنیکی است که می‌تواند زمانی مورد استفاده قرار گیرد که مایل به فرضیاتی در مورد توزیع احتمال داده‌ها باشیم. بر اساس توزیع احتمال نظری و داده‌های مشاهده شده، تابع درستنمایی یک عبارت احتمالی است که می‌تواند در مورد مجموعه خاصی از مقادیر پارامتر ساخته شود. اگر دو مجموعه از مقادیر پارامترها شناسایی شوند، مجموعه‌ای که احتمال بیشتری دارد با داده‌های مشاهده شده سازگارتر تلقی می‌شود.

توزیع احتمالی که اغلب در صورت وجود دو کلاس استفاده می‌شود، توزیع دو جمله‌ای است. [[33]](#footnote-33) این توزیع دارای یک پارامتر واحد است، *p که* احتمال یک رویداد یا یک کلاس خاص است. برای داده‌های گرنت، فرض کنید *p* احتمال موفقیت آمیز گرنت است. در کمک‌های بلاعوض قبل از سال 2008، در مجموع 6633 کمک بلاعوض وجود داشت که از این تعداد 3233 مورد موفق بودند. در اینجا، شکل تابع درستنمایی دو جمله‌ای خواهد بود

*L*(*p*) =

6633 3233

3233 *p*

(1 *- p*)

6633-3233

(12.1)

که در آن توان‌های *p* و 1 *- p* فرکانس‌های کلاس‌ها را در داده‌های مشاهده شده منعکس می‌کنند. بخش اول معادله "n انتخاب r" است و راه‌های احتمالی را که می‌تواند 3233 موفقیت و 3400 شکست در داده‌ها وجود داشته باشد را توضیح می‌دهد.

برآوردگر حداکثر درستنمایی مقدار *p را پیدا می‌کند* که ­بیشترین مقدار را برای *f (* *p* ) تولید می‌کند. به نظر می‌رسد که نسبت نمونه، 3233 */* 6633 = 0 *است.* 487، برآورد حداکثر احتمال در این وضعیت است.

با این حال، ما می‌دانیم که میزان موفقیت تحت تأثیر عوامل متعددی قرار می‌گیرد و مایلیم مدلی بسازیم که از آن عوامل برای تولید تخمین احتمالی دقیق تر استفاده کند. در این حالت، مدل را *مجدداً پارامتر می‌*کنیم تا *p* تابعی از این عوامل باشد. مانند رگرسیون خطی، مدل رگرسیون لجستیک علاوه بر پارامترهای شیب برای هر ترم مدل، یک برش دارد. با این حال، از آنجایی که لازم است احتمال رویداد بین 0 و 1 باشد، نمی‌توان تضمین کرد که یک مدل شیب و قطع مقادیر را در این محدوده محدود می‌کند. همانطور که قبلا در این فصل بحث شد، اگر *p* احتمال یک رویداد باشد، شانس این رویداد *p/ (*1 *- p* ) است. رگرسیون لجستیک شانس ورود رویداد را به‌عنوان یک تابع خطی مدل می‌کند:

log (1*—p)*

*— p* 0 + *P*1 *x* 1 + *• • •* + *Pp xp.*

(12.2)

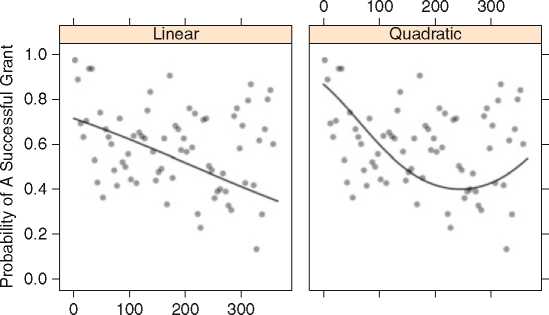
در اینجا *P* تعداد پیش‌بینی‌کننده‌ها است. سمت راست معادله معمولاً به‌عنوان *پیش‌بینی خطی شناخته می‌*شود. از آنجایی که لاگ شانس می‌تواند از *-* به تا *تا* متغیر باشد، هیچ نگرانی در مورد دامنه مقادیری که پیش‌بینی‌کننده‌های خطی ممکن است ایجاد کنند وجود ندارد. با جابجایی برخی اصطلاحات، به تابعی از احتمال رویداد برمی گردیم:

*p* — *?—tt. 7 7 -*

(12.3)

1 + exp [ *— (* *p* o + *p* 1 *x* i + *• • •* + *p p x p* )]

این تابع غیرخطی یک تابع سیگموئیدی از شرایط مدل است و ­تخمین‌های احتمال را بین 0 و 1 محدود می‌کند. همچنین، این مدل ­مرزهای کلاس خطی تولید می‌کند، مگر اینکه پیش‌بینی‌کننده‌های مورد استفاده در مدل باشند.



روز

شکل 12. 3: دو مدل رگرسیون لجستیک که احتمال یک گرنت موفق را به روز عددی سال مرتبط می‌کند. در این نمودار، مقادیر روز در دوره‌های 5 روزه قرار گرفتند. مدل متناسب از نسخه binned پیش‌بینی استفاده نمی‌کند. شانس ورود به سامانه به‌عنوان تابعی از روز سال مدل شد (به‌عنوان مثال، 1، 2،. *. . ،* 365)

نسخه‌های غیرخطی داده‌ها (مثلاً مقادیر مجذور یک پیش‌بینی به‌عنوان یکی از اصطلاحات مدل *x j* استفاده می‌شود).

اکنون که راهی برای ارتباط مدل خود با پارامتر توزیع دو جمله‌ای داریم، می‌توانیم مقادیر نامزد پارامترها ( *P* ) را پیدا کنیم و با داده‌های مشاهده شده، مقدار تابع درستنمایی را محاسبه کنیم. هنگامی که مقادیر *P را* یافتیم که به نظر می‌رسد احتمال داده‌های ما را به حداکثر می‌رسانند، این مقادیر برای پیش‌بینی نتایج نمونه استفاده می‌شوند.

رگرسیون لجستیک و رگرسیون خطی معمولی در دسته بزرگتری از تکنیک‌ها به نام مدل‌های خطی تعمیم‌یافته (GLMs) قرار می‌گیرند که توزیع‌های احتمال مختلفی را در بر می‌گیرد.  [دابسون](#bookmark1015) [( 2002](#bookmark1015) ) مقدمه‌ای عالی ­برای این مدل‌ها ارائه می‌دهد. آنها خطی هستند به این معنا که برخی از تابع‌های نتیجه با استفاده از پیش‌بینی‌کننده‌های خطی، مانند شانس ورود به سامانه در معادله مدل‌سازی می‌شوند.  [12. 2](#bookmark21) . آنها اغلب معادلات غیر خطی را تولید می‌کنند (مانند معادله *p* در معادله.  [12. 3 )](#bookmark21) . باز هم توجه داشته باشید که حتی اگر معادله *p* غیرخطی است، مرزهای طبقه‌بندی خطی را ایجاد می‌کند.

به‌عنوان مثال، ما می‌توانیم یک مدل رگرسیون لجستیک ساده را با استفاده از یک پیش‌بینی منفرد، مانند روز عددی سال، به داده‌های گرنت برسانیم. شکل [12. 3](#bookmark558) زمانی که داده‌ها در فواصل 5 روزه قرار می‌گیرند، احتمال مشاهده موفقیت آمیز گرنت را نشان می‌دهد. در این طرح، میزان موفقیت بیشتر مربوط به آغاز و پایان سال است. در واقع، برای داده‌های قبل از سال 2008، هیچ گرنت ناموفق در مجموعه 343 گرنت ارسال شده در روز اول سال وجود نداشت. در اواسط سال میزان پذیرش تقریباً کاهش می‌یابد اما نزدیک به پایان سال افزایش می‌یابد. یک مدل رگرسیون لجستیک ساده سعی می‌کند یک شیب مربوط به روز و همچنین یک عبارت عرض از مبدأ را تخمین بزند. با استفاده از داده‌های آموزشی، مدل برازش معمولی مقادیر مختلف این دو پارامتر را جستجو می‌کند تا ترکیبی را بیابد که با توجه به داده‌های آموزشی مشاهده‌شده، احتمال توزیع دوجمله‌ای را به حداکثر می‌رساند. در انتها، وقفه برآورد شده 919/0 و پارامتر شیب 0- تعیین *شد.* 0042. این به این معنی است که یک *کاهش* در روز در شانس ورود به سامانه 0. 0042 وجود دارد. برازش مدل در پانل سمت چپ شکل نشان داده شده است.  [12. 3 .](#bookmark558) این به اندازه کافی نشان دهنده روند در اواخر سال نیست. مدل دیگری را می‌توان ایجاد کرد که در آن پارامتر سوم با یک عبارت مجذور روز مطابقت دارد. برای این مدل، وقفه تخمینی اکنون 1. 88 می‌شود، شیب ترم روز خطی *-* 0 *است. 019 و شیب برای ترم درجه دوم -* 0 برآورد شد. 000038. پانل سمت راست شکل.  [12. 3](#bookmark558) بهبود آشکاری را در مدل نشان می‌دهد، اما افزایش نرخ موفقیت در پایان سال را کاملاً نشان نمی‌دهد. به‌عنوان شواهدی از نیاز به یک عبارت اضافی، مساحت زیر منحنی ROC برای مدل خطی 0. 56 بود که پس از استفاده از عبارت مدل اضافی به 0. 66 بهبود می‌یابد.

یک مدل رگرسیون لجستیک مؤثر مستلزم بازرسی از چگونگی نرخ موفقیت مربوط به هر یک از پیش‌بینی‌کننده‌های پیوسته است و بر این اساس، ممکن است شرایط مدل را برای محاسبه اثرات غیرخطی پارامتر کند. یک روش کارآمد برای انجام این کار در مورد بحث قرار گرفته است [هارل](#bookmark1018) [( 2001](#bookmark1018) ) که در آن اسپلاین‌های مکعبی محدود برای ایجاد نسخه‌های منعطف و تطبیقی از پیش‌بینی‌کننده‌ها استفاده می‌شود که می‌تواند بسیاری از انواع غیرخطی‌ها را ثبت کند. بخش محاسبات فصل ­جزئیات بیشتری در مورد این روش دارد. رویکرد دیگر یک مدل افزایشی تعمیم یافته است [( هستی و تیبشیرانی 1990](#bookmark1018) ; [هستی و همکاران 2008](#bookmark1018) ) که همچنین از روش‌های رگرسیون انعطاف‌پذیر (مانند splines) برای مدل‌سازی تطبیقی شانس ورود استفاده می‌کند. برای آشنایی بیشتر با این روش‌ها خواننده را به متون مرجع ارجاع می‌دهیم.

برای داده‌های گرنت، مجموعه کامل پیش‌بینی‌کننده‌ها در یک ­مدل رگرسیون لجستیک استفاده شد. سایر پیش‌بینی‌کننده‌های پیوسته برای ­روابط غیرخطی ارزیابی شدند. با این حال، بسیاری از پیش‌بینی‌کننده‌ها نقاط داده کمی در یک یا چند نقطه از توزیع‌ها دارند، مانند دو پیش‌بینی نشان‌داده‌شده در شکل.  [12. 1 .](#bookmark20) این مسأله در تجویز فرم عملکردی دقیق برای پیش‌بینی‌کننده‌ها را افزایش می‌دهد. با استفاده از این مجموعه پیش‌بینی، مدل رگرسیون لجستیک قادر به دستیابی به مساحت زیر منحنی ROC 0. 78، حساسیت 77 درصد و ویژگی 76. 1 درصد در مجموعه نگه‌داری 2008 بود.

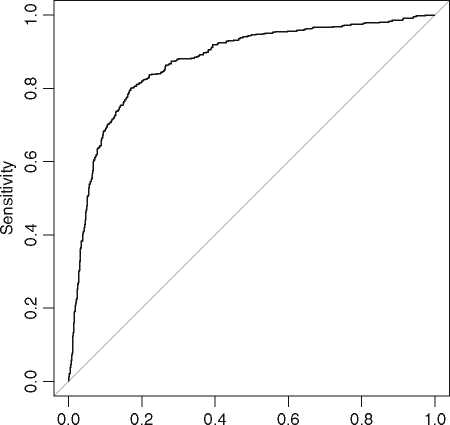
بسیاری از پیش‌بینی‌کننده‌های طبقه‌بندی دارای توزیع‌های پراکنده و نامتعادل هستند. به همین دلیل، ما انتظار داریم که مدلی که از مجموعه کامل پیش‌بینی‌کننده‌ها استفاده می‌کند، عملکرد بدتری نسبت به مجموعه‌ای داشته باشد که پیش‌نمایش‌های واریانس نزدیک به صفر ­حذف شده است. برای مجموعه کاهش یافته 253 متغیر، مساحت زیر منحنی ROC 0. 87، حساسیت 80. 4 درصد و ویژگی 82. 2 درصد بود (شکل 1.  [12. 4 )](#bookmark560) . ماتریس سردرگمی در جدول نشان داده شده است [12. 3](#bookmark560) . با این ­مدل خاص، با حذف این پیش‌بینی‌کننده‌ها، بهبود قابل‌توجهی حاصل شد.

برای رگرسیون لجستیک، آزمون‌های فرضیه آماری رسمی را می‌توان برای ارزیابی اینکه آیا ضرایب شیب برای هر پیش‌بینی از نظر آماری معنی‌دار هستند، ­انجام داد. آماره *Z* معمولاً برای این مدل‌ها استفاده می‌شود ( [دابسون 2002](#bookmark1015) )، و

جدول 12. 3: ماتریس سردرگمی مجموعه نگهدارنده 2008 برای مدل رگرسیون لجستیک

|  |  |  |
| --- | --- | --- |
|  | کلاس مشاهده شده | |
| موفقیت آمیز | ناموفق |
| موفقیت آمیز | 439 | 236 |
| ناموفق | 131 | 751 |

این مدل دارای دقت کلی 76. 4 درصد، حساسیت 77 درصد و ویژگی 76. 1 درصد بود.



1 - خاص بودن

شکل 12. 4: منحنی ROC برای مجموعه آزمون داده‌های گرنت با استفاده از مدل رگرسیون لجستیک. AUC 0. 87 است

اساساً معیاری از نسبت سیگنال به نویز است: شیب برآورد شده بر خطای استاندارد مربوطه تقسیم می‌شود. با استفاده از این آمار، ­عوامل پیش‌بینی را می‌توان رتبه‌بندی کرد تا بفهمند کدام عبارات بیشترین تأثیر را بر مدل داشته‌اند. برای این داده‌ها، پنج پیش‌بینی مهم عبارت بودند از تعداد گرنت ناموفق توسط محققین ارشد، تعداد گرنت موفق توسط محققین ارشد، باند ارزش قرارداد F، باند ارزش قرارداد E و روز عددی سال (مربع).

مدل رگرسیون لجستیک به دلیل سادگی و توانایی آن در بیان عبارات استنباطی در مورد اصطلاحات مدل بسیار محبوب است. به‌عنوان مثال، یک محقق ممکن است بخواهد به‌طور رسمی ارزیابی کند که آیا روز سال تقویمی رابطه آماری معناداری با احتمال گرنت دارد یا خیر.

پذیرش - پذیرفته شدن.  [هارل](#bookmark1018) [( 2001](#bookmark1018) ) یک منبع عالی برای توسعه مدل‌های آماری به منظور ایجاد اظهارات استنباطی در مورد ­پارامترهای مدل است.

این مدل همچنین زمانی می‌تواند مؤثر باشد که هدف صرفاً پیش‌بینی باشد، اما همانطور که در بالا نشان داده شد، کاربر را ملزم می‌کند تا بازنمایی‌های مؤثری ­از داده‌های پیش‌بینی را شناسایی کند که بهترین عملکرد را دارند. همانطور که در بخش‌های بعدی نشان داده خواهد شد، مدل‌های طبقه‌بندی دیگری وجود دارند که به‌طور تجربی این روابط را در دوره آموزش مدل استخراج می‌کنند. اگر مدل فقط برای پیش‌بینی مورد استفاده قرار گیرد، این تکنیک‌ها ممکن است سودمندتر باشند.

تحلیل تشخیص خطی

ریشه LDA به [فیشر ( 1936](#bookmark1016) ) و [ولش](#bookmark1027) [( 1939](#bookmark1027) ). هر یک از این محققین دیدگاه متفاوتی در مورد مسئله دستیابی به قوانین طبقه‌بندی بهینه داشتند. با این حال، همانطور که خواهیم دید، هر کدام قانون یکسانی را در تنظیمات طبقه‌بندی دو گروهی پیدا کردند. در این بخش، نکات برجسته هر دوی این رویکردها را ارائه خواهیم کرد تا جزئیات فنی لازم را به دست آوریم و همچنین در مورد چند ساختار ریاضی مورد نیاز برای ارتباط آن با سایر روش‌های مورد بحث در این فصل بحث خواهیم کرد.

برای مسأله طبقه بندی، [ولش](#bookmark1027) [( 1939](#bookmark1027) ) رویکرد به حداقل رساندن ­احتمال کل طبقه‌بندی اشتباه را اتخاذ کرد که به احتمالات طبقاتی ­و توزیع‌های چند متغیره پیش‌بینی‌کننده‌ها بستگی دارد. برای دیدن رویکرد ولش، ابتدا به درک اساسی از قانون بیز نیاز داریم[[34]](#footnote-34) که هست

*Pr* [ *Y = C* ] *Pr* [ *X |Y =* C *]* £   
C *= 1 Pr* [ *Y = Cl* ] *Pr* [ *X | Y* = *Cl* ]

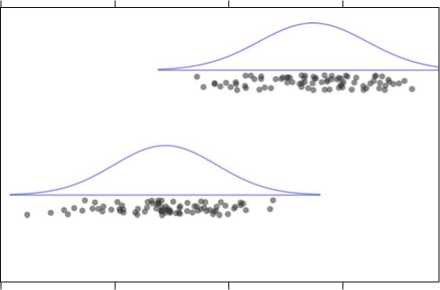
*Pr* [ *Y = C£X* ] =

(12.4)

*Pr* [ *Y = C £* ] به‌عنوان *احتمال* قبلی عضویت در کلاس *C شناخته می‌شود.* در عمل، این مقادیر یا شناخته شده هستند، یا با نسبت نمونه‌ها در هر کلاس تعیین می‌شوند، یا ناشناخته هستند که در این صورت، تمام مقادیر پیشین‌ها برابر هستند. *Pr* [ *X |Y* = *C* ] *احتمال مشروط* مشاهده پیش‌بینی *X* است، با توجه به اینکه داده‌ها از کلاس *C سرچشمه می‌گیرند.* در اینجا فرض می‌کنیم که داده‌ها از یک توزیع احتمال (مثلاً توزیع چند متغیره یا ­بد) تولید می‌شوند که سپس شکل ریاضی این کمیت را تعریف می‌کند. نتیجه این معادله *Pr* [ *Y* = *C |X* ] است که معمولاً به‌عنوان *احتمال خلفی نامیده می‌شود* که نمونه، *X،* عضوی از کلاس *C* است. برای توضیح دقیق تر این معادله شما را به بخش Sect ارجاع می‌دهیم.  [13. 6](#bookmark648) .

برای یک مسئله طبقه‌بندی دو گروهی، قاعده‌ای که احتمال کل طبقه‌بندی اشتباه را به حداقل می‌رساند، طبقه‌بندی *X* به گروه 1 است اگر *Pr* [ *Y* = *C* 1 *|X* ] *>Pr* [ *Y* = *C2 |X* ] و در گروه 2 اگر نابرابری معکوس می‌شود با استفاده از معادله 12. 4، این قانون مستقیماً به طبقه‌بندی *X* در گروه 1 اگر ترجمه می‌شود

*Pr* [ *Y* = *C* 1 ] *Pr* [ *X |Y* = *C* 1 ] *>Pr* [ *Y* = *C* 2 ] *Pr* [ *X |Y* = *C* 2 ]. (12. 5)



Class 2

Class 1

68

Predictor

Fig. 12.5: A single predictor is used to classify samples into two groups. The *blue figures* above each group represent the probability density function for a normal distribution determined by the class-specific means and variances

ما به راحتی می‌توانیم این قانون را به مورد بیش از دو گروه تعمیم دهیم. در این مجموعه، ­اگر *Pr* [ *Y = C e* ] *Pr* [ *X |Y* = *C e* ] بیشترین مقدار را در تمام کلاس‌های *C داشته باشد، X* را به گروه *C e* طبقه‌بندی می‌کنیم.

شکل [12. 5](#bookmark22) این را با یک پیش‌بینی و دو کلاس نشان می‌دهد (نقاط داده‌های فردی برای کاهش هم‌پوشانی آنها "تارپا" شده‌اند). تصاویر آبی بالای هر گروه از نقاط داده، تابع چگالی احتمال برای توزیع نرمال برای هر یک از کلاس‌ها هستند (یعنی *Pr* [ *X|Y* = *C* 1 ] و *Pr* [ *X |Y* = *C2* ]). از آنجایی که یک پیش‌بین واحد وجود دارد، یک نمونه جدید ­با یافتن مقدار آن در محور *x* طبقه‌بندی می‌شود، سپس مقدار هر یک از توابع چگالی احتمال برای هر کلاس تعیین می‌شود (علاوه بر احتمال کلی، *Pr* [ *X* ]، با ادغام هر دو گروه پیدا شد). فرض کنید یک نمونه جدید دارای مقدار 4 برای پیش‌بینی است. احتمال کلاس 2 تقریباً 0 است، بنابراین پیش‌بینی می‌شود که این نمونه متعلق به کلاس اول باشد.

از آنجایی که برای این مثال از یک پیش‌بین واحد استفاده می‌شود، پیچیدگی استفاده از قانون بیز در عمل را رد می‌کند. برای طبقه بندی، تعداد پیش‌بینی‌کننده‌ها تقریباً همیشه بیشتر از یک است و می‌تواند بسیار زیاد باشد. در موقعیت‌های واقعی تر، چگونه می‌توان کمیت‌هایی مانند *Pr* [ *X |Y* = *C e* ] را در ابعاد مختلف محاسبه کرد؟[[35]](#footnote-35) از چه توزیع‌های احتمال چند متغیره‌ای می‌توان برای این منظور استفاده کرد؟

یک سناریوی خاص که اغلب مورد استفاده قرار می‌گیرد این است که فرض کنیم توزیع پیش‌بینی‌کننده‌ها نرمال چند متغیره است. این توزیع دارای دو پارامتر است: بردار میانگین چند بعدی *p، e* و ماتریس کوواریانس X *e.* علاوه بر این، اگر فرض کنیم که میانگین گروه‌ها منحصر به فرد هستند (یعنی *p، e متفاوت* برای هر گروه)، اما ماتریس‌های کوواریانس در بین گروه‌ها یکسان هستند، می‌توانیم معادله را حل کنیم. 12. 5 یا مسئله چند کلاسه کلی تر برای یافتن ­تابع تمایز خطی گروه *اول* :

*X '* X 1 *^ £ -* 0. 5 *p، g* X 1 *^ £* + log ( *Pr* [ *Y* = *C $* ]). (12. 6)

در عمل، میانگین‌های نظری، *^،* با استفاده از میانگین کلاس خاص ( *x ^* ) تخمین زده می‌شود. ماتریس کوواریانس نظری، S نیز با ماتریس کوواریانس مشاهده شده داده‌ها تخمین زده می‌شود، S و *X* با نمونه مشاهده شده، u جایگزین می‌شود. در مثال ساده در شکل.  [12. 5 ،](#bookmark22) میانگین نمونه و واریانس داده‌ها برای تولید توزیع‌های احتمال نشان داده شده با رنگ آبی کافی بود. برای دو کلاس، میانگین‌ها و واریانس‌های مخصوص کلاس به همراه کوواریانس نمونه بین دو پیش‌بینی (برای پر کردن ماتریس کوواریانس نمونه) محاسبه می‌شود.

توجه کنید که معادله [12. 6](#bookmark563) یک تابع خطی در *X است و* مرزهای کلاس جداکننده را مشخص می‌کند. ­از این رو نام روش: LDA. یک تغییر جزئی در مفروضات - که ماتریس‌های کوواریانس در بین گروه‌ها یکسان نیستند - منجر به تحلیل متمایز درجه دوم می‌شود که در بخش توضیح داده خواهد شد.  [13. 1](#bookmark612) .

فیشر مسئله طبقه‌بندی را به روشی متفاوت فرموله کرد. در این رویکرد، او به دنبال یافتن ترکیب خطی پیش‌بینی‌کننده‌ها به‌گونه‌ای بود که واریانس بین گروهی نسبت به واریانس درون گروهی حداکثر شود. به عبارت دیگر، او می‌خواست ترکیبی از پیش‌بینی‌کننده‌ها را بیابد که حداکثر تفکیک را بین مراکز داده‌ها ایجاد می‌کند و در عین حال تغییرات در هر گروه از داده‌ها را به حداقل می‌رساند.

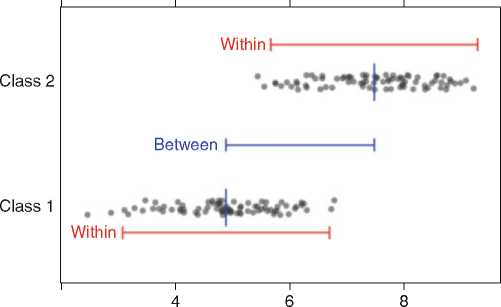
برای نشان دادن این مفهوم، شکل.  [12. 6](#bookmark564) آنالوگ شکل 1 است.  [12. 5 .](#bookmark22) در اینجا، نوارهای آبی نشان دهنده میانگین کلاس است. از آنجایی که یک پیش‌بینی وجود دارد، واریانس بین گروه‌ها مجذور تفاوت در این میانگین‌ها است. واریانس درون گروهی با واریانسی تخمین زده می‌شود که واریانس‌های پیش‌بینی را در هر گروه جمع می‌کند (با استفاده از نوارهای قرمز در شکل نشان داده شده است). در نظر گرفتن نسبت این دو کمیت، در واقع، نسبت سیگنال به نویز است. رویکرد فیشر ترکیب‌های خطی پیش‌بینی‌کننده‌ها ­را برای به حداکثر رساندن نسبت سیگنال به نویز تعیین می‌کند. مشابه بحث قبلی در مورد رویکرد ولش، با افزودن پیش‌بینی‌کننده‌های اضافی، وضعیت به‌طور فزاینده‌ای پیچیده‌تر می‌شود. واریانس‌های بین و درون گروهی به محاسبات پیچیده‌ای تبدیل می‌شوند که شامل ساختار کوواریانس پیش‌بینی‌کننده‌ها و غیره می‌شود.

ریاضی، اجازه دهید B نشان دهنده ماتریس کوواریانس بین گروهی و W نشان دهنده ماتریس کوواریانس درون گروهی باشد. سپس مسئله فیشر را می‌توان به‌عنوان یافتن مقدار *b به* گونه‌ای فرمول‌بندی کرد که

*V* B *b   
b ' W b*

(12.7)

به حداکثر می‌رسد. راه حل این مسئله بهینه‌سازی بردار ویژه مربوط به بزرگترین مقدار ویژه W *-* 1 B است. این بردار یک تفکیک کننده خطی است و متمایز کننده‌های بعدی از طریق همان عملیات یافت می‌شوند.



پیش‌بینی

شکل 12. 6: همان داده‌هایی که در شکل نشان داده شده است.  [12. 5](#bookmark22) . در اینجا، واریانس‌های بین و درون کلاس نشان داده شده است. محدوده‌های درون کلاس بر اساس میانگین *±* دو انحراف استاندارد است

زمان‌بندی منوط به این محدودیت است که جهت‌های جدید با تمایزات قبلی همبستگی ندارند.

برای ملموس تر کردن رویکرد فیشر، اجازه دهید تنظیمات دو گروهی را در نظر بگیریم. حل معادله [12. 7](#bookmark563) برای دو گروه تابع تمایز *S -* 1 را می‌دهد ( x i *—* x 2 ) که در آن *S -* 1 معکوس ماتریس کوواریانس داده‌ها است و در تفاوت بین بردارهای میانگین پیش‌بینی‌کننده‌ها برای هر گروه ضرب می‌شود (یعنی X 1 حاوی میانگین هر پیش‌بینی است که از داده‌های کلاس 1 محاسبه می‌شود). در عمل، یک نمونه جدید، u، بر روی تابع تفکیک به‌عنوان u *' S -* 1 پیش‌بینی می‌شود. ( x 1 *-* x 2 ) که یک امتیاز متمایز را برمی گرداند. اگر نمونه به میانگین گروه 1 نزدیکتر از میانگین گروه 2 در پروژه باشد، یک نمونه جدید در گروه 1 طبقه‌بندی می‌شود:­

| *b (* u *—* x i ) | *—* | *b (* u *—* x 2 ) | *<* 0 *\_*  (12. 8)

به‌عنوان یک تصویر پیچیده تر، شکل.  [12. 7](#bookmark565) مجموعه داده‌ای را با دو کلاس و دو پیش‌بینی *A* و *B* نشان می‌دهد. خط *A* = *B* به راحتی این دو مجموعه نقطه را به گروه‌های مجزا جدا می‌کند. با این حال، این خط تابع تمایز ­*نیست.* در عوض، تابع تفکیک متعامد به خطی است که آنها را در فضا از هم جدا می‌کند (شکل 2 را ببینید).  [12. 8 )](#bookmark565) . با دو پیش‌بینی، تابع تفکیک برای یک نمونه ناشناخته *u* است

شکل 12. 7: یک مثال ساده از دو گروه نمونه که به وضوح قابل تفکیک ­هستند

10

5

0

-5

-50 510

پیش‌بینی A

شکل 12. 8: خط تقریباً *A* = *B* بردار است که به صورت بصری ­دو گروه را از هم جدا می‌کند. ارزیابی عضویت کلاس با طرح یک نمونه بر روی بردار متمایز ( *فلش قرمز* ) و سپس محاسبه فاصله آن از میانگین برای هر گروه تعیین می‌شود. سپس نمونه به گروهی که میانگین نزدیکتر است طبقه‌بندی می‌شود. نمودارهای *جعبه* توزیع نمونه‌ها برای هر کلاس پس از انجام LDA است که حداکثر کردن تغییرات بین گروهی را نشان می‌دهد.

-3

-2

-1

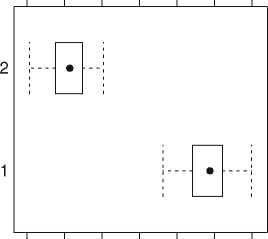
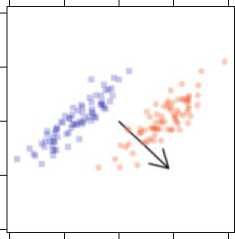
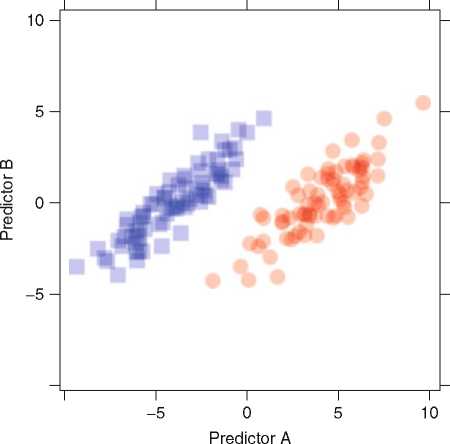
0

1

2

3

امتیاز تشخیصی



*D ( u* ) = u *' S* 1 ( x 1 *—* x s )

|  |  |
| --- | --- |
| *\_ f* ( *x* i *A - X s A* ) *S B \_*  *A  s* 2 *A s* 2 *B - s* 2 *AB*  /( *X* i *b — X s B* ) *Ss A*  + *u B*  2 2 2  *s* 2 *A s* 2 *B - s* 2 *AB* | ( *X* i *b — X s B* ) *S AB* A  *s* 2 *s* 2 *s* 2  *S A S B - S AB*  ( *X* 1 *A - x 2 A* ) *S AB*  *-*  22 2 *.*  *S A S B - S AB* |

در اینجا، *X 1 a* میانگین نمونه برای پیش‌بینی *A* است که تنها با استفاده از داده‌های کلاس اول محاسبه می‌شود. *X* 2 *a* میانگین نمونه برای *A* برای کلاس دوم است (نمونه مشابه برای پیش‌بینی *B* است). همچنین *s* 2 *A* واریانس نمونه برای پیش‌بینی *A (*محاسبه شده با داده‌های هر دو کلاس)، *s* 2 *B* واریانس نمونه برای پیش‌بینی *B* و *s AB* کوواریانس نمونه بین دو پیش‌بینی است.

برای این تابع، توجه داشته باشید که تمام واریانس‌های پیش‌بینی و کوواریانس بین پیش‌بینی در این معادله استفاده می‌شود. وقتی تعداد پیش‌بینی‌کننده‌ها زیاد باشد، معادله پیش‌بینی به تعداد بسیار زیادی پارامتر نیاز دارد تا تخمین زده شود. برای *P* = 2 و دو کلاس، این معادله از چهار میانگین و سه پارامتر واریانس استفاده می‌کند. به‌طور کلی، مدل به پارامترهای *CP* + *P (* *P* + 1) */ 2 با* پیش‌بینی‌کننده‌های *P و* کلاس‌های *C نیاز دارد.* در مقایسه با رگرسیون لجستیک، یک مدل مشابه فقط سه پارامتر را برآورد می‌کند. با افزایش تعداد پیش‌بینی‌ها، این تفاوت بین مدل‌ها بیشتر می‌شود. با این حال، مقدار پارامترهای اضافی در مدل‌های LDA این است که همبستگی‌های بین پیش‌بینی به صراحت توسط مدل مدیریت می‌شوند. در صورت وجود همبستگی‌های قابل‌توجه، این باید مزیتی برای LDA نسبت به رگرسیون لجستیک ایجاد کند، اگرچه هر دو مدل زمانی که چند خطی شدید شود، از بین می‌روند.

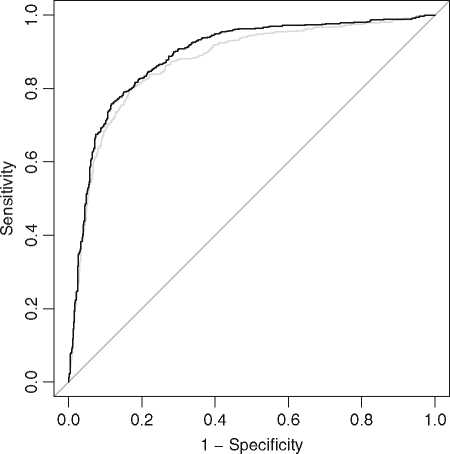
فرمول‌بندی فیشر از مسئله به معنای شهودی است، حل ریاضی آن آسان است و برخلاف رویکرد ولش، هیچ فرضی در مورد توزیع زیربنایی داده‌ها ندارد. بهینه‌سازی ریاضی ­حداکثر تعداد توابع متمایز قابل استخراج را محدود می‌کند تا کمتر از تعداد پیش‌بینی‌کننده‌ها یا یک کمتر از تعداد گروه‌ها باشد. برای مثال، اگر ده پیش‌بین و سه گروه داشته باشیم، حداکثر می‌توانیم دو بردار متمایز خطی استخراج کنیم. مشابه PCA، مقادیر ویژه در این مسئله نشان دهنده مقدار تغییرات توضیح داده شده توسط هر ­جزء W *-* 1 B است. از این رو، LDA عضوی از روتین‌های متغیر پنهان مانند PCA و حداقل مربعات جزئی (PLS) است. در عمل، تعداد ­بردارهای متمایز یک پارامتر تنظیمی است که با استفاده از رویکرد معمول اعتبارسنجی متقاطع یا نمونه‌گیری مجدد با معیارهای عملکرد مناسب، آن را تخمین می‌زنیم.

بررسی دقیق تابع تفکیک خطی منجر به دو ­یافته می‌شود که مشابه آن چیزی است که با رگرسیون خطی چندگانه مشاهده کردیم (بخش.  [6. 2 )](#bookmark286) . اول، راه حل LDA به معکوس کردن یک ماتریس کوواریانس بستگی دارد و یک راه حل منحصر به فرد تنها زمانی وجود دارد که این ماتریس معکوس باشد. درست مانند رگرسیون، این بدان معناست که داده‌ها باید شامل نمونه‌های بیشتری نسبت به ­پیش‌دیگرها باشند و پیش‌بینی‌کننده‌ها باید مستقل باشند (برای تعیین اینکه آیا ماتریس کوواریانس معکوس‌پذیر است، بخش محاسبات را ببینید). زمانی که تعداد نمونه‌ها بیشتر از پیش‌بینی‌کننده‌ها باشد، یا زمانی که پیش‌بینی‌کننده‌ها بسیار همبسته هستند، درست مانند رگرسیون، یک رویکرد رایج این است که ابتدا PCA را برای کاهش ابعاد و تولید ترکیب‌های پیش‌بینی نامرتبط جدید انجام دهیم. در حالی که نشان داده شده است که این رویکرد کار می‌کند، کاهش ابعاد در مورد ساختار کلاس داده‌ها بی اطلاع است. برای ادغام ساختار کلاس در کاهش ابعاد، استفاده از روش‌های PLSDA یا منظم‌سازی را توصیه می‌کنیم (به بخش‌های زیر مراجعه کنید). دوم، تابع متمایز خطی یک بردار *P-* بعدی است که مقادیر آن مستقیماً با پیش‌بینی‌کننده‌های اصلی جفت می‌شوند. بزرگی این مقادیر را می‌توان برای ­درک سهم هر پیش‌بینی در طبقه‌بندی نمونه‌ها و ارائه درک و تفسیری در مورد سامانه زیربنایی استفاده کرد.

از بحث بالا، متخصصان باید قبل از استفاده از LDA در پیش پردازش داده‌ها سختگیرانه عمل کنند. توصیه می‌کنیم پیش‌بینی‌کننده‌ها در مرکز و مقیاس‌بندی شوند و پیش‌بینی‌کننده‌های واریانس نزدیک به صفر حذف شوند. اگر ماتریس کوواریانس هنوز معکوس نیست، توصیه می‌کنیم از PLS یا یک رویکرد منظم‌سازی استفاده کنید. به‌طور مشابه، توصیه می‌کنیم در صورتی که پیش‌بینی‌کننده‌های بیشتری نسبت به نمونه‌ها وجود دارد، از روش‌های PLS یا منظم‌سازی (که در بخش‌های این فصل توضیح داده شده است) استفاده کنید. در همین راستا، متخصص هنگام استفاده از روال‌های اعتبارسنجی متقاطع برای روش‌هایی که به معکوس کردن ماتریس کوواریانس وابسته هستند، باید از تعداد نمونه‌ها نسبت به تعداد پیش‌بینی‌کننده‌ها آگاه باشد. به‌عنوان مثال، اگر تعداد نمونه‌ها 5 درصد بیشتر از تعداد پیش‌بینی‌کننده‌های مجموعه آموزشی باشد و اعتبارسنجی متقاطع 10 برابری را انتخاب کنیم، ماتریس کوواریانس برای هیچ یک از چین‌ها معکوس نخواهد بود زیرا همه تاها نمونه‌های کمتری نسبت به پیش‌بینی‌کننده‌ها خواهد داشت.

اکنون نحوه عملکرد LDA بر روی داده‌های گرنت را نشان خواهیم داد. از آنجایی که LDA به پیش‌بینی‌کننده‌های واریانس نزدیک به صفر و پیش‌بینی‌کننده‌های خطی حساس است، مجموعه پیش‌بینی را به ۲۵۳ پیش‌بینی کاهش داده‌ایم (شامل ترم مجذور روز مانند رگرسیون لجستیک). با استفاده از این زیرمجموعه پیش‌بینی‌ها، مساحت زیر منحنی ROC برای مجموعه نگهدارنده 2008 0. 89 است. جدول [12. 4](#bookmark567) ماتریس سردرگمی را برای این داده‌ها نشان می‌دهد و شکل 1.  [12. 9](#bookmark567) منحنی ROC مربوطه را نشان می‌دهد. خط خاکستری روشن در این نمودار نیز منحنی ROC را برای مدل رگرسیون لجستیک قبلی نشان می‌دهد.

همانطور که در بالا ذکر کردیم، بررسی ضرایب ­تابع غیر خطی تفکیک می‌تواند درکی از اهمیت نسبی پیش‌بینی‌کننده‌ها ارائه دهد. 5 پیش‌بینی برتر بر اساس بزرگی مطلق ­ضریب تابع متمایز عبارتند از: روز عددی سال (مربع) (2. 2)، روز عددی سال ( *-1. 9* )، تعداد گرنت ناموفق توسط محققین ارشد (-0 ). *62 )،* تعداد گرنت موفق توسط محققین ارشد (0. 58) و باند ارزش قرارداد A (0. 56). توجه داشته باشید که این فهرست حاوی چندین پیش‌بینی است که رویکرد تک متغیره با موفقیت ارسال گرنت مرتبط است. در اینجا، تعداد ارسال‌های گرنت ناموفق قبلی ­توسط بازرس ارشد با تعداد ارسال‌های موفق گرنت قبلی توسط محقق ارشد و بزرگترین طبقه‌بندی پولی که بصری است، رابطه معکوس دارد.



شکل 12. 9: منحنی ROC برای Holdout 2008 با استفاده از LDA. AUC 0. 89 است. خط *کمی سایه دار* منحنی ROC برای مدل رگرسیون لجستیک قبلی است

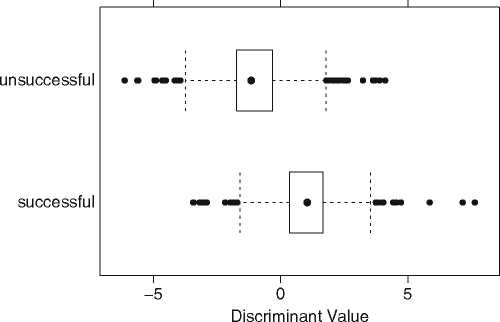
جدول 12. 4: ماتریس سردرگمی مجموعه Holdout 2008 برای مدل LDA

|  |  |  |
| --- | --- | --- |
|  | کلاس مشاهده شده | |
| موفقیت آمیز | ناموفق |
| موفقیت آمیز | 458 | 175 |
| ناموفق | 112 | 812 |

این مدل دارای دقت کلی 81. 6 درصد، حساسیت 80. 4 درصد و ویژگی 82. 3 درصد بود.

سپس می‌توانیم گرنت سال 2008 را بر روی این بردار تفکیک‌کننده خطی پیش‌بینی کنیم و توزیع امتیازهای متمایز را بررسی کنیم (شکل 2).  [12. 10 )](#bookmark568) . در حالی که توزیع‌ها برای برنامه‌های گرنت موفق و ناموفق همپوشانی دارند، LDA عملکرد طبقه‌بندی مناسبی را ارائه می‌دهد - به ویژه با توجه به اینکه LDA تمام روابط اساسی را در یک بعد خلاصه می‌کند (جدول [12. 4 )](#bookmark567) .

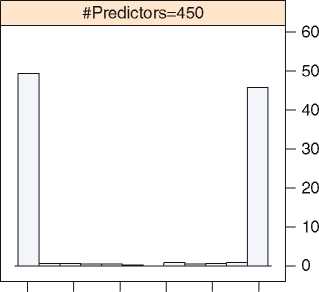
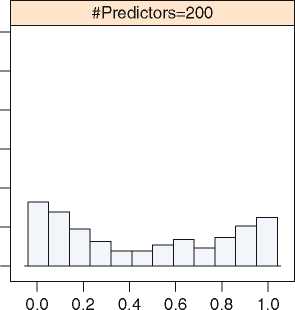
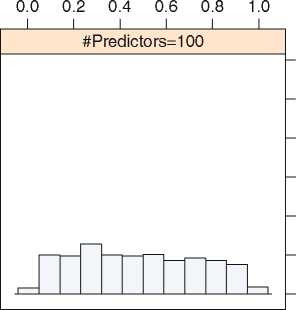
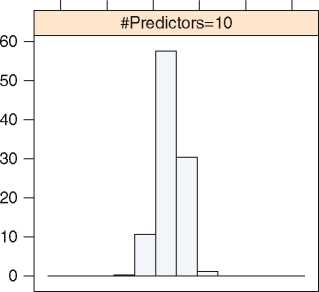
هنگامی که نمونه‌های بیشتری نسبت به پیش‌بینی‌کننده‌ها داشته باشیم، ماتریس کوواریانس در حالت ­عمودی است و داده‌ها را می‌توان به خوبی توسط یک ابر صفحه خطی از هم جدا کرد، آنگاه LDA یک مدل رضایت‌بخش پیش‌بینی تولید می‌کند که همچنین ­درک درستی از روابط اساسی بین پیش‌بینی‌کننده‌ها و پاسخ ارائه می‌دهد.



شکل 12. 10: نمودارهای جعبه‌ای از امتیازهای متمایز برای مجموعه نگهدارنده 2008

با این حال، کاربر باید آگاه باشد که سناریوی مجموعه داده‌ای وجود دارد که این الزامات اساسی را برآورده می‌کند، اما تخمین‌هایی از احتمال کلاس را ارائه می‌دهد که بیش از حد خوش بینانه هستند. به‌طور خاص، زمانی که تعداد نمونه‌ها به تعداد پیش‌بینی‌کننده‌ها نزدیک می‌شود، کاربر باید با احتمالات کلاس پیش‌بینی‌شده LDA بسیار محتاط باشد. ما از یک شبیه‌سازی ساده برای نشان دادن این یادداشت احتیاطی استفاده خواهیم کرد. برای 500 نمونه، مجموعه‌های داده‌ای حاوی 10، 100، 200 و 450 پیش‌بینی همگی از یک جمعیت عادی تصادفی تولید کرده‌ایم. پاسخ مجدد ­برای نمونه‌ها نیز به صورت تصادفی با قرار دادن 250 نمونه در هر دسته ایجاد شد. بنابراین، پیش‌بینی‌کننده‌ها و پاسخ برای هر یک از این مجموعه داده‌ها هیچ ارتباطی با یکدیگر ندارند. سپس مدل‌های LDA را بر روی هر یک از این مجموعه داده‌ها می‌سازیم و عملکرد را بررسی می‌کنیم. همانطور که انتظار داریم، دقت طبقه‌بندی مجموعه تست برای هر مجموعه داده تقریباً 50٪ است. از آنجایی که این داده‌ها کاملاً تصادفی هستند، ما همچنین انتظار داریم که ­احتمالات کلاس پیش‌بینی شده برای مجموعه آزمایشی نیز باید حدود 0. 5 باشد. این زمانی درست است که تعداد پیش‌بینی‌کننده‌ها نسبت به تعداد نمونه‌های نمونه کوچک ­باشد. اما با افزایش تعداد پیش‌بینی‌ها، احتمالات کلاس پیش‌بینی شده به 0 و 1 نزدیک تر می‌شوند (شکل 1).  [12. 11 )](#bookmark569) . در ارزش اسمی، این نتایج غیرمعمول به نظر ­می‌رسند: عملکرد مجموعه تست به ما می‌گوید که مدل به خوبی پرتاب سکه عمل می‌کند، اما مدل در مورد طبقه‌بندی نمونه‌ها در هر دسته بسیار مطمئن است.

چگونه می‌تواند این باشد؟ به نظر می‌رسد که این نتیجه گیری‌های به ظاهر متناقض ­به دلیل ساختار ریاضی LDA است. به یاد بیاورید که LDA یک بردار تفکیک بهینه را پیدا می‌کند. از نظر هندسی، اگر تعداد نمونه‌ها با تعداد پیش‌بینی‌کننده‌ها (یا ابعاد) برابر باشد، می‌توانیم حداقل یک بردار را پیدا کنیم ­که نمونه‌ها را کاملاً از هم جدا کند. ساده‌ترین حالت را در نظر بگیرید که دو نمونه و دو بعد داریم. تا زمانی که آن نمونه‌ها در یک مکان نباشند، می‌توانیم یک بردار (در واقع بی نهایت زیاد) پیدا کنیم.



مجموعه تست احتمالات برای کلاس #1

شکل 12. 11: هیستوگرام احتمالات کلاس مجموعه تست برای یک مثال دو کلاسه شبیه‌سازی شده که در آن همه پیش‌بینی‌کننده‌ها غیر اطلاعاتی هستند. با نزدیک شدن تعداد پیش‌بینی‌کننده‌ها به تعداد نمونه‌ها (500 در مجموعه آموزشی)، احتمالات کلاس شروع به واگرایی به سمت دو حد می‌کنند (اما، دقت کلی نزدیک به 50٪ باقی می‌ماند که دو نمونه را کاملاً از هم جدا می‌کند. سه نمونه (دو نمونه در یک کلاس و یکی در کلاس دیگر) در دو بعد نیز می‌توانند کاملاً از هم جدا شوند تا زمانی که نقاط روی یک خط مستقیم نباشند و تک نقطه بین دو نقطه دیگر از کلاس دیگر قرار نگیرد.

بدیهی است که این سناریوی داده می‌تواند منجر به تخمین‌های احتمال کلاسی شود که کالیبراسیون ضعیفی دارند (همانطور که در بخش بحث شد.  [11. 3 )](#bookmark17) . با توجه به این مسأله ذاتی با LDA و همچنین سایر الزامات اساسی آن، توصیه می‌کنیم که LDA در مجموعه داده‌هایی که حداقل 5 تا 10 برابر نمونه‌های بیشتری نسبت به پیش‌بینی‌کننده‌ها دارند، استفاده شود. هنگامی که نسبت به زیر 5 می‌رسد، باید در نتایج LDA احتیاط کرد.

در نهایت، مشابه رگرسیون لجستیک، هر پیش‌بینی‌ای که در یک مدل LDA استفاده می‌شود، می‌تواند تبدیل شود و در مدل گنجانده شود، مانند آنچه که با مقدار عددی مجذور روز سال در شکل مشاهده کردیم.  [12. 3](#bookmark558) . علاوه بر این، ­تمرین‌کننده می‌تواند عبارات متقابل (یعنی تعامل) را در میان پیش‌بینی‌کننده‌ها ایجاد کند. اتخاذ این رویکرد یکی از راه‌های فعال کردن LDA برای یافتن مرزهای متمایز غیرخطی است. با این حال، تبدیل متغیر یا تعاملات بین پیش‌بینی‌کننده‌ها فقط باید در نظر گرفته شوند، اگر دلیل خوبی برای این باور وجود داشته باشد که اطلاعات پیش‌بینی معنادار از طریق این پیش‌بینی‌کننده‌های اضافی وجود دارد. گنجاندن پیش‌بینی‌کننده‌های غیر اطلاعاتی اضافی، توانایی پیش‌بینی LDA را کاهش می‌دهد و می‌تواند از وارونگی ماتریس کوواریانس جلوگیری کند. اگر متخصص ­احتمال می‌دهد که ساختار غیرخطی بین پیش‌بینی‌کننده‌ها و نتیجه طبقه‌بندی وجود دارد، اما مطمئن نیست که کدام پیش‌بینی‌کننده‌ها در این رابطه دخیل هستند، توصیه می‌کنیم از روش‌های ارائه‌شده در فصل بعدی استفاده کنید.

تحلیل تفکیک حداقل مربعات جزئی

همانطور که بارها در طول فصل‌های قبلی اشاره کردیم، به صورت گذشته ­نگر یا آینده نگر، پیش‌بینی‌کننده‌های اندازه‌گیری شده برای هر مسأله خاص می‌توانند همبستگی زیادی داشته باشند یا می‌توانند از تعداد نمونه‌های جمع‌آوری شده بیشتر باشند. اگر هر یک از این شرایط درست باشد، نمی‌توان از روش معمول LDA به‌طور مستقیم برای یافتن تابع تفکیک کننده بهینه استفاده کرد.

درست مانند تنظیمات رگرسیون، می‌توانیم سعی کنیم داده‌های خود را به‌گونه‌ای پیش پردازش کنیم که پیش‌بینی‌کننده‌های همبسته بسیار را حذف کنیم. اگر ساختار همبستگی پیچیده‌تری ­در داده‌ها وجود داشته باشد یا اگر تعداد پیش‌بینی‌کننده‌ها همچنان از تعداد نمونه‌ها بیشتر باشد (یا نسبت نمونه‌ها به پیش‌بینی‌کننده‌ها خیلی کم است)، می‌توان از PCA برای کاهش بعد فضای پیش‌بینی استفاده کرد. با این حال، همانطور که ­قبلا در بخش بحث شد.  [6. 3 ،](#bookmark8) PCA ممکن است ترکیبات پیش‌بینی‌ای را ­که نمونه‌ها را به‌طور بهینه به گروه‌ها جدا می‌کنند، شناسایی نکند. به یاد بیاورید که هدف فیشر LDA یافتن فضای فرعی بود که تغییرات بین گروه را به حداکثر می‌رساند. از آنجایی که PCA هیچ یک از ­اطلاعات طبقه‌بندی پاسخ را در نظر نمی‌گیرد، انتظار نداریم که زیرفضای بهینه را برای طبقه‌بندی پیدا کند. به جای اتخاذ این رویکرد گام به گام (PCA- سپس-LDA) برای مسأله بیش از حد تعیین شده، توصیه می‌کنیم از PLS به منظور تبعیض استفاده کنید.

استفاده از PLS برای یک مسأله طبقه‌بندی حداقل به اواسط دهه 1980 برمی گردد [( برنتسون و ولد 1986](#bookmark1011) ). همانطور که در بخش رگرسیون ذکر شد، الگوریتم اصلی NIPALS در جامعه شیمی سنجی توسعه و اصلاح شد. جای تعجب نیست که این جامعه استفاده از PLS را به تنظیمات طبقه‌بندی کاوش و گسترش داد و این تکنیک را ­تحلیل متمایز PLS (یا PLSDA) نامید.  [دان و وولد](#bookmark1015) [( 1990](#bookmark1015) )، برای مثال، PLSDA را بر روی یک مثال تشخیص الگوی شیمی‌سنجی نشان داد و نشان داد که جداسازی بهتر نمونه‌ها در گروه‌ها را نسبت به روش سنتی PCA-سپس-LDA ارائه می‌دهد.

برای ایجاد شهود در مورد اینکه چرا PLS به‌طور طبیعی به ­تنظیمات طبقه‌بندی گسترش می‌یابد، اجازه دهید به‌طور خلاصه برای رگرسیون به PLS بازگردیم. به یاد بیاورید که PLS متغیرهای پنهانی را پیدا می‌کند که به‌طور همزمان ابعاد را کاهش می‌دهد و همبستگی را با مقدار پاسخ پیوسته به حداکثر می‌رساند (شکل 2 را ببینید). ­ [6. 9 )](#bookmark298) . در تنظیمات طبقه‌بندی برای یک مسئله دو گروهی، می‌توانیم ساده لوحانه از مقدار کلاس نمونه‌ها (که با 0 و 1 نشان داده می‌شود) به‌عنوان پاسخ برای این مدل استفاده کنیم. با توجه به آنچه در مورد PLS برای رگرسیون می‌دانیم، پس از آن انتظار داریم که متغیرهای پنهان برای کاهش ابعاد و در عین حال بهینه‌سازی همبستگی با بردار پاسخ طبقه‌ای انتخاب شوند. البته اگر هدف طبقه‌بندی باشد، بهینه‌سازی همبستگی هدف طبیعی نیست - در عوض، به حداقل رساندن ­خطای طبقه‌بندی اشتباه یا اهداف دیگری مرتبط با طبقه‌بندی رویکرد بهتری به نظر می‌رسد. علیرغم این واقعیت، PLSDA باید بهتر عمل کند زیرا اطلاعات گروه در حین تلاش برای کاهش ابعاد فضای پیش‌بینی در نظر گرفته می‌شود.

حتی اگر یک معیار همبستگی توسط PLS برای کاهش ابعاد با توجه به پاسخ استفاده می‌شود، معلوم می‌شود که این معیار اتفاقاً کار درستی را انجام می‌دهد. قبل از اینکه به دلیل درستی آن بپردازیم، ابتدا باید یک موضوع عملی را مورد بحث قرار دهیم: کدگذاری پاسخ. برای یک مسئله دو گروهی، کلاس‌ها در مجموعه‌ای از متغیرهای ساختگی 0/1 کدگذاری می‌شوند. با کلاس‌های *C،* نتایج مجموعه‌ای از متغیرهای ساختگی *C* است که در آن هر نمونه یک در ستون نشان‌دهنده کلاس مربوطه دارد. [[36]](#footnote-36) در نتیجه، پاسخ در داده‌ها با ماتریسی از متغیرهای ساختگی نشان داده می‌شود. به همین دلیل، مسأله را نمی‌توان با رویکرد رگرسیون PLS ­که در شکل 1 نشان داده شده است، حل کرد.  [6. 9](#bookmark298) و باید برای پاسخ چند متغیره به پارادایم حرکت کند.

استفاده از PLS در تنظیمات طبقه‌بندی با پاسخ چند متغیره، ارتباط ریاضی قوی با تحلیل همبستگی متعارف و LDA دارد [برای جزئیات فنی به Barker and Rayens ( 2003 ) مراجعه کنید]. با فرض ساختار کدگذاری بالا برای نتیجه، بارکر و راینز ( 2003 ) نشان دادند که جهت‌های PLS در این زمینه، بردارهای ویژه یک ماتریس کوواریانس بین گروهی (یعنی B از LDA) اندکی آشفته هستند. [[37]](#footnote-37) بنابراین، PLS به دنبال یافتن جداسازی بهینه گروه است در حالی که توسط اطلاعات بین گروه‌ها هدایت می‌شود. در مقابل، PCA به دنبال کاهش ابعاد با استفاده از تغییرات کل است که توسط ماتریس کوواریانس کلی پیش‌بینی‌کننده‌ها هدایت می‌شود.

این تحقیق منطق روشنی را برای انتخاب PLS نسبت به PCA در زمانی که کاهش ابعاد در هنگام تلاش برای طبقه‌بندی مورد نیاز است، ارائه می‌کند. با این حال، لیو

جدول 12. 5: ماتریس سردرگمی مجموعه Holdout 2008 برای مدل PLS

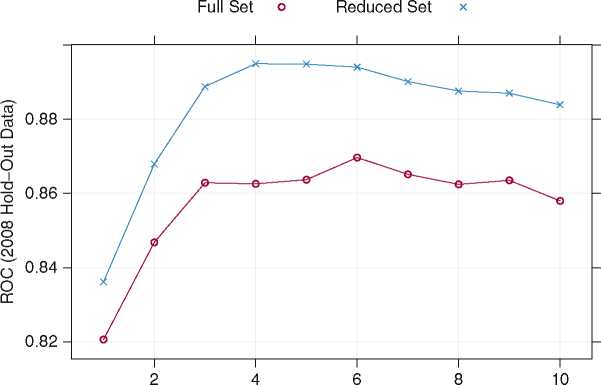
|  |  |  |
| --- | --- | --- |
|  | کلاس مشاهده شده | |
| موفقیت آمیز | ناموفق |
| موفقیت آمیز | 490 | 220 |
| ناموفق | 80 | 767 |

این مدل دارای دقت کلی 80. 7 درصد، حساسیت 86 درصد و ویژگی [77. 7 درصد بود. مجموعه](#bookmark1021) کاهش یافته پیش‌بینی‌کننده‌ها برای تولید این ماتریس استفاده شد

و Rayens [( 2007](#bookmark1021) ) اشاره می‌کنند که اگر کاهش ابعاد ضروری *نباشد* و طبقه‌بندی هدف باشد، LDA همیشه ­نرخ طبقه‌بندی نادرست کمتری نسبت به PLS ارائه می‌دهد. از این رو، LDA هنوز جای ضروری در جعبه ابزار طبقه‌بندی دارد.

دقیقاً مانند PLS برای رگرسیون، یک پارامتر تنظیم وجود دارد: تعداد ­متغیرهای پنهانی که باید حفظ شوند. هنگام انجام LDA بر روی داده‌های گرنت، مجموعه کاهش یافته پیش‌بینی‌کننده‌ها در صفحه شرح داده شده است [278](#bookmark20) استفاده شد (که پیش‌بینی‌کننده‌های واریانس نزدیک به صفر و پیش‌بینی‌کننده‌هایی که باعث همخطی شدید شدند) با مجذور عبارت اضافی برای روز سال استفاده شد. اما PLS می‌تواند در این شرایط مدلی تولید کند. عملکرد PLS، همانطور که به زودی خواهیم دید، زمانی که شامل پیش‌بینی‌کننده‌هایی می‌شود که حاوی ­اطلاعات پیش‌بینی کم یا بدون اطلاعات هستند، تحت‌تاثیر قرار می‌گیرد. اجرای PLS بر روی مجموعه کامل پیش‌بینی‌ها، یک مدل بهینه با مساحت زیر منحنی ROC 0. 87 بر اساس شش مؤلفه تولید می‌کند (شکل 2 را ببینید).  [12. 12 )](#bookmark25) ، حساسیت 83. 7٪ و ویژگی 77٪. این نتایج ROC اندکی بدتر از مدل LDA هستند، بنابراین آیا باید از این تعجب کنیم؟ پس از همه، ما پیش‌بینی‌کننده‌های *بیشتری را گنجانده ایم.* در واقع، از جمله پیش‌بینی‌کننده‌هایی که حاوی اطلاعات بسیار کم یا بدون اطلاعات در مورد پاسخ هستند، عملکرد یک مدل PLS را کاهش می‌دهند. برای جزئیات بیشتر در مورد این پدیده، بخش.  [19. 1](#bookmark867) .

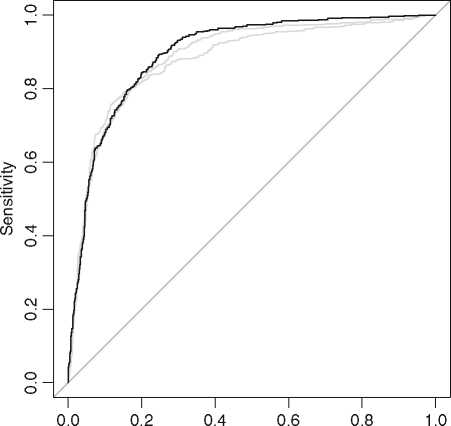
اگر PLS با استفاده از مجموعه‌ای بزرگتر از پیش‌بینی‌کننده‌ها بدتر از LDA عمل کند، گام منطقی بعدی بررسی عملکرد PLS با استفاده از مجموعه کاهش یافته پیش‌بینی‌کننده‌ها [(](#bookmark1021) که توسط LDA نیز استفاده می‌شود) است. [[38]](#footnote-38) ما از کار لیو و راینز [( 2007](#bookmark1021) ) می‌دانیم که LDA باید از نظر به حداقل رساندن خطاهای طبقه‌بندی نادرست از PLS بهتر عمل کند. برای این مسأله ما بهینه‌سازی ROC را انتخاب کرده ایم. با استفاده از این معیار آیا LDA همچنان از PLS بهتر عمل می‌کند؟ تعداد بهینه مولفه‌های PLS با مجموعه کاهش‌یافته پیش‌بینی‌کننده‌ها، 4 است، با ROC مربوطه 0. 89 (شکل 1).  [12. 12 )](#bookmark25) و ماتریس سردرگمی ارائه شده در جدول [12. 5](#bookmark24) . مجموعه کوچک‌تر پیش‌بینی‌کننده‌ها ROC را بهبود می‌بخشد، از اجزای کمتری برای رسیدن به آن مقدار استفاده می‌کند و به مقداری معادل عملکرد مدل LDA می‌رسد.



# اجزاء

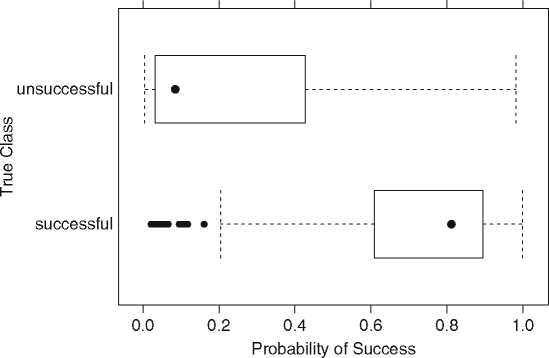
شکل 12. 12: مقادیر ROC بر حسب جزء برای PLS برای داده‌های اعطایی با استفاده از دو مجموعه پیش‌بینی. منحنی ROC دارای حداکثر مساحت با شش جزء در هنگام استفاده از همه پیش‌بینی‌کننده‌ها است. هنگام استفاده از زیرمجموعه کاهش یافته پیش‌بینی‌ها، منحنی ROC دارای حداکثر مساحت با چهار جزء است

به یاد بیاورید که PLSDA پاسخ را به‌عنوان مجموعه‌ای از متغیرهای ساختگی 0/1 رمزگذاری می‌کند. از آنجایی که PLS یک مدل خطی است، پیش‌بینی‌کننده‌های مدل PLSDA بین 0 و 1 محدود نمی‌شوند. کلاس نهایی توسط کلاسی با بزرگترین پیش‌بینی مدل تعیین می‌شود. با این حال، در صورت نیاز به احتمالات کلاس، پیش‌بینی‌کننده‌های مدل خام به پس پردازش نیاز دارند. رویکرد *softmax* قبلاً ­در بخش توضیح داده شده است. برای این منظور می‌توان از [11. 1 استفاده کرد.](#bookmark497) با این حال، تجربه ما با این تکنیک این است که احتمالات کلاس معنی‌داری را تولید نمی‌کند - برای مطمئن‌ترین پیش‌بینی‌کننده‌ها، احتمالات معمولاً نزدیک به 0 یا 1 نیستند. یک رویکرد جایگزین استفاده از *قانون بیز* برای ­تبدیل خروجی مدل اصلی به احتمالات کلاس است (شکل 2).  [12. 13 )](#bookmark573) . این تمایل به احتمالات کلاس معنادارتری دارد. یکی از مزایای استفاده از قانون بیز این است که می‌توان احتمال *قبلی را مشخص کرد.* این می‌تواند زمانی مهم باشد که داده‌ها شامل یک یا چند کلاس نادر باشد. در این شرایط، مجموعه آموزشی ممکن است به‌طور مصنوعی متعادل شود و از مشخصات احتمال قبلی برای تولید احتمالات دقیق تر استفاده شود. شکل [12. 14](#bookmark573) احتمالات کلاس را برای گرنت‌های سال 2008 نشان می‌دهد. [[39]](#footnote-39) در حالی که بین آنها همپوشانی وجود دارد



1 - خاص بودن

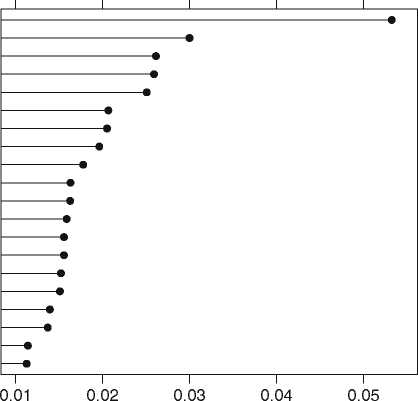
شکل 12. 13: منحنی ROC برای داده‌های نگهدارنده 2008 با استفاده از PLS ( *سیاه* ). AUC 0. 89 است. منحنی ROC برای LDA و رگرسیون لجستیک برای مقایسه روی هم قرار گرفته اند ( *خاکستری* ). هر سه روش به‌طور مشابه عمل می‌کنند



شکل 12. 14: نمودارهای جعبه‌ای برای احتمالات کلاس PLSDA مجموعه نگهدارنده 2008 محاسبه شده با استفاده از قانون بیز

ContractValueBandUnk SponsorUnk Jan Unsuccess. CI ContractValueBandA روز آگوست یکشنبه Success. CI Day2 ContractValueBandE GrantCat10A GrantCat30B ContractValueBandD ContractValueBandF ContractValueBandG

Sponsor24D Sponsor21A Sponsor2B NumSR



شکل 12. 15: امتیازهای اهمیت متغیر حداقل مربعات جزئی برای داده‌های کمکی کلاس‌ها، توزیع‌ها به درستی جابجا شده اند. احتمالات برای گرنت واقعا موفق بیشتر و برای گرنت ناموفق کم است.

Importance

همانطور که در تنظیمات رگرسیون، ما می‌توانیم اهمیت پیش‌بینی PLS را شناسایی کنیم (شکل 1).  [12. 15 )](#bookmark576) . برای داده‌های موجود، باند ارزش قرارداد ناشناخته در مقایسه با سایر پیش‌بینی‌کننده‌ها اهمیت نسبتاً زیادی دارد. مشابه LDA، موفقیت یا عدم موفقیت محقق ارشد به سمت بالای فهرست اهمیت شناور است. سایر پیش‌بینی‌کننده‌های مهم برای ­مدل طبقه‌بندی PLS، ارزش قرارداد، تعدادی دیگر از دسته‌های گرنت و ماه‌های ژانویه و آگوست است. جالب اینجاست که یکشنبه نیز در صدر فهرست قرار می‌گیرد.

در نهایت، اگر روابط غیرخطی بین پیش‌بینی‌کننده‌ها و پاسخ وجود داشته باشد و متخصص بخواهد از PLS برای یافتن این روابط استفاده کند، رویکردهای ارائه‌شده در بخش.  [6. 3](#bookmark8) را می‌توان به کار گرفت.

مدل‌های جریمه شده

مشابه روش‌های منظم‌سازی که در بخش بحث شده است.  [6. 4 ،](#bookmark311) بسیاری از مدل‌های طبقه‌بندی ­از جریمه‌ها (یا منظم‌سازی) برای بهبود برازش با داده‌ها، مانند ریج، استفاده می‌کنند. در بخش‌های بعدی، جریمه‌های مدل‌های غیرخطی ذاتی، مانند ماشین‌های بردار پشتیبان و شبکه‌های عصبی، مورد بحث قرار می‌گیرند.

به‌عنوان مثال، ممکن است یک عبارت جریمه برای مدل رگرسیون لجستیک به روشی که بسیار شبیه به رگرسیون پشته باشد، در نظر بگیرید. به یاد بیاورید که logis­

رگرسیون tic مقادیر پارامتری را پیدا می‌کند که تابع درستنمایی دوجمله ای، *L (* *p* ) را به حداکثر می‌رساند (به معادله [12. 1 مراجعه کنید )](#bookmark21) . یک رویکرد ساده برای منظم کردن این مدل، افزودن یک تابع جریمه مجذور به احتمال گزارش و یافتن تخمین‌های پارامتری است که حداکثر می‌کنند.

پ

log *L (* *P* ) *- A* ^ *p j.*

j = 1

[آیلرز و همکاران ( 2001](#bookmark1015) ) و [پارک و هیستی](#bookmark1023) [( 2008](#bookmark1023) ) این مدل را در ­متن متنی داده‌ها مورد بحث قرار می‌دهد که در آن تعداد زیادی پیش‌بینی و یک مجموعه آموزشی کوچک نمونه وجود دارد. در این شرایط، عبارت جریمه می‌تواند ضرایب مدل رگرسیون [[40]](#footnote-40)لجستیک را تثبیت کند. همانند رگرسیون برآمدگی، افزودن یک جریمه نیز می‌تواند یک اقدام متقابل در برابر پیش‌بینی‌کننده‌های بسیار همبسته ارائه کند.

به یاد بیاورید که روش دیگر برای منظم کردن مدل‌های رگرسیون خطی، اضافه کردن جریمه بر اساس مقادیر مطلق ضرایب رگرسیون است (شبیه ­به مدل ریج Sect.  [6. 4 )](#bookmark311) . مدل glmnet ( فریدمن [و همکاران 2010](#bookmark1016) ) از یک جریمه ریج مانند بر روی تابع درستنمایی دو جمله‌ای (یا چند جمله ای) استفاده می‌کند. مانند ریج، این منجر به ضرایب رگرسیون با مقادیر 0 مطلق می‌شود، بنابراین به‌طور همزمان منظم‌سازی و انتخاب ویژگی را به‌طور همزمان انجام می‌دهد. مدل‌های glmnet از پنالتی‌های ریج و ریج به‌طور همزمان استفاده می‌کنند، مانند شبکه الاستیک، اما ساختار پنالتی را کمی متفاوت می‌کند:

1 P P

(1 *- a )22 ZP* + *<&j I*

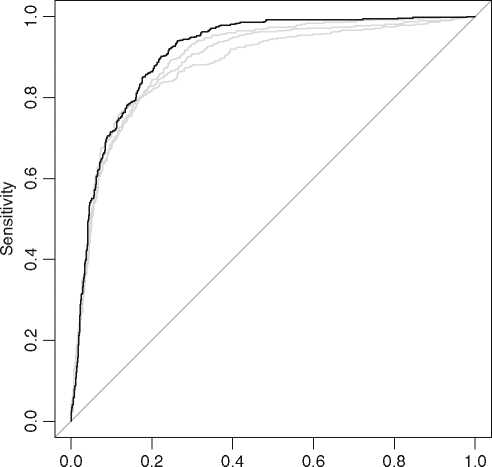
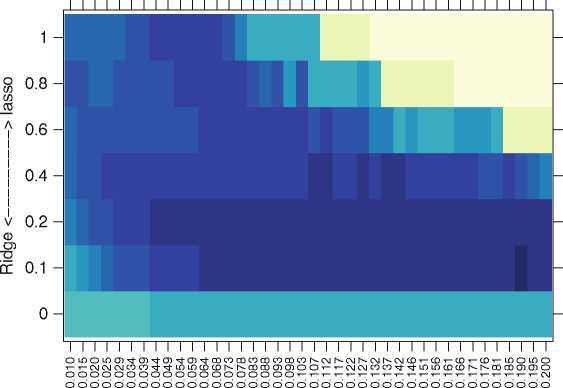
log *L*(*p*) *- A*

j = 1 j = 1

در اینجا، مقدار *a* «نسبت اختلاط» است که بین جریمه ریج خالص (وقتی *a* = 1) و جریمه رگرسیون مانند رج خالص ( *a* = 0) جابجا می‌شود. پارامتر تنظیم دیگر *A* کل مقدار جریمه را کنترل می‌کند.

برای داده‌های گرنت، مدل glmnet بر روی هفت مقدار پارامتر اختلاط *a* و 40 مقدار از مقدار کلی جریمه تنظیم شد. مجموعه کامل پیش‌بینی‌کننده‌ها در مدل استفاده شد. شکل 12. 16 یک نقشه حرارتی از ناحیه زیر منحنی ROC را برای این مدل‌ها نشان می‌دهد [.](#bookmark577) داده‌ها به نفع مدل‌هایی هستند که ترکیب بزرگ‌تری از جریمه‌های رج نسبت به پنالتی ریج دارند، اگرچه انتخاب‌های زیادی در این شبکه وجود دارد که قابل مقایسه هستند. ردیف پایین نقشه حرارتی نشان می‌دهد که یک راه حل کامل پشته بدون توجه به بزرگی جریمه ضعیف است. در پایان، تنظیمات عددی بهینه یک درصد اختلاط 0. 1 و مقدار 0. 19 برای مقدار تنظیم است. این تنظیمات تأثیر استفاده از تنها 44 پیش‌بینی از 1070 در مدل نهایی glmnet را داشت که به یک ناحیه زیر منحنی ROC دست یافت.

ناحیه زیر منحنی ROC



Amount of Regularization

**[**0.90

0.88

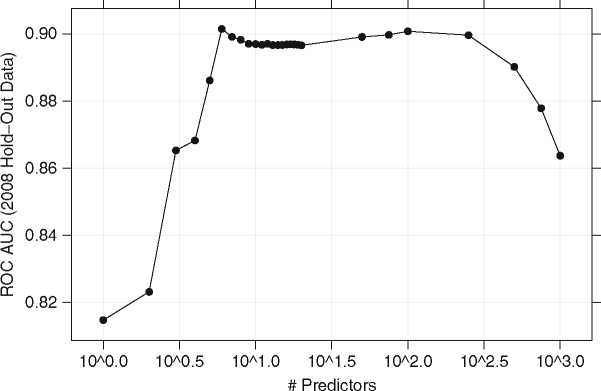
-0.86

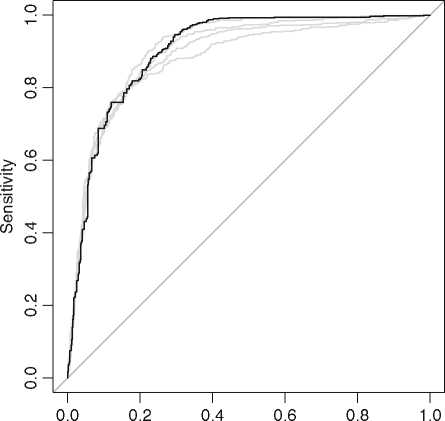
-0.84

-0.82

1 - Specificity

Fig. 12.16: *Top*: A heat map of the area under the ROC curve for the two glmnet tuning parameters. The numerically optimal settings are a mixing percentage of 0.1 and a value of 0.19 for the regularization amount. *Bottom*: The ROC curve for 2008 holdout data using glmnet the model (area under the ROC curve: 0.91)





1 - خاص بودن

شکل 12. 17: *بالا* : مشخصات پارامتر تنظیم برای مدل پراکنده LDA *پایین* : منحنی ROC برای مدل (AUC = 0. 901)

0. 91. مدل رگرسیون لجستیک قبلی که از مجموعه کاهش‌یافته پیش‌بینی‌کننده‌ها استفاده می‌کرد، منجر به AUC 0. 87 شد که نشان می‌دهد حذف روشی پیش‌بینی‌کننده‌های غیر اطلاعاتی، اثربخشی مدل را افزایش می‌دهد. سایر رویکردها برای انتخاب ویژگی *نظارت* شده در فصل بحث شده است.  [19](#bookmark864) .

روش دیگر، استراتژی‌های مجازات را می‌توان برای مدل‌های LDA اعمال کرد. مثلا، [کلمنسن و همکاران ( 2011](#bookmark1013) ) از این تکنیک با استفاده از مدل‌های LDA استفاده می‌کنند

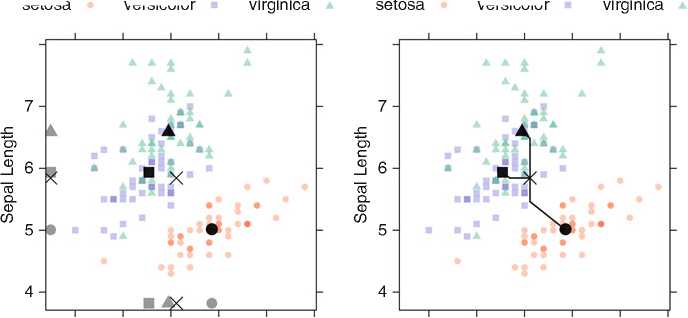
چارچوب *تحلیل تفکیک انعطاف‌پذیر (FDA)* شرح داده شده در بخش ها.  [13. 1](#bookmark612) و [13. 3](#bookmark629) . در این مدل از یک استراتژی شبکه الاستیک استفاده شده است. [جریمه‌های](#bookmark1027) *L* 1 اثر حذف پیش‌بینی‌کننده‌ها دارند در حالی که [جریمه L2](#bookmark1028) [ضرایب](#bookmark1027) *توابع* متمایز را به [سمت](#bookmark1028) [0](#bookmark1027) کوچک می‌کند. که حاوی ارجاعاتی به بسیاری از آثار قبلی است. همان مجازات ریج نیز برای مدل‌های متمایز کننده PLS اعمال شده است، به‌طوری که برخی از بارگذاری‌های PLS نیز حذف می‌شوند [( چانگ و کلس 2010](#bookmark1013) ).

مدل LDA جریمه شده از [کلمنسن و همکاران ( 2011](#bookmark1013) ) برای داده‌های گرنت اعمال شد. نرم‌افزار این مدل به کاربر اجازه می‌دهد تا تعداد پیش‌بینی‌کننده‌های حفظ شده را به‌عنوان پارامتر تنظیم (به جای مقدار جریمه *L* 1 ) مشخص کند. مدل بر روی این پارامتر و همچنین مقدار *L* 2 تنظیم شد پنالتی شکل [12. 17](#bookmark579) نتایج را برای یک مقدار از جریمه پشته نشان می‌دهد (تفاوت بسیار کمی در عملکرد در طیف وسیعی از مقادیر برای این پنالتی وجود دارد). زمانی که تعداد پیش‌بینی‌کننده‌ها نزدیک به حداکثر باشد، عملکرد متوسطی وجود دارد. با افزایش جریمه و حذف عوامل پیش‌بینی، عملکرد بهبود می‌یابد و نسبتاً ثابت می‌ماند تا زمانی که عوامل مهم حذف شوند. در این مرحله، عملکرد به‌طور چشمگیری سقوط می‌کند. در نتیجه فرآیند تنظیم، از شش پیش‌بینی در مدل استفاده شد که قابل رقابت با مدل‌های دیگر است (AUC 0. 9).

نزدیکترین مرکزهای کوچک شده

مدل مرکزی کوچک‌تر (همچنین به‌عنوان PAM، برای تحلیل پیش‌بینی ریزآرایه‌ها نیز شناخته می‌شود) یک مدل طبقه‌بندی خطی است که برای مسائل با ابعاد بالا به خوبی مناسب است [( تیبشیرانی و همکاران 2002](#bookmark1026) ، [2003](#bookmark1026) ; [گوو و همکاران 2007](#bookmark1017) ). برای هر کلاس، مرکز داده‌ها با در نظر گرفتن مقدار متوسط هر پیش‌بینی (در هر کلاس) در مجموعه آموزشی پیدا می‌شود. مرکز کلی با استفاده از داده‌های همه کلاس‌ها محاسبه می‌شود.

اگر یک پیش‌بینی حاوی اطلاعات زیادی برای یک کلاس خاص نباشد، مرکز آن برای آن کلاس احتمالاً نزدیک به مرکز کلی است. سه داده کلاس نشان داده شده در پانل سمت چپ شکل را در نظر بگیرید.  [12. 18](#bookmark582) . این مجموعه داده، داده معروف عنبیه فیشر/اندرسون است که در آن از چهار اندازه‌گیری کاسبرگ و گلبرگ زنبق برای طبقه‌بندی گل‌ها به یکی از سه گونه مختلف زنبق استفاده می‌شود: setosa، versicolor و virginica. در این نمودار، داده‌های کلاس‌های versicolor و virginica همپوشانی دارند، اما به خوبی از عنبیه‌های setosa جدا می‌شوند. مرکز برای بعد عرض کاسبرگ به صورت نمادهای خاکستری بالای محور *x نشان داده شده است.* مرکز ویرجینیکا (برای عرض کاسبرگ) بسیار نزدیک به مرکز کلی است و مرکز ورسیکالر کمی به مرکز کلی نزدیکتر از گلهای ستوزا است. این نشان می‌دهد که پیش‌بینی عرض کاسبرگ برای متمایز کردن گونه‌های ستوزا از دو گونه دیگر آموزنده‌ترین است. برای طول کاسبرگ (نشان داده شده در مجاورت محور *y* )، مرکز ورسیکالر بسیار نزدیک به مرکز داده‌ها است و دو گونه دیگر در منتهی الیه قرار دارند.



versicolor

versicolor

setosa

2.0 2.5 3.0 3.5 4.0

2.0 2.5 3.0 3.5 4.0

Sepal Width

Sepal Width

Fig. 12.18: *Left*: An example with three classes, their class-specific centroids (in *black*), and the overall centroid (*x*). The *grey symbols* along the axes are the centroids pro jected down into a single dimension. *Right*: The paths of the class centroids when shrunken to the center of the distributions

یکی از روش‌های طبقه‌بندی نمونه‌های ناشناخته، یافتن نزدیک‌ترین مرکز کلاس در فضای تمام‌بعدی و انتخاب آن کلاس برای پیش‌بینی است (یعنی مدل «نزدیک‌ترین مرکز»). به نظر می‌رسد که این رویکرد منجر به ایجاد مرزهای کلاس خطی می‌شود.

رویکرد اتخاذ شده توسط [تبشیرانی و همکاران. ( 2002](#bookmark1026) ) برای کوچک کردن کلاس ­ترویدهای مرکزی به مرکز کلی است. در انجام این کار، مرکزهایی که نزدیک‌تر به مرکز کلی شروع می‌شوند، قبل از سایرین به آن مکان می‌روند. به‌عنوان ­مثال، در بعد عرض کاسبرگ، مرکز ویرجینیکا قبل از دو مورد دیگر به مرکز خواهد رسید. برای این مدل، هنگامی که مرکز کلاس با مرکز کلی ملاقات کرد، دیگر بر طبقه‌بندی نمونه‌ها برای آن کلاس تأثیر نمی‌گذارد. باز هم، برای عرض کاسبرگ، هنگامی که مرکز ویرجینیکا به مرکز رسید، عرض کاسبرگ را فقط می‌توان برای طبقه‌بندی گل‌هایی که versicolor یا setosa هستند استفاده کرد. با انقباض کافی، امکان کوچک شدن تمام طبقات به مرکز وجود دارد. در صورتی که پیش‌بینی به مرکز مرکزی برسد، تاثیری بر مدل ندارد. در نتیجه، نزدیکترین مدل مرکزی کوچک شده نیز انتخاب ویژگی را در طول فرآیند آموزش مدل انجام می‌دهد.

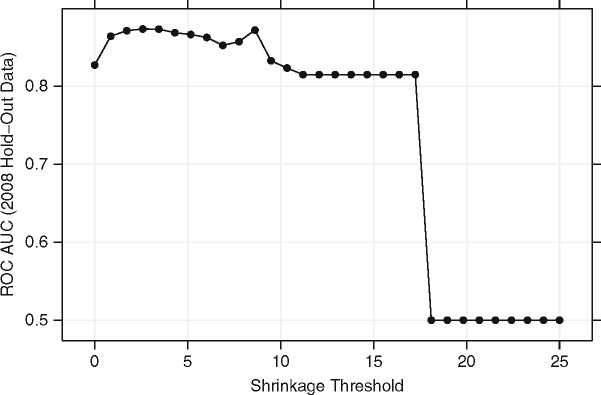
نزدیکترین روش سانتروئید کوچک شده دارای یک پارامتر تنظیم است: ­سن کوچک شدن. پانل سمت راست در شکل [12. 18](#bookmark582) مسیر مرکزها را بر روی ­مقادیر مختلف انقباض نشان می‌دهد. توجه داشته باشید که هر پیش‌بینی به صورت مورب به سمت مرکز حرکت می‌کند تا زمانی که یکی از مرکزهای کلاس خاص به مرکز برسد. در این مرحله کلاس‌ها به صورت تک بعدی به سمت مرکز حرکت می‌کنند. مرکز و مقیاس‌بندی پیش‌بینی‌کننده‌ها برای این مدل توصیه می‌شود.

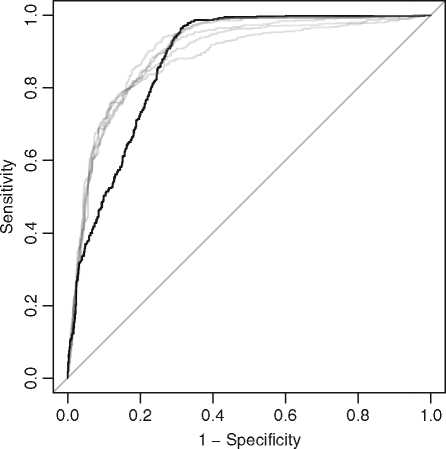
این مدل برای مسائل با تعداد زیادی پیش‌بینی به خوبی کار می‌کند، زیرا دارای انتخاب ویژگی داخلی است که توسط ­پارامتر تنظیم انقباض کنترل می‌شود. نزدیک‌ترین مرکز کوچک‌شده در ابتدا برای داده‌های پروفایل RNA که در آن تعداد پیش‌بینی‌کننده‌ها زیاد است (در چندین ­هزار ماسه) و تعداد نمونه‌ها کم است، توسعه داده شدند. اکثر مجموعه داده‌های پروفایل RNA کمتر از یک یا دویست نمونه دارند. در این سناریوی *n پایین و P زیاد،* داده‌ها احتمالاً نمی‌توانند یک مدل بسیار غیرخطی را پشتیبانی کنند ­و مرزهای طبقه‌بندی خطی انتخاب خوبی هستند. همچنین می‌توان از احتمالات کلاس قبلی به همراه فواصل بین مرکز کلاس و مرکز کلی برای تولید احتمالات کلاس استفاده کرد. امتیازهای اهمیت متغیر با استفاده از تفاوت بین مرکز کلاس به مرکز کلی محاسبه می‌شود (مقادیر مطلق بزرگتر نشان دهنده اهمیت مدل بالاتر است).

برای داده‌های گرنت، مجموعه کامل 1070 پیش‌بینی مورد ارزیابی قرار گرفت. این مدل بیش از 30 مقدار از پارامتر انقباض را تنظیم کرد که از 0 (به معنای انقباض بسیار کم و انتخاب ویژگی) تا 25 بود (شکل 1).  [12. 19 )](#bookmark27) . با مقادیر زیاد انقباض، تقریباً هیچ پیش‌بینی‌ای حفظ نمی‌شود و ناحیه زیر منحنی ROC بسیار ضعیف است. هنگامی که آستانه به ­تقریباً 17 کاهش می‌یابد، پنج پیش‌بینی اضافه شده است: تعداد گرنت ناموفق توسط محققین ارشد، حامی مالی ناشناخته، باند ارزش قرارداد A، باند ارزش قرارداد نامشخص ­و ماه ارسال ژانویه. افزودن این پیش‌بینی‌کننده‌ها تأثیر زیادی بر برازش مدل دارد. در راس منحنی (مقدار انقباض 2. 59)، متغیرهای مهم شروع به حذف می‌کنند و با افزایش مقدار انقباض، عدم برازش شروع می‌شود. اوج تیز در آستانه 8. 6 کنجکاو است. این افزایش با حذف دو پیش‌بینی مرتبط است: کد حامی 2B و باند ارزش قرارداد F. با این حال، مقدار انقباض بعدی سه پیش‌بینی اضافی ( ­باند ارزش قرارداد D، باند ارزش قرارداد E و باند ارزش قرارداد G) را حذف می‌کند. منجر به افت محسوس در ناحیه زیر منحنی ROC می‌شود. این جهش ناگهانی ­در عملکرد احتمالاً به دلیل این واقعیت است که فقط از یک نگهدارنده برای سنجش عملکرد استفاده می‌شود. رابطه واقعی بین عملکرد و انقباض به احتمال زیاد صاف تر از آنچه در این شکل نشان داده شده ­است خواهد بود. در بهترین آستانه، سطح زیر منحنی ROC با استفاده از 36 پیش‌بینی 0. 87 بود. حساسیت 83. 7٪ و ویژگی 77٪ برای داده‌های نگهداری سال 2008 بود. منحنی ROC نیز در شکل نشان داده شده است.  [12. 19](#bookmark27) .

12. 7 محاسبات

این بخش بسته‌های R زیر را مورد بحث قرار می‌دهد: AppliedPredictiveModeling، caret، glmnet، MASS، pamr، pls، pROC، rms، sparseLDA و انتخاب فرعی.





شکل 12. 19: *بالا* : مشخصات پارامتر تنظیم برای نزدیکترین ­مدل سنت تروئید کوچک شده. *پایین* : منحنی ROC برای مدل (AUC = 0. 873)

داده‌های درخواست گرنت را می‌توان در وب سایت Kaggle پیدا کرد. [[41]](#footnote-41) بسته R AppliedPredictiveModeling حاوی اسکریپت‌هایی است که می‌توانند برای ­بازتولید اشیاء و تحلیل‌های ارائه شده در اینجا استفاده شوند.

به دنبال رویکرد تقسیم داده‌ها که در بخش اول توضیح داده شد، دو چارچوب داده با داده‌های گرنت وجود دارد: آموزش شامل داده‌های قبل از 2008 و مجموعه نگه‌داری 2008 است که برای تنظیم مدل مورد استفاده قرار می‌گیرد، در حالی که آزمایش چارچوب داده تنها داده‌های کمکی سال 2008 دارد و نیست. تا فصل بعد استفاده شد بردار به نام pre2008 دارای شاخص‌های ردیف 6633 گرنت مجموعه آموزشی قبل از سال 2008 است (جدول را ببینید [12. 2](#bookmark557) برای خلاصه‌ای از استراتژی تقسیم داده ها).

بیشتر پیش‌بینی‌کننده‌های این مجموعه باینری هستند. به‌عنوان مثال، کدهای RFCD، کدهای SEO، حامیان مالی و دسته بندی‌های باند ارزش قرارداد ­در مقادیر باینری جداگانه با پیشوندهای شناسایی، مانند RFCD یا ContractValueBand وجود دارند. [[42]](#footnote-42) وقتی مقدار ناشناخته بود، یک متغیر ساختگی خاص ­برای این موقعیت ایجاد می‌شود، مانند SponsorUnk. متغیرهای ساختگی باینری برای ماه و روز هفته ارسال نیز وجود دارد.

علاوه بر این، تعداد و پیش‌بینی‌کننده‌های پیوسته مانند ­فراوانی‌های هر نقش مرتبط با گرنت وجود دارد. به‌عنوان مثال، NumCI و NumEA به ترتیب تعداد محققین ارشد و مشاوران خارجی در مورد گرنت هستند. تعداد افراد با نقش‌های نامشخص در پیش‌بینی numUnk ثبت می‌شود. پیش‌بینی‌کننده‌های شمارش مشابه نیز در داده‌ها وجود دارد، مانند تعداد افرادی که در یک چارچوب زمانی متولد شده‌اند (به‌عنوان مثال، CI. 1925 برای محققان ارشد ­متولد بین سال‌های 1925 و 1930)، تعداد متولد شده در یک منطقه خاص (مانند HV. استرالیا). و وضعیت مدرک آنها (به‌عنوان مثال، ECI. PhD ). تعداد ­گرنت موفق و ناموفق قبلی با پیش‌بینی‌کننده‌های Unsuccess. PS یا Success. CI برشمرده می‌شود. اطلاعات نشریه به دو صورت نمایش داده می‌شود. اول، مجموع هر نقش، مانند B. CI یا Astar. CI و همچنین تعداد کل در همه افراد ( AstarTotal ) یا همه انواع مجلات ( allPub ) در دسترس است.

روز تقویم سال به‌عنوان یک متغیر عددی ذخیره می‌شود.

در نهایت، نتیجه کلاس در ستونی به نام کلاس با سطوح موفق و ناموفق قرار می‌گیرد.

همانطور که در بخش‌های رگرسیون این کتاب نشان داده شد، مدل‌های مختلف محدودیت‌های متفاوتی بر روی انواع داده‌های قابل استفاده دارند. همانطور که ­قبلاً بحث شد، دو گروه‌بندی کلی از پیش‌بینی‌کننده‌ها ایجاد شد: مجموعه‌ای از پیش‌بینی‌کننده‌ها که شامل مجموعه کامل متغیرهای ساختگی باینری و داده‌های شمارش است و مجموعه کاهش‌یافته که برای پیش‌بینی‌کننده‌های واریانس نزدیک به صفر و پیش‌بینی‌کننده‌های بسیار همبسته فیلتر شد. برای مثال، ستون‌های AstarTotal، ATotal، BTotal و CTotal همگی به ستون allPub اضافه می‌شوند. در مجموعه کاهش یافته، allPub حذف شد. به‌طور مشابه، یک متغیر ساختگی برای ماه و یکی برای یک روز هفته نیز باید از مجموعه کاهش یافته حذف شود. دو ستون با کمترین فرکانس، Mar و Sun، از مجموعه کاهش یافته حذف شدند.

دو بردار کاراکتر به منظور مشخص کردن هر گروه ایجاد شد: fullSet و smallSet :

*> طول (fullSet)*

[1] 1070

*> هد (فول مجموعه)*

[1] "NumCI" "NumDR" "NumEA" "NumECI" "NumHV" "NumPS"

*> طول (کاهش مجموعه)*

[1] 252

*> head (reducedSet)*

[1] "NumCI" "NumDR" "NumECI" "NumPS" "NumSR" "NumSCI"

چگونه می‌توان مسائل هم خطی شدید (مانند ترکیبات خطی) را ­تشخیص داد؟ تابع trim. matrix در انتخاب فرعی یک ماتریس مربع و متقارن (مانند ماتریس کوواریانس) می‌گیرد و از یک الگوریتم برای حذف ترکیبات خطی استفاده می‌کند. به‌عنوان مثال، مجموعه کاهش یافته چنین مسائلی ندارد:

*> smallCovMat <- cov(training[, smallSet])*

*> کتابخانه (گزینه فرعی)*

*trimmingResults <- trim. matrix(reducedCovMat)*

*نام‌ها (نتایج برش)*

[1] "trimmedmat" "numbers. discarded" "names. discarded"

[4] "اندازه"

*## ببینید آیا هیچ یک از پیش‌بینی‌کننده‌ها حذف شده اند:*

*trimmingResults$names. رد شد*

کاراکتر (0)

با این حال، وقتی یک تابع یکسان را برای مجموعه کامل اعمال می‌کنیم، چندین پیش‌بینی شناسایی می‌شود:

*fullCovMat <- cov(training[, fullSet])*

*fullSetResults <- trim. matrix(fullCovMat)*

*## یک انتخاب متفاوت برای روز حذف بود*

*## توسط این تابع ساخته شده است*

*> fullSetResults$names. discarded*

[1] "NumDR" "PS. 1955" "CI. Dept1798" "PS. Dept3268" "PS. Faculty1"

[6] "DurationUnk" "ATotal" "Nov" "Sun"

تابع دیگری در بسته caret به نام findLinearCombos از روش مشابهی پیروی می‌کند اما به ماتریس مربع نیاز ندارد.

هنگام توسعه مدل‌ها، آموزش برای تنظیم پارامترها بر اساس منحنی ROC استفاده می‌شود. برای انجام این کار، یک تابع کنترل برای به دست آوردن نتایج مورد علاقه مورد نیاز است. برای این منظور از تابع caret trainControl استفاده می‌شود. ابتدا برای محاسبه مساحت زیر منحنی ROC، احتمالات کلاس باید تولید شود. به‌طور پیش‌فرض، train فقط پیش‌بینی‌کننده‌های کلاس را تولید می‌کند. گزینه classProbs را می‌توان در صورت نیاز به احتمالات مشخص کرد. همچنین به‌طور پیش فرض برای ارزیابی مدل از دقت کلی و آماره کاپا استفاده می‌شود. caret شامل یک تابع داخلی به نام twoClassSummary است که مساحت زیر منحنی ROC، حساسیت و ویژگی را محاسبه می‌کند. برای دستیابی به این اهداف، نحو به صورت زیر خواهد بود:

*ctrl <- trainControl(summaryFunction = twoClassSummary، + classProbs = TRUE)*

با این حال، در ابتدای فصل، یک طرح تقسیم داده ایجاد شد که مدل را بر اساس داده‌های قبل از 2008 ساخته و سپس از داده‌های نگهدارنده سال 2008 (در مجموعه آموزشی) برای تنظیم مدل استفاده کرد. برای انجام این کار، Train باید دقیقا بداند که از کدام نمونه‌ها هنگام تخمین پارامترها استفاده کند. آرگومان شاخص TrainControl ­این نمونه‌ها را شناسایی می‌کند. برای هر روش نمونه‌گیری مجدد، مجموعه‌ای از نمونه‌های نگهدارنده را می‌توان دقیقاً مشخص کرد. به‌عنوان مثال، با اعتبارسنجی متقاطع 10 برابری، نمونه‌های دقیقی که برای هر یک از 10 برابر حذف می‌شوند، با این گزینه مشخص می‌شوند. در این مورد، ایندکس ردیف‌هایی را مشخص می‌کند که با داده‌های قبل از سال 2008 مطابقت دارند. نحو دقیق باید این اعداد ردیف را در یک لیست بسته‌بندی کند (در صورتی که بیش از یک نگهدارنده وجود داشته باشد). به یاد بیاورید که بردار pre2008 شامل مکان‌های گرنت ارسال شده قبل از سال 2008 است. فراخوان TrainControl این است:

*ctrl <- trainControl( متد = "LGOCV"، summaryFunction = twoClassSummary، classProbs = TRUE، index = list(TrainSet = pre2008) )*

توجه داشته باشید که وقتی پارامترهای تنظیم با استفاده از برآوردهای عملکرد سال 2008 انتخاب شدند، مدل نهایی با تمام گرنت مجموعه آموزشی، از جمله موارد مربوط به سال 2008، مطابقت دارد.

در نهایت، برای اهداف توضیحی، ما باید پیش‌بینی‌کننده‌های گرنت سال 2008 را بر اساس مدل قبل از 2008 ذخیره کنیم (یعنی قبل از اینکه مدل نهایی دوباره با تمام داده‌های آموزشی تطبیق داده شود). آرگومان savePredictions این هدف را انجام می‌دهد:

*ctrl <- trainControl(روش = "LGOCV"،*

*summaryFunction = twoClassSummary، classProbs = TRUE،*

*index = list (TrainSet = pre2008)، savePredictions = TRUE )*

از آنجایی که بسیاری از مدل‌های توصیف‌شده در این متن از اعداد تصادفی استفاده می‌کنند، پیش از اجرای هر مدل، پایه تولیدکننده اعداد تصادفی تنظیم می‌شود تا بتوان محاسبات را دوباره تولید کرد. مقدار دانه 476 به‌طور تصادفی برای این فصل انتخاب شد.

رگرسیون لجستیک

تابع glm (برای GLMها) در پایه R معمولاً برای مطابقت با ­مدل‌های رگرسیون لجستیک استفاده می‌شود. نحو مشابه توابع مدل‌سازی قبلی است که از روش فرمول کار می‌کنند. به‌عنوان مثال، برای برازش با مدل نشان داده شده در پانل سمت چپ شکل.  [12. 3](#bookmark558) برای داده‌های قبل از 2008:

*سطوح (آموزش$Class)*

"موفق" "ناموفق"

*modelFit <- glm(Class ~ Day,*

*+ ## ردیف‌ها را برای داده‌های قبل از 2008 انتخاب کنید:*

*+ داده = آموزش [قبل از 2008،]،*

*+ ##* ' *خانواده* ' *به توزیع داده‌ها مربوط می‌شود.*

*+ ## مقدار* " *دو جمله* ای" *برای رگرسیون لجستیک استفاده می‌شود*

*+ خانواده = دوجمله ای)*

*> modelFit*

تماس: glm (فرمول = کلاس ~ روز، خانواده = دوجمله ای، داده = آموزش[pre2008، ])

ضرایب:

(عرض از مبدأ) روز

-0. 91934 0. 00424

درجه آزادی: 6632 مجموع (یعنی صفر). 6631 باقیمانده

انحراف تهی: 9190

انحراف باقیمانده: 8920 AIC: 8920

تابع glm سطح عامل *دوم* را به‌عنوان رویداد مورد علاقه در نظر می‌گیرد. از آنجایی که شیب برای روز سال مثبت است، نشان دهنده افزایش نرخ کمک‌های بلاعوض است. برای بدست آوردن احتمال موفقیت آمیز گرنت، از یک کم می‌کنیم:

*successProb <- 1 - پیش‌بینی(modelFit,*

*+ ## برای چند روز پیش‌بینی کنید*

*+ newdata = data. frame(Day = c(10, 150, 300,*

*350))*

*+ ## glm کلاس را پیش‌بینی نمی‌کند، اما می‌تواند*

*+ ## احتمال رویداد را تولید می‌کند*

*+ نوع = "پاسخ")*

*موفقیت پروب*

1234

0. 70619 0. 57043 0. 41287 0. 36262

برای اضافه کردن عبارت غیر خطی برای روز سال، فرمول قبلی به صورت زیر اضافه می‌شود:

*daySquaredModel <- glm(Class ~ Day + I(Day''2),*

*+ داده = آموزش [قبل از 2008،]،*

*+ خانواده = دوجمله ای)*

*daySquaredModel*

تماس: glm (فرمول = کلاس ~ روز + I(روز"2)، خانواده = دوجمله ای، داده = آموزش[pre2008، ])

ضرایب:

(عرض از مبدأ) روز اول (روز"2)

-1. 881341 0. 018622 -0. 000038

درجه آزادی: 6632 مجموع (یعنی صفر). 6630 باقی مانده

انحراف تهی: 9190

انحراف باقیمانده: 8720 AIC: 8730

تابع glm روش غیر فرمولی ندارد، بنابراین ایجاد مدل‌هایی با تعداد زیادی پیش‌بینی کار کمی بیشتر می‌طلبد. یک راه حل جایگزین در زیر نشان داده شده است.

R دیگر برای مدل لجستیک در بسته [p](#bookmark1018) مرتبط با Har ­rell [( 2001](#bookmark1018) )، به نام rms (برای استراتژی‌های مدل‌سازی رگرسیون) است. تابع lrm بسیار شبیه به glm است و شامل توابع کمکی است. به‌عنوان مثال، یک *اسپلاین مکعبی محدود* ابزاری برای برازش توابع غیرخطی انعطاف‌پذیر یک پیش‌بین است. برای روز سال:

*کتابخانه (rms)*

*rcsFit <- lrm(Class ~ rcs(Day), data = training[pre2008,]) > rcsFit*

مدل رگرسیون لجستیک

lrm (فرمول = کلاس ~ rcs (روز)، داده = آموزش[pre2008، ])

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | احتمال مدل  تست نسبت | | تبعیض  شاخص ها | | تبعیض رتبه  شاخص ها | |
| Obs | 6633 | LR chi2 | 461. 53 | R2 | 0. 090 | سی | 0. 614 |
| موفقیت آمیز | 3233 | df | 4 | g | 0. 538 | Dxy | 0. 229 |
| ناموفق | 3400 | Pr(> chi2) | <0. 0001 | گرم | 1. 713 | گاما | 0. 242 |
| حداکثر | مشتق | | 2e-06 |  |  | gp | 0. 122 | tau-a | 0. 114 |
|  |  |  |  | بریر | 0. 234 |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Coef | SE | Wald Z Pr(>|Z|) |
| عرض از مبدأ | -1. 6833 | 0. 1110 | -15. 16 <0. 0001 |
| روز | 0. 0124 | 0. 0013 | 9. 24 <0. 0001 |
| روز | -0. 0072 | 0. 0023 | -3. 17 0. 0015 |
| روز" | 0. 0193 | 0. 0367 | 0. 52 0. 6001 |
| روز'' | -0. 0888 | 0. 1026 | -0. 87 0. 3866 |

تابع lrm، مانند glm، احتمال سطح عامل دوم را مدل می‌کند. جدول پایین در خروجی مقادیر *p* را برای مولفه‌های غیرخطی مختلف اسپلاین مکعبی محدود نشان می‌دهد. از آنجایی که مقادیر *p* - برای سه جزء غیرخطی اول کوچک هستند، این نشان می‌دهد که یک رابطه غیرخطی بین کلاس و روز باید استفاده شود. این بسته حاوی تابع دیگری به نام Predict است که به سرعت یک نمایه پیش‌بینی در یک یا چند متغیر ایجاد می‌کند. مثلا کد

*dayProfile <- Predict(rcsFit,*

*+ ## محدوده متغیر نمودار را مشخص کنید*

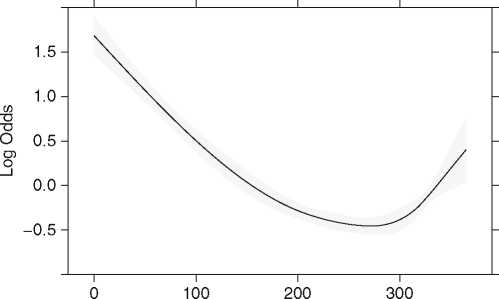
*+ روز = 0:365،*

*+ ## پیش‌بینی را ورق بزنید تا مدل مورد نظر را بدست آورید*

*+ ## گرنت موفق*

*+ سرگرمی = تابع (x) -x)*

*نمودار (dayProfile، ylab = "Log Odds")*



روز

شکل 12. 20: یک اسپلاین مکعبی محدود برای تولید در روز سال تولید ­شده توسط بسته rms. نوارهای *خاکستری* محدودیت‌های اطمینان در شانس ورود به سامانه هستند

تصویر موجود در شکل را تولید می‌کند.  [12. 20](#bookmark591) . استدلال سرگرم کننده علائم پیش‌بینی را تغییر می‌دهد به‌طوری که طرح احتمال یک گرنت موفق را منعکس می‌کند. از این نمودار، آشکار است که یک ترم درجه دوم برای روز سال، روندهای نشان داده شده توسط اسپلاین را تقریب می‌کند.

بسته rms شامل بسیاری از توابع مرتبط است، از جمله تکنیک‌های resam pling ­برای اعتبارسنجی مدل و توابع تجسم مدل. [هارل](#bookmark1018) را ببینید [( 2001](#bookmark1018) ) برای جزئیات روش‌ها و کد R.

برای مجموعه بزرگی از پیش‌بینی‌ها، روش فرمول برای تعیین مدل‌ها می‌تواند دست و پا گیر باشد. همانند فصل‌های قبلی، تابع آموزش می‌تواند مدل‌ها را به‌طور کارآمد برازش داده و اعتبارسنجی کند. برای رگرسیون لجستیک، train یک رابط برای تابع glm فراهم می‌کند که یک فرمول مدل را دور می‌زند، مستقیماً پیش‌بینی‌کننده‌های کلاس را تولید می‌کند و مساحت زیر منحنی ROC و سایر معیارها را محاسبه می‌کند.

قبل از برازش مدل، مجموعه داده‌ها و گروه‌های پیش‌بینی را با متغیر روز مربع افزایش می‌دهیم:

*train$Day2 <- training$Day"2*

*fullSet <- c(fullSet، "Day2")*

*smallSet <- c(reducedSet، "Day2")*

برای داده‌های گرنت، کدی که با یک مدل با مجموعه پیش‌بینی کامل مطابقت دارد:

*کتابخانه (کارت)*

*set. seed(476)*

*> lrFull <- train(training[,fullSet],*

*+ y = آموزش$Class،*

*+ روش = "glm"،*

*+ معیار = "ROC"،*

*+ trControl = ctrl)*

*> lrFull*

8190 نمونه

1071 پیش‌بینی

2 کلاس: "موفق"، "ناموفق"

بدون پیش پردازش

نمونه‌برداری مجدد: تقسیم‌های تمرین/تست مکرر (1 تکرار، 0. 75%)

خلاصه حجم نمونه: 6633

نتایج نمونه‌گیری مجدد

ROC Sens Spec 0. 78 0. 77 0. 76

توجه داشته باشید که بالای این خروجی نشان‌دهنده استفاده از 8190 گرنت است، اما « خلاصه اندازه‌های نمونه » مقدار 6633 نقطه داده را فهرست می‌کند. این ­عدد آخر منعکس کننده مجموعه واحد نمونه‌های قبل از سال 2008 است (به جدول مراجعه کنید [12. 2 )](#bookmark557) . " نتایج نمونه برداری مجدد " در واقع برآورد عملکرد ­مجموعه انتظاری سال 2008 است.

برای ایجاد یک مدل با مجموعه پیش‌بینی کوچکتر:

*> set. seed(476)*

*lrReduced <- train(training[,reducedSet],*

*+ y = آموزش$Class،*

*+ روش = "glm"،*

*+ معیار = "ROC"،*

*+ trControl = ctrl)*

*lrکاهش یافته است*

8190 نمونه

253 پیش‌بینی

2 کلاس: "موفق"، "ناموفق"

بدون پیش پردازش

نمونه‌برداری مجدد: تقسیم‌های تمرین/تست مکرر (1 تکرار، 0. 75%)

خلاصه حجم نمونه: 6633

نتایج نمونه‌گیری مجدد

ROC Sens Spec 0. 87 0. 8 0. 82

مانند تحلیل LDA، حذف پیش‌بینی‌کننده‌های واریانس نزدیک به صفر تأثیر مثبتی بر برازش مدل دارد. پیش‌بینی‌کننده‌های مجموعه نگهدارنده (گرنت سال 2008) در زیر شی pred موجود است :

*> head(lrReduced$pred)*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | پیش | obs | موفقیت آمیز | ناموفق | rowIndex | . پارامتر |
| 6634 | موفقیت آمیز | موفقیت آمیز | 0. 99878 | 0. 0012238 | 6634 | هیچ یک |
| 6635 | موفقیت آمیز | موفقیت آمیز | 0. 85151 | 0. 1484924 | 6635 | هیچ یک |
| 6636 | موفقیت آمیز | موفقیت آمیز | 0. 92019 | 0. 0798068 | 6636 | هیچ یک |
| 6637 | موفقیت آمیز | موفقیت آمیز | 0. 96694 | 0. 0330572 | 6637 | هیچ یک |
| 6639 | موفقیت آمیز | موفقیت آمیز | 0. 98928 | 0. 0107160 | 6638 | هیچ یک |
| 6642 | موفقیت آمیز | موفقیت آمیز | 0. 57563 | 0. 4243729 | 6639 | هیچ یک |

نمونه‌گیری مجدد

6634 TrainSet

6635 TrainSet

6636 TrainSet

6637 TrainSet

6639 TrainSet

6642 TrainSet

به ستونی در خروجی با برچسب. parameter توجه کنید. وقتی آموزش پیش‌بینی‌کننده‌ها را ذخیره می‌کند، این کار را برای هر پارامتر تنظیم انجام می‌دهد. این ستون در خروجی برای برچسب زدن مدلی که پیش‌بینی‌کننده‌ها را ایجاد کرده است استفاده می‌شود. این نسخه از رگرسیون لجستیک هیچ پارامتر تنظیمی ندارد، بنابراین پارامتر. یک مقدار واحد دارد ( "none" ).

از این داده‌ها، ماتریس سردرگمی را می‌توان محاسبه کرد:

*> confusionMatrix(data = lrReduced$pred$pred,*

*+ مرجع = lrکاهش$pred$obs)*

ماتریس سردرگمی و آمار

|  |  |  |
| --- | --- | --- |
|  | ارجاع | |
| پیش‌بینی | موفقیت آمیز | ناموفق |
| موفقیت آمیز | 458 | 176 |
| ناموفق | 112 | 811 |

دقت: 0. 815 95% فاصله اطمینان (CI): (0. 795, 0. 834)

بدون اطلاعات نرخ : 0. 634

P-Value [Acc > NIR]: < 2e-16

کاپا: 0. 611 آزمون مک نمار P-Value: 0. 000205

حساسیت: 0. 804

ویژگی: 0. 822

Pos Pred Value: 0. 722

ارزش Neg Pred: 0. 879

شیوع: 0. 366

میزان تشخیص: 0. 294

شیوع تشخیص: 0. 407

کلاس "مثبت": موفق

این نتایج با مقادیر نشان داده شده در بالا برای lrReduced مطابقت دارد. منحنی ROC را نیز می‌توان با استفاده از بسته pROC محاسبه و ترسیم کرد :

*smallRoc <- roc(response = lrReduced$pred$obs,*

*+ پیش‌بینی = lrکاهش$pred$موفق،*

*+ سطوح = دور (سطوح (lrکاهش$pred$obs)))*

*plot (reducedRoc، legacy. axes = TRUE)*

*auc(reducedRoc)*

مساحت زیر منحنی: 0. 872

تحلیل تشخیص خطی

یک تابع محبوب برای ایجاد مدل‌های LDA lda در بسته MASS است. ورودی این تابع می‌تواند یک فرمول و چارچوب داده یا ماتریسی از پیش‌بینی‌کننده‌ها و یک متغیر گروه‌بندی به‌عنوان عاملی باشد که حاوی اطلاعات عضویت در کلاس است. ما می‌توانیم مدل LDA را به صورت زیر برازش دهیم:

*کتابخانه (MASS)*

*## ابتدا داده‌ها را مرکز و مقیاس کنید*

*grantPreProcess <- preProcess(training[pre2008, smallSet])*

*grantPreProcess*

صدا زدن:

preProcess. default(x = training[pre2008, smallSet])

ایجاد شده از 6633 نمونه و 253 متغیر

پیش پردازش: متمرکز، مقیاس شده

*scaledPre2008 <- predict(grantPreProcess,*

*+ newdata = آموزش [pre2008, smallSet])*

*scaled2008HoldOut <- predict(grantPreProcess,*

*+ newdata = آموزش[-pre2008, smallSet])*

*ldaModel <- lda(x = scaledPre2008،*

*+ گروه‌بندی = آموزش$Class[pre2008])*

به یاد بیاورید که چون این داده‌ها شامل دو کلاس هستند، تنها یک بردار متمایز می‌توان به دست آورد. این بردار متمایز در شی ldaModel$scaling موجود است. شش ورودی اول این ماتریس عبارتند از:

*head(ldaModel$scaling)*

LD1

NumCI 0. 1301673

NumDR 0. 0017275

NumECI 0. 1219478

NumPS 0. 0042669

NumSR -0. 0642209

NumSCI -0. 0655663

این اطلاعات تفسیری در مورد پیش‌بینی‌کننده‌ها، روابط ­بین پیش‌بینی‌کننده‌ها و اگر داده‌ها متمرکز و مقیاس‌بندی شده‌اند، مقادیر اهمیت نسبی را ارائه می‌دهد. بردار تفکیک کننده در پیش‌بینی ­نمونه‌ها نقش دارد و بسته MASS این فرآیند را از طریق تابع پیش‌بینی ساده می‌کند. برای مجموعه آزمون داده‌های گرنت، پیش‌بینی‌کننده‌ها با نحو تولید می‌شوند:

*ldaHoldOutPredictions <- predict(ldaModel, scaled2008HoldOut)*

کلاس پیش‌بینی‌شده، احتمال پسین و مقدار متمایز خطی همگی در این شی وجود دارند، بنابراین کاربر را قادر می‌سازد (1) یک ماتریس سردرگمی از مقادیر مشاهده شده در مقابل مقادیر پیش‌بینی‌شده، (2) توزیع ­احتمالات پسین و/یا ایجاد کند. (3) توزیع مقادیر متمایز خطی.

یک مفهوم مستقیم از تنظیمات دو کلاسه این است که هیچ آموزشی در مورد تعداد بردارهای متمایز وجود ندارد که برای پیش‌بینی حفظ شود. هنگام کار با داده‌هایی که بیش از دو کلاس دارند، تعداد بهینه بردارهای متمایز خطی را می‌توان از طریق فرآیند اعتبارسنجی معمول تعیین کرد. از طریق تابع lda، تعداد متمایز کننده‌های خطی برای حفظ پیش‌بینی را می‌توان با گزینه dimen تابع پیش‌بینی تنظیم کرد. به‌طور ­معمول، این فرآیند بهینه‌سازی با عملکرد آموزش در بسته کارت خودکار می‌شود:

*set. seed(476)*

*> ldaFit1 <- train(x = training[, smallSet],*

*+ y = آموزش$Class،*

*+ روش = "lda"،*

*+ preProc = c ("مرکز"، "مقیاس")،*

*+ معیار = "ROC"،*

*+ ## در بالا تعریف شده است*

*+ trControl = ctrl)*

*> ldaFit1*

8190 نمونه

253 پیش‌بینی

2 کلاس: "موفق"، "ناموفق"

پیش پردازش: متمرکز، مقیاس شده

نمونه‌برداری مجدد: تقسیم‌های تمرین/تست مکرر (1 تکرار، 0. 75%)

خلاصه حجم نمونه: 6633

نتایج نمونه‌گیری مجدد

مشخصات ROC Sens

0. 89 0. 8 0. 82

هیچ تنظیم رسمی رخ نمی‌دهد زیرا فقط دو کلاس و بنابراین فقط یک بردار متمایز وجود دارد. ما می‌توانیم طبقات و احتمالات پیش‌بینی‌شده را برای مجموعه تست به روش معمول ایجاد کنیم:

*ldaTestClasses <- predict(ldaFit1,*

*+ newdata = تست[,reducedSet])*

*ldaTestProbs <- predict(ldaFit1,*

*+ newdata = تست[,reducedSet]،*

*+ نوع = "مسأله")*

هنگامی که مسأله بیش از دو کلاس را شامل می‌شود و ما می‌خواهیم تعداد بردارهای متمایز را بهینه کنیم، تابع آموزش همچنان می‌تواند با روش تنظیم شده روی " lda2 " و تنظیم طول روی حداکثر تعداد ابعادی که متخصص می‌خواهد ارزیابی کند، استفاده شود. .

تحلیل تفکیک حداقل مربعات جزئی

PLSDA را می‌توان با استفاده از تابع plsr در بسته pls با استفاده از یک ماتریس طبقه‌بندی که دسته‌های پاسخ را تعریف می‌کند، انجام داد. ما خواننده را به فرقه ارجاع می‌دهیم.  [6. 3](#bookmark8) برای توصیف تغییرات الگوریتمی PLS که مستقیماً به تنظیمات طبقه‌بندی گسترش می‌یابد.

بسته caret حاوی یک تابع ( plsda ) است که می‌تواند ­مدل PLS متغیر ساختگی مناسب برای داده‌ها ایجاد کند و سپس پیش‌بینی‌کننده‌های مدل خام را پس پردازش کند تا احتمالات کلاس را برگرداند. نحو بسیار شبیه به کد مدل رگرسیون برای PLS است که در بخش ارائه شده است.  [6. 3 .](#bookmark8) تفاوت اصلی این است که از متغیر عاملی برای نتیجه استفاده می‌شود.

به‌عنوان مثال، برای برازش مدل با مجموعه پیش‌بینی کاهش یافته:

*> plsdaModel <- plsda(x = training[pre2008,reducedSet],*

*+ y = آموزش [قبل از 2008، "کلاس"]،*

*+ ## داده‌ها باید برای PLS در یک مقیاس باشند. را*

*+ ##* ' *scale* ' *این مرحله پیش پردازش را اعمال می‌کند*

*+ مقیاس = درست،*

*+ ## برای محاسبه احتمالات از روش بیز استفاده کنید*

*+ probMethod = "Bayes"،*

*+ ## تعداد اجزای مورد نظر را برای مدل‌سازی مشخص کنید*

*+ ncomp = 4)*

*## مجموعه انتظاری 2008 را پیش‌بینی کنید*

*plsPred <- predict(plsdaModel,*

*+ newdata = آموزش[-pre2008, smallSet])*

*سر (plsPred)*

[1] موفق موفق موفق موفق موفق موفق

سطوح: موفق ناموفق

*plsProbs <- predict(plsdaModel,*

*+ newdata = آموزش[-pre2008, smallSet]،*

*+ نوع = "مسأله")*

*سر (plsProbs)*

[1] 0. 98842 0. 88724 0. 83455 0. 88144 0. 94848 0. 53991

شی plsdaModel تمام توابع مشابهی را به ارث می‌برد که از شیء که مستقیماً از تابع plsr می‌آید به دست می‌آید. به همین دلیل می‌توان از توابع دیگر بسته pls مانند بارگیری یا نمودار امتیاز استفاده کرد.

تابع آموزش همچنین می‌تواند با PLS در تنظیمات طبقه‌بندی استفاده شود. کد زیر ده جزء اول PLS را با توجه به ناحیه زیر منحنی ROC ارزیابی می‌کند و همچنین پیش‌بینی‌کننده‌ها را قبل از برازش مدل و پیش‌بینی نمونه به‌طور خودکار مرکز و مقیاس می‌کند:

*> set. seed(476)*

*> plsFit2 <- train(x = training[, smallSet],*

|  |  |
| --- | --- |
| *+*  *+*  *+*  *+*  *+*  *+* | *y = training$Class، متد = "pls"، tuneGrid = expand. grid(. ncomp = 1:10)، preProc = c("center","scale")، معیار = "ROC"، trControl = ctrl)* |

اولیه پیش‌بینی نمونه‌های جدید را ارزیابی می‌کند و type = "prob" احتمالات کلاس را برمی گرداند. محاسبه اهمیت متغیر همانطور که در شکل 1 نشان داده شده است.  [12. 15](#bookmark576) را می‌توان با کد زیر انجام داد:

*plsImpGrant <- varImp(plsFit2، scale = FALSE)*

*plsImpGrant*

pls اهمیت متغیر

تنها 20 متغیر مهم نشان داده شده است (از 253)

|  |  |
| --- | --- |
|  | به‌طور کلی |
| ContractValueBandUnk | 0. 0662 |
| حامی Unk Jan | 0. 0383  0. 0338 |
| ناموفق. CI | 0. 0329 |
| ContractValueBandA | 0. 0316 |
| روز  اوت  موفقیت. CI | 0. 0266  0. 0257  0. 0219 |
| GrantCat10A | 0. 0211 |
| روز 2  GrantCat30B | 0. 0209  0. 0202 |
| ContractValueBandE | 0. 0199 |
| ContractValueBandD | 0. 0193 |
| ContractValueBandF | 0. 0188 |
| ContractValueBandG | 0. 0184 |
| Sponsor24D Sponsor21A Sponsor2B NumSR | 0. 0172  0. 0169  0. 0147  0. 0144 |
| ژوئیه | 0. 0124 |

*> نمودار (plsImpGrant، top = 20، مقیاس = لیست (y = لیست (cex = 0. 95)))*

مدل‌های جریمه شده

بسته اولیه برای رگرسیون لجستیک جریمه‌شده glmnet است (اگرچه فصل بعدی نحوه تطبیق مدل‌های مشابه با استفاده از شبکه‌های عصبی را توضیح می‌دهد). تابع glmnet بسیار شبیه به تابع enet است که قبلا در بخش توضیح داده شد.  [6. 5](#bookmark321) . آرگومان‌های اصلی با داده‌ها مطابقت دارند: x *ماتریسی* از پیش‌بینی‌کننده‌ها و y عاملی از کلاس‌ها (برای رگرسیون لجستیک) است. علاوه بر این، بحث خانواده با توزیع نتیجه مرتبط است. برای دو کلاس، استفاده از خانواده = "دو جمله ای" با رگرسیون لجستیک مطابقت دارد و زمانی که سه یا بیشتر کلاس وجود دارد، خانواده "چند جمله ای" مناسب است.

این تابع به‌طور خودکار دنباله‌ای از مقادیر را برای مقدار تنظیم انتخاب می‌کند، اگرچه کاربر می‌تواند مقادیر خود را با گزینه لامبدا انتخاب کند. به یاد بیاورید که نوع تنظیم با پارامتر اختلاط *a تعیین می‌شود.* glmnet این پارامتر را به صورت پیش فرض آلفا = 1 قرار می‌دهد که مربوط به یک جریمه کامل ریج است.

تابع پیش‌بینی برای glmnet انواع مختلفی از مقادیر ­را پیش‌بینی می‌کند، از جمله: کلاس پیش‌بینی‌شده، پیش‌بینی‌کننده‌هایی که در مدل استفاده می‌شوند، و/یا تخمین‌های پارامتر رگرسیون. مثلا:

*کتابخانه (glmnet)*

*glmnetModel <- glmnet(x = as. matrix(training[,fullSet])،*

*+ y = آموزش$Class،*

*+ خانواده = "دوجمله ای")*

*## پیش‌بینی‌کننده‌ها را برای سه سطح مختلف منظم‌سازی محاسبه کنید.*

*## توجه داشته باشید که نتایج فاکتور نیستند*

*پیش‌بینی(glmnetModel,*

*+ newx = as. matrix(training[1:5,fullSet])،*

*+ s = c(0. 05، 0. 1، 0. 2)،*

*+ نوع = "کلاس")*

123

"موفق" "موفق" "ناموفق"

"موفق" "موفق" "ناموفق"

"موفق" "موفق" "ناموفق"

"موفق" "موفق" "ناموفق"

"موفق" "موفق" "ناموفق"

*## کدام پیش‌بینی‌کننده‌ها در مدل استفاده شده اند؟*

*پیش‌بینی(glmnetModel,*

*+ newx = as. matrix(training[1:5,fullSet])،*

*+ s = c(0. 05، 0. 1، 0. 2)،*

*+ نوع = "غیر صفر")*

$'1'

[1] 71 72 973 1027 1040 1045 1055

$'2'

[1] 1027 1040

$'3'

[1]

1040

به‌عنوان یک نکته جانبی، بسته glmnet تابعی به نام auc دارد. اگر بسته pROC قبل از بارگیری glmnet بارگیری شود، این پیام ظاهر می‌شود: « اشیاء(های) زیر از «package:pROC» پوشانده شده اند: auc. اگر این تابع در این نقطه فراخوانی شود، R نامشخص خواهد بود که در کدام مورد استفاده شود. دو رویکرد متفاوت برای مقابله با این موضوع وجود دارد:

اگر یکی از بسته‌ها دیگر مورد نیاز نباشد، می‌توان آن را با استفاده از detach (package:pROC) جدا کرد.

تابع مناسب را می‌توان با استفاده از قرارداد *فضای نام* هنگام فراخوانی تابع فراخوانی کرد. برای مثال، pROC:::auc و glmnet:::auc به توابع خاص اشاره می‌کنند.

نمونه بالقوه دیگری از این موضوع در زیر توضیح داده شده است.

تنظیم مدل با استفاده از ناحیه زیر منحنی ROC را می‌توان با آموزش انجام داد. برای داده‌های گرنت:

*## مقادیر تنظیم را مشخص کنید:*

*glmnGrid <- expand. grid(. alpha = c(0، 0. 1، 0. 2، 0. 4، 0. 6، 0. 8، 1)،*

*+ . lambda = seq(. 01،. 2، طول = 40))*

*set. seed(476)*

*> glmnTuned <- train(training[,fullSet],*

*+ y = آموزش$Class،*

*+ روش = "glmnet"،*

*+ tuneGrid = glmnGrid،*

*+ preProc = c ("مرکز"، "مقیاس")،*

*+ معیار = "ROC"،*

*+ trControl = ctrl)*

نقشه حرارتی در پانل بالای شکل.  [12. 16](#bookmark577) با استفاده از نمودار کد (glmnTuned، plotType = "سطح") تولید شد. توابع LDA مجازات شده را می‌توان در بسته‌های sparseLDA و PenalizedLDA یافت. تابع اصلی در بسته sparseLDA sda نامیده می‌شود. [[43]](#footnote-43) این تابع یک آرگومان برای پارامتر ridge به نام lambda دارد. پنالتی ریج را می‌توان به دو ­صورت ممکن با استدلال توقف بیان کرد. مقدار جریمه ریج با استفاده از یک عدد مثبت کنترل می‌شود (به‌عنوان مثال، توقف = 0. 01 ) یا، به‌طور متناوب، تعداد پیش‌بینی‌کننده‌های حفظ شده را می‌توان با استفاده از یک عدد صحیح *منفی انتخاب کرد (*به‌عنوان مثال، توقف = -6 برای شش پیش‌بینی). مثلا:

*کتابخانه (sparseLDA)*

*sparseLdaModel <- sda(x = as. matrix(training[,fullSet])،*

*+ y = آموزش$Class،*

*+ لامبدا = 0. 01،*

*+ توقف = -6)*

آرگومان = "sparseLDA" را می‌توان با train استفاده کرد. در این مورد، آموزش مدل را روی لامبدا و تعداد پیش‌بینی‌کننده‌های حفظ شده تنظیم می‌کند.

نزدیکترین مرکزهای کوچک شده

اصلی R برای این مدل در بسته pamr (برای " *P* redictive *A* تحلیل *M* icroarrays در R " یافت می‌شود. بسته دیگری، rda، شامل پسوندهایی برای مدلی است که در آن توضیح داده شده است [گوو و همکاران ( 2007](#bookmark1017) ).

نحو توابع در بسته pamr تا حدودی غیر استاندارد است. تابع آموزش مدل pamr. train است که داده‌های ورودی را در یک شی فهرست واحد با اجزای x و y می‌گیرد. قرارداد معمول برای مجموعه داده‌ها این است که نمونه‌ها در ردیف‌ها و ستون‌های مختلف برای پیش‌بینی‌کننده‌ها وجود داشته باشد. pamr. train به پیش‌بینی‌کننده‌های مجموعه آموزشی نیاز دارد که در قالب مخالف کدگذاری شوند، جایی که ردیف‌ها پیش‌بینی و ستون‌ها نمونه هستند. [[44]](#footnote-44) برای داده‌های گرنت، داده‌های ورودی به شکل زیر خواهد بود:

*## برای جابجایی داده‌ها، ابعاد را با استفاده از تابع t() تغییر دهید.*

*## این همچنین به‌طور ضمنی چارچوب داده‌های آموزشی را به یک ماتریس تبدیل می‌کند.*

*داده‌های ورودی <- لیست (x = t(آموزش[، مجموعه کامل])، y = آموزش$Class)*

نحو اصلی برای ایجاد مدل به شرح زیر است:

*کتابخانه (pamr)*

*nscModel <- pamr. train(data = inputData)*

به‌طور پیش فرض، تابع 30 مقدار انقباض مناسب را برای ارزیابی انتخاب می‌کند. گزینه‌هایی برای استفاده از مقادیر خاص برای مقدار انقباض، احتمالات قبلی و سایر جنبه‌های مدل وجود دارد. تابع pamr. predict پیش‌بینی‌کننده‌هایی را روی نمونه‌های جدید ایجاد می‌کند و همچنین تعیین می‌کند که کدام پیش‌بینی‌کننده‌های خاص در مدل برای یک مقدار انقباض مشخص استفاده شده‌اند. به‌عنوان مثال، برای تعیین مقدار انقباض 5:

*exampleData <- t(training[1:5, fullSet])*

*pamr. predict(nscModel، newx = exampleData، آستانه = 5)*

[1] موفق ناموفق موفق ناموفق موفق

سطوح: موفق ناموفق

*## کدام پیش‌بینی‌کننده‌ها در این آستانه استفاده شدند؟ پیش‌بینی*

*تابع ## اعداد ستون‌ها را برای پیش‌بینی‌کننده‌های حفظ شده نشان می‌دهد.*

*thresh17Vars <- pamr. predict(nscModel, newx = exampleData,*

*+ آستانه = 17، نوع = "غیر صفر")*

*fullSet[thresh17Vars]*

[1] " Unsuccess. CI" "SponsorUnk" "ContractValueBandA"

[4] "ContractValueBandUnk" "Jan"

این بسته همچنین دارای عملکردهایی برای اعتبار سنجی متقاطع *K* -fold برای انتخاب مقدار مناسب انقباض است، اما به یک نوع resam pling محدود می‌شود ­و مدل را با دقت کلی تنظیم می‌کند. نحو آموزش به صورت زیر است:

*## ما محدوده خاصی از پارامترهای تنظیم را در اینجا انتخاب کردیم:*

*nscGrid <- data. frame(. threshold = 0:25)*

*set. seed(476)*

*nscTuned <- train(x = training[,fullSet],*

*+ y = آموزش$Class،*

*+ روش = "پم"،*

*+ preProc = c ("مرکز"، "مقیاس")،*

*+ tuneGrid = nscGrid،*

*+ معیار = "ROC"،*

*+ trControl = ctrl)*

این رویکرد گزینه‌های بیشتری را برای تنظیم مدل (به‌عنوان مثال، استفاده از ناحیه زیر منحنی ROC) و همچنین یک نحو سازگار فراهم می‌کند. تابع پیش‌بینی برای آموزش نیازی به تعیین دستی مقدار انقباض توسط کاربر ندارد (مقدار بهینه تعیین شده توسط تابع به‌طور خودکار استفاده می‌شود).

تابع پیش‌بینی‌کننده‌ها پیش‌بینی‌کننده‌های مورد استفاده در معادله پیش‌بینی (در آستانه بهینه تعیین شده توسط آموزش ) را فهرست می‌کند. در مدل تیونینگ شده 36 مورد انتخاب شدند:

*> پیش‌بینی‌کننده‌ها (nscTuned)*

|  |  |  |
| --- | --- | --- |
| [1] "NumSR" | "موفقیت. CI" | "ناموفق. CI" |
| [4] "CI. Faculty13" | "CI. Faculty25" | "CI. Faculty58" |
| [7] "DurationGT15" | "Astar. CI" | "AstarTotal" |
| [10] "allPub" | "اسپانسر21A" | "Sponsor24D" |
| [13] "Sponsor2B" | "Sponsor34B" | "Sponsor4D" |
| [16] "Sponsor62B" | "Sponsor6B" | "Sponsor89A" |
| [19] "SponsorUnk" | "ContractValueBandA" | "ContractValueBandC |
| [22] "ContractValueBandD" | "ContractValueBandE" | "ContractValueBandF |
| [25] "ContractValueBandG" | "ContractValueBandUnk" | "GrantCat10A" |
| [28] "GrantCat30B" | "مرداد" | "دسامبر" |
| [31] "جان" | "ژوئیه" | "جمعه" |
| [34] "خورشید" | "روز" | "روز 2" |

همچنین، تابع varImp اهمیت متغیر را بر اساس فاصله بین کلاس Centroid و Centroid کلی برمی‌گرداند:

*varImp(nscTuned، scale = FALSE)*

اهمیت متغیر pam

تنها 20 متغیر مهم نشان داده شده است (از 1071)

|  |  |
| --- | --- |
|  | اهمیت |
| ContractValueBandUnk | -0. 2260 |
| SponsorUnk | 0. 1061 |
| ژان | 0. 0979 |
| ContractValueBandA | 0. 0948 |
| ناموفق. CI | -0. 0787 |
| روز | -0. 0691 |
| اوت | -0. 0669 |
| آفتاب | 0. 0660 |
| GrantCat10A | -0. 0501 |
| موفقیت. CI | 0. 0413 |

روز 2

-0.0397 0.0380

-0.0379 0.0344 0.0340 0.0333 0.0329

-0.0299

-0.0233

0.0224

ContractValueBandE GrantCat30B

ContractValueBandD حامی 21A

ContractValueBandF ContractValueBandG Sponsor24D Sponsor2B

NumSR

در این داده‌ها، علامت تفاوت، جهت تاثیر پیش‌بینی را نشان می‌دهد. به‌عنوان مثال، زمانی که باند پیمانکار ناشناخته است، تنها درصد کمی از کمک‌های بلاعوض موفق هستند (19. 4٪ در مقابل نرخ موفقیت پایه 46. 4٪). علامت منفی برای این پیش‌بینی نشان دهنده افت نرخ رویداد است. برعکس، زمانی که حامی ناشناخته باشد، میزان موفقیت بالاست (82. 2 درصد؛ جدول را ببینید [12. 1 )](#bookmark552) . فاصله برای این پیش‌بینی مثبت است که نشان دهنده افزایش نرخ رویداد است.

تمرینات

مجموعه داده‌های آسیب کبدی در فصل مقدماتی توضیح داده شد و شامل 281 ترکیب منحصربه‌فرد است که هر کدام به‌عنوان بدون آسیب کبدی، آسیب خفیف یا آسیب شدید طبقه‌بندی شده‌اند (شکل 1).  [1. 2 )](#bookmark84) . این ترکیبات با 184 صفحه نمایش بیولوژیکی (یعنی آزمایش) برای ارزیابی اثر هر ترکیب بر روی یک هدف مرتبط بیولوژیکی خاص در بدن مورد تحلیل قرار گرفتند. هر چه مقدار هر یک از این پیش‌بینی‌کننده‌ها بزرگتر باشد، فعالیت ترکیب بالاتر است. علاوه بر صفحه نمایش‌های بیولوژیکی، 192 پیش‌بینی اثر انگشت شیمیایی برای این ترکیبات تعیین شد. هر یک از این پیش‌بینی‌کننده‌ها نشان‌دهنده یک زیرساخت (یعنی یک اتم یا ترکیبی از اتم‌های درون ترکیب) هستند و یا شمارشی از تعداد زیرساخت‌ها هستند یا نشان‌دهنده حضور یا عدم وجود زیرساخت خاص هستند. هدف از این مجموعه داده ایجاد یک مدل پیش‌بینی برای آسیب کبدی است به‌طوری که سایر ترکیبات را بتوان برای احتمال ایجاد آسیب کبدی غربالگری کرد. R را راه اندازی کنید و از این دستورات برای بارگیری داده‌ها استفاده کنید:

*> کتابخانه (کارت)*

*> داده‌ها (کبدی)*

*> # برای مشاهده جزئیات بیشتر از ?hepatic استفاده کنید*

ماتریس‌های بیو و شیمی حاوی سنجش بیولوژیکی و ­پیش‌بینی‌کننده‌های اثر باله شیمیایی برای ۲۸۱ ترکیب هستند، در حالی که آسیب ناقل شامل طبقه‌بندی آسیب کبدی برای هر ترکیب است.

با توجه به عدم تعادل طبقه‌بندی در وضعیت آسیب کبدی، نحوه ایجاد یک مجموعه تمرین و آزمایش را توضیح دهید.

کدام آمار طبقه‌بندی را برای بهینه‌سازی این تمرین انتخاب می‌کنید ­و چرا؟

داده‌ها را به یک مجموعه آموزشی و آزمایشی تقسیم کنید، داده‌ها را از قبل پردازش کنید و مدل‌هایی را بسازید که در این فصل برای پیش‌بینی‌کننده‌های بیولوژیکی و به‌طور جداگانه برای پیش‌بینی‌کننده‌های اثرانگشت شیمیایی توضیح داده شده است. کدام مدل بهترین توانایی پیش‌بینی را برای پیش‌بینی‌کننده‌های بیولوژیکی دارد و عملکرد بهینه چیست؟ کدام مدل بهترین قابلیت پیش‌بینی را برای پیش‌بینی‌کننده‌های شیمیایی دارد و عملکرد بهینه چقدر است؟ بر اساس این نتایج، کدام مجموعه از پیش‌بینی‌کننده‌ها حاوی بیشترین اطلاعات در مورد سمیت کبدی است؟

برای مدل‌های بهینه برای پیش‌بینی‌کننده‌های بیولوژیکی و شیمیایی، پنج پیش‌بینی مهم کدامند؟

اکنون پیش‌بینی‌کننده‌های اثرانگشت بیولوژیکی و شیمیایی را در یک مجموعه پیش‌بینی ترکیب کنید. همان مجموعه‌ای از مدل‌های پیش‌بینی را که از قسمت (ج) ساخته اید، دوباره آموزش دهید. کدام مدل بهترین عملکرد پیش‌بینی را دارد؟ آیا عملکرد مدل بهتر از هر یک از بهترین مدل‌های قسمت (ج) است؟ پنج پیش‌بینی مهم برای مدل بهینه کدامند؟ اینها چگونه با پیش‌بینی‌کننده‌های بهینه از هر مجموعه پیش‌بینی منفرد مقایسه می‌شوند؟

در صورت وجود، از کدام مدل (هر مدل بیولوژی فردی یا اثر انگشت شیمیایی یا مدل پیش‌بینی ترکیبی)، برای پیش‌بینی سمیت کبدی ترکیبات استفاده می‌کنید؟ توضیح.

در ورزش [4. 4 ،](#bookmark7) مجموعه داده‌ای را توصیف کردیم که شامل 96 نمونه روغن ­هر کدام از یکی از هفت نوع روغن (کدو تنبل، آفتابگردان، بادام زمینی، زیتون، سویا، کلزا و ذرت) بود. بر روی هر نمونه گاز کروماتوگرافی انجام شد و درصد هر نوع 7 اسید چرب تعیین شد. ما می‌خواهیم از این داده‌ها برای ساخت مدلی استفاده کنیم که نوع روغن را بر اساس درصد اسیدهای چرب نمونه پیش‌بینی می‌کند.

مانند داده‌های آسیب کبدی، این داده‌ها از عدم تعادل شدید رنج می‌برند. با توجه به این عدم تعادل، آیا داده‌ها باید به مجموعه‌های آموزشی و آزمایشی تقسیم شوند؟

کدام آمار طبقه‌بندی را برای بهینه‌سازی این تمرین انتخاب می‌کنید ­و چرا؟

از بین مدل‌های ارائه شده در این فصل، کدامیک بر روی این داده‌ها بهترین عملکرد را دارد؟ کدام نوع روغن را مدل به دقت پیش‌بینی می‌کند؟ حداقل پیش‌بینی دقیق؟

وبسایت[[45]](#footnote-45) برای بسته نرم‌افزاری MLC++ حاوی تعدادی مجموعه داده یادگیری ماشین است. مجموعه داده "Churn" برای پیش‌بینی ریزش مشتریان مخابراتی بر اساس اطلاعات مربوط به حساب آنها ایجاد شده است. فایل‌های داده بیان می‌کنند که داده‌ها «مصنوعی هستند بر اساس ادعاهایی مشابه دنیای واقعی». داده‌ها شامل 19 پیش‌بینی مربوط به حساب مشتری است، مانند تعداد تماس‌های خدمات مشتری، کد منطقه و تعداد دقیقه ها. نتیجه این است که آیا مشتری سرگردان شده است یا خیر.

داده‌ها در بسته C50 موجود است و با استفاده از موارد زیر قابل بارگیری هستند:

*> کتابخانه (C50)*

*> داده (چرخش)*

*> ## دو شیء بارگذاری می‌شوند: churnTrain و churnTest*

*> خیابان (ChurnTrain)*

*> جدول (ChurnTrain$Class)*

با تجسم رابطه بین پیش‌بینی‌کننده‌ها و نتیجه، داده‌ها را کاوش کنید. آیا ویژگی‌های مهمی از خود داده‌های پیش‌بینی وجود دارد، مانند همبستگی‌های بین پیش‌بینی یا توزیع‌های منحط ­؟ آیا می‌توان از توابع بیش از یک پیش‌بینی برای مدل‌سازی موثرتر داده‌ها استفاده کرد؟

چند مدل پایه را در مجموعه آموزشی قرار دهید و آنها را از طریق نمونه برداری مجدد تنظیم کنید. برای ارزیابی اثربخشی مدل‌ها از چه معیارهایی باید استفاده کرد؟

از نمودارهای بالابر برای مقایسه مدل‌ها استفاده کنید. اگر بخواهید 80 درصد از مشتریان پرخاشگر را شناسایی کنید، چند مشتری دیگر نیز شناسایی می‌شوند؟

فصل 13

مدل‌های طبقه‌بندی غیرخطی

فصل قبل مدل‌هایی را تشریح کرد که ذاتاً خطی بودند - ساختار مدل مرزهای کلاس خطی ایجاد می‌کرد مگر اینکه توابع غیرخطی پیش‌بینی‌کننده‌ها به صورت دستی مشخص شوند. این فصل به برخی از مدل‌های غیرخطی ذاتی می‌پردازد. همانند بخش‌های رگرسیون، مدل‌های غیرخطی دیگری نیز وجود دارند که از درخت‌ها یا قوانین برای مدل‌سازی داده‌ها استفاده می‌کنند. اینها در فصل بعد مورد بحث قرار می‌گیرند.

با چند استثنا (مانند مدل‌های FDA، بخش.  [13. 3 )](#bookmark629) ، تکنیک‌های توصیف‌شده در این فصل زمانی که تعداد زیادی از پیش‌بینی‌کننده‌های غیر اطلاعاتی به‌عنوان ورودی استفاده می‌شوند، می‌توانند تأثیر منفی بگذارند. به این ترتیب، ترکیب این مدل‌ها با ابزارهای انتخاب ویژگی (شرح شده در فصل [19 )](#bookmark864) می تواند عملکرد را به میزان قابل‌توجهی افزایش دهد. تحلیل‌های نشان‌داده‌شده در این فصل ­بدون حذف نظارت‌شده پیش‌بینی‌کننده‌های غیر اطلاعاتی انجام می‌شوند، بنابراین عملکرد احتمالاً کمتر از آن چیزی است که با یک رویکرد جامع‌تر به دست می‌آید.

13. 1 تحلیل تفکیک غیر خطی

ما در فصل قبل دیدیم که مرزهای خطی ­تحلیل غیرقابل تفکیک خطی با ایجاد برخی مفروضات بسیار خاص برای توزیع‌های اساسی پیش‌بینی‌کننده‌ها به وجود آمده است. در این بخش، روش‌هایی را بررسی می‌کنیم که روش‌های تشخیص خطی همانطور که در فصل قبل توضیح داده شد، به منظور مدیریت داده‌هایی که به بهترین شکل توسط ساختارهای غیرخطی جدا می‌شوند، اصلاح می‌شوند. این روش‌ها شامل تحلیل تفکیک درجه دوم (QDA)، تحلیل متمایز منظم (RDA) و تحلیل تفکیک مخلوط (MDA) می‌باشد.

\_13،

©

تحلیل متمایز درجه دوم و منظم

به یاد بیاورید که تحلیل تفکیک خطی می‌تواند به گونه‌ای فرموله شود که مدل آموزش دیده احتمال کل طبقه‌بندی اشتباه را به حداقل برساند. نتیجه ­این فرض که پیش‌بینی‌کننده‌ها در هر کلاس ساختار کوواریانس مشترکی دارند، این بود که مرزهای کلاس، توابع خطی پیش‌بینی‌کننده‌ها هستند.

در مدل‌های متمایز درجه دوم، این فرض آرام‌تر می‌شود تا ساختار کوواریانس کلاس خاص را بتوان تطبیق داد. پیامد اصلی ­این تغییر این است که مرزهای تصمیم اکنون در فضای پیش‌بینی منحنی درجه دوم می‌شوند. افزایش پیچیدگی تابع تفکیک ممکن است عملکرد مدل را برای بسیاری از مسائل بهبود بخشد. با این حال، یکی دیگر از پیامدهای این تعمیم این است که الزامات داده‌ها ­دقیق‌تر می‌شوند. از آنجایی که از ماتریس‌های کوواریانس مخصوص کلاس استفاده می‌شود، معکوس ماتریس‌ها باید وجود داشته باشد. این بدان معنی است که تعداد پیش‌بینی‌کننده‌ها باید کمتر از تعداد موارد در هر کلاس باشد. همچنین، پیش‌بینی‌کننده‌های هر کلاس نباید دارای سطوح پاتولوژیک همخطی باشند. علاوه بر این، اگر اکثر پیش‌بینی‌کننده‌ها در داده‌ها، شاخص‌هایی برای دسته‌های گسسته باشند، QDA فقط می‌تواند آن‌ها را به‌عنوان توابع خطی مدل‌سازی کند، بنابراین اثربخشی مدل را محدود می‌کند.

در شرایط بهینه‌سازی ریاضی محض، LDA و QDA هر کدام احتمال کل طبقه‌بندی اشتباه را به حداقل می‌رسانند با این فرض که داده‌ها واقعاً می‌توانند توسط ابرصفحه‌ها یا سطوح درجه دوم جدا شوند. با این حال، واقعیت ممکن است این باشد که داده‌ها به بهترین وجه توسط ساختارهایی در جایی بین مرزهای کلاس خطی و درجه دوم جدا شوند. RDA، پیشنهاد شده توسط [فریدمن](#bookmark1016) [( 1989](#bookmark1016) )، یکی از راه‌های پل زدن سطوح جداکننده بین LDA و QDA است. در این رویکرد، فریدمن از ماتریس کوواریانس زیر حمایت کرد:

X *, (* *A* ) = *A* X *e* + (1 *- A* ) X *,*  (13. 1)

که در آن X *^* ماتریس کوواریانس کلاس *I* و X ماتریس کوواریانس تلفیقی ­در همه کلاس‌ها است. به راحتی می‌توان فهمید که پارامتر تنظیم، *A،* این روش را قادر می‌سازد تا ماتریس کوواریانس را بین LDA (وقتی *A* = 0) و QDA (وقتی *A* = 1) خم کند. اگر یک مدل روی *A تنظیم شود،* یک رویکرد داده محور می‌تواند برای انتخاب بین مرزهای خطی یا درجه دوم و همچنین مرزهایی که بین این دو قرار می‌گیرند استفاده شود.

RDA تعمیم دیگری از داده‌ها انجام می‌دهد: می‌توان به ماتریس کوواریانس ادغام شده اجازه داد از مقدار مشاهده شده خود به مقداری تبدیل شود که پیش‌بینی‌کننده‌ها مستقل فرض می‌شوند (همانطور که توسط یک ماتریس هویت نشان داده می‌شود):

X ( *y* ) = *Y* X + (1 *- Y* ) *a* 2 I *,*  (13. 2)

که در آن *a 2* واریانس مشترک همه پیش‌بینی‌کننده‌ها و I ماتریس هویت است (یعنی ورودی‌های مورب ماتریس 1 و همه ورودی‌های دیگر 0 هستند) که مدل را مجبور می‌کند فرض کند همه پیش‌بینی‌کننده‌ها مستقل هستند. مثال آشنای دو کلاسه با دو پیش‌بینی را که آخرین بار در فصل مشاهده شد، به یاد بیاورید.  [4](#bookmark192) (پ.  [69 )](#bookmark210) . همبستگی بالایی بین این پیش‌بینی‌کننده‌ها وجود دارد که نشان می‌دهد *7* مقدار نزدیک به 1 به احتمال زیاد مناسب هستند. با این حال، در ابعاد بالاتر، تشخیص بصری چنین الگوهایی به‌طور فزاینده‌ای دشوارتر می‌شود، بنابراین تنظیم یک مدل RDA روی *A* و *y* داده‌های مجموعه آموزشی را قادر می‌سازد تا مناسب‌ترین مفروضات را برای مدل تصمیم‌گیری کنند. با این حال، توجه داشته باشید که مگر اینکه *y* یک یا *A* صفر باشد، استانداردهای داده‌های دقیق تر QDA باید اعمال شوند.

تحلیل تفکیک مخلوط

MDA توسط [هستی و تبشیرانی](#bookmark1018) ( [1996](#bookmark1018) ) به‌عنوان توسعه LDA. LDA توزیعی از داده‌های پیش‌بینی را فرض می‌کند به‌طوری که میانگین‌های ­خاص کلاس متفاوت است (اما ساختار کوواریانس مستقل از کلاس‌ها است). MDA LDA را به روشی متفاوت تعمیم می‌دهد. این اجازه می‌دهد تا هر کلاس با *چندین* توزیع نرمال چند متغیره نمایش داده شود. این توزیع‌ها می‌توانند معنی‌های مختلفی داشته باشند، اما، مانند LDA، ساختارهای کوواریانس یکسان فرض می‌شوند. شکل [13. 1](#bookmark619) این ایده را با یک پیش‌بینی واحد ارائه می‌کند. در اینجا، هر کلاس با سه توزیع نرمال با میانگین‌های مختلف و واریانس‌های مشترک نشان داده می‌شود. اینها در واقع زیر کلاس‌های داده‌ها هستند. مدل ساز مشخص می‌کند که چند توزیع مختلف باید استفاده شود و مدل MDA مکان بهینه آنها را در فضای پیش‌بینی تعیین می‌کند.

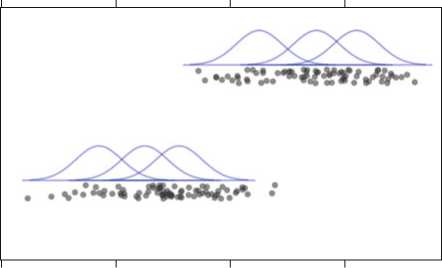
چگونه توزیع‌ها جمع می‌شوند تا بتوان یک پیش‌بینی کلاس را ­محاسبه کرد؟ در چارچوب قانون بیز (معادل 12. 4 )، MDA *Pr را اصلاح می‌کند* [ *X|Y* = *C* ]. توزیع‌های کلاس خاص با ایجاد یک مخلوط در هر کلاس در یک توزیع نرمال چند متغیره ترکیب می‌شوند. فرض کنید *D p k (* *x* ) تابع تفکیک برای *k* امین زیر کلاس در کلاس *I* است، ­تابع تمایز کلی برای کلاس *I* با آن متناسب خواهد بود.

*L t*

D p ( x ) ^ ^fyk D pk ( x ) ,   
k =1

که در آن *L* تعداد توزیع‌هایی است که برای کلاس *I استفاده می‌شود* و *^ pk* نسبت‌های اختلاط است که در طول آموزش تخمین زده می‌شود. این روی ­همه تابع متمایز می‌تواند احتمالات و پیش‌بینی‌کننده‌های کلاس را تولید کند.

برای این مدل، تعداد توزیع‌ها در هر کلاس، پارامتر تنظیم مدل است [(](#bookmark1018) نیازی نیست که آنها در هر کلاس برابر باشند). هستی و تیبشیرانی ( [1996](#bookmark1018) ) الگوریتم‌هایی را برای تعیین مقادیر شروع برای میانگین‌های کلاس خاص مورد نیاز برای هر توزیع، همراه با روال‌های بهینه‌سازی عددی برای حل معادلات غیر ضروری توصیف می‌کنند. همچنین، مشابه LDA، [کلمنسن و همکاران.](#bookmark1013) ( [2011](#bookmark1013) ) استفاده از جریمه‌های رج و ریج مانند را برای MDA توصیف می‌کند که انتخاب ویژگی را در مدل MDA ادغام می‌کند.



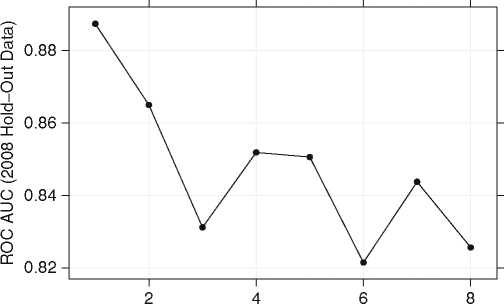
Class 2

Class 1

68

Predictor

Fig. 13.1: For a single predictor, three distinct subclasses are determined within each class using mixture discriminant analysis



#زیر کلاسها در هر کلاس

شکل 13. 2: مشخصات پارامتر تنظیم برای مدل MDA برای داده‌های گرنت. تعداد بهینه زیر کلاس‌ها 1 است که با اجرای LDA یکسان است

برای داده‌های گرنت، MDA بر روی تعداد زیر کلاس‌ها در هر گروه با مقادیر ممکن از 1 تا 8 تنظیم شد (شکل 1).  [13. 2 )](#bookmark619) . نواحی زیر منحنی ROC با استفاده از یک زیر کلاس در هر گروه که همان اجرای LDA است، بهینه شد. MDA ممکن است به دلیل تعداد زیاد پیش‌بینی‌کننده‌های باینری، با مرزهای متمایز پیچیده‌تر در این داده‌ها مخالف باشد.

13. 2 شبکه‌های عصبی

همانطور که در روش‌های طبقه‌بندی دیگر، مانند تحلیل تفکیک حداقل مربعات جزئی، دیده‌ایم، کلاس‌های *C* را می‌توان در ستون‌های دودویی *C* از متغیرهای ساختگی کدگذاری کرد و سپس به‌عنوان نتایج مدل استفاده کرد. اگرچه بحث قبلی در مورد شبکه‌های عصبی برای رگرسیون از یک پاسخ واحد استفاده می‌کرد، این مدل به راحتی می‌تواند خروجی‌های متعدد را هم برای رگرسیون ­و هم برای طبقه‌بندی انجام دهد. برای طبقه‌بندی شبکه‌های عصبی، این رویکردی است که در اینجا مورد بحث قرار می‌گیرد.

شکل [13. 3](#bookmark623) نموداری از معماری مدل را برای طبقه‌بندی نشان می‌دهد. به جای یک خروجی واحد (مانند شکل 1).  [7. 1](#bookmark343) برای رگرسیون)، لایه پایین دارای چندین گره برای هر کلاس است. توجه داشته باشید که بر خلاف شبکه‌های عصبی برای رگرسیون، یک تبدیل غیرخطی اضافی در ترکیب واحدهای پنهان استفاده می‌شود. هر کلاس با یک ترکیب خطی از واحدهای پنهان که بین صفر و یک تبدیل شده اند (معمولاً توسط یک تابع سیگموئیدی) پیش‌بینی می‌شود. با این حال، حتی اگر پیش‌بینی‌کننده‌ها بین صفر و یک هستند (به دلیل تابع سیگموئیدی اضافی)، آنها «احتمال‌مانند» نیستند زیرا با یک جمع نمی‌شوند. تبدیل softmax شرح داده شده در *بخش.*  [11. 1](#bookmark497) در اینجا استفاده می‌شود تا اطمینان حاصل شود که خروجی‌های شبکه عصبی با این محدودیت اضافی مطابقت دارند:

e f it ( x )   
f it - (x ) = Y ll e f il ( x ) '

*که* در آن *f (* *x* ) پیش‌بینی مدل کلاس اول و *i مین* نمونه است.

شبکه عصبی برای یافتن تخمین پارامترهای مناسب چه چیزی را باید بهینه کند؟ برای رگرسیون، مجموع خطاهای مجذور تمرکز بود و برای این مورد، برای رسیدگی به خروجی‌های متعدد با انباشتن خطاها در بین نمونه‌ها *و* کلاس‌ها، تغییر می‌یابد:

Cn

EE ( *y ii - f i (* *x* )) 2 *'*

جایی که *y ip* نشانگر 0/1 برای کلاس *I* است. برای طبقه بندی، این می‌تواند ­روش موثری برای تعیین مقادیر پارامتر باشد. کلاسی که بیشترین مقدار پیش‌بینی شده را دارد برای طبقه‌بندی نمونه استفاده می‌شود.

از طرف دیگر، تخمین‌های پارامتری را می‌توان یافت که می‌توانند احتمال توزیع برنولی را که مربوط به یک ­تابع دوجمله‌ای شباهت است ( معادل [12. 1 )](#bookmark21) با حجم نمونه *n* = 1:

Cn

EE *y ii'* ln &( *x* ) *•*  (13. 3)

این تابع همچنین *آنتروپی* یا آنتروپی *متقاطع را نام می‌برد* که در برخی از مدل‌های مبتنی بر درخت مورد بحث در فصل بعدی استفاده می‌شود (بخش.  [14 )](#bookmark683) . احتمال اعتبار نظری بیشتری نسبت به رویکرد خطای مربع دارد،

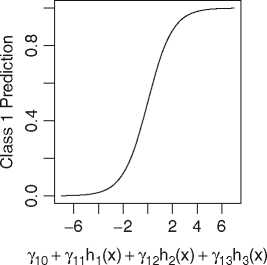
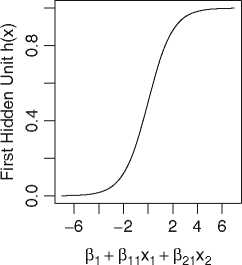
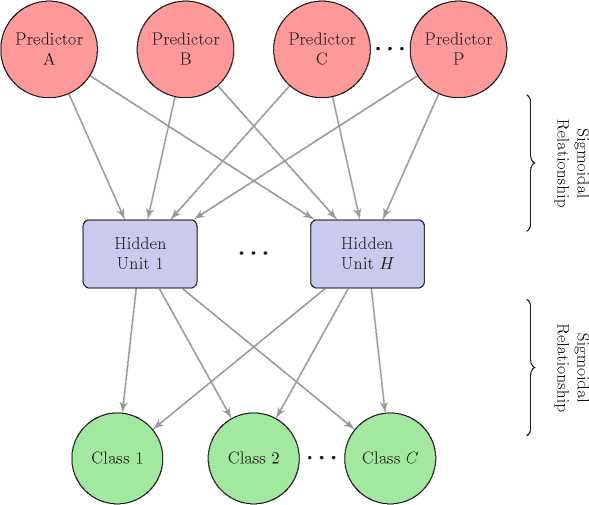
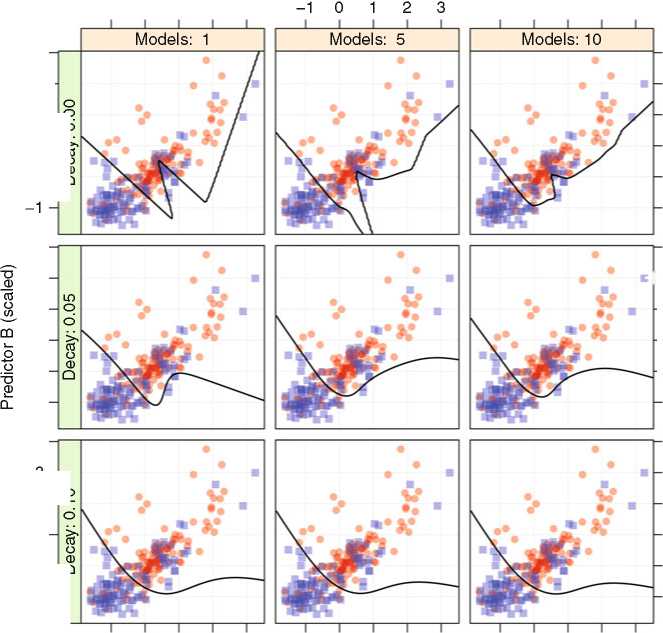


Fig. 13.3: A diagram of a neural network for classification with a single hidden layer. The hidden units are linear combinations of the predictors that have been transformed by a sigmoidal function. The output is also modeled by a sigmoidal function

اگرچه مطالعات نشان داده اند که تفاوت در عملکرد ناچیز است ( [Kline and Berardi 2005](#bookmark1020) ). با این حال، [راهب](#bookmark1011) [( 1995](#bookmark1011) ) پیشنهاد می‌کند که تابع آنتروپی باید با دقت بیشتری احتمالات کوچک را نسبت به آنهایی که توسط تابع مربع خطا ایجاد می‌شود، تخمین بزند.



Class 1 • Class 2

3 -I

-2

- 1

1-0

0 “|q

1 -

—

0

1

2

3

0

1

2

3

1

-1

-1

-1

0 - Q

Predictor A (scaled)

Fig. 13.4: Classification boundaries for neural networks with varying levels of smoothing and regularization. As weight decay and number of models increase, the boundaries become smoother

2-o

3-

2-1 —

1-3

مانند همتایان رگرسیونی خود، شبکه‌های عصبی برای طبقه‌بندی پتانسیل قابل‌توجهی برای برازش بیش از حد دارند. هنگام بهینه‌سازی مجموع مربعات خطا یا آنتروپی، کاهش وزن اندازه تخمین پارامترها را کاهش می‌دهد. این می‌تواند به مرزهای طبقه‌بندی بسیار هموارتر منجر شود. همچنین، همانطور که قبلاً بحث شد، میانگین‌گیری مدل به کاهش برازش بیش از حد کمک می‌کند. در این مورد، تخمین‌های احتمال کلاس ( *y* ^ ( *x* )) در سراسر شبکه‌ها میانگین می‌شوند و این مقادیر متوسط برای طبقه‌بندی نمونه‌ها استفاده می‌شوند.

شکل [13. 4](#bookmark624) نمونه‌هایی از مدل‌های متناسب با مقادیر مختلف پوسیدگی وزن و میانگین مدل را نشان می‌دهد. هر مدل با همان دانه تصادفی آغاز شد، از سه واحد پنهان استفاده شد و برای مجموع مجذور خطاها بهینه شد. ردیف اول مدل‌های بدون کاهش وزن، بیش برازش قابل‌توجهی را نشان می‌دهد و در این موارد، میانگین‌گیری مدل تأثیر حاشیه‌ای دارد. مقدار کم فروپاشی نشان‌داده‌شده در ردیف دوم بهبود را نشان می‌دهد (همانطور که میانگین‌گیری مدل نشان می‌دهد) اما هنوز هم زمانی که از یک شبکه استفاده می‌شود بیش از حد با داده‌های آموزشی تطبیق می‌یابد. بیشترین مقدار پوسیدگی وزن بهترین نتایج را بدون تاثیر میانگین‌گیری مدل نشان داد. برای این داده‌ها، یک مدل واحد با کاهش وزن احتمالا بهترین انتخاب است زیرا از نظر محاسباتی کم هزینه تر است.

بسیاری از جنبه‌های دیگر مدل‌های طبقه‌بندی شبکه‌های عصبی، همتایان رگرسیونی خود را منعکس می‌کنند. افزایش تعداد پیش‌بینی‌کننده‌ها یا واحدهای پنهان همچنان باعث ایجاد تعداد زیادی پارامتر در مدل می‌شود و می‌توان از همان روال‌های عددی مانند انتشار برگشتی برای تخمین این پارامترها استفاده کرد. پیش‌بینی‌کننده‌های خطی و غیر اطلاعاتی تأثیر قابل مقایسه‌ای بر عملکرد مدل خواهند داشت.

انواع مختلفی از شبکه‌های عصبی برای داده‌های گرنت مناسب بودند. اول، مدل‌های تک شبکه (یعنی بدون میانگین‌گیری مدل) با استفاده از آنتروپی برای تخمین ضرایب مدل مناسب بودند. مدل‌ها بر روی تعداد واحدها در لایه پنهان (از 1 تا 10) و همچنین میزان فروپاشی وزن ( *A* = 0، 0. 1، 1، 2) تنظیم شدند. بهترین مدل از هشت واحد پنهان با *A* = 2 استفاده کرد و دارای مساحت زیر منحنی ROC 0. 884 بود. پروفیل‌های پارامتر تنظیم مقدار قابل‌توجهی از تغییرات را نشان می‌دهند، بدون هیچ روند واضحی در سراسر پارامترهای تنظیم.

برای مقابله با این تغییرات، همان فرآیند تنظیم تکرار شد، اما 10 شبکه با داده‌ها متناسب شدند و نتایج آنها به‌طور میانگین محاسبه شد. در اینجا، بهترین مدل دارای شش واحد پنهان با *A* = 2 و دارای مساحت زیر منحنی ROC 0. 884 بود.

برای افزایش اثربخشی مدل، تبدیل‌های مختلف داده‌ها مورد ارزیابی قرار گرفت. یکی به ویژه، تبدیل نشانه فضایی، تاثیر مثبت قابل‌توجهی بر عملکرد شبکه‌های عصبی برای این داده‌ها داشت. وقتی با یک مدل شبکه منفرد ترکیب شد، سطح زیر منحنی 0. 903 بود. هنگامی که از میانگین مدل استفاده شد، سطح زیر منحنی ROC 0. 911 بود.

شکل [13. 5](#bookmark626) نمایه‌های پارامتر تنظیم را در مدل‌های مختلف به تصویر می‌کشد. هنگامی که هیچ تغییر داده‌ای استفاده نمی‌شود، میانگین‌گیری مدل عملکرد مدل‌ها را در تمام پارامترهای تنظیم افزایش می‌دهد. همچنین تأثیری بر هموار کردن تفاوت‌های بین مدل‌ها دارد. منحنی‌های پروفیل بسیار به هم نزدیکتر هستند. هنگامی که تبدیل علامت فضایی با مدل تک شبکه استفاده می‌شود، نسبت به مدل ­بدون تبدیل، پیشرفت را نشان می‌دهد. با این حال، به نظر می‌رسد عملکرد هنگام استفاده از میانگین‌گیری مدل و علامت فضایی بهینه می‌شود.

اوه

که در

O tr

کاهش وزن

0. 1 ■ 1

0

2

6

0. 80- |

0. 75-z

-0. 90

-0. 85

-0. 80

-0. 75

2

8 10

2 4

0. 90-o

03

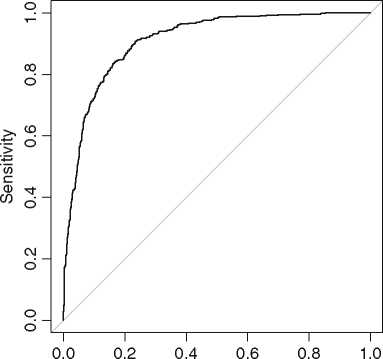
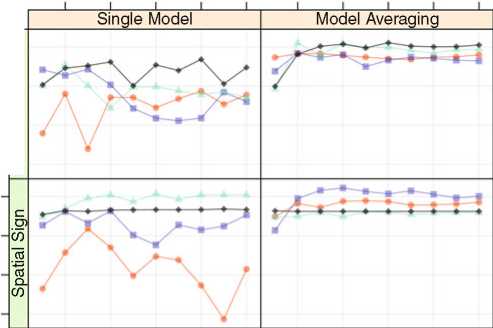
0. 85- I

46810

تعداد واحدهای پنهان

1 - خاص بودن

شکل 13. 5: *بالا* : مدل‌های موفقیت گرنت تحت چهار شرایط مختلف تنظیم شدند: با و بدون تغییر در پیش‌بینی‌کننده‌ها و با و بدون میانگین‌گیری مدل. *پایین* : منحنی ROC برای مجموعه نگهدارنده 2008 زمانی که یک شبکه میانگین مدل با تبدیل علامت فضایی استفاده می‌شود (مساحت زیر منحنی: 0. 911)



یک ماتریس پاسخ جدید از ستون‌های متغیر ساختگی باینری برای هر یک از کلاس‌های *C ایجاد کنید*

ایجاد یک مدل رگرسیون چند متغیره با استفاده از هر روشی که شیب‌ها و وقفه‌هایی را برای پیش‌بینی‌کننده‌ها یا توابع پیش‌بینی‌کننده‌ها ایجاد می‌کند (مانند رگرسیون خطی، MARS و غیره)

پس پردازش پارامترهای مدل با استفاده از تکنیک امتیازدهی بهینه

از ضرایب رگرسیون تنظیم شده به‌عنوان مقادیر متمایز استفاده کنید

الگوریتم 13. 1: الگوریتم تحلیل تفکیک انعطاف‌پذیر برای تعمیم مدل LDA [( هستی و همکاران 1994](#bookmark1018) )

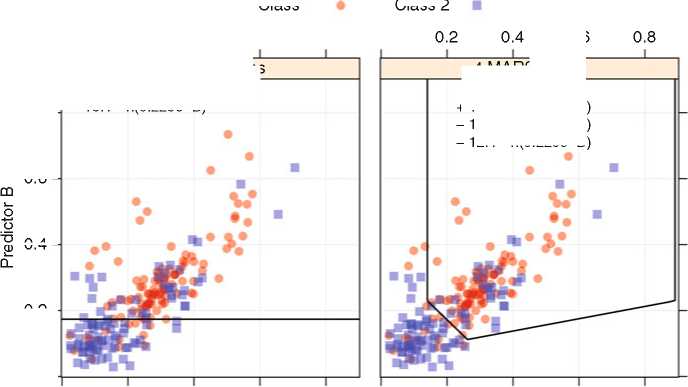
13. 3 تحلیل متمایز انعطاف پذیر

در فصل آخر، انگیزه تحلیل افتراقی خطی کلاسیک بر اساس به حداقل رساندن احتمال کل طبقه‌بندی اشتباه بود. به نظر می‌رسد که همان مدل را می‌توان به روشی کاملاً متفاوت استخراج کرد. [هستی و همکاران ( 1994](#bookmark1018) ) فرآیندی را توصیف می‌کند که در آن، برای کلاس‌های *C،* مجموعه‌ای از مدل‌های رگرسیون خطی *C* می‌توانند با شاخص‌های کلاس دودویی مناسب باشند و نشان دهند که ضرایب رگرسیون از این مدل‌ها می‌توانند پس پردازش شوند تا ضرایب متمایز را استخراج کنند (الگوریتم را ببینید).  [13. 1 )](#bookmark629) . این اجازه می‌دهد تا ایده تحلیل متمایز خطی به روش‌های مختلفی گسترش یابد. اول، بسیاری از مدل‌های موجود در Chaps. 6 و [7 ،](#bookmark342) مانند ریج، رگرسیون برآمدگی یا MARS، می‌تواند برای ایجاد متغیرهای متمایز گسترش یابد. به‌عنوان مثال، MARS می‌تواند برای ایجاد مجموعه‌ای از توابع لولا استفاده شود که منجر به توابع متمایز می‌شود که ترکیبی غیرخطی از پیش‌بینی‌کننده‌های اصلی هستند. به‌عنوان مثال دیگر، ریج می‌تواند توابع متمایز را با انتخاب ویژگی ایجاد کند. این چارچوب مفهومی به‌عنوان *تحلیل متمایز انعطاف‌پذیر (*FDA) نامیده می‌شود.

ما می‌توانیم ماهیت غیرخطی ریتم الگوی تفکیک انعطاف‌پذیر را ­با استفاده از MARS با داده‌های مثال در شکل نشان دهیم.  [4. 1](#bookmark196) (ص.  [63 )](#bookmark196) . به یاد بیاورید که MARS دو پارامتر تنظیم دارد: تعداد عبارت‌های حفظ شده و ­درجه پیش‌بینی‌کننده‌های درگیر در توابع لولا. اگر از یک مدل افزایشی (یعنی یک مدل درجه اول) استفاده کنیم، حداکثر تعداد عبارت‌های حفظ شده را به 2 محدود کنیم و یک پاسخ باینری از عضویت کلاس داشته باشیم، آنگاه تابع تفکیک ­کننده است.

*D (* *A, B* ) = 0. 911 *-* 19. 1 *xh (*0. 2295 *- B* )

در این معادله، *h (* *•* ) تابع لولا است که در معادله توضیح داده شده است.  [7. 1](#bookmark349) در صفحه [146](#bookmark349) . اگر تابع تفکیک بزرگتر از صفر باشد، نمونه کلاس اول پیش‌بینی می‌شود. در این مدل، معادله پیش‌بینی فقط از یک متغیر و پانل سمت چپ در شکل 1 استفاده می‌کند.  [13. 6](#bookmark630) مرزهای کلاس حاصل را نشان می‌دهد. مرز کلاس یک خط افقی است زیرا پیش‌بینی *B* تنها پیش‌بینی در تقسیم است.



Class 1

Class 2

0.6

0.8-

0.6-

0.4 -

0.2-1

0.2

0.4

0.6

0.8

2 MARS Terms

0.911 - 19.1 \* h(0.2295-B)

4 MARS Terms

-0.242

+ 11.6 \* h(A-0.1322)

13.9 \* h(A-0.2621)

12.1 \* h(0.2295-B)

پیش‌بینی A

شکل 13. 6: مرزهای طبقه‌بندی برای دو مدل FDA با سطوح مختلف ­پیچیده

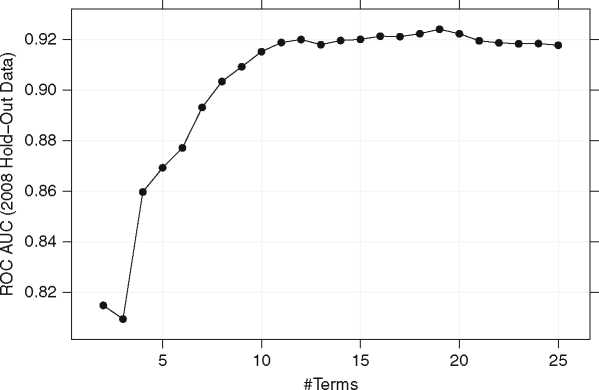
زمانی که MARS به شدت تحت فشار قرار می‌گیرد، اثربخشی FDA آشکار نیست. اگر حداکثر تعداد عبارت‌های حفظ شده به 4 کاهش یابد، معادله تمایز تخمین زده می‌شود

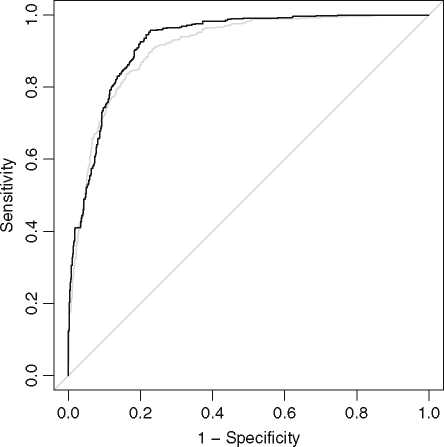
*D (* *A, B* ) = *-* 0. 242

+ 11. 6 *xh ( A -* 0. 1322 *) -* 13. 9 *xh ( A -* 0. 2621 *) -* 12. 1 *xh (* 0. 2295 *- B* ).

این مدل FDA از هر دو پیش‌بینی استفاده می‌کند و مرز کلاس آن در پانل سمت راست شکل نشان داده شده است.  [13. 6](#bookmark630) . به یاد بیاورید که تابع لولای MARS *h* یک طرف نقطه شکست را صفر می‌کند. به همین دلیل، توابع لولا مناطق خاصی از داده‌ها را جدا می‌کنند. به‌عنوان مثال، اگر *A<* 0. 1322 و *ب>* 0. 2295، هیچ یک از توابع لولا بر پیش‌بینی تأثیر نمی‌گذارد و قطع منفی در مدل نشان می‌دهد که تمام نقاط این ناحیه با کلاس دوم مطابقت دارند. با این حال، اگر *A >* 0. 2621 و *B <* 0. 2295، پیش‌بینی تابعی از هر سه تابع لولا است. اساساً، ویژگی‌های MARS نواحی چندبعدی چندبعدی فضای پیش‌بینی را جدا می‌کنند و یک کلاس مشترک را در این مناطق پیش‌بینی می‌کنند.

یک مدل FDA برای مدل درخواست گرنت تنظیم و آموزش داده شد. توابع لولای درجه یک MARS در جایی که تعداد ­عبارت‌های حفظ شده از 2 تا 25 متغیر بود، ارزیابی شدند.





شکل 13. 7: *بالا* : مشخصات تنظیم پارامتر برای مدل FDA. *پایین* : منحنی ROC FDA (مساحت زیر منحنی: 0. 924) در رابطه با منحنی مدل شبکه عصبی قبلی (به *رنگ خاکستری* ) نشان داده شده است.

ترم‌ها افزایش می‌یابد و حدود 15 ترم فلات می‌شود (شکل 2 را ببینید.  [13. 7 )](#bookmark631) . مقدار بهینه عددی ­19 بود، اگرچه به وضوح انعطاف پذیری در این پارامتر وجود دارد. برای این مدل، سطح زیر منحنی ROC برای داده‌های سال 2008 0. 924 با حساسیت 82. 5 درصد و ویژگی

86. 4 درصد اگرچه مدل FDA شامل 19 عبارت بود، 14 پیش‌بینی منحصر به فرد (از 1070 مورد ممکن) استفاده شد. همچنین، 9 عبارت مدل، توابع خطی ساده پیش‌بینی‌کننده‌های دسته‌بندی باینری بودند. معادله تمایز برای مدل است

*D (* *x* ) = 0. 85

0. 53 *x h (*1 *-* تعداد محققین ارشد)

0. 11 *x h (*تعداد گرنت موفق توسط محققین ارشد *-* 1)

1. 1 *x h (*1 *-* تعداد گرنت موفق توسط محققین ارشد)

0. 23 *x h (*تعداد گرنت ناموفق توسط محققین ارشد *-* 1) +1. 4 *x h (*1 *-* تعداد گرنت ناموفق توسط محققین ارشد) + 0. 18 *x h (*تعداد گرنت ناموفق توسط محققین ارشد *-* 4) *-* 0. 035 *x h (*8 *-* تعداد مقالات ژورنال A توسط همه محققین) *-* 0. 79 *x* حامی کد 24D

1 *عدد* حامی کد 59C

0. 98 *x* حامی کد 62B

1. 4 *x* حامی کد 6B

1. 2 *x* sp onsor ناشناخته

0. 34 *x* ارزش قرارداد باند B

1. *باند ارزش قرارداد* 5 *x* نامشخص

*-* 0. 34 *x* کد دسته‌بندی گرنت 30B

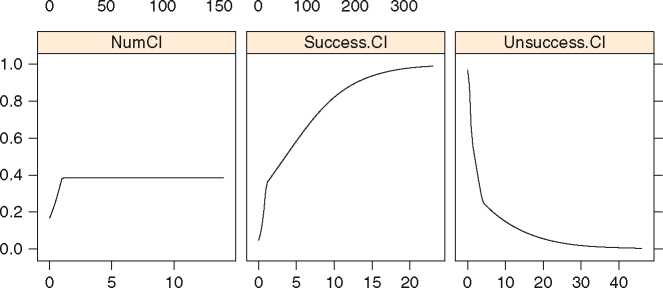
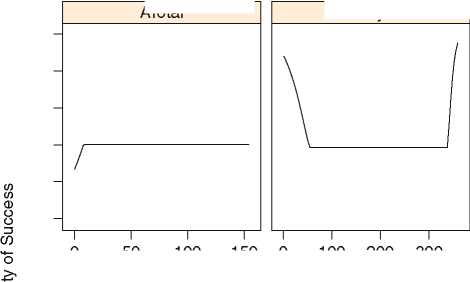
+0. 3 *بار* ارسال روز شنبه

+ 0. 022 *x h (*54 *—* روز عددی سال)

+ 0. 076 *x h (*روز عددی سال *-* 338).

از این معادله، تأثیر دقیق پیش‌بینی‌کننده‌ها بر مدل را می‌توان روشن کرد. به‌عنوان مثال، با افزایش تعداد محققین ارشد از صفر به یک، احتمال موفقیت آمیز گرنت افزایش می‌یابد. از آنجایی که عملکرد لولای مخالف حذف شد، داشتن بیش از یک محقق ارشد بر مدل تأثیر نمی‌گذارد. ­همچنین، احتمال موفقیت با تعداد ­گرنت موفق توسط محققین ارشد افزایش می‌یابد و با تعداد گرنت ناموفق توسط محققین اصلی کاهش می‌یابد. این نتیجه مشابه با مدل‌های قبلی است. برای روز سال، احتمال موفقیت آمیز بودن گرنت با ادامه سال کاهش می‌یابد و تا اواخر سال که احتمال موفقیت افزایش می‌یابد، تأثیری بر مدل ندارد.

تابع تفکیک نشان داده شده در بالا می‌تواند به علاوه برای تولید تخمین‌های احتمال کلاس تبدیل شود. از نظر بصری، روند احتمال برای پیش‌بینی‌کننده‌های پیوسته در شکل 1 نشان داده شده است.  [13. 8](#bookmark633) . به یاد بیاورید که از آنجایی که یک مدل افزایشی استفاده شده است، نمایه احتمال برای هر متغیر می‌تواند مستقل از بقیه در نظر گرفته شود. در اینجا، شرایط برای تعداد پژوهشگران ارشد ­و تعداد انتشارات در مجلات سطح *A* فقط تا یک نقطه بر پیش‌بینی تأثیر می‌گذارد. این نتیجه الگوریتم هرس elimi- است.



ATotal

Day

- 1.0

-0.8

-0.6

-0.4

-0.2

ra

-0.0

Fig. 13.8: Probability profiles for each of the continuous predictors used in the additive FDA model

نامگذاری یکی از جفت‌های بازتابنده هر پیش‌بینی. نمایه روز سال دارای دو عبارت است که از دو جفت منعکس شده متفاوت باقی می‌ماند. در نتیجه، این پیش‌بینی تنها در دوره‌های اولیه و اواخر سال بر مدل تأثیر می‌گذارد. در فصل آخر، شواهد خوبی وجود داشت مبنی بر اینکه این پیش‌بینی یک رابطه غیرخطی با نتیجه دارد که با افزودن یک تابع درجه دوم پیش‌بینی تقریبی شد. در اینجا، FDA نیز سعی می‌کند همین رابطه را تقریبی کند. یک پیش‌بینی، تعداد گرنت ناموفق توسط محققین ارشد، دارای شرایط متعددی در مدل است که در نمایه احتمال هموارتر منعکس می‌شود. از میان اصطلاحات دودویی، پیش‌بینی‌کننده‌های باند ارزش قرارداد B، باند ارزش قرارداد ناشناخته، کد دسته گرنت 30B، ­کد حامی 24D، کد حامی 59C، کد حامی 62B و کد حامی مالی 6B تأثیر مثبتی بر احتمال موفقیت داشتند. شرایط روز ارسال شنبه و حامی ناشناس با کاهش میزان موفقیت همراه بود.

بسته‌بندی مدل FDA را مجبور می‌کند تا روابط صاف‌تری بین پیش‌بینی‌کننده‌ها و نتیجه ایجاد کند. مدل‌های MARS ­پیش‌بینی‌کننده‌های نسبتاً ناپایداری هستند زیرا از جستجوی جامع داده‌ها و تقسیم‌ها استفاده می‌کنند.

بر اساس نقاط داده خاص در مجموعه آموزشی هستند. [[46]](#footnote-46) بسته‌بندی مدل FDA باعث می‌شود که تقسیم‌بندی‌های بیشتری برای پیش‌بینی‌کننده‌های مهم اضافه شود که منجر به تقریب بهتر می‌شود. با این حال، تجربه ما این است که بسته‌بندی مدل‌های MARS یا FDA تأثیری حاشیه‌ای بر عملکرد مدل دارد و افزایش تعداد عبارت‌ها، تفسیر معادله تمایز را کاهش می‌دهد (شبیه به روند نشان داده شده در شکل 2).  [8. 16 )](#bookmark415) .

از آنجایی که بسیاری از پیش‌بینی‌کننده‌ها در مدل FDA در مقیاس‌های مختلف هستند، استفاده از تابع متمایز برای کشف اینکه کدام متغیرها بیشترین تأثیر را بر نتیجه دارند، دشوار است. همان روش اندازه‌گیری اهمیت متغیر که در بخش.  [7. 2](#bookmark348) را می‌توان در اینجا به کار گرفت. پنج ­پیش‌بینی مهم به ترتیب عبارتند از: باند ارزش قرارداد ناشناخته، تعداد گرنت ناموفق توسط محققین ارشد، تعداد گرنت موفق توسط محققین ارشد، حامی ناشناخته و روز عددی سال.

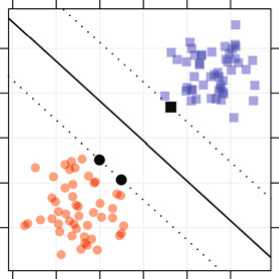
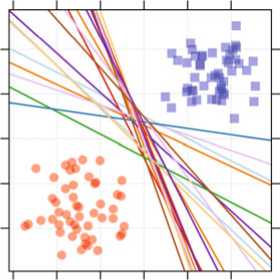
به‌عنوان جایگزینی برای استفاده از MARS در چارچوب FDA، [میلبورو](#bookmark1022) [( 2012](#bookmark1022) ) یک رویکرد دو مرحله‌ای با رگرسیون لجستیک را هنگامی که دو کلاس وجود دارد، توصیف می‌کند. در اینجا، یک مدل MARS اولیه برای پیش‌بینی متغیر پاسخ ساختگی باینری (یعنی دو مرحله اول در الگوریتم) ایجاد می‌شود.  [13. 1 )](#bookmark629) . پس از این، یک مدل رگرسیون لجستیک با ویژگی‌های MARS تولید شده توسط مدل متغیر ساختگی اصلی ایجاد می‌شود. تجربیات اولیه ما با این رویکرد این است که نتایج بسیار شبیه به مدل FDA را به همراه دارد.

ماشین‌های بردار پشتیبانی

ماشین‌های بردار پشتیبان دسته‌ای از مدل‌های آماری هستند که اولین بار در اواسط دهه 1960 توسط ولادیمیر واپنیک توسعه یافتند. در سال‌های بعد، این مدل به‌طور قابل‌توجهی به یکی از انعطاف‌پذیرترین و مؤثرترین ابزارهای یادگیری ماشین در دسترس تبدیل شد و [واپنیک](#bookmark1026) [( 2010](#bookmark1026) ) یک درمان جامع ارائه می‌دهد. نسخه رگرسیون این مدل‌ها قبلا در بخش بحث شده است.  [7. 3 ،](#bookmark356) که بسط مدل از توسعه اولیه آن در ­تنظیمات طبقه‌بندی بود. در اینجا ما مفاهیم مشابه SVM را برای رگرسیون و چیدمان مورد برای طبقه‌بندی لمس می‌کنیم.

مسأله رشک برانگیز نشان داده شده در پانل سمت چپ شکل را در نظر بگیرید.  [13. 9](#bookmark636) که در آن از دو متغیر برای پیش‌بینی دو دسته از نمونه‌ها استفاده می‌شود که کاملاً قابل تفکیک هستند. همانطور که در سمت چپ نشان داده شده است، تعداد زیادی (در واقع بی نهایت) از مرزهای خطی وجود دارد که این داده‌ها را کاملاً طبقه‌بندی می‌کند. با توجه به این موضوع، چگونه یک مرز کلاس مناسب را انتخاب کنیم؟ بسیاری از معیارهای عملکرد، مانند دقت، ناکافی هستند زیرا همه منحنی‌ها معادل تلقی می‌شوند. معیار مناسب‌تری برای قضاوت در مورد کارایی یک مدل چیست؟

Vapnik یک معیار جایگزین به نام ***حاشیه*** *تعریف کرد.* به زبان ساده، **حاشیه فاصله بین مرز طبقه‌بندی و نزدیکترین است**



—

4

2

0

2

4

-4

-2

0

2

Predictor A

Fig. 13.9: *Left*: A data set with completely separable classes. An infinite number of linear class boundaries would produce zero errors. *Right*: The class boundary associated with the linear maximum margin classifier. The solid black points indicate the support vectors

4

2

0

2

4

-4

-2

0

2

4

Predictor A

نقطه تنظیم آموزش به‌عنوان مثال، پانل سمت راست شکل.  [13. 9](#bookmark636) یک مرز طبقه‌بندی ممکن را به‌عنوان یک خط ثابت نشان می‌دهد. خطوط چین در دو طرف مرز در حداکثر فاصله از خط تا نزدیکترین داده‌های مجموعه آموزشی (با فاصله مساوی از خط مرز) قرار دارند. در این مثال، سه نقطه داده به همان اندازه نزدیک به مرز طبقه‌بندی هستند و با نمادهای سیاه و سفید برجسته شده اند. حاشیه تعریف شده توسط این نقاط داده را می‌توان کمی‌سازی کرد و برای ارزیابی مدل‌های ممکن استفاده کرد. در اصطلاح SVM، شیب و وقفه مرزی که بافر بین مرز و داده‌ها را به حداکثر می‌رساند، به‌عنوان طبقه‌بندی کننده حاشیه حداکثر شناخته می‌شود.

بیایید تعدادی از ساختارهای ریاضی SVM را در چارچوب یک مثال ساده به منظور درک بهتر عملکرد درونی روش بررسی کنیم. فرض کنید یک مسأله دو کلاسه داریم و نمونه‌های کلاس #1 را با مقدار 1 و نمونه‌های کلاس #2 را با *-* 1 کدگذاری می‌کنیم. همچنین، اجازه دهید بردار x *i* حاوی داده‌های پیش‌بینی برای یک نمونه مجموعه آموزشی باشد. طبقه‌بندی‌کننده حاشیه حداکثر یک مقدار تصمیم *D (* x ) ایجاد می‌کند که نمونه‌ها را طوری طبقه‌بندی می‌کند که اگر *D (* x ) *>* 0 باشد، نمونه‌ای را پیش‌بینی می‌کنیم که کلاس #1 باشد، در غیر این صورت کلاس #2. برای یک نمونه ناشناخته u، معادله تصمیم را می‌توان به شکلی مشابه به‌عنوان یک تابع متمایز خطی نوشت که بر حسب یک مقطع و شیب به‌عنوان پارامتر تعیین می‌شود.

*D (* u ) = *P* 0 + *$'* u

پ

= *P* 0 + *P j u j.*

j = 1

توجه داشته باشید که این معادله از دیدگاه پیش‌بینی‌کننده‌ها کار می‌کند. این معادله را می‌توان به گونه‌ای تبدیل کرد که حداکثر طبقه‌بندی کننده حاشیه را بتوان بر حسب هر نقطه داده در نمونه نوشت. این معادله ­را به

*P   
D (* u ) = *3 0* + £ *13 j u j   
j* = 1

n

*= 3* 0 + *y i a i X i* u (13. 4)

i = 1

با *i > 0 ( شبیه* معادله [7. 2 )](#bookmark9) . معلوم می‌شود که در حالت کاملاً قابل تفکیک، پارامترهای *a* برای همه نمونه‌هایی که در حاشیه نیستند دقیقاً صفر است. برعکس، مجموعه مقادیر غیرصفر *a* نقاطی هستند که روی مرز حاشیه قرار می‌گیرند (یعنی نقاط سیاه جامد در شکل 1).  [13. 9 )](#bookmark636) . به همین دلیل، معادله پیش‌بینی تنها تابعی از زیرمجموعه‌ای از نقاط مجموعه آموزشی است و به آنها *بردارهای پشتیبان گفته می‌شود.* جالب توجه است که تابع پیش‌بینی تنها تابعی از نمونه‌های مجموعه آموزشی است که نزدیک‌ترین به مرز هستند و با کمترین میزان اطمینان پیش‌بینی می‌شوند. [[47]](#footnote-47) از آنجایی که معادله پیش‌بینی صرفاً توسط این نقاط داده *پشتیبانی می‌شود،* طبقه‌بندی‌کننده حاشیه حداکثر معمولاً *ماشین بردار پشتیبان نامیده می‌شود.*

در اولین معاینه، معادله [13. 4](#bookmark33) ممکن است تا حدودی محرمانه به نظر برسد. با این حال، می‌تواند روشن کند که چگونه ماشین‌های بردار پشتیبان، نمونه‌های جدید را طبقه‌بندی می‌کنند. شکل را در نظر بگیرید.  [13. 10](#bookmark637) که در آن یک نمونه جدید که به صورت دایره خاکستری جامد نشان داده شده است، ­توسط مدل پیش‌بینی می‌شود. فاصله بین هر یک از بردارهای پشتیبان و نمونه جدید به صورت خطوط نقطه چین خاکستری است.

برای این داده‌ها، سه بردار پشتیبان وجود دارد و بنابراین حاوی تنها اطلاعات لازم برای طبقه‌بندی نمونه جدید است. گوشت معادله [13. 4](#bookmark33) جمع حاصل ضرب: علامت کلاس، پارامتر مدل و حاصل ضرب نقطه بین نمونه جدید و مقادیر پیش‌بینی بردار پشتیبان است. جدول زیر اجزای این مجموع را به تفکیک هر یک از سه بردار پشتیبان نشان می‌دهد:

|  |  |  |  |
| --- | --- | --- | --- |
| کلاس واقعی | محصول نقطه ای | *y من یک من* | تولید - محصول |
| SV 1 کلاس 2 | *-* 2. 4 | *-* 1 1. 00 | 2. 40 |
| SV 2 کلاس 1 | 5. 1 | 1 0. 34 | 1. 72 |
| SV 3 کلاس 1 | 1. 2 | 1 0. 66 | 0. 79 |

ضرب نقطه‌ای x *i* u را می‌توان به صورت ضرب فاصله x *i* از مبدأ، فاصله u از مبدأ و کسینوس زاویه بین x *i* و u نوشت [( دیلون و گلدشتاین 1984](#bookmark1014) ).



4

2

0

2

4

-4

-2

0

2

4

Predictor A

Fig. 13.10: Prediction of a new sample using a support vector machine. The final value of the decision equation is *D*(*u*) = 0*.*583. The grey lines indicate the distance of the new sample to the support vectors

بر اساس تخمین پارامتر *a i،* اولین بردار پشتیبان بیشترین تأثیر را روی معادله پیش‌بینی دارد (همه چیزهای دیگر برابر هستند) و دارای شیب منفی است. برای نمونه جدید ما، حاصلضرب نقطه‌ای منفی است، بنابراین سهم کل این نقطه مثبت است و پیش‌بینی را به سمت کلاس اول سوق می‌دهد (یعنی مقدار مثبت تابع تصمیم *D (* *u* )). دو بردار پشتیبان باقی مانده دارای محصولات نقطه مثبت و یک ­محصول بیش از همه هستند که مقدار تابع تصمیم را برای این نمونه افزایش می‌دهد. برای این مدل، عرض از مبدأ *-* 4 *است.* 372; بنابراین *D (* *u* ) برای نمونه جدید 0. 583 است. از آنجایی که این مقدار بزرگتر از صفر است، نمونه جدید بیشترین ارتباط را با کلاس اول دارد.

وقتی کلاس‌ها کاملاً قابل تفکیک نیستند چه چیزی وجود [دارد](#bookmark1014) ؟ Cortes و Vapnik [( 1995](#bookmark1014) ) برای تطبیق با این وضعیت، پسوندهایی را برای طبقه‌بندی‌کننده حداکثر حاشیه اولیه توسعه می‌دهند. فرمول‌بندی آنها هزینه‌ای را روی مجموع نقاط مجموعه آموزشی که در مرز یا سمت اشتباه مرز قرار دارند، می‌گذارد. هنگام تعیین برآورد مقادیر *a،* زمانی که نقاط داده در سمت اشتباه مرز کلاس یا داخل حاشیه قرار دارند، حاشیه جریمه می‌شود. ارزش هزینه یک پارامتر تنظیم برای مدل خواهد بود و مکانیزم اولیه برای کنترل پیچیدگی مرز است. به‌عنوان مثال، با افزایش هزینه خطاها، مرز طبقه‌بندی تغییر می‌کند و خود را منحرف می‌کند تا جایی که ممکن است بسیاری از مجموعه‌های آموزشی را به درستی طبقه‌بندی کند. شکل [4. 2](#bookmark196) در فصل [4](#bookmark192) این را نشان داد. پانل در سمت راست این شکل از ارزش هزینه نامناسب بالایی استفاده می‌کند که منجر به نصب بیش از حد شدید می‌شود.

بازتاب نظرات در بخش.  [7. 3](#bookmark356) ، اکثر مدل‌های منظم‌سازی مورد ­بحث در این کتاب، جریمه‌هایی را به ضرایب اضافه می‌کنند تا از برازش بیش از حد جلوگیری شود. جریمه‌های بزرگ، مشابه هزینه‌ها، محدودیت‌هایی را بر پیچیدگی مدل تحمیل می‌کند. برای ماشین‌های بردار پشتیبان، از مقادیر هزینه برای جریمه کردن تعداد *خطاها استفاده می‌شود.* در نتیجه، مقادیر هزینه بزرگ‌تر به جای محدود کردن مدل، پیچیدگی بیشتری را القا می‌کند.

تا کنون، ما مرزهای طبقه‌بندی خطی را برای این مدل‌ها در نظر گرفته ایم. در معادله [13. 4 ،](#bookmark33) به محصول نقطه‌ای x *i* u توجه کنید. از آنجایی که پیش‌بینی‌کننده‌ها به صورت خطی وارد این معادله می‌شوند، مرز تصمیم به ترتیب خطی است. [بوزر و همکاران ( 1992](#bookmark1011) ) ماهیت خطی مدل را ­با جایگزینی تابع هسته به جای ضربدر خطی ساده به مرزهای طبقه‌بندی غیرخطی گسترش داد:

n

*D (* u ) = *P* 0 + *^y i a i x i* u   
*i* = 1

n

= *P* 0 + *^y* i *a i K (* xi، *u* )، *i   
=* 1

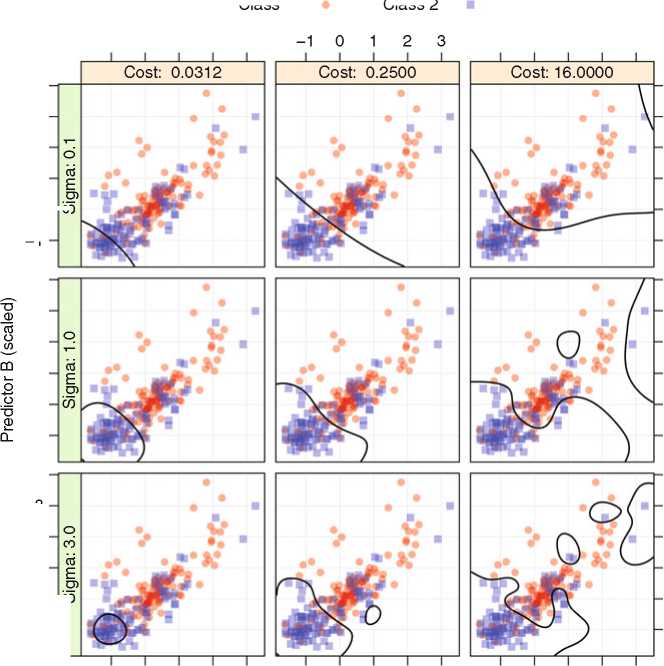
که در آن *K (* *•، •* ) یک *تابع هسته* از دو بردار است. برای حالت خطی، تابع هسته همان حاصلضرب داخلی x *i* u است. با این حال، همانطور که در SVM‌های رگرسیونی، سایر تبدیل‌های غیرخطی را می‌توان اعمال کرد، از جمله:

چند جمله‌ای = ( *مقیاس (* x *'* u ) + 1) *درجه*

تابع پایه شعاعی = exp( *— ct||* x *—* u *|* |2)   
مماس هذلولی = tanh ( *مقیاس (* x'u *) +* 1).

توجه داشته باشید که به دلیل حاصلضرب نقطه‌ای، داده‌های پیش‌بینی باید قبل از برازش در مرکز و مقیاس‌بندی شوند تا ویژگی‌هایی که مقادیر آن‌ها از نظر بزرگی بزرگ هستند بر محاسبات تسلط نداشته باشند.

*ترفند هسته* به مدل SVM اجازه می‌دهد تا مرزهای تصمیم‌گیری بسیار انعطاف پذیری تولید کند. انتخاب پارامترهای تابع هسته و ارزش هزینه پیچیدگی را کنترل می‌کند و باید به‌طور مناسب تنظیم شود تا مدل بیش از حد با داده‌های آموزشی مطابقت نداشته باشد. شکل [13. 11](#bookmark639) نمونه‌هایی از ­مرزهای طبقه‌بندی تولید شده توسط چندین مدل با استفاده از ترکیب‌های مختلف مقادیر پارامتر هزینه و تنظیم را نشان می‌دهد. هنگامی که ارزش هزینه کم است، مدل‌ها به وضوح از داده‌ها کمتر هستند. برعکس، زمانی که هزینه نسبتاً بالا باشد (مثلاً مقدار 16)، مدل می‌تواند داده‌ها را بیش برازش دهد، به خصوص اگر پارامتر هسته مقدار زیادی داشته باشد. استفاده از نمونه‌گیری مجدد برای یافتن تخمین‌های مناسب از این پارامترها منجر به یافتن تعادل معقولی بین برازش کم و بیش از حد می‌شود. بخش [4. 6](#bookmark6) از ماشین بردار پشتیبان تابع پایه شعاعی به‌عنوان مثالی برای تنظیم مدل استفاده کرد.



Class 1

Class 2

3

2

1

0 -lw

3

2

1

0

3

2

1

0 -|w

1 -

—

0

1

2

3

0

1

2

3

-1

-1

-1

-1

Predictor A (scaled)

Fig. 13.11: Classification boundaries for nine radial basis function support vector machine models varied over the cost parameter and the kernel param­eter (*a*)

ماشین‌های بردار پشتیبان در دسته کلی‌تری از روش‌ها*ی هسته* قرار می‌گیرند و این یک حوزه تحقیقاتی بسیار فعال برای مدتی بوده است. در اینجا، ما توسعه‌های مدل اصلی را مورد بحث قرار داده‌ایم تا نمونه‌های طبقه‌بندی‌شده اشتباه و مرزهای کلاس غیرخطی را مجاز کنیم. هنوز برنامه‌های افزودنی بیشتری ­برای ماشین‌های بردار پشتیبان، مانند مدیریت بیش از دو کلاس، توسعه داده شده است [( Hsu and Lin 2002](#bookmark1019) ; [Duan and Keerthi 2005](#bookmark1015) ). همچنین، انگیزه اصلی مدل، ایجاد یک مرز تصمیم‌گیری سخت به ­منظور طبقه‌بندی نمونه‌ها، بر خلاف تخمین احتمالات کلاس است. با این حال، [پلات ( 2000](#bookmark1023) ) روش‌های پس پردازش خروجی مدل SVM را برای تخمین احتمالات کلاس توصیف می‌کند. نسخه‌های جایگزینی از مدل ماشین بردار پشتیبان نیز وجود دارد، مانند ماشین‌های بردار پشتیبان حداقل مربعات [( Suykens and Vandewalle 1999](#bookmark1026) )، ماشین‌های بردار مرتبط [( Tiping 2001](#bookmark1026) ) و ماشین‌های بردار وارداتی [( Zhu and Hastie 2005](#bookmark1028) ).

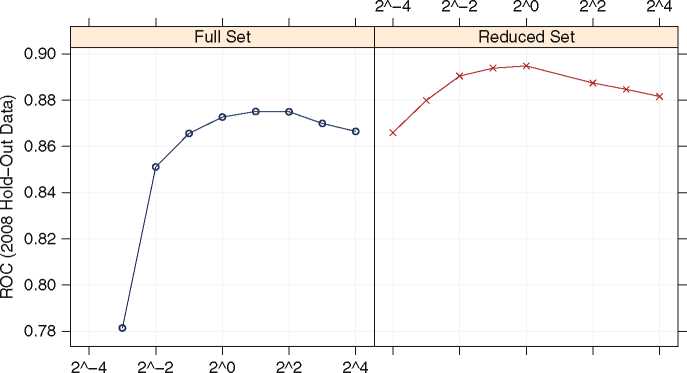
هسته‌های تخصصی نیز توسعه یافته اند. به‌عنوان مثال، برنامه QSAR ­که در بخش بحث شده است.  [6. 1](#bookmark281) و در سرتاسر فصل‌های رگرسیون از توصیف کننده‌های شیمیایی به‌عنوان پیش‌بینی استفاده شده است. شکل [6. 1](#bookmark282) فرمول شیمیایی آسپرین را نشان می‌دهد. به‌جای استخراج توصیف‌گرها از فرمول مولکولی، for ­mula را می‌توان به یک نمودار (یا شبکه) تبدیل کرد. یک کلاس تخصصی از توابع هسته، به نام *هسته‌های گراف،* می‌توانند مستقیماً محتوای فرمول شیمیایی را بدون استخراج متغیرهای توصیفگر به مدل مرتبط کنند [( Mahe et al. 2005](#bookmark1021) ; [ماهه و ورت 2009](#bookmark1021) ). به‌طور مشابه، هسته‌های مختلفی وجود دارد که می‌توانند در مسائل متن کاوی استفاده شوند. رویکرد "کیف کلمات" مجموعه‌ای از متن را با محاسبه فراوانی کلمات خاص خلاصه می‌کند. این شمارش‌ها به‌عنوان متغیرهای پیش‌بینی در مدل‌های طبقه‌بندی در نظر گرفته می‌شوند. چند مسأله در این رویکرد وجود دارد. اول، بار محاسباتی اضافی حاصل از استخراج متغیرهای پیش‌بینی می‌تواند مالیات باشد. ثانیاً، این رویکرد مبتنی بر اصطلاح، ترتیب متن را در نظر نمی‌گیرد. به‌عنوان مثال، متن «میراندا خرس را خورد» و «خرس میراندا را خورد» امتیاز یکسانی در مدل کیسه کلمات دارند اما معانی بسیار متفاوتی دارند. *هسته‌های رشته* [( Lodhi et al. 2002](#bookmark1021) ; [Cancedda و همکاران 2003](#bookmark1012) ) می‌تواند به‌طور مستقیم از کل متن یک سند استفاده ­کند و پتانسیل بیشتری برای یافتن روابط مهم نسبت به رویکرد کیسه‌ای از کلمات دارد.

برای داده‌های گرنت، چندین رویکرد برای استفاده از SVM وجود دارد. ما ­هسته تابع پایه شعاعی و همچنین هسته چند جمله‌ای ( ­که به صورت خطی یا درجه دوم به نظر می‌رسد) را ارزیابی کردیم. همچنین، هر دو مجموعه پیش‌بینی کامل و کاهش یافته مورد ارزیابی قرار گرفتند. همانطور که در فصل نشان داده خواهد شد.  [19 ،](#bookmark864) ماشین‌های بردار پشتیبان می‌توانند با گنجاندن پیش‌بینی‌کننده‌های غیر اطلاعاتی در مدل تأثیر منفی بگذارند.

برای هسته تابع پایه شعاعی، رویکرد تحلیلی برای ­تعیین پارامتر تابع پایه شعاعی ارزیابی شد. برای مجموعه کامل پیش‌بینی‌ها، برآورد *a* = 0 *بود.* 000559 و برای مجموعه کاهش یافته، مقدار *a* = 0 محاسبه شد. 00226. با این حال، این مدل‌ها عملکرد خوبی از خود نشان ندادند، بنابراین این پارامتر بر روی مقادیری که کوچکتر از تخمین‌های تحلیلی بودند، تغییر کرد. شکل [13. 12](#bookmark643) نتایج این مدل‌ها را نشان می‌دهد. مجموعه پیش‌بینی کوچکتر نتایج بهتری نسبت به مجموعه جامع‌تر به همراه دارد، با سطح بهینه زیر منحنی ROC 0. 895، حساسیت 84٪ و ویژگی 80. 4٪. همچنین، برای مجموعه کاهش‌یافته، مقادیر کوچک‌تر *a* نتایج بهتری ایجاد می‌کند، اگرچه مقادیر زیر 0. 001167 برازش مدل را بهبود نمی‌بخشد.

برای مدل‌های چند جمله‌ای، مقدار مناسبی از آزمون و خطا برای تعیین مقادیر مناسب برای ضریب مقیاس‌پذیری این هسته استفاده شد. مقادیر نامناسب منجر به مسائل عددی برای مدل‌ها می‌شود و مقادیر امکان‌پذیر این پارامتر بستگی به درجه چند جمله‌ای و پارامتر هزینه دارد. شکل [13. 13](#bookmark644) نتایج را برای مجموعه Holdout نشان می‌دهد. مدل‌هایی که با مجموعه پیش‌بینی‌کننده‌های کاهش‌یافته ساخته شده‌اند، به‌طور یکنواخت بهتر از مدل‌هایی که از مجموعه کامل استفاده می‌کنند، عمل می‌کنند. همچنین عملکرد بهینه برای مدل‌های خطی و درجه دوم حدود بود

سیگما = 0. 00023857 یا سیگما = 0. 00116699 x



هزینه

شکل 13. 12: مشخصات پارامتر تنظیم مدل SVM تابع پایه شعاعی برای داده‌های گرنت

همان این نشان می‌دهد که مدل‌ها عمدتاً از روابط خطی در داده‌ها استفاده می‌کنند. با توجه به اینکه بسیاری از پیش‌بینی‌کننده‌ها باینری هستند، این منطقی است. از بین این مدل‌ها بهترین سطح زیر منحنی ROC 0. 898 بوده است.

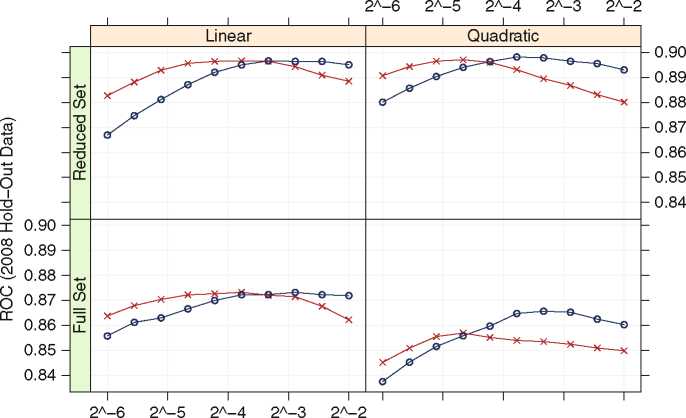
به‌طور کلی، مدل‌های ماشین بردار پشتیبان ­در مقایسه با مدل‌های ایجاد شده تاکنون، عملکرد رقابتی ندارند. بسیاری از مدل‌های خطی نشان داده شده در فصل.  [12](#bookmark550) عملکرد مشابه (یا بهتر) داشتند. مدل FDA در این فصل، تا کنون، موثرتر است. با این حال، در تجربه ما، مدل‌های SVM برای اکثر مسائل بسیار رقابتی هستند.

K -نزدیکترین همسایه ها

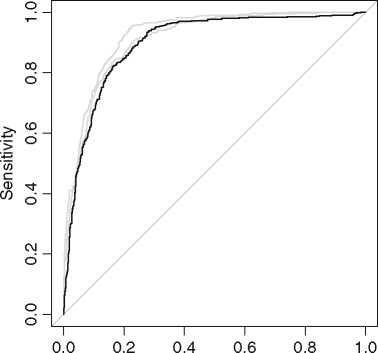
ما برای اولین بار با مدل *K-* نزدیکترین همسایه ( *K* NNs) برای طبقه‌بندی در بخش آشنا شدیم.  [4. 2](#bookmark199) هنگام بحث در مورد تنظیم مدل و مسأله بیش برازش. ما همچنین به‌طور گسترده در مورد *K* NN در زمینه رگرسیون در بخش آموخته ایم.  [7. 4 .](#bookmark363) در حالی که بسیاری از ایده‌های *K* NN برای رگرسیون مستقیماً در اینجا اعمال می‌شوند، ما جنبه‌های منحصربه‌فرد نحوه اعمال این روش در طبقه‌بندی را برجسته می‌کنیم.

روش‌های طبقه‌بندی که تاکنون مورد بحث قرار گرفته‌اند، مرزهای خطی یا غیرخطی را جستجو می‌کنند که به‌طور بهینه داده‌ها را از هم جدا می‌کنند. سپس از این مرزها برای پیش‌بینی طبقه‌بندی نمونه‌های جدید استفاده می‌شود. *K* NN رویکرد متفاوتی دارد

مقیاس = 0. 005 o مقیاس = 0. 010 x



هزینه



1 - خاص بودن

شکل 13. 13: *بالا* : پروفایل‌های عملکرد برای مدل درجه دوم SVM. *پایین* :

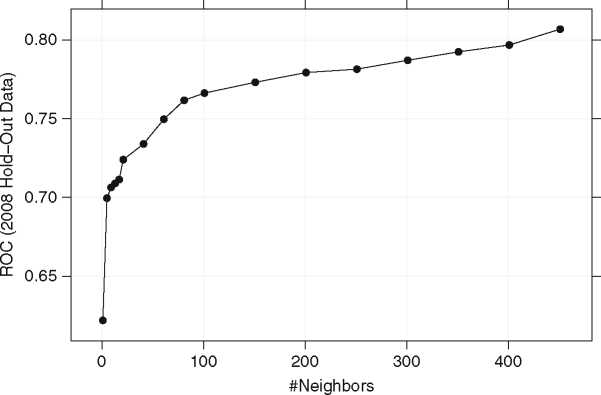
منحنی ROC برای مدل بهینه (مساحت زیر منحنی: 0. 895)

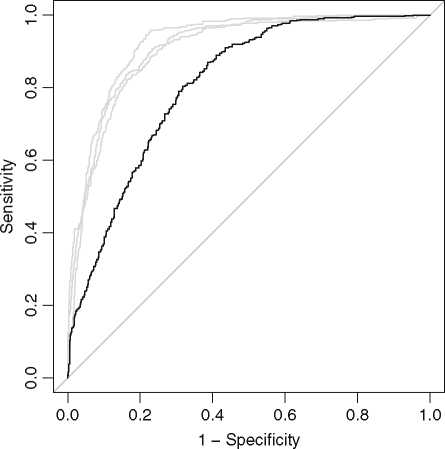
با استفاده از همسایگی جغرافیایی نمونه برای پیش‌بینی طبقه‌بندی نمونه. مشابه زمینه رگرسیون، *K* NN برای طبقه‌بندی یک نمونه جدید را با استفاده از *K-* نزدیکترین نمونه از مجموعه آموزشی پیش‌بینی می‌کند. "نزدیک" توسط یک معیار فاصله تعیین می‌شود، مانند اقلیدسی و مینکوفسکی (بخش.  [7. 4 )](#bookmark363) و انتخاب معیار به ویژگی‌های پیش‌بینی‌کننده بستگی دارد. برای هر معیار فاصله، یادآوری این نکته مهم است که مقیاس‌های اندازه‌گیری اصلی پیش‌بینی‌کننده‌ها بر محاسبات فاصله حاصل تأثیر می‌گذارد. این بدان معناست که اگر پیش‌بینی‌کننده‌ها در مقیاس‌های بسیار متفاوتی باشند، مقدار فاصله بین نمونه‌ها به سمت پیش‌بینی‌کننده‌هایی با مقیاس‌های بزرگ‌تر بایاس خواهد کرد. برای اینکه هر پیش‌بینی به‌طور مساوی در محاسبه فاصله مشارکت داشته باشد، توصیه می‌کنیم قبل از اجرای *K* NN همه پیش‌بینی‌کننده‌ها را در مرکز و مقیاس قرار دهید.

همانطور که در زمینه رگرسیون، برای تعیین طبقه‌بندی یک نمونه جدید، *K-* نزدیک‌ترین نمونه مجموعه آموزشی از طریق معیار فاصله تعیین می‌شود. برآوردهای احتمال کلاس برای نمونه جدید به‌عنوان ­نسبت همسایگان مجموعه آموزشی در هر کلاس محاسبه می‌شود. کلاس پیش‌بینی‌شده نمونه جدید، کلاسی است که بیشترین تخمین احتمال را دارد. اگر دو یا چند کلاس برای بالاترین تخمین با هم گره خورده باشند، این تساوی به‌طور تصادفی یا با نگاه کردن به *K* + 1 نزدیکترین همسایه شکسته می‌شود.

هر روشی با پارامترهای تنظیم می‌تواند مستعد بیش برازش باشد و *K* NN به ویژه در معرض این مسأله است همانطور که در شکل نشان داده شده است.  [4. 2](#bookmark196) . همسایه‌های خیلی کم منجر به برازش بسیار محلی می‌شود (یعنی برازش بیش از حد)، در حالی که همسایه‌های زیاد منجر به مرزهایی می‌شود که ممکن است ساختار جداکننده لازم را در داده‌ها قرار ندهند. بنابراین، برای تعیین مقدار بهینه *K* باید از روش اعتبارسنجی متقابل یا نمونه‌گیری مجدد استفاده کنیم.

برای داده‌های گرنت، محدوده همسایگی ارزیابی شده برای تنظیم بین 1 و 451 بود. شکل [13. 14](#bookmark648) نمایه تنظیم برای ناحیه زیر منحنی ROC را برای داده‌های نگهدارنده 2008 نشان می‌دهد. یک جهش مشخص در ­عملکرد پیش‌بینی از 1 به 5 همسایه و افزایش مداوم عملکرد از طریق محدوده تنظیم وجود دارد. جهش اولیه در عملکرد پیش‌بینی نشان می‌دهد که اطلاعات جغرافیایی محلی برای دسته‌بندی نمونه‌ها بسیار آموزنده است. علاوه بر این، افزایش پیوسته افزایشی در عملکرد پیش‌بینی نشان می‌دهد که همسایگی اطلاعات آموزنده برای طبقه‌بندی نمونه‌ها بسیار بزرگ است. این الگو برای *K* NN تا حدودی غیرمعمول است زیرا با افزایش تعداد همسایگان، ما شروع به کم شدن می‌کنیم و کاهش متناظر در عملکرد پیش‌بینی رخ می‌دهد، همانطور که در شکل نشان داده شده است.  [7. 10](#bookmark367) . در بیشتر مجموعه‌های داده، بعید است از این تعداد همسایه در پیش‌بینی استفاده کنیم. این مثال به شناسایی یک مسأله ناپایداری عددی با *K* NN کمک می‌کند: با افزایش تعداد همسایگان، احتمال پیوندها نیز افزایش می‌یابد. برای این مثال، اندازه محله بزرگتر از 451 منجر به تعداد زیادی کراوات می‌شود. سطح بهینه زیر منحنی ROC 0. 81 بود که در *K* = 451 رخ داد. نمودار پایین در شکل.  [13. 14 نمایه](#bookmark648) *K* NN ROC را با مشخصات SVM و FDA مقایسه می‌کند. برای این داده‌ها، توانایی پیش‌بینی *K* NN نسبت به سایر مدل‌های غیرخطی تنظیم‌شده پایین‌تر است. در حالی که اطلاعات جغرافیایی پیش‌بینی است، به اندازه مدل‌هایی که به دنبال یافتن مرزهای جداکننده بهینه جهانی هستند، مفید نیست.





شکل 13. 14: *بالا* : مشخصات تنظیم پارامتر برای مدل *K* NN. *پایین* :

منحنی ROC برای داده‌های مجموعه آزمایشی. مساحت زیر منحنی 0. 81 بود

بیز ساده

قاعده بیز قبلاً در زمینه تحلیل تفکیک خطی در فصل قبل مورد بحث قرار گرفت. این بخش در مورد آن بحث گسترش می‌یابد و بر یک مدل طبقه‌بندی خاص تمرکز می‌کند که مانند مدل‌های قبلی LDA، QDA و RDA، برحسب چگونگی ایجاد چگالی احتمال چند متغیره تعریف می‌شود.

قانون بیز به این سوال پاسخ می‌دهد که "بر اساس پیش‌بینی‌کننده‌هایی که مشاهده کرده ایم، احتمال اینکه نتیجه کلاس *C باشد چقدر است* ؟" از نظر ریاضی، *Y* متغیر کلاس باشد و *X* مجموعه‌ای از ­متغیرهای پیش فاکتور را نشان دهد. ما سعی می‌کنیم *Pr* [ *Y* = *C^X* ] را تخمین بزنیم که "با توجه به *X،* احتمال اینکه نتیجه کلاس اول باشد چقدر است؟ *"* قانون بیز ابزاری را برای پاسخ به این موضوع فراهم می‌کند:

*Pr* [ *Y* = *C£X* ] =

*Pr* [ *Y* ] *Pr* [ *X |Y* = *C* ]  
*Pr* [ *X* ]

(13.5)

*Pr* [ *Y* = *C|X* ] معمولاً به‌عنوان *احتمال پسین* کلاس نامیده می‌شود. اجزاء عبارتند از:

*Pr* [ *Y* ] احتمال *قبلی* نتیجه است. اساساً، بر اساس آنچه در مورد مسأله می‌دانیم، انتظار داریم که احتمال کلاس چقدر باشد؟ برای مثال، هنگام پیش‌بینی ریزش مشتری، شرکت‌ها معمولاً ایده خوبی از نرخ گردش کلی مشتریان دارند. برای مسائل مربوط به بیماری‌ها، این پیش از آن میزان شیوع بیماری در جمعیت خواهد بود (به بخش.  [11. 2](#bookmark512) در ص.  [254](#bookmark512) برای بحث).

*Pr* [ *X* ] احتمال مقادیر پیش‌بینی است. به‌عنوان مثال، اگر نمونه جدیدی پیش‌بینی شود، این الگو در مقایسه با داده‌های آموزشی چقدر محتمل است؟ به‌طور رسمی، این احتمال با استفاده از توزیع احتمال چند متغیره محاسبه می‌شود. در عمل، معمولاً برای کاهش پیچیدگی این محاسبه، مفروضات قابل‌توجهی مطرح می‌شود.

*Pr* [ *X|Y* = *C* ] *احتمال شرطی است.* برای داده‌های مرتبط با کلاس *C،* احتمال مشاهده مقادیر پیش‌بینی چقدر است؟ مشابه *Pr* [ *X* ]، این می‌تواند یک محاسبه پیچیده باشد مگر اینکه فرضیات دقیقی ایجاد شود.

مدل ساده بیز با فرض اینکه همه پیش‌بینی‌کننده‌ها مستقل از دیگران هستند، احتمالات مقادیر پیش‌بینی را ساده می‌کند. این یک فرض بسیار قوی است. برای اکثر مطالعات موردی و مثال‌های گویا در این متن، ادعای واقعی بودن این فرض دشوار است. با این حال، فرض استقلال کاهش قابل‌توجهی در پیچیدگی محاسبات به همراه دارد.

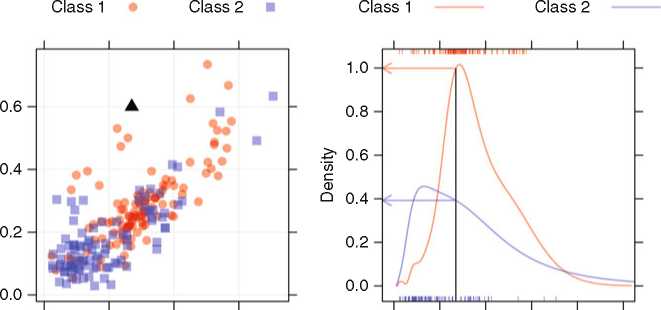
به‌عنوان مثال، برای محاسبه احتمال شرطی *Pr* [ *X |Y* = *C* ]، از حاصل ضرب چگالی احتمال برای هر پیش‌بینی منفرد استفاده می‌کنیم:

پ

*Pr* [ *X |Y* = *C* ] = H *Pr* [ *X j |Y* = *C* ]

j = 1

احتمال نامشروط *Pr* [ *X ] با فرض* استقلال فرمول مشابهی را به همراه دارد. ­برای تخمین احتمالات فردی، یک فرض



m

CD

0.0

0.2

0.4

0.6

0.0

0.2

Predictor A

0.4 0.6 0.8 1.0

Predictor A

Fig. 13.15: *Left*: A plot of two class illustrative data where a new sample (the *solid triangle*) is being predicted. *Right*: Conditional density plots of predictor *A* created using a nonparametric density estimate. The value of predictor *A* for the new sample is shown by the *vertical black line*

نرمال بودن ممکن است برای پیش‌بینی‌کننده‌های پیوسته (با استفاده از میانگین نمونه و واریانس مجموعه آموزشی) ساخته شود. روش‌های دیگر، مانند برآوردگرهای ناپارامتری چگالی هسته [( Hardle et al. 2004](#bookmark1017) )، می‌توانند با انعطاف‌پذیری بیشتری چگالی احتمال را تخمین بزنند. برای پیش‌بینی‌کننده‌های طبقه بندی، توزیع احتمال را می‌توان با فرکانس‌های مشاهده شده در داده‌های مجموعه آموزشی تعیین کرد.

به‌عنوان مثال، شکل.  [13. 15](#bookmark650) مثال تشریح دو کلاسه آشنا را نشان می‌دهد. در پنل سمت چپ، داده‌های آموزشی نشان داده شده است. واضح است که بعید است این دو پیش‌بینی مستقل باشند (همبستگی آنها 0. 78 است). فرض کنید یک نمونه جدید (که به صورت یک مثلث سیاه و سفید نشان داده شده است) نیاز به پیش‌بینی دارد. برای محاسبه احتمال شرطی کلی *Pr* [ *X |K* = *C* ]، هر پیش‌بینی به‌طور جداگانه در نظر گرفته می‌شود. برای پیش‌بینی *A،* دو چگالی شرطی در پانل سمت راست شکل نشان داده شده است.  [13. 15](#bookmark650) با یک خط سیاه عمودی که مقدار نمونه جدید را برای این پیش‌بینی نشان می‌دهد. برای داده‌های مجموعه آموزشی، با استفاده از این پیش‌بینی به تنهایی، به نظر می‌رسد که کلاس اول بسیار محتمل تر باشد.

برای تولید احتمال کلاس *Pr* [ *X |K* = *C* ] برای کلاس اول، دو مقدار احتمال شرطی برای پیش‌بینی‌کننده‌های *A* و *B تعیین می‌شود و* سپس برای محاسبه احتمال شرطی کلی برای کلاس با هم ضرب می‌شوند.

برای *Pr* [ *X* ] یک روش مشابه اتفاق می‌افتد به جز احتمالات برای پیش‌بینی‌کننده‌های *A* و *B* که از کل مجموعه آموزشی (یعنی هر دو کلاس) تعیین می‌شود. برای مثال در شکل.  [13. 15 ،](#bookmark650) همبستگی بین پیش‌بینی‌کننده‌ها نسبتاً قوی است که نشان می‌دهد که نمونه جدید بسیار بعید است.

جدول 13. 1: فراوانی‌ها و احتمالات شرطی *Pr* [ *X|K* = *C* ] برای روز هفته

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| روز | شمردن | | درصد از کل | |
| موفق ناموفق | | موفق ناموفق | |
| دوشنبه | 749 | 803 | 9. 15 | 9. 80 |
| سه شنبه ها | 597 | 658 | 7. 29 | 8. 03 |
| چهارشنبه | 588 | 752 | 7. 18 | 9. 18 |
| پنج شنبه | 416 | 358 | 5. 08 | 4. 37 |
| جمعه | 606 | 952 | 7. 40 | 11. 62 |
| نشست | 619 | 861 | 7. 56 | 10. 51 |
| آفتاب | 228 | 3 | 2. 78 | 0. 04 |

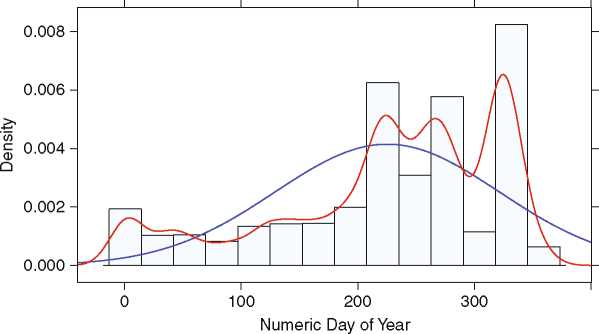
با این حال، با استفاده از فرض استقلال، این احتمال احتمالاً بیش از حد برآورد می‌شود.

احتمال قبلی به مدل ساز اجازه می‌دهد تا احتمال نهایی را به سمت ­یک یا چند کلاس *کج کند.* به‌عنوان مثال، هنگام مدل‌سازی یک رویداد نادر، نمونه‌برداری انتخابی از داده‌ها معمول است تا توزیع کلاس در مجموعه آموزشی متعادل‌تر باشد. با این حال، مدل ساز ممکن است بخواهد مشخص کند که این رویداد واقعاً نادر است با انتساب یک احتمال قبلی کم به آن. اگر به صراحت پیش از این داده نشده باشد، قرارداد استفاده از نسبت‌های مشاهده شده از مجموعه آموزشی برای تخمین قبلی است.

با توجه به چنین فرض شدید و غیر واقعی، چرا باید این مدل را در نظر گرفت؟ اول، مدل ساده بیز را می‌توان به سرعت محاسبه کرد، حتی برای مجموعه‌های آموزشی بزرگ. به‌عنوان مثال، زمانی که پیش‌بینی‌کننده‌ها همگی دسته‌بندی هستند، جداول جستجوی ساده با توزیع‌های فرکانس مجموعه آموزشی همه چیز مورد نیاز است. ثانیاً، با وجود چنین فرض قوی، مدل در بسیاری از موارد به صورت رقابتی عمل می‌کند.

قانون بیز اساساً یک گزاره احتمال است. احتمالات کلاس ایجاد می‌شود و کلاس پیش‌بینی شده کلاسی است که با بزرگترین احتمال کلاس مرتبط است. گوشت مدل، تعیین احتمالات مشروط و غیرشرطی مرتبط با پیش‌بینی‌کننده‌ها است. برای ­پیش‌بینی‌کننده‌های پیوسته، می‌توان مفروضات توزیعی ساده‌ای مانند نرمال بودن را انتخاب کرد. چگالی‌های ناپارامعیار (مانند آنچه در شکل 2 نشان داده شده است).  [13. 16 )](#bookmark652) می تواند برآوردهای احتمالی انعطاف پذیرتری را تولید کند. برای داده‌های درخواست گرنت، پیش‌بینی برای روز عددی سال دارای چندین فریم زمانی است که در آن تعداد نامحدودی از گرنت ارسال شده است. در این شکل، منحنی سیاه برای توزیع نرمال بسیار گسترده است و تفاوت‌های ظریف داده‌ها را نشان نمی‌دهد. منحنی قرمز تخمین ناپارامعیار است و به نظر می‌رسد روندهای داده‌ها را با وفاداری بالاتر تولید می‌کند.

برای پیش‌بینی‌کننده‌های طبقه‌بندی، توزیع فراوانی پیش‌بینی در مجموعه آموزشی برای تخمین *Pr* [ *X* ] و *Pr* [ *X|K* = *C* ] استفاده می‌شود. جدول [13. 1](#bookmark651)



شکل 13. 16: دو رویکرد برای تخمین تابع چگالی *Pr* [ *X* ] برای روز سال. خط *آبی* بر اساس یک توزیع نرمال است در حالی که *خط قرمز* با استفاده از یک تخمین‌گر چگالی ناپارامعیار تولید می‌شود.

فرکانس‌های مشاهده شده را برای روز هفته‌ای که در آن گرنت ارسال شده است را نشان می‌دهد. ستون‌هایی که درصد کل را نشان می‌دهند، تخمین‌های *Pr* [ *X|y* = *C* ] برای هر کلاس هستند. هنگامی که یک نمونه جدید پیش‌بینی می‌شود، یک جستجوی ساده در این جدول برای تخمین احتمالات استفاده می‌شود.

یک مسأله آشکار، به خصوص برای اندازه نمونه‌های کوچک، زمانی رخ می‌دهد که یک یا چند فرکانس صفر باشد. اگر یک پیش‌بینی نمونه‌های مجموعه آموزشی برای یک کلاس خاص نداشته باشد، احتمال شرطی صفر خواهد بود و از آنجایی که ­توانایی‌های احتمالی با هم ضرب می‌شوند، یک پیش‌بینی احتمال پسین را مجبور به صفر می‌کند. یک روش برای جلوگیری از این مسأله استفاده از *اصلاح لاپلاس* یا *صاف کردن لاپلاس است* [( Niblett 1987](#bookmark1023) ; [زادروزنی و الکان 2001](#bookmark1028) ; [Provost و Domingos 2003](#bookmark1023) ) که در آن همان ضریب تصحیح، معمولاً ­بین یک و دو، به صورت‌گر اضافه می‌شود. برای مخرج، فرکانس‌ها با ضریب تصحیح ضربدر تعداد ­مقادیر پیش‌بینی افزایش می‌یابد. به‌عنوان مثال، فرکانس‌های بسیار پایینی برای گرنت ارسال شده در روز یکشنبه وجود دارد. برای تصحیح احتمالات شدید، ضریب تصحیح یک فرکانس مشاهده شده را به 229 و 4 تغییر می‌دهد، اما مخرج هفت افزایش می‌یابد. با توجه به حجم نمونه بزرگ برای مجموعه آموزشی، این تصحیح تنها تأثیر کمی دارد ( ­نرخ موفقیت تخمینی در روز یکشنبه از 2. 78٪ به 2. 79٪ افزایش یافته است). با این حال، هر سه گرنت ناموفق در جدول پس از سال 2008 ارائه شده اند. آموزش در مورد داده‌های قبل از 2008 احتمال صفر را ایجاد می‌کند. در این حالت، تصحیح مقدار یک، احتمال کمک‌های بلاعوض را به 0. 02% تغییر می‌دهد در حالی که ­ضریب تصحیح دو، مقدار را به 0. 03% افزایش می‌دهد. برای اندازه‌های مجموعه آموزشی کوچکتر، اصلاح می‌تواند تأثیر مثبت قابل‌توجهی بر سلول‌های از دست رفته در جدول داشته باشد.

برای داده‌های گرنت، بسیاری از پیش‌بینی‌کننده‌ها شمارش بودند. اگرچه اینها اعداد هستند، اما مقادیر گسسته‌ای هستند و می‌توانند به‌عنوان دسته‌بندی در نظر گرفته شوند. در بسیاری از موارد، توزیع فرکانس مشاهده شده فشرده است. به‌عنوان مثال، در مجموعه آموزشی، تعداد محققین ارشد در بخش 2678 چهار مقدار بین 0 و 3 را می‌گیرد و دارای توزیع بسیار راست می‌باشد. برخورد با چنین پیش‌بینی دانه‌ای به‌گونه‌ای که گویی توسط ­توزیع متقارن یا بد ایجاد شده است، ممکن است تخمین‌های احتمال ضعیفی را ایجاد کند. برای این تحلیل، مجموعه کاهش‌یافته پیش‌بینی‌کننده‌ها به‌گونه‌ای ارزیابی شد که همه پیش‌بینی‌کننده‌ها با کمتر از 15 مقدار ممکن به‌عنوان گسسته در نظر گرفته شدند و احتمالات آنها با استفاده از توزیع فراوانی آنها محاسبه شد (مانند روز هفته نشان داده شده در جدول).  [13. 1](#bookmark651) . 14 پیش‌بینی با بیش از 15 ارزش منحصربه‌فرد، از جمله تعداد گرنت موفق توسط محققین ارشد، تعداد مقالات ژورنال A *\** توسط محققین ارشد و روز عددی سال وجود داشت.

این پیش‌بینی‌کننده‌ها با استفاده از یک توزیع نرمال یا یک چگالی ناپارامعیار (نوع چگالی به‌عنوان یک پارامتر تنظیم در نظر گرفته شد) مدل‌سازی شدند و از اصلاح لاپلاس 2 استفاده شد. هنگام استفاده از توزیع نرمال برای پیش‌بینی‌کننده‌های پیوسته، سطح زیر منحنی 0. 78، حساسیت 58. 8 درصد و ویژگی 79. 6 درصد برآورد شد. با استفاده از تخمین ناپارامتری ­ric از چگالی احتمال، سطح زیر منحنی ROC به 0. 81 بهبود می‌یابد که مربوط به افزایش حساسیت (64. 4٪) و ویژگی (82. 4٪) است. متأسفانه، عملکرد این مدل با *K* NN برابر است که به‌طور قابل ملاحظه‌ای کمتر از نتایج مدل‌های دیگر در این فصل است.

بخش [11. 1](#bookmark497) نشان داد که قانون بیز می‌تواند برای کالیبره کردن ­تخمین‌های احتمالی کلاس استفاده شود. برای انجام این کار، کلاس‌های واقعی به‌عنوان *Y استفاده می‌شوند،* اما مقادیر احتمال کلاس ­برای مجموعه آموزشی به‌عنوان "پیش‌بینی" استفاده می‌شود و *Pr* [ *X |K* = *C* ] از پیش‌بینی‌کننده‌های مدل در مجموعه آموزشی تعیین می‌شود. هنگامی که نمونه‌های جدید ­پیش‌بینی می‌شوند، احتمالات کلاسی که توسط مدل تولید می‌شوند با استفاده از قانون بیز برای بهبود کالیبراسیون پس پردازش می‌شوند. از قضا، احتمالات طبقاتی ایجاد شده با اعمال قانون بیز به روش معمولی، خود به خوبی کالیبره نمی‌شوند. با افزایش تعداد پیش‌بینی‌کننده‌ها ( ­نسبت به حجم نمونه)، احتمالات پسین شدیدتر می‌شوند (مشابه مشاهدات مربوط به تحلیل متمایز خطی نشان داده شده در شکل 1).  [12. 11 )](#bookmark569) . به یاد بیاورید که QDA بر اساس قانون بیز (با استفاده از چند متغیره یا ­بدخیم بودن برای پیش‌بینی ها) و نتایج QDA نشان داده شده در شکل است.  [11. 1](#bookmark502) کالیبراسیون ضعیفی را با دو متغیر پیش‌بینی نشان داد (اما با کالیبراسیون مجدد با استفاده از برنامه دیگری از قانون بیز بهبود یافت).

13. 7 محاسبات

بسته‌های R زیر در این فصل مورد بحث قرار می‌گیرند: caret، earth، kernlab، klaR، MASS، mda، nnet و rrcov. این بخش همچنین از همان اشیاء R ایجاد شده در فصل آخر که حاوی داده‌ها هستند (مانند آموزش قاب داده ) استفاده می‌کند.

تحلیل تفکیک غیر خطی

تعدادی بسته برای انجام انواع تحلیل تفکیک غیرخطی که قبلا در این فصل توضیح داده شد در دسترس است. QDA در تابع qda در MASS و همچنین یک نسخه مقاوم در برابر بیرونی در تابع QdaCov در بسته rrcov پیاده‌سازی شده است. RDA در تابع rda در بسته klaR موجود است و MDA را می‌توان در بسته mda یافت. نحو برای این مدل‌ها بسیار شبیه است و ما استفاده از آنها را با برازش یک مدل MDA به داده‌های گرنت نشان خواهیم داد.

تابع mda دارای یک رابط فرمول مدل است. پارامتر تنظیم تعداد زیر کلاس‌ها در هر کلاس است که لازم نیست برای هر کلاس یکسان باشد. به‌عنوان مثال، برای تطبیق یک مدل MDA به داده‌های گرنت با سه گروه فرعی در هر کلاس:

*> کتابخانه (mda)*

*> mdaModel <- mda(Class ~. ,*

*+ ## داده‌ها را به پیش‌بینی‌کننده‌های مربوطه کاهش دهید*

*+ ## متغیر کلاس برای استفاده از میانبر فرمول بالا*

*+ داده = آموزش [pre2008، c("Class"، smallSet)]،*

*+ زیر کلاس‌ها = 3)*

*> mdaModel*

صدا زدن:

mda(فرمول = کلاس ~. ، داده = آموزش[pre2008، c("Class"، smallSet)]، زیر کلاس‌ها = 3)

ابعاد: 5

درصد واریانس بین گروهی توضیح داده شده: v1 v2 v3 v4 v5

72. 50 92. 57 96. 10 98. 66 100. 00

درجات آزادی (در هر بعد): 253

خطای طبقه‌بندی اشتباه آموزش: 0. 18709 (N = 6633)

انحراف: 6429. 499

*> پیش‌بینی (mdaModel,*

*+ newdata = head(training[-pre2008, smallSet]))*

[1]

هر یک از این مدل‌های متمایز غیرخطی را می‌توان ساخت و پارامترهای تنظیم بهینه را می‌توان با استفاده از بسته caret پیدا کرد. گزینه trControl برای داده‌های گرنت همانطور که در بخش توضیح داده شد تنظیم شده است.  [12. 7](#bookmark585) و در اینجا استفاده خواهد شد:

*set. seed(476)*

*mdaFit <- train(training[,reducedSet], training$Class,*

*+ روش = "mda"،*

*+ معیار = "ROC"،*

*+ tuneGrid = expand. grid(. subclasses = 1:8)،*

*+ trControl = ctrl)*

نحو مشابهی را می‌توان برای RDA (با استفاده از روش = "rda" ) و QDA (مقادیر روش "rda" یا "QdaCov" برای نسخه مقاوم در برابر خارج در بسته rrcov استفاده کرد.

نسخه جریمه شده MDA نیز در بسته sparseLDA با عملکرد smda موجود است. دیدن [کلمنسن و همکاران ( 2011](#bookmark1013) ) برای جزئیات بیشتر.

شبکه‌های عصبی

R زیادی برای شبکه‌های عصبی وجود دارد، از جمله nnet، RSNNS، qrnn و neuralnet. دو منبع برای استفاده از شبکه‌های عصبی [g در](#bookmark1026) R عبارتند از Venables و Ripley [( 2002](#bookmark1026) ) و Sect. 7 از [برگمیر و بنیتز ( 2012](#bookmark1011) ).

تحلیل در اینجا بر روی بسته nnet متمرکز است. نحو بسیار شبیه به مدل‌های رگرسیون با چند استثنا است. آرگومان linout باید روی FALSE تنظیم شود زیرا اکثر مدل‌های طبقه‌بندی از یک ­تبدیل مدال علامت برای مرتبط کردن واحدهای پنهان به خروجی‌ها استفاده می‌کنند. مجموع خطاهای مجذور یا تخمین‌های آنتروپی پارامترهای مدل و آرگومان‌های منطقی ­softmax و آنتروپی بین این دو جابجا می‌شوند.

بسته دارای یک رابط فرمول و یک رابط برای عبور ماتریس‌ها یا فریم‌های داده برای پیش‌بینی‌کننده‌ها و نتیجه است. برای دومی، نتیجه نمی‌تواند یک متغیر عامل باشد و باید به مجموعه‌ای از شاخص‌های باینری *C تبدیل شود.* این بسته حاوی یک تابع، class. ind است که در انجام این تبدیل مفید است:

*head(class. ind(training$Class))* موفق ناموفق [1،] 1 0

[2،] 1 0

[3،] 1 0

[4،] 1 0

[5،] 0 1

[6،] 1 0

استفاده از رابط فرمول برای جا دادن یک مدل ساده:

*set. seed(800)*

*nnetMod <- nnet(Class ~ NumCI + CI. 1960،*

*+ داده = آموزش [قبل از 2008،]،*

*+ اندازه = 3، پوسیدگی = 0. 1)*

# وزن: 13 مقدار اولیه 4802. 892391 iter 10 مقدار 4595. 629073 iter 20 مقدار 4584. 893054 iter 30 مقدار 4582. 614616 iter 40 مقدار iter 4581. 010289 iter 10 مقدار 4581. 010289 iter 4581. 010289 مقدار 4581. 010289 iter 4581. 010289 iter 10.

ارزش نهایی 4580. 756133

همگرا شد

*> nnetMod*

یک شبکه 2-3-1 با 13 وزن

ورودی: NumCI CI. 1960

خروجی(های): کلاس

گزینه‌ها بودند - آنتروپی برازش واپاشی = 0. 1

*پیش‌بینی (nnetMod، newdata = head (تست))*

[,1]

6641 0. 5178744

6647 0. 5178744

6649 0. 5138892

6650 0. 5837029

6655 0. 4899851

6659 0. 5701479

*پیش‌بینی (nnetMod، newdata = head (تست)، نوع = "کلاس")*

[1] "ناموفق" "ناموفق" "ناموفق" "ناموفق"

"موفق" "ناموفق"

وقتی سه یا چند کلاس مدل می‌شوند، فراخوانی اولیه برای پیش‌بینی، ستون‌هایی را برای هر کلاس تولید می‌کند.

مانند قبل، آموزش برای این تابع یک پوشش برای تنظیم مدل بر روی میزان پوسیدگی وزن و تعداد واحدهای پنهان فراهم می‌کند. از همان کد مدل استفاده می‌شود ( متد = "nnet" ) و هر یک از مدل‌ها در دسترس است، اگرچه train بردارهای فاکتور را برای کلاس‌ها اجازه می‌دهد (با استفاده از class. ind به صورت داخلی ­متغیرهای ساختگی را کد می‌کنند). همچنین، مانند رگرسیون، مدل av ­eraging را می‌توان از طریق تابع مستقل avNNet یا با استفاده از آموزش (با متد = "avNNet" ) استفاده کرد.

مدل نهایی برای داده‌های گرنت دارای نحو زیر است:

*nnetGrid <- expand. grid(. size = 1:10,*

*+ . decay = c(0، 0. 1، 1، 2))*

*maxSize <- max(nnetGrid$. size)*

*numWts <- 1\*(maxSize \* (length(reducedSet) + 1) + maxSize + 1)*

*set. seed(476)*

*nnetFit <- train(x = training[,reducedSet],*

*+ y = آموزش$Class،*

*+ روش = "nnet"،*

*+ معیار = "ROC"،*

*+ preProc = c ("مرکز"، "مقیاس"، "SpatialSign")،*

*+ tuneGrid = nnetGrid،*

*+ ردیابی = نادرست،*

*+ حداکثر = 2000،*

*+ MaxNWts = numWts،*

*+ ## ctrl در فصل قبل تعریف شد*

*+ trControl = ctrl)*

تحلیل تفکیک انعطاف پذیر

بسته mda شامل یک تابع ( fda ) برای ساخت این مدل است. مدل رابط فرمول را می‌پذیرد و یک گزینه ( متد ) دارد که روش دقیق تخمین پارامترهای رگرسیون را مشخص می‌کند. برای استفاده از FDA با MARS، دو رویکرد وجود دارد. روش = mars از پیاده‌سازی MARS ­در بسته mda استفاده می‌کند. با این حال، بسته زمین که قبلا در بخش توضیح داده شد.  [7. 5](#bookmark367) ، متناسب با مدل MARS با طیف وسیع‌تری از گزینه‌ها است. در اینجا، بسته ارت را بارگذاری کنید و سپس روش = earth را مشخص کنید. به‌عنوان مثال، یک مدل ساده FDA برای داده‌های درخواست گرنت می‌تواند به‌عنوان ایجاد شود

*کتابخانه (mda)*

*کتابخانه (زمین)*

*fdaModel <- fda(Class ~ Day + NumCI, data = training[pre2008,],*

*+ روش = زمین)*

آرگومان‌های تابع زمین مانند nprune را می‌توان در هنگام فراخوانی fda مشخص کرد و به زمین ارسال می‌شود. مدل MARS در موضوعی به نام برازش موجود است :

*خلاصه (fdaModel$fit)*

فراخوانی: زمین (x=x، y=تتا، وزن=وزن)

(Intercept) h(Day-91) h(Day-202) h(Day-228) h(228-Day) h(Day-282) h(Day-319) h(Day-328) h(Day-332) h(Day-336) h(1-NumCI)

coefficients

1.41053449

-0.01348332

0.03259400

-0.02660477

-0.00997109

-0.00831905

0.17945773

-0.51574151

0.50725158

-0.20323060

0.11782107

11 مورد از 12 عبارت و 2 مورد از 2 پیش‌بینی انتخاب شد

اهمیت: روز، NumCI

تعداد اصطلاحات در هر درجه تعامل: 1 10 (مدل افزودنی)

GCV 0. 8660403 RSS 5708. 129 GRSq 0. 1342208 RSq 0. 1394347

توجه داشته باشید که ضرایب مدل نشان داده شده در اینجا پس پردازش نشده اند. ضرایب مدل نهایی را می‌توان با coef(fdaModel) پیدا کرد. برای پیش‌بینی:

*پیش‌بینی (fdaModel, head(training[-pre2008,]))*

[1] موفق موفق موفق موفق موفق موفق

سطوح: موفق ناموفق

تابع آموزش را می‌توان با روش = "fda" برای تنظیم این مدل بر روی تعداد عبارت‌های حفظ شده استفاده کرد. علاوه بر این، تابع varImp از این بسته، اهمیت پیش‌بینی را به همان روشی که برای مدل‌های MARS تعیین می‌کند (شرح شده در بخش.  [7. 2 )](#bookmark348) .

ماشین‌های بردار پشتیبانی

همانطور که در فصل رگرسیون بحث شد، تعدادی بسته R با پیاده‌سازی برای ماشین بردار پشتیبان و سایر روش‌های هسته، از جمله e1071، kernlab، klaR و svmPath وجود دارد. جامع‌ترین آنها بسته kernlab است.

نحو برای مدل‌های طبقه‌بندی SVM تا حد زیادی مشابه ­حالت رگرسیون است. اگرچه پارامتر اپسیلون فقط برای رگرسیون مرتبط است، چند پارامتر دیگر برای طبقه‌بندی مفید هستند:

آرگومان منطقی prob. model ksvm را برای تخمین مجموعه پارامترهای اضافی برای یک تابع سیگموئیدی برای ترجمه مقادیر تصمیم SVM به احتمالات کلاس با استفاده از روش [پلات ( 2000](#bookmark1023) ). اگر این گزینه روی TRUE تنظیم نشود، احتمالات کلاس قابل پیش‌بینی نیست.

class. weights هزینه‌های نامتقارن را به هر کلاس اختصاص می‌دهد [(](#bookmark1023) Osuna et al. [1997](#bookmark1023) ). این می‌تواند به ویژه زمانی مهم باشد که یک یا چند نوع خاص از خطاها مضرتر از سایرین باشند یا زمانی که عدم تعادل کلاسی شدیدی وجود دارد که مدل را به کلاس اکثریت بایاس می‌کند (به فصل 16 مراجعه کنید ). نحو در اینجا استفاده از یک بردار *نامگذاری شده* از وزن یا هزینه است. به‌عنوان مثال، اگر تمایل به بایاس مدل گرنت برای شناسایی گرنت ناموفق وجود داشته باشد، آنگاه دستور نحوی خواهد بود.

*class. weights = c (موفق = 1، ناموفق = 5)*

این یک خطای منفی کاذب را پنج برابر بیشتر از یک ­خطای مثبت کاذب هزینه می‌کند. توجه داشته باشید که اجرای وزن کلاس در ksvm بر کلاس پیش‌بینی شده تأثیر می‌گذارد، اما مدل احتمال کلاس تحت تأثیر وزن‌ها (در این پیاده سازی) قرار نمی‌گیرد. این ویژگی در فصل استفاده شده است.  [17 .](#bookmark795)

کد زیر یک تابع پایه شعاعی را با مجموعه کاهش یافته پیش‌بینی‌کننده‌ها در داده‌های گرنت منطبق می‌کند:

*set. seed(202)*

*sigmaRangeReduced <- sigest(as. matrix(training[,reducedSet]))*

*svmRGridReduced <- expand. grid(. sigma = sigmaRangeReduced[1]،*

*+ . C = 2" (دنباله (-4، 4)))*

*set. seed(476)*

*svmRMmodel <- train(training[,reducedSet], training$Class,*

*+ روش = "svmRadial"،*

*+ معیار = "ROC"،*

*+ preProc = c ("مرکز"، "مقیاس")،*

*+ tuneGrid = svmRGridReduced،*

*+ مناسب = نادرست،*

*+ trControl = ctrl)*

*svmRMmodel*

8190 نمونه

252 پیش‌بینی

2 کلاس: "موفق"، "ناموفق"

پیش پردازش: متمرکز، مقیاس شده

نمونه‌برداری مجدد: تقسیم‌های تمرین/تست مکرر (1 تکرار، 0. 75%)

خلاصه حجم نمونه: 6633

نمونه برداری مجدد از نتایج در پارامترهای تنظیم:

|  |  |  |  |
| --- | --- | --- | --- |
| سی | ROC | حس | مشخصات |
| 0. 0625 | 0. 866 | 0. 775 | 0. 787 |
| 0. 125 | 0. 88 | 0. 842 | 0. 776 |
| 0. 25 | 0. 89 | 0. 867 | 0. 772 |
| 0. 5 | 0. 894 | 0. 851 | 0. 784 |
| 1 | 0. 895 | 0. 84 | 0. 804 |
| 2 | NaN | 0. 814 | 0. 814 |
| 4 | 0. 887 | 0. 814 | 0. 812 |
| 8 | 0. 885 | 0. 804 | 0. 814 |
| 16 | 0. 882 | 0. 805 | 0. 818 |

پارامتر تنظیم 'sigma' در مقدار 0. 00117 ثابت نگه داشته شد

برای انتخاب مدل بهینه با استفاده از بزرگترین مقدار از ROC استفاده شد.

مقادیر نهایی مورد استفاده برای مدل C = 1 و سیگما = 0. 00117 بود.

هنگامی که نتیجه یک عامل است، تابع به‌طور خودکار از prob. model = TRUE استفاده می‌کند.

سایر توابع هسته را می‌توان از طریق آرگومان‌های کرنل و kpar تعریف کرد.

پیش‌بینی نمونه‌های جدید از الگوی مشابه سایر توابع پیروی می‌کند:

*کتابخانه (kernlab)*

*پیش‌بینی (svmRModel، newdata = head(training[-pre2008, smallSet]))*

[1]

|  |  |  |
| --- | --- | --- |
| *> پیش‌بینی (svmRModel، newdata = head(training[-pre2008, smallSet])* | | |
| *+* | *نوع* | *= "مسأله")* |
|  | موفقیت آمیز | ناموفق |
| 1 | 0. 9522587 | 0. 04774130 |
| 2 | 0. 8510325 | 0. 14896755 |
| 3 | 0. 8488238 | 0. 15117620 |
| 4 | 0. 9453771 | 0. 05462293 |
| 5 | 0. 9537204 | 0. 04627964 |
| 6 | 0. 5009338 | 0. 49906620 |

K -نزدیکترین همسایه ها

برازش یک مدل طبقه‌بندی *K* NN دارای نحوی مشابه با برازش یک مدل رگرسیونی است. در این تنظیمات، بسته caret با متد تنظیم شده روی "knn" مدل را تولید می‌کند. نحوی که برای تولید بالای شکل استفاده می‌شود.  [13. 14](#bookmark648) است

*set. seed(476)*

*knnFit <- train(training[,reducedSet], training$Class,*

*روش = "knn"،*

*معیار = "ROC"،*

*+*

*+*

*+*

*+*

*+*

*+*

*preProc = c("مرکز"، "مقیاس")، tuneGrid = data. frame(. k = c(4\*(0:5)+1، trControl = ctrl)*

*20\*(1:5)+1,*

*50\*(2:9)+1)),*

کد زیر داده‌های مجموعه تست و منحنی ROC مربوطه را پیش‌بینی می‌کند:

*knnFit$pred <- ادغام (knnFit$pred، knnFit$bestTune)*

*knnRoc <- roc(پاسخ = knnFit$pred$obs،*

*+ پیش‌بینی = knnFit$pred$موفق،*

*+ سطوح = دور (سطوح (knnFit$pred$obs)))*

*طرح (knnRoc، legacy. axes = TRUE)*

بیز ساده

دو عملکرد اصلی برای قرار دادن مدل‌های ساده بیز در R عبارتند از: ساده‌بازی‌ها در پکیج e1071 و NaiveBayes در بسته‌بندی klaR. هر دو اصلاحات لاپلاس را ارائه می‌دهند، اما نسخه موجود در بسته klaR این گزینه را دارد که از ­تخمین‌های چگالی شرطی که انعطاف پذیرتر هستند استفاده کند.

هر دو تابع، رویکردهای فرمول و غیرفرمول را برای تعیین ­شرایط مدل می‌پذیرند. با این حال، تغذیه این مدل‌ها با متغیرهای ساختگی باینری (به‌جای متغیر عامل) مسأله‌ساز است، زیرا دسته‌های جداگانه به‌عنوان داده‌های عددی در نظر گرفته می‌شوند و مدل ­تابع چگالی احتمال (یعنی *Pr* [ *X* ]) را از یک توزیع پیوسته تخمین می‌زند. مانند گاوسی.

برای پیروی از استراتژی توضیح داده شده در بالا که در آن بسیاری از پیش‌بینی‌کننده‌ها به متغیرهای عاملی تبدیل می‌شوند، نسخه‌های جایگزینی از مجموعه‌های آموزشی و آزمایشی ایجاد می‌کنیم:

*## برخی از پیش‌بینی‌کننده‌ها قبلاً به‌عنوان فاکتور ذخیره شده اند*

*عوامل <- c("SponsorCode"، "ContractValueBand"، "Month"، "Weekday")*

*## سایر پیش‌بینی‌کننده‌ها را از مجموعه کاهش یافته دریافت کنید*

*nbPredictors <- factorPredictors[factorPredictors %in% smallSet]*

*nbPredictors <- c(nbPredictors, factor)*

*## تره فرنگی فقط مورد نیاز است*

*nbTraining <- training[, c("Class"، nbPredictors)]*

*nbTesting <- testing[, c("Class"، nbPredictors)]*

*## از طریق پیش‌بینی‌کننده‌ها حلقه بزنید و برخی را به فاکتور تبدیل کنید*

*برای (i در nbPredictors)*

*+{*

*+ varLevels <- sort(unique(training[,i]))*

*+ if(length(varLevels) <= 15)*

*+{*

*+ nbTraining[, i] <- factor(nbTraining[,i],*

*+ سطوح = چسباندن (varLevels))*

*+ nbTesting[, i] <- factor(nbTesting[,i],*

*+ سطوح = چسباندن (varLevels))*

*+}*

*+}*

اکنون می‌توانیم از رابط فرمول تابع NaiveBayes برای ایجاد یک مدل استفاده کنیم:

*کتابخانه (klaR)*

*nBayesFit <- NaiveBayes(Class ~. ,*

*+ داده = nbTraining[pre2008،]،*

*+ ## باید برآورد ناپارامعیار*

*+ ## استفاده شود؟*

*+ userkernel = TRUE،*

*+ ## مقدار تصحیح لاپلاس*

*+ fL=2)*

*پیش‌بینی (nBayesFit، newdata = head(nbTesting))*

کلاس $

6641 6647 6649 6650 6655 6659

موفق موفق موفق موفق موفق موفق موفق

سطوح: موفق ناموفق

$ پسین

موفق ناموفق

6641 0. 9937862 6. 213817e-03

6647 0. 8143309 1. 856691e-01

6649 0. 9999078 9. 222923e-05

6650 0. 9992232 7. 768286e-04

6655 0. 9967181 3. 281949e-03

6659 0. 9922326 7. 767364e-03

در برخی موارد، یک هشدار ظاهر می‌شود: "احتمال عددی 0 برای همه کلاس‌ها با مشاهده 1. " تابع پیش‌بینی برای این مدل آرگومانی به نام آستانه دارد که مقادیر صفر را با یک عدد کوچک و غیر صفر (به‌طور پیش فرض 0. 001) جایگزین می‌کند.

تابع آموزش با روش تخمین چگالی (یعنی userkernel ) و اصلاح لاپلاس به‌عنوان پارامترهای تنظیم رفتار می‌کند. به‌طور پیش فرض، تابع احتمالات را با توزیع نرمال و روش ناپارامعیار (و بدون اصلاح لاپلاس) ارزیابی می‌کند.

تمرینات

از داده‌های آسیب کبدی از مجموعه تمرینات قبلی استفاده کنید (Exer ­cise [12. 1 )](#bookmark605) . به یاد بیاورید که ماتریس‌های زیستی و شیمیایی حاوی سنجش بیولوژیکی و پیش‌بینی اثر انگشت شیمیایی برای ۲۸۱ ترکیب هستند، در حالی که آسیب ناقل شامل طبقه‌بندی آسیب کبدی برای هر ترکیب است.

با همان مجموعه‌های آموزشی و آزمایشی و همچنین مراحل پیش پردازشی که در کار قبلی خود روی این داده‌ها انجام دادید، کار کنید. با استفاده از همان آمار طبقه‌بندی قبلی، مدل‌هایی را که در این فصل برای پیش‌بینی‌کننده‌های بیولوژیکی و به‌طور جداگانه برای پیش‌بینی‌کننده‌های اثرانگشت شیمیایی توضیح داده شده است، بسازید. کدام مدل بهترین توانایی پیش‌بینی را برای پیش‌بینی‌کننده‌های بیولوژیکی دارد و عملکرد بهینه چیست؟ کدام مدل بهترین توانایی پیش‌بینی را برای پیش‌بینی‌کننده‌های شیمیایی دارد و ­عملکرد بهینه آن چقدر است؟ آیا ساختار غیرخطی این مدل‌ها به بهبود عملکرد طبقه‌بندی کمک می‌کند؟

برای مدل‌های بهینه برای پیش‌بینی‌کننده‌های بیولوژیکی و شیمیایی، پنج پیش‌بینی مهم کدامند؟

اکنون پیش‌بینی‌کننده‌های اثرانگشت بیولوژیکی و شیمیایی را در یک مجموعه پیش‌بینی ترکیب کنید. همان مجموعه‌ای از مدل‌های پیش‌بینی را که از قسمت (الف) ساخته اید، دوباره آموزش دهید. کدام مدل بهترین عملکرد پیش‌بینی را دارد؟ آیا عملکرد مدل بهتر از هر یک از بهترین مدل‌های قسمت (الف) است؟ 5 پیش‌بینی مهم برای مدل بهینه کدامند؟ اینها چگونه با پیش‌بینی‌کننده‌های بهینه از هر مجموعه پیش‌بینی منفرد مقایسه می‌شوند؟ چگونه این پیش‌بینی‌کننده‌های مهم پیش‌بینی‌کننده‌های مدل‌های خطی را مقایسه می‌کنند؟

در صورت وجود، از کدام مدل (هر مدل بیولوژی فردی یا اثر انگشت شیمیایی یا مدل پیش‌بینی ترکیبی)، برای پیش‌بینی سمیت کبدی ترکیبات استفاده می‌کنید؟ توضیح.

از داده‌های اسید چرب مجموعه تمرین قبلی استفاده کنید (ورزش [12. 2 )](#bookmark31) .

از همان روش تقسیم داده‌ها (در صورت وجود) و مراحل پیش پردازش استفاده کنید که در فصل قبل انجام دادید. با استفاده از همان آمار طبقه‌بندی قبلی، مدل‌هایی را که در این فصل توضیح داده شده است برای این داده‌ها بسازید. کدام مدل بهترین توانایی پیش‌بینی را دارد؟ عملکرد این مدل بهینه در مقایسه با عملکرد بهترین مدل خطی چگونه است؟ آیا بر اساس این مقایسه استنباط می‌کنید که داده‌ها دارای مرزهای جدایی غیرخطی هستند؟

مدل بهینه کدام نوع روغن را دقیق‌تر پیش‌بینی می‌کند؟ حداقل پیش‌بینی دقیق؟

فصل 14

طبقه‌بندی درختان و بر اساس قانون

مدل ها

درخت‌های طبقه‌بندی در خانواده مدل‌های مبتنی بر درخت قرار می‌گیرند و مانند درخت‌های رگرسیونی، از دستورات if-then تودرتو تشکیل می‌شوند. برای مسئله آشنای دو کلاسه نشان داده شده در دو فصل آخر، یک درخت طبقه‌بندی ساده ممکن است باشد

اگر پیش‌بینی B >= 0. 197 باشد، آنگاه

| اگر پیش‌بینی A >= 0. 13 باشد، کلاس = 1

| دیگری کلاس = 2

دیگری کلاس = 2

در این مورد، فضای پیش‌بینی دوبعدی به سه ناحیه (یا گره‌های پایانی) بریده می‌شود و در هر ناحیه، نتیجه به «کلاس 1» یا «کلاس 2» طبقه‌بندی می‌شود. شکل [14. 1](#bookmark686) درخت را در فضای پیش‌بینی نشان می‌دهد. درست مانند تنظیمات رگرسیون، دستورات تودرتوی if-then را می‌توان در قوانینی مانند

if Predictor A >= 0.13

if Predictor A >= 0.13

if Predictor A < 0.13

and Predictor B >= and Predictor B < then Class = 2

0.197 then Class = 1

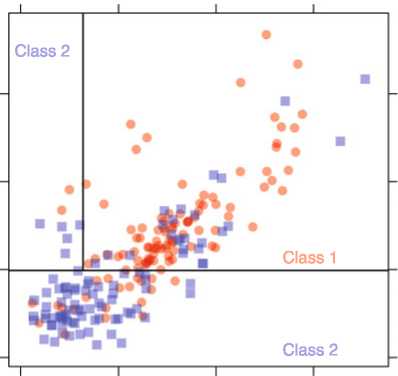
0.197 then Class = 2

واضح است که ساختار درختان و قوانین مشابه ساختاری است که در تنظیمات رگرسیون دیدیم. و مزایا و نقاط ضعف درختان در تنظیمات طبقه‌بندی نیز مشابه است: آنها می‌توانند بسیار قابل تفسیر باشند، می‌توانند بسیاری از انواع پیش‌بینی‌کننده‌ها و همچنین داده‌های از دست رفته را مدیریت کنند، اما از بی ثباتی مدل رنج می‌برند و ممکن است عملکرد پیش‌بینی مطلوبی را ایجاد نکنند. با این حال، فرآیند یافتن تقسیم‌بندی‌ها و قوانین بهینه به دلیل تغییر در معیارهای بهینه‌سازی که در ادامه توضیح داده خواهد شد، کمی متفاوت است.

جنگل‌های تصادفی، تقویت و سایر روش‌های گروهی با استفاده از ­درخت‌ها یا قوانین طبقه‌بندی شده نیز به همین ترتیب به این تنظیمات بسط داده شده‌اند و در فرقه‌ها مورد بحث قرار می‌گیرند.  [14. 3](#bookmark699) تا [14. 6 .](#bookmark715)

\_14،

©



Class 1 • Class 2 ■

14. 1 درختان طبقه‌بندی اساسی

همانند درختان رگرسیون، هدف درختان طبقه‌بندی تقسیم داده‌ها به گروه‌های کوچکتر و همگن تر است. همگنی در این زمینه به این معنی است که گره‌های تقسیم خالص تر هستند (یعنی شامل نسبت بزرگتری از یک کلاس در هر گره). یک راه ساده برای تعریف خلوص در طبقه‌بندی، به حداکثر رساندن دقت یا معادل آن با به حداقل رساندن خطای طبقه‌بندی اشتباه است. با این حال، دقت به‌عنوان معیار خلوص کمی گمراه‌کننده است، زیرا تمرکز این معیار بر پارتیشن‌بندی داده‌ها به گونه‌ای است که طبقه‌بندی اشتباه را به حداقل می‌رساند ­تا تمرکز بر پارتیشن‌بندی داده‌ها به گونه‌ای که نمونه‌ها را عمدتاً در یک کلاس قرار می‌دهد.

دو معیار جایگزین، **شاخص جینی** [( Breiman et al. 1984](#bookmark1012) ) **و آنتروپی** متقاطع که به‌عنوان **انحراف یا اطلاعات** نیز نامیده می‌شود (در ادامه در این بخش تعریف می‌شود)، تمرکز را از دقت به خلوص تغییر می‌دهد. برای مسئله دو کلاسه، شاخص جینی برای یک گره معین به صورت تعریف شده است

*p* 1 (1 *- p* 1 )+ *p* 2 (1 *- p* 2 ) *,*  (14. 1)

که در آن *p* 1 و *p* 2 به ترتیب احتمالات کلاس 1 و 2 هستند. از آنجایی که این یک مسئله دو کلاسه است *p* 1 + *p* 2 = 1 و بنابراین معادله.  [14. 1](#bookmark686) را می‌توان به‌طور معادل 2 *p* 1 *p* 2 نوشت. به راحتی می‌توان دید که شاخص جینی زمانی که هر یک از احتمالات کلاس به سمت صفر هدایت می‌شود، به حداقل می‌رسد، به این معنی که گره نسبت به یکی از کلاس‌ها خالص است. برعکس، شاخص جینی زمانی به حداکثر می‌رسد که *p* 1 = *p* 2 , موردی که در آن گره کمترین خلوص را دارد.

هنگام کار با یک پیش‌بینی پیوسته و یک پاسخ طبقه‌بندی، فرآیند یافتن نقطه تقسیم بهینه مشابه فرآیندی است که در بخش دیدیم.  [8. 1 .](#bookmark392) ابتدا نمونه‌ها بر اساس مقادیر پیش‌بینی آنها مرتب می‌شوند. سپس نقاط تقسیم، نقاط میانی بین هر مقدار پیش‌بینی منحصربه‌فرد هستند. اگر پاسخ باینری باشد، این فرآیند یک جدول احتمالی 2 *× 2 را در هر نقطه تقسیم ایجاد می‌کند.* این جدول را می‌توان به صورت کلی نشان داد

|  |  |  |  |
| --- | --- | --- | --- |
| کلاس 1 کلاس 2 | | | |
| *>* تقسیم | *n* 11 | *n* 12 | *n* +1 |
| *<* شکافتن | *n* 21 | *n* 22 | *n* +2 |
|  | *n* 1+ | *n* 2+ | *n* |

شاخص جینی قبل از تقسیم خواهد بود

*جینی (*قبل از تقسیم) = 2

و شاخص جینی را می‌توان پس از تقسیم در هر یک از موارد جدید محاسبه کرد

and 2 n—nn^} n+2 n+2

گره‌هایی با مقادیر 2 *n — ] n — ] n* +1 *n* +1

for greater than and

به ترتیب کمتر یا مساوی با تقسیم. این مقادیر با استفاده از نسبت نمونه‌ها در هر بخش از تقسیم به‌عنوان وزن با ( *n* +1) و (*n* ^) ترکیب می‌شوند که وزن‌های مربوطه را برای بزرگتر و کمتر یا مساوی تقسیم می‌کنند. پس از کمی ساده‌سازی، شاخص جینی برای ارزیابی تقسیم به صورت زیر خواهد بود:

*جینی (*پس از تقسیم) = 2 [( *n il }* f + ( *n* 21 ) < *^* 22^'.

*nn* +1 *nn* +2

حال به مثال ساده ارائه شده در شکل توجه کنید.  [14. 1 ،](#bookmark686) که در آن ­جدول اقتضایی برای تقسیم پیش‌بینی B به شرح زیر است:

کلاس 1 کلاس 2

*B>* 0. 197 91 30 121

*B <* 0. 197 20 67 87

شاخص جینی برای نمونه‌ها در *B >* 0. 197 تقسیم 0. 373 و برای نمونه‌های با *B <* 0 *خواهد بود.* 197 0. 354 خواهد بود. برای تعیین اینکه آیا این یک تقسیم کلی خوب است، این مقادیر باید با هم ترکیب شوند که با وزن دادن هر مقدار خلوص با نسبت نمونه‌ها در گره نسبت به تعداد کل نمونه‌ها در گره والد انجام می‌شود. در این مورد، وزن برای *B>* 0. تقسیم 197 زمانی 0. 582 و 0. 418 خواهد بود که *B <* 0 *باشد.* 197. اندازه‌گیری کلی شاخص جینی برای این تقسیم ( 0. 582) ( *0.* 373) + ( *0.* 418) ( *0.* 354) = 0 *خواهد بود.* 365.

در اینجا ما فقط یک نقطه تقسیم ممکن را ارزیابی کرده ایم. با این حال، الگوریتم‌های پارتیشن‌بندی تقریباً تمام نقاط تقسیم را ارزیابی می‌کنند[[48]](#footnote-48) و مقدار نقطه تقسیم را انتخاب کنید که معیار خلوص را به حداقل می‌رساند. فرآیند تقسیم در هر پارتیشن تازه ایجاد شده ادامه می‌یابد، بنابراین عمق درخت افزایش می‌یابد، تا زمانی که معیارهای توقف برآورده شود (مانند حداقل تعداد نمونه در یک گره یا حداکثر عمق درخت).

درختانی که برای داشتن حداکثر عمق ساخته شده اند به دلیل بیش برازش داده‌های آموزشی بدنام هستند. درخت قابل تعمیم تر درختی است که نسخه هرس شده درخت اولیه است و می‌تواند با تنظیم هزینه-پیچیدگی تعیین شود که در آن معیار خلوص با ضریب تعداد کل گره‌های انتهایی درخت جریمه می‌شود. عامل پیچیدگی هزینه، پارامتر پیچیدگی نامیده می‌شود و می‌تواند در فرآیند تنظیم گنجانده شود تا بتوان یک مقدار بهینه را تخمین زد. جزئیات بیشتر در مورد این فرآیند را می‌توان در بخش یافت.  [8. 1 .](#bookmark392)

پس از هرس درخت، می‌توان از آن برای پیش‌بینی استفاده کرد. در طبقه بندی، هر گره پایانه بردار احتمالات کلاس را بر اساس مجموعه آموزشی تولید می‌کند که سپس به‌عنوان پیش‌بینی برای یک نمونه جدید استفاده می‌شود. در مثال ساده بالا، اگر یک نمونه جدید مقدار Predictor *B* = 0 *را داشته باشد.* 10، سپس بردار احتمال کلاس پیش‌بینی شده به ترتیب برای کلاس 1 و کلاس 2 (0. 23، 0. 77 *)* خواهد *بود.*

مشابه درختان رگرسیون، درختان طبقه‌بندی می‌توانند داده‌های از دست رفته را مدیریت کنند. در ساخت درخت فقط نمونه‌هایی با اطلاعات گم نشده برای ایجاد شکاف در نظر گرفته می‌شوند. در پیش‌بینی، می‌توان از تقسیم‌های جایگزین به جای تقسیم‌هایی که داده‌های گمشده برای آن وجود دارد استفاده کرد. به همین ترتیب، اهمیت متغیر را می‌توان برای درختان طبقه‌بندی با ارزیابی بهبود کلی در معیارهای بهینه‌سازی برای هر پیش‌بینی محاسبه کرد. بخش را ببینید.  [8. 1](#bookmark392) برای توضیح موازی در رگرسیون.

هنگامی که پیش‌بینی پیوسته است، فرآیند پارتیشن‌بندی برای تعیین نقطه تقسیم بهینه ساده است. وقتی پیش‌بینی مقوله‌ای باشد، فرآیند می‌تواند چند مسیر به همان اندازه قابل توجیه را طی کند که یکی از آنها ­با رویکرد مدل‌سازی آماری سنتی متفاوت است. به‌عنوان مثال، یک مدل رگرسیون لجستیک را در نظر بگیرید که شیب‌ها و وقفه‌های مرتبط با پیش‌بینی‌کننده‌ها را تخمین می‌زند. برای پیش‌بینی‌کننده‌های طبقه‌بندی، مجموعه‌ای از متغیرهای ساختگی باینری (بخش. ­ [3. 6 )](#bookmark169) ایجاد شده است که دسته‌ها را به بیت‌های مستقل از اطلاعات تجزیه می‌کند. سپس هر یک از این متغیرهای ساختگی به‌طور جداگانه در مدل گنجانده شده است. مدل‌های درختی همچنین می‌توانند پیش‌بینی‌کننده‌های دسته‌بندی را حذف کنند. ارزیابی ­خلوص برای هر یک از این پیش‌بینی‌کننده‌های جدید ساده است، زیرا هر پیش‌بینی دقیقاً یک نقطه تقسیم دارد.

برای مدل‌های درختی، روش تقسیم ممکن است بتواند انشعابات پویاتری از داده‌ها را ایجاد کند، مانند گروه‌هایی از دو یا چند دسته در دو طرف تقسیم. با این حال، برای انجام این کار، الگوریتم باید پیش‌بینی‌کننده‌های طبقه‌بندی را به‌عنوان مجموعه‌ای مرتب از بیت‌ها در نظر بگیرد. بنابراین، هنگام برازش درختان و مدل‌های مبتنی بر قانون، متخصص باید در مورد درمان داده‌های پیش‌بینی طبقه‌ای، انتخابی انجام دهد:

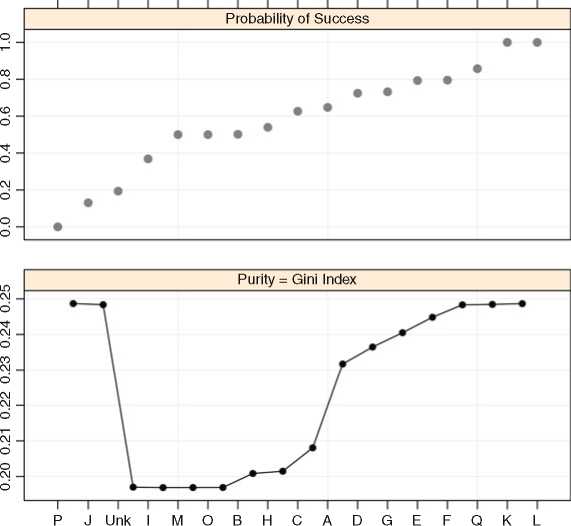
هر پیش‌بینی طبقه‌ای را می‌توان به‌عنوان یک موجودیت واحد وارد مدل کرد تا مدل تصمیم بگیرد که چگونه مقادیر را گروه‌بندی یا تقسیم کند. در متن، به این موضوع با استفاده از *دسته‌های گروه‌بندی اشاره می‌شود.*

پیش‌بینی‌کننده‌های طبقه‌ای ابتدا به متغیرهای ساختگی باینری تجزیه می‌شوند. به این ترتیب، متغیرهای ساختگی به‌دست‌آمده به‌طور مستقل در نظر گرفته می‌شوند و تقسیم‌های باینری را برای دسته‌ها ایجاد می‌کنند. در واقع، تقسیم بر روی یک متغیر ساختگی باینری قبل از مدل‌سازی، تقسیم «یک در مقابل همه» از دسته‌ها را تحمیل می‌کند. این رویکرد با استفاده از *دسته‌های مستقل برچسب‌گذاری می‌شود.*

اینکه کدام رویکرد مناسب تر است به داده‌ها و مدل بستگی دارد. برای مثال، اگر زیرمجموعه‌ای از دسته‌بندی‌ها به شدت پیش‌بینی نتایج باشند، احتمالاً اولین رویکرد بهترین است. با این حال، همانطور که بعدا خواهیم دید، این انتخاب می‌تواند تأثیر قابل‌توجهی بر پیچیدگی مدل و در نتیجه عملکرد داشته باشد. در بخش‌های بعدی، مدل‌هایی با استفاده از *هر دو* رویکردی که در بالا توضیح داده شد، ایجاد می‌شود تا ارزیابی شود که کدام رویکرد سودمند است. خلاصه‌ای از تفاوت‌های این دو رویکرد در شکل 1 خلاصه شده است.  [14. 14](#bookmark729) در ص.  [402](#bookmark729) این باب.

برای نشان دادن فرآیند پارتیشن‌بندی برای یک پیش‌بینی طبقه بندی، مدل CART داده‌های گرنت نشان داده شده در شکل 14. 3 را در نظر بگیرید. اولین تقسیم برای این داده‌ها در باند ارزش قرارداد است که دارای 17 دسته ممکن است و مقادیر I، J، P و Unknown را در یک پارتیشن و دسته‌های باقیمانده را در پارتیشن دیگر قرار می‌دهد. از دیدگاه ترکیبی، با افزایش تعداد دسته‌های ممکن، تعداد سفارش‌های دسته‌بندی ممکن به صورت فاکتوری افزایش می‌یابد. بنابراین رویکرد الگوریتمی باید یک مسیر منطقی اما حریصانه برای مرتب کردن دسته‌ها پیش از تعیین تقسیم بهینه در پیش بگیرد. یک رویکرد این است که دسته‌ها را بر اساس نسبت نمونه‌ها در یک کلاس انتخابی سفارش دهید. نمودار بالایی در شکل [14. 2](#bookmark688) احتمال درخواست موفقیت آمیز گرنت را در هر باند ارزش قرارداد به ترتیب از کم موفق‌ترین به موفق‌ترین نشان می‌دهد. برای محاسبه شاخص جینی، نقاط تقسیم، تقسیم‌بندی بین هر یک از دسته‌های مرتب شده است، با دسته‌های سمت چپ در یک گروه و دسته‌های سمت راست در گروه دیگر قرار می‌گیرند. نتایج حاصل از این پارتیشن‌های متوالی در نمودار پایین ارائه شده است. واضح است که افزودن نمونه‌هایی از دسته ناشناخته به نمونه‌های دسته‌های P و J شاخص جینی را تا حد زیادی کاهش می‌دهد. در حالی که دیدن آن از شکل دشوار است، حداقل مقدار در نقطه تقسیم بین دسته‌های I و M رخ می‌دهد. بنابراین، الگوریتم انتخاب می‌کند که نمونه‌هایی را از باند ارزش قرارداد I، J، P و Unknown در یک پارتیشن و نمونه‌های باقی‌مانده را در پارتیشن دیگر قرار دهد. تنها با استفاده از این تقسیم، مدل یک نمونه جدید را به‌عنوان ناموفق طبقه‌بندی می‌کند اگر دارای باند ارزش قرارداد I , J , P یا Unknown باشد و در غیر این صورت موفق باشد.

ادامه فرآیند درخت‌سازی با پیش‌بینی‌کننده‌ها به‌عنوان دسته‌های گروه‌بندی شده و هرس از طریق پیچیدگی هزینه درخت را در شکل 14. 3 تولید می‌کند. از آنجا که پیش‌بینی‌کننده‌ها کدگذاری شده‌اند، تفسیر درخت بدون دانش عمیق از داده‌ها دشوار است. با این حال، هنوز هم می‌توان از ساختار درختی برای به دست آوردن بینش نسبت به ارتباط پیش‌بینی‌کننده‌ها با پاسخ استفاده کرد. همچنین می‌توانیم متغیرهای دسته‌بندی شده مانند حامی را ببینیم



باند ارزش قرارداد

شکل 14. 2: *بالا* : نمودار پراکندگی احتمال موفقیت مرتب ( محور *y* ) برای هر باند ارزش قرارداد. *پایین* : نمایه شاخص جینی در هر تقسیم سفارشی. شاخص جینی برای نقاط تقسیم بین دسته‌های ناشناخته، I، M، O و B تقریباً معادل است و حداقل بین دسته‌های I و M رخ می‌دهد.

کد، روز هفته و ماه به موفقیت گرنت مرتبط هستند. مدل دسته‌بندی‌شده دارای مساحت زیر منحنی ROC 0. 91 با استفاده از 16 گره پایانه است.

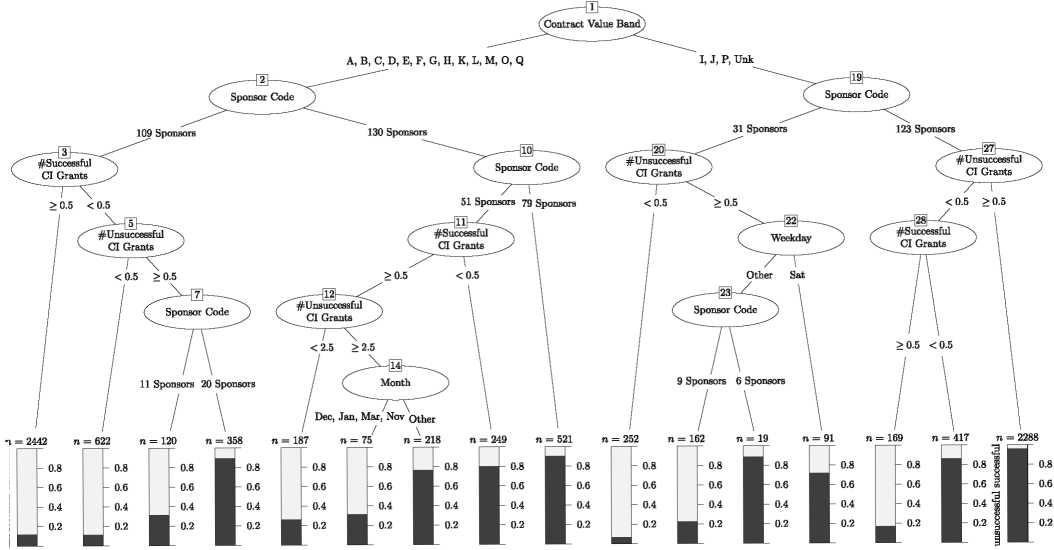
یک مدل CART نیز با استفاده از پیش‌بینی‌کننده‌های دسته‌بندی مستقل ساخته شد. از آنجایی که این رویکرد پیش‌بینی‌کننده‌های بیشتری ایجاد می‌کند، انتظار داریم که درخت هرس شده دارای گره‌های پایانی بیشتری باشد. برخلاف شهود، درخت نهایی هرس شده دارای 16 گره است و در شکل 14. 4 نشان داده شده است. این درخت دارای AUC 0. 912 است و شکل.  [14. 5](#bookmark689) عملکرد خود را با پیش‌بینی‌کننده‌های دسته‌بندی شده مقایسه می‌کند. برای درختان طبقه‌بندی با استفاده از CART، هیچ ­تفاوت عملی در عملکرد پیش‌بینی هنگام استفاده از دسته‌های گروه‌بندی شده یا پیش‌بینی‌کننده‌های دسته‌های مستقل برای داده‌های گرنت وجود ندارد.

مقایسه شکل. 14. 3 و 14. 4 چند شباهت و تفاوت جالب را بین نحوه مدیریت یک مدل درختی دسته‌بندی شده در مقابل پیش‌بینی‌کننده‌های مستقل برجسته می‌کند. ابتدا، توجه داشته باشید که سطوح بالای درختان معمولاً ­با انتخاب هر نوار ارزش قرارداد، کد حامی و یکسان هستند.

**ناموفق موفق**

شکل 14. 3: مدل نهایی CART برای داده‌های گرنت با استفاده از پیش‌بینی‌کننده‌های دسته‌بندی گروه‌بندی شده

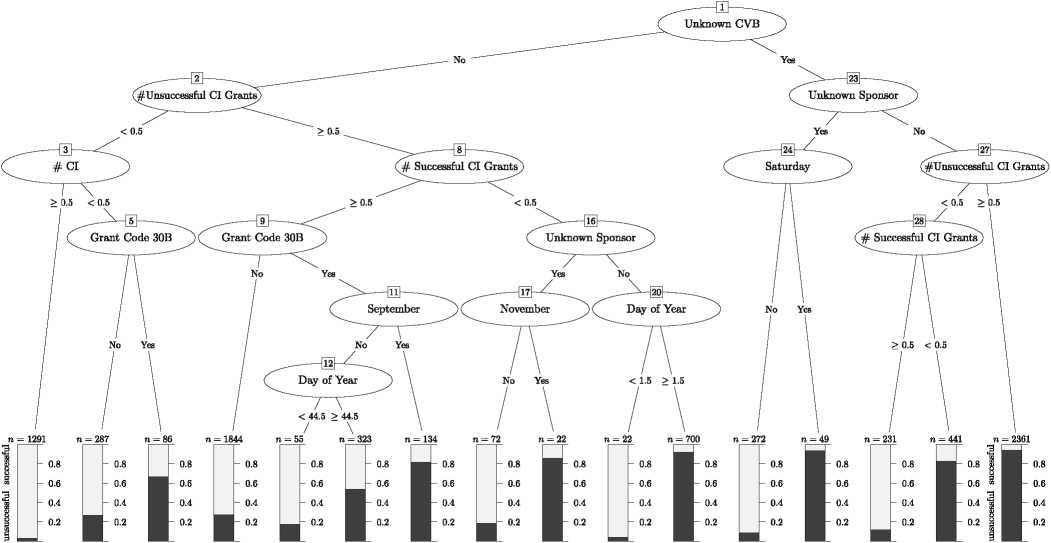
14. 1 درختان طبقه‌بندی پایه 375

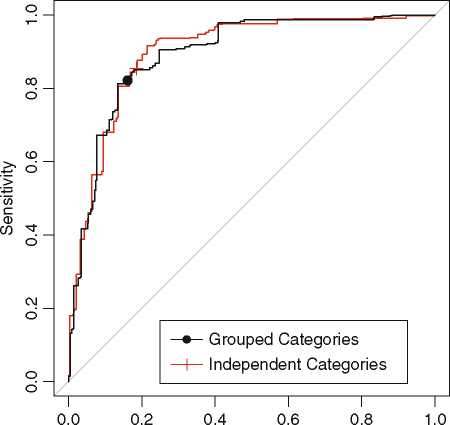


شکل 14. 4: مدل نهایی CART برای داده‌های گرنت

با استفاده از پیش‌بینی‌کننده‌های طبقه‌بندی مستقل

376 14 طبقه‌بندی درختان و مدل‌های مبتنی بر قانون





1 - خاص بودن

شکل 14. 5: منحنی CART ROC برای داده‌های نگهدارنده. هنگام استفاده از دسته‌های گروه‌بندی شده، سطح زیر منحنی 0. 89 بود. با دسته بندی‌های مستقل، AUC نیز 0. 89 بود

تعداد گرنت ناموفق و موفق توسط محققین ارشد در چهار سطح اول. اگرچه درختان اطلاعات مهم مشابهی را شناسایی می‌کنند، تفسیر درخت دسته‌بندی مستقل بسیار آسان‌تر از درخت دسته‌بندی شده است. به‌عنوان مثال، در حالی که پیش‌بینی باند ارزش قرارداد به‌عنوان اولین تقسیم در هر درخت انتخاب می‌شود، درخت دسته مستقل نشان می‌دهد که ارزش Unknown برای ایجاد گره‌های بعدی که خالص تر هستند بسیار حیاتی است. بدون تولید نمودار خلوص دسته‌های مرتب شده، اهمیت باند ناشناخته در گروه‌بندی باندهای I، J، P و Unknown برای درخت دسته‌بندی‌شده پنهان می‌شود. تضادهای مشابهی را می‌توان با پیش‌بینی‌کننده‌های ماه و روز هفته ایجاد کرد، جایی که درخت دسته‌بندی مستقل بینش بیشتری در مورد اهمیت ماه‌ها و روزهای کاری خاص ارائه می‌کند. بنابراین، در مورد درخت‌ها، ایجاد پیش‌بینی‌کننده‌های دسته‌بندی مستقل ممکن است تفسیر ارزشمندی در مورد رابطه بین پیش‌بینی‌کننده‌ها و پاسخی ارائه دهد که هنگام برخورد با پیش‌بینی‌کننده‌ها به‌عنوان دسته‌های گروه‌بندی‌شده به آسانی در دسترس نیست.

رویکرد دیگر برای طبقه‌بندی درختان، مدل C4. 5 است [( کوینلان](#bookmark1024) [1993b](#bookmark1024) ). در اینجا، معیارهای تقسیم بر اساس نظریه اطلاعات است [( والاس](#bookmark1027) [2005](#bookmark1027) ; [کاور و توماس 2006](#bookmark1014) ). فرض کنید می‌خواهیم اطلاعاتی مانند توزیع احتمال کلاس‌ها در ­گره پایانی یک درخت را در یک سری پیام به هم منتقل کنیم. اگر توزیع احتمال بسیار نامتعادل باشد، احتمال زیادی وجود دارد که نمونه به کلاس اکثریت تعلق داشته باشد، بنابراین عدم قطعیت در هنگام حدس زدن کمتر است. با این حال، اگر احتمالات کلاس در گره زوج باشد، عدم قطعیت بالایی در مورد کلاس واقعی یک نمونه وجود دارد. اگر می‌خواهیم محتوای توزیع احتمال را در یک سری پیام‌ها به اشتراک بگذاریم، به‌طور متوسط، زمانی که میزان عدم قطعیت در پیام وجود دارد، اطلاعات بیشتری باید منتقل شود.  [شانون](#bookmark1025) [( 1948](#bookmark1025) ) و دیگران نظریه‌ای را برای ارتباط اطلاعات ایجاد کردند. کمیتی که آنها *آمار اطلاعاتی می‌نامند* نشان دهنده میانگین تعداد بیت‌های مورد نیاز برای برقراری ارتباط در یک پیام است.

در زمینه ما، فرض کنید *C* = 2 کلاس وجود دارد و احتمال کلاس اول *p است.* تعریف رسمی آمار اطلاعاتی است

*اطلاعات* = *-* [ *p log* 2 *p* +(1 *- p* ) *log* 2 (1 *- p* )].

هنگامی که *p* = 0، مرسوم است که 0 *log* 2 (0) = 0 داشته باشیم. همانطور که قبلا ذکر شد، واحدها *بیت نامیده می‌شوند.*

برای داده‌های دو کلاس نشان داده شده در شکل.  [14. 1](#bookmark686) ، کلاس‌ها تقریباً یکسان هستند. اگر *p* نسبت نمونه‌ها در کلاس اول باشد، *p* = 0 *است.* 53. از این، میانگین تعداد بیت‌های اطلاعات برای حدس زدن کلاس واقعی (یعنی اطلاعات) 0. 997 خواهد بود. حال یک وضعیت نامتعادل را در نظر بگیرید که در آن تعداد کمتری از نمونه‌ها در کلاس 1 بودند ( 0. 10 = *p* ). در این مورد، اطلاعات 0. 46 بیت خواهد بود که کوچکتر است زیرا عدم تعادل کلاس، حدس زدن تصادفی کلاس واقعی را آسان تر می‌کند. [[49]](#footnote-49) این معیار قبلاً دو بار مورد بحث قرار گرفته است: به‌عنوان یک تابع هدف برای شبکه‌های عصبی (معادل [13. 3 )](#bookmark622) و رگرسیون لجستیک (در معادله [12. 1](#bookmark21) با یک نقطه داده واحد).

این چه ارتباطی با تعیین انشعابات دارد؟ با استفاده از نماد جدول احتمالی کلی از بالا، کل محتوای اطلاعاتی داده‌ها قبل از تقسیم خواهد بود

*اطلاعات (*قبل از تقسیم) = *-* [ *n i+ x log 2 (* *n i+* }] *-* [ *n* 2+ *x log 2 (* *n 2 ^* }].

دوباره، زمانی که *n* 1+ =0 یا *n* 2+ = 0، سنتی است که عبارات داخل براکت‌ها را صفر می‌کنیم.

ما می‌توانیم بهبود معیارهای اطلاعاتی را که با ایجاد شکاف‌ها در درخت طبقه‌بندی القا می‌شود، اندازه‌گیری کنیم. به *دست آوردن اطلاعات*[[50]](#footnote-50) (یا به سادگی *سود* ) خواهد بود

*افزایش (* *تقسیم* ) = *اطلاعات (*قبل از تقسیم) *- inf o (*پس از تقسیم).

تقسیم‌بندی‌هایی با دستاوردهای اطلاعاتی بیشتر جذاب‌تر از مواردی هستند که سود کمتری دارند.

برای تقسیم باینری که در جدول بالا نشان داده شده است، اطلاعات پس از تقسیم، مجموع مقادیر اطلاعات هر یک از پارتیشن‌های به دست آمده است. به‌عنوان مثال، اطلاعات برای داده‌ها با مقادیر بیشتر از مقدار تقسیم است

*inf o*(greater) = *-*

*— x log2 n —  
n*+1 *n*+1

n12

*x log*2

n+1

('—2') ~ .  
n+1

فرمول داده‌های طرف دیگر تقسیم مشابه است. کل اطلاعات پس از تقسیم، میانگین وزنی این مقادیر است که در آن وزن‌ها به تعداد نمونه‌های موجود در برگ‌های شکاف مربوط می‌شود.

*اطلاعات (*پس از تقسیم) = —+ 1 *اطلاعات (*بزرگتر) +— *+2 اطلاعات (*کمتر از).

با بازگشت به داده‌های دو کلاس، تقسیم پیش‌بینی *B* را با مقدار 0. 197 در نظر بگیرید. اطلاعات زمانی که *B >* 0 *باشد.* 197 0. 808 است و در طرف دیگر تقسیم، مقدار 0. 778 است که با نسبت نمونه‌ها در هر طرف تقسیم وزن می‌شود، کل اطلاعات 0. 795 است، افزایش 0. 997 *-* 0. 795 = 0. 201، از طرف دیگر، فرض کنید، تقسیم دیگری انتخاب شده است که کاملاً غیر اطلاعاتی است، اطلاعات پس از تقسیم مانند قبل از تقسیم خواهد بود، بنابراین سود صفر خواهد بود.

برای پیش‌بینی‌کننده‌های پیوسته، یک درخت می‌تواند با جستجوی پیش‌بینی و تقسیم منفرد ساخته شود که سود اطلاعات را به حداکثر می‌رساند. [[51]](#footnote-51) برای این داده‌ها، این افزایش در هنگام تقسیم پیش‌بینی *B* در 0. 197 بزرگترین است و این شکاف نشان‌داده‌شده در شکل است.  [14. 1 .](#bookmark686) همچنین مشخص می‌شود که این تقسیم بهترین تقسیم برای معیار Gini است که توسط CART استفاده می‌شود.

این استراتژی یک مسأله دارد. از آنجایی که پیش‌بینی‌کننده‌ها ممکن است تعداد متفاوتی ­از مقادیر ممکن داشته باشند، معیار به دست آوردن اطلاعات در برابر پیش‌بینی‌کننده‌هایی که تعداد زیادی پیامدهای ممکن دارند (یعنی، ­پیش‌بینی‌کننده‌های طبقه‌بندی را با تنها چند مقدار متمایز نسبت به پیش‌بینی‌کننده‌های پیوسته ترجیح می‌دهند) بایاس دارد. این پدیده شبیه به بایاس قبلاً مورد بحث برای درختان رگرسیون در بخش است.  [8. 1 .](#bookmark392) در این مورد، بایاس به توانایی الگوریتم برای تقسیم پیش‌بینی‌کننده‌های طبقه‌بندی به روش‌های مختلف مربوط می‌شود (به‌جای تقسیم باینری در پیش‌بینی‌کننده‌های پیوسته). انشعابات چند طرفه احتمالاً سود بیشتری خواهند داشت. برای تصحیح بایاس، از *نسبت بهره* استفاده می‌شود که ­بهره را با اندازه‌گیری مقدار اطلاعات موجود در خود تقسیم تقسیم می‌کند. [Quinlan ( 1993b](#bookmark1024) ) نمونه‌های اضافی از این محاسبات را نشان می‌دهد در حالی که Quin ­lan [( 1996b](#bookmark1024) ) اصلاحات این روش را برای پیش‌بینی‌کننده‌های پیوسته با استفاده از اصل حداقل طول توصیف (MDL) توصیف می‌کند.

هنگام ارزیابی تقسیم‌بندی‌های پیش‌بینی‌کننده‌های طبقه‌بندی، یک استراتژی این است ­که پیش‌بینی را با استفاده از تقسیم‌بندی‌های چند طرفه به‌گونه‌ای که یک تقسیم جداگانه برای هر دسته وجود داشته باشد، نشان دهیم. هنگامی که یک پیش‌بینی تعداد زیادی از مقادیر ممکن را داشته باشد، این می‌تواند به درختان بیش از حد پیچیده منجر شود. برای مثال، پیش‌بینی کد حامی در داده‌های گرنت دارای 298 مقدار منحصربه‌فرد است. اگر این پیش‌بینی مهم در نظر گرفته می‌شد، یک تقسیم اولیه ۲۹۸ جهتی از داده‌ها ایجاد می‌شد (قبل از هرس). پس از فرآیند هرس شرح داده شده در زیر، برخی از این تقسیم‌بندی‌ها احتمالاً ترکیب و ساده می‌شوند.

فصل 7 از [کوینلان](#bookmark1024) [( 1993b](#bookmark1024) ) یک رویکرد اصلاح شده را برای ایجاد ­تقسیمات چند طرفه توصیف می‌کند که توانایی گروه‌بندی دو یا چند دسته را دارند. قبل از ارزیابی یک پیش‌بینی طبقه‌بندی به‌عنوان یک متغیر تقسیم، مدل ابتدا نسبت بهره را زمانی که پیش‌بینی به صورت زیر نمایش داده می‌شود، برمی‌شمارد:

یک تقسیم چند طرفه با تعداد زیادی تقسیم به اندازه مقادیر متمایز (یعنی رویکرد پیش فرض که در آن هر دسته یک تقسیم جداگانه است).

تقسیم‌های چند طرفه برای همه ترکیب‌های ممکن زمانی که دو دسته با هم گروه‌بندی می‌شوند و بقیه به‌طور جداگانه تقسیم می‌شوند.

بر اساس نتایج این نمایش‌های پیش‌بینی، از یک الگوریتم حریص ­برای یافتن بهترین دسته‌ها برای ادغام استفاده می‌شود. در نتیجه، نمایش‌های احتمالی زیادی از پیش‌بینی مقوله‌ای وجود دارد. هنگامی که مدل گروه بندی‌های نهایی را ساخت، نسبت بهره برای این پیکربندی محاسبه می‌شود. هنگام جستجوی بهترین متغیر تقسیم، این نسبت با سایر پیش‌بینی‌کننده‌ها مقایسه می‌شود. این فرآیند هر بار که مدل جستجوی یک متغیر تقسیم جدید را انجام می‌دهد، تکرار می‌شود. این گزینه از نظر محاسباتی گران است و اگر پیش‌بینی‌کننده‌های طبقه‌بندی فقط چند سطح ممکن را داشته باشند، ممکن است تأثیر حداقلی بر درخت داشته باشد. متأسفانه این گزینه در اجرای C4. 5 که در حال حاضر موجود است (در مجموعه نرم‌افزار Weka با نام J48 ) موجود نیست. تأثیر این گزینه روی داده‌ها را نمی‌توان مستقیماً در اینجا نشان داد، اما بعداً هنگام توصیف C5. 0 (نخست C4. 5 ) نشان داده می‌شود. از آنجایی که این می‌تواند تأثیر عمیقی بر مدل داشته باشد، برای متمایز کردن نسخه‌ها، این نسخه از C4. 5 را با عنوان J48 برچسب گذاری می‌کنیم.

هنگام ساخت درخت‌هایی با مجموعه‌های آموزشی حاوی مقادیر پیش‌بینی گمشده، C4. 5 چندین تنظیمات را در فرآیند آموزش انجام می‌دهد:

هنگام محاسبه به دست آوردن اطلاعات، آمار اطلاعات ­با استفاده از داده‌های گم نشده محاسبه می‌شود و سپس با کسری از داده‌های گم نشده در تقسیم مقیاس‌بندی می‌شود.

به یاد بیاورید که C4. 5 با تعدیل آمار بهره با مقدار اطلاعات برای پیش‌بینی، با بایاس انتخاب سروکار دارد. هنگامی که پیش‌بینی حاوی مقادیر گم شده باشد، تعداد شاخه‌ها یک عدد افزایش می‌یابد. داده‌های از دست رفته به‌عنوان یک دسته یا مقدار "اضافی" پیش‌بینی در نظر گرفته می‌شوند.

در نهایت، زمانی که توزیع کلاس برای تقسیم‌های حاصل تعیین می‌شود، مقادیر پیش‌بینی گمشده به صورت *کسری* به هر کلاس کمک می‌کنند. سهم ­کسری از نقاط داده بر اساس توزیع کلاس از مقادیر غیر از دست رفته است. به‌عنوان مثال، فرض کنید 11 نمونه در حال تقسیم شدن هستند و یک مقدار گم شده است. اگر سه نمونه کلاس #1 و بقیه کلاس #2 باشند، مقدار از دست رفته 0. 30 به کلاس #1 و 0. 70 به کلاس #2 (در هر دو طرف تقسیم) کمک می‌کند.

به دلیل این حسابداری، توزیع فرکانس کلاس در هر گره ممکن است شامل اعداد کامل نباشد. همچنین تعداد خطاها در گره ترمینال می‌تواند کسری باشد.

مانند CART، C4. 5 یک درخت بزرگ می‌سازد که احتمالاً بیش از حد داده‌ها را تطبیق می‌دهد و سپس درخت را با دو استراتژی مختلف هرس می‌کند:

حذف ساده یک درخت فرعی

*بلند کردن* یک درخت فرعی به‌طوری که جایگزین گره‌ای بالاتر از درخت شود.

در حالی که CART از هرس پیچیدگی هزینه استفاده می‌کند، هرس *بدبینانه* ارزیابی می‌کند که آیا درخت باید ساده شود یا خیر. موردی را در نظر بگیرید که یک درخت فرعی کاندید حذف است. هرس بدبینانه تعداد خطاها را با و بدون درخت فرعی تخمین می‌زند. با این حال، به خوبی شناخته شده است که میزان خطای ظاهری بسیار خوش بینانه است. برای مقابله با این، هرس بدبینانه یک حد اطمینان بالای تعداد خطاها را محاسبه می‌کند - این *تخمین بدبینانه* از تعداد خطاها است. این با و بدون درخت فرعی محاسبه می‌شود. اگر تعداد تخمینی خطاهای بدون درخت فرعی کمتر از درختی باشد که شامل آن می‌شود، درخت فرعی از مدل هرس می‌شود.

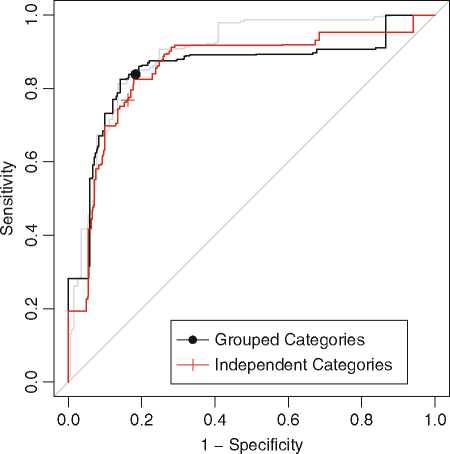
هنگام تعیین میزان خطای تخمینی، C4. 5 از یک سطح اطمینان پیش‌فرض برای فاصله 0. 25 (به نام *ضریب اطمینان* ) استفاده می‌کند. این را می‌توان ­یک پارامتر تنظیم برای مدل در نظر گرفت، زیرا افزایش ضریب اطمینان منجر به درختان بزرگتر می‌شود. این رویکرد در حالی که شهودی است، بر پایه‌های آماری متزلزلی استوار است.  [کوینلان](#bookmark1024) [( 1993b](#bookmark1024) ) این را تصدیق می‌کند و می‌گوید که رویکرد

«به مفاهیم آماری نمونه‌گیری و محدودیت‌های اطمینان خشونت می‌کند، بنابراین استدلال را باید با کمی نمک در نظر گرفت».

گفته می‌شود، این تکنیک می‌تواند بسیار موثر باشد و از نظر محاسباتی کارآمدتر از استفاده از اعتبارسنجی متقاطع برای تعیین اندازه مناسب درخت است.

هنگامی که درخت رشد کرد و هرس شد، یک نمونه جدید با حرکت در مسیر مناسب تا رسیدن به گره پایانه طبقه‌بندی می‌شود. در اینجا، کلاس اکثریت برای داده‌های مجموعه آموزشی که در گره ترمینال قرار می‌گیرند برای پیش‌بینی یک نمونه جدید استفاده می‌شود. یک *مقدار اطمینان،* مشابه یک ­احتمال کلاس، همچنین می‌تواند بر اساس فرکانس‌های کلاس مرتبط با گره‌های پایانه محاسبه شود.  [کوینلان](#bookmark1024) [( 1993b](#bookmark1024) ) توضیح می‌دهد که چگونه محدوده‌های بالا و پایین برای عوامل اطمینان را می‌توان از محاسبات مشابه الگوریتم هرس بدبینانه که در بالا توضیح داده شد به دست آورد.

هنگام *پیش‌بینی* نمونه‌ای با یک یا چند مقدار گمشده، نمونه دوباره به صورت کسری رفتار می‌شود. هنگامی که یک تقسیم برای متغیری که در داده‌ها وجود ندارد مواجه می‌شود، هر مسیر ممکن به پایین درخت تعیین می‌شود. به‌طور معمول، کلاس پیش‌بینی‌شده براساس کلاسی است که بیشترین فرکانس را از یک گره پایانی دارد. از آنجایی که مقدار گمشده ممکن است در بیش از یک گره ترمینال قرار گیرد، هر کلاس یک رأی وزنی برای تعیین کلاس پیش‌بینی شده نهایی دریافت می‌کند. وزن کلاس برای همه مربوطه



شکل 14. 6: منحنی‌های J48 ROC برای داده‌های نگهدارنده با استفاده از دو روش برنامه متفاوت ­برای مدیریت پیش‌بینی‌کننده‌های طبقه‌بندی می‌شوند. نمادها ( *دایره پر شده* و *بعلاوه* ) برش احتمال 50 درصد را نشان می‌دهند. نواحی زیر منحنی‌ها هنگام استفاده از دسته بندی‌های گروه‌بندی شده 0. 835 و هنگام استفاده از گروه‌های گربه مستقل ­0. 842 بود. خط *خاکستری* مربوط به مدل قبلی CART است

گره‌های پایانه جمع می‌شوند و کلاس مرتبط با بیشترین وزن کل برای پیش‌بینی نمونه استفاده می‌شود. به این ترتیب، هر گره پایانه با ارتباط احتمالی با نمونه به پیش‌بینی کلی کمک می‌کند.

درختان J48 برای داده‌های درخواست گرنت ایجاد شدند. اگرچه ­ضریب اطمینان را می‌توان به‌عنوان یک پارامتر تنظیم در نظر گرفت، تجربه ما این است که مقدار پیش فرض (0. 25) به خوبی کار می‌کند. دو مدل با استفاده از دو ­رویکرد متفاوت برای نمایش پیش‌بینی‌کننده‌های طبقه‌بندی مناسب بودند. بر اساس بحث قبلی، این انتظار وجود دارد که برخورد مقوله‌ها به‌عنوان یک مجموعه منسجم منجر به درختی بسیار بزرگ‌تر از درختی با استفاده از دسته‌های مستقل شود. این دقیقاً در مورد این داده‌ها صادق است. گروه‌بندی دسته‌ها منجر به یک درخت هرس شده با 2918 گره پایانی شد. این در درجه اول به دلیل تعداد زیادی تقسیم با استفاده از کد حامی بود. 2,384 تقسیم از 2,918 (82 %) در ­این پیش‌بینی است. هنگام استفاده از دسته‌های مستقل، درخت بسیار کوچکتر بود (821 گره ترمینال).

سطح زیر منحنی ROC برای مدل بزرگ 0. 835 بود، در مقایسه با 0. 842 هنگام استفاده از دسته‌های مستقل. شکل [14. 6](#bookmark692) دو منحنی ROC و نقاط روی هر منحنی را نشان می‌دهد که مربوط به ­قطع احتمالی پیش‌فرض 50 درصد است. از این رو، واضح است که ویژگی‌ها برای هر رویکرد تقریباً یکسان است (81. 7٪ برای مدل بزرگتر در مقابل 83. 8٪)، اما یک

تفاوت قابل‌توجهی در حساسیت مدل ها؛ مدل پیچیده تر منجر به حساسیت 83. 9٪ شد در حالی که مدل طبقه مستقل توانایی نسبتاً ضعیفی برای پیش‌بینی گرنت موفق (با حساسیت 76. 8٪) داشت. با این حال، این آمار بر اساس قطع اسمی 50 درصد برای موفقیت است. منحنی‌ها به‌طور قابل‌توجهی همپوشانی دارند و برش‌های متناوب نتایج تقریباً یکسانی را ایجاد می‌کنند (به بخش مراجعه کنید.  [16. 4 )](#bookmark757) .

در حالی که درختان طبقه‌بندی CART و C4. 5 بیشترین استفاده را دارند، تحقیقات گسترده‌ای در این زمینه و بسیاری از پیشنهادات دیگر برای ­مدل‌های مبتنی بر درخت انجام شده است. به‌عنوان مثال، همانطور که در بخش درختان رگرسیون مورد بحث قرار گرفت، درختان استنتاج مشروط [( Hothorn et al. 2006](#bookmark1018) ) از بایاس انتخاب در طول تقسیم جلوگیری می‌کنند. همچنین چندین تکنیک وجود دارد [(](#bookmark1013) [فرانک و همکاران 1998](#bookmark1016) ; [Loh 2002](#bookmark1021) ; چان و لو [2004](#bookmark1013) ; [زایلیس و همکاران 2008](#bookmark1028) ) که از مدل‌های پیچیده‌تری در گره‌های ترمینال استفاده می‌کنند، مشابه M5 و Cubist. انواع دیگر اسپلیت را می‌توان به کار گرفت. مثلا، [بریمن و همکاران ( 1984](#bookmark1012) ) ایده تقسیم بر روی یک ترکیب خطی از پیش‌بینی‌کننده‌ها را معرفی کرد. این *درختان مایل* ممکن است زمانی مفید باشند که طبقات به صورت خطی قابل تفکیک باشند که تقریب آنها در تقسیم بندی‌های سنتی مسأله است.  [منزه و همکاران ( 2011](#bookmark1022) ) مدل‌های مجموعه درختی را با درختان مورب مورد بحث قرار می‌دهد.

مدل‌های مبتنی بر قانون

همانطور که قبلاً بحث شد، مدل‌های مبتنی بر قانون از یک یا چند عبارت شرطی مستقل تشکیل شده‌اند. بر خلاف درختان، یک نمونه ممکن است از روی مجموعه‌ای از قوانین پیش‌بینی شود. قوانین به‌عنوان طبقه‌بندی‌کننده سابقه طولانی دارند و در این بخش روش‌هایی برای ایجاد قوانین طبقه‌بندی بحث می‌شود.

قوانین C4. 5

چندین فلسفه و الگوریتم مختلف برای ایجاد مدل‌های مبتنی بر قانون از درختان طبقه‌بندی وجود دارد. برخی از اولین‌ها توسط [کوینلان](#bookmark1023) [( 1987](#bookmark1023) ) و [کوینلان](#bookmark1024) [( 1993b](#bookmark1024) ). این مدل که C4. 5Rules نام دارد، بر اساس روش درختی C4. 5 است که در بخش آخر توضیح داده شد. برای شروع، یک درخت هرس نشده ایجاد می‌شود، سپس هر مسیری که از درخت می‌گذرد به یک قانون منفرد تبدیل می‌شود.

با توجه به این مجموعه اولیه، هر قانون به صورت جداگانه ارزیابی می‌شود تا ارزیابی شود که آیا می‌توان آن را با حذف عبارات در عبارت شرطی تعمیم داد. فرآیند هرس در اینجا شبیه به هرس درختان C4. 5 است. برای یک قاعده، مدل ابتدا یک نرخ خطای بدبینانه پایه را محاسبه می‌کند، سپس هر یک از شرایط موجود در قانون را به صورت مجزا حذف می‌کند. پس از حذف یک شرط، میزان خطای بدبینانه مجددا محاسبه می‌شود. اگر هر میزان خطا از خط پایه کوچکتر باشد، شرط مربوط به کمترین میزان خطا حذف می‌شود. این فرآیند تا زمانی تکرار می‌شود که همه شرایط بالاتر از نرخ پایه باشند یا همه شرایط حذف شوند. در مورد دوم، قانون به‌طور کامل از مدل هرس می‌شود. جدول زیر فرآیند هرس را با قانون پنج شرط برای داده‌های گرنت نشان می‌دهد:

میزان خطای بدبینانه

|  |  |  |  |
| --- | --- | --- | --- |
| شرایط. شرط | پاس 1 پاس 2 | | پاس 3 |
| *خط پایه* | *14. 9* | *5. 8* | *5. 2* |
| روز اول سال | 12. 9 | 5. 2 |  |
| صفر گرنت ناموفق (CI) | 77. 3 | 53. 5 | 50. 7 |
| تعداد CI | 42. 0 | 21. 6 | 19. 7 |
| تعداد SCI | 18. 0 | 8. 1 | 7. 3 |
| گرنت موفق صفر (CI) | 5. 8 |  |  |

در اولین گذر، حذف شرط مرتبط با عدم موفقیت گرنت توسط یک محقق ارشد کمترین تأثیر را بر میزان خطا دارد، بنابراین این شرط از قانون حذف می‌شود. سه بار هرس مورد نیاز بود تا اینکه هیچ یک از نرخ‌های خطا کمتر از نرخ پایه نبود. همچنین توجه داشته باشید که با هر تکرار میزان خطای بدبینانه کاهش می‌یابد. در نهایت، به نظر می‌رسد که شرط مربوط به صفر بودن گرنت ناموفق برای یک محقق ارشد، بیشترین اهمیت را برای این قانون داشته باشد، زیرا نرخ خطا زمانی که شرط از قانون حذف می‌شود، بزرگترین است.

پس از هرس کردن شرایط *در* هر قانون، مجموعه قوانین مرتبط با هر کلاس به‌طور جداگانه پردازش می‌شوند تا قوانین را کاهش داده و سفارش دهند. ابتدا، قوانین زائد یا ناکارآمد با استفاده از اصل MDL حذف می‌شوند ­[نگاه کنید به [Quinlan and Rivest ( 1989](#bookmark1024) ) و Chap. 5 از [کوینلان](#bookmark1024) [( 1993b](#bookmark1024) )]. یک معیار MDL ایجاد می‌شود که عملکرد و پیچیدگی یک مجموعه قوانین را در بر می‌گیرد - برای دو مجموعه قوانین با عملکرد معادل، مجموعه ساده‌تری از قوانین مورد علاقه معیار است. در هر کلاس، یک گروه اولیه از گروه‌ها جمع‌آوری می‌شود ­به‌طوری که هر نمونه مجموعه آموزشی حداقل تحت پوشش یک قانون قرار می‌گیرد. اینها در مجموعه قوانین اولیه ترکیب می‌شوند. با شروع این مجموعه، روش‌های جستجو (مانند تپه‌نوردی حریصانه یا بازپخت شبیه‌سازی‌شده) برای افزودن و حذف قوانین استفاده می‌شود تا زمانی که امکان بهبود بیشتر در مجموعه قوانین وجود نداشته باشد. دومین عملیات اصلی در یک کلاس این است که قوانین را از دقیق‌ترین به کمترین دقت مرتب کنیم.

هنگامی که مجموعه قوانین در هر کلاس نهایی شد، کلاس‌ها بر اساس دقت مرتب می‌شوند و یک کلاس پیش فرض برای نمونه‌هایی انتخاب می‌شود که قوانین مرتبطی ندارند. هنگام پیش‌بینی یک نمونه جدید، هر قانون به ترتیب ارزیابی می‌شود تا زمانی که یکی از آنها ارضا شود. کلاس پیش‌بینی شده با کلاس اولین قانون فعال مطابقت دارد.

1 تکرار

یک درخت طبقه‌بندی هرس شده ایجاد کنید

مسیر درختی را با بیشترین پوشش مشخص کنید

این مسیر را به‌عنوان یک قانون به مجموعه قوانین اضافه کنید

نمونه‌های مجموعه آموزشی تحت پوشش قانون را حذف کنید

6 تا زمانی که *تمام نمونه‌های مجموعه آموزشی تحت یک قانون قرار گیرند*

الگوریتم 14. 1: الگوریتم PART برای ساختن مبتنی بر قانون

مدل‌ها [( فرانک و ویتن 1998](#bookmark1016) )

*بخش*

قوانین C4. 5 از این فلسفه پیروی می‌کند که مجموعه اولیه قوانین کاندید به‌طور همزمان توسعه داده می‌شوند و سپس به یک مدل بهبودیافته پس از پردازش تبدیل می‌شوند. روش دیگر، ­قوانین را می‌توان به صورت تدریجی ایجاد کرد. به این ترتیب، یک قانون جدید می‌تواند با مجموعه قوانین قبلی سازگار شود و ممکن است روندهای مهم در داده‌ها را به‌طور موثرتری ثبت کند.

فرانک و ویتن [( 1998](#bookmark1016) ) مدل قانون دیگری به نام PART را که در الگوریتم نشان داده شده است، توصیف می‌کنند [14. 1 .](#bookmark699) در اینجا، یک درخت C4. 5 هرس شده از داده‌ها ایجاد می‌شود و مسیر عبور از درخت که بیشترین نمونه‌ها را پوشش می‌دهد، به‌عنوان یک قانون حفظ می‌شود. نمونه‌های تحت پوشش قانون از مجموعه داده‌ها حذف می‌شوند و این فرآیند تا زمانی تکرار می‌شود که همه نمونه‌ها حداقل توسط یک قانون پوشش داده شوند. اگرچه مدل از درخت برای ایجاد قوانین استفاده می‌کند، اما هر قانون به‌طور جداگانه ایجاد می‌شود و آزادی بالقوه بیشتری برای انطباق با داده‌ها دارد.

مدل PART برای داده‌های گرنت اندکی مدل دسته‌بندی شده را ترجیح داد. برای این مدل، نتایج بهبودی بالاتر و فراتر از مدل‌های قبلی را نشان نمی‌دهند: حساسیت برآورد شده 77. 9 درصد، ­ویژگی 80. 2 درصد و سطح زیر منحنی ROC (نشان داده نشده) 0. 809 بود. این مدل شامل 360 قانون بود. از این تعداد، 181 کمک بلاعوض را به‌عنوان موفق طبقه‌بندی می‌کنند در حالی که 179 دیگر گرنت را به‌عنوان ناموفق طبقه‌بندی می‌کنند. در اینجا، پنج ­پیش‌بینی مؤثر عبارتند از کد حامی (332 قانون)، باند ارزش قرارداد (30 قانون)، تعداد گرنت ناموفق توسط محققین ارشد (27 قانون)، تعداد ­گرنت موفق توسط محققین ارشد (26 قانون) و تعداد محققان ارشد (23 قانون).

درختان کیسه ای

کیسه‌بندی برای طبقه‌بندی یک اصلاح ساده برای کیسه‌بندی برای رگرسیون است (بخش.  [8. 4 )](#bookmark411) . به‌طور خاص، درخت رگرسیون در الگوریتم [8. 1](#bookmark413) با یک درخت طبقه‌بندی نشده هرس نشده برای مدل‌سازی کلاس‌های *C جایگزین می‌شود.* مانند رگرسیون

جدول 14. 1: ماتریس سردرگمی مجموعه Holdout 2008 برای مدل جنگل تصادفی

|  |  |  |
| --- | --- | --- |
|  | کلاس مشاهده شده | |
| موفقیت آمیز | ناموفق |
| موفقیت آمیز | 491 | 144 |
| ناموفق | 79 | 843 |

این مدل دارای دقت کلی 85. 7 درصد، حساسیت 86. 1 درصد و ویژگی 85. 4 درصد است.

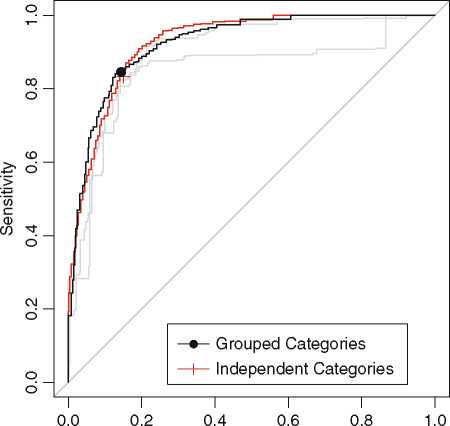
تنظیمات، هر مدل در مجموعه برای پیش‌بینی کلاس نمونه جدید استفاده می‌شود. از آنجایی که هر مدل وزن برابری در مجموعه دارد، هر مدل را می‌توان به‌عنوان رأی دادن به کلاسی که فکر می‌کند نمونه جدید متعلق به آن است، در نظر گرفت. سپس تعداد کل آرا در هر کلاس بر تعداد کل مدل‌های مجموعه ( *M* ) تقسیم می‌شود تا یک بردار احتمال پیش‌بینی‌شده برای نمونه تولید شود. سپس نمونه جدید در گروهی که بیشترین رای و در نتیجه بیشترین احتمال را دارد طبقه‌بندی می‌شود.

برای داده‌های گرنت، مدل‌های بسته‌بندی با استفاده از هر دو استراتژی برای پیش‌بینی‌کننده‌های طبقه‌بندی شده ساخته شدند. همانطور که در فصل درختان رگرسیون بحث شد، عملکرد کیسه‌بندی اغلب با حدود 50 درخت در فلات است، بنابراین 50 درخت به‌عنوان تعداد درخت برای هر یک از این مدل‌ها انتخاب شد. شکل [14. 7](#bookmark703) عملکرد گروه کیسه‌بندی را با استفاده از دسته‌های مستقل یا گروه‌بندی شده نشان می‌دهد. هر دوی این منحنی‌های ROC صاف‌تر از منحنی‌های تولید شده با درخت‌های طبقه‌بندی یا J48 هستند که نشان‌دهنده توانایی کیسه‌بندی در کاهش واریانس از طریق مجموعه است. علاوه بر این، هر دو مدل کیسه‌بندی دارای AUC بهتری (0. 92 برای هر دو) نسبت به مدل‌های قبلی هستند. برای این داده‌ها، به نظر می‌رسد هیچ تفاوت آشکاری در عملکرد برای بسته‌بندی هنگام استفاده از ­دسته‌های مستقل یا گروه‌بندی‌شده وجود ندارد. منحنی‌ها، حساسیت‌ها و ویژگی‌های ROC تقریباً یکسان هستند. عملکرد مجموعه نگهدارنده در شکل.  [14. 7](#bookmark703) بهبودی را نسبت به نتایج J48 نشان می‌دهد (شکل 1).  [14. 6 )](#bookmark692) .

مشابه با تنظیم رگرسیون، معیارهای اهمیت متغیر را می‌توان ­با جمع کردن مقادیر اهمیت متغیر از درختان منفرد در مجموعه محاسبه کرد. اهمیت متغیر 16 پیش‌بینی برتر برای ­مجموعه مدل‌های بسته‌بندی شده مستقل و گروه‌بندی شده در شکل 1 ارائه شده است.  [14. 15](#bookmark738)  و مقایسه این نتایج در تمرین به خواننده واگذار شده است [14. 1 .](#bookmark737)

جنگل‌های تصادفی

جنگل‌های تصادفی برای طبقه‌بندی به یک تغییر ساده در الگوریتم رگرسیون جنگل تصادفی نیاز دارد (الگوریتم [8. 2 )](#bookmark421) : به جای درخت طبقه‌بندی استفاده می‌شود



1 - خاص بودن

شکل 14. 7: منحنی‌های ROC برای مدل درخت طبقه‌بندی کیسه ای. مساحت زیر منحنی‌ها برای هر دو مدل 0. 92 بود. حساسیت و ویژگی به ترتیب 98/82 و 71/85 بود

یک درخت رگرسیون مانند کیسه‌بندی، هر درخت در جنگل برای طبقه‌بندی یک نمونه جدید رای می‌دهد و نسبت آرا در هر کلاس در سراسر مجموعه، بردار احتمال پیش‌بینی‌شده است.

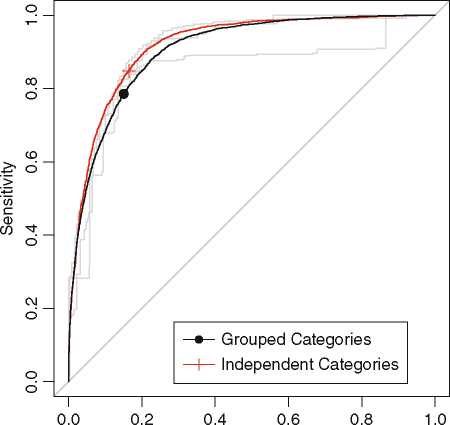
در حالی که نوع درخت در الگوریتم تغییر می‌کند، پارامتر تنظیم تعداد پیش‌بینی‌کننده‌های تصادفی انتخاب شده برای انتخاب در هر تقسیم یکسان است (با *m try نشان داده می‌شود* ). همانند رگرسیون، ایده پشت پیش‌بینی‌کننده‌های نمونه‌گیری تصادفی در طول آموزش، همبستگی بین درختان جنگل است. برای مسائل طبقه بندی، [بریمن](#bookmark1012) [( 2001](#bookmark1012) ) توصیه می‌کند که *m try* را روی جذر تعداد پیش‌بینی‌کننده‌ها تنظیم کنید. برای تنظیم *m try،* توصیه می‌کنیم با پنج مقدار شروع کنید که تا حدودی به‌طور مساوی در محدوده 2 تا *P قرار دارند،* جایی که *P* تعداد پیش‌بینی‌کننده‌ها است. ما به همین ترتیب توصیه می‌کنیم که با مجموعه‌ای از 1000 درخت شروع کنید و اگر عملکرد هنوز به یک فلات نزدیک نشده است، این تعداد را افزایش دهید.

در بیشتر موارد، جنگل تصادفی برای طبقه‌بندی دارای خواص بسیار مشابه ­با آنالوگ رگرسیون است که قبلاً مورد بحث قرار گرفت، از جمله:

مدل نسبتاً به مقادیر *m try حساس نیست.*

مانند بسیاری از درختان، نیازهای پیش پردازش داده حداقل است.

معیارهای عملکرد خارج از کیسه را می‌توان محاسبه کرد، از جمله دقت، حساسیت، ویژگی و ماتریس‌های سردرگمی.



1 - خاص بودن

شکل 14. 8: منحنی‌های ROC برای مدل جنگل تصادفی. سطح زیر منحنی برای دسته‌های مستقل 0. 92 و برای مدل دسته‌بندی گروه‌بندی شده AUC 0. 9 بود.

یکی از تفاوت‌ها توانایی در کلاس‌های وزنی متفاوت است. این جنبه از مدل بیشتر در فصل مورد بحث قرار گرفته است. 16.

مدل‌های جنگل تصادفی بر روی هر دو مدل گربه مستقل و گروه‌بندی شده ساخته شدند. پارامتر تنظیم، *m try،* در مقادیری ­از 5 تا 1000 ارزیابی شد. برای دسته‌های مستقل، مقدار تنظیم شده بهینه *m try* 100 و برای دسته‌های گروه‌بندی شده نیز مقدار 250 بود.  [14. 8](#bookmark704) نتایج را ارائه می‌دهد و در این مورد، دسته‌های مستقل دارای AUC کمی بالاتر (0. 92) نسبت به رویکرد دسته‌بندی شده (0. 9) هستند. مدل پیش‌دیکتور باینری ­نیز حساسیت بهتری دارد (86. 1% در مقابل 84. 7%) اما ویژگی کمی بدتر (85. 4% در مقابل 87. 2%) دارد.

برای تک درختان، اهمیت متغیر را می‌توان با تجمیع بهبود در هدف بهینه‌سازی برای هر پیش‌بینی تعیین کرد. برای ­جنگل‌های تصادفی، معیارهای بهبود (پیش‌فرض معمولاً شاخص جینی است) در سراسر مجموعه جمع‌آوری می‌شود تا یک معیار اهمیت متغیر کلی ایجاد کند. از طرف دیگر، تأثیر پیش‌بینی‌کننده‌ها بر مجموعه را می‌توان با استفاده از رویکرد جایگشت محاسبه کرد [( Breiman 2000](#bookmark1012) ) همانطور که در بخش بحث شد.  [8. 5](#bookmark419) . مقادیر ­متغیر متغیر بر اساس بهبود جمع‌آوری شده ­برای داده‌های گرنت برای هر دو نوع پیش‌بینی محاسبه شده است و مهم‌ترین پیش‌بینی‌کننده‌ها در شکل ارائه شده‌اند.  [14. 15 .](#bookmark738) تفسیر در تمرین به خواننده واگذار می‌شود [14. 1 .](#bookmark737)

درختان استنتاج شرطی همچنین می‌توانند به‌عنوان یادگیرنده پایه برای جنگل‌های تصادفی استفاده شوند. اما پیاده‌سازی‌های کنونی این روش ­برای مسائلی که اندازه نسبی داده‌های گرنت هستند، سنگین هستند. مقایسه عملکرد جنگل‌های تصادفی با استفاده از درختان CART و درختان استنتاج شرطی در تمرین بررسی شده است.  [14. 3](#bookmark737) .

14. 5 تقویت

اگرچه قبلاً در مورد تقویت در تنظیمات رگرسیون بحث کرده‌ایم، این روش در ابتدا برای مسائل طبقه‌بندی توسعه داده شد [( والیانت 1984](#bookmark1026) ؛ [کرنز و والیانت 1989](#bookmark1019) ) که در آن بسیاری از طبقه‌بندی‌کننده‌های ضعیف (مثلاً طبقه‌بندی‌کننده‌ای که به‌طور حاشیه‌ای بهتر از تصادفی پیش‌بینی می‌کند) در یک ترکیب شدند. طبقه‌بندی قوی انواع زیادی از الگوریتم‌های تقویت وجود دارد و در اینجا به مهم‌ترین آنها می‌پردازیم.

AdaBoost

در اوایل دهه 1990 چندین الگوریتم تقویت کننده ظاهر شد [(](#bookmark1025) [Schapire](#bookmark1016) [1990](#bookmark1025) ; Fre ­und [1995](#bookmark1016) ) برای اجرای نظریه اصلی.  [فروند و شاپیره](#bookmark1016) [( 1996](#bookmark1016) ) سرانجام اولین اجرای عملی نظریه تقویت را در الگوریتم معروف AdaBoost خود ارائه کردند. یک نسخه بصری در ­الگوریتم ارائه شده است [14. 2 .](#bookmark711)

برای خلاصه کردن الگوریتم، AdaBoost دنباله‌ای از سیفایرهای کلاس ضعیف تولید می‌کند که در هر تکرار الگوریتم بهترین طبقه‌بندی کننده را بر اساس وزن نمونه فعلی پیدا می‌کند. نمونه‌هایی که در تکرار k به اشتباه طبقه‌بندی شده اند در *تکرار (* *k* + 1) وزن بیشتری دریافت می‌کنند، در حالی که نمونه‌هایی که به درستی طبقه‌بندی شده اند وزن کمتری در تکرار بعدی دریافت می‌کنند. این بدان معناست که نمونه‌هایی که طبقه‌بندی آنها دشوار است، وزن‌های بزرگ‌تری دریافت می‌کنند تا زمانی که الگوریتم مدلی را شناسایی کند که به درستی این نمونه‌ها را طبقه‌بندی کند. بنابراین، هر تکرار الگوریتم برای یادگیری ­جنبه‌های متفاوتی از داده‌ها، با تمرکز بر مناطقی که شامل نمونه‌هایی با طبقه‌بندی دشوار هستند، مورد نیاز است. در هر تکرار، *وزن مرحله* بر اساس میزان خطا در آن تکرار محاسبه می‌شود. ماهیت وزن مرحله شرح داده شده در الگوریتم [14. 2](#bookmark711) به این معنی است که مدل‌های دقیق تر دارای مقادیر مثبت بالاتر و مدل‌های دقیق تر دارای ­مقادیر منفی کمتری هستند. [[52]](#footnote-52) سپس توالی کلی طبقه‌بندی‌کننده‌های وزن‌دار در یک مجموعه ترکیب می‌شوند و پتانسیل قوی‌تری برای طبقه‌بندی بهتر از هر یک از طبقه‌بندی‌کننده‌ها دارند.

بگذارید یک کلاس با مقدار +1 و دیگری با مقدار -1 نمایش داده شود

بگذارید هر نمونه وزن اولیه یکسانی داشته باشد (1 */n* )

برای *k =1toK* انجام دهید

یک طبقه‌بندی ضعیف را با استفاده از نمونه‌های وزنی تنظیم کنید و خطای طبقه‌بندی اشتباه مدل *k را محاسبه کنید (* *err k* )

*k* را به صورت ln ((1 *- err k* ) */err k* ) محاسبه کنید.

وزن‌های نمونه را به‌روزرسانی کنید و وزن بیشتری به نمونه‌های پیش‌بینی‌شده اشتباه و وزن کمتری به نمونه‌های پیش‌بینی‌شده درست بدهید.

پایان

پیش‌بینی طبقه‌بندی‌کننده تقویت‌شده را برای هر نمونه با ضرب *مقدار k* امین مرحله در *k* امین مدل پیش‌بینی و جمع کردن این مقادیر در عرض *k محاسبه* کنید. اگر این مجموع مثبت است، نمونه را در کلاس +1 و در غیر این صورت کلاس -1 طبقه‌بندی کنید.

الگوریتم 14. 2: الگوریتم AdaBoost برای مسائل دو کلاسه

تقویت را می‌توان برای هر تکنیک طبقه‌بندی اعمال کرد، اما درختان طبقه‌بندی یک روش محبوب برای تقویت هستند، زیرا می‌توان آنها را با محدود کردن عمق درخت برای ایجاد درختانی با شکاف‌های کم (همچنین به‌عنوان کنده) به یادگیرندگان ضعیف تبدیل کرد.  [بریمن](#bookmark1012) [( 1998](#bookmark1012) ) توضیحی در مورد اینکه چرا ­درختان طبقه‌بندی به ویژه برای تقویت کار می‌کنند، توضیح می‌دهد. از آنجایی که درخت‌های طبقه‌بندی یک تکنیک بایاس کم/واریانس بالا هستند، مجموعه درختان به کاهش واریانس کمک می‌کند و نتیجه‌ای را ایجاد می‌کند که دارای بایاس کم و واریانس کم است. کار از طریق لنز الگوریتم AdaBoost، [جانسون و راینز ( 2007](#bookmark1019) ) نشان دادند که روش‌های واریانس کم را نمی‌توان تا حد زیادی از طریق تقویت بهبود بخشید. بنابراین، روش‌های تقویتی مانند LDA یا *K* NN به اندازه روش‌های تقویت‌کننده مانند شبکه‌های عصبی ( Freund and Schapire [1996](#bookmark1016) ) یا بیز ساده [( Bauer and Kohavi 1999](#bookmark1011) [) بهبودی نشان نمی‌دهند](#bookmark1016) .

افزایش گرادیان تصادفی

همانطور که در بخش ذکر شد.  [8. 6](#bookmark425) ، [فریدمن و همکاران ( 2000](#bookmark1016) ) برای ارائه ­بینش آماری از الگوریتم AdaBoost کار کرد. برای مسئله طبقه‌بندی، آنها نشان دادند که می‌توان آن را به‌عنوان یک مدل افزودنی مرحله‌ای رو به جلو تفسیر کرد که تابع تلفات نمایی را به حداقل می‌رساند. این چارچوب به تعمیم‌های الگوریتمی مانند Real AdaBoost، Gentle AdaBoost و LogitBoost منجر شد. متعاقباً، این تعمیم‌ها در چارچوب یکپارچه‌ای به نام ماشین‌های تقویت گرادیان قرار گرفتند که قبلاً در فصل درختان رگرسیون مورد بحث قرار گرفت.

1 همه پیش‌بینی‌کننده‌ها را برای نمونه log-shans اولیه کرد: *fi 0 ' 1 =* log 1^.

برای *تکرار j* = 1. *. . M* انجام دهید

باقیمانده (یعنی گرادیان) *z i* = *y i — p i را محاسبه کنید*

به‌طور تصادفی از داده‌های آموزشی نمونه برداری کنید

یک مدل درختی را در زیر مجموعه تصادفی با استفاده از باقیمانده‌ها به‌عنوان نتیجه آموزش دهید

تخمین گره پایانی باقیمانده‌های پیرسون را محاسبه کنید:

\_ 1 /n £ n ( y i - P i )

r i = 1 /n £ n P i (1 - P i )

( j )

مدل فعلی را با استفاده از *f i* = *f i* + *Xf i به روز کنید*

پایان

الگوریتم 14. 3: تقویت گرادیان ساده برای طبقه‌بندی (2-کلاس)

مشابه تنظیمات رگرسیون، زمانی که از درختان به‌عنوان یادگیرنده پایه استفاده می‌شود، تقویت گرادیان پایه دارای دو پارامتر تنظیم است: عمق درخت (یا عمق *تعامل ) ­*و تعداد تکرارها. یک فرمول از مدل‌های تقویت گرادیان تصادفی، احتمال رویداد را شبیه به آنچه در رگرسیون لجستیک دیدیم، توسط

P = 1 .

1 + *گسترش* [ *-f (* *x* )]

که در آن *f (* *x* ) یک پیش‌بینی مدل در محدوده [ *—تا، تا* ] است. برای مثال، تخمین اولیه مدل می‌تواند شانس ثبت نمونه باشد، *fi 0 ' 1* = log *1 — p که* در آن *p* نسبت نمونه یک کلاس از مجموعه آموزشی است.

با استفاده از توزیع برنولی، الگوریتم تقویت گرادیان تصادفی برای دو کلاس در الگوریتم نشان داده شده است.  [14. 3](#bookmark712) .

کاربر می‌تواند الگوریتم را به‌طور خاص با انتخاب یک ­تابع از دست دادن مناسب و گرادیان مربوطه تنظیم کند [( Hastie et al. 2008](#bookmark1018) ). Shrinkage را می‌توان در مرحله آخر الگوریتم پیاده‌سازی کرد [14. 3](#bookmark712) . علاوه بر این، این ­الگوریتم را می‌توان با افزودن یک طرح نمونه‌گیری تصادفی قبل از اولین مرحله در حلقه For داخلی، در چارچوب تقویت گرادیان تصادفی قرار داد. جزئیات در مورد این فرآیند را می‌توان در بخش یافت.  [8. 6](#bookmark425) .

برای داده‌های گرنت، یک شبکه پارامتر تنظیم ساخته شد که در آن ­عمق تعامل بین 1 تا 9، تعداد درختان از 100 تا 2000 و انقباض از 0. 01 تا 0. 1 متغیر بود. این شبکه برای ساخت یک مدل تقویتی به کار گرفته شد که در آن متغیرهای طبقه‌بندی به‌عنوان ­دسته‌های مستقل و به‌طور جداگانه به‌عنوان دسته‌های گروه‌بندی شده در نظر گرفته شدند. برای مدل مقوله مستقل، سطح بهینه زیر منحنی ROC 0. 94، با عمق اندرکنش 9، تعداد درختان 1300 و انقباض 0. 01 بود. برای مدل دسته‌بندی شده، مساحت بهینه زیر منحنی ROC 0. 92، با عمق اندرکنش 7، تعداد درختان 100 و انقباض 0. 01 بود (نگاه کنید به

شکل.  [14. 9 )](#bookmark716) . در این حالت، مدل مقوله مستقل بهتر از مدل دسته‌بندی شده بر اساس ROC عمل می‌کند. با این حال، تعداد درختان در هر مدل به‌طور قابل‌توجهی متفاوت بود که به‌طور منطقی از آنجایی که مجموعه پیش‌بینی باینری بزرگ‌تر از دسته‌های گروه‌بندی شده است، نتیجه می‌شود.

بررسی نمایه‌های پارامتر تنظیم برای دسته‌بندی گروه‌بندی‌شده و پیش‌بینی‌کننده‌های دسته‌بندی مستقل که در شکل‌ها نشان داده شده‌اند.  [14. 10](#bookmark716) و [14. 11](#bookmark717) چند تضاد جالب را نشان می‌دهد. اولاً، تقویت پیش‌بینی‌کننده‌های دسته‌بندی مستقل تقریباً به‌طور یکسان عملکرد پیش‌بینی بهتری در بین تنظیمات پارامتر ­تنظیم نسبت به تقویت پیش‌بینی‌کننده‌های دسته‌بندی‌شده دارد. این الگو احتمالاً به این دلیل است که فقط یک مقدار برای بسیاری از پیش‌بینی‌کننده‌های دسته‌بندی مهم حاوی اطلاعات پیش‌بینی معنادار است. بنابراین، درختانی که از پیش‌بینی‌کننده‌های دسته‌بندی مستقل استفاده می‌کنند، راحت‌تر می‌توانند آن اطلاعات را ­به سرعت پیدا کنند که سپس فرآیند تقویت را هدایت می‌کند. در پیش‌بینی‌کننده‌های دسته‌بندی گروه‌بندی‌شده، افزایش پارامتر انقباض تقریباً به‌طور یکنواخت ­عملکرد پیش‌بینی را در عمق درخت درجه‌بندی می‌کند. این نتایج نشان می‌دهد که برای پیش‌بینی‌کننده‌های دسته‌بندی‌شده، تقویت بیشتر اطلاعات پیش‌بینی خود را ­از یک درخت اولیه با اندازه متوسط به دست می‌آورد که با AUCهای قابل مقایسه بین یک درخت واحد (0. 89) و درخت تقویت‌شده بهینه (0. 92) مشهود است.

تقویت پیش‌بینی‌کننده‌های دسته‌بندی مستقل نشان می‌دهد که با افزایش تعداد درختان، عملکرد مدل برای مقادیر کم انقباض بهبود می‌یابد و برای مقادیر بالاتر انقباض کاهش می‌یابد. اما، چه مقدار کمتر یا بالاتر از انقباض انتخاب شود، هر رویکرد اوج عملکرد پیش‌بینی را در ROC تقریباً 0. 94 پیدا می‌کند. این نتیجه برای این داده‌ها نشان می‌دهد که تقویت می‌تواند تنظیمات بهینه را نسبتاً سریع و بدون نیاز به انقباض زیاد پیدا کند.

اهمیت متغیر برای تقویت در تنظیم طبقه‌بندی به روشی مشابه با تنظیم رگرسیون محاسبه می‌شود: در هر درخت در مجموعه، بهبود بر اساس معیارهای تقسیم برای هر پیش‌بینی جمع می‌شود. سپس این مقادیر اهمیت در کل گروه تقویت کننده به‌طور میانگین محاسبه می‌شوند.

14. 6 C5. 0

C5. 0 نسخه پیشرفته‌تری از مدل طبقه‌بندی C4. 5 Quinlan است که دارای ویژگی‌های اضافی مانند افزایش هزینه و نابرابر برای انواع مختلف خطاها است. مانند C4. 5، نسخه‌های مبتنی بر درخت و قانون دارد و بسیاری از الگوریتم‌های اصلی خود را با نسخه‌های قبلی خود به اشتراک می‌گذارد. برخلاف C4. 5 یا Cubist، ادبیات بسیار کمی در مورد بهبودها وجود دارد و توضیحات ما عمدتاً از ارزیابی کد منبع برنامه است که در سال 2011 در دسترس عموم قرار گرفت.

این مدل ویژگی‌ها و گزینه‌های زیادی دارد و بحث ما به چهار بخش جداگانه تقسیم می‌شود: ایجاد یک درخت طبقه‌بندی واحد، cor-

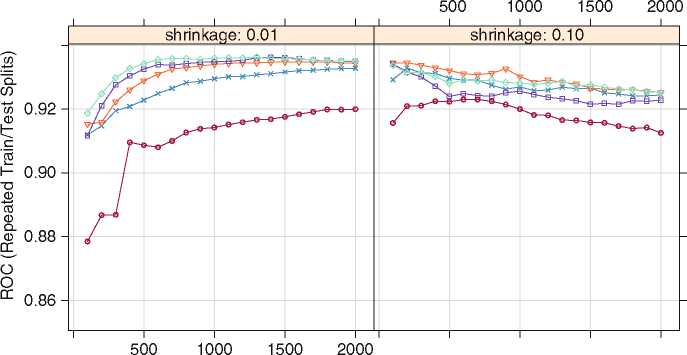
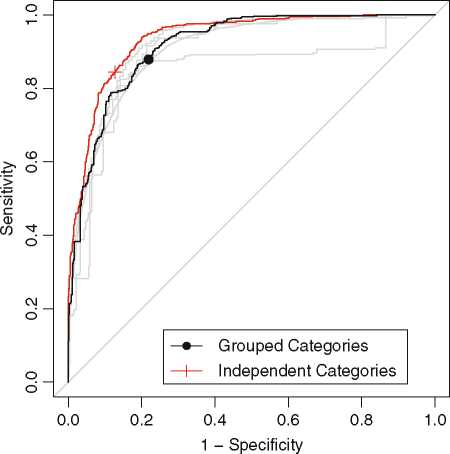
شکل 14. 9: منحنی‌های ROC برای مدل درخت تقویت شده. سطح زیر منحنی برای دسته‌های مستقل 0. 936 و برای مدل دسته‌بندی‌شده AUC 0. 916 بود.

1 o 5 V 9

3 x 7 o

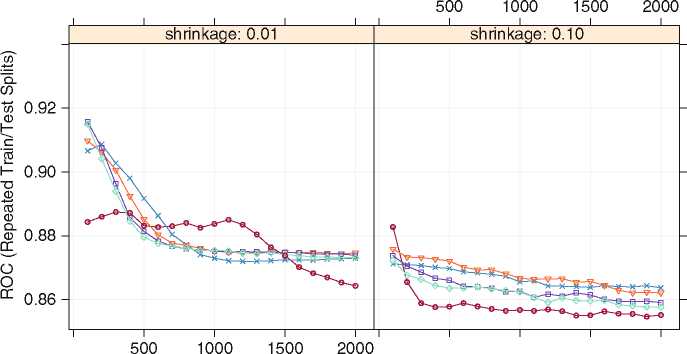
#درختان

شکل 14. 10: تنظیم پروفایل‌های پارامتر برای مدل درختی تقویت شده با استفاده از دسته‌های گروه‌بندی شده



1 o 5 v 9

3 » 7 o



#درختان

شکل 14. 11: تنظیم پروفایل‌های پارامتر برای مدل درختی تقویت شده با استفاده از ­دسته‌های مستقل

پاسخ به مدل مبتنی بر قانون، روش تقویت C5. 0 و ویژگی‌های متفرقه الگوریتم (به‌عنوان مثال، اهمیت متغیر و غیره).

*درختان طبقه بندی*

C5. 0 چندین پیشرفت اساسی دارند که احتمالاً درختان کوچکتری تولید می‌کنند. به‌عنوان مثال، الگوریتم شرایط غیر اتفاقی را برای تقسیم‌ها با چندین دسته ترکیب می‌کند. همچنین یک فرآیند هرس جهانی نهایی را انجام می‌دهد ­که سعی می‌کند درختان فرعی را با رویکرد هزینه-پیچیدگی حذف کند. در اینجا، درخت‌های فرعی حذف می‌شوند تا زمانی که نرخ خطا از یک خطای استاندارد نرخ پایه (یعنی بدون هرس) بیشتر شود. آزمایش اولیه نشان می‌دهد که این روش‌های اضافی تمایل به ایجاد درختان ساده تر از الگوریتم قبلی دارند.

اسمی C5. 0 با داده‌های گرنت با پیش‌دیگرهای طبقه‌ای ­که به‌عنوان مجموعه‌های منسجم در نظر گرفته می‌شوند، مناسب بود. درخت دارای 86 گره پایانی بود و منجر به ایجاد مساحت زیر منحنی ROC 0. 685 شد. پنج پیش‌بینی پرکار در درخت عبارت بودند از: باند ارزش قرارداد (شش تقسیم)، روز عددی سال (شش تقسیم)، کد حامی (پنج تقسیم)، کد دسته (چهار تقسیم) و روز هفته (چهار تقسیم). به یاد بیاورید که درخت J48 مشابه دارای گره‌های ترمینال بسیار بیشتری بود (2918) که در اصل به دلیل نحوه تقسیم‌بندی بر روی ­متغیرهای دسته‌بندی با مقادیر زیادی مانند کد حامی بود. درخت C5. 0 با استفاده از الگوریتم اکتشافی شرح داده شده در بخش از این مسئله جلوگیری می‌کند.  [14. 1](#bookmark686) که تلاش می‌کند دسته‌ها را در دو یا چند گروه کوچکتر ادغام کند. اگر این گزینه در C5. 0 خاموش باشد، درخت بسیار بزرگتر است (213 گره پایانه) به دلیل پیش‌بینی‌کننده‌های طبقه‌بندی شده. با این حال، مساحت زیر منحنی ROC برای درخت بزرگتر (0. 685) تقریباً مشابه درخت کوچکتر است.

هیچ‌یک از مدل‌های C5. 0 به اندازه درخت J48 که قبلاً توضیح داده شد، نزدیک نمی‌شوند. برای J48 و C5. 0 (بدون گروه‌بندی)، پیش‌بینی‌کننده‌های دسته‌بندی با مقادیر زیاد در تقسیم‌های بیشتری استفاده می‌شوند و در هر تقسیم، زمانی که از گزینه گروه‌بندی استفاده نمی‌شود، تمایل به ایجاد بیش از دو شاخه دارند.

*قوانین طبقه بندی*

فرآیند مورد استفاده برای ایجاد قوانین مشابه C4. 5 است. یک درخت اولیه رشد می‌کند، به قوانین تبدیل می‌شود، سپس قوانین فردی از طریق هرس ساده‌سازی می‌شوند و یک رویه کلی در کل مجموعه استفاده می‌شود تا به‌طور بالقوه تعداد قوانین تشکیل دهنده را کاهش دهد. فرآیند هرس کردن شرایط در یک قانون و ساده‌سازی مجموعه قوانین از ­C4. 5 پیروی می‌کند، اما C5. 0 قوانین را ترتیب نمی‌دهد. در عوض، هنگام پیش‌بینی نمونه‌های جدید، C5. 0 از *تمام* قوانین فعال استفاده می‌کند که هر کدام به محتمل‌ترین کلاس رای می‌دهند. آرای هر طبقه با ­مقادیر اطمینان وزن می‌شود و طبقه مرتبط با بالاترین رای استفاده می‌شود. با این حال، ارزش اطمینان پیش‌بینی‌شده، مقداری است که با خاص‌ترین قانون فعال مرتبط است. به یاد بیاورید که C4. 5 قوانین را مرتب می‌کند و از اولین قانون فعال برای پیش‌بینی استفاده می‌کند.

داده‌های گرنت با این الگوریتم تحلیل شد. مدل مبتنی بر قانون شامل 22 قانون با مساحت تخمینی زیر منحنی ROC 0. 675 است. پیچیدگی مدل بسیار ساده تر از PART است. وقتی بر اساس ارزش اطمینان قاعده ترتیب داده شود، سه قانون اصلی برای پیش‌بینی یک گرنت موفق عبارتند از:

(اولین روز سال)

(تعداد محققین ارشد *>* 0) و (تعداد سرپرستان اصلی *<* 0) و (تعداد محققین ارشد دانش آموز *<* 0) و (تعداد گرنت ناموفق توسط محققین ارشد *<* 0) و (کد SEO = 730106) و (روز عددی سال *<* 209)

(تعداد محققین ارشد خارجی *<* 0) و (تعداد بازرسان ارشد متولد حدود 1975 *<* 0) و (تعداد گرنت موفق توسط بازرس ارشد *<* 0) و (روز عددی سال *>* 109) و ( ­نامعلوم کد دسته) و (روز هفته در سه شنبه، جمعه، دوشنبه، چهارشنبه، پنجشنبه)

به‌طور مشابه، سه قانون اصلی برای گرنت ناموفق عبارتند از:

(تعداد گرنت ناموفق توسط محققین ارشد *>* 0) و ( ­تعداد روز سال *>* 327) و (کد حامی در 2B، 4D، 24D، 60D، 90B، 32D، 176D، 7C، 173A، 269A) و ( باند ارزش قرارداد در Unk، J) و (CategoryCode در 10A، 30B، 30D، 30C)

(تعداد محققین ارشد *<* 1) و (تعداد گرنت ناموفق توسط محققین ارشد *>* 0) و (تعداد مقالات مجله B توسط محققین ارشد *>* 3) و (کد حامی = 4D) و (باند ارزش قرارداد در B، Unk، J) و (ماه در نوامبر، دسامبر، فوریه، مارس، می، ژوئن)

(تعداد محققین ارشد *>* 0) و (تعداد بازجویان ارشد ­متولد حدود 1945 *<* 0) و (تعداد گرنت موفق توسط محققین ارشد *<* 0) و (روز عددی سال *>* 209) و (کد حامی در 21A، 60D، 172D، 53A، 103C، 150B، 175C، 93A، 207C، 294B)

11 قانون برای پیش‌بینی گرنت موفق و 11 قانون برای نتایج ناموفق وجود داشت. پیش‌بینی‌کننده‌های درگیر در بیشتر قوانین عبارت بودند از: تعداد گرنت ناموفق توسط محققین ارشد (11 قانون)، باند ارزش قرارداد (9 قانون)، کد دسته (8 قانون)، روز عددی سال (8 قانون) و ماه (5 قانون). قوانین).

C5. 0 دارای ویژگی‌های دیگری برای مدل‌های مبتنی بر قانون است. به‌عنوان مثال، مدل می‌تواند *باندهای کاربردی ایجاد کند.* در اینجا، سودمندی به‌عنوان افزایش خطا که هنگام حذف قانون از مجموعه رخ می‌دهد، اندازه‌گیری می‌شود. قوانین با یک الگوریتم تکراری مرتب می‌شوند: مدل قانون را با کوچکترین ابزار حذف می‌کند و ابزارهای مفید را برای قوانین دیگر دوباره محاسبه می‌کند. ترتیب حذف قوانین اهمیت آنها را مشخص می‌کند. به‌عنوان مثال، اولین قانون حذف شده با کمترین ابزار و آخرین قانون با بالاترین سود مرتبط است. باندها گروهی از قوانین با اندازه تقریباً مساوی بر اساس ترتیب کاربرد (بزرگترین تا کوچکترین) هستند. رابطه بین نرخ خطای تجمعی را می‌توان با اضافه شدن گروه‌هایی از قوانین به مدل مشخص کرد.

*افزایش*

C5. 0 شبیه به الگوریتم AdaBoost است که قبلاً توضیح داده شد در معنای اصلی: مدل‌ها به ترتیب برازش می‌شوند و هر تکرار وزن‌های مورد را بر اساس دقت پیش‌بینی نمونه تنظیم می‌کند. با این حال، برخی از تفاوت‌های قابل‌توجه وجود دارد. ابتدا، C5. 0 تلاش می‌کند تا درخت‌هایی را ایجاد کند که تقریباً به اندازه درخت اول باشند، با وادار کردن درخت‌ها به تعداد تقریباً همان تعداد گره پایانی در هر مورد به‌عنوان درخت اولیه. تکنیک‌های تقویت قبلی، پیچیدگی درخت را به‌عنوان یک پارامتر تنظیم در نظر می‌گرفتند. ثانیاً، این مدل پیش‌بینی‌کننده‌های درختان تشکیل‌دهنده را به‌طور متفاوتی با AdaBoost ترکیب می‌کند. هر مدل تقویت شده مقادیر اطمینان را برای هر کلاس همانطور که در بالا توضیح داده شد محاسبه می‌کند و میانگین ساده‌ای از این مقادیر ­محاسبه می‌شود. کلاس با بیشترین ارزش اطمینان انتخاب می‌شود. وزن مرحله در طول فرآیند آموزش مدل محاسبه نمی‌شود. سوم، C5. 0 دو نوع "تحلیل بیهودگی" را در طول آموزش مدل انجام می‌دهد. اگر مدل بسیار مؤثر باشد (یعنی مجموع وزن‌های نمونه‌های طبقه‌بندی‌شده اشتباه کمتر از 0. 10 باشد) یا اگر بسیار ناکارآمد باشد (مثلاً میانگین وزن نمونه‌های نادرست بیشتر از 50 باشد، مدل به‌طور خودکار تقویت را متوقف می‌کند». ­٪. همچنین، پس از نیمی از تکرارهای تقویتی درخواستی، هر نمونه ارزیابی می‌شود تا مشخص شود که آیا پیش‌بینی صحیح امکان‌پذیر است یا خیر. اگر اینطور نباشد، مورد از محاسبات بعدی حذف می‌شود.

در نهایت، C5. 0 از یک طرح وزنی متفاوت در طول آموزش مدل استفاده می‌کند. ابتدا مقداری نماد:

*N* = اندازه مجموعه آموزشی

*N -* = تعداد نمونه‌های طبقه‌بندی نادرست

*w k* = وزن مورد برای نمونه در *k* امین تکرار تقویتی

*S* + = مجموع اوزان برای نمونه‌هایی که به درستی طبقه‌بندی شده اند

*S -* = مجموع وزن برای نمونه‌های طبقه‌بندی نادرست

الگوریتم با تعیین نقطه میانی بین مجموع وزن‌ها برای نمونه‌های طبقه‌بندی اشتباه و نیمی از مجموع وزن‌ها شروع می‌شود.

*1 . 1* 1

*midpoint = —*

2( *s-* + *s*+) *- s-*

= 4( *s + - s-).*

از این رو، نمونه‌های طبقه‌بندی شده به درستی با معادله تنظیم می‌شوند

*s* + *- نقطه میانی*

kk s +

و نمونه‌های طبقه‌بندی شده با استفاده از به روز رسانی می‌شوند

*نقطه میانی*

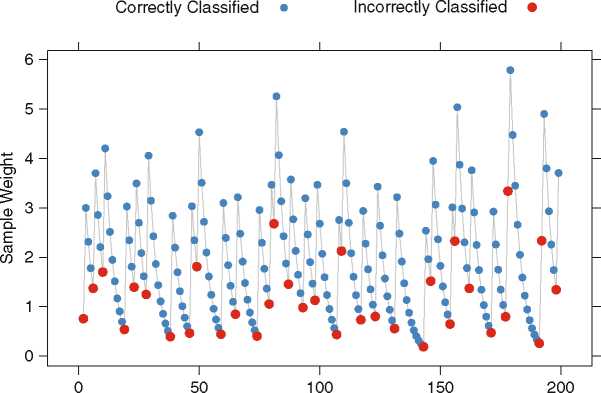
*W k = W k - 1* + *N.*

این طرح به روز رسانی یک جهش مثبت بزرگ در وزن‌ها زمانی که یک نمونه به اشتباه پیش‌بینی می‌شود، می‌دهد. هنگامی که یک نمونه به درستی پیش‌بینی می‌شود، ماهیت ضربی معادله باعث می‌شود وزن‌ها آهسته‌تر و با نرخ کاهشی کاهش یابد، زیرا نمونه به درستی پیش‌بینی می‌شود. شکل [14. 12](#bookmark721) نمونه‌ای از تغییر وزن را برای یک نمونه در چندین تکرار تقویتی نشان می‌دهد.

Quinlan [( 1996a](#bookmark1024) ) چندین آزمایش را با تقویت و بسته‌بندی مدل‌های مبتنی بر درخت توصیف می‌کند، از جمله چندین آزمایش که در آن‌ها تقویت C4. 5 منجر به مدل کمتر مؤثری شد.

*سایر جنبه‌های مدل*

C5. 0 اهمیت پیش‌بینی را با تعیین درصد نمونه‌های مجموعه آموزشی که در تمام گره‌های پایانی پس از تقسیم قرار می‌گیرند، اندازه‌گیری می‌کند. برای



تقویت تکرار

شکل 14. 12: نمونه‌ای از طرح وزن دهی نمونه با استفاده از C5. 0 هنگام تقویت

به‌عنوان مثال، پیش‌بینی در اولین تقسیم به‌طور خودکار دارای اندازه‌گیری اهمیت ­100٪ است زیرا همه نمونه‌ها تحت تأثیر این تقسیم قرار می‌گیرند. سایر پیش‌بینی‌کننده‌ها ممکن است اغلب در تقسیم‌بندی‌ها استفاده شوند، اما اگر گره‌های پایانی تنها تعداد انگشت ­شماری از نمونه‌های مجموعه آموزشی را پوشش دهند، امتیازات اهمیت ممکن است نزدیک به صفر باشد. همین استراتژی برای مدل‌های مبتنی بر قانون و نسخه‌های تقویت‌شده مدل اعمال می‌شود.

C5. 0 همچنین گزینه‌ای برای باز کردن یا حذف پیش‌بینی‌کننده‌ها دارد: یک ریتم الگوی اولیه ­*نشان می‌دهد* که کدام پیش‌بینی‌کننده‌ها با نتیجه ارتباط دارند و مدل نهایی تنها از پیش‌بینی‌کننده‌های مهم ایجاد می‌شود. برای انجام این کار، مجموعه آموزشی به‌طور تصادفی به دو نیم تقسیم می‌شود و درختی به منظور ارزیابی سودمندی پیش‌بینی‌کننده‌ها ایجاد می‌شود (به آن درخت برنده می‌گویند). دو رویه اهمیت هر پیش‌بینی را برای مدل مشخص می‌کند:

پیش‌بینی‌کننده‌ها اگر در هیچ شکافی در درخت برنده نباشند، بی‌اهمیت در نظر گرفته می‌شوند.

نیمی از نمونه‌های مجموعه آموزشی که برای ایجاد درخت برنده گنجانده نشده‌اند، برای تخمین میزان خطای درخت استفاده می‌شوند. نرخ خطا نیز بدون هر پیش‌بینی تخمین زده می‌شود و با نرخ خطا در زمانی که همه پیش‌بینی‌کننده‌ها استفاده می‌شوند، مقایسه می‌شود. اگر نرخ خطا بدون پیش‌بینی بهبود یابد، بی‌ربط تلقی می‌شود و موقتاً حذف می‌شود.

هنگامی که فهرست آزمایشی پیش‌بینی‌کننده‌های غیر اطلاعاتی ایجاد شد، C5. 0 درخت را بازسازی ­می‌کند. اگر میزان خطا بدتر شده باشد، فرآیند winnowing غیرفعال می‌شود و هیچ پیش‌بینی‌ای حذف نمی‌شود.

پس از ایجاد پیش‌بینی‌کننده‌های مهم (در صورت وجود)، فرآیند آموزشی معمولی C5. 0 با مجموعه آموزشی کامل استفاده می‌شود، اما *تنها با* پیش‌بینی‌کننده‌هایی که از روند برنده شدن جان سالم به در برده‌اند.

به‌عنوان مثال، C5. 0 داده‌های گرنت را به قسمت‌های تقریباً مساوی تقسیم کرد، یک درخت روی نیمی از داده‌ها ساخت و از نیمه دوم برای تخمین میزان خطا در حدود 14. 6٪ استفاده کرد. وقتی پیش‌بینی مربوط به تعداد محققین ارشد دانشجو حذف شد، میزان خطا کمی کاهش یافت و به 14. 2 درصد رسید. با توجه به این موضوع، تعداد محققین ارشد دانشجویی از بررسی بیشتر کنار گذاشته شد. در مقابل، زمانی که باند ارزش قرارداد حذف شد، میزان خطا به 24. 8 درصد افزایش یافت. این پیش‌بینی برای مدل‌های بعدی C5. 0 حفظ شد.

*داده‌های اعطایی*

برای داده‌های گرنت، چندین تغییر از مدل C5. 0 مورد ارزیابی قرار گرفت:

مدل‌های تک درختی و مبتنی بر قانون

درخت و قوانین با تقویت (تا 100 تکرار)

همه پیش‌بینی‌کننده‌ها و مجموعه برنده شده

دو رویکردی که قبلاً برای مدیریت ­عوامل پیش‌بینیی طبقه‌ای توضیح داده شد

برای آخرین مجموعه از شرایط مدل، تفاوت بسیار کمی در مدل‌ها وجود داشت. شکل [14. 13](#bookmark728) منحنی‌های ROC را برای دو روش رمزگذاری پیش‌بینی‌کننده‌های طبقه‌بندی نشان می‌دهد. منحنی‌ها تقریباً یکسان هستند.

پانل بالای شکل.  [14. 13](#bookmark728) مشخصات تنظیم را برای مدل‌های C5. 0 با دسته بندی‌های گروه‌بندی شده نشان می‌دهد. هنگامی که الگوریتم winnowing اعمال شد، کاهش جزئی در عملکرد وجود داشت، اگرچه این احتمالاً در نویز تجربی داده‌ها است. افزایش آشکارا تأثیر مثبتی برای این مدل‌ها داشت و پس از حدود 50 تکرار، بهبودی نهایی وجود دارد. اگرچه قوانین منفرد بدتر از درختان منفرد بودند، اما تقویت بیشترین تأثیر را بر روی مدل‌های مبتنی بر قانون داشت که در نهایت عملکرد برتری داشتند. سطح بهینه زیر منحنی ROC برای این مدل 0. 942 بود که بهترین سطح در بین مدل‌ها مشاهده شد.

چه پیش‌بینی‌کننده‌هایی در مدل استفاده شده است؟ اول، ممکن است مفید باشد که بدانید هر یک از پیش‌بینی‌کننده‌ها چند بار در یک قانون در تمام تکرارهای تقویت استفاده شده است. اکثریت وسیعی از پیش‌بینی‌کننده‌ها به ندرت مورد استفاده قرار گرفتند. 99٪ از پیش‌بینی‌کننده‌ها در کمتر از 0. 71٪ از قوانین استفاده شده است. ده پیش‌بینی متداول عبارتند از: باند ارزش قرارداد (9. 2%)، تعداد گرنت ناموفق توسط محققین ارشد (8. 3%)، تعداد گرنت موفق توسط محققین ارشد (7. 1%)، روز عددی سال (6. 3%). کد دسته (6%)، ماه (3. 5%)، روز هفته (3. 1%)، کد حامی (2. 8%)، تعداد محققین ارشد خارجی (1. 1%) و تعداد مقالات ژورنال C توسط رئیس پژوهشگران (0. 9%). همانطور که قبلاً توضیح داده شد، پیش‌بینی‌کننده‌ها را می‌توان بر اساس مقادیر اهمیتشان رتبه‌بندی کرد، همانطور که با درصد کل نمونه‌های ­تحت پوشش پیش‌بینی اندازه‌گیری می‌شود. با افزایش، این معیار کمتر آموزنده است زیرا پیش‌بینی در تقسیم اول 100٪ اهمیت دارد. در این مدل که تعداد قابل‌توجهی از تکرارهای تقویتی استفاده شد، 40 ­پیش‌دیگر دارای مقادیر اهمیت 100 درصد بودند. این مدل تنها از 357 پیش‌بینی (24 درصد) استفاده کرده است.

مقایسه دو رمزگذاری پیش‌بینی‌کننده‌های طبقه‌بندی شده

همه مدل‌های مناسب در این فصل از دو روش برای رمزگذاری ­پیش‌بینی‌کننده‌های طبقه‌بندی استفاده کردند. شکل [14. 14](#bookmark729) نتایج مجموعه نگهدارنده را برای هر مدل و رویکرد نشان می‌دهد. به‌طور کلی، تفاوت زیادی در ناحیه زیر منحنی ROC بین دو کدگذاری مشاهده نشد. J48 حساسیت ­را با متغیرهای ساختگی باینری مجزا از دست داد، در حالی که افزایش گرادیان تصادفی و PART هنگام استفاده از متغیرهای گروه‌بندی شده، ویژگی‌های خود را از دست می‌دهند. در برخی موارد، رمزگذاری‌ها بر پیچیدگی مدل تأثیر داشتند. همانطور که در شکل نشان داده شده است، برای درختان تقویت شده، انتخاب کدگذاری‌ها منجر به پروفایل‌های تنظیم بسیار متفاوتی شد.  [14. 10](#bookmark716) و [14. 11](#bookmark717) . برون یابی این یافته‌ها به مدل‌های دیگر و مجموعه داده‌های دیگر دشوار است و به همین دلیل، ممکن است ارزش داشته باشد که هر دو رمزگذاری را در مرحله آموزش مدل امتحان کنید.

محاسبه

این بخش از توابع بسته‌های زیر استفاده می‌کند: C50، caret، gbm، ipred، partykit، pROC، randomForest و RWeka. این بخش همچنین از همان R ob ject‌های ایجاد شده در بخش استفاده می‌کند.  [12. 7](#bookmark585) که حاوی داده‌های Grant Applications (مانند آموزش قاب داده‌ها ) است.

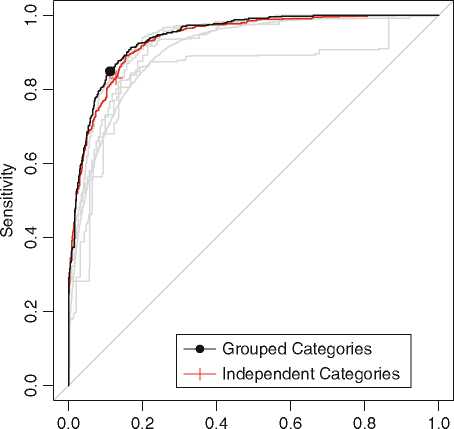
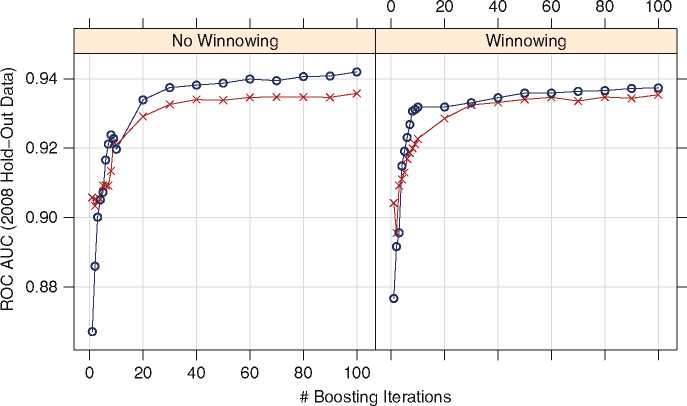
علاوه بر مجموعه‌ای از متغیرهای ساختگی که در بخش توضیح داده شده است.  [12. 7](#bookmark585) ، ­تعدادی از پیش‌بینی‌کننده‌های طبقه‌بندی به‌عنوان فاکتورهای R کدگذاری می‌شوند: SponsorCode، ContractValueBand، CategoryCode و Weekday. هنگام برازش مدل‌هایی با ­دسته‌های مستقل برای این پیش‌بینی‌کننده‌ها، از بردار کاراکتر fullSet استفاده می‌شود. هنگامی که پیش‌بینی‌کننده‌های طبقه‌بندی را به‌عنوان یک مجموعه منسجم در نظر می‌گیریم، فهرست جایگزینی از پیش‌بینی‌کننده‌ها در فاکتور بردار Predictors وجود دارد که شامل ­نسخه‌های فاکتوری از داده‌های مربوطه است. علاوه بر این، رشته کاراکتر factorForm یک فرمول R است که با استفاده از همه پیش‌بینی‌کننده‌های موجود در factorPredictors ایجاد شده است (و بسیار طولانی است).

مقدار زیادی از نحو نشان داده شده در این بخش مشابه سایر ­بخش‌های محاسباتی است، به خصوص قسمت قبلی مربوط به درختان رگرسیون. تمرکز در اینجا بر روی تفاوت‌های ظریف توابع مدل فردی و تفسیر خروجی آنها خواهد بود. برخی از کدها برای بازسازی تحلیل‌ها در این فصل نشان داده شده است.

قوانین نوع مدل   
o درخت x

1 - خاص بودن

شکل 14. 13: *بالا* : مشخصات تنظیم پارامتر برای مدل C5. 0 با استفاده از دسته‌های گروه‌بندی شده. *پایین* : منحنی‌های ROC برای مدل‌های تقویت‌شده C5. 0. نسخه‌های گروه‌بندی‌شده و مستقل مدل تقریباً یکسان هستند، با مساحت زیر منحنی ROC 0. 942





Grouped o

Independent x

Fig. 14.14: The effect of different methods of representing categorical predic­tors in tree- and rule-based models. “Grouped” indicates that the categories for a predictor were treated as a cohesive set, while “independent” indicates that the categories were converted to independent dummy variables prior to modeling

یک برنامه جامع برای مدل‌های نشان داده شده در فهرست فصل بسته AppliedPredictiveModeling موجود است.

*درختان طبقه بندی*

تعدادی بسته R برای ساخت درختان طبقه‌بندی منفرد وجود دارد. بسته اولیه rpart است. همانطور که در رگرسیون بحث شد، تابع فقط از روش فرمول برای مشخص کردن شکل دقیق مدل استفاده می‌کند.

تعداد زیادی پیش‌بینی برای داده‌های گرنت وجود دارد و همانطور ­که قبلا ذکر شد، یک فرمول R به صورت برنامه نویسی برای مدل‌سازی کلاس‌ها برای دسته‌های گروه‌بندی شده ایجاد شد. نحو زیر یک مدل CART را با این پیش‌بینی‌کننده‌ها با استراتژی تقسیم داده ما مطابقت می‌دهد:

*کتابخانه (rpart)*

*cartModel <- rpart(factorForm, data = training[pre2008,])*

این به‌طور خودکار درخت را با استفاده از ­روش اعتبارسنجی متقاطع داخلی رشد می‌دهد و هرس می‌کند. یک استدلال مهم برای طبقه‌بندی parms است. در اینجا، تغییرات متعددی در فرآیند آموزش مدل می‌توان اعلام کرد، مانند احتمالات قبلی نتیجه و نوع تقسیم (اعم از ضریب جینی یا آمار اطلاعات). این مقادیر باید در یک لیست باشند. [[53]](#footnote-53) برای جزئیات بیشتر به ?rpart مراجعه کنید. همچنین، آرگومان کنترلی می‌تواند رویه برازش را از نظر روش‌های عددی (مانند عمق درخت) سفارشی کند.

خروجی مدل تا حدودی با درختان رگرسیون متفاوت است. برای نشان دادن این، یک مدل کوچکتر با دو پیش‌بینی ایجاد می‌کنیم:

*rpart(کلاس ~ NumCI + روز هفته، داده = آموزش[قبل از 2008،])*

n= 6633

گره)، تقسیم، n، از دست دادن، yval، (yprob) \* نشان دهنده گره پایانی است

root 6633 3200 ناموفق (0. 49 0. 51)

روز هفته = یکشنبه 223 0 موفق (1. 00 0. 00) \*

روز هفته = جمعه، دوشنبه، شنبه، پنجشنبه، سه شنبه، چهارشنبه 6410 3000 ناموفق (0. 47 0. 53)

روز هفته = دوشنبه، پنج شنبه، سه شنبه 2342 1000 موفق (0. 57 0. 43) \*

روز هفته = جمعه، شنبه، چهارشنبه 4068 1700 ناموفق (0. 41 0. 59) \*

خروجی متغیر/مقدار تقسیم را به همراه تعداد نمونه در شاخه تقسیم شده نشان می‌دهد (223 برای گره دوم در خروجی بالا). کلاس اکثریت نیز چاپ می‌شود (موفق برای گره 2) و احتمالات کلاس پیش‌بینی شده برای نمونه‌هایی که به این گره ختم می‌شوند.

نحو پیش‌بینی تقریباً مشابه سایر مدل‌ها در R است. تابع پیش‌بینی، به‌طور پیش فرض، احتمالاتی را برای هر کلاس تولید می‌کند. با استفاده از predict(object, type = "class") یک بردار عامل از کلاس برنده تولید می‌شود.

پیاده‌سازی R از C4. 5 در بسته RWeka در تابعی به نام J48 است. تابع همچنین یک فرمول مدل می‌گیرد:

*> کتابخانه (RWeka)*

*> J48 (کلاس ~ NumCI + روز هفته، داده = آموزش[قبل از 2008،])*

درخت J48 هرس شده

روز هفته = جمعه: ناموفق (1422. 0/542. 0)

روز هفته = دوشنبه: موفق (1089. 0/455. 0)

روز هفته = شنبه

| NumCI <= 1: ناموفق (1037. 0/395. 0)

| NumCI > 1

| | NumCI <= 3: ناموفق (378. 0/185. 0)

| | NumCI > 3: موفق (61. 0/26. 0)

روز هفته = یکشنبه: موفق (223. 0)

روز هفته = پنجشنبه

| NumCI <= 0: ناموفق (47. 0/21. 0)

| NumCI > 0: موفق (520. 0/220. 0)

روز هفته = سه شنبه

| NumCI <= 2

| | NumCI <= 0: ناموفق (45. 0/21. 0)

| | NumCI > 0: موفق (585. 0/251. 0)

| NumCI > 2: ناموفق (56. 0/22. 0)

روز هفته = چهارشنبه: ناموفق (1170. 0/521. 0)

Number of Leaves

12

اندازه درخت : 18

به یاد بیاورید که این اجرای C4. 5 تلاشی برای گروه‌بندی دسته‌ها قبل از هرس ندارد. تابع پیش‌بینی به‌طور خودکار ­کلاس‌های برنده را تولید می‌کند و احتمال‌های کلاس را می‌توان از پیش‌بینی (شیء، نوع = "مسائل") به دست آورد.

هنگام تجسم درختان CART یا J48، تابع نمودار از بسته partykit می‌تواند نمایشگرهای دقیق ایجاد کند. اشیاء باید با as. party به کلاس مناسب تبدیل شوند و سپس تابع plot.

یک درخت C5. 0 را می‌توان از بسته C50 ایجاد کرد :

*کتابخانه (C50)*

*C5tree <- C5. 0 (کلاس ~ NumCI + روز هفته، داده = آموزش[قبل از 2008،])*

*C5tree*

صدا زدن:

C5. 0. formula(فرمول = کلاس ~ NumCI + روز هفته، داده = آموزش[قبل از 2008، ])

درخت طبقه بندی

تعداد نمونه: 6633

تعداد پیش‌بینی ها: 2

اندازه درخت: 2

گزینه‌های غیر استاندارد: تلاش برای گروه‌بندی ویژگی ها

*خلاصه (C5tree)*

صدا زدن:

C5. 0. formula(فرمول = کلاس ~ NumCI + روز هفته، داده = آموزش[قبل از 2008، ])

C5. 0 [نسخه 2. 07 GPL Edition] پنجشنبه 6 دسامبر 2012 13:53:14

کلاس مشخص شده توسط ویژگی «نتیجه»

6633 مورد (3 ویژگی) را از undefined. data بخوانید

درخت تصمیم:

روزهای هفته در سه شنبه، دوشنبه، پنجشنبه، یکشنبه: موفقیت آمیز (2565/1010)

روزهای هفته در جمعه، چهارشنبه، شنبه: ناموفق (4068/1678)

ارزیابی داده‌های آموزشی (6633 مورد):

درخت تصمیم

خطاهای اندازه

2 2688(40. 5%) <<

(الف) (ب) <- طبقه‌بندی شده به‌عنوان

1555 1678 (a): کلاس موفق

1010 2390 (ب): کلاس ناموفق

استفاده از ویژگی:

100. 00٪ روز هفته

زمان: 0. 0 ثانیه

توجه داشته باشید که برخلاف J48، این تابع می‌تواند مقادیر روزهای هفته را از گروه‌هایی از مقادیر تقسیم کند. عملکرد کنترل برای این مدل ( C5. 0Control ) این ویژگی را خاموش می‌کند ( زیر مجموعه = FALSE ). گزینه‌های دیگری مانند win ­nowing و ضریب اطمینان برای تقسیم در اینجا موجود است. مانند J48، تابع پیش‌بینی پیش‌فرض کلاس‌ها را تولید می‌کند و type = "prob" احتمالات را تولید می‌کند.

لفاف‌هایی برای این مدل‌ها با استفاده از آموزش تابعی caret وجود دارد. به‌عنوان مثال، برای برازش مدل دسته‌بندی شده برای CART، از:

*> set. seed(476)*

*> rpartGrouped <- train(x = training[,factorPredictors],*

*+ y = آموزش$Class،*

*+ روش = "rpart"،*

*+ آهنگ طول = 30،*

*+ معیار = "ROC"،*

*+ trControl = ctrl)*

به یاد بیاورید که موضوع ctrl مشخص می‌کند که کدام داده‌ها در مجموعه نگهدارنده هستند و چه معیارهای عملکردی باید محاسبه شوند (به‌عنوان مثال، حساسیت، ویژگی و ناحیه زیر منحنی ROC). کدهای مدل درختان J48 و C5. 0 به ترتیب J48 و C5. 0Tree هستند. تفاوت اصلی در اینجا بین آموزش و تابع مدل اصلی، یک رابط یکپارچه برای مدل‌ها و توانایی تنظیم مدل‌ها با معیارهای جایگزین، مانند سطح زیر منحنی ROC است.

توجه داشته باشید که rpart، C5. 0 و J48 از روش فرمول متفاوت از اکثر توابع دیگر استفاده می‌کنند. معمولاً، روش فرمول به‌طور خودکار هر پیش‌بینی طبقه‌ای را به مجموعه‌ای از متغیرهای ساختگی باینری تجزیه می‌کند. این ­توابع به ماهیت طبقه‌بندی داده‌ها احترام می‌گذارند و این پیش‌بینی‌کننده‌ها را به‌عنوان مجموعه‌های گروه‌بندی شده از دسته‌ها در نظر می‌گیرند (مگر اینکه داده‌ها قبلاً به متغیرهای ساختگی تبدیل شده باشند). تابع آموزش از قرارداد رایج‌تر در R پیروی می‌کند، یعنی ایجاد متغیرهای ساختگی قبل از مدل‌سازی. این دلیل اصلی ­است که قطعه کد بالا با روش غیر فرمول هنگام فراخوانی آموزش نوشته شده است.

*قوانین*

چندین مدل مبتنی بر قانون در بسته RWeka وجود دارد. تابع PART مدل‌هایی را بر اساس ایجاد می‌کند [فرانک و ویتن](#bookmark1016) [( 1998](#bookmark1016) ). نحو آن شبیه J48 است :

*PART(کلاس ~ NumCI + روز هفته، داده = آموزش[قبل از 2008،])*

لیست تصمیم بخشی

روز هفته = جمعه: ناموفق (1422. 0/542. 0)

روز هفته = شنبه و

NumCI <= 1: ناموفق (1037. 0/395. 0)

روز هفته = دوشنبه: موفق (1089. 0/455. 0)

روز هفته = پنجشنبه و

NumCI > 0: موفق (520. 0/220. 0)

روز هفته = چهارشنبه: ناموفق (1170. 0/521. 0)

روز هفته = سه شنبه و

NumCI <= 2 و

NumCI > 0: موفق (585. 0/251. 0)

روز هفته = شنبه و

NumCI <= 3: ناموفق (378. 0/185. 0)

روز هفته = یکشنبه: موفق (223. 0)

روز هفته = سه شنبه: ناموفق (101. 0/43. 0)

روز هفته = شنبه: موفق (61. 0/26. 0)

: ناموفق (47. 0/21. 0)

تعداد قوانین: 11

سایر توابع RWeka برای قوانین را می‌توان در صفحه راهنما یافت ?Weka\_ classifier\_rules.

قوانین C5. 0 با استفاده از تابع C5. 0 به روش درختان ایجاد می‌شوند، اما با گزینه قوانین = TRUE :

*قوانین C5 <- C5. 0 (کلاس ~ NumCI + روز هفته، داده = آموزش[قبل از 2008،]،*

*+ قوانین = درست)*

*> قوانین C5*

صدا زدن:

C5. 0. formula(فرمول = کلاس ~ NumCI + روز هفته، داده

= آموزش [قبل از 2008، ]، قوانین = TRUE)

مدل مبتنی بر قانون

تعداد نمونه: 6633

تعداد پیش‌بینی ها: 2

تعداد قوانین: 2

گزینه‌های غیر استاندارد: تلاش برای گروه‌بندی ویژگی ها

*> خلاصه (قوانین C5)*

صدا زدن:

C5. 0. formula(فرمول = کلاس ~ NumCI + روز هفته، داده = آموزش[قبل از 2008، ]، قوانین = درست)

C5. 0 [نسخه 2. 07 GPL Edition] پنجشنبه 6 دسامبر 2012 13:53:14

کلاس مشخص شده توسط ویژگی «نتیجه»

6633 مورد (3 ویژگی) را از undefined. data بخوانید

قوانین:

قانون 1: (2565/1010، آسانسور 1. 2)

روزهای هفته در سه شنبه، دوشنبه، پنج شنبه، یکشنبه

-> کلاس موفقیت آمیز [0. 606]

قانون 2: (4068/1678، lift 1. 1)

روزهای هفته در جمعه، چهارشنبه، شنبه

-> کلاس ناموفق [0. 587]

کلاس پیش فرض: ناموفق

ارزیابی داده‌های آموزشی (6633 مورد):

قوانین بدون خطا

2 2688(40. 5%) <<

(الف) (ب) <- طبقه‌بندی شده به‌عنوان

1555 1678 (a): کلاس موفق

1010 2390 (ب): کلاس ناموفق

استفاده از ویژگی:

100. 00٪ روز هفته

زمان: 0. 0 ثانیه

پیش‌بینی از همان نحوی که در بالا گفته شد پیروی می‌کند. امتیازهای اهمیت متغیر برای درختان و قوانین C5. 0 با استفاده از تابع C5imp یا تابع varImp در بسته caret محاسبه می‌شود.

هنگام کار با عملکرد آموزش، کدهای مدل C5. 0Rules و PART در دسترس هستند.

بسته‌های دیگر برای تک درختان عبارتند از party (درخت استنتاج شرطی)، درخت (درخت CART)، oblique. tree (درختان مایل)، partDSA (برای مدل [Molinaro و همکاران ( 2010](#bookmark1022) )) و evtree (درختانی که با استفاده از الگوریتم‌های ژنتیک توسعه یافته اند. ). دسته دیگری از روش‌های پارتیشن‌بندی که در اینجا مورد بحث قرار نگرفته اند به نام رگرسیون منطقی [( Ruczinski و همکاران 2003](#bookmark1025) ) در چندین بسته از جمله Logi ­cReg پیاده‌سازی شده اند.

*درختان کیسه ای*

بسته اولیه کیسه‌بندی درخت ipred است. تابع bagging نسخه‌های بسته‌بندی شده درختان rpart را با استفاده از روش فرمول ایجاد می‌کند (یک تابع دیگر، ipredbagg، از روش غیر فرمول استفاده می‌کند). نحو آشنا است:

*> بسته‌بندی (کلاس ~ روز هفته + NumCI، داده = آموزش[قبل از 2008،])*

آرگومان nbagg تعداد درختان این مجموعه را کنترل می‌کند (به‌طور پیش فرض 25 درخت). پیش‌فرض برای روش پیش‌بینی استاندارد، تعیین کلاس برنده است و نوع = "prob" احتمالات را تولید می‌کند.

یکی دیگر از عملکردهای بسته مراقبت به نام کیسه، مدل‌های کیف را به‌طور کلی ایجاد می‌کند (یعنی مدل‌هایی غیر از درخت).

*جنگل تصادفی*

پورت R برنامه جنگل تصادفی اصلی در بسته randomForest ­موجود است و نحو اصلی آن با کد درخت رگرسیون نشان داده شده در صفحه یکسان است.  [215](#bookmark448) . مقدار پیش‌فرض *m try «^p* با رگرسیون متفاوت است. یکی از گزینه‌ها، قطع، مخصوص طبقه‌بندی است و برش (های) رای‌گیری را برای تعیین طبقه برنده از مجموعه درختان کنترل می‌کند. این گزینه خاص هنگام استفاده از تابع پیش‌بینی جنگل تصادفی نیز در دسترس است.

مدل فرمول و نحو غیر فرمول را می‌گیرد. در هر صورت، هر پیش‌بینی طبقه‌ای که به‌عنوان متغیرهای فاکتور R کدگذاری می‌شود، به‌عنوان یک گروه در نظر گرفته می‌شود. نحو پیش‌بینی به‌طور پیش‌فرض برای تولید کلاس برنده است، اما آرگومان type به پیش‌بینی مقادیر دیگری مانند احتمالات کلاس ( نوع = "prob" ) یا تعداد واقعی رای نوع = "رای" اجازه می‌دهد.

یک مثال اساسی برای داده‌های گرنت، با خروجی، این است:

*کتابخانه (جنگل تصادفی)*

*randomForest (کلاس ~ NumCI + روز هفته، داده = آموزش[قبل از 2008،])*

صدا زدن:

randomForest (فرمول = کلاس ~ NumCI + روز هفته، داده = آموزش[قبل از 2008، ]) نوع جنگل تصادفی: طبقه بندی

تعداد درخت: 500

تعداد متغیرهای امتحان شده در هر تقسیم: 1

برآورد OOB از میزان خطا: 40. 06٪

ماتریس سردرگمی:

موفق ناموفق class. error

موفق 1455 1778 0. 5499536

ناموفق 879 2521 0. 2585294

از آنجایی که فقط دو پیش‌بینی گنجانده شده است، در هر تقسیم فقط یک پیش‌بینی به‌طور تصادفی انتخاب می‌شود.

این تابع تخمین خطای خارج از کیسه و همچنین ماتریس سردرگمی مشابه را چاپ می‌کند. برآوردهای خارج از کیسه حساسیت و نرخ مثبت کاذب (یعنی 1-ویژگی) در زیر ستون class. error نشان داده شده است.

کد مدل برای تنظیم یک مدل جنگل تصادفی با آموزش "rf" است.

سایر توابع جنگل‌های تصادفی عبارتند از cforest (در بسته حزبی )، obliqueRF (جنگل‌های درختان مایل در بسته اریب )، rFerns (برای مدل سرخس تصادفی از [اوزویسال و همکاران ( 2010](#bookmark1023) ) در بسته rFerns ) و RRF (مدل‌های جنگل تصادفی منظم در بسته RRF ).

*درختان تقویت شده*

بسته درختی تقویت شده اولیه در R gbm است که تقویت گرادیان tic stochas را پیاده‌سازی می‌کند. ­تفاوت اصلی بین تقویت رگرسیون و درختان طبقه بندی، انتخاب توزیع داده‌ها است. تابع gbm فقط می‌تواند دو مسأله کلاس را در خود جای دهد و استفاده از توزیع = "bernoulli" در اینجا انتخاب مناسبی است. گزینه دیگر توزیع = "adaboost" برای تکرار تابع ضرر استفاده شده توسط آن متدولوژی است.

یکی از مسائلی که هنگام استفاده از gbm برای طبقه‌بندی وجود دارد این است که انتظار می‌رود نتیجه به صورت 0/1 کدگذاری شود. نمونه‌ای از یک مدل ساده برای داده‌های گرنت خواهد بود

*کتابخانه (gbm)*

*forGBM <- آموزش*

*forGBM$Class <- ifelse(forGBM$Class == "موفق"، 1، 0)*

*> gbmModel <- gbm(کلاس ~ NumCI + روز هفته،*

*+ داده = forGBM[pre2008،]،*

*+ توزیع = "برنولی"،*

*+ interaction. depth = 9،*

*+ n. trees = 1400،*

*+ انقباض = 0. 01،*

*+ ## این تابع مقادیر زیادی تولید می‌کند*

*## از خروجی به‌طور پیش فرض. پرمخاطب = FALSE)*

تابع پیش‌بینی برای این مدل کلاس برنده را پیش‌بینی نمی‌کند. با استفاده از predict ( gbmModel، type = "response") احتمال کلاس برای کلاس رمزگذاری شده به صورت 1 محاسبه می‌شود (در مثال ما، یک گرنت موفق به صورت 1 کدگذاری شده است). این را می‌توان به یک متغیر عامل با کلاس برنده تبدیل کرد:

*gbmPred <- پیش‌بینی(gbmModel,*

*+ newdata = head(training[-pre2008,])،*

*+ نوع = "پاسخ"،*

*+ ## تعداد درختان باید باشد*

*+ ## به صراحت تنظیم شده است*

*+ n. trees = 1400)*

*gbmPred*

[1] 0. 5697346 0. 5688882 0. 5688882 0. 5688882 0. 5697346 0. 5688882

*gbmClass <- ifelse(gbmPred > 0. 5، "موفق"، "ناموفق")*

*gbmClass <- factor(gbmClass، سطوح = سطوح (training$Class))*

*gbmClass*

[1] موفق موفق موفق موفق موفق موفق

سطوح: موفق ناموفق

تطبیق این مدل با آموزش این فرآیند را به میزان قابل‌توجهی ساده می‌کند. برای ­مثال، یک متغیر عامل می‌تواند به‌عنوان فرمت نتیجه استفاده شود ( آموزش به‌طور خودکار تبدیل را انجام می‌دهد). هنگام پیش‌بینی کلاس برنده، یک فاکتور تولید می‌شود. اگر احتمالات کلاس مورد نیاز است، پیش‌بینی (شیء، نوع = "prob") را مشخص کنید ( تابع پیش‌بینی آموزش به‌طور خودکار از تعداد درخت‌هایی استفاده می‌کند که در طول تنظیم مدل بهینه بودند).

الگوریتم اصلی AdaBoost در بسته ada موجود است. یکی دیگر از عملکردهای تقویت درختان، blackboost در بسته بوست است. این بسته همچنین دارای توابعی برای تقویت انواع دیگر مدل‌ها (مانند رگرسیون لجستیک) مانند بسته bst است.

برای آموزش نسخه‌های تقویت شده C5. 0 از آرگومان آزمایشی (با مقادیر بین 1 تا 100) استفاده می‌شود.

*کتابخانه (C50)*

*C5Boost <- C5. 0 (کلاس ~ NumCI + روز هفته، داده = آموزش[قبل از 2008،]،*

*+ آزمایش = 10)*

*C5Boost*

صدا زدن:

C5. 0. formula(فرمول = کلاس ~ NumCI + روز هفته، داده

= آموزش [قبل از 2008، ]، آزمایشات = 10)

درخت طبقه بندی

تعداد نمونه: 6633

تعداد پیش‌بینی ها: 2

تعداد تکرارهای تقویتی: 10 مورد درخواست. 6 به دلیل توقف زود هنگام استفاده شده است

اندازه درخت متوسط: 2. 5

گزینه‌های غیر استاندارد: تلاش برای گروه‌بندی ویژگی ها

به‌طور پیش‌فرض، الگوریتم دارای تست‌های داخلی است که ارزیابی می‌کند که آیا تقویت مؤثر است یا خیر ­و زمانی که تشخیص دهد که دیگر مؤثر نیست، آموزش مدل را متوقف می‌کند (به این پیام توجه کنید که ده تکرار درخواست شده است اما تنها شش تکرار به دلیل توقف زودهنگام استفاده شده است). این ویژگی را می‌توان با استفاده از C5. 0Control (EarlyStopping = FALSE) رد کرد.

این مدل‌ها را می‌توان با آموزش با استفاده از مقادیر متد gbm , ada , یا C5. 0 تنظیم کرد.

تمرینات

اهمیت متغیر برای کیسه‌بندی، جنگل‌های تصادفی و تقویت هم برای دسته‌های مستقل و هم برای پیش‌بینی‌کننده‌های مدل عامل محاسبه شده است. 16 پیش‌بینی مهم برای هر روش و مجموعه پیش‌بینی در شکل 1 ارائه شده است.  [14. 15](#bookmark738) .

در هر تکنیک مدلسازی، کدام عوامل بین دسته مستقل و مدل‌های عامل مشترک است؟

چگونه این نتایج با پرکارترین پیش‌بینی‌کننده‌های موجود در نتایج مدل PART که در بخش بحث شده است مقایسه می‌شود.  [14. 2 ؟](#bookmark695)

برای داده‌های ریزش توضیح داده شده در تمرین [12. 3](#bookmark31) :

چند درخت اصلی را در مجموعه آموزشی قرار دهید. آیا کد منطقه باید به‌عنوان متغیرهای ساختگی مستقل رمزگذاری شود یا به‌عنوان مجموعه‌ای از مقادیر گروه‌بندی شده باشد؟

آیا بسته‌بندی عملکرد درختان را بهبود می‌بخشد؟ تقویت ­ing چطور؟

مدل‌های مبتنی بر قانون را در داده‌ها اعمال کنید. عملکرد چگونه است؟ آیا قوانین منطقی هستند؟

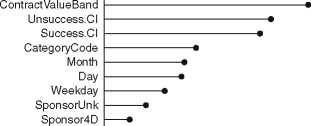
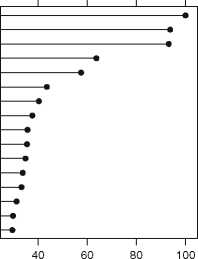
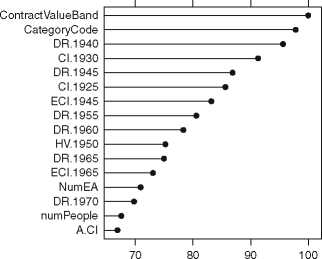
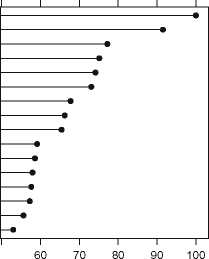
از نمودارهای بالابر برای مقایسه مدل‌های درختی یا قانون با بهترین تکنیک‌های فصل‌های قبلی استفاده کنید.

ورزش [12. 1](#bookmark605) شرح مفصلی از مجموعه داده‌های آسیب کبدی را ارائه می‌دهد که در آن هدف علمی اولیه برای این داده‌ها ایجاد مدلی برای پیش‌بینی وضعیت آسیب کبدی است. به یاد بیاورید که جنگل‌های تصادفی را می‌توان با درختان CART یا درختان استنتاج شرطی انجام داد. R را راه اندازی کنید و از این دستورات برای بارگیری داده‌ها استفاده کنید:

*> کتابخانه (کارت)*

*> داده (کبدی)*

یک مدل جنگل تصادفی را با استفاده از درختان CART و درختان استنتاج شرطی ­برای پیش‌بینی‌کننده‌های شیمی، با استفاده از آماره کاپا به‌عنوان معیار به شرح زیر برازش دهید:

ContractValueBandUnk Unsuccess. CI Success. CI SponsorUnk Day NumCI GrantCat30B Sat

**Bagging (Independent Categories)**

**Bagging (Grouped Categories)**

Sun A.CI Success.PS ContractValueBandG allPub

Unsuccess.HV Unsuccess.PS DR.1960 DR.PhD Unsuccess.CI DR.Dept3123 DurationGT15 PS.SouthAfrica ECI.Australia DR.Faculty19 DR.English

Importance

Importance

**Random Forests (Independent Categories)**

**Random Forests (Grouped Categories)**

Unsuccess.CI ContractValueBandUnk Success.CI SponsorUnk Day GrantCat30B ContractValueBandA GrantCat10A Jan Sat

Sponsor21A allPub

Sponsor4D Nov Aug

Sponsor24D

Importance

Sponsor2B allPub NumECI Sponsor24D NumSCI Sponsor21A numPeople

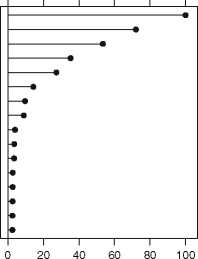
20 40 60 80 100

Importance

**GBM (Independent Categories)**

**GBM (Grouped Categories)**

Sponsor21A ContractValueBandA allPub NumSCI Sponsor2B Sponsor6B Sponsor24D GrantCat10A

شکل 14. 15: مقایسه‌ای از اهمیت متغیر برای روش‌های مجموعه‌ای از بسته بندی، جنگل‌های تصادفی و تقویت برای هر دو دسته‌های مستقل و پیش‌بینی دسته‌های گروه‌بندی شده

Importance

SponsorCode ContractValueBand Unsuccess.CI Success.CI NumCI Month Weekday Day SEO750604

RFCD260109 RFCD249999 RFCD300102 RFCD270601 RFCD291104 CI.Dept2893 CI.Dept248

0 20406080100

Importance

*> کتابخانه (کارت)*

*> set. seed(714)*

*> indx <-createFolds(آسیب، returnTrain = TRUE)*

*> ctrl <- trainControl(روش = "cv"، index = indx)*

*> mtryValues <- c(5، 10، 25، 50، 75، 100)*

*> rfCART <- آموزش (شیمی، آسیب،*

*+ روش = "rf"،*

*+ معیار = "کاپا"،*

*+ ntree = 1000،*

*+ tuneGrid = data. frame (. mtry = mtryValues))*

*> rfcForest <- آموزش(شیمی، آسیب،*

*+ روش = "جنگل"،*

*+ معیار = "کاپا"،*

*+ tuneGrid = data. frame (. mtry = mtryValues))*

کدام مدل عملکرد بهتری دارد و پارامترهای تنظیم مربوطه کدامند؟

برای بدست آوردن زمان محاسبه برای هر مدل از دستور زیر استفاده کنید:

*> rfCART$times$ everything*

*> rfcForest$times$ everything*

کدام مدل زمان محاسبات کمتری می‌برد؟ با توجه به مبادله بین عملکرد و زمان محاسبه، کدام مدل را ترجیح می‌دهید؟

از دستور زیر برای به دست آوردن اهمیت متغیر برای ده پیش‌بینی برتر برای هر مدل استفاده کنید:

*> varImp(rfCART)*

*> varImp(rfcForest)*

آیا تفاوت‌های قابل‌توجهی در اهمیت متغیر بین ده پیش‌بینی برتر برای هر مدل وجود دارد؟ دلایل احتمالی تفاوت‌ها را توضیح دهید.

فصل 15

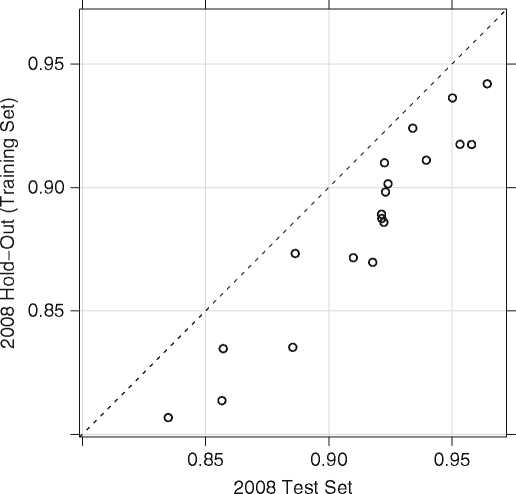
خلاصه‌ای از مدل‌های درخواست گرنت

سه فصل قبل از انواع فلسفه‌ها و تکنیک‌های مختلف برای پیش‌بینی موفقیت گرنت استفاده کردند. به‌طور کلی، ما بر این باوریم که ارزیابی تعدادی مدل مختلف برای هر مجموعه داده عاقلانه است، زیرا دشوار است که بدانیم کدام *یک به‌طور پیشینی* و با هر درجه‌ای از اطمینان به خوبی عمل می‌کنند.

قبل از مقایسه مدل‌ها، به یاد بیاورید که فرآیند تقسیم داده‌ها از گرنت قبل از سال 2008 برای تنظیم مدل استفاده می‌کرد و پس از تعیین پارامترهای نهایی، کل مجموعه آموزشی (که شامل کسری از گرنت 2008 بود) برای برازش با آن استفاده شد. مدل. مجموعه آزمایشی 518 که هنوز در تحلیل ما استفاده نشده است، به‌طور کامل از گرنت سال 2008 تشکیل شده است. همانطور که در بخش بحث شد.  [12. 1 ،](#bookmark550) عملکرد در ­مجموعه آزمایشی احتمالاً بهتر از نتایج تولید شده در فرآیند تنظیم است زیرا پارامترهای مدل نهایی بر اساس برخی از اطلاعات سال 2008 است.

شکل [15. 1](#bookmark39) تصویری از ناحیه زیر منحنی ROC را برای مدل‌های نهایی با دو مجموعه داده نشان می‌دهد. هر نقطه در طرح برای مدل خاصی از سه فصل آخر است. همبستگی بین دو تخمین بالا است (0. 96)، اگرچه مجموعه نگهدارنده تنظیم بدبینانه تر از مجموعه آزمایشی بود. به‌طور متوسط، مجموعه تست 0. 029 واحد بزرگتر بود. مجدداً، این تفاوت کوچک اما قابل تکرار احتمالاً به دلیل این واقعیت است که پیش‌بینی‌کننده‌های مجموعه آزمایشی از مدلی ساخته شده است که با اطلاعاتی در سال 2008 ساخته شده است و این تخمین‌ها احتمالاً برای پیش‌بینی کمک‌های جدیدتر مرتبط هستند. یک نکته مهم این است که، علیرغم تفاوت‌های بین عملکرد مجموعه تست و نگهداری، رتبه‌بندی عملکرد مدل بدون توجه به فرآیند تقسیم داده‌ها تقریباً یکسان است (جدول را ببینید.  [15. 1 )](#bookmark743) .

شکل [15. 2](#bookmark744) مساحت زیر منحنی‌های ROC را برای هر مدل نشان می‌دهد که با استفاده از گرنت‌های مجموعه آزمایشی تخمین زده می‌شود. میله‌ها نشان دهنده 95 درصد فواصل اطمینان هستند که با استفاده از راه انداز به دست آمده اند [( رابین و همکاران 2011](#bookmark1024) ). این برآوردها از عدم قطعیت به دو صورت کمک می‌کند. اولاً، فواصل زمانی به مصرف کنندگان مدل درکی در مورد اینکه مدل ممکن است چقدر خوب یا بد باشد، می‌دهد. این بازه کمیت تغییر در مدل را نشان می‌دهد، اما همچنین منعکس کننده داده‌ها است. به‌عنوان مثال، مجموعه‌های کوچکتر تست یا نویز (یا برچسب گذاری اشتباه) در ­پاسخ (به بخش مراجعه کنید.  [20. 2 )](#bookmark923) می تواند منجر به فواصل بیشتر شود. به این ترتیب اعتماد به نفس



شکل 15. 1: مقایسه‌ای از مناطق نگهداشته شده در سال 2008 در زیر منحنی ROC و مقادیر AUC که از ارزیابی مجموعه آزمایش حاصل می‌شود. هر نقطه نشان دهنده مدلی است که در سه فصل آخر برآورد شده است

فاصله زمانی به اندازه‌گیری وزن شواهد موجود هنگام مقایسه مدل‌ها کمک می‌کند. دومین مزیت فواصل اطمینان، تسهیل مبادلات بین مدل‌ها است. اگر فواصل اطمینان برای دو مدل به‌طور قابل‌توجهی بیش از ­یک دور باشد، این نشانه‌ای از هم ارزی (آماری) بین این دو است و ممکن است دلیلی برای ترجیح مدل کمتر پیچیده یا قابل تفسیرتر باشد.

در نهایت، قبل از مقایسه مدل‌ها، نظر ارائه شده در بخش را تکرار می‌کنیم.  [14. 7 :](#bookmark727) اثربخشی نسبی مدل‌ها برای این داده‌ها را نمی‌توان برای تعمیم به موقعیت‌های دیگر استفاده کرد. یک مدل با رتبه پایین برای این داده‌ها ممکن است در مجموعه داده‌های متفاوت از سایرین پیشی بگیرد. هر نتیجه‌ای که از این تحلیل‌ها به دست می‌آید باید «گزاره‌های آینده‌نگر» در نظر گرفته شود و باید با احتیاط همراه باشد.

در بین مدل‌ها، کدام پیش‌بینی بیشترین تأثیر را داشت؟[[54]](#footnote-54) در بسیاری از مدل‌ها، میزان موفقیت و شکست تاریخی برای محقق ارشد ­نقش مهمی در پیش‌بینی نتایج گرنت ایفا می‌کند («هیچ چیز مانند

جدول 15. 1: عملکرد مدل رتبه‌بندی شده در سال 2008 (مجموعه آموزشی) و مجموعه آزمایشی 2008

|  |  |  |
| --- | --- | --- |
| مدل | مجموعه آزمایشی Holdout | |
| C5. 0 | 1 | 1 |
| درخت تقویت شده | 2 | 4 |
| درخت کیسه دار | 4 | 3 |
| جنگل تصادفی | 5 | 2 |
| FDA | 3 | 6 |
| شبکه‌های عصبی | 6 | 5 |
| LDA پراکنده | 8 | 7 |
| glmnet | 7 | 9 |
| SVM (چند جمله ای) | 9 | 8 |
| سبد خرید | 12 | 10 |
| LDA | 10 | 12 |
| MDA | 11 | 11 |
| رگرسیون لجستیک | 14 | 14 |
| نزدیکترین مرکزهای کوچک شده | 13 | 15 |
| PLS | 15 | 13 |
| J48 | 16 | 16 |
| بخش | 17 | 17 |
| بیز ساده لوح | 18 | 18 |
| KNN | 19 | 19 |

روش‌ها دارای رتبه‌بندی منحنی ROC مشابهی بین دو مجموعه هستند (همبستگی رتبه سطح زیر منحنی: 0. 96)

موفقیت»). به نظر می‌رسید که فقدان اطلاعاتی در این داده‌ها رخ می‌دهد. مقادیر ناشناخته نوار ارزش قرارداد و کد حامی به شدت در بسیاری از مدل‌ها استفاده شد. این خود احتمالاً جایگزین یا سیگنالی برای برخی اطلاعات دیگر است. به‌طور مشابه، چرا یک اثر فصلی قوی وجود داشت به‌طوری که کمک‌های بلاعوض در اواخر دسامبر و اوایل ژانویه (به جدول مراجعه کنید) [12. 1](#bookmark552) و شکل.  [12. 3](#bookmark558) و [13. 8 )](#bookmark633) آیا احتمال موفقیت بیشتری داشتید؟ در این مورد، ممکن است دشوار باشد که بگوییم آیا افزایش موفقیت به دلیل محققینی است که گرنت را در این مدت ارائه می‌کنند یا اینکه بهبود به دلیل بازبینی کنندگان گرنت بوده است. در نهایت، گرنت با کد دسته 30B با کاهش نرخ موفقیت همراه بود. درک این روند به اضافه کردن قطعه دیگری به پازل برای کسانی که علاقه مند به بهبود میزان موفقیت هستند کمک می‌کند.

C5. 0

شبکه‌های عصبی درختی تقویت شده درختی   
تصادفی   
با کیسه جنگل

FDA

LDA SVM (چند جمله ای) glmnet پراکنده

سبد خرید   
MDA

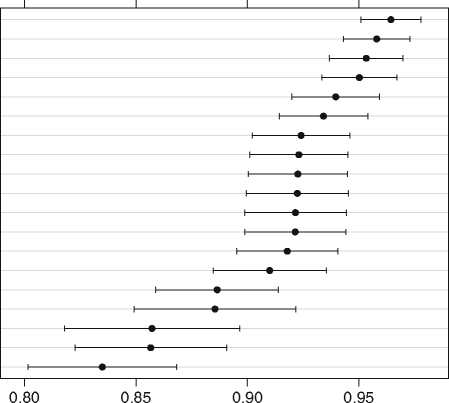
LDA PLS

رگرسیون لجستیک نزدیکترین مرکزهای کوچک شده

J48

بخش

ساده لوح بیز KNN

شکل 15. 2: نمودار AUC منحنی ROC مجموعه تست و فواصل اطمینان 95 درصد مربوط به آنها

Test Set ROC Curve AUC

فصل 16

راهکارهایی برای عدم تعادل شدید طبقاتی

هنگام مدل‌سازی کلاس‌های گسسته، فرکانس‌های نسبی کلاس‌ها می‌تواند تأثیر قابل‌توجهی بر اثربخشی مدل داشته باشد. عدم تعادل زمانی رخ می‌دهد که یک یا چند کلاس نسبت به سایر کلاس‌ها نسبت بسیار پایینی در داده‌های آموزشی داشته باشند. عدم تعادل می‌تواند در هر مجموعه داده یا برنامه‌ای وجود داشته باشد و از این رو، متخصص باید از پیامدهای مدل‌سازی این نوع داده آگاه باشد.

در اینجا چند تنظیمات عملی وجود دارد که در آن عدم تعادل کلاس اغلب رخ می‌دهد:

تبلیغات آنلاین: تبلیغی به بیننده ارائه می‌شود که باعث ایجاد حس می‌*شود.* نرخ *کلیک* تعداد دفعاتی است که بر روی یک تبلیغ کلیک شده است تقسیم بر تعداد کل نمایش‌ها و بسیار پایین است [( ریچاردسون و همکاران 2007).](#bookmark1024)  نرخی را کمتر از 2. 4 درصد ذکر کنید.

تحقیقات دارویی: غربالگری با توان عملیاتی بالا یک تکنیک تجربی است که در آن تعداد زیادی مولکول (10000s) به سرعت برای فعالیت بیولوژیکی ارزیابی می‌شوند. معمولاً فقط چند مولکول فعالیت بالایی نشان می‌دهند. بنابراین، فراوانی ترکیبات جالب کم است.

مطالبات بیمه: آرتیس و همکاران. ( 2002 ) ادعاهای خسارت بیمه خودرو را در اسپانیا بین سالهای 1993 و 1996 بررسی کرد. از دعاوی تحت حسابرسی، میزان تقلب تقریباً 22 درصد برآورد شد.

این فصل تاثیر عدم تعادل طبقاتی را بر معیارهای عملکرد، روش‌شناسی برای پیش‌بینی‌کننده‌های مدل پس از پردازش و مدل‌های پیش‌بینی ­که می‌توانند این موضوع را در طول آموزش کاهش دهند، مورد بحث قرار می‌دهد. با این حال، قبل از ادامه، مطالعه موردی دیگری را شرح خواهیم داد که از آن برای نشان دادن رویکردهای مختلف برای رسیدگی به عدم تعادل طبقاتی استفاده خواهیم کرد.

16. 1 مطالعه موردی: پیش‌بینی مالکیت سیاست کاروان

یک مجموعه داده تولید شده توسط شبکه تحقیقاتی هوش محاسباتی و یادگیری (CoIL) برای نشان دادن روش‌هایی برای مبارزه با عدم تعادل طبقاتی استفاده می‌شود. چالش CoIL در سال 2000 پیش‌بینی این بود که آیا مشتریان بیمه کاروان خریداری خواهند کرد یا خیر [[55]](#footnote-55).  [ون در پوتن و ون سامرن](#bookmark1026) [( 2004](#bookmark1026) ) این داده‌ها، چالش و برخی از راه حل‌های مسأله را مورد بحث قرار می‌دهند. بحث ما در مورد این داده‌ها به نشان دادن اثر عدم تعادل طبقاتی محدود می‌شود.

نتیجه، چه مشتری بیمه کاروان خریداری کرده باشد، بسیار نامتعادل است و تنها 6 درصد از مشتریان بیمه نامه خریداری کرده اند. پیش‌بینی‌کننده‌ها در مجموعه داده شامل موارد زیر بودند:

تعیین زیرمجموعه مشتری، مانند «خانواده‌های سنتی» یا « ­خانواده‌های جوان مرفه». 39 مقدار منحصر به فرد وجود داشت، اگرچه بسیاری از زیرشاخه‌ها کمتر از 5٪ از مشتریان را تشکیل می‌دهند.

عوامل جمعیتی، مانند مذهب، سطح تحصیلات، طبقه اجتماعی، درآمد و 38 مورد دیگر. مقادیر پیش‌بینی‌کننده‌ها از داده‌های سطح کد پستی مشتق شده‌اند، بنابراین مشتریان ساکن در کد پستی یکسان مقادیر یکسانی برای این ویژگی‌ها خواهند داشت. [[56]](#footnote-56)

اطلاعات مالکیت محصول، مانند تعداد (یا مشارکت در) خط‌مشی‌های انواع مختلف.

در مجموع 85 پیش‌بینی وجود داشت. بسیاری از پیش‌بینی‌کننده‌های طبقه‌بندی دارای 10 سطح یا بیشتر بودند و پیش‌بینی‌کننده‌های مبتنی بر شمارش نسبتاً پراکنده بودند (یعنی مقدار کمی غیر صفر).

برای نشان دادن روش‌های مختلف با این داده‌ها، از نمونه‌گیری تصادفی طبقه‌ای (که در آن لایه متغیر پاسخ بود) برای ایجاد سه مجموعه داده مختلف استفاده شد:

مجموعه آموزشی از مشتریان ( *n* = 6877) که برای تخمین پارامترهای مدل، مدل‌های تنظیم و غیره استفاده می‌شود.

*ارزیابی* کوچکی از مشتریان ( *n* = 983) که برای توسعه تکنیک‌های پس از پردازش، مانند برش‌های احتمال جایگزین استفاده می‌شود.

یک مجموعه تست مشتری ( *n* = 1962) که صرفاً برای ارزیابی نهایی مدل‌ها استفاده می‌شود

نرخ مشتریان با سیاست‌های کاروان در هر یک از این مجموعه داده‌ها تقریباً در هر سه مجموعه داده یکسان بود (به ترتیب 6٪، 5. 9٪ و 6٪).

اثر عدم تعادل طبقاتی

برای شروع، از سه مدل پیش‌بینی برای مدل‌سازی داده‌ها استفاده شد: جنگل تصادفی، یک مدل تحلیل تفکیک انعطاف‌پذیر (با توابع لولای MARS) و رگرسیون لجستیک. برای تنظیم مدل‌ها، از اعتبارسنجی متقاطع 10 برابری استفاده شد. هر نمونه نگهدارنده تقریباً شامل 687 مشتری بود که باید ارائه دهند

جدول 16. 1: نتایج برای سه مدل پیش‌بینی با استفاده از مجموعه ارزیابی

مدل کاپا حساسیت ویژه ROC AUC

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| جنگل تصادفی | 93. 5 | 0. 091 | 6. 78 | 99. 0 | 0. 757 |
| FDA (MARS) | 93. 8 | 0. 024 | 1. 69 | 99. 7 | 0. 754 |
| رگرسیون لجستیک | 93. 9 | 0. 027 | 1. 69 | 99. 8 | 0. 727 |

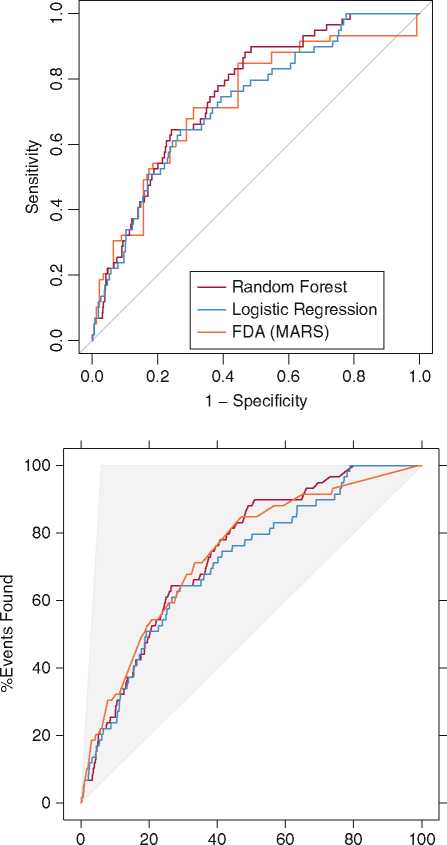
برآوردهای منطقی از عدم قطعیت برای انتخاب مدل بهینه، ناحیه زیر منحنی مشخصه عملکرد گیرنده (ROC) بهینه شد. [[57]](#footnote-57)

مدل جنگل تصادفی از 1500 درخت در جنگل استفاده کرد و بر روی 5 مقدار پارامتر *m try تنظیم شد (*بخش.  [8. 5 )](#bookmark419) ؛ مدل نهایی دارای مقدار بهینه *m try* برابر با 126 بود. مدل FDA از ویژگی‌های درجه اول استفاده می‌کرد و بیش از 25 مقدار برای تعداد عبارت‌های حفظ شده تنظیم شد. فرآیند ­نمونه‌گیری مجدد مشخص کرد که 13 عبارت مدل مناسب هستند. رگرسیون لجستیک از یک مدل افزودنی ساده (یعنی بدون برهمکنش یا عبارت غیرخطی) با مجموعه پیش‌بینی کاهش‌یافته استفاده کرد (بسیاری از پیش‌بینی‌کننده‌های واریانس نزدیک به صفر حذف شدند تا مدل به یک راه‌حل پایدار منجر شود).

تعدادی از معیارهای عملکرد مختلف از جمله: دقت کلی، کاپا، سطح زیر منحنی ROC، حساسیت و ­ویژگی (که در آن خط مشی خریداری شده به‌عنوان "رویداد" مورد علاقه تعریف می‌شود) برآورد شد. همه مدل‌ها نمونه‌ها را در مجموعه داده‌های ارزیابی پیش‌بینی کردند و نتایج بسیار مشابهی را به دست آوردند، همانطور که در جدول نشان داده شده است.  [16. 1](#bookmark41) . در هر مدل، هر الگویی که برای پیش‌بینی نتیجه مفید بود، توسط درصد زیادی از مشتریان بدون بیمه کاروانی تحت تأثیر قرار گرفت. در واقع، با وجود 59 مشتری دارای بیمه در مجموعه ارزیابی، هیچ یک از مدل‌ها بیش از 13 مشتری در مجموعه ارزیابی را دارای بیمه پیش‌بینی نکردند. مفهوم این نتایج این ­است که مدل‌ها به ویژگی خوبی دست می‌یابند (زیرا تقریباً برای هر مشتری هیچ بیمه پیش‌بینی نمی‌شود) اما حساسیت ضعیفی دارند.

عدم تعادل همچنین تأثیر شدیدی بر احتمالات طبقاتی پیش‌بینی‌شده داشت. به‌عنوان مثال، در مدل جنگل تصادفی، 82 درصد از مشتریان ­احتمال پیش‌بینی بیمه 10 درصد یا کمتر را دارند. این توزیع احتمال پیش‌بینی‌شده بسیار چپ چپ برای دو مدل دیگر نیز رخ می‌دهد.

شکل [16. 1](#bookmark752) نمودارهای بالابر و منحنی‌های ROC را برای مجموعه ارزیابی نشان می‌دهد. منحنی‌های بالابر بسیار شبیه به هم هستند و می‌توانند برای تعیین تعداد افراد مورد نیاز برای جذب درصد خاصی از افرادی که ممکن است بیمه نامه را خریداری کنند، استفاده شود. به‌عنوان مثال، برای یافتن حدود 60 درصد از افراد دارای سیاست با استفاده از مدل جنگل تصادفی، حدود 30 درصد از جمعیت باید نمونه برداری شوند. منحنی‌های ROC همپوشانی قابل‌توجهی را نشان می‌دهند و مدل‌ها را متمایز نمی‌کنند.



٪ مشتریان ارزیابی شده است

شکل 16. 1: *بالا* : منحنی‌های ROC مجموعه ارزیابی برای هر یک از سه خط پایه

مدل ها. *پایین* : نمودارهای بالابر مربوطه

در این شکل، شباهت زیادی بین منحنی‌های lift و ROC وجود دارد. وقتی کلاس‌ها متعادل‌تر هستند، نمودارهای لیفت و منحنی‌های ROC این‌قدر شبیه نیستند. بلکه شباهتی که در شکل 1 می‌بینیم.  [16. 1](#bookmark752) به دلیل عدم تعادل کلاس است. به‌طور کلی، این مورد نیست، اما مصنوع از عدم تعادل شدید طبقاتی است. وقتی کلاس‌ها متعادل تر باشند، بعید است که منحنی‌ها مشابه باشند

الگوها به‌عنوان مثال، حداکثر مساحت در منحنی لیفت با نرخ رویداد در داده‌ها محدود می‌شود (همانطور که با ناحیه پس زمینه خاکستری در نمودار لفت نشان داده شده است) در حالی که منحنی ROC چنین محدودیتی ندارد.

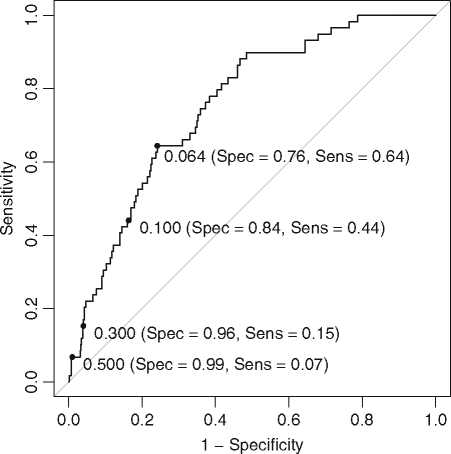
بخش‌های باقی‌مانده درباره راهبردهای دیگر برای غلبه بر عدم تعادل طبقاتی بحث می‌کنند. در بخش بعدی بحث می‌شود که چگونه می‌توان از تنظیم مدل برای ­افزایش حساسیت کلاس اقلیت در Sect استفاده کرد.  [16. 4](#bookmark757) نشان می‌دهد که چگونه می‌توان برش‌های احتمال جایگزین را از داده‌ها برای بهبود نرخ خطا در کلاس اقلیت استخراج کرد. این به معنای پس پردازش پیش‌بینی‌کننده‌های مدل برای تعریف مجدد پیش‌بینی‌کننده‌های کلاس است. تغییرات وزن مورد و احتمالات قبلی نیز مورد بحث قرار گرفته است. بخش دیگری بررسی می‌کند که چگونه داده‌های آموزشی را می‌توان برای کاهش عدم تعادل کلاس قبل از آموزش مدل تغییر داد. آخرین رویکرد شرح داده شده در بخش.  [16. 6](#bookmark764) نشان می‌دهد که چگونه، برای برخی از مدل‌ها، فرآیند آموزش مدل را می‌توان تغییر داد تا بر ­دقت مدل برای کلاس(های) کمتر متداول تأکید شود. در این مورد، پارامترهای مدل برآورد شده به جای پردازش پس از خروجی مدل، اصلاح می‌شوند.

تیونینگ مدل

ساده‌ترین رویکرد برای مقابله با اثرات منفی عدم تعادل طبقاتی، تنظیم مدل برای به حداکثر رساندن دقت طبقه(های) اقلیت است. برای پیش‌بینی بیمه، تنظیم مدل برای به حداکثر رساندن حساسیت ممکن است به حساسیت زدایی فرآیند آموزش نسبت به درصد بالایی از داده‌ها بدون سیاست‌های کاروان در مجموعه آموزشی کمک کند. مدل جنگل تصادفی که برای این داده‌ها تنظیم شد، روند معنی‌داری در حساسیت در سراسر پارامتر تنظیم نشان نداد. مدل FDA روندی را نشان داد. با افزایش تعداد اصطلاحات مدل، حساسیت از 0% برای مدل‌های بسیار ساده به 5. 4% با حفظ 16 عبارت افزایش یافت. این بهبود جزئی در حساسیت عملاً هیچ هزینه‌ای برای ویژگی ندارد. با توجه به اینکه افزایش حساسیت به اندازه کافی بالا نیست که قابل قبول تلقی شود، این رویکرد برای حل مسأله برای این مجموعه داده خاص موثر نیست. استفاده از تنظیم مدل برای این منظور در بخش بازبینی شده است.  [16. 8](#bookmark771) .

قطع‌های جایگزین

هنگامی که دو دسته نتیجه ممکن وجود دارد، روش دیگری برای افزایش ­دقت پیش‌بینی نمونه‌های کلاس اقلیت، تعیین برش‌های جایگزین برای احتمالات پیش‌بینی‌شده است که به‌طور موثر تعریف یک رویداد پیش‌بینی‌شده را تغییر می‌دهد. ساده‌ترین رویکرد استفاده از منحنی ROC است زیرا حساسیت و ویژگی را در یک زنجیره از برش‌ها محاسبه می‌کند. با استفاده از این منحنی می‌توان تعادل مناسبی بین حساسیت و ویژگی تعیین کرد.



شکل 16. 2: منحنی ROC جنگل تصادفی برای پیش‌بینی کلاس‌ها با استفاده از مجموعه ارزیابی. عدد سمت چپ نشان دهنده قطع احتمال است و اعداد داخل پرانتز به ترتیب ویژگی و حساسیت هستند. چندین برش احتمال ممکن استفاده می‌شود، از جمله آستانه از نظر هندسی نزدیک به مدل کامل (0. 064)

شکل [16. 2](#bookmark758) منحنی ROC را برای مدل جنگل تصادفی بر اساس مجموعه ارزیابی نشان می‌دهد. چندین برش بر روی منحنی نشان داده شده است و آشکار است که کاهش برش برای احتمال پاسخگویی، ­حساسیت را افزایش می‌دهد (به قیمت ویژگی). ممکن است شرایطی وجود داشته باشد که مبادله حساسیت/ویژگی را بتوان بدون به خطر انداختن جدی ­دقت طبقه اکثریت (که البته به زمینه مسأله بستگی دارد) انجام داد.

چندین تکنیک برای تعیین یک برش جدید وجود دارد. ابتدا، اگر هدف خاصی وجود دارد که باید برای حساسیت یا ویژگی برآورده شود، این نقطه را می‌توان در منحنی ROC پیدا کرد و برش مربوطه را می‌توان تعیین کرد. روش دیگر یافتن نقطه‌ای در منحنی ROC است که نزدیک‌ترین (یعنی کوتاه‌ترین فاصله) به مدل کامل (با حساسیت 100٪ و ویژگی 100٪) است که با گوشه سمت چپ بالای نمودار مرتبط است. در شکل [16. 2](#bookmark758) ، مقدار برش 0. 064 نزدیکترین به مدل کامل خواهد بود. روش دیگری برای تعیین برش از شاخص *J Youden استفاده می‌کند (*به بخش مراجعه کنید.  [11. 2 )](#bookmark512)  که نسبت نمونه‌های پیش‌بینی‌شده درست را برای هر دو گروه رویداد و غیر رویداد اندازه‌گیری می‌کند. این شاخص را می‌توان برای هر قطعی که برای ایجاد منحنی ROC استفاده می‌شود محاسبه کرد. برش مرتبط با بزرگترین مقدار شاخص یودن نیز ممکن است عملکرد برتر نسبی را نشان دهد

جدول 16. 2: ماتریس‌های سردرگمی برای مجموعه آزمایشی برای جنگل‌های تصادفی با استفاده از برش‌های پیش فرض و جایگزین

|  |  |  |  |
| --- | --- | --- | --- |
|  | 0. 50 قطع | | 0. 064 قطع |
| بیمه بدون بیمه | | بیمه بدون بیمه |
| بیمه | 11 | 19 | 71 441 |
| بیمه غیر بیمه ای | 105 | 1827 | 45 1,405 |

به مقدار پیش فرض 50٪. برای منحنی ROC جنگل تصادفی، برشی که شاخص یودن (0. 021) را به حداکثر می‌رساند، مشابه نزدیک‌ترین نقطه به مدل بهینه است.

با استفاده از مجموعه ارزیابی، حساسیت پیش‌بینی‌شده برای برش جدید 0. 064 64. 4 درصد است که نسبت به مقدار تولید شده توسط قطع پیش‌فرض بهبود قابل‌توجهی است. نتیجه قطع جدید این است که برآورد می‌شود ویژگی از 99٪ به 75. 9٪ کاهش یابد. این ممکن است بر اساس زمینه نحوه استفاده از مدل قابل قبول باشد یا نباشد.

در تحلیل ما، برش متناوب برای مدل از مجموعه‌های آموزشی یا آزمایشی مشتق نشده است. استفاده از یک مجموعه داده مستقل برای استخراج برش، به خصوص برای نمونه‌های کوچک مهم است. اگر از پیش‌بینی‌کننده‌های مجموعه آموزشی استفاده شود، احتمالاً یک بایاس خوش‌بینانه بزرگ در احتمالات کلاس وجود دارد که منجر به ارزیابی‌های نادرست از حساسیت و ویژگی می‌شود. اگر مجموعه تست استفاده شود، دیگر منبع بی طرفی برای قضاوت در مورد عملکرد مدل نیست. مثلا، [ایوالد](#bookmark1015) [( 2006](#bookmark1015) ) از طریق شبیه‌سازی دریافتند که *استخراج post hoc* برش‌ها می‌تواند عملکرد مجموعه تست را اغراق کند.

جدول [16. 2](#bookmark759) حاوی ماتریس‌های سردرگمی است که از مجموعه آزمایشی برای برش‌های پیش فرض و متناوب به دست آمده است. پیش‌بینی‌کننده‌ها از بریدگی 50 درصد، همان مقادیر عملکرد نشان‌داده‌شده توسط اعتبارسنجی متقابل (حساسیت 9. 5 درصد و ویژگی 99 درصد) را منعکس می‌کنند. برای برش جدید، حساسیت مجموعه تست 61. 2٪ بود در حالی که ویژگی 76. 1٪ محاسبه شد. این تفاوت تخمینی در حساسیت بین دو مجموعه داده (64. 4٪ و 61. 2٪) احتمالاً به دلیل عدم قطعیت بالا در معیار است (که خود نتیجه کم بودن فراوانی رویدادها است). این دو مقدار ممکن است با توجه به نویز آزمایشی معادل باشند. برای تکرار، این برش ناشی از نزدیک‌ترین حالت ممکن به مدل بهینه است. این ممکن است در زمینه مسائل دیگر قابل اجرا نباشد که در آن موارد می‌توان مبادلات جایگزین انجام داد.

شایان ذکر است که هسته اصلی مدل تغییر نکرده است. از همان پارامترهای مدل استفاده می‌شود. تغییر برش برای افزایش ­حساسیت، اثربخشی پیش‌بینی کلی مدل را افزایش نمی‌دهد. تأثیر اصلی که یک برش جایگزین دارد، ایجاد معاوضه بین انواع خاصی از خطاها است. به‌عنوان مثال، در یک ماتریس سردرگمی، برش‌های متناوب ­فقط می‌توانند نمونه‌ها را در ردیف‌های ماتریس به سمت بالا و پایین حرکت دهند (بر خلاف انتقال آنها از قطرهای خارج به مورب). به عبارت دیگر، استفاده از یک برش جایگزین باعث جدایی بیشتر بین طبقات نمی‌شود.

شکل [16. 2](#bookmark758) و جدول [16. 2](#bookmark759) نشان می‌دهد که چرا، برای بسیاری از مسائل طبقه بندی، مقایسه مدل‌ها بر اساس حساسیت و ویژگی پیش فرض ­ممکن است گمراه کننده باشد. از آنجایی که ممکن است یک برش بهتر امکان‌پذیر باشد، تحلیل منحنی ROC می‌تواند منجر به بهبود این معیارها شود. همانطور که در شکل نشان داده شده است.  [16. 2](#bookmark758) ، عدم تعادل طبقاتی می‌تواند این موضوع را بیشتر تشدید کند. در نتیجه، معیارهای عملکرد مستقل از بریدگی‌های احتمال (مانند ناحیه زیر منحنی ROC) احتمالاً تضادهای معنی‌داری بین مدل‌ها ایجاد می‌کنند. با این حال، برخی از مدل‌های پیش‌بینی فقط پیش‌بینی‌کننده‌های کلاس گسسته را تولید می‌کنند.

تنظیم احتمالات قبلی

برخی از مدل‌ها از احتمالات قبلی مانند Bayes ساده و طبقه‌بندی‌کننده تحلیل متمایز استفاده می‌کنند. مگر اینکه به صورت دستی مشخص شود، این مدل‌ها معمولاً مقدار پیشین‌ها را از داده‌های آموزشی استخراج می‌کنند.  [ویس و پروست ( 2001a](#bookmark1027) ) پیشنهاد می‌کنند که پیش‌بینی‌کننده‌هایی که عدم تعادل طبقاتی طبیعی را منعکس می‌کنند، پیش‌بینی‌کننده‌های مادی را به طبقه اکثریت بایاس می‌کنند. استفاده از اولویت‌های متعادل تر یا یک مجموعه آموزشی متعادل ممکن است به مقابله با عدم تعادل کلاس کمک کند.

برای داده‌های بیمه، اولویت‌ها به ترتیب 6 درصد و برای بیمه شدگان 94 درصد است. احتمال پیش‌بینی‌شده داشتن بیمه ­برای هر سه مدل به شدت چپ است و تنظیم پیشین‌ها می‌تواند توزیع احتمال را از مقادیر کوچک دور کند. به‌عنوان مثال، اولویت‌های جدید 60 درصدی برای بیمه شده و 40 درصدی برای افراد غیر بیمه شده در مدل FDA، احتمال داشتن بیمه را به میزان قابل‌توجهی افزایش می‌دهد. با قطع پیش‌فرض، ­پیش‌بینی‌کننده‌های مدل جدید دارای حساسیت 71. 2 درصد و ویژگی 66. 9 درصد در مجموعه آزمایشی هستند. با این حال، احتمالات کلاس جدید رتبه‌بندی مشتریان در مجموعه آزمایشی را تغییر نداد و مدل دارای همان منطقه زیر منحنی ROC مانند مدل قبلی FDA است. مانند تاکتیک‌های قبلی برای یک برش جایگزین، این استراتژی مدل را تغییر نداد، اما اجازه می‌دهد تا مبادلات متفاوتی بین حساسیت و ویژگی وجود داشته باشد.

وزن‌های پرونده نابرابر

بسیاری از مدل‌های پیش‌بینی برای طبقه‌بندی، این توانایی را دارند که از *وزن‌های موردی استفاده کنند* که در آن می‌توان به هر نقطه داده جداگانه در مرحله آموزش مدل تأکید بیشتری کرد. برای مثال، رویکردهای تقویتی که قبلاً در مورد درختان طبقه‌بندی و رگرسیون مورد بحث قرار گرفت، می‌توانند دنباله‌ای از مدل‌ها را ایجاد کنند که هر کدام وزن‌های مورد متفاوتی را در هر تکرار اعمال می‌کنند.

یک رویکرد برای ایجاد تعادل مجدد مجموعه آموزشی، افزایش وزن برای نمونه‌ها در کلاس‌های اقلیت است [( Ting 2002](#bookmark1026) ). برای بسیاری از مدل‌ها، این می‌تواند به‌عنوان داشتن نقاط داده تکراری یکسان با مقادیر پیش‌بینی دقیقاً یکسان تفسیر شود. برای مثال، رگرسیون لجستیک می‌تواند از وزن‌های موردی در این راه استفاده کند. این روش برای برخورد با عدم تعادل کلاس مربوط به روش‌های نمونه‌گیری است که در بخش بعدی بحث می‌شود.

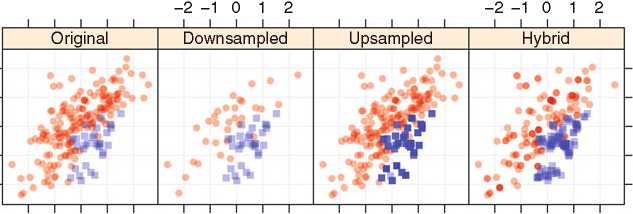
روش‌های نمونه گیری

هنگامی که دانش *پیشینی* از عدم تعادل کلاس وجود دارد، یک روش ساده برای کاهش تأثیر آن بر آموزش مدل، انتخاب یک نمونه مجموعه آموزشی است که نرخ رویداد تقریباً برابری در طول جمع‌آوری داده‌های اولیه داشته باشد (به‌عنوان مثال، Artis و همکاران 2002 را ببینید ). . اساساً به جای اینکه مدل با عدم تعادل سروکار داشته باشد، می‌توانیم سعی کنیم فرکانس‌های کلاس را متعادل کنیم. اتخاذ این رویکرد مسأله عدم تعادل اساسی را که گریبانگیر آموزش مدل می‌شود را از بین می‌برد. با این حال، اگر مجموعه آموزشی برای متعادل بودن نمونه برداری شود، مجموعه آزمون باید نمونه برداری شود تا با وضعیت طبیعی سازگارتر باشد و باید عدم تعادل را منعکس کند تا بتوان برآوردهای صادقانه از عملکرد آینده را محاسبه کرد.

اگر روش نمونه‌گیری *پیشینی امکان‌پذیر نباشد، روش‌های نمونه‌گیری تعقیبی وجود دارد* که می‌تواند به کاهش اثرات عدم تعادل در طول ­آموزش مدل کمک کند. دو رویکرد *تعقیبی کلی عبارتند از نمونه‌گیری پایین* و *نمونه برداری* از داده ها. نمونه‌برداری بالا به هر تکنیکی گفته می‌شود که نقاط داده اضافی را برای بهبود تعادل در بین کلاس‌ها شبیه‌سازی یا القا می‌کند، در حالی که نمونه‌برداری پایین به هر تکنیکی اطلاق می‌شود که تعداد نمونه‌ها را برای بهبود تعادل بین کلاس‌ها کاهش دهد.

لینگ و لی [( 1998](#bookmark1021) ) یک رویکرد برای نمونه‌گیری بالا ارائه می‌کنند که در آن موارد از طبقات اقلیت با جایگزینی نمونه‌برداری می‌شوند تا زمانی که هر طبقه تقریباً به همان تعداد برسد. برای داده‌های بیمه، مجموعه آموزشی شامل 6466 مشتری غیر بیمه نامه و 411 مشتری بیمه شده بود. اگر داده‌های کلاس اقلیت اصلی را حفظ کنیم، با افزودن 6055 نمونه تصادفی (با جایگزینی) کلاس اقلیت برابر با اکثریت است. در انجام این کار، برخی از نمونه‌های کلاس اقلیت ممکن است در مجموعه آموزشی با فرکانس نسبتاً بالایی نشان داده شوند، در حالی که هر نمونه در کلاس اکثریت یک تحقق واحد در داده‌ها دارد. این بسیار شبیه به رویکرد وزن مورد نشان داده شده در بخش قبلی است، با وزن‌های مختلف در هر مورد.

نمونه‌گیری پایین، نقاط داده را از کلاس ­اکثریت انتخاب می‌کند تا کلاس اکثریت تقریباً به اندازه کلاس(های) اقلیت باشد. چندین رویکرد برای نمونه‌گیری پایین وجود دارد. اول، یک رویکرد اساسی این است که به‌طور تصادفی از کلاس‌های اکثریت نمونه برداری کنید تا همه کلاس‌ها تقریباً یک اندازه داشته باشند. یک ­رویکرد دیگر، گرفتن یک نمونه بوت استرپ در همه موارد است به‌طوری که کلاس‌ها در مجموعه بوت استرپ متعادل باشند. مزیت این روش این است که انتخاب بوت استرپ را می‌توان بارها اجرا کرد تا تخمین زده شود



0J

Classi Class2

2

1 -

0-

1 -

—

■2-

—J

-2 -1 0 1 2

-2 -1 0 1 2

Predictor A

Fig. 16.3: *From left to right*: The original simulated data set and realizations of a down-sampled version, an up-sampled version, and sampling using SMOTE where the cases are sampled and/or imputed

از تغییرات را می‌توان در مورد نمونه‌گیری پایین به دست آورد. یکی از پیاده‌سازی جنگل‌های تصادفی می‌تواند ذاتاً با کنترل فرآیند نمونه‌گیری بوت استرپ در یک *متغیر طبقه‌بندی، نمونه‌برداری کند.* اگر کلاس به‌عنوان ­متغیر طبقه‌بندی استفاده شود، نمونه‌های بوت استرپ ایجاد می‌شوند که تقریباً اندازه هر کلاس یکسان هستند. این نسخه‌های *نمونه‌برداری شده داخلی* از مجموعه آموزشی سپس برای ساخت درخت‌ها در مجموعه استفاده می‌شوند.

روش نمونه برداری بیش از حد اقلیت مصنوعی (SMOTE) که توسط [Chawla و همکارانش توضیح داده شده است. ( 2002](#bookmark1013) )، یک روش نمونه‌گیری داده‌ای است که بسته به کلاس از نمونه‌گیری بالا و پایین استفاده می‌کند و دارای سه پارامتر عملیاتی است ­: مقدار نمونه‌گیری بالا، مقدار نمونه‌برداری پایین و تعداد همسایه‌هایی که برای نسبت دادن موارد جدید استفاده می‌شوند. برای نمونه برداری برای کلاس اقلیت، SMOTE موارد جدید را ترکیب می‌کند. برای انجام این کار، یک نقطه داده به صورت ­تصادفی از کلاس اقلیت انتخاب می‌شود و نزدیکترین همسایه K آن ( *K NNs* ) تعیین می‌شود.  [چاولا و همکاران ( 2002](#bookmark1013) ) در تحلیل‌های خود از پنج همسایه استفاده کردند، اما بسته به داده‌ها می‌توان از مقادیر متفاوتی استفاده کرد. نقطه داده مصنوعی جدید ترکیبی تصادفی از پیش‌بینی‌کننده‌های ­نقطه داده انتخاب‌شده به‌طور تصادفی و همسایگان آن است. در حالی که الگوریتم SMOTE نمونه‌های جدیدی را از طریق نمونه‌گیری بالا به کلاس اقلیت اضافه می‌کند، همچنین می‌تواند موارد را از کلاس اکثریت از طریق نمونه‌گیری تصادفی پایین بیاورد تا به تعادل مجموعه آموزشی کمک کند.

شکل [16. 3](#bookmark768) نمونه‌ای از روش‌های نمونه‌گیری با استفاده از مجموعه داده‌های شبیه‌سازی شده را نشان می‌دهد. داده‌های اصلی شامل 168 نمونه از کلاس اول و 32 نمونه از کلاس دوم (نسبت 5. 2:1) است. نسخه پایین نمونه داده‌ها، حجم کل نمونه را به 64 مورد کاهش داد که به‌طور مساوی بین کلاس‌ها تقسیم شد. داده‌های نمونه‌گیری شده دارای 336 مورد است که اکنون 168 رویداد وجود دارد. ­نسخه SMOTE داده‌ها دارای عدم تعادل کمتری است (با نسبت 1. 3:1) که ناشی از داشتن 128 نمونه از کلاس اول و 96 نمونه از کلاس دوم است. در بیشتر موارد، نمونه‌های مصنوعی جدید بسیار شبیه به سایر موارد در مجموعه داده‌ها هستند. فضایی که طبقه اقلیت را در بر می‌گیرد به جای گسترش، پر می‌شود.

لازم به ذکر است که هنگام استفاده از نسخه‌های اصلاح شده مجموعه آموزشی، تخمین‌های نمونه‌گیری مجدد از عملکرد مدل می‌تواند بایاس شود. برای مثال، اگر داده‌ها به‌صورت نمونه‌گیری بالا انجام شوند، روش‌های نمونه‌گیری مجدد احتمالاً در مواردی که برای ساخت مدل و همچنین مجموعه نگهدارنده استفاده می‌شوند، نمونه مشابهی دارند که منجر به نتایج خوش‌بینانه می‌شود. با وجود این، روش‌های نمونه‌گیری مجدد همچنان می‌توانند در تنظیم مدل‌ها مؤثر باشند.

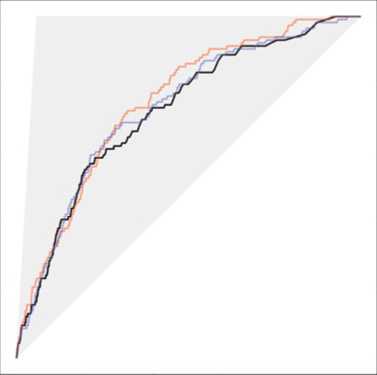
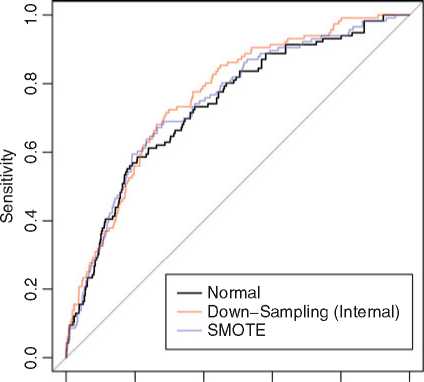
تحقیقات قابل‌توجهی در مورد اثربخشی ­استفاده از روش‌های نمونه‌گیری برای مبارزه با توزیع‌های طبقاتی ناهنجار انجام شده است، به ویژه [ویس و پرووست ( 2001b](#bookmark1027) )، باتیستا و همکاران. ( 2004 ) [ون هالس و همکاران](#bookmark1026) [( 2007](#bookmark1026) ) [بورز و ون دن پول](#bookmark1012) [( 2009](#bookmark1012) )، و [جئتراکول و همکاران ( 2010](#bookmark1019) ). این نشریات و سایر نشریات نشان می‌دهند که در بسیاری از موارد، نمونه‌گیری می‌تواند مسائل ناشی از عدم تعادل را کاهش دهد، اما هیچ برنده مشخصی در بین ­رویکردهای مختلف وجود ندارد. همچنین، بسیاری از تکنیک‌های مدل‌سازی واکنش‌های متفاوتی نسبت به ­نمونه‌برداری نشان می‌دهند که ایده یک دستورالعمل ساده برای استفاده از آن را پیچیده‌تر می‌کند.

این روش‌های نمونه‌گیری برای مدل‌های جنگل تصادفی برای داده‌های بیمه با استفاده از فرآیند تنظیم مشابه مدل اصلی اعمال شد. منحنی‌های ROC و لیفت برای مجموعه ارزیابی برای سه مدل جنگل تصادفی در شکل نشان داده شده است.  [16. 4](#bookmark772) . خلاصه‌های عددی مدل‌ها در جدول موجود است [16. 3 .](#bookmark773) در این جدول مساحت زیر منحنی ROC برای مجموعه‌های ارزیابی و تست محاسبه شده است. با استفاده از مجموعه ارزیابی، برش‌های جدید برای هر مدل با انتخاب نقطه روی منحنی ROC نزدیک‌ترین به مدل بهینه به دست آمد. مقادیر حساسیت و ویژگی در این جدول نتیجه اعمال آن برش‌ها در مجموعه آزمایشی است.

نتایج نشان می‌دهد که روش نمونه‌گیری بالا هیچ بهبود واقعی در سطح زیر منحنی نداشت. SMOTE بهبودی را در ­مجموعه ارزیابی نشان داد، اما افزایش سطح زیر منحنی ROC در مجموعه آزمایشی بزرگ‌تر تکرار نشد. نمونه برداری ساده از داده‌ها نیز تأثیر محدودی بر عملکرد مدل داشت. با این حال، نمونه برداری پایین در داخل مدل جنگل تصادفی دارای مناطق قوی زیر منحنی ROC در هر دو مجموعه داده بود. این ممکن است به دلیل استفاده از تحقق مستقل کلاس اکثریت در هر درخت باشد. در کل، نتایج مختلط است. در حالی که این پیشرفت‌ها متوسط هستند، روش‌های نمونه‌گیری این مزیت را دارند که مبادلات بهتری بین حساسیت و ویژگی (برخلاف اصلاح برش‌ها یا احتمالات قبلی) فراهم می‌کنند.

آموزش حساس به هزینه

به جای بهینه‌سازی معیارهای عملکرد معمولی، مانند دقت یا ناخالصی، برخی از مدل‌ها می‌توانند به‌طور متناوب یک تابع هزینه یا ضرر را بهینه کنند که به‌طور متفاوت انواع خاصی از خطاها را وزن می‌کند. مثلا ممکن است باشد



<D *>* LU

0.0 0.2 0.4 0.6 0.8 1.0

1 - Specificity

100-

80-

60-

40-

20-

0-

0

20

40

60

80

100

٪ مشتریان ارزیابی شده است

شکل 16. 4: *بالا* : تست منحنی‌های ROC مجموعه‌ای برای سه مدل جنگل تصادفی.

*پایین* : نمودارهای بالابر مربوطه برای این باور که طبقه‌بندی اشتباه رویدادهای واقعی (منفی‌های کاذب) *X* برابر پرهزینه تر از پیش‌بینی نادرست غیرحوادث است (مثبت‌های غلط) است. گنجاندن ­هزینه‌های خاص در طول آموزش مدل ممکن است مدل را به سمت کمتر سوق دهد

جدول 16. 3: خلاصه‌ای از نتایج برای مدل‌های تصادفی جنگل با مکانیسم‌های نمونه‌گیری مختلف

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| روش | ارزیابی | تست | | |
| ROC | ویژگی حساسیت ROC | | |
| اصلی | 0. 757 | 0. 738 | 64. 4 | 75. 9 |
| نمونه برداری پایین | 0. 794 | 0. 730 | 81. 4 | 70. 3 |
| نمونه برداری پایین (داخلی) | 0. 792 | 0. 764 | 78. 0 | 68. 3 |
| نمونه‌گیری بالا | 0. 755 | 0. 739 | 71. 2 | 68. 1 |
| SMOTE | 0. 767 | 0. 747 | 78. 0 | 67. 7 |

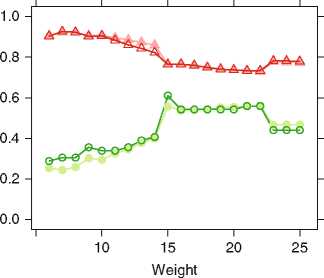
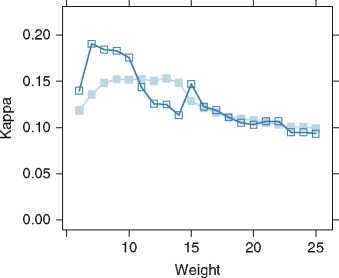
مقادیر حساسیت و ویژگی مجموعه آزمون با استفاده از برش بهینه حاصل از منحنی ROC مجموعه ارزیابی تعیین شد.

کلاس‌های مکرر برخلاف استفاده از برش‌های جایگزین، هزینه‌های نابرابر می‌تواند بر پارامترهای مدل تأثیر بگذارد و بنابراین پتانسیل ایجاد بهبودهای واقعی در طبقه‌بندی کننده را دارد.

برای مدل‌های ماشین بردار پشتیبان (SVM)، هزینه‌ها را می‌توان با کلاس‌های خاصی مرتبط کرد (برخلاف انواع خاص خطا). به یاد بیاورید که این مدل‌ها پیچیدگی را با استفاده از تابع هزینه کنترل می‌کنند که اگر نمونه‌ها در سمت نادرست مرز کلاس فعلی قرار گیرند، جریمه را افزایش می‌دهد. برای عدم تعادل طبقاتی، هزینه‌های نابرابر برای هر طبقه می‌تواند پارامترها را برای افزایش یا کاهش حساسیت مدل به کلاس‌های خاص تنظیم کند [( وروپولوس و همکاران.](#bookmark1027)  [1999](#bookmark1027) ). توجه داشته باشید که این رویکرد با رویکردی که انواع خاصی از خطاها می‌توانند هزینه‌های متفاوتی داشته باشند متفاوت است. برای ماشین‌های بردار پشتیبان (SVM)، می‌توان به کل کلاس اهمیت بیشتری داد. برای دو کلاس، این دو رویکرد مشابه هستند.

یکی از پیامدهای این رویکرد این است که احتمالات کلاس را نمی‌توان ­برای مدل، حداقل در پیاده‌سازی موجود، ایجاد کرد. بنابراین ما نمی‌توانیم منحنی ROC را محاسبه کنیم و باید از یک معیار عملکرد متفاوت استفاده کنیم. در عوض، اکنون از آماره، حساسیت و ویژگی کاپا برای ارزیابی تأثیر طبقات وزنی استفاده می‌کنیم.

برای مدل SVM، ما هر دو مدل وزنی و بدون وزن را تحلیل کردیم و نتایج عملکرد را مقایسه کردیم. هنگام تنظیم مدل‌ها، ­SVM بدون وزن به مقدار نسبتاً زیادی از پارامتر هزینه SVM (256) برای بهینه‌سازی آمار کاپا نیاز داشت. با اعمال مدل بدون وزن در مجموعه آزمون، آماره کاپا 0. 121 با حساسیت متناظر 15. 5 درصد و ویژگی 95. 7 درصد بود. اثرات وزن‌های بین 6 تا 25 برای مجموعه آموزشی (با استفاده از نمونه‌گیری مجدد) و مجموعه ارزیابی ارزیابی شد. در محدوده وزن انتخاب شده، عملکرد مدل بر اساس آماره کاپا، حساسیت و ویژگی برای دو مجموعه داده بسیار مشابه بود (شکل 1).  [16. 5 )](#bookmark774) . وزنه‌های کوچکتر کاپا را بهینه می‌کنند، در حالی که در حد متوسط



Kappa (CV)

Kappa (Eval.) □

Fig. 16.5: Tuning results for class weighted support mined with cross-validation and the evaluation set

Sens (CV) Spec (CV)

Sensitivity (Eval.) o Specificity (Eval.) △

vector machine deter-

وزن‌های بالا حساسیت را بهینه می‌کند. با استفاده از طرح، مدل ساز می‌تواند ویژگی‌های عملیاتی مناسب مدل را برای مسأله مورد نظر همانطور که در بخش توضیح داده شده است، تصمیم بگیرد.  [16. 3](#bookmark757) .

علاوه بر این، بسیاری از مدل‌های درخت طبقه‌بندی می‌توانند هزینه‌های متفاوتی را شامل شوند، از جمله درختان CART و C5. 0. هزینه بالقوه یک پیش‌بینی چندین عامل را در نظر می‌گیرد [( جانسون و ویچرن 2001](#bookmark1019) ):

هزینه یک اشتباه خاص

احتمال انجام آن اشتباه

احتمال قبلی کلاس ها

برای نشان دادن اینکه چگونه می‌توان هزینه‌ها را در نظر گرفت، اجازه دهید به مفاهیم و نمادهای استفاده شده در بخش بازگردیم.  [13. 6 .](#bookmark648) فرض کنید *n i* نشان دهنده احتمال قبلی ­یک نمونه در کلاس *i* باشد و فرض کنید *Pr* [ *j |i* ] احتمال پیش‌بینی اشتباه یک نمونه کلاس *i* به‌عنوان کلاس *j* باشد. برای یک مسئله دو کلاسه، احتمال کل طبقه‌بندی اشتباه یک نمونه پس از آن است

*Pr* [2 *1* 1] *n* i + *Pr* [1 *1* 2] *n 2*

در عمل ما به ندرت *Pr* [ *j |i* ] را می‌شناسیم و در عوض از احتمال تخمین زده شده کلاس i، *p i استفاده می‌*کنیم.

برای دو کلاس، معادله فوق می‌تواند به یک قانون برای طبقه‌بندی یک نمونه به کلاس 1 تبدیل شود، اگر

ص 1 > n 2

P 2 n 1

این قانون فرض می‌کند که هزینه‌های طبقه‌بندی اشتباه یک نمونه کلاس 1 به کلاس 2 یا یک نمونه کلاس 2 به کلاس 1 برابر است. به راحتی می‌توان معادله را برای در نظر گرفتن سناریویی که در آن هزینه‌ها یکسان نیست، گسترش داد.

فرض کنید *C (* *j |i* ) هزینه پیش‌بینی اشتباه یک نمونه کلاس *i* به‌عنوان کلاس *j* باشد. سپس کل هزینه مورد انتظار طبقه‌بندی اشتباه است

هزینه مورد انتظار = *C (2 |* 1) *Pr [2 |* 1] *n* 1 + *C (*1 *|* 2) *Pr* [1 *|* 2] *n 2*

و مرز تصمیم تجدید نظر شده برای طبقه‌بندی یک نمونه به کلاس 1 است

*P 1 > ( C (*1 *1* 2)   
*p 2 C (*2 *|* 1)

ni

برای پیشین‌های مساوی، کلاس اول فقط زمانی پیش‌بینی می‌شود که نسبت احتمال بیشتر از نسبت هزینه باشد. اگر *p* 1 = 0. 75 و *p* 2 = 0. 25، کلاس دوم زمانی پیش‌بینی می‌شود که هزینه پیش‌بینی اشتباه آن کلاس بیشتر از 3. 0 باشد.

هنگامی که هزینه هر نوع خطا مشخص شد، درختان CART می‌توانند آنها را در فرآیند آموزش بگنجانند.  [بریمن و همکاران ( 1984](#bookmark1012) ) استفاده از معیار جینی تعمیم یافته را مورد بحث قرار می‌دهد:

*جینی \** = *C (*1 *1* 2) *p* 1 (1 *- p* 1 ) + *C (*2 *1* 1) *p* 2 (1 *- p* 2 )   
= [ *C (*1 *|* 2) + *C (*2 *|* 1)] *p* 1 *ص* 2

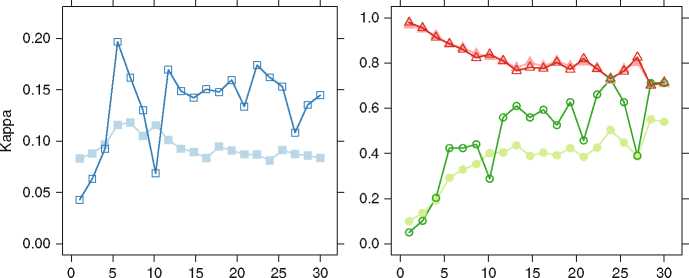
که در آن *p* 1 و *p* 2 احتمالات کلاس مشاهده شده ناشی از تقسیم هستند. توجه داشته باشید که در این مورد، هنگام استفاده از شاخص جینی برای تعیین تقسیم‌ها، این دو هزینه با هم جمع می‌شوند. با بیش از دو کلاس، همین مسئله به وجود می‌آید. به این ترتیب، استفاده از هزینه‌ها با ماتریس جینی، هزینه‌ها را متقارن می‌کند، به این معنا که هزینه‌های *C (* *i|j* ) و *C (* *j|i* ) هنگام تعیین هزینه‌های کلی برای تقسیم میانگین می‌شوند.  [بریمن و همکاران ( 1984](#bookmark1012) ) همچنین اشاره می‌کند که در برخی موارد، استفاده از هزینه‌های نابرابر معادل استفاده از احتمالات قبلی اصلاح شده می‌شود. در مثال‌های خود، آنها اشاره می‌کنند که درختان حساس به هزینه کوچک‌تر از درختانی هستند که با روش اسمی تولید می‌شوند. استدلال این بود که در مرحله رشد، درخت حریصانه شکافته می‌شود تا پیش‌بینی‌کننده‌های دقیقی از خطاهای بالقوه پرهزینه ایجاد کند، اما به سختی کار نمی‌کند تا در مورد خطاهای باقی‌مانده که هزینه‌های کمتری دارند، دقیق باشد.

برای درختان (و قوانین)، احتمالات کلاس (یا مقادیر اطمینان) پیش‌بینی‌شده ممکن است با پیش‌بینی‌کننده‌های کلاس گسسته زمانی که ­هزینه‌های نابرابر استفاده می‌شود، سازگار نباشد. پیش‌بینی کلاس نهایی برای یک نمونه تابعی از احتمال کلاس *و* ساختار هزینه است. همانطور که قبلا نشان داده شد، ­توانایی‌های احتمالی کلاس در گره پایانه ممکن است به‌طور قابل ملاحظه‌ای به نفع یک کلاس باشد، اما هزینه مورد انتظار زیادی نیز دارد. به همین دلیل، بین مقادیر اطمینان و کلاس پیش‌بینی شده قطع ارتباط وجود دارد. بنابراین، احتمالات کلاس ساده (یا مقادیر اطمینان) نباید در این شرایط استفاده شوند.

برای نشان دادن آموزش مدل حساس به هزینه، درخت‌های CART منفرد با داده‌ها با هزینه‌های مساوی و همچنین طیف وسیعی از مقادیر هزینه (2. 5 تا 30) مشابه مدل SVM وزن‌دار قبلی، متناسب شدند. شکل [16. 6](#bookmark42) نتایج مجموعه‌های آموزشی و ارزشیابی را نشان می‌دهد. روندهای نشان داده شده در این کرت‌ها پر سر و صدا هستند (به دلیل ناپایداری درختان CART)، به ویژه برای حساسیت که دارای

کاپا (CV) ■ حساسیت (CV) ویژگی (CV)

کاپا (Eval. ) □ حساسیت (Eval. ) o خاص (Eval. ) △



هزینه هزینه

شکل 16. 6: نتایج تنظیم برای مدل‌های درختی طبقه‌بندی حساس به هزینه ­که با اعتبارسنجی متقاطع و مجموعه ارزیابی تعیین می‌شود.

حجم نمونه کوچکتر تخمین نمونه‌گیری مجدد حساسیت بدبینانه تر از مجموعه ارزیابی بود. نتایج نشان می‌دهد که حساسیت و ویژگی مدل در مقادیر هزینه بالا شروع به همگرایی می‌کند.

به یاد داشته باشید که C5. 0 از معیار Gini برای تقسیم استفاده نمی‌کند. با هزینه‌ها، از فرمول قبلی استفاده می‌شود و احتمالات قبلی برابر را فرض می‌کند [[58]](#footnote-58):

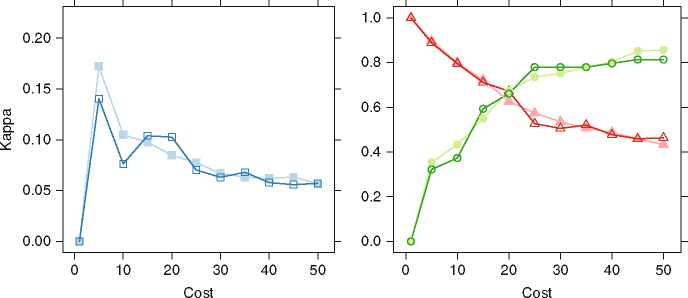
هزینه مورد انتظار = *C (*2 *|* 1) *p* 1 + *C (*1 | *2* ) *p2*

مجدداً، نمونه‌های جدید به‌عنوان کلاس مرتبط با کمترین هزینه مورد انتظار پیش‌بینی می‌شوند.

برای C5. 0، مدل‌ها بر روی نوع مدل (یعنی درختان یا قوانین) و تعداد تکرارهای تقویت کننده تنظیم شدند. شکل [16. 7](#bookmark778) الگوهای ­یادگیری حساس به هزینه را با هزینه‌های منفی کاذب از 5 تا 50 نشان می‌دهد. مانند تحلیل SVM، منحنی‌ها بین مجموعه‌های آموزش و ارزیابی سازگار هستند. برای بهینه‌سازی تطابق از طریق آماره کاپا، مقادیر هزینه‌های کوچکتری مورد نیاز است و برای افزایش حساسیت، هزینه‌های متوسط تا زیاد مورد نیاز است. در اینجا، یک مبادله واضح بین حساسیت و ویژگی وجود دارد، اما به نظر می‌رسد که حساسیت برای مقادیر هزینه‌های بیشتر از 25 یا 30 بالاست.

کاپا (CV) ■ حساسیت (CV) ویژگی (CV)

کاپا (Eval. ) □ حساسیت (Eval. ) o خاص (Eval. ) △



شکل 16. 7: پروفایل‌های عملکرد برای مدل‌های C5. 0 با استفاده از یادگیری حساس به هزینه

16. 9 محاسبات

این بخش از بسته‌های زیر استفاده می‌کند: caret، C50، DMwR، DWD، kernlab، pROC و rpart.

داده‌های بیمه در بسته DWD موجود است و به صورت زیر قابل بارگیری است:

*کتابخانه (DWD)*

*داده (ticdata)*

چندین متغیر عامل در مجموعه داده وجود دارد. بسیاری از سطوح فاکتور دارای کاراکترهای غیر استاندارد مانند کاما "%" و مقادیر دیگر هستند. وقتی اینها به ستون‌های متغیر ساختگی تبدیل می‌شوند، مقادیر قوانین نامگذاری متغیرهای جدید را نقض می‌کنند. برای دور زدن این موضوع، اسامی را دوباره رمزگذاری می‌کنیم تا ساده‌تر باشند:

*recodeLevels <- تابع(x) +{*

*+ x <- as. numeric(x)*

*+ ## صفرها را به نسخه متنی اضافه کنید :*

*+ x <- gsub(" "، "0",format(as. numeric(x)))*

*+ فاکتور (x)*

*+}*

*## پیدا کنید کدام ستون‌ها فاکتورهای منظم یا فاکتورهای مرتب هستند*

*isOrdered <- unlist(lapply(ticdata, is. ordered))*

*isFactor <- unlist(lapply(ticdata, is. factor))*

*convertCols <- names(isOrdered)[isOrdered | isFactor]*

*for(i در convertCols) ticdata[,i] <- recodeLevels(ticdata[,i])*

*## سطح* " *بیمه* " *را اولین سطح عامل قرار دهید*

*ticdata$CARAVAN <- factor(as. character(ticdata$CARAVAN)، + level = rev (levels(ticdata$CARAVAN)))*

مجموعه آموزش و آزمون با استفاده از نمونه‌گیری تصادفی طبقه‌ای ایجاد شد:

*کتابخانه (کارت)*

*## ابتدا، مجموعه آموزشی را تقسیم کنید*

*set. seed(156)*

*split1 <- createDataPartition(ticdata$CARAVAN، p = 0. 7)[[1]]*

*دیگر <- ticdata[-split1,]*

*آموزش <- ticdata[ split1,]*

*## حالا مجموعه‌های ارزیابی و تست را ایجاد کنید*

*set. seed(934)*

*split2 <- createDataPartition(other$CARAVAN، p = 1/3)[[1]]*

*ارزیابی <- دیگر[ split2,]*

*آزمایش <- other[-split2,]*

*## نام‌های پیش‌بینی را تعیین کنید*

*پیش‌بینی‌کننده‌ها <- names(training)[names(training) != "CARAVAN"]*

متغیرهای ساختگی برای چندین مدل مناسب در این بخش مفید هستند. تابع randomForest یک محدودیت دارد که همه عوامل پیش‌بینی نباید بیش از 32 سطح داشته باشند. پیش‌بینی نوع مشتری 39 سطح دارد، بنابراین یک مجموعه پیش‌بینی از متغیرهای ساختگی برای این و سایر مدل‌ها با استفاده از تابع model. matrix ایجاد می‌شود:

*## ستون اول وقفه است که حذف می‌شود:*

*trainingInd <- data. frame(model. matrix(CARAVAN ~. ,*

*+ داده = آموزش))[،-1]*

*ارزیابیInd <- data. frame(model. matrix(CARAVAN ~. ,*

*+ داده = ارزیابی))[،-1]*

*testingInd <- data. frame(model. matrix(CARAVAN ~. ,*

*+ داده = آزمایش))[،-1]*

*## نتیجه را دوباره به مجموعه داده اضافه کنید*

*TrainingInd$CARAVAN <- train$CARAVAN*

*ارزیابیInd$CARAVAN <- ارزیابی$CARAVAN*

*testingInd$CARAVAN <- testing$CARAVAN*

*## یک مجموعه پیش‌بینی بدون توزیع‌های بسیار پراکنده و نامتعادل تعیین کنید:*

*isNZV <- nearZeroVar(trainingInd)*

*noNZVSet <- names(trainingInd)[-isNZV]*

برای به دست آوردن معیارهای مختلف عملکرد، دو تابع wrapper ایجاد شد:

*## برای دقت، کاپا، ناحیه زیر منحنی ROC،*

*## حساسیت و ویژگی:*

*fiveStats <- function(. . . ) c(twoClassSummary(. . . )*

*+ خلاصه پیش فرض (. . . ))*

*## همه چیز به جز ناحیه زیر منحنی ROC:*

*تابع fourStats <- (داده، lev = سطوح (data$obs)، مدل = NULL)*

*+{*

*+*

*+ accKapp <- postResample(data[, "pred"], data[, "obs"])*

*+ خارج <- c(accKapp,*

*+ حساسیت (داده[، "pred"]، داده[، "obs"]، lev[1])،*

*+ ویژگی (داده[، "pred"]، داده[، "obs"]، lev[2]))*

*+ names(out)[3:4] <- c("Sens"، "Spec")*

*+ بیرون*

*+}*

دو تابع کنترلی برای موقعیت‌هایی که احتمالات کلاس را می‌توان ایجاد کرد و زمانی که نمی‌توان ایجاد کرد، ایجاد می‌شود:

*ctrl <- trainControl(روش = "cv"،*

*+ classProbs = TRUE،*

*+ summaryFunction = پنج آمار،*

*+ verboseIter = TRUE)*

*ctrlNoProb <- ctrl*

*ctrlNoProb$summaryFunction <- fourStats*

*ctrlNoProb$classProbs <- FALSE*

سه مدل پایه با نحو مطابقت داشتند:

*set. seed(1410)*

*rfFit <- train(CARAVAN ~. , data = trainingInd,*

*+ روش = "rf"،*

*+ trControl = ctrl،*

*+ ntree = 1500،*

*+ تنظیم طول = 5،*

*+ معیار = "ROC")*

*set. seed(1410)*

*lrFit <- train(CARAVAN ~. ,*

*+ داده = trainingInd[، noNZVSet]،*

*+ روش = "glm"،*

*+ trControl = ctrl،*

*+ معیار = "ROC")*

*set. seed(1401)*

*fdaFit <- train(CARAVAN ~. ، داده = آموزش،*

*+ روش = "fda"،*

*+ tuneGrid = data. frame (. درجه = 1،. nprune = 1:25)،*

*+ معیار = "ROC"،*

*+ trControl = ctrl)*

*>*

یک چارچوب داده برای قرار دادن پیش‌بینی‌کننده‌های مدل‌های مختلف استفاده می‌شود:

*> evalResults <- data. frame(CARAVAN = ارزیابی$CARAVAN)*

*> evalResults$RF <- predict(rfFit,*

|  |  |
| --- | --- |
| *+ داده‌های جدید*  *+ نوع =* | *= ارزیابیInd, "prob")[,1]* |

*evalResults$FDA <- predict(fdaFit,*

*+ داده‌های جدید = ارزیابی[، پیش‌بینی ها]،*

*+ type = "Prob")[,1]*

*evalResults$LogReg <- predict(lrFit,*

*+ newdata = valuationInd[, noNZVSet]،*

*+ type = "Prob")[,1]*

منحنی‌های ROC و lift از این اشیا ایجاد می‌شوند. مثلا:

*کتابخانه (pROC)*

*rfROC <- roc(evalResults$CARAVAN، evalResults$RF،*

*+ سطوح = دور (سطوح (evalResults$CARAVAN)))*

*## برای مدل‌ها برچسب ایجاد کنید:*

*آزمایشگاه‌ها <- c(RF = "جنگل تصادفی"، LogReg = "رگرسیون لجستیک"،*

*+ FDA = "FDA (MARS)")*

*lift1 <- lift(CARAVAN ~ RF + LogReg + FDA، داده = evalResults،*

*+ برچسب‌ها = آزمایشگاه ها)*

*rfROC*

صدا زدن:

roc. default(response = evalResults$CARAVAN، پیش‌بینی = evalResults$RF، سطوح = rev(سطوح(evalResults$CARAVAN)))

داده ها: evalResults$RF در 924 کنترل (evalResults$CARAVAN غیر بیمه) < 59 مورد (evalResults$CARAVAN بیمه).

مساحت زیر منحنی: 0. 7569

*آسانسور 1*

صدا زدن:

lift. formula(x = CARAVAN ~ RF + LogReg + FDA، داده = evalResults، برچسب‌ها = آزمایشگاه ها)

مدل‌ها: جنگل تصادفی، رگرسیون لجستیک، FDA (MARS)

رویداد: بیمه (6%)

برای رسم منحنی ها:

*نمودار (rfROC، legacy. axes = TRUE)*

*xyplot(lift1,*

*+ ylab = "% رویدادها پیدا شد"، xlab = "% مشتریان ارزیابی شدند"،*

*+ lwd = 2، نوع = "l")*

قطع‌های جایگزین

پس از ایجاد منحنی ROC، چندین تابع در بسته pROC وجود دارد که می‌توان از آنها برای بررسی قطع‌های احتمالی استفاده کرد. تابع coords نقاط روی منحنی ROC را برمی گرداند و همچنین برش‌های جدید را استخراج می‌کند. آرگومان‌های اصلی x هستند که مشخص می‌کند چه چیزی باید برگردانده شود. مقدار x = "همه" مختصات منحنی و برش‌های مرتبط با آنها را برمی گرداند. مقدار "بهترین" یک برش جدید ایجاد می‌کند. استفاده از x = "best" در ارتباط با best. method (یا "youden" یا "closest. topleft" ) می‌تواند آموزنده باشد:

*> rfThresh <- coords(rfROC، x = "best"، best. method = "closest. topleft") > rfThresh*

حساسیت ویژگی آستانه 0. 06433333 0. 75865801 0. 64406780

برای این، کلاس‌های پیش‌بینی شده جدید را می‌توان محاسبه کرد:

*newValue <- factor(ifelse(evalResults$RF > rfThresh,*

*+ "بیمه"، "غیر بیمه")،*

*+ سطوح = سطوح (evalResults$CARAVAN))*

روش‌های نمونه گیری

بسته caret دارای دو عملکرد downSample و upSample است که فرکانس‌های کلاس را دوباره تنظیم می‌کند. هر کدام آرگومان‌هایی را برای پیش‌بینی‌کننده‌ها (که x نامیده می‌شوند ) و کلاس نتیجه ( y ) می‌گیرند. هر دو تابع یک چارچوب داده را با نسخه نمونه از مجموعه آموزشی برمی‌گردانند:

*set. seed(1103)*

*upSampledTrain <- upSample(x = training[,predictors],*

*+ y = آموزش$CARAVAN،*

*+ ## نام متغیر کلاس را یکسان نگه دارید:*

*+ yname = "کاروان")*

*کم نور (آموزش)*

[1] 6877 86

*> dim (upSampledTrain)*

[1] 12932 86

*جدول (upSampledTrain$CARAVAN)*

بیمه غیر بیمه ای

6466

6466

تابع down-sampling دستور یکسانی دارد. یک تابع برای SMOTE را می‌توان در بسته DMwR یافت. یک فرمول مدل را به‌عنوان ورودی، همراه با پارامترهایی (مانند مقدار بیش و کم نمونه برداری و تعداد همسایگان) می‌گیرد. نحو اصلی است

*کتابخانه (DMwR)*

*set. seed(1103)*

*smoteTrain <- SMOTE(CARAVAN ~. ، داده = آموزش)*

*کم نور (smoteTrain)*

[1] 2877 86

*> جدول (smoteTrain$CARAVAN)*

بیمه غیر بیمه ای

1644

1233

این مجموعه داده‌ها را می‌توان به‌عنوان ورودی در کد مدل‌سازی قبلی استفاده کرد.

آموزش حساس به هزینه

SVM‌های دارای وزن کلاس را می‌توان با استفاده از بسته kernlab ایجاد کرد. سینتکس تابع ksvm مانند توضیحات قبلی است، اما آرگومان class. weights استفاده می‌شود. تابع آموزش مالیات همگام مشابهی دارد :­

*کتابخانه (kernlab)*

*## ما در محدوده هزینه زیادی آموزش خواهیم داد، بنابراین سیگما را از قبل محاسبه می‌کنیم*

*پارامتر ## و ایجاد یک شبکه تنظیم سفارشی:*

*set. seed(1157)*

*sigma <- sigest(CARAVAN ~. , data = trainingInd[, noNZVSet], frac = 0. 75) > names(sigma) <- NULL*

*svmGrid <- data. frame(. sigma = sigma[2]،*

*+ C = 2~ seq(-6، 1، طول = 15))*

*## احتمالات کلاس را نمی‌توان با وزن کلاس ایجاد کرد، بنابراین*

*## از شی کنترلی* " *ctrlNoProb* " *برای جلوگیری از تخمین زدن استفاده کنید*

*## منحنی ROC.*

*set. seed(1401)*

*SVMwts <- آموزش(CARAVAN ~. ,*

*+ داده = trainingInd[، noNZVSet]،*

*+ روش = "svmRadial"،*

*+ tuneGrid = svmGrid،*

*+ preProc = c ("مرکز"، "مقیاس")،*

*+ class. weights = c (بیمه = 18، بیمه غیر بیمه = 1)،*

*+ معیار = "حس"،*

*+ trControl = ctrlNoProb)*

*SVMwts*

6877 نمونه

203 پیش‌بینی

2 کلاس: "بیمه"، "بیمه غیر بیمه"

پیش پردازش: متمرکز، مقیاس شده

نمونه‌گیری مجدد: اعتبارسنجی متقاطع (10 برابر)

خلاصه حجم نمونه: 6189، 6190، 6190، 6189، 6189، 6189،

نمونه برداری مجدد از نتایج در پارامترهای تنظیم:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| سی | دقت | کاپا | حس | مشخصات |
| 0. 0156 | 0. 557 | 0. 0682 | 0. 742 | 0. 545 |
| 0. 0221 | 0. 614 | 0. 0806 | 0. 691 | 0. 609 |
| 0. 0312 | 0. 637 | 0. 0864 | 0. 669 | 0. 635 |
| 0. 0442 | 0. 644 | 0. 0883 | 0. 662 | 0. 643 |
| 0. 0625 | 0. 658 | 0. 0939 | 0. 657 | 0. 658 |
| 0. 0884 | 0. 672 | 0. 0958 | 0. 633 | 0. 674 |
| 0. 125 | 0. 684 | 0. 101 | 0. 625 | 0. 688 |
| 0. 177 | 0. 7 | 0. 106 | 0. 611 | 0. 705 |
| 0. 25 | 0. 711 | 0. 108 | 0. 591 | 0. 719 |
| 0. 354 | 0. 724 | 0. 111 | 0. 572 | 0. 734 |
| 0. 5 | 0. 737 | 0. 112 | 0. 543 | 0. 75 |

|  |  |
| --- | --- |
| 0. 707 0. 75  1 0. 765  1. 41 0. 776  2 0. 791 | 0. 109 0. 506 0. 765  0. 104 0. 46 0. 785  0. 097 0. 416 0. 799  0. 102 0. 394 0. 817 |
| از پارامتر تنظیم Sens برای مقادیر نهایی استفاده شد | 'سیگما' در مقدار 0. 00245 ثابت نگه داشته شد مدل بهینه را با استفاده از بزرگترین مقدار انتخاب کنید.  برای مدل استفاده شده C = 0. 0156 و سیگما = 0. 00245 بود. |

(ستون‌های انحراف استاندارد برای صرفه‌جویی در فضا نشان داده نشدند) پیش‌بینی از نحوی مشابه مدل‌های بدون وزن استفاده می‌کند.

برای مدل‌های CART حساس به هزینه، بسته rpart با آرگومان parms استفاده می‌شود که فهرستی از گزینه‌های برازش است. یک گزینه، ضرر، می‌تواند ماتریسی از هزینه‌ها را در نظر بگیرد:

*costMatrix <- ماتریس(c(0، 1، 20، 0)، ncol = 2)*

*rownames(costMatrix) <- سطوح (training$CARAVAN)*

*colnames(costMatrix) <- سطوح (training$CARAVAN) > costMatrix*

بیمه غیر بیمه ای

بیمه 0 20

بیمه غیر بیمه 1 0

در اینجا، هزینه منفی کاذب 20 برابر بیشتر از مثبت کاذب خواهد بود. متناسب با مدل:

*> کتابخانه (rpart)*

*> set. seed(1401)*

*> cartهزینه‌ها <- آموزش(x = آموزش[,پیش‌بینی‌کنندگان]،*

|  |  |
| --- | --- |
| *+*  *+* | *y = train$CARAVAN، روش = "rpart"،* |
| *+*  *+* | *trControl = ctrlNoProb، معیار = "کاپا"،* |
| *+*  *+* | *طول = 10، parms = لیست (از دست دادن = هزینه ماتریس))* |

مشابه مدل ماشین بردار پشتیبان، نحو برای تولید پیش‌بینی‌کننده‌های کلاس مانند مدل اسمی است. با این حال، احتمالات کلاس ایجاد شده از این مدل ممکن است با کلاس‌های پیش‌بینی شده (که تابعی از هزینه و احتمالات هستند) مطابقت نداشته باشد.

C5. 0 با گرفتن ماتریس هزینه، نحوی مشابه با rpart دارد، اگرچه این تابع از جابجایی ساختار ماتریس هزینه استفاده شده توسط rpart استفاده می‌کند :

*ماتریس c5 <- ماتریس(c(0، 20، 1، 0)، ncol = 2)*

*rownames(c5Matrix) <- level (training$CARAVAN)*

*colnames(c5Matrix) <- سطوح (training$CARAVAN)*

*ماتریس c5*

بیمه غیر بیمه ای

بیمه 0 1

بیمه بدون بیمه 20 0

*> کتابخانه (C50)*

*> set. seed(1401)*

*> C5Cost <- train(x = آموزش[، پیش‌بینی ها]،*

*y = training$CARAVAN,*

*+*

*+*

*+*

*+*

*+*

*روش = "C5. 0"، معیار = "کاپا"، هزینه = ماتریس c5،*

*trControl = ctrlNoProb)*

هنگام به کارگیری هزینه‌ها، تابع پیش‌بینی برای این مدل فقط کلاس‌های گسسته را تولید می‌کند (یعنی بدون احتمال).

تمرینات

مجموعه داده‌های "بزرگسال" در مخزن یادگیری ماشین UCI از سوابق سرشماری مشتق شده است. [[59]](#footnote-59) در این داده‌ها، هدف پیش‌بینی این است که آیا درآمد یک فرد بزرگ (در سال 1994 بیش از 50 هزار دلار تعریف شد) یا کم است. پیش‌بینی‌کننده‌ها شامل سطح تحصیلات، نوع شغل (به‌عنوان مثال، هرگز کار نکرده و دولت محلی)، سود / زیان سرمایه، ساعات کار در هفته، کشور بومی و غیره است. [[60]](#footnote-60) پس از فیلتر کردن داده‌هایی که کلاس نتیجه ناشناخته است، 48842 رکورد باقی مانده بود. اکثریت داده‌ها با سطح درآمد کمی (75. 9 درصد) مرتبط بودند.

داده‌ها در بسته قوانین موجود است و نسخه مناسب را می‌توان با استفاده از داده (AdultUCI) بارگیری کرد.

داده‌ها را بارگذاری کنید و پیش‌بینی‌کننده‌ها را از نظر توزیع و همبستگی‌های بالقوه آنها بررسی کنید.

تقسیم مناسب داده‌ها را تعیین کنید.

چندین مدل طبقه‌بندی برای این داده‌ها بسازید. آیا نتایج به نفع طبقه کم درآمد است؟

آیا معاوضه خوبی بین حساسیت و ویژگی وجود دارد؟

از روش‌های نمونه‌گیری برای بهبود برازش مدل استفاده کنید.

آیا مدل‌های حساس به هزینه به عملکرد کمک می‌کنند؟

داده‌های بازاریابی مستقیم از [لاروس](#bookmark1020) [( 2006](#bookmark1020) ، فصل 7) قبلاً ­در فصل مورد بحث قرار گرفت.  [11](#bookmark497) را می‌توان در وب سایت نویسنده یافت. [[61]](#footnote-61) هدف از تحلیل پیش‌بینی این بود که کدام مشتریان به یک فرصت تبلیغاتی از طریق پست پاسخ خواهند داد.

از 65220 مشتری در مجموعه داده، 16. 6٪ به تبلیغات پاسخ دادند.

پیش‌بینی‌کننده‌ها در مجموعه داده‌ها شامل

عادات خرج کردن، به‌طور کلی و تفکیک شده بر اساس ماه و مکان‌های فروشگاهی که اغلب بازدید می‌شود

انواع محصولات خریداری شده

زمان بین بازدیدها

نرخ پاسخ تبلیغات تاریخی

عضویت در خوشه مشتری از پیش تعریف شده

Larose [( 2006](#bookmark1020) ) عدم تعادل طبقاتی را برای این مسأله مورد بحث قرار داد و ­چندین تکنیک را برای دستیابی به نتایج مؤثر نشان داد.

داده‌ها را در R بخوانید، تحلیل‌های اکتشافی انجام دهید و بهترین روش را برای رمزگذاری پیش‌بینی‌کننده‌ها تعیین کنید.

یک تقسیم مناسب از داده‌ها را تعیین کنید و چندین مدل طبقه‌بندی برای این داده‌ها بسازید.

نمودارهای بالابر را برای درک یک استراتژی احتمالی در رابطه با تعداد مشتریانی که باید با آنها تماس گرفت تا حدود 60 درصد از پاسخ دهندگان را جذب کنید، بسازید.

از روش‌های نمونه‌گیری با چندین مدل استفاده کنید. آیا این مدل‌ها نمودارهای بالابر بهتری دارند و آیا می‌توان از آنها برای تماس با مشتریان کمتری برای دستیابی به نرخ پاسخگویی 60 درصدی (از کسانی که تبلیغ را دریافت کردند) استفاده کرد؟

فصل 17

مطالعه موردی: برنامه ریزی شغلی

محیط‌های محاسباتی با کارایی بالا (HPC) توسط بسیاری از سازمان‌های فناوری و تحقیقاتی برای تسهیل محاسبات در مقیاس بزرگ استفاده می‌شود. یک محیط HPC معمولاً از شبکه‌های «گره‌های محاسباتی» متعددی تشکیل می‌شود ­که برای انجام محاسبات مورد نیاز کاربران کار می‌کنند. اینها می‌توانند به روش‌های مختلفی ساختار شوند، مانند شبکه‌ای از رایانه‌های زیادی که هر کدام پردازنده‌های کمتری دارند ­یا شبکه‌ای از رایانه‌های کمتر با پردازنده‌های زیاد. مقدار حافظه در هر یک ممکن است از محیطی به محیط دیگر متفاوت باشد و اغلب بر اساس تعادل بین منابع موجود و ماهیت انواع محاسبات مورد نیاز یک موسسه ساخته می‌شود.

یک واحد خاص از محاسبات (که در اینجا به‌طور کلی *کار نامیده می‌*شود) می‌تواند توسط کاربران برای اجرا در محیط HPC راه اندازی شود. مسئله این است که در بسیاری از موارد، تعداد زیادی برنامه به صورت همزمان اجرا می‌شود و محیط مجبور است این مشاغل را به گونه‌ای مدیریت کند که نتایج شغلی را به بهترین نحو بازگرداند. این ممکن است یک کار پیچیده باشد. مثلا:

مواقعی وجود دارد که تعداد مشاغل موجود بیشتر از ظرفیت محیط است. در این موارد، برخی از مشاغل باید قبل از راه‌اندازی معلق بمانند تا منابع مناسب در دسترس باشند.

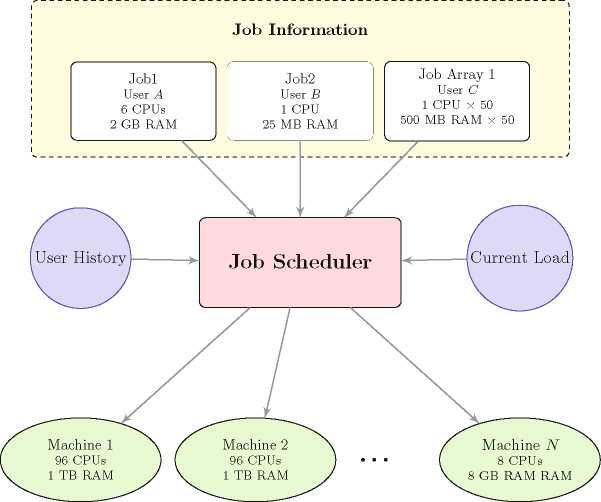
فرکانس ارسال توسط کاربران ممکن است متفاوت باشد. اگر یک کاربر تعداد زیادی کار را به‌طور همزمان ارسال کند، ممکن است برای سامانه نامطلوب باشد که به یک کاربر اجازه دهد اکثریت منابع را به هزینه سایر کاربران مصرف کند.

ممکن است با تمام کارهای محاسباتی به یک اندازه برخورد نشود. پروژه‌های با اولویت بالاتر ممکن است به دسترسی ممتازتری به منابع سخت افزاری نیاز داشته باشند. به‌عنوان مثال، مشاغل در یک شرکت داروسازی که برای پشتیبانی از ارسال مقررات حساس به زمان لازم است، ­ممکن است نیاز به اولویت بیشتری نسبت به مشاغل مرتبط با وظایف پژوهش محور داشته باشند.

یک راه حل رایج برای این مسأله استفاده از *زمانبندی کار* است - نرم‌افزاری که مشاغل را برای ارسال‌ها اولویت‌بندی می‌کند، منابع محاسباتی را مدیریت می‌کند و کارهای ارسالی را برای به حداکثر رساندن کارایی آغاز می‌کند. زمان‌بند می‌تواند یک سامانه صف را بر اساس عوامل متعددی مانند منابع مورد نیاز اجرا کند

\_17،

©



شکل 17. 1: یک مثال فرضی از زمانبندی کار در یک محیط محاسباتی با کارایی بالا

یک کار (به‌عنوان مثال، تعداد پردازنده‌های مورد نیاز، حافظه مورد نیاز)، اولویت پروژه و بار فعلی محیط. زمان‌بند ممکن است سابقه ارسال کاربران خاص را هنگام واگذاری کارها به منابع محاسباتی در نظر بگیرد.

شکل [17. 1](#bookmark796) شماتیکی از چنین سامانه‌ی را نشان می‌دهد. بالا سه شغل فرضی را نشان می‌دهد که همزمان ارسال می‌شوند. دو مورد اول هر کدام یک سری محاسبات دارند اما نیازهای متفاوتی دارند. به‌عنوان مثال، اولین کار می‌تواند یک محاسبات علمی باشد که محاسبات را در چندین پردازنده در یک ماشین فیزیکی پخش می‌کند. شغل دوم ممکن است یک پرس و جوی کوتاه پایگاه داده باشد که یک دانشمند به صورت تعاملی از یک وب سایت انجام می‌دهد. انتظار می‌رود این کار به سرعت با سربار کم اجرا شود. مورد سوم مجموعه منسجمی از محاسبات را نشان می‌دهد که به یکباره راه‌اندازی می‌شوند اما با مجموعه‌ای از محاسبات مستقل (مانند شبیه‌سازی) مطابقت دارند که از همان برنامه استفاده می‌کنند. این *آرایه شغلی* 50 محاسبه مستقل را راه‌اندازی می‌کند که می‌توانند روی ماشین‌های مختلف اجرا شوند، اما باید به‌عنوان یک موجودیت واحد مدیریت و نظارت شوند. مجدداً هدف زمانبندی تخصیص این مشاغل به سامانه به نحوی است که کارایی را به حداکثر برساند.

کارایی زمانبندی می‌تواند به‌طور قابل‌توجهی تحت تأثیر میزان و کیفیت اطلاعات شغلی که در زمان ارسال مشخص است، قرار گیرد. این به زمان‌بندی اجازه می‌دهد تا کارها را در زمان مناسب روی ماشین‌های مناسب قرار دهد. برای محاسباتی که چندین بار تکرار می‌شوند، ثبت اطلاعات در مورد مشاغل موجود و سپس پیش‌بینی منابع در مشاغل جدید ساده است. به‌عنوان مثال، پیش‌بینی زمان اجرا یا حداکثر حافظه مورد نیاز برای یک کار در زمان ارسال ممکن است مفید باشد. لازمه چنین پیش‌بینی این است که سریع و دقیق باشد. با این حال، الزامات دقت باید این واقعیت را در نظر بگیرد که اشتباهات پیش‌بینی احتمالاً تأثیر یکسانی بر ­عملکرد محیط ندارند. به‌عنوان مثال، اگر مقدار پیش‌بینی‌شده حافظه مورد نیاز به‌شدت دست‌کم گرفته شود، این می‌تواند باعث بارگیری بیش از حد منابع فیزیکی ماشین شود و ممکن است به شدت بر تمام کارهای آن ماشین تأثیر بگذارد. هزینه این نوع خطاها باید زیاد باشد تا زمانبند از این مسأله جلوگیری کند. مسأله برعکس درست نیست: برآورد بیش از حد ­نیازهای حافظه ممکن است باعث استفاده کم غیرطبیعی از یک منبع سخت افزاری شود، اما بر مشاغل موجود در سامانه تأثیر مهمی نخواهد گذاشت. هزینه‌ای باید به این خطا اختصاص داده شود، اما احتمالاً به اندازه سایر خطاها نیست.

به‌عنوان مثال، Pfizer تعداد زیادی کار را در محیط HPC خود اجرا می‌کند. یک دسته از مشاغل برای مشخص کردن ترکیبات به‌طور منظم اجرا می‌شود که برخی از آنها می‌توانند از نظر محاسباتی سنگین باشند. در طی یک دوره زمانی، اطلاعات استفاده از منابع ­در مورد این دسته از مشاغل ثبت شد. علاوه بر این، چندین ویژگی کار وجود دارد که می‌توان در زمان راه اندازی کار جمع‌آوری کرد:

پروتکل: چندین تکنیک تحلیلی مختلف، به نام *پروتکل،* می‌تواند برای محاسبه نتایج برای هر ترکیب استفاده شود. پروتکل‌ها به صورت حروف A تا O کدگذاری می‌شوند. پروتکل J بیشترین استفاده را داشت (22. 8٪ از تمام مشاغل) در حالی که پروتکل K به ندرت (0. 1٪) فراخوانی می‌شد.

تعداد ترکیبات: تعداد ترکیبات پردازش شده توسط کار. این عدد در بین داده‌ها بسیار متفاوت بود.

تعداد فیلدهای ورودی: هر وظیفه می‌تواند تعدادی فیلد ورودی مختلف را در داده‌ها پردازش کند، بسته به اینکه دانشمند چه چیزی را می‌خواهد تحلیل کند. این پیش‌بینی نیز به درستی منحرف است.

تعداد تکرارها: هر پروتکل می‌تواند برای تعداد از پیش تعیین شده تکرار اجرا شود. پیش فرض 20 تکرار است که بیشتر در این داده‌ها رخ می‌دهد.

تعداد مشاغل معلق: شمارشی از تعداد مشاغل معلق در زمان راه اندازی ثبت شد. این به معنای اندازه‌گیری حجم کاری محیط ­در زمان پرتاب است. در صورت مساوی بودن همه موارد دیگر، تعداد کارهای معلق نباید مستقیماً بر زمان اجرای کارها تأثیر بگذارد، اما ممکن است میزان منابع مورد استفاده در آن زمان را تحت تأثیر قرار دهد. برای این داده‌ها، بیشتر مشاغل در زمانی راه اندازی شدند که منابع موجود وجود داشت، بنابراین هیچ شغلی معلق نبود. با این حال، تعداد کمی از درخواست‌ها زمانی که سخت‌افزار مورد تقاضا بود (هزاران شغل در حال حاضر معلق بودند) انجام شد.

زمان روز: زمانی از روز (زمان استاندارد شرقی) که کار راه اندازی شد (0 تا 24). این توزیع چندوجهی است که نشان دهنده آنلاین شدن کاربران در سه منطقه زمانی مختلف در دو قاره است.

روز هفته: روزی از هفته که کار راه اندازی شد.

زمان اجرا برای هر کار ثبت شد. زمان صرف شده در حالت‌های معلق یا معلق در این مقادیر محاسبه نمی‌شود. در حالی که نتیجه ماهیت مستمر است، زمان‌بندی‌کننده نیاز به یک نمایش کیفی برای این اطلاعات دارد. مشاغل باید به‌عنوان بسیار سریع (1 متر یا کمتر)، سریع (1-50 متر)، متوسط (5-30 متر)، یا طولانی (بیش از 30 متر) طبقه‌بندی شوند. اکثر مشاغل در دو گروه بسیار سریع (51. 1٪) یا دسته سریع (31. 1٪) قرار می‌گیرند در حالی که تنها 11. 9٪ به‌عنوان متوسط و 6٪ طولانی طبقه‌بندی شدند.

هدف از این آزمایش پیش‌بینی کلاس مشاغل با استفاده از اطلاعات ارائه شده در جدول است [17. 1](#bookmark799) . بدیهی است که انواع خطاها یکسان نیستند و باید جریمه بیشتری برای طبقه‌بندی مشاغل به‌عنوان مشاغل بسیار کوتاه و در واقع طولانی وجود داشته باشد. همچنین، از آنجایی که معادله پیش‌بینی باید در نرم‌افزار پیاده‌سازی شود و به سرعت محاسبه شود، مدل‌هایی با معادلات پیش‌بینی ساده‌تر ترجیح داده می‌شوند.

در طی دو سال، تغییراتی در برخی از سخت افزارهای محیط HPC ایجاد شد. در نتیجه، یک کار اجرا شده بر روی دو نسخه مختلف سخت افزار ممکن است زمان اجرا متفاوتی را ایجاد کند. این اثر باعث افزایش تغییرات ذاتی در زمان‌های اجرا می‌شود و می‌تواند منجر به برچسب‌گذاری نادرست کلاس‌های مشاهده‌شده شود. در حالی که هیچ راهی برای رسیدگی به این موضوع در تحلیل وجود ندارد، هر مدلی برای طبقه‌بندی زمان‌های اجرا باید در طول زمان مورد بازبینی قرار گیرد (با این فرض که سخت افزار جدید زمان اجرا را کاهش می‌دهد).

قبل از پرداختن به مدل‌سازی، بررسی داده‌ها مهم است. قبل از مدل‌سازی، می‌توان انتظار داشت که محرک‌های اصلی زمان اجرا، تعداد ترکیب‌ها، تعداد وظایف و اینکه کدام پروتکل در حال اجرا است باشد. به‌عنوان یک بررسی اولیه، شکل.  [17. 2](#bookmark802) رابطه بین پروتکل‌ها و کلاس‌های زمان اجرا را با استفاده از *نمودار موزاییکی نشان می‌*دهد. در اینجا پهنای کادرها نشان دهنده تعداد کارهای اجرا شده برای پروتکل است (مثلاً پروتکل J بیشترین و K کمترین اجرا شده است). برای پیش‌بینی کارهای کند، می‌توان دید که تنها چند پروتکل زمان اجرای طولانی را تولید می‌کنند. سایر موارد، مانند پروتکل D، به احتمال زیاد بسیار سریع اجرا می‌شوند. با توجه به این روابط، ممکن است انتظار داشته باشیم که اطلاعات پروتکل ­به‌طور بالقوه برای مدل مهم باشد.

شکل 17. 3 تجسم دیگری از داده‌ها را با استفاده از *نمودار جدول نشان می‌*دهد. در اینجا، داده‌ها توسط کلاس مرتب می‌شوند و سپس در 100 برش قرار می‌گیرند. در هر برش، مقدار متوسط پیش‌بینی‌کننده‌های عددی تعیین می‌شود. به‌طور مشابه، توزیع فراوانی پیش‌بینی‌کننده‌های طبقه‌بندی شده تعیین می‌شود. نتایج برای هر پیش‌بینی در ستون‌هایی نشان داده می‌شود تا مدل‌ساز بتواند درک بهتری از ارتباط هر پیش‌بینی با نتیجه داشته باشد. در شکل می‌بینیم که:

کارهای مرتبط با تعداد زیادی از ترکیبات معمولاً در زمان اجرا بزرگ یا متوسط هستند.

بسیاری از مشاغل با طول متوسط زمانی ارائه شدند که تعداد مشاغل معلق بسیار زیاد بود. با این حال، این روند خود را در مشاغل بسیار طولانی بازتولید نمی‌کند. به همین دلیل، مهم خواهد بود

جدول 17. 1: پیش‌بینی‌کننده‌ها برای داده‌های زمانبندی کار   
7 متغیر 4331 مشاهده

پروتکل

n از دست رفته منحصر به فرد

4331 0 14

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | آ | سی | دی | E | اف | جی | اچ | من | JK | ال | م | ن | O |
| فرکانس | 94 | 160 | 149 | 96 | 170 | 155 | 321 | 381 | 989 6 | 242 | 451 | 536 | 581 |
| % | 2 | 4 | 3 | 2 | 4 | 4 | 7 | 9 | 230 | 6 | 10 | 12 | 13 |

ترکیبات lii.

n گمشده منحصر به فرد میانگین 0. 05 0. 10 0. 25 0. 50 0. 75 0. 90 0. 95

4331 0 858 497. 7 27 37 98 226 448 967 2512

کمترین: 20 21 22 23 24

بالاترین: 14087 14090 14091 14097 14103

فیلدهای ورودی I

n گمشده منحصر به فرد میانگین 0. 05 0. 10 0. 25 0. 50 0. 75 0. 90 0. 95 4331 0 1730 1537 26 48 134 426 991 4165 7594

کمترین: 10 11 12 13 14

بالاترین: 36021 45420 45628 55920 56671

تکرارها

n گمشده منحصر به فرد میانگین 0. 05 0. 10 0. 25 0. 50 0. 75 0. 90 0. 95

4331 0 11 29. 24 10 20 20 20 20 50 100

10 11 15 20 30 40 50 100 125 150 200

فرکانس 272 9 2 3568 3 7 153 188 1 2 126

% 600820044003

NumPending I

n گمشده منحصر به فرد میانگین 0. 05 0. 10 0. 25 0. 50 0. 75 0. 90 0. 95 4331 0 303 53. 39 0. 0 0. 0 0. 0 0. 0 0. 0 33. 0 145. 5

کمترین : 0 1 2 3 4، بالاترین: 3822 3870 3878 5547 5605

ساعت . . . . لیلللللللللللللللللللللللل .

n گمشده منحصر به فرد میانگین 0. 05 0. 10 0. 25 0. 50 0. 75 0. 90 0. 95

4331 0 924 13. 73 7. 025 9. 333 10. 900 14. 017 16. 600 18. 250 19. 658

کمترین : 0. 01667 0. 03333 0. 08333 0. 10000 0. 11667 بالاترین: 23. 20000 23. 21667 23. 35000 23. 80000 23. 98333

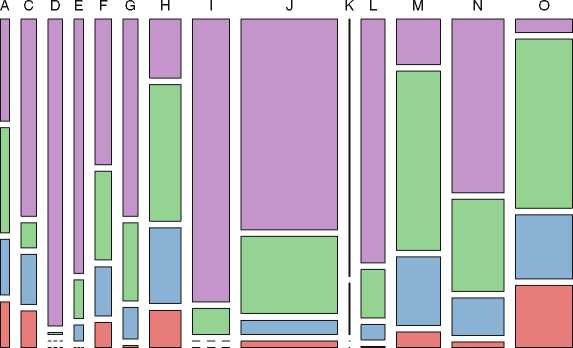
روز

n از دست رفته منحصر به فرد 4331 0 7

دوشنبه سه شنبه چهارشنبه پنج شنبه جمعه شنبه یکشنبه

فرکانس 692 900 903 720 923 32 161

% 1621211721 1 4



شکل 17. 2: تعداد زیادی موزاییک از فرکانس‌های کلاس برای هر پروتکل. عرض جعبه‌ها بر اساس تعداد کارهای مشاهده شده در مجموعه داده‌ها تعیین می‌شود

آینده نگر این مشاهدات را تأیید کنید تا مطمئن شوید که تصادفی از این مجموعه داده خاص نیست.

• هنگامی که تعداد تکرارها زیاد باشد، کار تمایل به طولانی شدن دارد.

یکی از کاستی‌های این تجسم خاص این است که ­روابط بین پیش‌بینی‌کننده‌ها را مبهم می‌کند. نمودارهای همبستگی و ماتریس‌های نمودار پراکندگی روش‌های موثری برای یافتن این نوع روابط هستند.

علاوه بر این، شکل [17. 4](#bookmark803) نمودارهای پراکندگی را برای تعداد ترکیبات در مقابل تعداد فیلدهای ورودی بر اساس پروتکل نشان می‌دهد. در این طرح‌ها، مشاغل بر اساس کلاس رنگ آمیزی می‌شوند. برای برخی موارد، مانند پروتکل‌های A، C، D، H، I و K، به نظر می‌رسد که تعداد ترکیبات و فیلدها اطلاعاتی را برای تمایز ­کلاس‌ها فراهم می‌کنند. با این حال، این روابط خاص کلاس هستند. الگوهای پروتکل‌های I و K متفاوت است. یکی دیگر از جنبه‌های جالب این تحلیل این است که الگوی همبستگی بین تعداد ترکیبات در مقابل تعداد فیلدهای ورودی، پروتکل خاص است. برای مثال، همبستگی بسیار کمی بین دو پیش‌بینی برای برخی از پروتکل‌ها (مانند J، O ) و همبستگی قوی در برخی دیگر (مانند D، E و H ) وجود دارد. این الگوها ممکن است مهم باشند و زمانی که مدل ساز به نقاط داده واقعی نگاه می‌کند، کشف می‌شوند.

17. 1 تقسیم داده‌ها و استراتژی مدل

4331 نمونه موجود است. 80٪ برای آموزش الگوریتم‌ها استفاده می‌شود در حالی که مابقی برای ارزیابی مدل‌های نامزد نهایی استفاده می‌شود. داده‌ها با استفاده از نمونه‌گیری تصادفی طبقه‌ای برای حفظ کلاس تقسیم شدند

پروتکل کلاس

log (ترکیبات) ورودی فیلدها گزارش (تغییرات) گزارش (تعداد در انتظار) ساعت روز

دوشنبه

سه شنبه

چهارشنبه

پنج شنبه

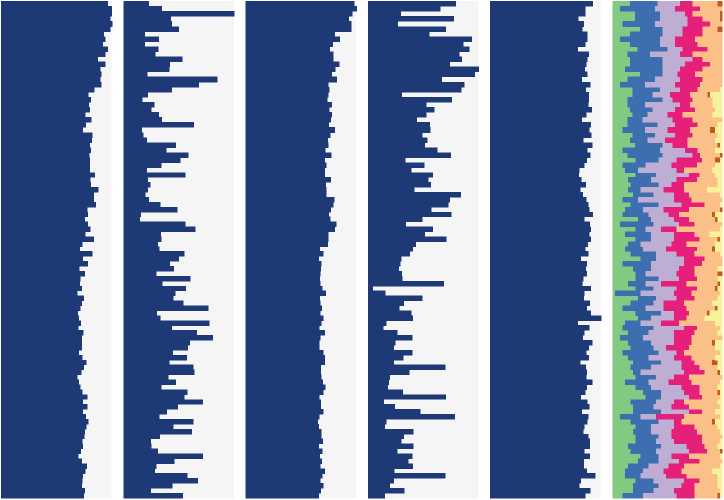
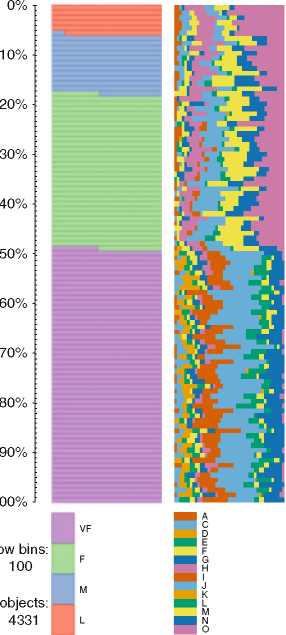
جمعه

نشست

آفتاب

شکل 17. 3: نمودار جدولی از داده ها

17. 1 تقسیم داده‌ها و استراتژی مدل 451



VF • F Ma ال

10A4.0

10A3.5

10A3.0

10A2.5

10A2.0

10A1.5

10A4.0

10A3.5

10A3.0

10A2.5

10A2.0

10A1.5

| J 1 1 1  A | J 1 1 1  C | J 1 1 1  D | J 1 1 1  E |
| --- | --- | --- | --- |
| **• 7 •** | **! • ••** | / | Ri a  7 |
| F | G | H | I |
| **SwJ 1 ♦ ♦ ♦** | **>\***  • •< | jjjr |  |
| j | K | L | M |
| **♦4H** | •  • | • |  |
| N | O | - 10A4.0  - 10A3.5  - 10A3.0  - 10A2.5  - 10A2.0  - 10A1.5 | |
| **\* \*\*\*\*>?•**  **Illi** | **1**  **Illi** |

10A1 10A3

10A1 10A3

10A1 10A3

Input Fields

10A4.0

10A3.5

10A3.0

10A2.5

10A2.0

10A1.5

شکل 17. 4: نمودار پراکندگی تعداد ترکیبات در مقابل تعداد فیلدها برای هر پروتکل. نقاط با توزیع کلاس زمان اجرا آنها از نتیجه رنگ می‌شوند. پنج تکرار از اعتبارسنجی متقاطع 10 برابری برای تنظیم مدل‌ها استفاده شد.

به جای ایجاد مدل‌هایی که دقت کلی یا آمار کاپا را به حداکثر می‌رسانند، از یک تابع هزینه سفارشی برای دادن وزن بیشتر به خطاهایی استفاده می‌شود که در آن مشاغل طولانی و متوسط به اشتباه به‌عنوان سریع یا بسیار سریع طبقه‌بندی می‌شوند. جدول [17. 2](#bookmark804) نشان می‌دهد که چگونه هزینه‌های مرتبط با هر نوع خطا بر معیار کلی عملکرد تأثیر می‌گذارد. هزینه بسیار سنگین است به‌طوری که مشاغل طولانی ­به صف‌ها (یا سخت‌افزار) که برای کارهای کوچک و سریع طراحی شده‌اند، ارسال نمی‌شوند. مشاغل با طول متوسط ­نیز به دلیل طبقه‌بندی اشتباه به‌عنوان مشاغل کارآمدتر جریمه می‌شوند.

جدول 17. 2: ساختار هزینه مورد استفاده برای بهینه‌سازی مدل

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | کلاس مشاهده شده | | | |
| VF | اف | م | ال |
| VF | 0 | 1 | 5 | 10 |
| اف | 1 | 0 | 5 | 5 |
| م | 1 | 1 | 0 | 1 |
| ال | 1 | 1 | 1 | 0 |

مشاغل طولانی‌تر که به اشتباه به‌عنوان سریع طبقه‌بندی می‌شوند، در این معیار جریمه بیشتری می‌شوند

بهترین رویکرد ممکن برای آموزش مدل زمانی است که از این تابع هزینه برای تخمین پارامترهای مدل استفاده شود (به بخش [16. 8 )](#bookmark771) . چند مدل مبتنی بر درخت این امکان را می‌دهند، اما بیشتر مدل‌هایی که در اینجا در نظر گرفته شده‌اند، این امکان را ندارند.

یک سری مدل برای مجموعه آموزشی مناسب بود. ترکیب پارامتر تنظیم ­مرتبط با کوچکترین مقدار هزینه متوسط برای مدل نهایی انتخاب شد و همراه با کل مجموعه آموزشی استفاده شد. مدل‌های زیر مورد بررسی قرار گرفتند:

تحلیل تفکیک خطی: این مدل با استفاده از مجموعه معادلات استاندارد و همچنین با نسخه جریمه شده که ­انتخاب ویژگی را در طول آموزش مدل انجام می‌دهد ایجاد شده است. اندازه‌های زیرمجموعه پیش‌بینی از 2 تا 112 در ارتباط با چندین مقدار جریمه پشته بررسی شدند: 0، 0. 01، 0. 1، 1 و 10.

تحلیل تفکیک حداقل مربعات جزئی: مدل PLS با تعداد مؤلفه‌ها از 1 تا 91 برازش داشت.

شبکه‌های عصبی: مدل‌ها با واحدهای پنهان از 1 تا 19 و 5 ارزش فروپاشی وزنی برازش داشتند: 0، 0. 001، 0. 01، 0. 1 و 0. 5.

تحلیل تفکیک انعطاف پذیر: از توابع لولای درجه یک MARS استفاده شد و تعداد عبارت‌های حفظ شده از 2 تا 23 متغیر بود.

ماشین‌های بردار پشتیبان (SVM): دو مدل مختلف با تابع پایه شعاعی مطابقت داشتند. یکی با استفاده از وزن‌های مساوی در هر کلاس و دیگری که در آن به مشاغل متوسط وزن پنج برابری داده شد و مشاغل طولانی ­ده برابر وزن داده شدند. در هر مورد، محاسبات تحلیلی برای تخمین تابع هسته RBF همراه با مقادیر هزینه از 2 *تا* 2 استفاده شد. به 2 12 در مقیاس ورود به سامانه‌.

درختان تک سبد خرید: به‌طور مشابه، مدل‌های CART با هزینه‌های مساوی برای هر کلاس و مدل دیگر که در آن هزینه‌ها شبیه به هزینه‌های موجود در جدول بود، مناسب بودند.  [17. 2 .](#bookmark804) در هر مورد، مدل بیش از 20 مقدار از پارامتر پیچیدگی تنظیم شد.

درختان سبد خرید کیسه ای: این مدل‌ها از 50 درخت سبد خرید کیسه‌ای با و بدون ترکیب ساختار هزینه استفاده می‌کردند.

جنگل‌های تصادفی: مدل از 2000 درخت در جنگل استفاده کرد و روی 6 مقدار پارامتر تنظیم تنظیم شد.

C5. 0 : این مدل با مدل‌های مبتنی بر درخت و قانون، با و بدون winnowing، با مدل‌های تکی و با حداکثر 100 تکرار تقویت ارزیابی شد. یک نسخه جایگزین از این مدل مناسب بود که از ساختار هزینه نشان داده شده در جدول استفاده کرد [17. 2 .](#bookmark804)

برای این کاربرد، بایاس نسبت به مدل‌هایی وجود دارد که برای پیش‌بینی‌کننده‌های سریع مفید هستند. اگر معادله(های) پیش‌بینی مدل را بتوان به راحتی در نرم‌افزار (مانند یک درخت ساده یا شبکه عصبی) کدگذاری کرد یا از یک رابط خط فرمان ساده (مانند C5. 0 ) اجرا کرد، مدل بر ترجیح داده می‌شود. دیگران.

17. 2 نتایج

نتایج مدل در شکل نشان داده شده است.  [17. 5](#bookmark807) که در آن نمودارهای جعبه‌ای از برآوردهای نمونه‌گیری مجدد از میانگین ارزش هزینه نشان داده شده است. مدل‌های خطی، مانند LDA و PLS، در اینجا خوب عمل نکردند. انتخاب ویژگی به مدل تفکیک خطی کمکی نکرد، اما این ممکن است به دلیل ناتوانی آن مدل در ­تعیین مرزهای کلاس غیرخطی باشد. FDA همچنین از نظر هزینه عملکرد ضعیفی را نشان داد.

مجموعه‌ای از مدل‌ها با هزینه‌های متوسط وجود دارد که احتمالاً معادل هستند، عمدتاً SVM و روش‌های مختلف مجموعه درختی. استفاده از هزینه/وزن اثرات مثبت قابل‌توجهی بر روی درخت سبد خرید و SVMها داشت. شکل [17. 6](#bookmark808) نمایه‌های نمونه‌گیری مجدد را برای این دو مدل نشان می‌دهد

PLS

FDA

LDA

LDA (Sparse) سبد خرید

شبکه‌های عصبی

SVM

بسته‌بندی (هزینه ها)

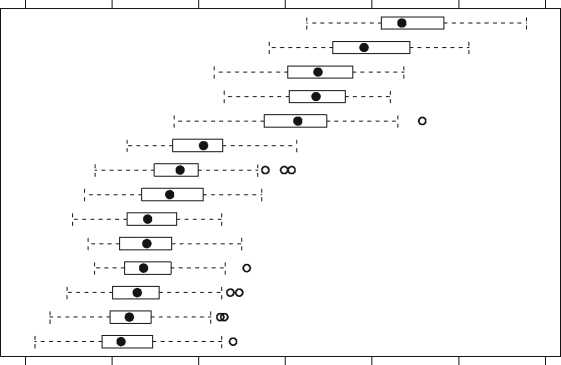
سبد خرید (هزینه ها)

C5. 0 (هزینه ها)

SVM (وزن ها)

کوله بری

جنگل‌های تصادفی C5. 0

شکل 17. 5: میانگین هزینه نمونه برداری مجدد پروفایل برای مدل‌های مختلف

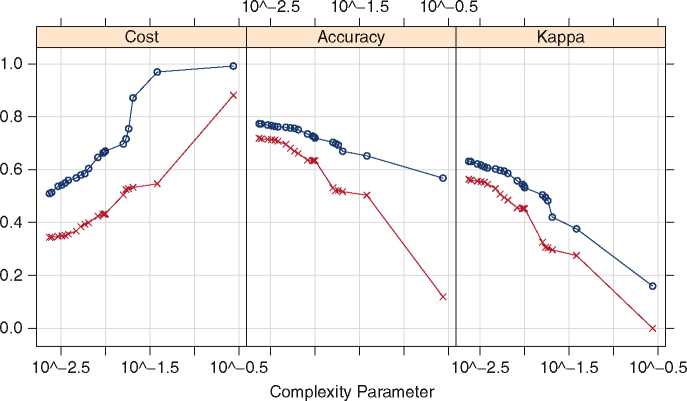
0.2

0.3

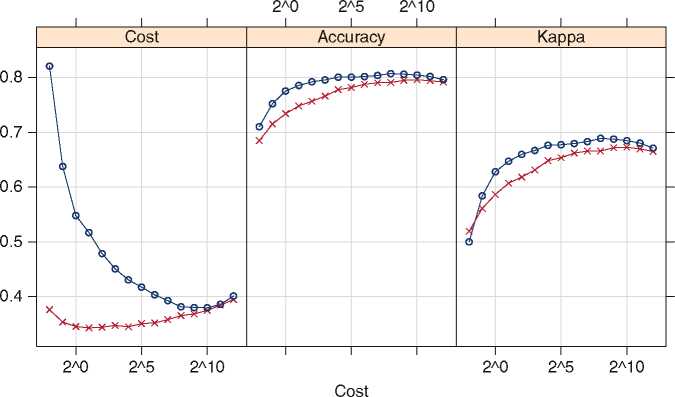
0.4 0.5 0.6 0.7 0.8

Cost

هزینه‌های برابر o هزینه‌های نابرابر x



وزن‌های مساوی o وزن‌های نابرابر x



شکل 17. 6: تأثیر ترکیب ساختار هزینه در فرآیند آموزش CART ( *بالا* ) و وزنه‌ها برای ماشین‌های بردار پشتیبان ( *پایین* )

شرایط برآورد آنها از هزینه، دقت کلی و آمار کاپا. نتایج مدل CART نشان می‌دهد که استفاده از هزینه تأثیر منفی بر دقت و کاپا دارد اما به‌طور طبیعی برآورد هزینه را بهبود می‌بخشد. مهم نیست معیار، فرآیند تنظیم همان مدل CART را انتخاب می‌کرد

جدول 17. 3: ماتریس‌های سردرگمی نمونه برداری مجدد برای جنگل تصادفی و ­مدل‌های CART حساس به هزینه

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | سبد خرید حساس به هزینه | | | | جنگل‌های تصادفی | | | |
| VF | اف | م | ال | VF | اف | م | ال |
| VF | 157. 0 | 24. 6 | 2. 0 | 0. 2 | 164. 7 | 18. 0 | 1. 3 | 0. 2 |
| اف | 10. 3 | 43. 2 | 3. 1 | 0. 2 | 11. 9 | 83. 7 | 11. 5 | 1. 8 |
| م | 9. 6 | 38. 3 | 34. 5 | 5. 8 | 0. 2 | 5. 5 | 27. 4 | 1. 8 |
| ال | 0. 0 | 1. 7 | 1. 6 | 14. 6 | 0. 0 | 0. 6 | 0. 9 | 17. 0 |

هر مقدار میانگین تعداد کارهایی است که در آن سلول در 50 مجموعه داده نگهدارنده رخ داده است. ستون‌ها کلاس‌های شغلی واقعی برای آموزش نهایی هستند. نتایج ماشین بردار پشتیبانی (SVM) تا حدودی متفاوت است. استفاده از وزن‌های کلاس نیز تأثیر منفی جزئی بر دقت و کاپا داشت، اما بهبود در هزینه برآورد شده قابل‌توجه بود. همچنین، مدل بدون وزن با پارامتر هزینه SVM بسیار بالاتر (و در نتیجه مدل‌های پیچیده تر) نسبت به نسخه وزن دار بهینه می‌شود.

به‌طور عجیبی، مدل C5. 0 بدون هزینه بسیار خوب عمل کرد، اما افزودن ساختار هزینه به فرآیند درخت سازی، هزینه تخمینی مدل را *افزایش داد.* همچنین، کیسه‌بندی درختان CART با هزینه‌ها تأثیر منفی کمی بر مدل داشت

درختان به وضوح برای این داده‌ها خوب عمل کردند، همانطور که ماشین‌های برداری و شبکه‌های عصبی را پشتیبانی کردند. آیا بین مدل‌های برتر تفاوت زیادی وجود دارد؟ یک رویکرد برای بررسی این نتایج، نگاه کردن به یک ماتریس سردرگمی است که در سراسر نمونه‌های مجدد ایجاد شده است. به یاد بیاورید که برای نمونه برداری مجدد، 50 مجموعه نگهدارنده وجود داشت که به‌طور متوسط شامل حدود 347 شغل بود. برای هر یک از این مجموعه‌های نگهدارنده، یک ماتریس سردرگمی محاسبه شد و میانگین ماتریس اختلاط ­با میانگین فرکانس‌های سلولی محاسبه شد.

جدول [17. 3](#bookmark809) دو جدول از این قبیل را برای مدل جنگل‌های تصادفی و مدل سبد خرید حساس به هزینه نشان می‌دهد. برای جنگل تصادفی، میانگین تعداد مشاغل طولانی که به اشتباه به‌عنوان خیلی سریع طبقه‌بندی شدند 0. 2 بود در حالی که همین مقدار برای درخت طبقه‌بندی 0. 24 بود. درخت CART برای کارهای سریع در مقایسه با مدل جنگل تصادفی دقت بسیار ضعیفی را نشان می‌دهد. با این حال، برعکس برای مشاغل نسبتا طولانی صادق است. جنگل تصادفی 72. 56% از آن مشاغل را (به‌طور متوسط) به اشتباه طبقه‌بندی کرد، در حالی که 65. 52% برای تک درخت بود. برای کارهای طولانی، تک درخت دارای نرخ خطای بالاتری نسبت به روش گروهی است. چگونه این دو مدل با استفاده از مجموعه تست مقایسه می‌شوند؟ هزینه مجموعه آزمون برای جنگل تصادفی 0. 316 بود در حالی که درختان طبقه‌بندی تک دارای هزینه متوسط 0. 37 بود. جدول [17. 4](#bookmark812) ماتریس‌های سردرگمی را برای دو مدل نشان می‌دهد. روندها در مجموعه آزمایشی بسیار شبیه به برآوردهای نمونه‌گیری مجدد است. تک درخت با کارهای سریع و طولانی بدتر عمل می‌کند در حالی که جنگل‌های تصادفی مسائلی در پیش‌بینی متوسط دارند.

جدول 17. 4: مجموعه آزمایشی ماتریس‌های سردرگمی برای جنگل تصادفی و ­مدل‌های CART حساس به هزینه

سبد خرید حساس به هزینه جنگل‌های تصادفی

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | VF | اف | م | ال | VF | اف | م | ال |
| VF | 383 | 61 | 5 | 1 | 414 | 45 | 3 | 0 |
| اف | 32 | 106 | 7 | 2 | 28 | 206 | 27 | 5 |
| م | 26 | 99 | 87 | 15 | 0 | 18 | 71 | 6 |
| ال | 1 | 3 | 3 | 33 | 0 | 0 | 1 | 40 |

ستون‌ها مشاغل طولانی طبقات شغلی واقعی هستند. به‌طور خلاصه، تفاوت‌های کلی بین این دو مدل زیاد نیست.

17. 3 محاسبات

داده‌ها در بسته AppliedPredictiveModeling موجود است. پس از بارگذاری داده‌ها، یک مجموعه آموزشی و آزمایشی ایجاد شد:

*> کتابخانه (AppliedPredictiveModeling)*

*> داده (HPC)*

*> set. seed(1104)*

*inTrain <- createDataPartition(schedulingData$Class,*

*+ p = 0. 8،*

*+ لیست = نادرست)*

*schedulingData$NumPending <- schedulingData$NumPending + 1*

*trainData <- schedulingData[ inTrain,]*

*testData <- schedulingData[-inTrain,]*

از آنجایی که هزینه‌های تعریف شده در جدول [17. 2](#bookmark804) برای قضاوت در مورد مدل‌ها استفاده خواهد شد، توابع برای تخمین این مقدار از مجموعه‌ای از کلاس‌های مشاهده شده و پیش‌بینی شده نوشته شده است:

*هزینه <- تابع (pred، obs)*

*+{*

*+ isNA <- is. na(pred)*

*+ if(!all(isNA))*

*+{*

*+ pred <- pred[!isNA]*

*+ obs <- obs[!isNA]*

*+ هزینه <- ifelse(pred == obs, 0, 1)*

*+ if(any(pred == "VF" & obs == "L"))*

*+ هزینه[pred == "L" & obs == "VF"] <- 10*

*+ if(any(pred == "F" & obs == "L"))*

*+ هزینه[pred == "F" & obs == "L"] <- 5*

*+ if(any(pred == "F" & obs == "M"))*

*+ هزینه[pred == "F" & obs == "M"] <- 5*

*+ if(any(pred == "VF" & obs == "M"))*

*+ هزینه[pred == "VF" & obs == "M"] <- 5*

*+ خارج <- میانگین (هزینه)*

*+ } other out <- NA*

*+ بیرون*

*+}*

*costSummary <- تابع (داده، lev = NULL، مدل = NULL) +{*

*+ if (is. character(data$obs)) data$obs <- factor(data$obs,*

*سطوح = lev)*

*+ c(postResample(data[، "pred"]، data[، "obs"])،*

*+ هزینه = هزینه (داده[، "پیش از پیش"]، داده[، "obs"]))*

*+}*

تابع دوم در شیء کنترلی برای محاسبات آینده استفاده می‌شود:

*ctrl <- trainControl(method = "repeatedcv"، repeats = 5،*

*+ summaryfunction = هزینه خلاصه)*

برای مدل‌های درختی حساس به هزینه، یک نمایش ماتریسی از هزینه‌ها نیز ایجاد شد:

*costMatrix <- ifelse(diag(4) == 1, 0, 1)*

*costMatrix[1,4] <- 10*

*costMatrix[1,3] <- 5*

*costMatrix[2,4] <- 5*

*costMatrix[2,3] <- 5*

*rownames(costMatrix) <- level (trainData$Class)*

*colnames(costMatrix) <- level(trainData$Class) > costMatrix*

روش‌های مبتنی بر درخت از دسته‌های مستقل استفاده نمی‌کنند، اما مدل‌های دیگر نیاز دارند که پیش‌بینی‌کننده‌های طبقه‌بندی (مثلاً پروتکل) به متغیرهای ساختگی تجزیه شوند. یک فرمول مدل ایجاد شد که log چندین پیش‌بینی را تبدیل می‌کند (با توجه به چولگی نشان داده شده در جدول [17. 1 )](#bookmark799) :

*modForm <- as. formula(Class ~ Protocol + log10(Compounds) +*

*+ log10 (فیلدهای ورودی) + log10 (تکرارها) +*

*+ log10 (NumPending) + ساعت + روز)*

ویژگی‌های مدل‌های متناسب با داده‌ها را می‌توان در فهرست فصل بسته AppliedPredictiveModeling یافت و از مالیات همگام‌سازی مشابه ­با کد نشان‌داده‌شده در فصل‌های قبل پیروی کرد. با این حال، فراخوانی تابع مدل حساس به هزینه و وزنی هستند

*## سبد خرید حساس به هزینه*

*set. seed(857)*

*rpFitCost <- train(x = trainData[، پیش‌بینی ها]،*

*+ y = trainData$Class،*

*+ روش = "rpart"،*

*+ معیار = "هزینه"،*

*+ حداکثر کردن = نادرست،*

|  |  |
| --- | --- |
| *+* | *طول کوک = 20،* |
| *+* | *## rpart ماتریس هزینه را طوری ساختار می‌دهد که* |
| *+* | *## کلاس‌های واقعی در ردیف هستند، بنابراین ما* |
| *+* | *## ماتریس هزینه را جابجا کنید* |
| *+* | *parms =list(loss = t(costMatrix))،* |
| *+* | *trControl = ctrl)* |

*> ## C5. 0 حساس به هزینه*

*> set. seed(857)*

*> c50Cost <- train(x = trainData[، پیش‌بینی ها]*

|  |  |
| --- | --- |
| *+*  *+* | *y = trainData$Class، متد = "C5. 0"،* |
| *+*  *+* | *معیار = "هزینه"، حداکثر کردن = نادرست،* |
| *+*  *+* | *هزینه‌ها = costMatrix، tuneGrid = expand. grid(. trials = c(1, (1:10)\*10)* |
| *+*  *+* | *. model = "درخت"،*  *. winnow = c(درست، نادرست))* |
| *+* | *trControl = ctrl)* |

*> ## درختان بسته‌بندی شده حساس به هزینه*

*> rpCost <- تابع (x, y)*

*+{*

*+ costMatrix <- ifelse(diag(4) == 1, 0, 1)*

*+ costMatrix[4, 1] <- 10*

*+ costMatrix[3, 1] <- 5*

*+ costMatrix[4, 2] <- 5*

*+ costMatrix[3, 2] <- 5*

*+ کتابخانه (rpart)*

*+ tmp <- x*

*+ tmp$y <- y*

*+ rpart(y~. ,*

*+ داده = tmp،*

*+ control = rpart. control (cp = 0)،*

*+ parms = لیست (از دست دادن = هزینه ماتریس))*

*+}*

*> rpPredict <- تابع(شیء، x) پیش‌بینی(شیء، x)*

*> rpAgg <- تابع (x، نوع = "کلاس")*

*+{*

*+ ادغام شده <- x[[1]] \* NA*

*+ n <- nrow (تلفیقی)*

*+ کلاس‌ها <- colnames (تلفیقی)*

*+ برای (i در 1:ncol(pooled))*

*+{*

*+ tmp <- lapply(x، تابع (y، col) y[، col]، col = i)*

*+ tmp <- do. call("rbind"، tmp)*

*+ ادغام شده[، i] <- اعمال (tmp، 2، میانه)*

*+}*

*+ ادغام شده <- اعمال (ترکیبی، 1، تابع (x) x/جمع (x))*

*+ if (n != nrow(pooled)) pooled <- t(pooled)*

*+ out <- factor(classes[apply(pooled, 1, which. max)],*

*+ سطوح = کلاس ها)*

*+ بیرون*

*+}*

*> set. seed(857)*

*> rpCostBag <- train(trainData[, predictors],*

|  |  |
| --- | --- |
| *+*  *+*  *+*  *+*  *+*  *+*  *+*  *+* | *trainData$Class، "bag"،*  *B = 50،*  *bagControl = bagControl (مناسب = rpCost، predict = rpPredict، aggregate = rpAgg، downSample = FALSE)*  *trControl = ctrl)* |

*>*

|  |
| --- |
| *> ## SVM وزنی* |
| *> set. seed(857)* |
| *> svmRFitCost <- train(modForm, data = trainData,* |
| *+ روش = "svmRadial"،* |
| *+ معیار = "هزینه"،* |
| *+ حداکثر کردن = نادرست،* |
| *+ preProc = c ("مرکز"، "مقیاس")،* |
| *+ class. weights = c(VF = 1,F=1,* |
| *+ M = 5، L = 10)* |
| *+ آهنگ طول = 15،* |
| *+ trControl = ctrl)* |

نسخه‌های نمونه‌گیری مجدد از ماتریس‌های سردرگمی با استفاده از تابع confusionMatrix بر روی اشیاء تولید شده توسط تابع آموزش، مانند

*> confusionMatrix (rpFitCost، norm = "هیچ")*

اعتبار متقاطع (10 برابر، 5 بار تکرار) ماتریس سردرگمی

(تعداد ورودی‌ها غیر عادی هستند)

ارجاع

پیش‌بینی VF FML

VF 157. 0 24. 6 2. 0 0. 2

F 10. 3 43. 2 3. 1 0. 2

M 9. 6 38. 3 34. 5 5. 8

L 0. 0 1. 7 1. 6 14. 6

هنجار تعیین می‌کند که چگونه شمارش خام از هر نمونه مجدد باید نرمال شود. مقدار "none" به تعداد میانگین در هر سلول جدول منجر می‌شود. استفاده از norm = "کلی" ابتدا ورودی‌های سلول را بر تعداد کل نقاط داده در جدول تقسیم می‌کند، سپس این درصدها را میانگین می‌کند.

قسمت چهارم

ملاحظات دیگر

فصل 18

اندازه‌گیری اهمیت پیش‌بینی

اغلب، ما تمایل داریم که قدرت رابطه بین پیش‌بینی‌کننده‌ها و نتیجه را کمی کنیم. از آنجایی که تعداد ویژگی‌ها زیاد می‌شود، تحلیل اکتشافی همه پیش‌بینی‌کننده‌ها ممکن است غیرممکن باشد و تمرکز ­بر آنهایی که روابط قوی با نتیجه دارند، ممکن است یک استراتژی تریاژ مؤثر باشد. پیش‌بینی‌کننده‌های رتبه‌بندی به این روش می‌توانند هنگام غربال کردن مقادیر زیادی از داده‌ها بسیار مفید باشند.

یکی از دلایل اصلی برای اندازه‌گیری قدرت یا ارتباط پیش‌بینی‌ها، فیلتری است که باید به‌عنوان ورودی در یک مدل استفاده شود. این *انتخاب ویژگی فوق‌العاده ­*می‌تواند مبتنی بر داده‌های موجود باشد. نتایج ­فرآیند فیلتر کردن، همراه با تخصص موضوع، می‌تواند گامی حیاتی در ایجاد یک مدل پیش‌بینی موثر باشد. همانطور که در فصل بعدی مشاهده خواهد شد، بسیاری از الگوریتم‌های انتخاب ویژگی برای فیلتر کردن بر یک امتیاز ارتباط کمی تکیه می‌کنند.

بسیاری از مدل‌های پیش‌بینی دارای اندازه گیری‌های داخلی یا *ذاتی* با اهمیت پیش‌بینی هستند و در فصل‌های قبلی مورد بحث قرار گرفته اند. برای ­مثال، MARS و بسیاری از مدل‌های مبتنی بر درخت، افزایش عملکرد را که هنگام افزودن هر پیش‌بینی به مدل رخ می‌دهد، نظارت می‌کنند. برخی دیگر مانند رگرسیون خطی یا رگرسیون لجستیک می‌توانند از کمی‌سازی بر اساس ضرایب مدل یا معیارهای آماری (مانند آماره *t* ) استفاده کنند. روش‌های مورد بحث در این فصل مختص هیچ مدل پیش‌بینی نیستند. اگر یک ­مدل کارآمد ایجاد شده باشد، امتیازهای به دست آمده از آن مدل احتمالاً قابل اعتمادتر از روش‌های مورد بحث در این فصل هستند، زیرا آنها مستقیماً با مدل مرتبط هستند.

در این فصل، مفهوم اهمیت متغیر به معنای کمی‌سازی کلی رابطه بین پیش‌بینی و خروجی در نظر گرفته شده ­است. بیشتر روش‌های مورد بحث نمی‌توانند مدل‌ساز را از *ماهیت* رابطه آگاه کنند، مانند "افزایش پیش‌بینی منجر به کاهش نتیجه می‌شود. " چنین خصوصیات دقیق فقط می‌تواند ­از مدل‌هایی ناشی شود که فرم‌های پارامعیار دقیقی مانند رگرسیون خطی یا لجستیک، خطوط رگرسیون تطبیقی چند متغیره و چند مدل دیگر دارند. به‌عنوان مثال، فرقه.  [8. 5](#bookmark419) امتیازهای اهمیت متغیر را برای جنگل تصادفی مورد بحث قرار داد. که در

\_18،

© ، این اندازه‌گیری ارتباط پیش‌بینی با جابجایی هر پیش‌بینی به‌صورت جداگانه و ارزیابی کاهش عملکرد زمانی که اثر پیش‌بینی نفی می‌شود، به دست می‌آید. در این مورد، کاهش قابل‌توجهی در عملکرد نشان دهنده یک پیش‌بینی مهم است. در حالی که این می‌تواند یک رویکرد موثر باشد، اما مدل ساز را در مورد شکل دقیق رابطه روشن نمی‌کند. با وجود این، این اندازه‌گیری‌ها می‌توانند برای راهنمایی کاربر برای تمرکز بیشتر روی پیش‌بینی‌کننده‌های خاص از طریق تجسم‌ها و ابزارهای دیگر مفید باشند.

بسیاری از امتیازهای اهمیت متغیر به نوع داده‌ها اختصاص دارند. برای ­مثال، تکنیک‌های پیامدهای عددی ممکن است برای نتایج طبقه‌بندی مناسب نباشند. این فصل بر اساس ماهیت نتیجه تقسیم شده است. بخش ­\_ [18. 1](#bookmark825) تکنیک‌هایی را برای نتایج عددی و بخش بحث می‌کند.  [18. 2](#bookmark831) بر نتایج طبقه‌بندی شده تمرکز دارد. در نهایت، بخش.  [18. 3](#bookmark837) رویکردهای کلی‌تری را برای مسأله مورد بحث قرار داد. چندین مثال از فصل‌های قبلی برای نشان دادن تکنیک‌ها استفاده شده است.

18. 1 نتایج عددی

برای پیش‌بینی‌کننده‌های عددی، رویکرد کلاسیک برای کمی کردن هر رابطه با نتیجه، از آماره همبستگی نمونه استفاده می‌کند. این کمیت ­ارتباط *خطی را اندازه‌گیری می‌کند.* اگر رابطه تقریباً خطی یا منحنی باشد، ضریب همبستگی اسپیرمن (بخش 5. 1 ) ممکن است مؤثرتر باشد. این معیارها باید برآوردهای تقریبی از رابطه در نظر گرفته شوند و ممکن است برای روابط پیچیده تر مؤثر نباشند.

به‌عنوان مثال، داده‌های QSAR مورد استفاده در فصل‌های قبلی رگرسیون حاوی تعدادی پیش‌بینی عددی بود. شکل [18. 1](#bookmark826) نمودارهای پراکندگی دو پیش‌بینی و نتیجه را نشان می‌دهد. رابطه بین حلالیت و تعداد اتم‌های کربن نسبتاً خطی و مستقیم است. همبستگی ساده 0 *-* تخمین زده شد. 61 و همبستگی رتبه *-* 0 *بود.* 67. برای پیش‌بینی سطح، وضعیت پیچیده تر است. گروهی از ترکیبات با سطح کم وجود دارد و برای این ترکیبات حلالیت ­نیز کم است. بقیه ترکیبات تا یک نقطه حلالیت بالاتری دارند و پس از آن روند حاشیه‌ای وجود دارد. این می‌تواند یک قطعه اطلاعات ارزشمند برای مدل ساز باشد و توسط آماره همبستگی به دست نمی‌آید. برای این پیش‌بینی، همبستگی ساده 0. 27 و همبستگی رتبه‌ای 0. 14 برآورد شد. این مقادیر نسبتاً پایین هستند و بر این اساس در رتبه پایین قرار می‌گیرند (هفدهمین از 20 پیش‌بینی).

یک جایگزین، استفاده از روش‌های انعطاف‌پذیرتر است که ممکن است قادر به مدل‌سازی روابط عمومی غیرخطی باشند. یکی از این تکنیک‌ها، مدل رگرسیون وزن‌دار محلی [(](#bookmark1013) که بیشتر به‌عنوان LOESS شناخته می‌شود) کلیولند و دولین [( 1988](#bookmark1013) ) است. این تکنیک بر اساس یک سری رگرسیون چند جمله‌ای است که داده‌ها را در محله‌های کوچک مدل‌سازی می‌کند (شبیه به محاسبه ­میانگین متحرک). این رویکرد می‌تواند در ایجاد روندهای رگرسیون هموار که بسیار تطبیقی هستند، مؤثر باشد. خطوط قرمز در شکل [18. 1](#bookmark826) LOESS هستند

NumCarbon

SurfaceArea2

o

<z>

-5

**7.**

\*•

**.1**

10-

10

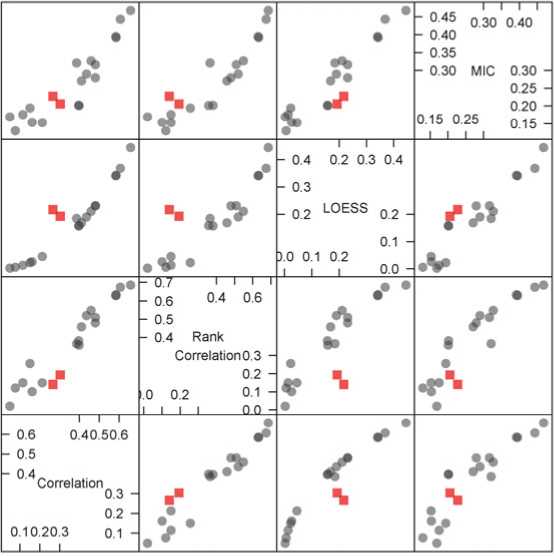
15

20

شکل 18. 1: نمودارهای پراکندگی برای دو پیش‌بینی عددی (در مقیاس‌های تبدیل شده) برای داده‌های حلالیت. خطوط *قرمز* مدل صاف‌تر هستند و مناسب‌های *نرم‌تر هستند.* از مدل مناسب، شبه *R* 2 آمار را می‌توان به دست آمده از باقی مانده محاسبه کرد. برای داده‌های حلالیت نشان داده شده در شکل.  [18. 1](#bookmark826) ، شبه LOESS *R* 2 برای مساحت سطح 0. 22 بود که بالاتر از امتیازهای همبستگی (7 از 20) است.

شکل [18. 2](#bookmark827) رابطه بین امتیازهای اهمیت را برای همه پیش‌بینی‌کننده‌های پیوسته نشان می‌دهد. سومین معیار، ضریب اطلاعات حداکثر (MIC)، در بخش بحث شده است.  [18. 3 . همبستگی](#bookmark837) رتبه و LOESS شبه *R2* به استثنای دو پیش‌بینی سطح، نتایج یکسانی را ارائه می‌دهند. هر دو روش منعکس کننده سه خوشه متمایز از پیش‌بینی‌کننده‌ها، بر اساس رتبه‌هایشان هستند: مجموعه‌ای از چهار پیش‌بینی امتیاز بالا (دو پیش‌بینی به‌طور قابل‌توجهی در طرح همپوشانی دارند)، مجموعه‌ای از پیش‌بینی‌کننده‌های با اهمیت متوسط و مجموعه دیگری با اهمیت بسیار پایین. تنها ناسازگاری بین روش‌ها این است که دو پیش‌بینی سطح سطح با استفاده از LOESS نسبت به روش‌های دیگر امتیاز بیشتری کسب می‌کنند. این پیش‌بینی‌کننده‌ها به صورت مربع‌های قرمز در تصویر نشان داده شده اند.

هر یک از این تکنیک‌ها هر پیش‌بینی را بدون در نظر گرفتن دیگران ارزیابی می‌کند. این می‌تواند به دو صورت بالقوه گمراه کننده باشد. ابتدا، اگر دو عامل پیش‌بینی ­با پاسخ و با یکدیگر همبستگی زیادی داشته باشند، رویکرد تک متغیره هر دو را مهم تشخیص می‌دهد. همانطور که دیدیم، برخی از مدل‌ها با گنجاندن این اطلاعات اضافی تأثیر منفی خواهند داشت. رویکردهای پیش پردازش مانند حذف پیش‌بینی‌کننده‌های بسیار همبسته می‌تواند این مسأله را کاهش دهد. دوم، رویکرد اهمیت تک متغیره در شناسایی گروه‌هایی از پیش‌بینی‌کننده‌ها که با هم رابطه قوی با پاسخ دارند، شکست خواهد خورد. به‌عنوان مثال، دو پیش‌بینی ممکن است ارتباط زیادی با پاسخ نداشته باشند. با این حال، تعامل آنها ممکن است باشد. همبستگی‌های تک متغیره این رابطه پیش‌بینی را نشان نمی‌دهند. یک استراتژی معقول این است که به ­جای استفاده از رتبه‌بندی به‌عنوان تنها روش برای درک روندهای اساسی، این جنبه‌های پیش‌بینی‌کننده‌ها را بیشتر مورد بررسی قرار دهیم. دانستن اینکه کدام رابطه را باید کاوش کرد اغلب به دانش تخصصی در مورد داده‌ها نیاز دارد.

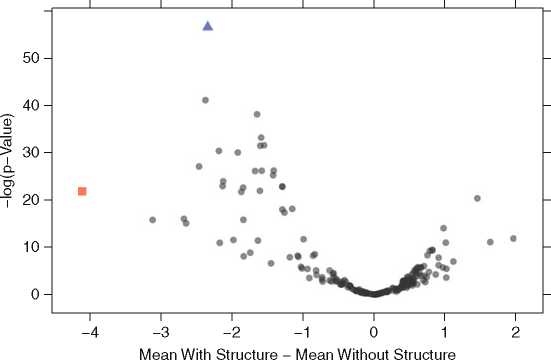


ماتریس طرح پراکندگی

شکل 18. 2: معیارهای اهمیت ماتریس نمودار پراکندگی برای 20 پیش‌بینی پیوسته در داده‌های حلالیت. دو پیش‌بینی سطح به صورت *مربع‌های قرمز نشان داده شده اند*

هنگامی که پیش‌بینی‌کننده‌ها مقوله‌ای هستند، روش‌های متفاوتی مورد نیاز است. در داده‌های حلالیت، 208 پیش‌بینی اثر انگشت وجود داشت که متغیرهای شاخصی هستند که ساختارهای اتمی خاصی را در ترکیب نشان می‌دهند. ساده‌ترین روش برای تعیین ارتباط هر پیش‌بینی باینری، ارزیابی این است که آیا میانگین نتیجه در هر دسته متفاوت است یا خیر. اثر انگشت FP175 را در نظر بگیرید. تفاوت بین مقدار حلالیت متوسط با و بدون ساختار *-* 0 *است.* 002. با توجه به اینکه حلالیت از *-* 11 متغیر است. 6 تا 1. 6، این احتمالاً ناچیز است. از طرف دیگر، اثر انگشت FP044 دارای تفاوت *-* 4 *است.* 1 واحد ورود به سامانه‌. این بزرگتر است اما به تنهایی ممکن است آموزنده نباشد. تغییرات در داده‌ها نیز باید در نظر گرفته شود.

طبیعی‌ترین روش برای مقایسه میانگین دو گروه، آماره *t استاندارد* است که اساساً نسبت سیگنال به نویز است (تفاوت ­میانگین‌ها بر اساس تابعی از متغیرهای گروه‌ها تقسیم می‌شود). با این روش می‌توان یک مقدار *p* تولید کرد که در آن فرضیه صفر این است که تفاوتی بین گروه‌ها وجود ندارد. فرض برای آمار



شکل 18. 3: نمودار آتشفشانی از نتایج آزمون ­*t که برای پیش‌بینی* اثر انگشت اعمال می‌شود. *مربع قرمز* اثر انگشت FP044 است (که بیشترین تفاوت را در میانگین دارد) در حالی که *مثلث آبی* اثر انگشت FP076 است و از نظر آماری مهم‌ترین است.

این است که داده‌ها به‌طور معمول توزیع می‌شوند. اگر بعید است که این فرض درست باشد، روش‌های دیگر (مانند آزمون جمع رتبه ویلکاکسون) ممکن است مناسب تر باشند.

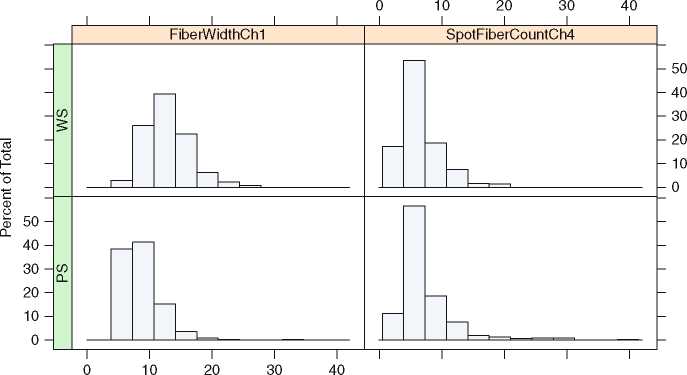
برای هر یک از 208 پیش‌بینی اثر انگشت، از آزمون *t استفاده شد.* به جای تکیه صرفاً بر مقدار *p،* شکل.  [شکل 18. 3](#bookmark828) یک *نمودار آتشفشانی را نشان می‌دهد* که در آن یک نسخه ترانس ­شکل از *p -value* در محور *y* و تفاوت میانگین حلالیت در محور ­*x* نشان داده شده است. مقادیر بالاتر محور *y* نشان‌دهنده اهمیت آماری قوی است (یعنی *p-value کم* ). اکثر تفاوت‌های میانگین ­منفی هستند که نشان می‌دهد حلالیت بدون ساختار بیشتر است. FP044 بیشترین تفاوت را دارد (و به صورت مربع قرمز نشان داده می‌شود). با این حال، این تفاوت از نظر آماری مهم‌ترین نیست. اثر انگشت FP076 از نظر آماری مهم‌ترین است (به صورت مثلث آبی نشان داده شده است) و تفاوت در میانگین *-* 2 *دارد.* 3 واحد ورود به سامانه‌. همانطور که قبلا ذکر شد، آماره *t* یک نسبت سیگنال به نویز است. در حالی که اثر انگشت FP044 بزرگترین سیگنال را دارد، اما تغییرات قابل‌توجهی نیز دارد که از اهمیت آن می‌کاهد. این در فواصل اطمینان 95 درصدی برای تفاوت‌های میانگین منعکس می‌شود: (3. 6، 4. 7) برای FP044 و (2. 1، 2. 6) برای اثر انگشت FP076. اینکه کدام پیش‌بینی مهم تر است ­به زمینه و مجموعه داده‌های خاص بستگی دارد. اگر سیگنال به اندازه کافی بزرگ باشد، مدل ممکن است بتواند بر نویز غلبه کند. در موارد دیگر، تفاوت‌های کوچکتر اما دقیق تر ممکن است عملکرد بهتری ایجاد کند.

هنگامی که پیش‌بینی بیش از دو مقدار دارد، می‌توان از مدل تحلیل واریانس (ANOVA) برای مشخص کردن اهمیت آماری پیش‌بینی‌کننده‌ها استفاده کرد. با این حال، اگر مشخص شود که ابزارهای مقوله‌ها متفاوت هستند، گام طبیعی بعدی این است که کشف کنیم که کدام‌یک متفاوت هستند. به همین دلیل، ممکن است مفید باشد که دسته‌ها را به چندین متغیر ساختگی باینری تجزیه کنیم و روشی را که در بالا ذکر شد برای تعیین ارتباط هر دسته اعمال کنیم.

نتایج طبقه‌بندی شده

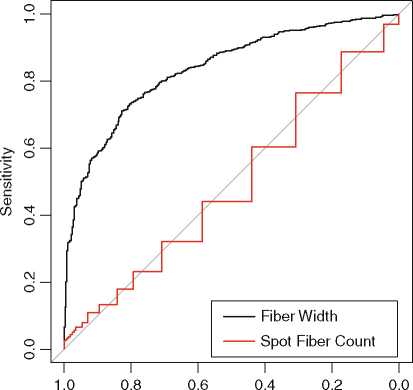
با نتایج طبقه‌بندی شده و پیش‌بینی‌کننده‌های عددی، چندین رویکرد برای تعیین کمیت اهمیت پیش‌بینی وجود دارد. داده‌های ذکر بخش تصویر از فصل. ­ [3](#bookmark131) برای نشان دادن تکنیک‌ها استفاده خواهد شد. شکل [18. 4](#bookmark831) هیستوگرام دو پیش‌بینی (عرض فیبر برای کانال 1 و تعداد فیبر نقطه‌ای برای کانال 4) را برای هر کلاس نشان می‌دهد. عرض فیبر یک تغییر در مقادیر متوسط بین کلاس‌ها را نشان می‌دهد در حالی که توزیع تعداد فیبر نقطه‌ای بین طبقات بسیار مشابه به نظر می‌رسد.

یک رویکرد زمانی که دو کلاس وجود دارد، استفاده از ناحیه زیر منحنی ROC برای تعیین کمیت ارتباط پیش‌بینی است. بخش ها [11. 3](#bookmark17) و [16. 4](#bookmark757) از منحنی ROC با احتمالات کلاس پیش‌بینی شده به‌عنوان ورودی استفاده کرد. در اینجا، از داده‌های پیش‌بینی به‌عنوان ورودی منحنی ROC استفاده می‌کنیم. اگر پیش‌بینی می‌توانست کلاس‌ها را کاملاً از هم جدا کند، یک بریدگی برای پیش‌بینی وجود دارد که حساسیت و ویژگی 1 را به دست می‌آورد و سطح زیر منحنی یک می‌شود. مانند قبل، یک پیش‌بینی کاملاً نامربوط، مساحتی زیر منحنی تقریباً 0. 5 خواهد داشت. شکل [18. 5](#bookmark832) مساحت زیر منحنی را برای پیش‌بینی‌کننده‌های عرض فیبر و تعداد فیبر نقطه‌ای نشان می‌دهد. منطقه تحت ROC



شکل 18. 4: هیستوگرام برای دو پیش‌بینی در داده‌های بخش‌بندی سلول،

برای هر کلاس جدا شده است



اختصاصی

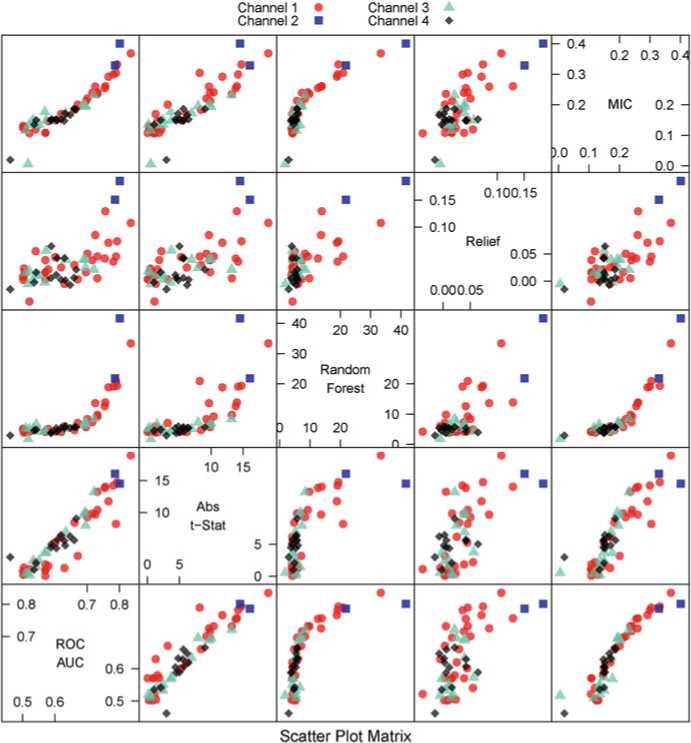
شکل 18. 5: منحنی‌های ROC برای دو پیش‌بینی در داده‌های تقسیم‌بندی سلولی

منحنی برای داده‌های عرض فیبر 0. 836 بود در حالی که مقدار برای تعداد فیبر نقطه‌ای 0. 538 بود. این بازتاب روندهایی است که در شکل 1 مشاهده می‌شود.  [18. 4](#bookmark831) که در آن عرض فیبر جدایی بیشتری را بین طبقات نشان داد.

هنگامی که چندین کلاس وجود دارد، پسوندهای منحنی ROC توسط [هانلی و مک نیل](#bookmark1017) [( 1982](#bookmark1017) ) [دلونگ و همکاران ( 1988](#bookmark1014) ) [ونکاترامان](#bookmark1027) [( 2000](#bookmark1027) ) و [پپه و همکاران ( 2009](#bookmark1023) ) نیز می‌تواند مورد استفاده قرار گیرد. از طرف دیگر، مجموعه‌ای از منحنی‌های ROC "یک در مقابل همه" را می‌توان با جمع کردن همه کلاس‌ها به جز یکی با هم ایجاد کرد. به این ترتیب، یک AUC جداگانه برای هر کلاس وجود خواهد داشت و ارتباط کلی را می‌توان با استفاده از میانگین یا حداکثر AUC در کلاس‌ها تعیین کرد.

یکی دیگر از رویکردهای ساده تر، آزمایش این است که آیا مقادیر میانگین پیش‌بینی‌کننده‌ها در هر کلاس متفاوت است یا خیر. این شبیه به تکنیک توضیح داده شده در بخش قبل است، اما، در این زمینه، پیش‌بینی به‌عنوان نتیجه در نظر گرفته می‌شود. نمودارهای آتشفشانی را نمی‌توان استفاده کرد زیرا هر یک از تفاوت‌ها ممکن است در مقیاس‌های مختلف باشد. با این حال، آمار *t می‌*تواند برای مقایسه پیش‌بینی‌کننده‌ها در مقیاس‌های مختلف استفاده شود. برای عرض فیبر، مقدار آماره *t -* 19 بود که نشان دهنده سیگنال واضح بالای نویز است. سیگنال برای تعداد فیبر نقطه‌ای بسیار ضعیف تر بود، با مقدار 3.

برای داده‌های تقسیم‌بندی، 58 پیش‌بینی پیوسته با استفاده از هر یک از این روش‌ها همراه با امتیاز اهمیت متغیر تصادفی جنگل و دو روشی که در بخش توضیح داده شده‌اند، ارزیابی شدند.  [18. 3](#bookmark837) (MIC و Relief). روش‌ها تقریباً با هم هماهنگ هستند (شکل 2).  [18. 6 )](#bookmark833) . سطح زیر منحنی ROC یک رابطه تنگ و منحنی خطی با امتیاز جنگل تصادفی و یک رابطه خطی قوی با آماره *t* نشان می‌دهد. آماره *t* و تصادفی



شکل 18. 6: یک ماتریس نمودار پراکندگی از چندین معیار مختلف برای رتبه‌بندی ­پیش‌دگرها با مجموعه داده‌های تقسیم‌بندی تصویر

به نظر می‌رسد جنگل همان پیش‌بینی‌کننده‌ها را به‌عنوان کم‌اهمیت‌تر رتبه‌بندی می‌کند، اما تفاوت‌هایی در نمرات آن‌هایی وجود دارد که به‌عنوان مهم‌ترین کمیت تعیین شده‌اند. سه ­پیش‌دیگر معمولاً در بین روش‌ها رتبه‌بندی بسیار بالایی دارند: شدت متوسط (کانال 2)، عرض فیبر (کانال 1) و شدت کل (کانال 2).

وقتی پیش‌بینی مقوله‌ای است، چندین معیار وجود دارد که ممکن است مناسب باشند. برای پیش‌بینی‌کننده‌های باینری و دو کلاس، یک روش مؤثر برای اندازه‌گیری ارتباط، نسبت شانس است. به یاد بیاورید که شانس یک احتمال *p (*1 *- p* ) است. احتمال یک رویداد را می‌توان برای هر دو سطح محاسبه کرد

جدول 18. 1: آمار برای سه پیش‌بینی دسته‌بندی باینری در داده‌های گرنت

|  |  |
| --- | --- |
|  | موفقیت عطا کن  بله خیر % یا *p-value* نسبت سود |
| اسپانسر  62B دیگر | 7 44 14  3226 3356 49 6. 0 2. 6e *-* 07  0. 04726 |

CVB

باند ناشناخته 644 2075 24

|  |  |
| --- | --- |
| گروه شناخته شده | 2589 1325 66 6. 3 1. 7e *-* 263  0. 13408 |
| کد RFCD  240302  کد دیگر | 13 15 46  3220 3385 49 1. 1 8. 5e *-* 01  0. 00017 |

پیش‌بینی اگر اینها با *p* 1 و نشان داده شوند [*p* 2 ،](#bookmark1011) نسبت شانس ( بلند و آلتمن [2000](#bookmark1011) ؛ آگرستی 2002 ) است و نشان دهنده افزایش شانس رویداد هنگام رفتن از سطح اول پیش‌بینی به سطح دیگر است.

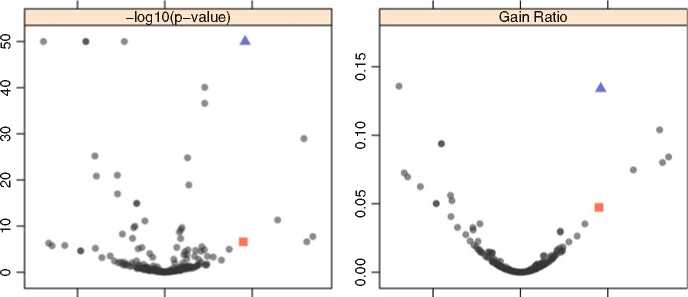
OR =

*P* 1(1 *- P*2)

*P*2(1 *- P*1)

برای نشان دادن این رویکرد، از داده‌های درخواست گرنت از فصل‌های قبلی استفاده می‌شود. برای این داده‌ها 226 پیش‌بینی باینری با حداقل 25 ورودی در هر خانه از جدول 2 *×* 2 بین پیش‌بینی و نتیجه وجود دارد. جدول [18. 1](#bookmark834) جدول‌های متقابل را برای سه پیش‌بینی در داده‌های گرنت نشان می‌دهد. برای باند ارزش قرارداد، احتمال موفقیت گرنت زمانی تعیین می‌شود که باند شناخته شده باشد ( *P* 1 = 0. 661 *)* و ناشناخته ( *P* 2 = 0. 237). *نسبت شانس* متناظر ­6. 3 است، به این معنی که با مشخص شدن باند، شانس موفقیت بیش از شش برابر افزایش می‌یابد. مانند بحث در مورد تفاوت ­در میانگین، نسبت شانس فقط سیگنال را منعکس می‌کند، نه نویز را. اگرچه فرمول‌هایی برای محاسبه فاصله اطمینان برای نسبت شانس وجود دارد، یک رویکرد رایج تر انجام یک آزمون فرضیه آماری است که نسبت شانس برابر با یک است (یعنی شانس مساوی بین سطوح پیش‌بینی). هنگامی که دو مقوله و دو مقدار برای پیش‌بینی وجود دارد، می‌توان از ­آزمون عمل فیشر ( Agresti 2002 ) برای ارزیابی این فرضیه استفاده کرد و می‌توان آن را با *P-* value حاصل خلاصه کرد. برای این پیش‌بینی، *P- value به‌طور* موثر ­صفر بود که نشان می‌دهد سطوح باند ارزش قرارداد با موفقیت گرنت مرتبط است. جدول [18. 1](#bookmark834) پیش‌بینی دیگری را برای حامی 62B نشان می‌دهد که دارای نسبت شانس مشابه (OR = 6) است، اما تعداد گرنت با این حامی بسیار کم است. در اینجا، *P-* value هنوز کوچک بود، اما بسیار پایین‌تر رتبه‌بندی می‌شد

-2 0 2



-2 0 2

Log Odds Ratio

شکل 18. 7: نسبت شانس ورود به سامانه در برابر دو آمار دیگر ترسیم شده است. *سمت چپ* : تابعی از *p* -values از آزمون دقیق فیشر نشان داده شده است. بیشترین مقدار در داده‌ها 263 است. این تصویر در 50 کوتاه می‌شود تا پرسپکتیو حفظ شود. *راست* : نسبت بهره برای هر پیش‌بینی. *مربع قرمز* پیش‌بینی حامی 62B و *مثلث آبی* برای نوار ارزش قرارداد ناشناخته است

با استفاده از معناداری آماری پانل سمت چپ شکل.  [18. 7](#bookmark837) نمودار آتشفشانی را برای نسبت شانس و *p -value مربوطه* برای 226 پیش‌بینی گر دودویی نشان می‌دهد.

هنگامی که بیش از دو کلاس وجود دارد یا پیش‌بینی‌کننده‌ها بیش از دو سطح دارند، می‌توان از روش‌های دیگری استفاده کرد. هنوز هم می‌توان از آزمون دقیق فیشر برای اندازه‌گیری ارتباط بین پیش‌بینی و کلاس‌ها استفاده کرد. از طرف دیگر، نسبت بهره مورد استفاده برای C4. 5 را می‌توان برای کمی کردن رابطه بین دو متغیر، در جایی که بزرگتر بهتر است، اعمال کرد. به یاد بیاورید که نسبت بهره برای تعداد سطوح در پیش‌بینی تنظیم می‌شود تا بایاس در برابر ویژگی‌های دارای سطوح کم را حذف کند. به همین دلیل، به دست آوردن اطلاعات را می‌توان برای همه پیش‌بینی‌کننده‌های طبقه بندی، صرف نظر از ویژگی‌های آنها اعمال کرد. جدول [18. 1](#bookmark834) سود اطلاعاتی را برای سه پیش‌بینی گرنت نشان می‌دهد و شکل.  [18. 7](#bookmark837) در مقابل آمار افزایش در مقابل نسبت شانس است. مانند *p-value* برای آزمون دقیق فیشر، آمار افزایش، علیرغم شباهت‌های نسبت شانس، پیش‌بینی باند ارزش قرارداد را نسبت به حامی 62B ترجیح می‌دهد.

سایر رویکردها

الگوریتم امداد ( Kira [and Rendell 1992](#bookmark1020) ) یک روش عمومی برای تعیین کمیت اهمیت پیش‌بینی است. در ابتدا برای ­مسائل طبقه‌بندی با دو کلاس توسعه داده شد، اما برای کار در سطح a گسترش یافته است

نمرات پیش‌بینی *S j* را به صفر مقداردهی کنید

برای *i* = 1. *. . m به‌طور تصادفی انتخاب شده نمونه مجموعه آموزشی (R i )* انجام دهید

در مجموعه آموزشی نزدیکترین گل را پیدا کنید و ضربه بزنید

برای *j* = 1. *. . p متغیرهای پیش‌بینی* انجام می‌دهند

امتیاز را برای هر پیش‌بینی بر اساس نزدیکی تنظیم کنید

*Rj به نزدیکترین* رد و ضربه بزنید:

*S j* = *S j -diff j (* *R j ,Hit* ) 2 */m* + *diff j (* *R j , Miss* ) 2 */m*

پایان

پایان

الگوریتم 18. 1: الگوریتم اصلی Relief برای رتبه‌بندی متغیرهای پیش‌بینی در مدل‌های طبقه‌بندی با دو کلاس

طیف وسیع‌تری از مسائل می‌تواند پیش‌بینی‌کننده‌های پیوسته و همچنین متغیرهای ساختگی را در خود جای دهد و می‌تواند روابط غیرخطی بین پیش‌بینی‌کننده‌ها و نتیجه را تشخیص دهد. از نقاط انتخاب شده تصادفی و نزدیکترین همسایگان آنها برای ارزیابی هر پیش‌بینی به صورت مجزا استفاده می‌کند.

برای یک پیش‌بینی خاص، امتیاز تلاش می‌کند تا جدایی ­بین کلاس‌ها را در بخش‌های جدا شده از داده‌ها مشخص کند. الگوریتم [18. 1](#bookmark838) روش را نشان می‌دهد.

برای یک نمونه مجموعه آموزشی به‌طور تصادفی انتخاب شده، الگوریتم نزدیک‌ترین نمونه‌ها را از هر دو کلاس پیدا می‌کند (به نام‌های "Hits" و "Miss"). برای هر پیش‌بینی، اندازه‌گیری تفاوت در مقادیر پیش‌بینی بین نقطه تصادفی داده‌های [p](#bookmark1020) و ضربه‌ها و خطاها محاسبه می‌شود. برای پیش‌بینی‌کننده‌های پیوسته، Kira و Rendell [( 1992](#bookmark1020) ) پیشنهاد کردند که فاصله بین دو نقطه بر گستره کلی پیش‌بینی تقسیم شود:

diff( *x, y* ) = ( *x - y* ) */C*

که در آن *C* یک ثابت برای پیش‌بینی است که تفاوت را بین 0 و 1 مقیاس می‌کند. برای داده‌های باینری (یعنی 0/1)، یک نشانگر ساده برای هم ارزی می‌توان استفاده کرد:

diff( *x,y* )= *|x - y|*

به‌طوری که این مقادیر نیز بین 0 تا 1 باشد.

امتیاز کلی ( *Sj ) انباشته‌ای* از این تفاوت‌ها است به‌طوری که اگر ضربه دور از مقدار تصادفی انتخاب شده باشد، امتیاز کاهش می‌یابد، اما اگر فاصله دور باشد، امتیاز افزایش می‌یابد. ایده این است که پیش‌بینی‌ای که جدایی بین کلاس‌ها را نشان می‌دهد باید در نزدیکی‌ها ضربه می‌خورد و دوردست را از دست می‌داد. با توجه به این، نمرات بزرگتر نشان دهنده پیش‌بینی‌کننده‌های مهم است.

به‌عنوان مثال، شکل.  [18. 8](#bookmark839) مجموعه‌ای از داده‌ها را برای چندین پیش‌بینی ارائه می‌دهد. پانل بالایی یک پیش‌بینی را نشان می‌دهد که به‌طور کامل کلاس‌ها را از هم جدا می‌کند. فرض کنید روش نمونه‌گیری دومین تا آخرین نمونه را در سمت چپ انتخاب می‌کند

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| • • • • | • |  | ■ | ■ ■ ■ | ■ |
| 1  -1. 0  1 1 | -0. 5  من | من 0. 0  پیش‌بینی A  من | . 0. 5  من | من  1. 0  من | من |
| • ■ • | ■ | • ■ | • | ■ • | ■ |
| 1 1  -1. 5 -1. 0 | من  -0. 5 | من  0. 0 | l 0. 5 | l 1. 0 | جی  1. 5 |

Class 1

Class 2 ■

پیش‌بینی B

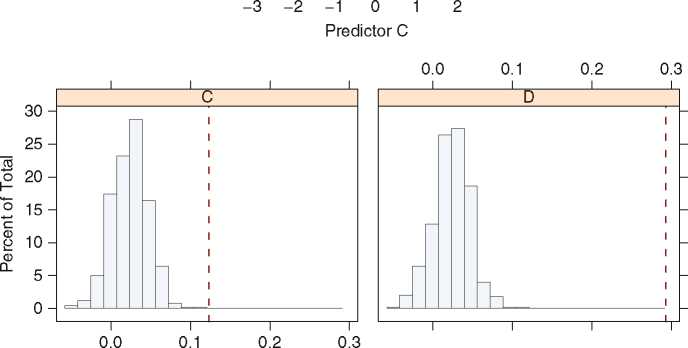
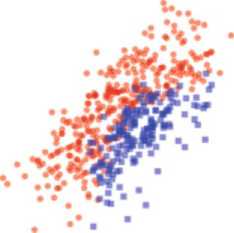
شکل 18. 8: سه مجموعه داده که الگوریتم Relief را نشان می‌دهد. در *پانل بالا،* پیش‌بینی *A* جداسازی کامل را با امتیاز ReliefF 0. 19 نشان می‌دهد. پانل *میانی* یک پیش‌بینی کاملاً غیر اطلاعاتی با امتیاز ReliefF مربوطه 0 را نشان می‌دهد (پیش‌بینی *A* = *-* 1. 1). نزدیکترین ضربه در دو طرف این نقطه داده است، با اختلاف تقریباً 0. 2 (در اینجا از ثابت صرف نظر می‌کنیم). نزدیکترین فاصله دور است ( *A* = 0. 5 *)* با اختلاف متناظر *-* 1. 61. اگر این اولین ست پوینت آموزشی بود که به‌طور تصادفی انتخاب می‌شد، امتیاز می‌شد

*S A* = 0 *-* 0. 2 2 + *-* 1. 61 2 = 2. 55

این مقدار بر *m تقسیم می‌شود* و مقادیر جدید جمع می‌شود. به‌عنوان مثال متقابل، پیش‌بینی *B* در شکل.  [18. 8](#bookmark839) کاملاً غیر آموزنده است. هر ضربه و از دست دادن در مجاورت هر نمونه تصادفی انتخاب شده است، بنابراین تفاوت‌ها همیشه یکدیگر را جبران می‌کنند و امتیاز این پیش‌بینی *S B* = 0 است.

این روش متعاقباً توسط [کونوننکو ( 1994](#bookmark1020) ). الگوریتم اصلاح‌شده که ReliefF نام دارد، از بیش از یک نزدیکترین همسایه استفاده می‌کند، از یک معیار تفاوت اصلاح‌شده استفاده می‌کند و اجازه می‌دهد تا بیش از دو کلاس و همچنین مقادیر پیش‌بینی از دست رفته را داشته باشد. علاوه بر این، [Robnik-Sikonja و Kononenko ( 1997](#bookmark1024) ) الگوریتم را برای رگرسیون (یعنی نتایج عددی) تطبیق دادند.

پانل پایین شکل.  [18. 9](#bookmark840) یک مجموعه داده فرضی را با دو کلاس و دو پیش‌بینی همبسته (که با *C* و *D مشخص می‌شوند* ) نشان می‌دهد. مرز کلاس با استفاده از یک معادله مدل رگرسیون لجستیک ساده با دو اثر اصلی قوی و اثر متقابل متوسط ایجاد شد:



2

1

0

-1

-2

Relief Score

Fig. 18.9: *Top*: two predictors with distinct class boundary. In this example, the ReliefF scores for predictors *C* and *D* are 0.17 and 0.25, respectively. *Bot­tom*: the permutation distribution of the scores when there is no relationship between the predictors and the classes. The *dashed lines* indicate the observed ReliefF scores

log (i^) =

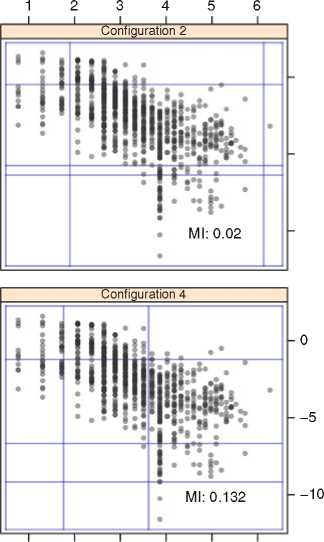
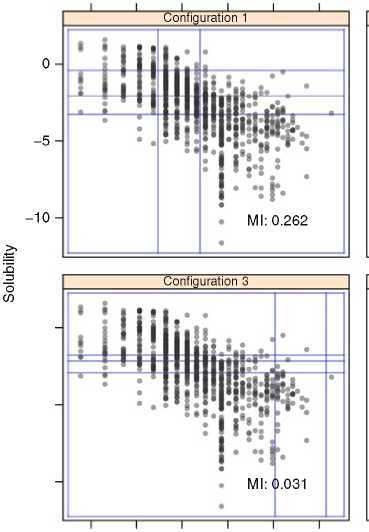
*-*4*C-*4*D*+2*CD*

داده‌های نشان داده شده در شکل.  [18. 9](#bookmark840) با این مرز کلاس شبیه‌سازی شد. برای این داده‌ها، یک مدل رگرسیون لجستیک قادر به دستیابی به نرخ دقت اعتبار متقاطع 90. 6٪ بود. با دانستن معادله مدل واقعی، هر دو پیش‌بینی باید به‌عنوان مهم تعیین شوند. از آنجایی که یک اثر متقابل در معادله مدل وجود دارد، می‌توان انتظار داشت که الگوریتم ReliefF سیگنال بزرگ‌تری را نسبت به سایر روش‌هایی که تنها یک پیش‌بینی را در یک زمان در نظر می‌گیرند، نشان دهد. برای این داده‌ها، الگوریتم ReliefF با *m* = 50 نمونه تصادفی و *k* = 10-نزدیک‌ترین همسایه استفاده شد. دو پیش‌بینی قبل از اجرای الگوریتم مرکز و مقیاس‌بندی شدند. نمرات پیش‌بینی‌کننده‌های *C* و *D* به ترتیب 0. 17 و 0. 25 بود. نواحی زیر منحنی‌های ROC برای پیش‌بینی‌کننده‌های *C* و *D* متوسط، با مقادیر 0. 65 و 0. 69 بود.

چه برشی برای این امتیازات باید استفاده شود؟ [کیرا و رندل](#bookmark1020) [( 1992](#bookmark1020) ) پیشنهاد می‌کند که یک آستانه "می تواند بسیار کوچکتر از 1 */^Jam* " باشد که در آن *a* نرخ مثبت کاذب مورد نظر است (یعنی درصد زمانی که یک پیش‌بینی نامربوط مهم ارزیابی می‌شود). با استفاده از این معیار با نرخ مثبت کاذب 5 درصد، مقادیر بیشتر از 0. 63 ممکن است مهم در نظر گرفته شوند. جایگزین دیگر استفاده از یک روش آماری تثبیت شده به نام *تست جایگشت است* [( خوب](#bookmark1017) [2000](#bookmark1017) ) برای ارزیابی نمرات. در اینجا، برچسب‌های کلاس واقعی به‌طور تصادفی با هم مخلوط می‌شوند و امتیازات ReliefF بارها مجدداً محاسبه می‌شوند. وقتی پیش‌بینی هیچ ارتباطی ندارد، این باید تا حدودی توزیع امتیازات را ارائه دهد. از این رو، امتیاز مشاهده شده را می‌توان در مقابل این توزیع قرار داد تا مشخص شود که چقدر نسبت به این فرض که پیش‌بینی مرتبط نیست، افراطی است. با استفاده از 500 جایگشت تصادفی داده‌ها، توزیع امتیازات برای پیش‌بینی‌کننده‌های *C* و *D* شبیه داده‌های توزیع شده نرمال بود (شکل 2).  [18. 9 )](#bookmark840) . برای پیش‌بینی *C* , مقادیر امتیاز از 0 *- متغیر بود.* 05 تا 0. 11، در حالی که برای پیش‌بینی *D،* محدوده *-* 0 *بود.* 05 تا 0. 11. در این زمینه، دو مقدار مشاهده شده برای داده‌های شکل 1.  [18. 9](#bookmark840) نشان دهنده پیش‌بینی‌کننده‌های بسیار مهم است. با توجه به شکل‌های متقارن و توزیع شده معمولی جایگشت‌های تصادفی، می‌توان تعیین کرد که امتیازات پیش‌بینی‌کننده‌های *C* و *D* 4. 5 و 11. 6 انحراف استاندارد از آنچه که باید به‌طور تصادفی انتظار می‌رود فاصله دارد. با توجه به افراطی بودن نمرات، نتیجه‌گیری می‌تواند این باشد که هر دو پیش‌بینی فاکتورهای مهمی برای جداسازی طبقات هستند.

برای داده‌های تقسیم‌بندی سلولی، شکل.  [18. 6 نمرات](#bookmark833) ReliefF را برای هر پیش‌بینی نشان می‌دهد. این مقادیر با معیارهای دیگر همبستگی متوسطی دارند اما تغییرات بیشتری را در روابط نشان می‌دهند. به‌عنوان مثال، ناحیه زیر منحنی ROC نویز کمتری را در روابط با سایر معیارها نشان می‌دهد. این ممکن است به دلیل معیارهای اندازه‌گیری جنبه‌های مختلف داده‌ها باشد یا ممکن است به دلیل نمونه‌گیری تصادفی مورد استفاده توسط ReliefF باشد.

رشف و همکاران [( 2011](#bookmark1024) ) معیار جدیدی را برای تعیین کمیت رابطه بین دو متغیر به نام MIC توصیف می‌کند. روش آنها ­منطقه دو بعدی تعریف شده توسط پیش‌بینی و نتیجه را به مجموعه‌هایی از ­شبکه‌های دو بعدی تقسیم می‌کند. به‌عنوان مثال، شکل.  [18. 10](#bookmark842) نمودار پراکندگی ­داده انحلال پذیری را نشان می‌دهد که در آن چهار شبکه تصادفی 4 *×* 3 برای پیش‌بینی تعداد اتم‌های کربن ایجاد شده است. در داخل هر شبکه، تعداد نقاط داده محاسبه شده و برای محاسبه آمار *اطلاعات متقابل استفاده می‌شود* [( Brillinger 2004](#bookmark1012) ) که مربوط به معیارهای اطلاعاتی شرح داده شده در فصل است.  [14](#bookmark683) برای درخت‌های تصمیم C4. 5 و C5. 0. بسیاری از پیکربندی‌های مختلف با اندازه شبکه یکسان ارزیابی می‌شوند ­و بزرگترین مقدار اطلاعات متقابل تعیین می‌شود. این فرآیند برای بسیاری از اندازه‌های شبکه مختلف (مثلاً 2 *\** 2، 10 *\** 3) تکرار می‌شود. خلاصه مقادیر اطلاعات متقابل در همه سطل‌ها نرمال‌سازی می‌شود و بزرگترین مقدار در بین تمام پیکربندی‌های bin به‌عنوان قدرت ارتباط بین پیش‌بینی و نتیجه استفاده می‌شود.



1 23456

تعداد اتم‌های کربن

شکل 18. 10: نمونه‌هایی از نحوه ایجاد معیار حداکثر اطلاعات (MIC). هر پانل یک پیکربندی تصادفی از شبکه‌های 4 *×* 3 را برای یکی از پیش‌بینی‌کننده‌های حلالیت با آمار اطلاعات متقابل مربوطه نشان می‌دهد.

نویسندگان نشان می‌دهند که این روش می‌تواند انواع زیادی از روابط، مانند امواج گناه، بیضی‌ها و دیگر الگوهای بسیار غیرخطی را ­تشخیص دهد. این معیار مقیاسی مشابه با آماره همبستگی ساده دارد که در آن مقدار صفر نشان دهنده هیچ رابطه‌ای نیست، در حالی که مقدار یک نشان دهنده یک رابطه بسیار قوی است. یک مسأله بالقوه در مورد چنین تکنیک کلی این است که ممکن است تحت برخی شرایط به خوبی دیگران عمل نکند. برای مثال، اگر رابطه واقعی بین پیش‌بینی و نتیجه خطی باشد، آماره همبستگی ساده ممکن است عملکرد بهتری داشته باشد زیرا مختص روندهای خطی است.

آمار MIC قبلا در شکل 1 نشان داده شده است.  [18. 1](#bookmark826) برای پیش‌بینی حلالیت پیوسته. در اینجا، مقادیر MIC رابطه قوی با سایر معیارها دارند. برای داده‌های تقسیم‌بندی سلولی نشان داده شده در شکل.  [18. 6](#bookmark833) ، آماره MIC یک همبستگی قوی با آماره *t مطلق،* مساحت زیر منحنی ROC و نمرات جنگل تصادفی دارد. همبستگی کمتری با ReliefF وجود دارد.

18. 4 محاسبات

این بخش از توابع بسته‌های R زیر استفاده می‌کند: AppliedPredictive-Modeling، caret، CORElearn، minerva، pROC و randomForest.

داده‌های تقسیم‌بندی سلولی از فصل.  [3](#bookmark131) مورد استفاده است و در بسته‌بندی کارت یافت می‌شود. داده‌های حلالیت از Chaps. از 6 تا 9 نیز استفاده می‌شود و می‌توان آنها را در بسته AppliedPredictiveModeling پیدا کرد. در نهایت، داده‌های درخواست گرنت از Chaps.  [12](#bookmark550) تا 15 نیز استفاده می‌شود (به بخش مراجعه کنید.  [12. 7](#bookmark585) برای خواندن و پردازش این داده ها).

نتایج عددی

برای تخمین همبستگی بین پیش‌بینی‌کننده‌ها و نتیجه، از تابع cor استفاده می‌شود. مثلا،

*کتابخانه (AppliedPredictiveModeling)*

*داده (حلالیت)*

*cor(solTrainXtrans$NumCarbon، solTrainY)*

[1] -0. 6067917

برای به دست آوردن نتایج برای همه پیش‌بینی‌کننده‌های عددی، می‌توان از تابع application برای انجام محاسبات یکسان در بسیاری از ستون‌ها استفاده کرد.

*## تعیین کنید که کدام ستون‌ها دارای رشته "FP" در نام و*

*## برای بدست آوردن پیش‌بینی‌کننده‌های عددی اینها را حذف کنید*

*fpCols<- grepl("FP"، names(solTrainXtrans))*

*## برای به دست آوردن نام‌های پیش‌بینی عددی، این‌ها را حذف کنید*

*numericPreds <- names(solTrainXtrans)[!fpCols]*

*corrValues <- application(solTrainXtrans[, numericPreds]،*

*+ MARGIN = 2،*

*+ FUN = تابع (x، y) cor(x، y)،*

*+ y = solTrainY)*

*head (corrValues)*

MolWeight NumAtoms NumNonHAtoms NumBonds NumNonHBonds -0. 6585284 -0. 4358113 -0. 5836236 -0. 4590395 -0. 5851968

NumMultBonds

-0. 4804159

با بدست آوردن همبستگی رتبه، تابع corr یک روش گزینه = "Spearman" دارد.

نرم‌افزار LOESS با تابع لس در کتابخانه آمار قابل دسترسی است. برای تعیین مدل از روش فرمول استفاده می‌شود:

*نرمتر <- لس (solTrainY ~ solTrainXtrans$NumCarbon)*

*صاف تر*

صدا زدن:

لس (فرمول = solTrainY ~ solTrainXtrans$NumCarbon)

تعداد مشاهدات: 951

تعداد پارامترهای معادل: 5. 3

خطای استاندارد باقی مانده: 1. 548

تابع شبکه xyplot برای نمایش برازش LOESS مناسب است:

*xyplot(solTrainY ~ solTrainXtrans$NumCarbon،*

*+ نوع = c("p"، "صاف")،*

*+ xlab = "# کربن"،*

*+ ylab = "حلالیت")*

تابع caret filterVarImp با گزینه nonpara = TRUE (برای ­رگرسیون غیرپارامعیار) یک مدل LOESS برای هر پیش‌بینی ایجاد می‌کند و رابطه با نتیجه را کمی می‌کند:

*loessResults <- filterVarImp(x = solTrainXtrans[, numericPreds],*

*+ y = solTrainY،*

*+ ناپارا = درست)*

*سر (نتایج لوس)*

به‌طور کلی

مول وزن 0. 4443931

NumAtoms 0. 1899315

NumNonHAtoms 0. 3406166

NumBonds 0. 2107173

NumNonHBonds 0. 3424552

NumMultBonds 0. 2307995

بسته minerva را می‌توان برای محاسبه آمار MIC بین پیش‌بینی‌کننده‌ها و نتایج استفاده کرد. تابع معدن چندین کمیت از جمله مقدار MIC را محاسبه می‌کند:

*کتابخانه (minerva)*

*micValues <- mine(solTrainXtrans[, numericPreds], solTrainY)*

*## چندین آمار محاسبه شده است*

*نام‌ها (micValues)*

[1] "MIC" "MAS" "MEV" "MCN" "MICR2"

*> head(micValues$MIC)*

Y

مول وزن 0. 4679277

NumAtoms 0. 2896815

NumNonHAtoms 0. 3947092

NumBonds 0. 3268683

NumNonHBonds 0. 3919627

NumMultBonds 0. 2792600

برای پیش‌بینی‌کننده‌های طبقه‌بندی، تابع ­t. test ساده، تفاوت در میانگین و مقدار *p را محاسبه می‌کند.* برای یک پیش‌بینی:

*> t. test(solTrainY ~ solTrainXtrans$FP044)*

ولچ دو نمونه t-test

داده: solTrainY توسط solTrainXtrans$FP044

t = 15. 1984، df = 61. 891، p-value < 2. 2e-16

فرضیه جایگزین: تفاوت واقعی در میانگین برابر با 0 نیست

فاصله اطمینان 95 درصد:

3. 569300 4. 650437

برآورد نمونه:

میانگین در گروه 0 میانگین در گروه 1 -2. 472237 -6. 582105

این رویکرد را می‌توان با استفاده از اعمال به روشی مشابه آنچه در بالا برای همبستگی‌ها نشان داده شده است، به همه پیش‌بینی‌کننده‌ها گسترش داد.

*> getTstats <- تابع (x, y)*

*+{*

*+ tTest <- t. test(y~x)*

*+ خارج <- c(tStat = tTest$statistic, p = tTest$p. value)*

*+ out > tVals <- اعمال (solTrainXtrans[, fpCols]،*

*MARGIN = 2، FUN = getTstats، y = solTrainY)*

*## ابعاد را تغییر دهید*

*tVals <- t(tVals)*

*سر (tVals)*

tStat. t

FP001 -4. 022040 6. 287404e-05

FP002 10. 286727 1. 351580e-23

FP003 -2. 036442 4. 198619e-02

FP004 -4. 948958 9. 551772e-07

FP005 10. 282475 1. 576549e-23

FP006 -7. 875838 9. 287835e-15

نتایج طبقه‌بندی شده

تابع filterVarImp همچنین زمانی که متغیر نتیجه یک متغیر فاکتور R باشد، ناحیه زیر منحنی ROC را محاسبه می‌کند :

*کتابخانه (کارت)*

*داده (بخش‌بندی داده ها)*

*cellData <- زیر مجموعه (segmentationData, Case == "Train")*

*cellData$Case <- cellData$Cell <- NULL*

*## کلاس در ستون اول قرار دارد*

*head(names(cellData))*

[1] "Class" "AngleCh1" "AreaCh1" "AvgIntenCh1"

[5] "AvgIntenCh2" "AvgIntenCh3"

*> rocValues <- filterVarImp(x = cellData[,-1],*

*+ y = cellData$Class)*

*> ## ستون برای هر کلاس ایجاد می‌شود*

*> head(rocValues)*

PS WS

AngleCh1 0. 5025967 0. 5025967

AreaCh1 0. 5709170 0. 5709170

AvgIntenCh1 0. 7662375 0. 7662375

AvgIntenCh2 0. 7866146 0. 7866146

AvgIntenCh3 0. 5214098 0. 5214098

AvgIntenCh4 0. 6473814 0. 6473814

این یک لفاف ساده برای عملکردهای roc و auc در بسته‌بندی pROC است. وقتی سه یا چند کلاس وجود دارد، filterVarImp منحنی‌های ROC را برای هر کلاس در مقابل کلاس‌های دیگر محاسبه می‌کند و سپس بزرگترین ناحیه زیر منحنی را برمی‌گرداند.

آمار Relief را می‌توان با استفاده از بسته CORElearn محاسبه کرد. تابع attrEval چندین نسخه از Relief را محاسبه می‌کند (با استفاده از گزینه برآوردگر ):

*> کتابخانه (CORElearn)*

*> reliefValues <- attrEval(Class ~. , data = cellData,*

*+ ## روش‌های امدادی زیادی وجود دارد*

*+ ## موجود است. ?attrEval را ببینید*

*+ برآوردگر = "ReliefFequalK"،*

*+ ## تعداد نمونه‌های آزمایش شده:*

*+ ReliefIterations = 50)*

*> سر (مقادیر تسکین)*

AngleCh1 AreaCh1 AvgIntenCh1 AvgIntenCh2 AvgIntenCh3 AvgIntenCh4 0. 01631332 0. 02004060 0. 09402596 0. 17200400 0. 09268398 0. 0267216

همچنین می‌توان از این تابع برای محاسبه نسبت بهره، شاخص جینی و سایر امتیازات استفاده کرد. برای استفاده از یک رویکرد جایگشت برای بررسی مقادیر مشاهده شده از آمار ReliefF، بسته AppliedPredictiveModeling دارای تابع permuteRelief است :

*> perm <- permuteRelief(x = cellData[,-1],*

*+ y = cellData$Class،*

*+ nperm = 500،*

*+ برآوردگر = "ReliefFequalK"،*

*+ ReliefIterations = 50)*

ReliefF تغییر یافته در یک شی فرعی به نام جایگشت وجود دارد :

*> head(perm$permutations)*

ارزش پیش‌بینی

1 AngleCh1 -0. 009364024

2 AngleCh1 0. 011170669

3 AngleCh1 -0. 020425694

4 AngleCh1 -0. 037133238

5 AngleCh1 0. 005334315

6 AngleCh1 0. 010394028

توزیع جایگشت برای نمرات ReliefF می‌تواند مفید باشد. توگرام‌های او، مانند آنچه در شکل 1 نشان داده شده است.  [18. 9 را](#bookmark840) می توان با نحو ایجاد کرد:

*> هیستوگرام(~ مقدار|پیش‌بینی،*

*+ داده = perm$ جایگشت)*

همچنین، نسخه‌های استاندارد شده امتیازها در موضوع فرعی به نام استاندارد شده قرار دارند و تعداد انحرافات استانداردی را نشان می‌دهند که مقادیر ­ReliefF مشاهده شده (یعنی بدون جایگشت) از مرکز توزیع جایگشت شده هستند :

*> سر (perm$استاندارد)*

AngleCh1 AreaCh1 AvgIntenCh1 AvgIntenCh2 AvgIntenCh3 AvgIntenCh4

-1. 232653 3. 257958 3. 765691 8. 300906 4. 054288 1. 603847

آمار MIC را می‌توان مانند قبل محاسبه کرد اما با رمزگذاری متغیر ساختگی دودویی کلاس ها:

*> micValues <- mine(x = cellData[,-1],*

*+ y = ifelse(cellData$Class == "PS"، 1، 0))*

*> head(micValues$MIC)*

Y

AngleCh1 0. 1310570

AreaCh1 0. 1080839

AvgIntenCh1 0. 2920461

AvgIntenCh2 0. 3294846

AvgIntenCh3 0. 1354438

AvgIntenCh4 0. 1665450

برای محاسبه نسبت شانس و آزمون آماری ارتباط، تابع fisher. test در کتابخانه آمار می‌تواند اعمال شود. به‌عنوان مثال، برای محاسبه این آمار برای اشیاء گرانت ایجاد شده در بخش.  [12. 7 :](#bookmark585)

*> Sp62BTable <- table(training[pre2008, "Sponsor62B"],*

*+ آموزش [قبل از 2008، "کلاس"])*

*Sp62BTable*

موفق ناموفق 0 3226 3356

17 44

*fisher. test (Sp62BTable)*

تست دقیق فیشر برای داده‌های شمارش

داده ها: Sp62BTable

p-value = 2. 644e-07

فرضیه جایگزین: نسبت شانس واقعی برابر با 1 نیست

فاصله اطمینان 95 درصد:

2. 694138 15. 917729

برآورد نمونه:

نسبت شانس

6. 040826

وقتی پیش‌بینی بیش از دو کلاس دارد، یک نسبت شانس منفرد را نمی‌توان محاسبه کرد، اما هنوز هم می‌توان از *p-value برای ارتباط استفاده کرد:*

*> ciTable <- table(training[pre2008, "CI. 1950"],*

*+ آموزش [قبل از 2008، "کلاس"])*

*> ciTable* موفق ناموفق 0 2704 2899

1 476 455

245 39

37 7

41 0

*> fisher. test (ciTable)*

تست دقیق فیشر برای داده‌های شمارش

داده ها: ciTable p-value = 0. 3143 فرضیه جایگزین: دو طرفه

در برخی موارد، آزمایش دقیق فیشر ممکن است از نظر محاسباتی بازدارنده باشد.

در این موارد، آزمون *x 2* برای ارتباط را می‌توان محاسبه کرد:

*> DayTable <- table(training[pre2008, "Weekday"],*

*+ آموزش [قبل از 2008، "کلاس"])*

*> DayTable* با موفقیت ناموفق جمعه 542 880

دوشنبه 634 455

شنبه 615 861

خورشید 223 0

پنجشنبه 321 246

سه شنبه 377 309

چهارشنبه 521 649

*> chisq. test (DayTable)* آزمون Chi-squared پیرسون

داده ها: DayTable

X-squared = 400. 4766، df = 6، p-value < 2. 2e-16

امتیازهای اهمیت مبتنی بر مدل

همانطور که در فصل‌های قبلی توضیح داده شد، بسیاری از مدل‌ها دارای رویکردهای داخلی برای اندازه‌گیری اثر کل پیش‌بینی‌کننده‌ها بر روی مدل هستند. بسته caret شامل یک کلاس کلی برای محاسبه یا برگرداندن این مقادیر است. تا زمان نگارش این مقاله، متدهایی برای 27 کلاس R وجود دارد، از جمله: C5. 0، JRip، PART، RRF، RandomForest، bagEarth، classbagg، cubist، dsa، earth، fda، gam، gbm، glm، glmnet، lm، multi. , mvr , nnet , pamrtrained , plsda , randomForest , regbagg , rfe , rpart , sbf و train.

برای نشان دادن، یک مدل جنگل تصادفی برای داده‌های بخش‌بندی سلولی مناسب بود:

*> کتابخانه (جنگل تصادفی)*

*> set. seed(791)*

*> rfImp <- randomForest(Class ~. , data = segTrain,*

*+ ntree = 2000،*

*+ اهمیت = درست)*

بسته randomForest حاوی تابعی به نام اهمیت است که معیار مربوطه را برمی گرداند. تابع varImp در بین مدل‌ها استاندارد می‌کند:

*> head(varImp(rfImp))*

PS WS

AngleCh1 -1. 002852 -1. 002852

AreaCh1 8. 769884 8. 769884

AvgIntenCh1 21. 460666 21. 460666

AvgIntenCh2 22. 377451 22. 377451

AvgIntenCh3 7. 690371 7. 690371

AvgIntenCh4 9. 108741 9. 108741

توجه داشته باشید که برخی از مدل‌ها برای هر کلاس نمره جداگانه‌ای دریافت می‌کنند در حالی که برخی دیگر این کار را نمی‌کنند.

هنگام استفاده از تابع train، تابع varImp کد مناسب را بر اساس مقدار آرگومان متد اجرا می‌کند. هنگامی که مدل دارای یک تابع داخلی برای اندازه‌گیری اهمیت نباشد، آموزش از رویکرد کلی‌تری (همانطور که در بالا توضیح داده شد) استفاده می‌کند.

تمرینات

برای داده‌های ریزش توضیح داده شده در تمرین [12. 3](#bookmark31) :

همبستگی بین پیش‌بینی‌کننده‌ها را محاسبه کنید. آیا رابطه قوی ­بین این متغیرها وجود دارد؟ این چگونه با آنچه که با این ویژگی‌ها انتظار می‌رود مقایسه می‌شود؟

اهمیت پیش‌بینی‌کننده‌های طبقه‌بندی شده (یعنی کد منطقه، طرح پست صوتی و غیره) را به صورت جداگانه با استفاده از مجموعه آموزشی ارزیابی کنید.

همچنین نمرات اهمیت پیش‌بینی‌کننده‌های پیوسته را به صورت جداگانه برآورد کنید.

اکنون از ReliefF برای تخمین مشترک اهمیت پیش‌بینی‌کننده‌ها استفاده کنید. آیا در رتبه‌بندی تفاوت وجود دارد؟ چرا و چرا نه؟

برای داده‌های نوع روغن شرح داده شده در تمرین [4. 4 ،](#bookmark7) نمرات اهمیت متغیر را برآورد کنید. ­آیا تعداد زیاد کلاس‌ها بر روند کمی‌سازی اهمیت‌ها تأثیر می‌گذارد؟

داده‌های UCI Abalone ( [http://archive. ics. uci. edu/ml/datasets/Abalone](http://archive.ics.uci.edu/ml/datasets/Abalone) ) از داده‌های 4177 آبالون تشکیل شده است. داده‌ها شامل اندازه‌گیری‌هایی از نوع (مرد، ماده و نوزاد)، طولانی‌ترین اندازه‌گیری پوسته، قطر، قد و چندین وزن (کل، بسته، احشاء و پوسته) است. نتیجه تعداد حلقه هاست. سن آبالون تعداد حلقه‌ها به اضافه 1. 5 است.

داده‌ها در بسته AppliedPredictiveModeling موجود است:

*> کتابخانه (AppliedPredictiveModeling)*

*> داده (abalone)*

*> str(abalone)*

|  |  |  |
| --- | --- | --- |
| data. frame': | 4177 obs. | از 9 متغیر: |
| نوع $ : | فاکتور w/ 3 | سطوح "F"، "I"، "M": 3 3 1 3 2 2 1 1 3 1 |
| LongestShell $: | عدد 0. 455 0 | 0. 35 0. 53 0. 44 0. 33 0. 425 0. 53 0. 545. . . |
| قطر دلار : | عدد 0. 365 0 | 0. 265 0. 42 0. 365 0. 255 0. 3 0. 415 0. 425. |
| $ ارتفاع : | شماره 0. 095 0 | 0. 09 0. 135 0. 125 0. 08 0. 095 0. 15 0. 125 |
| $ کل وزن: | شماره 0. 514 0 | 0. 226 0. 677 0. 516 0. 205. . . |
| $ ShuckedWeight: | شماره 0. 2245 | 0. 0995 0. 2565 0. 2155 0. 0895. . . |
| $ Viscera Weight: | عدد 0. 101 0 | 0. 0485 0. 1415 0. 114 0. 0395. . . |
| $ ShellWeight: | عدد 0. 15 0. | 07 0. 21 0. 155 0. 055 0. 12 0. 33 0. 26. . . |
| حلقه‌های $ : | int 15 7 9 | 10 7 8 20 16 9 19. . . |

*> سر (آبالون)*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | LongestShell Diameter Height WholeWeight ShuckedWeight را تایپ کنید | | | | | |
| 1 | م | 0. 455 | 0. 365 | 0. 095 | 0. 5140 | 0. 2245 |
| 2 | م | 0. 350 | 0. 265 | 0. 090 | 0. 2255 | 0. 0995 |
| 3 | اف | 0. 530 | 0. 420 | 0. 135 | 0. 6770 | 0. 2565 |
| 4 | م | 0. 440 | 0. 365 | 0. 125 | 0. 5160 | 0. 2155 |
| 5 | من | 0. 330 | 0. 255 | 0. 080 | 0. 2050 | 0. 0895 |
| 6  1  2  3  4  5  6 | من 0. 425 0. 300 0. 095  حلقه‌های VisceraWeight ShellWeight 0. 1010 0. 150 15  0. 0485 0. 070 7  0. 1415 0. 210 9  0. 1140 0. 155 10  0. 0395 0. 055 7  0. 0775 0. 120 8 | | | | 0. 3515 | 0. 1410 |

داده‌ها را برای ارزیابی روابط عملکردی بین پیش‌بینی‌کننده‌ها و نتیجه ترسیم کنید.

از نمودارهای پراکنده و نمودارهای همبستگی برای درک چگونگی ارتباط پیش‌بینی‌کننده‌ها با یکدیگر استفاده کنید.

امتیازهای اهمیت متغیر را برای هر پیش‌بینی تخمین بزنید. رویکردی برای تعیین مجموعه کاهش یافته از پیش‌بینی‌کننده‌های غیر زائد ایجاد کنید.

تحلیل مؤلفه‌های اصلی را برای پیش‌بینی‌کننده‌های پیوسته اعمال کنید تا مشخص کنید که چه تعداد از اطلاعات اساسی متمایز در داده‌ها وجود دارد. آیا استخراج ویژگی به این داده‌ها کمک می‌کند؟

فصل 19

مقدمه‌ای بر انتخاب ویژگی

تعیین اینکه کدام پیش‌بینی‌کننده‌ها باید در یک مدل گنجانده شوند، ­به یکی از مهم‌ترین سؤالات تبدیل می‌شود، زیرا داده‌ها به‌طور فزاینده‌ای ابعاد بالایی دارند. مثلا:

در تجارت، شرکت‌ها در حال حاضر در ذخیره و دسترسی به حجم زیادی از اطلاعات مشتریان و محصولات خود مهارت بیشتری دارند. پایگاه‌های داده بزرگ اغلب برای کشف روابط حیاتی استخراج می‌شوند [( Lo 2002](#bookmark1021) ).

در تحقیقات دارویی، شیمیدان‌ها می‌توانند هزاران عامل پیش‌بینی را ­با استفاده از روش روش کمی ساختار-فعالیت (QSAR) ­که در فصل‌های رگرسیون توضیح داده شده است، برای توصیف عددی جنبه‌های مختلف مولکول‌ها محاسبه کنند. به‌عنوان مثال، یک مجموعه نرم‌افزاری محبوب 17 طعم از سطح یک ترکیب را محاسبه می‌کند. این پیش‌بینی‌کننده‌ها می‌توانند مقوله‌ای یا پیوسته باشند و به راحتی به ده‌ها هزار عدد می‌رسند.

در زیست شناسی، مجموعه وسیعی از پیش‌بینی‌کننده‌های بیولوژیکی را می‌توان در یک زمان بر روی نمونه‌ای از مواد بیولوژیکی مانند خون اندازه‌گیری کرد. ریزآرایه‌های نمایانگر بیان RNA می‌توانند هزاران توالی RNA را به‌طور همزمان اندازه‌گیری کنند. همچنین، ریزآرایه‌های DNA و فناوری‌های توالی‌یابی می‌توانند ساختار ژنتیکی یک نمونه را به‌طور جامع تعیین کنند و تعداد زیادی پیش‌بینی عددی را تولید کنند. این فناوری‌ها در طول زمان به سرعت پیشرفت کرده‌اند و مقادیر بیشتری از اطلاعات را ارائه می‌دهند.

از نقطه نظر عملی، مدلی با پیش‌بینی‌کننده‌های کمتر ممکن است قابل تفسیرتر و کم‌هزینه‌تر باشد، به‌ویژه اگر برای اندازه‌گیری پیش‌بینی‌کننده‌ها هزینه وجود داشته باشد. از نظر آماری، برآورد ­پارامترهای کمتر اغلب جذاب تر است. همچنین، همانطور که به زودی خواهیم دید، برخی از مدل‌ها ممکن است تحت تاثیر منفی پیش‌بینی‌کننده‌های غیر اطلاعاتی قرار گیرند.

برخی از مدل‌ها به‌طور طبیعی در برابر پیش‌بینی‌کننده‌های غیر اطلاعاتی مقاوم هستند. برای مثال، مدل‌های مبتنی بر درخت و قانون، MARS و ریج، ­انتخاب ویژگی را به‌طور ذاتی هدایت می‌کنند. به‌عنوان مثال، اگر یک پیش‌بینی در هیچ شکافی در طول ساخت یک درخت استفاده نشود، معادله پیش‌بینی از نظر عملکردی مستقل از پیش‌بینی است.

\_19،

©

تمایز مهمی که در انتخاب ویژگی باید قائل شد، روش‌های *نظارت شده* و *بدون نظارت* است. هنگامی که در حین حذف عوامل پیش‌بینی، نتیجه نادیده گرفته می‌شود، این تکنیک ­*بدون نظارت است.* نمونه‌هایی از این فیلترها در فصل توضیح داده شد.  [3](#bookmark131) و شامل حذف پیش‌بینی‌کننده‌هایی است که همبستگی بالایی با سایر پیش‌بینی‌کننده‌ها دارند یا دارای توزیع‌های بسیار پراکنده و نامتعادل هستند (یعنی پیش‌بینی‌کننده‌های واریانس نزدیک به صفر). در هر مورد، نتیجه مستقل از محاسبات فیلتر است. برای روش‌ها*ی نظارت‌شده،* پیش‌بینی‌کننده‌ها به‌طور خاص به منظور افزایش دقت یا یافتن زیرمجموعه‌ای از پیش‌بینی‌کننده‌ها برای کاهش پیچیدگی مدل انتخاب می‌شوند. در اینجا، نتیجه معمولاً برای تعیین کمیت اهمیت پیش‌بینی‌کننده‌ها استفاده می‌شود (همانطور که در فصل نشان داده شده است.  [18 )](#bookmark822) .

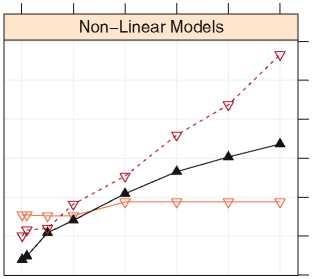
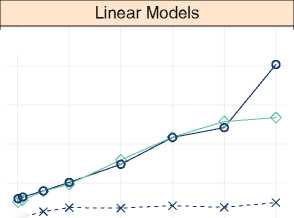
مسائل مربوط به هر نوع انتخاب ویژگی بسیار متفاوت است و ادبیات این موضوع بسیار زیاد است. بخش‌های بعدی چندین موضوع مهم از جمله نیاز به انتخاب ویژگی، رویکردهای معمولی و مسائل رایج را برجسته می‌کنند.

19. 1 پیامدهای استفاده از پیش‌بینی‌کننده‌های غیر اطلاعاتی

انتخاب ویژگی در درجه اول بر حذف پیش‌بینی‌کننده‌های غیر اطلاعاتی یا اضافی از مدل متمرکز است. مانند بسیاری از موضوعات مورد بحث در این متن، اهمیت انتخاب ویژگی بستگی به مدلی دارد که استفاده می‌شود. بسیاری از مدل‌ها، به‌ویژه مدل‌هایی که مبتنی بر شیب‌های رگرسیون و ­برش‌ها هستند، پارامترها را برای هر عبارت در مدل تخمین می‌زنند. به همین دلیل، وجود متغیرهای غیر اطلاعاتی می‌تواند عدم قطعیت را به پیش‌بینی‌کننده‌ها اضافه کند ­و اثربخشی کلی مدل را کاهش دهد.

حلالیت داده‌های QSAR که در فصل تحلیل شد. 6 - 9 و [20](#bookmark918) دوباره برای نشان دادن تأثیر پیش‌بینی‌کننده‌های غیر اطلاعاتی بر مدل‌های مختلف استفاده خواهد شد. داده‌ها به احتمال زیاد حاوی پیش‌بینی‌کننده‌های غیر اطلاعاتی هستند. با وجود این، پس از افزودن پیش‌بینی‌کننده‌های نامربوط، چندین مدل را تنظیم و بازسازی خواهیم کرد. برای انجام این کار، داده‌های پیش‌بینی اصلی با به هم ­زدن تصادفی ردیف‌هایشان (به‌طور مستقل برای هر ستون) تغییر می‌کنند. در نتیجه، نباید هیچ ارتباطی بین پیش‌بینی‌کننده‌های جدید و مقادیر حلالیت وجود داشته باشد. این روش ماهیت ­پیش‌بینی‌کننده‌های فردی (یعنی اثر انگشت با توزیع فرکانس یکسان) را حفظ می‌کند، اما اثر جانبی آن حذف همبستگی‌های بین پیش‌بینی است. اگر گنجاندن پیش‌بینی‌کننده‌های مرتبط آسیب بیشتری به یک مدل خاص اضافه کند، آن تأثیر در این تمرین منعکس نخواهد شد.

برای تعیین کمیت اثر پیش‌بینی‌کننده‌های اضافی، مدل‌هایی ایجاد شدند که از ۲۲۸ پیش‌بینی اصلی در داده‌ها استفاده می‌کردند و سپس با ۱۰، ۵۰، ۱۰۰، ۲۰۰، ۳۰۰، ۴۰۰ یا ۵۰۰ پیش‌بینی غیر اطلاعاتی اضافی تکمیل شدند. مدل‌ها تنظیم و نهایی شدند و مقادیر RMSE مجموعه آزمون محاسبه شد. مدل‌های ارزیابی شده در این روش رگرسیون خطی، جزئی بودند



0.7- \*

1.1 -

1.0-

0.9-

0.8-

<D CO to <D

LU

CO s *tr*

Trees

-1.1

-1.0

**~~X< X~~**

-0.9

-x x

-0.8

Linear Regression

Regression Tree

MARS

Random Forests

Partial Least Squares

Support Vector Machine lasso

Neural Networks

■a -0. 7

100 200 300 400 500

تعداد پیش‌بینی‌کننده‌های غیر اطلاعاتی اضافی

شکل 19. 1: مجموعه آزمایشی پروفایل‌های RMSE برای مدل‌های حلالیت زمانی که پیش‌بینی‌کننده‌های غیر اطلاعاتی حداقل مربعات، درخت‌های رگرسیون منفرد، خطوط رگرسیون تطبیقی چند متغیره، جنگل‌های تصادفی، شبکه‌های عصبی و ماشین‌های بردار پشتیبان تابع پایه شعاعی اضافه می‌شوند.

شکل [19. 1](#bookmark868) نتایج مجموعه تست را نشان می‌دهد. همانطور که انتظار می‌رفت، درختان رگرسیون و مدل‌های MARS به دلیل انتخاب ویژگی داخلی تحت تأثیر قرار نگرفتند. جنگل‌های ­رندوم کاهش متوسطی را در عملکرد نشان دادند. مسئله در اینجا این است که انتخاب تصادفی پیش‌بینی‌کننده‌ها برای تقسیم می‌تواند مدل را وادار کند که برخی از پیش‌بینی‌کننده‌های بی‌اهمیت را شامل شود. با این حال، گنجاندن آنها تأثیر جدی بر مدل کلی ندارد. مدل‌های ساختاری پارامعیار، مانند رگرسیون خطی، حداقل مربعات جزئی و شبکه‌های عصبی، بیشتر تحت تأثیر قرار گرفتند. به نظر می‌رسد شبکه‌های عصبی گسترده‌ترین مسائل را دارند، شاید به دلیل تعداد زیاد پارامترهای اضافه شده به مدل. به‌عنوان مثال، یک شبکه با 5 واحد پنهان در ابتدا 961 پارامتر رگرسیون را تخمین زده است. افزودن 500 پیش‌بینی نامربوط، تعداد پارامترها را به 3461 افزایش می‌دهد. با توجه به اینکه 951 نقطه داده در ­مجموعه آموزشی وجود دارد، ممکن است حل این مسأله بدون بیش برازش دشوار باشد. ماشین‌های بردار پشتیبان نیز افزایش قابل‌توجهی در RMSE داشتند که با نظرات ارائه شده توسط [هستی و همکاران ( 2008](#bookmark1018) ، فصل 12).

با توجه به تأثیر منفی احتمالی، نیاز به یافتن زیرمجموعه کوچکتری از پیش‌بینی‌کننده‌ها وجود دارد. هدف اصلی ما کاهش تعداد آنها به گونه‌ای است که عملکرد را به حداکثر برساند. این مشابه بحث‌های قبلی است: چگونه می‌توانیم پیچیدگی را بدون تأثیر منفی بر اثربخشی مدل کاهش دهیم؟

19. 2 رویکردهای کاهش تعداد پیش‌بینی ها

جدای از مدل‌هایی با انتخاب ویژگی داخلی، بیشتر رویکردها برای کاهش ­تعداد [پیش‌بینی‌کننده‌ها](#bookmark1019) را می‌توان در دو دسته اصلی قرار داد ( جان و همکاران [1994](#bookmark1019) ):

روش‌های *Wrapper چندین مدل را با استفاده از روش‌هایی که پیش‌بینی‌کننده‌ها را اضافه و/یا حذف می‌کنند، ارزیابی می‌کنند تا ترکیب بهینه‌ای را پیدا کنند که عملکرد مدل را به حداکثر می‌رساند.* در اصل، روش‌های wrapper الگوریتم‌های جستجویی هستند که با پیش‌بینی‌کننده‌ها به‌عنوان ورودی رفتار می‌کنند و از عملکرد مدل به‌عنوان خروجی برای بهینه‌سازی استفاده می‌کنند.

*روش‌های فیلتر،* ارتباط پیش‌بینی‌کننده‌ها را خارج از ­مدل‌های پیش‌بینی ارزیابی می‌کنند و متعاقباً فقط پیش‌بینی‌کننده‌هایی را مدل‌سازی می‌کنند که برخی از معیارها را پاس می‌کنند. به‌عنوان مثال، برای مسائل طبقه‌بندی، هر پیش‌بینی را می‌توان به صورت جداگانه ارزیابی کرد تا بررسی شود که آیا رابطه قابل قبولی بین آن و کلاس‌های مشاهده‌شده وجود دارد یا خیر. سپس فقط پیش‌بینی‌کننده‌هایی با روابط مهم در یک مدل طبقه‌بندی گنجانده می‌شوند.  [Saeys و همکاران ( 2007](#bookmark1025) ) روش‌های فیلتر نظرسنجی.

هر دو رویکرد مزایا و معایبی دارند. روش‌های فیلتر معمولاً از نظر محاسباتی کارآمدتر از روش‌های wrapper هستند، اما معیار انتخاب ارتباط مستقیمی با اثربخشی مدل ندارد. همچنین، اکثر روش‌های فیلتر، هر پیش‌بینی را به‌طور جداگانه ارزیابی می‌کنند و در ­نتیجه، پیش‌بینی‌کننده‌های اضافی (یعنی با همبستگی زیاد) ممکن است انتخاب شوند و برهمکنش‌های مهم بین متغیرها قابل تعیین کمی نباشند. نقطه ضعف روش wrapper این است که بسیاری از مدل‌ها ارزیابی می‌شوند (که ممکن است به تنظیم پارامتر نیز نیاز داشته باشند) و در نتیجه زمان محاسبه افزایش می‌یابد. همچنین خطر نصب بیش از حد با لفاف‌ها افزایش می‌یابد.

دو بخش زیر این روش‌ها را با جزئیات بیشتری تشریح می‌کند و از یک مطالعه موردی برای نشان دادن کاربرد آنها استفاده می‌شود.

19. 3 روش‌های بسته بندی

همانطور که قبلاً گفته شد، روش‌های wrapper جستجوی پیش‌بینی‌کننده‌ها را انجام می‌دهند تا تعیین کنند که وقتی وارد مدل می‌شوند، بهترین نتایج را ایجاد می‌کنند. یک مثال ساده، انتخاب رو به جلو کلاسیک برای رگرسیون خطی (الگوریتم [19. 1 )](#bookmark874) . در اینجا، پیش‌بینی‌کننده‌ها (یک در یک زمان) در مدل رگرسیون خطی فعلی ارزیابی می‌شوند. یک آزمون فرضیه آماری را می‌توان برای مشاهده اینکه آیا هر یک از پیش‌بینی‌کننده‌های جدید اضافه شده از نظر آماری معنی دار هستند (در آستانه از پیش تعریف شده) انجام داد. اگر حداقل یکی از پیش‌بینی‌کننده‌ها دارای *p-value* زیر آستانه ­قدیمی باشد، پیش‌بینی مرتبط با کوچک‌ترین مقدار به مدل اضافه می‌شود و فرآیند دوباره شروع می‌شود. الگوریتم زمانی متوقف می‌شود که هیچ یک از مقادیر *p* برای پیش‌بینی‌کننده‌های باقی مانده از نظر آماری معنی دار نباشد. در این طرح، رگرسیون خطی *پایه یادگیرنده* و انتخاب رو به جلو *روش جستجو است.* تابع *هدف،* کمیتی است که بهینه شده است که در این مورد، اهمیت آماری دارد که با *p-value نشان داده می‌شود.*

چند مسأله در این رویکرد وجود دارد:

روش جستجوی رو به جلو *حریصانه است،* به این معنی که ­راه‌حل‌های گذشته را دوباره بررسی نمی‌کند.

استفاده از آزمون‌های فرضیه مکرر به این شیوه، بسیاری از ویژگی‌های آماری آن‌ها را باطل می‌کند، زیرا همان داده‌ها بارها ارزیابی می‌شوند. شکل [19. 2](#bookmark875) برای یک تصویر غیر فنی از این موضوع.

به حداکثر رساندن اهمیت آماری ممکن است با به حداکثر رساندن کمیت‌های مبتنی بر دقت مرتبط‌تر یکسان نباشد.

1 یک مدل اولیه ایجاد کنید که فقط شامل یک عبارت عرض از مبدأ باشد.

2 repeat

برای *هر پیش‌بینی‌ای که در مدل فعلی* نیست

3

4

5

6

7

8

9

10

11

با افزودن پیش‌بینی به مدل فعلی، یک مدل کاندید ایجاد کنید

از آزمون فرضیه برای برآورد اهمیت آماری عبارت مدل جدید استفاده کنید

پایان

اگر *کوچکترین مقدار p کمتر از آستانه گنجاندن باشد،* مدل فعلی را به‌روزرسانی کنید تا عبارتی مطابق با مهم‌ترین پیش‌بینی آماری را شامل شود.

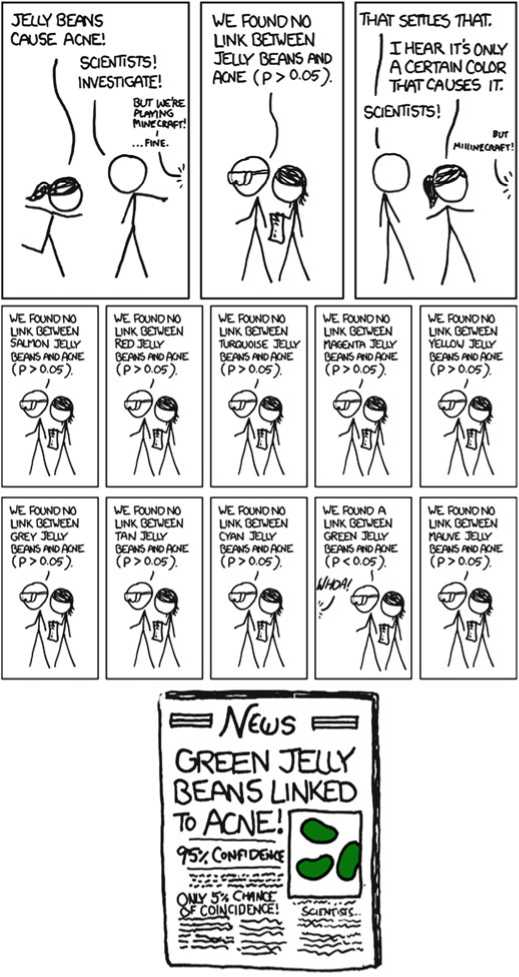
دیگر

| متوقف کردن

پایان

12 تا زمانی که *هیچ پیش‌بینی آماری معنی داری خارج از مدل باقی نماند*

الگوریتم 19. 1: انتخاب رو به جلو کلاسیک برای مدل‌های رگرسیون خطی



شکل 19. 2: خطرات آزمایش مکرر همان داده‌ها (رندال مونرو، [http://xkcd. com/882](http://xkcd.com/882) ، اصلاح شده برای فضا)

در مورد اول، رویه‌های جستجوی پیچیده‌تر ممکن است مؤثرتر باشند. چندین الگوریتم از این دست در زیر مورد بحث قرار گرفته است. موضوع دوم به‌طور ­گسترده مورد مطالعه قرار گرفته است [( درکسن و کسلمن 1992](#bookmark1014) ; [اولدن و جکسون 2000](#bookmark1023) ; [هارل 2001](#bookmark1018) ; [ویتینگهام و همکاران 2006](#bookmark1027) ). [[62]](#footnote-62) [هارل](#bookmark1018) [( 2001](#bookmark1018) ) استفاده از *p* -values را در انتخاب مدل خودکار خلاصه می‌کند:

*. .* اگر این روش به تازگی به‌عنوان یک روش آماری پیشنهاد شده بود، به احتمال زیاد رد می‌شد، زیرا این روش همه اصول تخمین آماری ­و آزمون فرضیه‌ها را نقض می‌کند.

مسئله دوم و سوم را می‌توان با استفاده از برخی معیارهای عملکرد پیش‌بینی مانند RMSE، دقت طبقه‌بندی یا خطای زیر منحنی ROC کاهش داد.

فرض کنید که RMSE به جای معنی‌داری آماری، تابع هدف باشد. در اینجا، الگوریتم یکسان خواهد بود اما ­عوامل پیش‌بینی را به مدل اضافه می‌کند که منجر به کوچک‌ترین مدل RMSE می‌شود. این فرآیند تا زمانی ادامه خواهد داشت که تعدادی از پیش‌بینی‌کننده‌های از پیش تعریف شده به دست آید یا از مدل کامل استفاده شود. از این فرآیند، RMSE را می‌توان برای تعیین نقطه‌ای که خطا شروع به افزایش کرد، نظارت کرد. اندازه زیر مجموعه ­مرتبط با کوچکترین RMSE انتخاب خواهد شد. همانطور که می‌توان حدس زد، مسأله اصلی در اینجا به دست آوردن تخمین‌های خوب از میزان خطا است که در معرض تطابق بیش از حد ناشی از مدل یا فرآیند انتخاب ویژگی نیست (به بخش [19. 5](#bookmark889) زیر).

چندین معیار دیگر وجود دارد که می‌تواند عملکرد را بر اساس تعداد پیش‌بینی در مدل جریمه کند. برای مثال، هنگام انتخاب بین دو مدل با RMSE یکسان، مدل ساده‌تر با پیش‌بینی‌کننده‌های کمتر ترجیح داده می‌شود. برای رگرسیون خطی، آماری که معمولاً مورد استفاده قرار می‌گیرد، *معیار اطلاعات آکایک* است، نسخه‌ای جریمه‌شده از مجموع مربعات خطا: که در آن *P* تعداد عبارت‌های مدل است. در اینجا، هیچ گزاره استنباطی درباره این کمیت بیان نمی‌شود و با وجود یکسان بودن همه چیزهای دیگر، سادگی ­بر پیچیدگی ترجیح داده می‌شود. آمار مشابه *AIC* برای مدل‌های دیگر مانند رگرسیون لجستیک نیز موجود است. جستجوی فضای پیش‌بینی برای مدلی با کوچک‌ترین *AI C* ممکن است به جلوگیری از برازش بیش از حد کمک کند. [[63]](#footnote-63)

AIC = *n* log

(£(*y. - y.* )2)

+2*P*

رویکرد دیگر انتخاب ویژگی مبتنی بر همبستگی است [( هال و اسمیت](#bookmark1017) [1997](#bookmark1017) ) که تلاش می‌کند بهترین زیرمجموعه پیش‌بینی‌کننده‌ها را بیابد که همبستگی قوی با نتیجه اما همبستگی بین پیش‌بینی ضعیف دارند. برای انجام این کار، یک معیار ممکن است

*P P Ry*

*G — '/ — =*

*P +* + *P ( P -* 1) *R x*

که در آن *Ry* معیاری از همبستگی بین پیش‌بینی نامزد و نتیجه است و *Rx میانگین همبستگی بین پیش‌بینی* فعلی و پیش‌بینی‌کننده‌هایی است که قبلاً در زیر مجموعه گنجانده شده‌اند. در هر مرحله از جستجو، این معیار می‌تواند پیش‌بینی‌کننده‌ها را به‌عنوان تابعی از اثربخشی و افزونگی رتبه‌بندی کند.

برای مدل‌هایی که *برای پیش‌بینی،* به جای توضیح ساخته شده‌اند، دو نکته کلی وجود دارد:

بسیاری از انتقادات به روش‌های لفاف که در بالا ذکر شد[[64]](#footnote-64) تمرکز بر استفاده از آزمون‌های فرضیه‌های آماری.

مانند بسیاری از موارد دیگر ارائه شده در این متن، روش‌شناسی مبتنی بر ­اصول آماری دوگانه ممکن است همچنان به مدل‌های بسیار دقیق منجر شود. حفاظت کلیدی در این موارد یک فرآیند اعتبارسنجی روشمند و کامل با داده‌های مستقل است.

زیربخش‌های زیر روش‌های مختلف جستجو را برای استفاده با روش‌های wrapper توضیح می‌دهند.

انتخاب رو به جلو، عقب و گام به گام

انتخاب رو به جلو کلاسیک قبلاً در الگوریتم توضیح داده شده بود [19. 1 .](#bookmark874)  *انتخاب گام به گام* یک اصلاح رایج است که در آن، پس از اضافه شدن هر متغیر کاندید ­به مدل، هر عبارت برای حذف از مدل مجددا ارزیابی می‌شود. در برخی موارد، آستانه *p-value* برای افزودن و حذف ­پیش‌دگرها می‌تواند کاملاً متفاوت باشد [( درکسن و کزلمن، 1992](#bookmark1014) ). اگرچه این امر باعث می‌شود روند جستجو کمتر حریص باشد، اما مسأله ­آزمایش مجدد فرضیه‌ها را تشدید می‌کند. در *انتخاب معکوس،* مدل اولیه شامل همه پیش‌بینی‌کننده‌ها است که سپس به‌طور مکرر حذف می‌شوند تا مشخص شود کدامیک به‌طور قابل‌توجهی به مدل کمک نمی‌کنند. این رویه‌ها را می‌توان با استفاده از معیارهای غیر استنتاجی، مانند آماره *AIC،* برای افزودن یا حذف پیش‌بینی‌کننده‌ها از مدل، بهبود بخشید.

گیون و همکاران [( 2002](#bookmark1017) ) یک الگوریتم انتخاب به عقب (به نام *حذف ویژگی بازگشتی* ) را توصیف کرد که از نصب مجدد بسیاری از مدل‌ها در هر مرحله از جستجو جلوگیری می‌کند. هنگامی که مدل کامل ایجاد می‌شود، معیاری از اهمیت متغیر ­محاسبه می‌شود که پیش‌بینی‌کننده‌ها را از مهم‌ترین به حداقل رتبه‌بندی می‌کند. محاسبات اهمیت می‌تواند مبتنی بر مدل باشد (به‌عنوان مثال، معیار اهمیت تصادفی جنگل ­) یا با استفاده از یک رویکرد کلی تر که مستقل از مدل کامل است. در هر مرحله از جستجو، کم اهمیت‌ترین پیش‌بینی‌کننده‌ها هستند

|  |  |
| --- | --- |
| 1 با استفاده از همه پیش‌بینی‌کننده‌های *P،* مدل را روی مجموعه آموزشی تنظیم/آموزش دهید | |
| 2 محاسبه عملکرد مدل | |
| 3 اهمیت یا رتبه‌بندی متغیرها را محاسبه کنید | |
| 4 برای *هر زیر مجموعه اندازه S i، i* = 1. *. . S* انجام دهید | |
| 5 | مهم‌ترین متغیرهای *S i* را حفظ کنید |
| 6 | [اختیاری] داده‌ها را از قبل پردازش کنید |
| 7 | پیش‌بینی‌کننده‌های *S i،* مدل را روی مجموعه آموزشی تنظیم/آموزش دهید |
| 8 | محاسبه عملکرد مدل |
| 9 | [اختیاری] رتبه‌بندی هر پیش‌بینی را دوباره محاسبه کنید |
| 10 پایان | |
| 11 مشخصات عملکرد را روی *S i محاسبه کنید* | |
| 12 تعداد مناسب پیش‌بینی‌کننده‌ها را تعیین کنید (یعنی *S i* | |
| مرتبط با بهترین عملکرد) | |
| 13 مدل نهایی را بر اساس *S i بهینه برازش دهید* | |

الگوریتم 19. 2: انتخاب به عقب از طریق الگوریتم حذف ویژگی بازگشتی­ [گیون و همکاران ( 2002](#bookmark1017) )

به‌طور مکرر قبل از بازسازی مدل حذف شده است. مانند قبل، هنگامی که یک مدل جدید ایجاد می‌شود، تابع هدف برای آن مدل تخمین زده می‌شود. این ­فرآیند برای برخی از دنباله‌های از پیش تعریف شده ادامه می‌یابد و اندازه زیر مجموعه مربوط ­به بهترین مقدار تابع هدف به‌عنوان مدل نهایی استفاده می‌شود. این فرآیند با جزئیات بیشتر در الگوریتم توضیح داده شده است [19. 2](#bookmark880) و در بخش نشان داده شده است.  [19. 6 .](#bookmark46)

در حالی که به راحتی می‌توان الگوریتم RFE را به‌عنوان یک جعبه سیاه در نظر گرفت، ملاحظاتی وجود دارد که باید در نظر گرفته شوند. به‌عنوان مثال، هنگامی که نتیجه بیش از دو کلاس دارد، برخی از کلاس‌ها ممکن است درجه زیادی از جدایی از بقیه مجموعه آموزشی داشته باشند. به این ترتیب، دستیابی به نرخ خطای کمتری برای این کلاس‌ها نسبت به سایرین آسان‌تر است. وقتی پیش‌بینی‌کننده‌ها برای انتخاب رتبه‌بندی می‌شوند، پیش‌بینی‌کننده‌های مرتبط با کلاس‌های «آسان» ممکن است موقعیت‌های بالاترین رتبه‌ها را اشباع کنند. در نتیجه کلاس‌های دشوار نادیده گرفته می‌شوند و میزان خطای بالایی را حفظ می‌کنند. در این مورد، امتیازهای اهمیت ویژه کلاس می‌تواند به انتخاب مجموعه متعادل‌تری از پیش‌بینی‌کننده‌ها در تلاش برای متعادل کردن میزان خطا در همه کلاس‌ها کمک کند.

بازپخت شبیه‌سازی شده

تعداد زیادی از روش‌های جستجوی مدرن وجود دارد که می‌تواند برای مسأله انتخاب ویژگی اعمال شود. *بازپخت شبیه‌سازی شده* [( بوهاچفسکی و همکاران 1986](#bookmark1011) ) فرآیند خنک‌سازی فلز را تقلید می‌کند. الگوریتم [19. 3](#bookmark881) این فرآیند را به تفصیل شرح می‌دهد. یک زیر مجموعه اولیه از پیش‌بینی‌کننده‌ها انتخاب شده و برای تخمین استفاده می‌شود

1 یک زیرمجموعه تصادفی اولیه از پیش‌بینی‌کننده‌ها ایجاد کنید

2 برای *تکرار i* = 1. *. . t* do

به‌طور تصادفی مجموعه بهترین پیش‌بینی فعلی را مختل کنید

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

[اختیاری] داده‌ها را از قبل پردازش کنید

با استفاده از این مجموعه پیش‌بینی، مدل را تنظیم/آموزش دهید

محاسبه عملکرد مدل ( *E i* )

اگر *E i < E بهترین* است

مجموعه پیش‌بینی فعلی را به‌عنوان بهترین بپذیرید

مجموعه E بهترین = E i

دیگر

احتمال پذیرش پیش‌بینی فعلی را محاسبه کنید

مجموعه  *p i a* = exp [( *E best -E i* ) */T* ]

یک عدد تصادفی *U* بین [0،1] ایجاد کنید، اگر *p a <U* سپس

مجموعه پیش‌بینی فعلی را به‌عنوان بهترین بپذیرید

مجموعه E بهترین = E i

دیگر

| پایان مجموعه بهترین پیش‌بینی فعلی را حفظ کنید

پایان

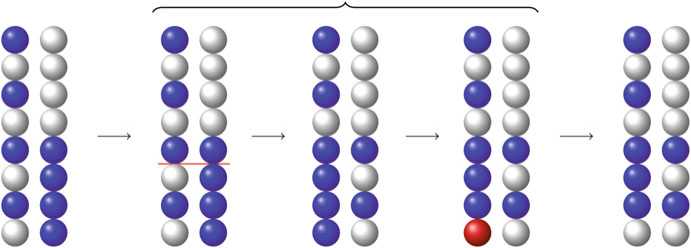
20 پایان

21 مجموعه پیش‌بینی مرتبط با کوچکترین *E i* را در تمام تکرارها تعیین کنید

22 مدل را با این مجموعه پیش‌بینی نهایی کنید

الگوریتم 19. 3: بازپخت شبیه‌سازی شده برای انتخاب ویژگی. *E* معیاری از عملکرد است که در آن مقادیر کوچک بهترین هستند و *T* یک ­مقدار دمایی است که در طول تکرار تغییر می‌کند

عملکرد مدل (در اینجا به‌عنوان *E* 1 نشان داده شده است، برای میزان خطای اولیه). زیرمجموعه پیش‌بینی فعلی کمی تغییر کرده و مدل دیگری با نرخ خطای تخمینی *E* 2 ایجاد می‌شود. اگر مدل جدید نسبت به مدل قبلی (یعنی *E* 2 *< E* 1 ) بهبود یافته باشد، مجموعه ویژگی‌های جدید پذیرفته می‌شود. با این حال، اگر بدتر باشد، همچنان ممکن است بر اساس احتمال *p i a که i* تکرار فرآیند است، پذیرفته شود. این احتمال به گونه‌ای تنظیم شده است که در طول زمان کاهش یابد، به‌طوری که با بزرگ شدن *i،* احتمال پذیرفته شدن پیکربندی کمتر از حد مطلوب بسیار بعید است. این فرآیند برای تعدادی از تکرارهای از پیش تعیین شده و بهترین زیر مجموعه متغیر در تمام تکرارها ادامه می‌یابد.



متقاطع متقاطع جهش

محل

شکل 19. 3: شماتیک فاز تولید مثل یک الگوریتم ژنتیک

استفاده می‌شود. ایده این است که از یک بهینه محلی اجتناب کنید (راه حلی که در حال حاضر بهترین است اما در کل بهترین نیست). با پذیرش راه‌حل‌های «بد»، الگوریتم می‌تواند جستجو را در فضاهای دیگر ادامه دهد و بنابراین حریص کمتری دارد.

*الگوریتم‌های ژنتیک*

*الگوریتم‌های* ژنتیک (GAs) [( هلند )](#bookmark1018) یکی از نزدیکان بازپخت شبیه‌سازی شده [1975](#bookmark1018) ; [گلدبرگ 1989](#bookmark1017) ). این ابزار بهینه‌سازی بر اساس اصول تکاملی ­زیست شناسی جمعیت است و نشان داده شده است که در یافتن راه حل‌های بهینه توابع پیچیده و چند متغیره موثر است. به‌طور خاص، GAها برای تقلید از فرآیند تکاملی ساخته شده‌اند که با اجازه دادن به جمعیت فعلی ­راه‌حل‌ها برای تولید مثل، کودکانی را تولید می‌کنند که برای زنده ماندن با هم رقابت می‌کنند. سپس به مناسب‌ترین بازماندگان اجازه تولید مثل داده می‌شود و نسل بعدی کودکان ایجاد می‌شود. با گذشت زمان، نسل‌ها به یک فلات برازش اندام همگرا می‌شوند [( هلند 1992](#bookmark1018) ) و یک راه حل بهینه می‌تواند انتخاب شود.

همانطور که تا کنون دیده‌ایم، مسئله انتخاب ویژگی ذاتاً یک مسئله بهینه‌سازی پیچیده است، جایی که ما به دنبال ترکیبی از ویژگی‌هایی هستیم که پیش‌بینی بهینه پاسخ را فراهم می‌کند. برای به کارگیری GA در این راستا، باید مسئله انتخاب ویژگی را بر حسب ماشین آلات GA چارچوب‌بندی کنیم. سوژه‌های این ماشین کروموزوم‌هایی هستند که از ژن تشکیل شده و بر اساس برازش آن‌ها ارزیابی می‌شوند. برای ایجاد نسل بعدی فرزندان، دو کروموزوم از طریق فرآیند متقاطع و جهش تولید مثل می‌کنند که در شکل 1 نشان داده شده است.  [19. 3](#bookmark882) . نشان داده شده است که GAها ابزارهای موثر انتخاب ویژگی در زمینه‌های شیمی سنجی هستند [( Lavine et al.](#bookmark1020)  [2002](#bookmark1020) )، تحلیل تصویر [( Bhanu and Lin 2003](#bookmark1011) ) و امور مالی [( Min et al. 2006](#bookmark1022) ; [Huang et al. 2012](#bookmark1019) ).

معیارهای توقف، تعداد فرزندان برای هر نسل ( *GenSize* ) و احتمال جهش ( *p m* ) را تعریف کنید.

یک مجموعه تصادفی اولیه از *m* کروموزوم‌های دوتایی، هر کدام به طول *p را تولید کنید*

تکرار

برای *هر کروموزوم* انجام دهید

4

5

6

7

8

9

10

11

یک مدل را تنظیم و آموزش دهید و برازش هر کروموزوم را محاسبه کنید

پایان

برای *تولید مثل k* =1. *. . GenSize/* 2 do

دو کروموزوم را بر اساس معیار برازش انتخاب کنید

متقاطع: به‌طور تصادفی یک مکان را انتخاب کنید و ژن‌های هر کروموزوم را فراتر از جایگاه‌ها مبادله کنید

جهش: تغییر تصادفی مقادیر دوتایی هر ژن در هر کروموزوم فرزند جدید با احتمال *p m*

پایان

12 تا زمانی *که معیارهای توقف برآورده شود*

الگوریتم 19. 4: الگوریتم ژنتیک برای انتخاب ویژگی

در زمینه انتخاب ویژگی، کروموزوم یک بردار دوتایی است که طول آن برابر با تعداد پیش‌بینی‌کننده‌های مجموعه داده است. هر ورودی دوتایی کروموزوم یا ژن، نشان دهنده وجود یا عدم وجود هر پیش‌بینی در داده‌ها است. برازش کروموزوم توسط مدل با استفاده از پیش‌بینی‌کننده‌های نشان داده شده توسط بردار باینری تعیین می‌شود. بنابراین، GAها وظیفه یافتن راه‌حل‌های بهینه را از 2 *n* ترکیب ممکن از مجموعه‌های پیش‌بین دارند.

برای شروع فرآیند جستجو، GA‌ها اغلب با انتخاب تصادفی کروموزوم‌ها از جمعیت همه کروموزوم‌های ممکن آغاز می‌شوند. برازش هر کروموزوم محاسبه می‌شود که احتمال انتخاب کروموزوم را برای فرآیند تولید مثل تعیین می‌کند. سپس دو کروموزوم از جمعیت فعلی بر اساس معیار برازش اندام انتخاب می‌شوند و اجازه تولید مثل دارند. در مرحله تولید مثل، دو کروموزوم والد در یک موقعیت تصادفی تقسیم می‌شوند (که جایگاه‌های دیگر نیز نامیده *می‌شود* ) و سر یک کروموزوم با دم کروموزوم دیگر ترکیب می‌شود و بالعکس. پس از تقاطع، ورودی‌های جداگانه کروموزوم‌های جدید را می‌توان به‌طور تصادفی برای جهش انتخاب کرد که در آن مقدار باینری فعلی به مقدار دیگر تغییر می‌کند. الگوریتم [19. 4](#bookmark883) این مراحل را فهرست می‌کند.

فاز متقاطع نسل‌های بعدی را به سمت بهینه‌ها در زیرفضاهای مواد ژنتیکی مشابه سوق می‌دهد. به عبارت دیگر، فضای فرعی جستجو به فضایی که توسط مناسب‌ترین کروموزوم‌ها تعریف شده است، باریک می‌شود. این بدان معنی است که الگوریتم می‌تواند در یک بهینه محلی به دام بیفتد. در زمینه انتخاب ویژگی، این بدان معنی است که ویژگی‌های انتخاب شده ممکن است ­یک مدل بهینه تولید کنند، اما سایر زیر مجموعه‌های ویژگی بهینه تر ممکن است وجود داشته باشند. فاز جهش الگوریتم را قادر می‌سازد تا با ایجاد اختلال در ­ماده ژنتیکی، از بهینه‌های محلی فرار کند. معمولاً احتمال جهش پایین نگه داشته می‌شود (مثلاً *p m <* 0. 05). با این حال، اگر تمرین‌کننده نگران بهینه‌های محلی باشد، احتمال جهش را می‌توان افزایش داد. اثر افزایش احتمال جهش کاهش همگرایی به یک راه حل بهینه است.

روش‌های فیلتر

همانطور که قبلاً گفته شد، روش‌های فیلتر پیش‌بینی‌کننده‌ها را قبل از آموزش ­مدل ارزیابی می‌کنند و بر اساس این ارزیابی، زیرمجموعه‌ای از پیش‌بینی‌کننده‌ها در ­مدل وارد می‌شوند. از آنجایی که امتیازدهی پیش‌بینی‌کننده‌ها با مدل قطع شده است، بسیاری از معیارهای اهمیت متغیر در فصل مورد بحث قرار گرفتند.  [18](#bookmark822) برای فیلتر کردن متغیرها مناسب است. بیشتر این تکنیک‌ها تک ­متغیره هستند، به این معنی که هر پیش‌بینی را به صورت مجزا ارزیابی می‌کنند. در این حالت، وجود پیش‌بینی‌کننده‌های همبسته، انتخاب پیش‌بینی‌کننده‌های مهم، اما اضافی را ممکن می‌سازد. پیامدهای آشکار این موضوع این است که پیش‌بینی‌کننده‌های زیادی انتخاب می‌شوند و در نتیجه مسائل هم خطی ایجاد می‌شود. [گیون و الیسیف](#bookmark1017) [( 2003](#bookmark1017) ) چندین جنبه از افزونگی پیش‌بینی در طول فیلتر کردن را مورد بحث قرار می‌دهد.

همچنین، اگر از آزمون‌های فرضیه برای تعیین اینکه کدام پیش‌بینی‌کننده‌ها رابطه آماری معنی‌داری با نتیجه دارند (مانند آزمون *t* ) استفاده شود، مسأله *تعدد* می‌تواند رخ دهد (نگاه کنید به [وست فال و یانگ](#bookmark1027) [( 1993](#bookmark1027) ) و شکل.  [19. 2 )](#bookmark875) . برای مثال، اگر سطح اطمینان *a =* 0 *باشد.* 05 به‌عنوان یک آستانه *p-value* برای معنی‌داری استفاده می‌شود، هر آزمون فردی دارای نرخ نظری مثبت کاذب 5٪ است. با این حال، هنگامی که تعداد زیادی آزمون آماری همزمان انجام می‌شود، احتمال مثبت کاذب کلی به‌طور تصاعدی افزایش می‌یابد. برای توضیح این موضوع، رویه‌های تعدیل ­*p -value می‌توانند* نرخ مثبت کاذب را کنترل کنند. *تصحیح* بونفرونی [( بلند و آلتمن 1995](#bookmark1011) ) یکی از این روش‌ها است. اگر از قطع *p-value a* برای تعریف معنی‌داری آماری برای هر یک از آزمون‌های *M* استفاده شود، استفاده از یک برش جایگزین *a/M* شدت را افزایش می‌دهد و به کنترل احتمال نتایج مثبت کاذب کمک می‌کند. با این حال، این روش می‌تواند بسیار محافظه کارانه باشد و تعداد نتایج مثبت واقعی را محدود کند. سوسک‌های دیگر برای مقابله با [چندگانگی](#bookmark1027) را می‌توان در Westfall and Young [( 1993](#bookmark1027) ) یافت. همچنین، Ahdesmaki و Strimmer ( 2010 ) یک آزمون *t* اصلاح شده را پیشنهاد می‌کنند که تعداد زیادی از آزمون‌های انجام شده و همچنین همبستگی‌های بین پیش‌بینی را به حساب می‌آورد.

در حالی که روش‌های فیلتر ساده و سریع هستند، ماهیت موضوعی برای این روش وجود دارد. اکثر روش‌های امتیازدهی نقطه برش آشکاری ندارند تا مشخص کنند کدام پیش‌بینی به اندازه کافی برای وارد شدن به مدل مهم است. حتی در مورد آزمون‌های فرضیه‌های آماری، کاربر همچنان باید ­سطح اطمینان را برای اعمال در نتایج انتخاب کند. در عمل، یافتن یک مقدار مناسب برای مقدار اطمینان *a* ممکن است نیاز به چندین ارزیابی داشته باشد تا زمانی که عملکرد قابل قبولی حاصل شود.

بایاس انتخاب

در حالی که برخی از روش‌های فیلترینگ یا روش‌های جستجو مؤثرتر از سایرین هستند، سؤال مهم‌تر به نحوه محاسبه عملکرد مدل مربوط می‌شود (مخصوصاً وقتی حجم نمونه کوچک است). بیش برازش ­پیش‌دیگرها با داده‌های آموزشی ممکن است رخ دهد و بدون اعتبارسنجی مناسب، ممکن است مورد توجه قرار نگیرد. مثلا، [گیون و همکاران ( 2002](#bookmark1017) ) حذف ویژگی بازگشتی را با مدل‌های طبقه‌بندی ماشین بردار پشتیبان برای مجموعه داده‌های ریزآرایه سرطان روده بزرگ نشان داد. در این داده‌ها، اندازه‌گیری روی 2000 توالی RNA منحصر به فرد به‌عنوان پیش‌بینی وضعیت بیماری (سرطان یا نرمال) در مجموعه‌ای از 62 بیمار مورد استفاده قرار گرفت. هیچ مجموعه آزمایشی برای تأیید نتایج استفاده نشد. برای نظارت بر عملکرد، اعتبار سنجی متقاطع ترک یک خروجی برای هر مدل استفاده شد (در Line [8](#bookmark880) در الگوریتم [19. 2 )](#bookmark880) . تحلیل آنها نشان داد که مدل SVM می‌تواند به دقت بیش از 95 درصد تنها با استفاده از 4 پیش‌بینی و دقت 100 درصد با مدل‌هایی با استفاده از 8-256 پیش‌بینی دست یابد.

*پس* از انتخاب ویژگی‌ها، نرخ‌های خطای ترک بر اساس مدل SVM بود. می‌توان تصور کرد که اگر انتخاب ویژگی با مجموعه داده کمی متفاوت تکرار شود، نتایج ممکن است تغییر کند. معلوم می‌شود که در برخی موارد، عدم قطعیت ناشی از انتخاب ویژگی می‌تواند بسیار بزرگتر از عدم قطعیت مدل (پس از انتخاب ویژگی ها) باشد. برای نشان دادن این موضوع، Ambroise و McLachlan ( 2002 ) همان روش RFE را با مجموعه داده‌های یکسانی انجام دادند، اما برچسب‌های کلاس را به هم زدند (برای وادار کردن همه پیش‌بینی‌کنندگان به غیر اطلاعاتی بودن). آنها نشان دادند که استراتژی اعتبار سنجی متقاطع ترک یکی از استفاده شده توسط [گیون و همکاران](#bookmark1017) [( 2002](#bookmark1017) ) حتی زمانی که پیش‌بینی‌کننده‌ها کاملاً غیر اطلاعاتی هستند، به خطاهای صفر دست می‌یابند.

خطای منطقی در رویکرد اصلی واضح است. یک مدل از مجموعه آموزشی ایجاد شد و با استفاده از این داده‌ها، پیش‌بینی‌کننده‌ها ارزیابی و رتبه‌بندی شدند. اگر مدل فقط با استفاده از پیش‌بینی‌کننده‌های مهم دوباره تنظیم شود، عملکرد تقریباً در همان مجموعه داده بهبود می‌یابد. علاوه بر این، نسبت *P* به *n* بسیار زیاد است (2000:62) که به‌طور محسوسی احتمال مهم اعلام شدن یک پیش‌بینی کاملاً نامربوط را به‌طور اتفاقی افزایش می‌دهد.

خطای روش‌شناختی به این دلیل رخ داد که انتخاب ویژگی به‌عنوان بخشی از فرآیند ساخت مدل در نظر گرفته نشد. به این ترتیب، باید در ­روش نمونه‌گیری مجدد گنجانده شود تا تغییرات انتخاب ویژگی ­در نتایج ثبت شود. رویه اعتبارسنجی متقاطع یک خروجی روی خط [8](#bookmark880) در الگوریتم [19. 2](#bookmark880) از مراحل خارج از فرآیند آموزش مدلی که اندازه‌گیری می‌کند بی اطلاع است. برای مثال، حتی اگر فرآیند انتخاب ویژگی پس از ارزیابی تعداد زیادی از مدل‌های کاندید به مدل واقعی برسد، تخمین‌های عملکرد باید منعکس‌کننده فرآیندی باشد که منجر به نتیجه شده است.

برای نمونه‌برداری مجدد مناسب از فرآیند انتخاب ویژگی، یک حلقه نمونه‌برداری مجدد «بیرونی» مورد نیاز است که کل فرآیند را در بر می‌گیرد. الگوریتم [19. 5](#bookmark46) چنین طرح نمونه‌گیری مجدد را برای حذف ویژگی بازگشتی نشان می‌دهد. در لاین [1 ،](#bookmark46) نمونه برداری مجدد به گونه‌ای اعمال می‌شود که کل فرآیند انتخاب ویژگی در داخل باشد. مثلا،

اگر اعتبارسنجی متقاطع 10 برابری در این حلقه اولیه وجود داشت، 90٪ از داده‌ها برای انجام انتخاب ویژگی و 10٪ نگهدارنده برای ارزیابی عملکرد استفاده می‌شود (به‌عنوان مثال، Line [10 )](#bookmark46) برای هر زیر مجموعه از پیش‌بینی ها. کل فرآیند انتخاب ویژگی 9 بار دیگر با مجموعه‌ای متفاوت از نمونه‌های نگهدارنده انجام می‌شود. در پایان، این ده مجموعه نگهدارنده تعداد بهینه پیش‌بینی‌کننده‌ها را در مدل نهایی تعیین می‌کنند (خط [14 )](#bookmark46) . با توجه به این نتیجه، کل مجموعه آموزشی برای رتبه‌بندی پیش‌بینی‌کننده‌ها و آموزش مدل نهایی استفاده می‌شود (خطوط [16](#bookmark46) و [به ترتیب 17](#bookmark46) ). توجه داشته باشید که ممکن است برای بهینه‌سازی هر پارامتر تنظیم در مدل، یک لایه "داخلی" اضافی از نمونه برداری مجدد مورد نیاز باشد (خطوط [3](#bookmark46) و [9 )](#bookmark46) .

Ambroise و McLachlan ( 2002 ) همچنین نشان دادند که وقتی روش‌های نمونه‌گیری مجدد از راه‌انداز، اعتبارسنجی متقاطع 10 برابری یا روش‌های نمونه‌گیری مجدد مجموعه آزمایشی مکرر به درستی استفاده شد، نتایج مدل به درستی در حدود مقدار نرخ بدون اطلاعات تعیین شد.

لایه نمونه‌گیری مجدد اضافی می‌تواند تأثیر منفی قابل‌توجهی بر کارایی محاسباتی فرآیند انتخاب ویژگی داشته باشد. با این حال، به خصوص با ست‌های آموزشی کوچک، این فرآیند به شدت شانس بیش برازش پیش‌بینی‌کننده‌ها را کاهش می‌دهد.

نکته مهم این است که گاهی اوقات مرتکب خطا در اعتبارسنجی ­نتایج روش‌های انتخاب ویژگی آسان است. مثلا، [کاستالدی و همکاران](#bookmark1013) [( 2011](#bookmark1013) ) بررسی مقالاتی را انجام داد که از تکنیک‌های طبقه‌بندی با داده‌های بیولوژیکی (مثلاً ریزآرایه‌های RNA، پروتئین‌ها) استفاده می‌کردند و دریافتند که 64 درصد از تحلیل‌ها به‌طور مناسب فرآیند انتخاب ویژگی را تأیید نمی‌کنند. آنها همچنین نشان دادند که این تحلیل‌ها بین تخمین‌های عملکرد نمونه‌گیری مجدد و محاسبه‌شده از یک مجموعه آزمون مستقل تفاوت معنی‌داری داشتند (نتایج مجموعه آزمون بدبینانه‌تر بود).

خطر بیش برازش در این روش به انتخاب ویژگی‌های بازگشتی یا به‌طور کلی بسته بندی‌ها محدود نمی‌شود. هنگام استفاده از روش‌های جستجو یا فیلترهای دیگر برای کاهش تعداد پیش‌بینی‌کننده‌ها، همچنان خطر وجود دارد.

شرایط زیر احتمال بایاس انتخاب را افزایش می‌دهد:

مجموعه داده‌ها کوچک است.

تعداد پیش‌بینی‌کننده‌ها زیاد است (از آنجایی که احتمال اینکه یک پیش‌بینی غیر اطلاعاتی به اشتباه به‌عنوان مهم اعلام شود، افزایش می‌یابد).

مدل پیش‌بینی قدرتمند است (به‌عنوان مثال، مدل‌های جعبه سیاه) که به احتمال زیاد بیش از حد داده‌ها برازش دارند.

هیچ مجموعه تست مستقلی در دسترس نیست.

وقتی مجموعه داده‌ها بزرگ است، مجموعه‌های داده جداگانه را برای انتخاب ویژگی‌ها، تنظیم مدل‌ها و اعتبارسنجی مدل نهایی (و مجموعه ویژگی‌ها) توصیه می‌کنیم. برای مجموعه‌های آموزشی کوچک، نمونه برداری مجدد مناسب بسیار مهم است. اگر مقدار داده‌ها خیلی کم نیست، توصیه می‌کنیم یک مجموعه آزمایشی کوچک را کنار بگذارید تا دو بار بررسی کنید که خطای فاحشی مرتکب نشده باشد.

1 برای *هر تکرار نمونه برداری مجدد* انجام دهید

داده‌ها را از طریق نمونه‌گیری مجدد به آموزش و مجموعه تست/بازداشت تقسیم کنید

2

3

4

5

6

7

8

9

10

11

12

پیش‌بینی‌کننده‌های *P،* مدل را روی مجموعه آموزشی تنظیم/آموزش دهید

محاسبه عملکرد مدل

اهمیت یا رتبه‌بندی متغیرها را محاسبه کنید

برای *هر زیر مجموعه اندازه S i , i* = 1. *. . S* انجام دهید

مهم‌ترین متغیرهای *S i* را حفظ کنید

[اختیاری] داده‌ها را از قبل پردازش کنید

پیش‌بینی‌کننده‌های *S i،* مدل را روی مجموعه آموزشی تنظیم/آموزش دهید

محاسبه عملکرد مدل با استفاده از نمونه‌های نگهداری شده [اختیاری] محاسبه مجدد رتبه‌بندی برای هر پایان پیش‌بینی

13 پایان

مشخصات عملکرد را بر روی *S i* با استفاده از نمونه‌های نگه‌داشته‌شده محاسبه کنید

تعداد مناسب پیش‌بینی‌کننده‌ها را تعیین کنید

رتبه‌های نهایی هر پیش‌بینی را تعیین کنید

با استفاده از مجموعه آموزشی اصلی، مدل نهایی را بر اساس *S i بهینه تنظیم کنید*

الگوریتم 19. 5: حذف ویژگی‌های بازگشتی با resam Pling ­مناسب

. 6 مطالعه موردی: پیش‌بینی اختلال شناختی

بیماری آلزایمر (AD) یک اختلال اختلال شناختی است که با از دست دادن حافظه و کاهش توانایی‌های عملکردی بالاتر و فراتر از حد معمول برای یک سن مشخص می‌شود. شایع‌ترین علت زوال عقل در سالمندان است. از نظر بیولوژیکی، بیماری آلزایمر با پلاک‌های مغزی آمیلوئید *ft (A ft ) و همچنین گره‌های مغزی مرتبط با نوعی از پروتئین تاو مرتبط است.*

تشخیص AD بر شاخص‌های بالینی متمرکز است که پس از آشکار شدن، ­نشان می‌دهد که پیشرفت بیماری شدید است و به سختی قابل برگشت است. تشخیص زودهنگام بیماری آلزایمر می‌تواند به بهبود قابل‌توجهی ­در مراقبت از بیمار منجر شود. به این ترتیب، علاقه به شناسایی *نشانگرهای زیستی وجود دارد که* مقادیر قابل اندازه‌گیری هستند که شامل ارزیابی بالینی نمی‌شوند. [[65]](#footnote-65) [د لئون و کلانک](#bookmark1014) [( 2006](#bookmark1014) ) و [هاپل و همکاران ( 2010](#bookmark1017) ) شامل بحث گسترده‌ای در مورد نشانگرهای زیستی برای بیماری آلزایمر است.

اگرچه تصویربرداری متخصصی ممکن است در پیش‌بینی شروع بیماری مفید باشد، اما علاقه‌ای به بیومارکرهای بالقوه مایع کم‌هزینه نیز وجود دارد که می‌توانند از پلاسما یا مایع مغزی نخاعی (CSF) به دست آیند. در حال حاضر ­چندین بیومارکر غیرتصویربرداری پذیرفته شده وجود دارد: سطوح پروتئین اشکال خاصی از پروتئین‌های A *P* و Tau و ژنوتیپ آپولیپوپروتئین E. برای دومی، سه نوع اصلی وجود دارد: E2، E3 و E4. E4 آللی است که بیشترین ارتباط را با AD دارد [( Kim et al. 2009](#bookmark1019) ; [بو 2009](#bookmark1012) ). دقت پیش آگهی ممکن است با افزودن نشانگرهای زیستی دیگر به این لیست بهبود یابد.

کریگ شاپیرو و همکاران [( 2011](#bookmark1014) ) یک مطالعه بالینی بر روی 333 بیمار را توصیف می‌کند که ­برخی از آنها دارای اختلال شناختی خفیف (اما مشخص شده) و همچنین افراد سالم هستند. نمونه CSF از تمام افراد مورد مطالعه گرفته شد. هدف از این مطالعه تعیین این بود که آیا افراد در حالات اولیه آسیب می‌توانند از افراد سالم از نظر شناختی متمایز شوند یا خیر. داده‌های جمع‌آوری شده در مورد هر موضوع شامل:

مشخصات دموگرافیک مانند سن و جنسیت

ژنوتیپ آپولیپوپروتئین E

اندازه‌گیری پروتئین A *P،* Tau و یک نسخه فسفریله Tau (به نام pTau)

اندازه‌گیری پروتئین 124 نشانگر زیستی اکتشافی و

نمرات دمانس بالینی

برای این تحلیل‌ها، نمرات را به دو کلاس تبدیل کرده ایم: آسیب دیده و سالم. هدف از این تحلیل، ایجاد مدل‌های طبقه‌بندی با استفاده از داده‌های دموگرافیک و سنجش برای پیش‌بینی اینکه کدام بیماران در مراحل اولیه بیماری هستند، است.

با توجه به حجم نسبتاً کوچک نمونه، می‌توان استدلال کرد که بهترین ­راهبرد برای تقسیم داده‌ها، گنجاندن همه موضوعات در مجموعه آموزشی برای به حداکثر رساندن مقدار اطلاعات برای تخمین پارامترها و انتخاب پیش‌بینی‌کننده‌ها است. با این حال، با نسبت پیش‌بینی‌کننده‌ها به نقاط داده در این داده‌ها، امکان بایاس انتخاب نسبتاً زیاد است. به همین دلیل، مجموعه کوچکی از آزمودنی‌ها به‌عنوان مجموعه‌ای از آزمون برای تأیید اینکه آیا ­خطای روش‌شناختی فاحشی مرتکب نشده‌اند، عقب‌نشینی می‌شوند. مجموعه آزمون شامل 18 آزمودنی دارای اختلال و 48 فرد سالم بود. هر معیاری از عملکرد محاسبه شده از این داده‌ها دارای عدم قطعیت بالایی خواهد بود اما برای تشخیص بیش برازش به دلیل انتخاب ویژگی کافی است.

برای 267 آزمودنی در مجموعه آموزشی، پنج تکرار از ­اعتبار سنجی متقاطع 10 برابری برای ارزیابی روال‌های انتخاب ویژگی استفاده خواهد شد (به‌عنوان مثال، روش نمونه‌گیری مجدد "بیرونی"). اگر مدل‌ها دارای پارامترهای تنظیم اضافی باشند، از اعتبارسنجی متقاطع 10 برابری ساده استفاده می‌شود. تنظیم این مدل‌ها در ­هر مرحله از روال انتخاب ویژگی اتفاق می‌افتد. در طول انتخاب ویژگی و مدل

در مورد دوم، اطلاعات کلسترول به‌عنوان نقاط پایانی جانشین برای ویژگی‌های واقعا مهم (مثلاً مرگ و میر یا عوارض) در نظر گرفته می‌شود.

جدول 19. 1: فرکانس‌های مجموعه آموزشی برای دو رمزگذاری داده‌های ژنوتیپ

|  |  |  |  |
| --- | --- | --- | --- |
|  | E2/E2 E2/E3 E2/E4 E3/E3 E3/E4 E4/E4 | E2 E3 E4 | |
| دچار اختلال شده است | 0 6 1 27 32 7 | 7 | 65 40 |
| کنترل | 2 24 6 107 51 4 | 32 | 182 61 |

با تنظیم، ناحیه زیر منحنی ROC (برای احتمالات کلاس پیش‌بینی‌شده) بهینه شد.

شکل [19. 4](#bookmark895) ماتریس همبستگی 124\*124 پیش‌بینی‌کننده‌ها را نشان می‌دهد. بسیاری از همبستگی‌های قوی بین پیش‌بینی وجود دارد، همانطور که با نواحی قرمز تیره و آبی نشان داده می‌شود. میانگین همبستگی زوجی 0. 27 بود. حداقل و حداکثر همبستگی *-* 0 *بود.* به ترتیب 93 و 0. 99. بسیاری از همبستگی‌ها در گروه بزرگی از پیش‌بینی‌کننده‌ها قرار دارند (همانطور که توسط بلوک قرمز بزرگ روی مورب نشان داده شده است). این ممکن است تأثیر منفی بر روند مدل‌سازی و انتخاب ویژگی داشته باشد. تحلیل نشان داده شده در زیر از تمام پیش‌بینی‌کننده‌ها برای مدل‌سازی داده‌ها استفاده می‌کند. اعمال یک فیلتر بدون نظارت برای کاهش مجموعه ویژگی قبل از تحلیل ممکن است به بهبود نتایج کمک کند (به تمرین مراجعه کنید [19. 1 )](#bookmark914) .

تقریباً همه پیش‌بینی‌کننده‌ها در داده‌ها پیوسته هستند. با این حال، ژنوتیپ آپولیپوپروتئین E نیست. برای سه نوع ژنتیکی (E2، E3 و E4)، شش مقدار ممکن وجود دارد، زیرا یک نسخه از ژن از هر والدین به ارث می‌رسد. تفکیک این مقادیر در مجموعه آموزشی در جدول نشان داده شده است [19. 1 .](#bookmark894) وقتی به ترکیب‌های ژنتیکی مادری/پدری تقسیم ­می‌شوند، برخی از انواع فرکانس‌های بسیار کوچکی دارند (مانند E2/E2). برخی از ­مدل‌های پیش‌بینی ممکن است مسائلی در تخمین برخی پارامترها داشته باشند، به خصوص اگر فرکانس‌ها در طول نمونه برداری مجدد کاهش یابد. یک رمزگذاری جایگزین برای این داده‌ها ایجاد سه نشانگر باینری برای هر آلل است (به‌عنوان مثال، E2، E3 و E4). این نسخه از اطلاعات در سه ستون سمت راست جدول نشان داده شده است [19. 1](#bookmark894) . از آنجایی که این فرکانس‌ها به اندازه جفت‌های ژنوتیپ کم نیستند، این رمزگذاری برای همه مدل‌های پیش‌بینی نشان داده شده در اینجا استفاده می‌شود.

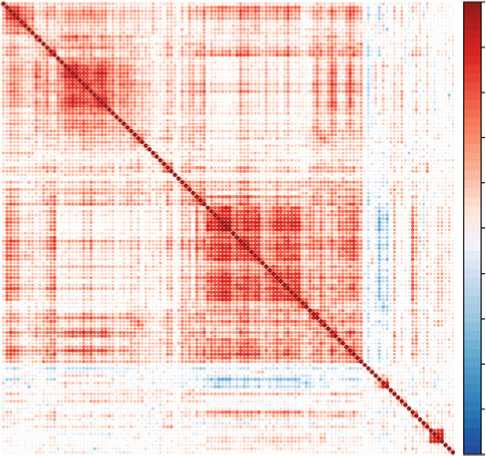
برای نشان دادن انتخاب ویژگی، حذف ویژگی بازگشتی برای چندین مدل و 66 اندازه زیر مجموعه از 1 تا 131 استفاده شد. مدل‌هایی که نیاز به تنظیم پارامتر دارند در هر تکرار حذف ویژگی تنظیم شدند. مدل‌های زیر مورد ارزیابی قرار گرفتند:

جنگل‌های تصادفی: مقدار پیش فرض *m try* = *^p* در هر تکرار استفاده شد و 1000 درخت در جنگل استفاده شد.

تحلیل تفکیک خطی: از رویکرد استاندارد استفاده شد (یعنی بدون جریمه یا انتخاب ویژگی داخلی).

رگرسیون لجستیک نامنظم: مدلی با اثرات اصلی در نظر گرفته شد.

*K* -نزدیکترین همسایه: مدل بر روی تعداد فرد همسایه از 5 تا 43 تنظیم شده است.

1

0. 8

0. 6

0. 4

0. 2

0

-0. 2

-0. 4

-0. 6

-0. 8

-1

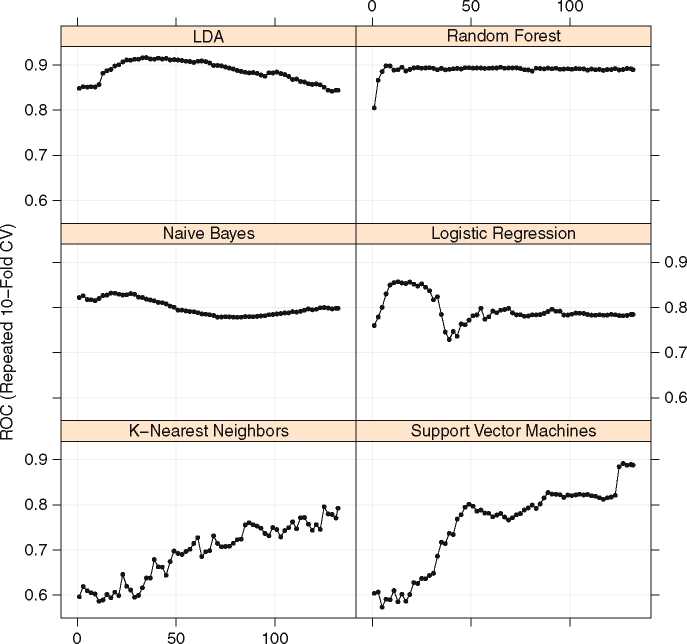
شکل 19. 4: بین همبستگی‌های پیش‌بینی برای داده‌های AD. هر *سطر* و *ستون* یکی از پیش‌بینی‌کننده‌ها را نشان می‌دهد. ترتیب ردیف‌ها و ستون‌ها با استفاده از روش‌های خوشه‌بندی مرتب شده اند

Naive Bayes: تخمین‌های ناپارامتری هسته برای پیش‌بینی‌کننده‌های پیوسته استفاده شد

ماشین‌های بردار پشتیبان: از یک هسته تابع پایه شعاعی استفاده شد. برآورد تحلیلی *a* همراه با مقادیر هزینه از 2 *\_* 2 تا 2 9 استفاده شد.

نمرات اهمیت متغیر جنگل تصادفی (بر اساس اولین مدل اولیه) پیش‌بینی‌کننده‌های آن مدل را رتبه‌بندی کرد، در حالی که رگرسیون لجستیک از قدر مطلق آماره *Z* برای هر پارامتر مدل استفاده کرد. مدل‌های دیگر پیش‌بینی‌کننده‌ها را با سطح زیر منحنی ROC برای هر پیش‌بینی جداگانه رتبه‌بندی کردند.

شکل [19. 5](#bookmark896) نمایه‌های نمونه برداری مجدد از فرآیند انتخاب ویژگی را برای هر مدل نشان می‌دهد. جنگل تصادفی تغییر بسیار کمی را نشان داد تا اینکه ­در نهایت پیش‌دیگرهای مهم از مدل حذف شدند. این تا حدودی قابل انتظار است، زیرا، حتی اگر جنگل‌های تصادفی حداقل فیلترهای تعبیه‌شده را از پیش‌بینی‌کننده‌ها انجام می‌دهند، پیش‌بینی‌کننده‌های غیر اطلاعاتی تأثیر بسیار کمی بر پیش‌بینی‌کننده‌های مدل دارند. تعداد بهینه پیش‌بینی‌کننده‌های برآورد شده برای مدل 7 بود، اگرچه آزادی عمل قابل‌توجهی در این تعداد وجود دارد. LDA بهبود زیادی را نشان داد و در 35 پیش‌بینی به اوج خود رسید و منطقه زیر منحنی ROC کمی بهتر از مدل جنگل تصادفی تخمین زده شد. رگرسیون لجستیک تا حدود 50 پیش‌بینی تغییر بسیار کمی را در عملکرد نشان داد.



تعداد پیش‌بینی ها

شکل 19. 5: نمایه‌های نمونه برداری مجدد برای روش RFE برای چندین مدل

زمانی که پس از افت اولیه، عملکرد به‌طور قابل‌توجهی بهبود یافت. بهترین اندازه زیرمجموعه برای این مدل 13 بود. با این حال، پس از حذف پیش‌بینی‌کننده‌های مهم، کاهش قابل‌توجهی در عملکرد مدل وجود دارد. ­با حذف پیش‌بینی‌کننده‌ها، Naive Bayes بهبود جزئی را نشان داد که در حدود ­17 پیش‌بینی به اوج خود رسید. برخلاف بقیه، این مدل با باقی ماندن تعداد کمی از پیش‌بینی‌کننده‌ها، کاهش شدید عملکرد را نشان نداد. هم ماشین‌های بردار پشتیبان و هم نزدیک‌ترین همسایه‌های *K* به‌طور قابل‌توجهی با حذف پیش‌بینی‌کننده‌ها آسیب دیدند و با مجموعه کامل پیش‌بینی در بهترین حالت خود قرار گرفتند. در مورد ماشین‌های بردار پشتیبان، عملکرد ضعیف ممکن است با بایاس انتخاب ارتباطی نداشته باشد. بررسی دقیق‌تر نتایج SVM نشان می‌دهد که مدل‌ها دارای ویژگی بالا و حساسیت پایین هستند (نتایج را در ­بخش محاسبات زیر ببینید). به عبارت دیگر، نتایج ضعیف احتمالاً نتیجه عدم تعادل طبقاتی در داده‌ها است.

جدول 19. 2: نتایج تأیید متقابل برای انتخاب ویژگی بازگشتی

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | مجموعه کامل | | | مجموعه کاهش یافته | | | | *p* -value |
| ROC | CI | | اندازه | ROC | CI | |
| LDA | 0. 844 | ( *0.* 82 *-* | 0. 87) | 35 | 0. 916 | ( *0.* 90 *-* | 0. 94) | 0 |
| RF | 0. 891 | ( *0.* 87 *-* | 0. 91) | 7 | 0. 898 | ( *0.* 88 *-* | 0. 92) | 0. 1255 |
| SVM | 0. 889 | ( *0.* 87 *-* | 0. 91) | 127 | 0. 891 | ( *0.* 87 *-* | 0. 91) | 0. 0192 |
| ثبت لجستیک | 0. 785 | ( *0.* 76 *-* | 0. 81) | 13 | 0. 857 | ( *0.* 83 *-* | 0. 88) | 0 |
| N. Bayes | 0. 798 | ( *0.* 77 *-* | 0. 83) | 17 | 0. 832 | ( *0.* 81 *-* | 0. 86) | 0. 0002 |
| *K* -NN | 0. 849 | ( *0.* 83 *-* | 0. 87) | 125 | 0. 796 | ( *0.* 77 *-* | 0. 82) | 1. 0000 |

ستون "CI" مربوط به 95٪ فواصل اطمینان است در حالی که ستون *p-value* مربوط به یک آزمون آماری است که ارزیابی می‌کند آیا مقدار ROC برای مدل کاهش یافته بزرگتر از منحنی مرتبط با همه پیش‌بینی‌کننده‌ها است یا خیر.

جدول [19. 2](#bookmark897) شامل خلاصه‌ای از نتایج نمونه‌گیری مجدد برای هر مدل است. ناحیه زیر منحنی ROC برای مدل کامل (یعنی همه پیش‌بینی‌کننده‌ها) در مقابل مدل‌های حاصل از ­فرآیند حذف ویژگی بازگشتی نشان داده شده است. روی میز [19. 2](#bookmark897) ، 95% فواصل اطمینان نیز نشان داده شده است. اینها از 50 تخمین نمونه برداری مجدد که با ­اعتبارسنجی متقاطع مکرر تولید شده بودند، محاسبه شدند. از بین مدل‌های ارزیابی شده، LDA بهترین تخمین ­عملکرد را دارد. با این حال، بر اساس فواصل اطمینان، این مقدار از نظر عملکرد مشابه سایر مدل‌ها (حداقل در خطای آزمایشی منعکس شده در فواصل زمانی)، از جمله جنگل تصادفی (انتخاب ویژگی پس از آن) و ­ماشین‌های بردار پورت پشتیبانی (قبل یا بعد از انتخاب ویژگی) است. . ستون *p -value* با یک آزمون آماری همراه است که در آن فرضیه صفر این است که ­مجموعه پیش‌بینی کاهش‌یافته دارای مقدار ROC بزرگ‌تر از مجموعه کامل است. این مقدار با استفاده از فرآیند نتیجه نمونه‌گیری مجدد زوجی که در بخش توضیح داده شده است محاسبه شد.  [4. 8 .](#bookmark232) اگرچه مدل‌های ماشین بردار پشتیبان و مدل‌های *K-* نزدیک‌ترین همسایه به‌طور قابل‌توجهی بهبود نیافته‌اند، شواهد قابل‌توجهی وجود دارد که مدل‌های دیگر با حذف ویژگی بازگشتی بهبود یافته‌اند.

آیا مجموعه آزمون روندهای مشابهی را با نتایج نمونه‌گیری مجدد نشان داد؟ شکل [19. 6](#bookmark898) تخمین مساحت زیر منحنی ROC را برای مدل‌های کاهش یافته نشان می‌دهد. محور *y* مطابق با مقادیر ROC نمونه برداری مجدد (و فاصله‌های اطمینان ­) که قبلاً در جدول نشان داده شده است.  [19. 2 . محور](#bookmark897) *y مقادیر مشابهی در بین مدل‌ها دارد، اما* این مقادیر از منحنی‌های ROC با استفاده از پیش‌بینی‌کننده‌های روی مجموعه آزمایشی 66 نفر محاسبه شد. عدم قطعیت بسیار بیشتری در نتایج مجموعه آزمون وجود داشت. عرض فواصل اطمینان مجموعه آزمون بیش از 3 برابر عرض فواصل مربوطه از نمونه برداری مجدد است. همبستگی متوسطی بین مقادیر ROC محاسبه شده با استفاده از مجموعه آزمون و نمونه‌گیری مجدد وجود دارد. نتایج مجموعه آزمون نسبت به ­نتایج شبیه‌سازی مجدد برای بیز ساده‌لوح، رگرسیون لجستیک و نزدیک‌ترین همسایگان *K خوش‌بینانه‌تر بود.*

1. 00

KNN o

LDA x

Logistic Reg. v

N. Bayes □ RF

SVM a

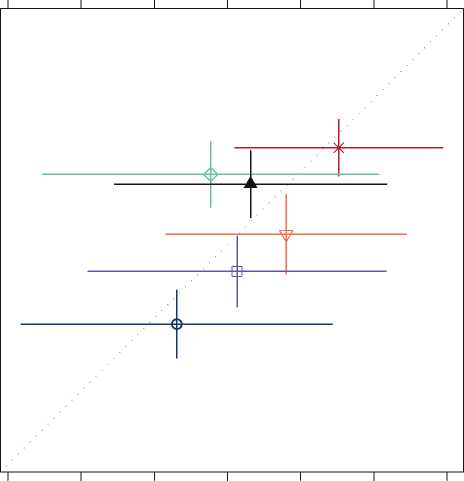
0. 95

0. 90

0. 85

0. 80

0. 75

0. 75 0. 80 0. 85 0. 90 0. 95 1. 00

نتایج مجموعه تست

تخمین مساحت زیر منحنی ROC برای مدل کاهش یافته با استفاده از مجموعه تست و اعتبارسنجی متقابل مکرر. میله‌های افقی و عمودی فواصل اطمینان 95 درصدی با استفاده از هر مدل روش هستند. همچنین، مدل ماشین بردار پشتیبان با ­اعتبارسنجی متقاطع بالاتر از مجموعه آزمون قرار گرفت. علیرغم این تناقضات، نتایج مجموعه آزمون عموماً نتایجی را که در جدول مشاهده می‌شود، تقویت می‌کند [19. 2 ;](#bookmark897) فرآیند انتخاب ویژگی ­تمایل به نشان دادن مزایایی برای چندین مدل داشت و منجر به بایاس شدید انتخاب نشد.

دو مدل پیشرو، LDA و جنگل‌های تصادفی، کاملاً متفاوت هستند. آیا این مدل‌ها همان پیش‌بینی‌کننده‌ها را انتخاب کردند؟ این یک سوال تا حدودی ذهنی ­است زیرا همانطور که در شکل نشان داده شده است.  [19. 5 ،](#bookmark896) هیچ یک از پروفایل‌های این مدل‌ها یک قله واحد نداشت. به همین دلیل، اندازه بهینه زیر مجموعه مشخص نیست. با این حال، با استفاده از نتیجه عددی بهینه، جنگل‌های تصادفی 7 و LDA 35 را انتخاب کردند. همه پیش‌بینی‌کننده‌های جنگل تصادفی در مدل LDA گنجانده شده‌اند، اما 28 پیش‌بینی LDA در مدل جنگل تصادفی نیستند. شکل [19. 7](#bookmark900) تخمین‌های اهمیت متغیر را برای هر دو مدل نشان می‌دهد (به یاد بیاورید که سطح زیر منحنی ROC برای LDA استفاده شده است). در این نمودار، مقادیر اهمیت، میانگین‌هایی هستند که در طول فرآیند نمونه‌گیری مجدد برای اندازه زیر مجموعه نهایی یافت می‌شوند. به‌عنوان مثال، نقاطی در طرح برای پیش‌بینی‌کننده‌ها وجود دارد که فقط ­در مدل LDA موجود است. این پیش‌بینی‌کننده‌ها دارای مقادیر جنگل تصادفی هستند زیرا آن پیش‌بینی حداقل در یکی از 50 نمونه مجدد با اندازه زیر مجموعه 7 پیش‌بینی انتخاب شده است. نمودار فرش روی محور *y* مربوط به مساحت زیر منحنی ROC برای پیش‌بینی‌کننده‌هایی است که در هیچ یک از مدل‌های جنگل تصادفی انتخاب نشده‌اند. مقدار مناسبی از تطابق بین این دو مدل وجود دارد. همبستگی رتبه‌ای بین پیش‌بینی‌کننده‌های رایج 0. 44 است. پیش‌بینی‌کننده‌های ناهماهنگ برای هر دو مدل نمرات اهمیت پایینی دارند. هر دو مدل حاوی دو پیش‌بینی، سنجش‌های پروتئین A ­*P و Tau بودند که تأثیر زیادی* روی مدل‌ها داشتند. سومین پیش‌بینی مهم برای هر دو مدل، سنجش تاو اصلاح شده بود. پس از این، امتیازات اهمیت با LDA/ROC با استفاده از MMP10 و جنگل تصادفی با استفاده از VEGF مخالفت می‌کنند.

این امر بر این ایده تأکید می‌کند که انتخاب ویژگی روش ضعیفی برای ­تعیین مهم‌ترین متغیرها در *داده‌ها* است، برخلاف اینکه کدام پیش‌بینی‌کننده‌ها بیشتر بر *مدل تأثیر گذاشته اند.* مدل‌های مختلف امتیازهای ­متفاوتی به پیش‌بینی‌کننده‌ها می‌دهند. برای ارزیابی اینکه کدام پیش‌بینی‌کننده‌ها با نتیجه ارتباط دارند، روش‌های آماری کلاسیک رایج بسیار مناسب‌تر هستند. مثلا، [کریگ شاپیرو و همکاران ( 2011](#bookmark1014) ) همچنین از تحلیل کوواریانس و سایر مدل‌های استنباطی برای ارزیابی ارتباط‌های احتمالی استفاده کرد. این ­تکنیک‌ها به گونه‌ای طراحی شده‌اند که نسبت به هر روش اهمیت متغیر، اظهارات احتمالی و باکیفیت بالاتری در مورد نشانگرهای زیستی بالقوه ارائه کنند. در نهایت، جنبه‌های بیولوژیکی تحلیل به‌طور بالقوه بر همه ارزیابی‌های تجربی برتری ­دارد. در حالی که این مطالعه با هدف یافتن نشانگرهای زیستی جدید انجام می‌شود، به درجه خاصی از مشروعیت علمی و همچنین آزمایشات آینده نگر برای تأیید بیشتر نتایج یافت شده در این داده‌ها نیاز است.

استفاده از فیلترها با تحلیل تفکیک خطی نیز مورد بررسی قرار گرفت. برای پیش‌بینی‌کننده‌های پیوسته، از آزمون *t ساده* برای تولید یک مقدار *p* برای تفاوت میانگین پیش‌بینی بین دو کلاس استفاده شد. برای پیش‌بینی‌کننده‌های طبقه‌بندی (به‌عنوان مثال، جنسیت، ژنوتیپ آپولیپوپروتئین E)، از آزمون دقیق فیشر برای محاسبه مقدار *p* استفاده شد که ارتباط بین پیش‌بینی و کلاس را آزمایش می‌کند. هنگامی که مقادیر *p* محاسبه شد، از دو رویکرد برای فیلتر کردن استفاده شد:

پیش‌بینی‌کننده‌ها را با مقادیر *p خام* کمتر از 0. 05 حفظ کنید، به‌عنوان مثال، نرخ مثبت کاذب 5 درصد را برای هر مقایسه جداگانه بپذیرید.

یک تصحیح بونفرونی را اعمال کنید تا زمانی که مقادیر *p* آنها کمتر از 0. 000379 یا 0 باشد، پیش‌بینی‌کننده‌ها حفظ شوند. 05/132. *\_*

برای رویکرد اول، فرآیند 47 پیش‌بینی از 132 مورد را حفظ کرد. این فرآیند نیز با استفاده از همان فرآیند اعتبارسنجی متقاطع مکرر مورد استفاده برای تحلیل لفاف، نمونه‌برداری مجدد شد. در بین 50 نمونه مجدد، فیلتر به‌طور متوسط 46. 06 پیش‌بینی را انتخاب کرد، اگرچه این عدد از 38 تا 57 متغیر بود. با استفاده از زیر مجموعه پیش‌بینی، یک مدل LDA مناسب بود. مساحت مجموعه تست حاصل در زیر منحنی ROC برای این مدل 0. 918 بود. هنگامی که فعالیت‌های فیلترینگ و مدل‌سازی مجدداً نمونه‌برداری شدند، ناحیه مشابهی در زیر منحنی ایجاد شد (AUC: 0. 917). مجموعه فیلتر شده متغیرها شامل اعضای نسبتاً کوچکی بود

فقط LDA o هر دو x

0. 80

0. 75

0. 70 - x

0. 60 -

6 8 10 12 14 16

اهمیت متغیر جنگل تصادفی

مقایسه دو روش برای تعیین اهمیت متغیر. مقادیر جنگل تصادفی با مدل‌هایی با 7 پیش‌بینی مطابقت دارد در حالی که مقادیر ROC بر اساس 35 پیش‌بینی برتر (یعنی بهترین زیر مجموعه برای LDA) است. مقادیر در بین 50 نمونه مجدد به‌طور میانگین محاسبه می‌شوند. مقادیر «فقط LDA» پیش‌بینی‌کننده‌هایی هستند که در مدل نهایی LDA بودند، اما در مدل جنگل تصادفی نیستند. نشانه‌های هش، نه محور *y،* برای پیش‌بینی‌کننده‌های LDA هستند که در هیچ یک از 50 مدل جنگل تصادفی نمونه‌برداری مجدد بین همبستگی‌ها با یک استثنا انتخاب نشده‌اند. پیش‌بینی‌کننده‌های Tau و pTau به شدت مرتبط هستند (همبستگی: 0. 91). تکرار این فرآیند بدون پیش‌بینی pTau منجر به تغییر مؤثری در ناحیه زیر منحنی ROC نشد.

هنگامی که فیلتر *p-value* برای کاهش تعداد نتایج مثبت کاذب اصلاح می‌شود، تنها 17 پیش‌بینی انتخاب شدند (میانگین 13. 46 در طول اعتبارسنجی متقاطع انتخاب شدند). در اینجا، تأثیر آن بر مدل چندان واضح نیست. سطح مجموعه آزمایشی زیر منحنی ROC بهبود جزئی داشت (AUC: 0. 926)، اما تخمین نمونه‌برداری مجدد به‌طور قابل‌توجهی بدتر بود، با ناحیه زیر منحنی 0. 856. با توجه به حجم نمونه، تغییرات قابل‌توجهی در مجموعه آزمون وجود دارد. فاصله اطمینان 95 درصد برای ناحیه زیر منحنی (0. 841، 1) است. در حالی که فرآیند اعتبار سنجی متقاطع تنها 27 بیمار را به‌طور متوسط نگه داشت، این فرآیند 50 بار تکرار شد. به همین دلیل، ممکن است در برآوردهای نمونه‌گیری مجدد بدبینانه تر اعتبار بیشتری وجود داشته باشد.

19. 7 محاسبات

این بخش داده‌ها و/یا توابع بسته‌های زیر را مورد بحث قرار می‌دهد: AppliedPredictiveModeling، caret، klaR، leaps، MASS، pROC، rms و آمار.

داده‌ها در بسته AppliedPredictiveModeling موجود است. اشیاء داده شامل یک چارچوب داده از پیش‌بینی‌کننده‌ها به نام پیش‌بینی و یک بردار فاکتور از مقادیر کلاس به نام تشخیص (با سطوح مختل شده و کنترل) است. برای آماده‌سازی داده‌ها برای تحلیل از کد زیر استفاده شده است:

*کتابخانه (AppliedPredictiveModeling)*

*داده‌ها (بیماری آلزایمر)*

*## به صورت دستی متغیرهای ساختگی جدید ایجاد کنید*

*پیش‌بینی $E2 <- پیش‌بینی $E3 <- پیش‌بینی $E4 <- 0*

*predictors$E2[grepl("2"، predictors$Genotype)] <- 1*

*predictors$E3[grepl("3"، predictors$Genotype)] <- 1*

*predictors$E4[grepl("4"، predictors$Genotype)] <- 1 >*

*## داده‌ها را با استفاده از نمونه‌گیری طبقه‌ای تقسیم کنید*

*set. seed(730)*

*تقسیم <- createDataPartition (تشخیص، p = 0. 8، لیست = FALSE)*

*## در یک قاب داده ترکیب کنید*

*adData <- پیش‌بینی ها*

*adData$Class <- تشخیص*

*آموزش <- adData[ split, ]*

*آزمایش <- adData[-split, ]*

*## یک برداری از نام متغیرهای پیش‌بینی را ذخیره کنید*

*predVars <- names(adData)[!(names(adData) %in% c("Class"، "Genotype"))]*

*## محاسبه مساحت زیر منحنی ROC، حساسیت، ویژگی،*

*## دقت و کاپا >*

*fiveStats <- function(. . . ) c(twoClassSummary(. . . )*

*+ خلاصه پیش فرض (. . . ))*

*## مجموعه داده‌های نمونه‌گیری مجدد را برای استفاده برای همه مدل‌ها ایجاد کنید*

*set. seed(104)*

*index <- createMultiFolds(training$Class, times = 5)*

*## یک برداری از اندازه‌های زیر مجموعه برای ارزیابی ایجاد کنید*

*varSeq <- seq(1، طول(predVars)-1، توسط = 2)*

کد برای بازتولید محاسبات در این فصل گسترده است و در بسته AppliedPredictiveModeling یافت می‌شود. این بخش ­نشان می‌دهد که چگونه انتخاب ویژگی را می‌توان در R برای زیر مجموعه‌ای از ­yses مقعدی انجام داد.

انتخاب رو به جلو، عقب و گام به گام

چندین تابع R برای این دسته از wrapper‌ها وجود دارد:

مرحله در بسته آماری می‌تواند برای جستجوی زیر مجموعه‌های مناسب برای مدل‌های خطی رگرسیون و تعمیم یافته (به ترتیب از توابع lm ­و glm ) استفاده شود. آرگومان جهت، روش جستجو را کنترل می‌کند

(به‌عنوان مثال "هر دو"، "عقب" یا "به جلو"). یک تابع کلی تر، تابع stepAIC در بسته MASS است که می‌تواند انواع مدل‌های اضافی را مدیریت کند. در هر صورت، از آماره AIC (یا انواع آن) به‌عنوان تابع هدف استفاده می‌شود.

تابع fastbw در بسته rms جستجوهای مشابهی را انجام می‌دهد اما انتخاب اختیاری اما توصیه نشده استفاده از *p* -values به‌عنوان تابع هدف را دارد.

تابع regsubsets در بسته جهش عملکردی مشابه دارد.

بسته klaR شامل تابع کلاس مرحله است تا ­فضای پیش‌دیکتور را برای مدل‌هایی جستجو کند که نرخ‌های دقت تایید شده متقاطع را به حداکثر می‌رسانند. این تابع دارای روش‌های داخلی برای چندین مدل مانند lda است، اما می‌توان آن را به‌طور گسترده‌تری تعمیم داد.

آموزش تابعی بسته caret دارای بسته‌بندی‌هایی برای جهش، stepAIC و stepclass است، به‌طوری که می‌توان کل فرآیند انتخاب ویژگی را مجدداً نمونه‌برداری کرد و خطر بایاس انتخاب کاهش می‌یابد.

به‌عنوان مثال، برای استفاده از stepAIC با رگرسیون لجستیک، تابع یک مدل اولیه را به‌عنوان ورودی می‌گیرد. برای نشان دادن تابع، از یک مدل کوچک استفاده شده است:

*اولیه <- glm(کلاس ~ tau + VEGF + E4 + IL\_3، داده = آموزش،*

*+ خانواده = دوجمله ای)*

*کتابخانه (MASS)*

*stepAIC (ابتدایی، جهت = "هر دو")*

شروع: AIC=189. 46

کلاس ~ tau + VEGF + E4 + IL\_3

|  |  |  |  |
| --- | --- | --- | --- |
|  | دی اف | انحراف | AIC |
| - IL\_3 | 1 | 179. 69 | 187. 69 |
| - E4 | 1 | 179. 72 | 187. 72 |
| <هیچ> |  | 179. 46 | 189. 46 |
| - VEGF | 1 | 242. 77 | 250. 77 |
| - تاو | 1 | 288. 61 | 296. 61 |

مرحله: AIC=187. 69

کلاس ~ تاو + VEGF + E4

|  |  |  |  |
| --- | --- | --- | --- |
|  | دی اف | انحراف | AIC |
| - E4 | 1 | 179. 84 | 185. 84 |
| <هیچ> |  | 179. 69 | 187. 69 |
| + IL\_3 | 1 | 179. 46 | 189. 46 |
| - VEGF | 1 | 248. 30 | 254. 30 |
| - تاو | 1 | 290. 05 | 296. 05 |

مرحله: AIC=185. 84

کلاس ~ تاو + VEGF

|  |  |  |  |
| --- | --- | --- | --- |
|  | دی اف | انحراف | AIC |
| <هیچ> |  | 179. 84 | 185. 84 |
| + E4 | 1 | 179. 69 | 187. 69 |
| + IL\_3 | 1 | 179. 72 | 187. 72 |
| - VEGF | 1 | 255. 07 | 259. 07 |

- tau 1 300. 69 304. 69

تماس: glm (فرمول = کلاس ~ tau + VEGF، خانواده = دو جمله ای، داده = آموزش)

ضرایب:

(عرض از مبدأ) تاو VEGF

9. 8075 -4. 2779 0. 9761

درجه آزادی: 266 مجموع (یعنی صفر). 264 باقیمانده

: 313. 3

انحراف باقیمانده: 179. 8 AIC: 185. 8

تابع یک شی glm را با مجموعه پیش‌بینی نهایی برمی گرداند. سایر توابع ذکر شده در بالا از استراتژی‌های مشابهی استفاده می‌کنند.

حذف ویژگی بازگشتی

بسته‌های caret و varSelRF دارای توابعی برای حذف ویژگی‌های بازگشتی ­هستند. در حالی که تابع varSelRF در varSelRF مختص جنگل‌های تصادفی است، تابع rfe در caret یک چارچوب کلی برای هر مدل پیش‌بینی است. برای دومی، توابع از پیش تعریف‌شده‌ای برای جنگل‌های تصادفی، تحلیل تفکیک خطی، درختان کیسه‌دار، بی‌بی‌های ساده، مدل‌های خطی تعمیم‌یافته، مدل‌های رگرسیون خطی و رگرسیون لجستیک وجود دارد. توابع جنگل تصادفی در لیستی به نام rfFuncs قرار دارند :

*کتابخانه (کارت)*

*## توابع جنگل تصادفی داخلی در rfFuncs هستند.*

*> str(rfFuncs)*

لیست 6

خلاصه $: تابع (داده، lev = NULL، مدل = NULL)

$ fit : تابع (x، y، اولین، آخرین،. . . )

$ pred :function (شیء، x)

رتبه $: تابع (شیء، x، y)

$ selectSize:function (x، معیار، حداکثر کردن)

$ selectVar :function (y، اندازه)

هر کدام از این تابع‌ها یک مرحله را در الگوریتم تعریف می‌کنند [19. 2](#bookmark880) :

تابع خلاصه نحوه ارزیابی پیش‌بینی‌کننده‌ها را مشخص می‌کند (خط [10](#bookmark880) در الگوریتم [19. 2 )](#bookmark880) .

تابع برازش به کاربر اجازه می‌دهد تا مدل را مشخص کند و ­تنظیم پارامترها را انجام دهد (خطوط [19. 2 ، 6 ،](#bookmark880) و [12 )](#bookmark880) .

تابع pred پیش‌بینی‌کننده‌هایی را برای نمونه‌های جدید تولید می‌کند.

تابع رتبه معیارهای اهمیت متغیر را تولید می‌کند (خط [2 )](#bookmark880) .

تابع selectSize اندازه زیر مجموعه پیش‌بینی مناسب را انتخاب می‌کند (خط [11 )](#bookmark880) .

تابع selectVar انتخاب می‌کند که کدام متغیرها در مدل نهایی استفاده شوند.

این گزینه‌ها قابل تغییر هستند. به‌عنوان مثال، برای محاسبه مجموعه گسترده‌ای از معیارهای عملکرد نشان داده شده در بالا،

*newRF <- rfFuncs*

*newRF$summary <- fiveStats*

برای اجرای رویه RFE برای جنگل‌های تصادفی، نحو است

*> ## تابع کنترل شبیه به trainControl():*

*> ctrl <- rfeControl(روش = "repeatedcv",*

|  |  |
| --- | --- |
| *+* | *تکرار = 5،* |
| *+*  *+* | *پرمخاطب = TRUE، توابع = newRF،* |
| *+* | *شاخص = شاخص)* |

*> set. seed(721)*

*> rfRFE <- rfe(x = آموزش[, predVars]،*

|  |  |
| --- | --- |
| *+*  *+* | *y = train$Class، اندازه‌ها = varSeq،* |
| *+*  *+* | *معیار = "ROC"، rfeControl = ctrl،* |
| *+*  *+*  *> rfRFE* | *## اکنون گزینه‌ها را به randomForest() ntree = 1000 منتقل کنید* |

انتخاب ویژگی بازگشتی

روش نمونه‌گیری مجدد بیرونی: اعتبارسنجی متقاطع (10 برابر، 5 بار تکرار)

عملکرد نمونه برداری مجدد از اندازه زیر مجموعه:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| متغیرها | ROC | حس | دقت مشخصات | | کاپا انتخاب شد |
| 1 | 0. 8051 | 0. 5375 | 0. 8806 | 0. 7869 | 0. 4316 |
| 3 | 0. 8661 | 0. 6407 | 0. 9167 | 0. 8415 | 0. 5801 |
| 5 | 0. 8854 | 0. 6736 | 0. 9365 | 0. 8645 | 0. 6386 |
| 7 | 0. 8980 | 0. 6571 | 0. 9414 | 0. 8637 | 0. 6300 \* |
| 9 | 0. 8978 | 0. 6850 | 0. 9506 | 0. 8779 | 0. 6679 |
| 11 | 0. 8886 | 0. 6750 | 0. 9609 | 0. 8825 | 0. 6756 |
| 13 | 0. 8895 | 0. 6604 | 0. 9609 | 0. 8786 | 0. 6636 |
| 15 | 0. 8950 | 0. 6586 | 0. 9629 | 0. 8794 | 0. 6628 |
| 17 | 0. 8867 | 0. 6554 | 0. 9621 | 0. 8780 | 0. 6576 |
| 19 | 0. 8900 | 0. 6418 | 0. 9642 | 0. 8758 | 0. 6514 |
| 129 | 0. 8923 | 0. 4439 | 0. 9826 | 0. 8351 | 0. 4947 |
| 131 | 0. 8918 | 0. 4439 | 0. 9836 | 0. 8360 | 0. 4976 |
| 132 | 0. 8895 | 0. 4439 | 0. 9815 | 0. 8345 | 0. 4963 |
| 5 متغیر برتر (از 7):  Ab\_42، tau، p\_tau، VEGF، FAS  (این خروجی کوتاه شده است و ستون‌ها برای انحرافات استاندارد هستند | | | | | |
| برای قرار گرفتن در صفحه حذف شدند. ) فرآیند پیش‌بینی جدید | | | | نمونه‌ها ساده است: | |

*> پیش‌بینی (rfRFE، سر (تست))*

|  |  |
| --- | --- |
| پیش  2 کنترل  6 آسیب دیده  15 کنترل  16 آسیب دیده  33 کنترل  38 آسیب دیده | اختلال در کنترل  0. 291 0. 709  0. 695 0. 305  0. 189 0. 811  0. 794 0. 206  0. 302 0. 698  0. 930 0. 070 |

توابع داخلی کلاس‌ها و احتمالات را برای طبقه‌بندی پیش‌بینی می‌کنند.

همچنین توابع داخلی برای انتخاب ویژگی‌های بازگشتی برای مدل‌هایی که نیاز به تنظیم مجدد در هر تکرار دارند وجود دارد. به‌عنوان مثال، برای جا دادن ماشین‌های بردار پشتیبان:

*svmFuncs <- caretFuncs*

*svmFuncs$summary <- fivestats*

*> ctrl <- rfeControl(روش = "repeatedcv",*

|  |  |
| --- | --- |
| *+*  *+* | *تکرار = 5، پرمخاطب = TRUE،* |
| *+*  *+* | *توابع = svmFuncs، index = index)* |

*> set. seed(721)*

*> svmRFE <- rfe(x = آموزش[, predVars]،*

|  |  |
| --- | --- |
| *+* | *y = آموزش$کلاس،* |
| *+*  *+* | *اندازه = varSeq، معیار = "ROC"،* |
| *+*  *+* | *rfeControl = ctrl،*  *## اکنون گزینه‌هایی برای آموزش ()* |
| *+*  *+* | *روش = "svmRadial"، طول = 12،* |
| *+*  *+* | *preProc = c("مرکز"، "مقیاس")،*  *## در زیر فرآیند نمونه‌گیری مجدد داخلی را مشخص می‌کند* |
| *+*  *+* | *trControl = trainControl (روش = "cv"، verboseIter = FALSE،* |
| *+*  *> svmRFE* | *classProbs = TRUE))* |

انتخاب ویژگی بازگشتی

روش نمونه‌گیری مجدد بیرونی: اعتبارسنجی متقاطع (10 برابر، 5 بار تکرار)

عملکرد نمونه برداری مجدد از اندازه زیر مجموعه:

متغیرهای ROC Sens Spec Accuracy Kappa انتخاب شده   
1 0. 6043 0. 000000 0. 9959 0. 7237 -0. 005400

3 0. 6071 0. 005714 0. 9858 0. 7178 -0. 010508

5 0. 5737 0. 000000 0. 9979 0. 7252 -0. 002718

7 0. 5912 0. 005357 0. 9969 0. 7259 0. 002849

9 0. 5899 0. 000000 0. 9979 0. 7252 -0. 002799

11 0. 6104 0. 000000 0. 9959 0. 7237 -0. 005625

13 0. 5858 0. 000000 0. 9979 0. 7252 -0. 002829

|  |  |  |
| --- | --- | --- |
| 121 0. 8172 0. 513571 0. 9241 | 0. 8116 | 0. 473426 |
| 123 0. 8210 0. 514286 0. 9199 | 0. 8087 | 0. 469536 |
| 125 0. 8844 0. 608571 0. 9559 | 0. 8610 | 0. 613538 |
| 127 0. 8914 0. 671786 0. 9548 | 0. 8775 | 0. 666157 |
| 129 0. 8877 0. 647500 0. 9445 | 0. 8632 | 0. 629154 |
| 131 0. 8891 0. 644643 0. 9487 | 0. 8655 | 0. 631925 |
| 132 0. 8879 0. 647143 0. 9455 | 0. 8639 | 0. 630313 |

5 متغیر برتر (از 127): Ab\_42، tau، p\_tau، MMP10، MIF

در اینجا می‌توانیم ببینیم که عملکرد ضعیف به عدم تعادل طبقاتی مربوط می‌شود. این مدل به سمت ویژگی بالا بایاس دارد زیرا اکثر نمونه‌ها کنترل هستند.

صفحه [[66]](#footnote-66)وب کرت شامل جزئیات و نمونه‌های بیشتر مربوط به rfe است.

روش‌های فیلتر

caret تابعی به نام sbf (برای Selection By Filter) دارد که می‌تواند برای غربالگری پیش‌بینی‌کننده‌ها برای مدل‌ها و تخمین عملکرد با استفاده از نمونه‌گیری مجدد استفاده شود. هر تابعی را می‌توان برای نمایش پیش‌بینی‌کننده‌ها نوشت.

به‌عنوان مثال، برای محاسبه مقدار *p* برای هر پیش‌بینی، بسته به نوع داده، می‌توان از روش زیر استفاده کرد:

*> pScore <- تابع (x, y)*

*+{*

*+ numX <- طول (یکتا (x))*

*+ if (numX > 2)*

*+{*

*+ ## با مقادیر زیادی در x، یک آزمون t را محاسبه کنید*

*+ out <- t. test(x ~ y)$p. value*

*+ }دیگر {*

*+ ## برای پیش‌بینی‌کننده‌های باینری، نسبت شانس == 1 را آزمایش کنید*

*+* ## *تست دقیق فیشر*

*+ out <- fisher. test(factor(x)، y)$p. value*

*+}*

*+ بیرون*

*+}*

*## امتیازها را برای هر یک از ستون‌های پیش‌بینی اعمال کنید*

*امتیازات <- اعمال (X = آموزش[، predVars]،*

*+ MARGIN = 2،*

*+ سرگرمی = pScore،*

*+ y = آموزش$Class)*

*> دم (نمرات)*

p\_tau Ab\_42 مرد E4 E3

1. 699064e-07 8. 952405e-13 1. 535628e-02 6. 396309e-04 1. 978571e-01

E2

1. 774673e-01

یک تابع همچنین می‌تواند برای اعمال اصلاح *p-value طراحی شود،* مانند رویه Bonferroni:

*> pCorrection <- تابع (امتیاز، x، y)*

*+{*

*+ ## گزینه‌های x و y توسط بسته caret مورد نیاز هستند*

*+ ## اما در اینجا استفاده نمی‌شود*

*+ امتیاز <- p. adjust(score, "bonferroni")*

*+ ## یک بردار منطقی برای تصمیم‌گیری در مورد پیش‌بینی‌کننده‌ها برگردانید*

*+ ## برای حفظ پس از فیلتر*

*+ دروازه بان <- (امتیاز <= 0. 05)*

*+ نگهبانان*

*+}*

*> دم (pCorrection(نمرات))*

p\_tau Ab\_42 مرد E4 E3 E2

TRUE TRUE FALSE FALSE FALSE FALSE

مانند قبل، caret شامل تعدادی توابع داخلی برای روش‌های فیلتر می‌شود: رگرسیون خطی، جنگل‌های تصادفی، درختان بسته‌شده، تحلیل تشخیص خطی و بیز ساده ( برای جزئیات بیشتر به ?rfSBF مراجعه کنید). به‌عنوان مثال، idaSBF دارای عملکردهای زیر است:

*> str(ldaSBF)*

لیست 5

خلاصه $: تابع (داده، lev = NULL، مدل = NULL)

$ fit :function (x، y،. . . )

$ pred :function (شیء، x)

امتیاز $: تابع (x، y)

فیلتر $: تابع (امتیاز، x، y)

این توابع مشابه توابع نشان داده شده برای rfe هستند. تابع امتیاز مقدار کمی از اهمیت را محاسبه می‌کند (مثلاً *p - مقادیر تولید شده توسط* تابع pScore قبلی ­). فیلتر تابع این مقادیر (و داده‌های مجموعه آموزشی خام) را می‌گیرد و تعیین می‌کند کدام پیش‌بینی از فیلتر عبور می‌کند.

برای داده‌های نشانگر زیستی، مدل LDA فیلتر شده با استفاده مناسب بود

*ldaWithPvalues <- ldaSBF ldaWithPvalues$score <- pScore ldaWithPvalues$خلاصه <- fiveStats ldaWithPvalues$فیلتر <- pCorrection sbfCtrl <- sbfControl ==روش = "repeatedWithPvaluesit, متد = "repeatedWithPvaluessv,"*

*> > > > > + + + + > + + +*

*ldaFilter <- sbf(training[, predVars]، training$Class، tol = 1. 0e-12، sbfControl = sbfCtrl)*

*> ldaFilter*

انتخاب بر اساس فیلتر

روش نمونه‌گیری مجدد بیرونی: اعتبارسنجی متقاطع (10 برابر، 5 بار تکرار)

عملکرد نمونه برداری مجدد:

ROC Sens Spec Accuracy Kappa ROCSD SensSD SpecSD AccuracySD 0. 9168 0. 7439 0. 9136 0. 867 0. 6588 0. 06458 0. 1778 0. 05973 0. 0567

KappaSD 0. 1512

با استفاده از مجموعه آموزشی، 47 متغیر انتخاب شدند:

آلفا\_1\_آنتی تریپسین، آپولیپوپروتئین\_D، B\_لنفوسیت\_شیمیایی\_جاذب\_BL، مکمل\_3، کورتیزول. . .

در طول نمونه‌گیری مجدد، 5 متغیر برتر انتخاب شده (از 66 متغیر ممکن):

Ab\_42 (100%)، کورتیزول (100%)، کراتین\_کیناز\_MB (100%)،

سیستاتین\_C (100%)، E4 (100%)

به‌طور متوسط، 46. 1 متغیر انتخاب شد (حداقل = 38، حداکثر = 57)

باز هم، وب‌سایت بسته caret جزئیات بیشتری در مورد عملکردهای rfe و sbf دارد، از جمله ویژگی‌هایی که در اینجا نشان داده نشده‌اند.

تمرینات

برای داده‌های نشانگر زیستی، تعیین کنید که آیا همبستگی بین پیش‌بینی نشان داده شده در شکل.  [19. 4](#bookmark895) بر فرآیند انتخاب ویژگی تأثیر دارد. به‌طور مشخص:

یک فیلتر اولیه از پیش‌بینی‌کننده‌ها ایجاد کنید که پیش‌بینی‌کننده‌ها را حذف می‌کند تا ­میزان چندخطی بودن داده‌ها را قبل از مدل‌سازی به حداقل برساند.

مدل‌های انتخاب ویژگی بازگشتی را دوباره تنظیم کنید.

آیا نمایه‌های RFE نشان داده شده در شکل.  [19. 4](#bookmark895) تغییر قابل‌توجهی دارد؟ کدام ­مدل‌ها احتمالاً تحت تأثیر چند خطی بودن قرار می‌گیرند؟

برای ارزیابی یک مدل LDA جریمه شده از فرآیند نمونه‌گیری مجدد استفاده کنید. عملکرد چگونه مقایسه می‌شود؟ آیا الگوی انتخاب متغیر یکسانی ­در هر دو مدل ارائه شده است؟

ترکیب‌های مختلف فیلترها و مدل‌های پیش‌بینی را روی داده‌های نشانگر زیستی اعمال کنید. آیا پیش‌بینی‌کننده‌های یکسانی در فیلترهای مختلف حفظ می‌شوند؟ آیا مدل‌ها به یک مجموعه پیش‌بینی واکنش متفاوتی نشان می‌دهند؟

تمرین یادآوری [7. 2](#bookmark382) از کجا یک ابزار شبیه سازی [فریدمن](#bookmark1016) [( 1991](#bookmark1016) ) ­از یک تابع غیرخطی از پیش‌بینی‌کننده‌ها استفاده کرد:

*y =* 10sin( *nx* 1 *x* 2 ) + 20( *x* 3 *—* 0. 5) 2 + 10 *x* 4 + 5 *x* 5 + *e* که در آن پیش‌بینی‌کننده‌های *x* 1 تا *x* 5 توزیع یکنواخت دارند و خطای *e* معمولاً با صفر توزیع می‌شود. میانگین و یک انحراف استاندارد که می‌تواند مشخص شود:

یک مجموعه آموزشی و آزمایشی را با *n* = 500 نمونه در هر مجموعه داده شبیه‌سازی کنید. با استفاده از نمودارهای پراکنده، نمودارهای همبستگی، نمودارهای جدولی و سایر تجسم‌ها، پیش‌بینی‌کننده‌ها را در مقابل نتیجه در مجموعه آموزشی ترسیم کنید.

برای انتخاب ویژگی از الگوریتم‌های رو به جلو، عقب و گام به گام استفاده کنید. آیا مدل‌های نهایی مجموعه کاملی از پیش‌بینی‌کننده‌ها را انتخاب کردند؟ چرا و چرا نه؟ اگر از ­اعتبارسنجی متقاطع با این ابزارهای جستجو استفاده می‌شد، آیا عملکرد مشابه مجموعه تست بود؟

از انتخاب ویژگی بازگشتی با چندین مدل مختلف استفاده کنید؟ این روش‌ها چگونه عمل کردند؟ آیا پیش‌بینی‌کننده‌های آموزنده انتخاب شدند؟

روش‌های فیلتر را با استفاده از فیلترهایی که در آن هر پیش‌بینی به‌طور جداگانه ارزیابی می‌شود و دیگران به‌طور همزمان ارزیابی می‌شوند (به‌عنوان مثال، الگوریتم ReliefF ) روی داده‌ها اعمال کنید. آیا دو پیش‌بینی تعاملی ( *x* 1 و *x* 2 ) انتخاب شدند؟ آیا یکی بیشتر از دیگری مورد توجه قرار گرفت؟

حجم نمونه مجموعه آموزشی را کاهش دهید و تعداد بیشتری پیش‌بینی غیر آموزنده اضافه کنید. این روش‌های جستجو در شرایط شدیدتر چگونه عمل می‌کنند؟

برای داده‌های بخش‌بندی سلول:

برای تعیین مجموعه‌ای بهینه از پیش‌بینی‌کننده‌ها از روش‌های فیلتر و پوشش استفاده کنید.

برای تحلیل تفکیک خطی و رگرسیون لجستیک، از نسخه‌های جایگزین این مدل‌ها با انتخاب ویژگی داخلی استفاده کنید (مثلاً glmnet و LDA پراکنده). رویکردهای مختلف از نظر عملکرد، تعداد پیش‌بینی‌کننده‌های مورد نیاز و زمان تمرین چگونه با هم مقایسه می‌شوند؟

فصل 20

عواملی که می‌توانند بر عملکرد مدل تأثیر بگذارند

چندین فصل قبل بر روی مسائل فنی ­مدل‌های پیش‌بینی، مانند برازش بیش از حد و عدم تعادل طبقاتی تمرکز کرده‌اند. اغلب، موفقیت واقعی ممکن است به جنبه‌هایی از مسأله بستگی داشته باشد که مستقیماً به خود مدل مربوط نمی‌شود. برای مثال، ممکن است محدودیت‌هایی در مورد آنچه داده‌ها می‌توانند پشتیبانی کنند وجود داشته باشد یا ممکن است موانع ظریف مرتبط با هدف تلاش مدل‌سازی وجود داشته باشد. یکی از موضوعاتی که در اینجا مورد بحث قرار نگرفته مربوط به پذیرش مدل‌سازی است، به‌ویژه در مناطقی که این تکنیک‌ها به‌عنوان جدید یا مخرب تلقی می‌شوند. آیرس ( 2007 ) بحث گسترده‌ای در مورد این موضوع ارائه می‌دهد. یکی دیگر از جنبه‌های مهم ­مدل‌سازی که در این فصل مورد بحث قرار نگرفته است، *مهندسی ویژگی است.* یعنی روش‌هایی برای رمزگذاری یک یا چند پیش‌بینی برای مدل.

این فصل چندین جنبه مهم ایجاد و حفظ ­مدل‌های پیش‌بینی را مورد بحث قرار می‌دهد. بخش اول به خطاهای "نوع III" می‌پردازد: ایجاد مدلی که به سوال اشتباه پاسخ می‌دهد. یک مثال گویا برای نشان دادن این موضوع استفاده می‌شود که به دلیل مسائل نمونه‌گیری در مجموعه آموزشی، مدل پیش‌بینی‌کننده‌هایی ارائه می‌دهد که به سؤالی متفاوت از سؤال مورد علاقه پاسخ می‌دهد.

نویز یا خطا درجات مختلفی از تأثیر بر عملکرد پیش‌بینی مدل‌ها دارد ­و در اکثر مجموعه‌های داده به سه شکل کلی رخ می‌دهد:

از آنجایی که بسیاری از پیش‌بینی‌کننده‌ها اندازه‌گیری می‌شوند، دارای سطحی از نویز نظام‌مند مرتبط با سامانه اندازه‌گیری هستند. هرگونه نویز اضافی در پیش‌بینی‌کننده‌ها احتمالاً مستقیماً از طریق معادله پیش‌بینی مدل منتشر می‌شود و منجر به عملکرد ضعیف می‌شود.

راه دومی که می‌توان نویز را به داده‌ها وارد کرد، گنجاندن پیش‌بینی‌کننده‌های غیر اطلاعاتی است (مثلاً پیش‌بینی‌کننده‌هایی که هیچ ارتباطی با پاسخ ندارند). برخی از مدل‌ها توانایی فیلتر کردن اطلاعات نامربوط را ­دارند و از این رو عملکرد پیش‌بینی آنها نسبتاً بی‌تأثیر است.

سومین راه ورود نویز به فرآیند مدلسازی از طریق متغیر پاسخ ­است. مانند پیش‌بینی‌ها، برخی از نتایج را می‌توان با درجه‌ای از نویز نظام‌مند و ناخواسته اندازه‌گیری کرد. این نوع خطا باعث ایجاد یک حد بالایی در عملکرد مدل می‌شود که هیچ پیش پردازش، ­پیچیدگی مدل یا تنظیمی نمی‌تواند بر آن غلبه کند. به‌عنوان مثال، اگر یک نتیجه طبقه‌بندی شده اندازه‌گیری شده در داده‌های آموزشی 10 درصد موارد اشتباه برچسب گذاری شود، بعید است.

\_20،

© که هر مدلی واقعاً می‌تواند به بیش از 90 درصد دقت دست یابد. البته، مدل ساز از این موضوع آگاه نخواهد بود و ممکن است زمان قابل‌توجهی را صرف تعقیب نویز کند.

این جنبه‌های مدل‌سازی در فرقه‌ها بررسی می‌شوند.  [20. 2](#bookmark923) و [20. 3](#bookmark928) . بخش [20. 4](#bookmark49) در حالی که بخش.  [20. 5 پیامدهای](#bookmark940) *برون یابی* مدل را بررسی می‌کند. ما این فصل را با مروری بر اینکه چگونه داده‌های بزرگ می‌توانند بر عملکرد مدل تأثیر بگذارند، پایان می‌دهیم.

20. 1 خطاهای نوع III

یکی از رایج‌ترین اشتباهات در مدل‌سازی، ایجاد مدلی است که به سؤال اشتباه پاسخ می‌دهد، در غیر این صورت به‌عنوان خطای نوع III [( Kimball ) شناخته می‌شود.](#bookmark1020)  [1957](#bookmark1020) ). اغلب، تمایل به تمرکز بر جزئیات فنی و نادیده گرفتن ناخواسته ماهیت واقعی مسأله وجود دارد. به عبارت دیگر، تمرکز بر استراتژی کلی مسأله و نه فقط تاکتیک‌های فنی راه حل بالقوه بسیار مهم است. به‌عنوان مثال، در برنامه‌های کاربردی تجاری، هدف تقریباً همیشه به حداکثر رساندن سود است. زمانی که نتیجه مشاهده‌شده مقوله‌ای باشد (مثلاً خرید/عدم خرید یا ریزش/نگهداری)، مهم است که عملکرد مدل و پیش‌بینی‌کننده‌های کلاس را به سود مورد انتظار مرتبط کنیم.

یک مثال ظریف تر از یک استراتژی مدل‌سازی مسأله ساز مربوط به *مدل‌سازی پاسخ* در بازاریابی است. مثال بازاریابی مستقیم مورد بحث در فصل را به یاد بیاورید.  [11](#bookmark497) که در آن با گروهی از مشتریان یک فروشگاه لباس برای تبلیغات تماس گرفته شد. برای هر مشتری، *پاسخ* ثبت شد (یعنی اینکه آیا خریدی انجام شده است یا خیر).

هدف واقعی فروشگاه پوشاک افزایش سود است، اما این کمپین خاص از همه جمعیت‌های مناسب نمونه‌گیری نکرد. فقط از مشتریانی استفاده کرد که با آنها تماس گرفته شده بود و این فرض را ایجاد کرد که *همه مشتریان* رفتار نشان داده شده در این جمعیت را تقلید می‌کنند. [[67]](#footnote-67) هر مدلی که از این داده‌ها ساخته می‌شود، *تنها در صورت تماس با مشتری* به پیش‌بینی احتمال خرید محدود می‌شود. این بیانیه مشروط برخلاف هدف افزایش سود کلی است. در واقع، ارائه یک تبلیغ برای مشتریانی که همیشه پاسخگو هستند، سود را کاهش می‌دهد.

به‌عنوان مثال، زیرجمعیتی از مشتریان وجود دارند که بدون توجه به یک پیشنهاد تبلیغاتی، خرید انجام می‌دهند. این پاسخ دهندگان احتمالاً در داده‌های مشاهده شده هستند و مدل قادر به تشخیص دلایل پاسخ نیست. هدف از تمرین ساخت مدل افزایش سود از طریق ارائه پیشنهادهای تبلیغاتی فقط به مشتریانی است که بدون استفاده از آن پاسخی نخواهند داشت.

سیگل [( 2011](#bookmark1025) ) چهار مورد ممکن را بیان می‌کند:

تماسی وجود ندارد

پاسخ بدون پاسخ

تماس با پاسخ *AB*

*سی دی* بدون پاسخ

سلول‌های جدول عبارتند از:

*A* : مشتریانی که بدون توجه به تماس پاسخ می‌دهند

*ب* : مشتریانی که صرفاً به دلیل تبلیغ پاسخ خواهند داد

*ج* : مشتریانی که به تبلیغات پاسخ منفی می‌دهند و اگر با آنها تماس گرفته نمی‌شد، پاسخ می‌دادند

*د* : مشتریانی که مطلقاً علاقه‌ای به پاسخگویی ندارند

برای افزایش سود، مدلی که دقیقاً پیش‌بینی می‌کند کدام مشتریان در سلول *B* هستند، مفیدترین است زیرا این جمعیت مربوط به *سود جدید است.*

پیامدهای منفی بالقوه برای مدل پاسخ ساده عبارتند از:

نرخ پاسخ بیش از حد برآورد شده است زیرا شامل مشتریانی است که همیشه پاسخ می‌دهند. سود کلی برآورد شده سود خالص نیست زیرا سود پایه تعبیه شده است. به‌عنوان مثال، منحنی افزایش تولید شده از یک مدل پاسخ ممکن است نشان دهد که با تماس با 30٪ از مشتریان، نرخ پاسخ 70٪ می‌تواند به دست آید. شانس خوبی وجود دارد که مشتریانی که همیشه پاسخ می‌دهند با اطمینان از مدل امتیاز بگیرند و درصدی از 30٪ تعیین شده برای تماس را مصرف کنند.

هزینه‌ها با ارسال تبلیغات برای مشتریان در سلول‌های *C* یا *D افزایش می‌یابد.*

برای ساختار هزینه که توسط [Larose ( 2006](#bookmark1020) ، فصل 7)، هزینه ارتقاء نسبتاً پایین بود. با این حال، در برخی شرایط، هزینه‌ها می‌تواند بسیار بیشتر باشد. به‌عنوان مثال، قرار دادن مشتریان در معرض تبلیغات ناخواسته می‌تواند تأثیر مخربی بر احساسات آنها نسبت به تجارت داشته باشد. همچنین، در جایی که از مدل‌های پاسخ برای حفظ مشتری استفاده می‌شود، این احتمال وجود دارد که تماس *باعث* ریزش شود زیرا یادآوری می‌کند که ممکن است معامله بهتری با شرکت دیگری پیدا کنند.

تا زمانی که تبلیغات وارد عمل نشود، نمی‌توان این مسائل را رعایت کرد.

تکنیک‌هایی که سعی در درک تأثیرات پاسخ مشتری دارند، مدل‌سازی افزایش نامیده [*می‌شوند* (](#bookmark1025) *Segel* [2011](#bookmark1025) ; [رادکلیف و ساری 2011](#bookmark1024) ; Rzepakowski و Jaroszewicz [2012](#bookmark1025) )، *مدل‌های آسانسور واقعی* [( Lo 2002](#bookmark1021) )، *حالت‌های بالابر خالص،* *مدل‌های بالابر افزایشی،* یا *مدل‌سازی پاسخ واقعی.*  [Lo ( 2002](#bookmark1021) ) پیشنهاد می‌کند که یک گروه کنترلی از مشتریانی که با آنها تماس گرفته نشده است می‌توان برای ایجاد یک مدل پاسخ جداگانه برای متمایز ساختن مشتریان در سلول‌های *A* و *C* از سلول‌های *B* و *D استفاده کرد (*اگرچه *B* و *D* را نمی‌توان با مدل متمایز کرد. ). در ارتباط با یک مدل پاسخ سنتی که با استفاده از مشتریانی که با آنها تماس گرفته شده ایجاد می‌شود، یک استراتژی امتیازدهی می‌تواند یافتن مشتریانی با مقادیر بزرگ باشد.

*P r* [پاسخ *|* تماس] *- P r* [پاسخ *|* بدون تماس].

با کم کردن احتمال آن دسته از مشتریانی که بدون تماس پاسخ می‌دهند، از ابزار دقیق‌تری برای یافتن مجموعه مشتری مناسب استفاده می‌شود. اشکال این روش غیر مستقیم بودن آن است. همچنین، این احتمال وجود دارد که پیش‌بینی‌کننده‌های مدل بسیار همبسته باشند، زیرا آنها رویدادهای بسیار مشابهی را مدل‌سازی می‌کنند. اگر چنین باشد، ممکن است مشتریان بسیار کمی وجود داشته باشند که دو [احتمال](#bookmark1024) مشابه برای آنها نباشد. رادکلیف و ساری [( 2011](#bookmark1024) ) مدل‌های مبتنی بر درخت را توصیف می‌کنند که در آن بالابردن مستقیماً با گروه کنترلی از مشتریانی که با آنها تماس گرفته نشده است مدل‌سازی می‌شود.

روش دیگر می‌تواند استفاده از تکنیک‌های نمونه‌گیری پیچیده تر برای ایجاد یک مجموعه آموزشی مناسب باشد. برای جدول بالا امکان تماس و عدم تماس با همان مشتری وجود ندارد. با این حال، در تحقیقات متخصصی، این مسأله اغلب هنگام ارزیابی یک درمان جدید در برابر درمان موجود با آن مواجه می‌شود. در اینجا، آزمایش‌های بالینی گاهی اوقات از *نمونه‌های همسان استفاده می‌کنند.* دو آزمودنی پیدا شد که تقریباً یکسان هستند و به‌طور تصادفی در ­گروه‌های درمان قرار می‌گیرند. ایده این است که تنها عامل متمایز کننده درمان است و پاسخ بیمار را می‌توان با دقت بیشتری نسبت به بدون تطابق تخمین زد. ایده مهم در اینجا این است که آزمودنی‌ها دیگر واحد آزمایشی نیستند. جفت منطبق به خودی خود به نقطه داده اولیه در تحلیل تبدیل می‌شود.

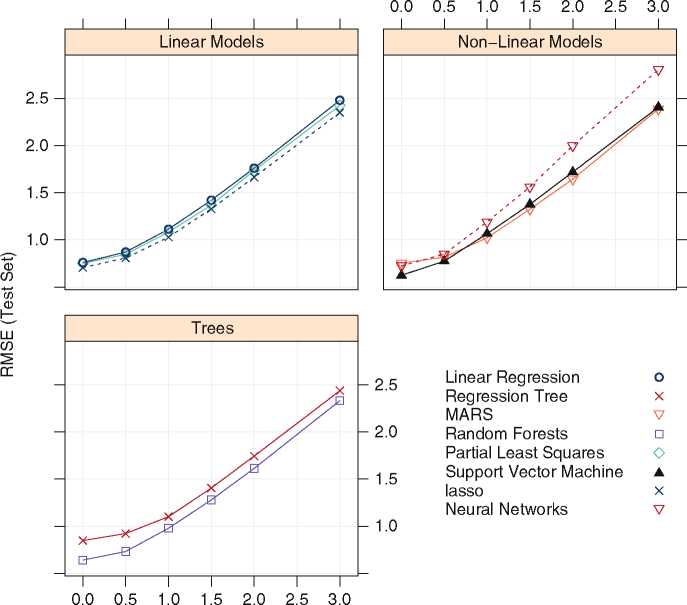
همین استراتژی را می‌توان برای این وضعیت به کار برد. فرض کنید یک نمونه اولیه از مشتریان را می‌توان با سایرینی که ویژگی‌های مشابهی مانند عوامل جمعیت شناختی و سطح درآمد دارند، مطابقت داد. در هر جفت منطبق، یک تبلیغ می‌تواند به‌طور تصادفی به یک مشتری در جفت اختصاص داده شود. اکنون، برای هر جفت همسان، جدول 2 *× 2 بالا می‌*تواند ایجاد شود. اگر نتایج در تمام جفت‌های همسان جمع شوند، می‌توان یک مدل طبقه‌بندی برای چهار نتیجه مختلف ( *A و D* ) ایجاد کرد و مشتریان را بر اساس احتمال قرار گرفتن در کلاس B که گروه مورد علاقه است، امتیاز داد. این رویکرد به‌طور مستقیم جمعیت مورد نظر را در یک مدل واحد مدل می‌کند.

خطای اندازه‌گیری در نتیجه

حداقل میزان خطای یک مدل چقدر است؟ مدل رگرسیون خطی نشان داده شده در معادله را به یاد بیاورید. 6. 1 :

*y i* = *b 0* + *b* 1 *X i* 1 + *b 2 X i 2* + + *b p X ip* + *e i.*

باقیمانده *e i* فرض می‌شود که مقداری واریانس به *صورت 2 نشان داده می‌شود.* اگر ساختار مدل واقعی را بدانیم (یعنی پیش‌بینی‌کننده‌های دقیق و رابطه آن‌ها با نتیجه)، آنگاه عدد *2 کمترین* خطای ممکن را نشان می‌دهد یا *خطای غیرقابل کاهش را نشان می‌دهد.* با این حال، ما معمولاً ساختار مدل واقعی را نمی‌دانیم، بنابراین این مقدار برای شامل *خطای مدل (*به‌عنوان مثال، خطا) افزایش می‌یابد.



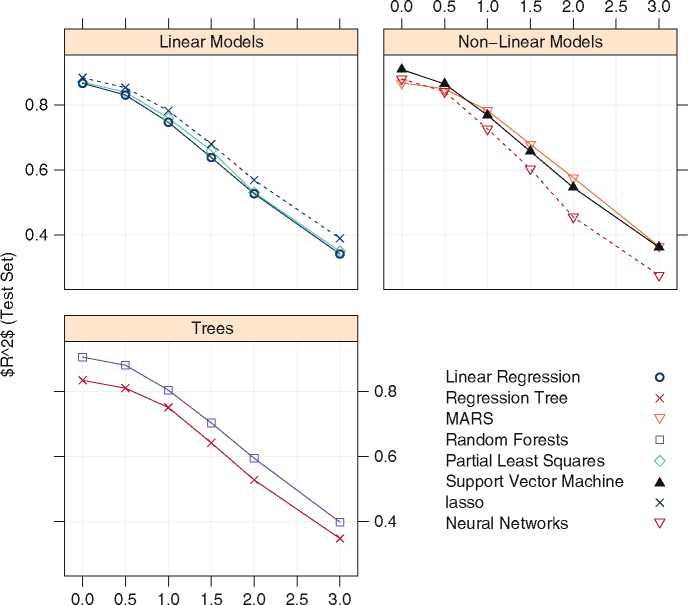
Factor of Additional RMSE

مجموعه تست پروفایل‌های RMSE برای مدل‌های حلالیت زمانی که نویز سامانه اندازه‌گیری مربوط به عدم برازش افزایش می‌یابد). در طول فرآیند مدلسازی، هدف از بین بردن خطای مدل است.

Fig.

با این حال، مؤلفه دیگری وجود دارد که به 2 کمک می‌*کند* که از طریق فرآیند مدل‌سازی حذف نمی‌شود. اگر نتیجه شامل *نویز اندازه‌گیری قابل‌توجه باشد،* خطای کاهش ناپذیر در بزرگی افزایش می‌یابد. ریشه میانگین مربعات خطا و *R2* سپس به دلیل این خطا دارای کران‌های پایین و بالایی هستند. بنابراین، عبارت خطا، علاوه بر اینکه شامل تغییرات در پاسخ است که توسط مدل توضیح داده نشده است، خطای سامانه اندازه‌گیری را جمع‌آوری می‌کند. هرچه سامانه اندازه‌گیری و محدودیت‌های آن و همچنین رابطه بین پیش‌بینی‌کننده‌ها و پاسخ را بهتر درک کنیم، بهتر می‌توانیم محدودیت‌های عملکرد مدل را پیش‌بینی کنیم. همانطور که در مقدمه این فصل ذکر شد، مسأله مشابهی در طبقه‌بندی رخ می‌دهد.

برای نشان دادن تأثیر درجه خطا در پاسخ بر عملکرد مدل، داده‌های QSAR حلالیت را مجدداً بررسی می‌کنیم (فصل 6 تا 9 ). پاسخ برای این داده‌ها، گزارش اندازه‌گیری حلالیت است (شکل 2).  [6. 2 )](#bookmark283) ،



عامل RMSE اضافی

شکل 20. 2: مجموعه تست *R* 2 پروفایل برای مدل‌های حلالیت زمانی که نویز سامانه اندازه‌گیری افزایش می‌یابد

و پیش‌بینی‌کننده‌ها یا پیوسته و باینری بودند (بخش.  [6. 1 )](#bookmark281) . بهترین ­مدل رگرسیون گوش خطی برای این داده‌ها دارای RMSE 0. 7 بود که شامل خطای سامانه اندازه‌گیری، خطای عدم برازش و خطاهای ناشی از ­عوامل پیش‌بینی مربوطه است که در مدل گنجانده نشده‌اند. برای مثال، از این مقدار به‌عنوان سطح پایه برای خطا استفاده می‌کنیم و خطا در پاسخ را به‌طور متناسب افزایش می‌دهیم. برای انجام این کار، نویز را به مقدار حلالیت ورود به سامانه هر ترکیب اضافه کرده‌ایم که از توزیع نرمال با میانگین صفر و ­انحراف استاندارد مدل رگرسیون خطی RMSE ضرب در ضریب بین 0 تا 3 می‌شود. برای هر سطح از نویز اضافی، مدل‌های زیر را آموزش و تنظیم کردیم: رگرسیون خطی، حداقل مربعات جزئی، درخت‌های رگرسیون تک، خطوط رگرسیون تطبیقی چند متغیره، جنگل‌های تصادفی، شبکه‌های عصبی و ماشین‌های بردار پشتیبان تابع پایه شعاعی. عملکرد برای هر یک از این مدل‌ها بر اساس RMSE و *R* 2 ارزیابی می‌شود برای مجموعه تست و نتایج در شکل نشان داده شده است.  [20. 1](#bookmark924) و [20. 2 .](#bookmark925) واضح است که عملکرد بدون توجه به نوع مدل بدتر می‌شود (مدل‌های خطی، مدل‌های غیرخطی یا درختان) و به نسبت میزان خطای اضافی که در پاسخ گنجانده شده است بدتر می‌شود.

دو نکته مهم از این تصویر ساده وجود دارد. اول، این نوع نویز، نویز است که هیچ مدلی نمی‌تواند آن را پیش‌بینی کند - ما به سادگی در آن گیر کرده ایم و نمی‌توانیم از کف RMSE یا *R* 2 عبور کنیم. سقف. بنابراین، هرچه مدل ساز بیشتر در مورد سامانه اندازه‌گیری بداند، بهتر می‌تواند ­انتظارات در مورد عملکرد مدل را درک کند. دوم، با افزایش نویز، مدل‌ها از نظر عملکرد پیش‌بینی‌شان عملاً غیرقابل تشخیص ­می‌شوند. این بدان معناست که مزایایی که برخی از مدل‌های پیچیده‌تر، مانند مجموعه‌ها، دارند، تنها زمانی سودمند هستند که خطای سامانه اندازه‌گیری نسبتاً کم باشد. این منطقی است زیرا ساختار پیچیده‌ای که یک مدل (مانند یک مجموعه) می‌تواند پیدا کند، با افزایش نویز مبهم تر می‌شود. بنابراین، زمانی که نویز سامانه اندازه‌گیری زیاد است، احتمالاً می‌توانیم با یک مدل ساده و کارآمد محاسباتی به همان خوبی عمل کنیم.

خطای اندازه‌گیری در پیش‌بینی ها

همانطور که در چندین مجموعه داده نشان داده شده است، بسیاری از پیش‌بینی‌کننده‌ها محاسبه می‌شوند. به‌عنوان مثال، داده‌های حلالیت حاوی اثر انگشت‌هایی بود که وجود یا عدم وجود یک ساختار شیمیایی خاص را نشان می‌داد. در متن کاوی، پیش‌بینی‌کننده‌ها ممکن است بسامد برخی کلمات یا عبارات مهم موجود در متن باشد. در موارد دیگر، پیش‌بینی‌کننده‌ها مشاهده می‌شوند یا ممکن است محصول برخی فرآیندهای خارجی باشند. مثلا:

داده‌های تقسیم‌بندی سلولی جنبه‌های مختلف سلول‌ها مانند مساحت هسته را اندازه‌گیری کردند.

داده‌های گرنت اطلاعاتی را در مورد تعداد گرنت موفق و ناموفق جمع‌آوری کرد.

مدل‌های آماری سنتی معمولاً فرض می‌کنند که پیش‌بینی‌کننده‌ها ­بدون خطا اندازه‌گیری می‌شوند، اما همیشه اینطور نیست. تأثیر تصادفی بودن در پیش‌بینی‌کننده‌ها می‌تواند شدید باشد، بسته به عوامل متعددی: میزان تصادفی بودن، اهمیت پیش‌بینی‌کننده‌ها، نوع مدل مورد استفاده و همچنین عوامل دیگر. در برخی موارد، اگر داده‌های اولیه‌ای که در ­مجموعه آموزشی قرار دارند در شرایط بسیار کنترل شده تولید شوند، تصادفی بودن را می‌توان پنهان کرد. به‌عنوان مثال، فرض کنید داده‌ها توسط شخصی که به صورت دستی یک شی را اندازه‌گیری می‌کند (مانند رتبه بندی) تولید می‌شود. ممکن است تفاوت‌هایی در نحوه ­درک افراد از شی وجود داشته باشد که در نتیجه نویز رتبه‌دهنده به رتبه‌دهنده ایجاد می‌شود. اگر از یک ارزیاب واحد برای داده‌های مجموعه آموزشی استفاده شود، اما نرخ دیگری برای داده‌های جدید استفاده شود، بایاس بین ارزیاب‌ها احتمالاً مسائلی را ایجاد می‌کند.

به‌عنوان مثالی دیگر، داده‌های مخلوط بتن را که قبلاً بحث شد در نظر بگیرید. با وجود اینکه نسبت‌ها و مقادیر دقیق هر یک از ترکیبات مخلوط ­مشخص است، در هنگام ایجاد مخلوط‌ها مقداری انحراف در این مقادیر وجود دارد. اگر فرآیند ایجاد بتن پیچیده باشد، مقدار واقعی استفاده شده ممکن است با فرمول متفاوت باشد (اگرچه این تفاوت اندازه‌گیری یا مشاهده نمی‌شود).

برای نشان دادن تأثیر نویز پیش‌بینی تصادفی بر روی مدل‌ها، یک موج سینوسی  *ساده* با خطای تصادفی شبیه‌سازی شد. شکل [20. 3](#bookmark932) داده‌های اصلی را در پانل با برچسب "SD = 0" نشان می‌دهد. در اینجا، داده‌های محور *x* دارای مقادیر مساوی بین 2 و 10 هستند. نتیجه با افزودن مقدار کمی نویز معمولی به داده‌ها ایجاد شد. در هر پانل، رابطه واقعی بین پیش‌بینی و پاسخ به صورت یک خط مشکی یکپارچه نشان داده می‌شود. برای برازش داده‌ها از دو مدل رگرسیون مختلف استفاده شد. ستون سمت چپ پانل‌ها یک مدل رگرسیون خطی معمولی را نشان می‌دهد که در آن شکل مدل واقعی (یعنی *sin (* *x* )) برای داده‌ها مناسب است. ستون سمت راست پانل‌ها مطابق با درخت رگرسیون CART است. منحنی‌های نصب شده از این مدل‌ها به صورت خطوط قرمز ضخیم نشان داده شده است.

ردیف‌های موجود در شکل [20. 3](#bookmark932) همان داده‌ها بدون نویز (در ردیف بالا) هستند و زمانی که خطای تصادفی به صورت تدریجی به مقادیر پیش‌بینی اضافه می‌شود. به‌طور خاص، مقادیر ­نرمال تصادفی با انحرافات استاندارد مختلف به مقادیر واقعی اضافه می‌شوند. منحنی‌های زنگ‌شکل آبی کوچک در این شکل، توزیع احتمال یک نقطه داده را با میانگین شش و ­انحراف استاندارد مربوطه را نشان می‌دهند. مقادیر محور *y* در همه پانل‌ها یکسان است.

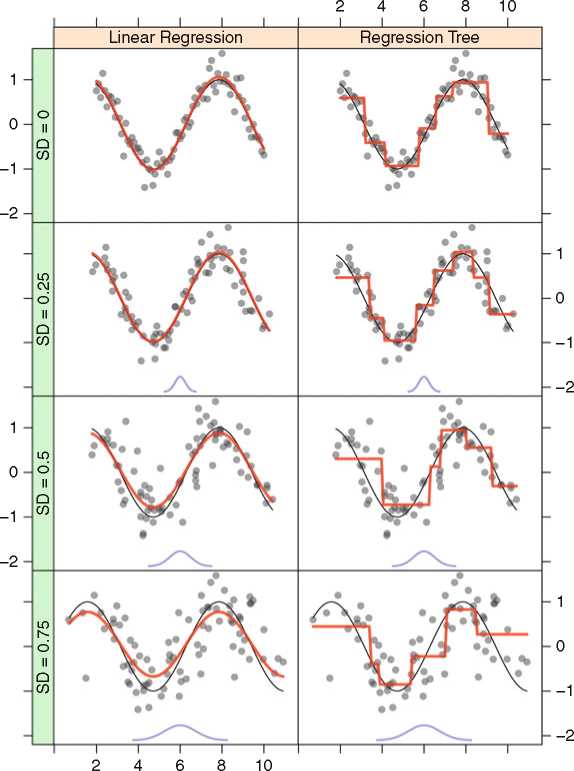
رگرسیون خطی قادر است به‌طور موثر رابطه واقعی را بدون نویز اضافی مدل کند. همانطور که نویز به پیش‌بینی‌کننده‌ها اضافه می‌شود، مدل رگرسیون خطی در نقاط اوج و نادر منحنی ضعیف می‌شود. این اثر با انحراف استاندارد 0. 75 بسیار برجسته می‌شود. با این حال، تصویر ارزیابی تا حدودی خوش بینانه است زیرا فرض می‌کند که مدل ساز از قبل رابطه واقعی بین *x* و *y را می‌داند* و مدل را با این دانش مشخص کرده است.

بدون نویز اضافی، درخت رگرسیون نیز الگو را در محدوده داده‌های مشاهده شده به خوبی تقریب می‌کند. نتایج درخت رگرسیون مسأله سازتر است زیرا الگوریتم باید الگوی داده‌ها را به صورت تجربی تعیین کند. علاوه بر این، این مدل خاص ناپایدار است (یعنی بایاس کم اما واریانس بالا). در این حالت، تفاوت بین رابطه پیش‌بینی‌شده و واقعی با مقادیر کمتر خطای تصادفی آشکارتر می‌شود.

خطاهای اندازه‌گیری در پیش‌بینی‌کننده‌ها می‌تواند مسائل قابل‌توجهی را هنگام ساخت مدل‌ها ایجاد کند، به‌ویژه از نظر تکرارپذیری نتایج در مجموعه داده‌های آینده (مشابه با برازش بیش از حد). در این مورد، نتایج آتی ممکن است ضعیف باشد زیرا داده‌های پیش‌بینی اساسی با مقادیر مورد استفاده در مجموعه آموزشی متفاوت است.

مطالعه موردی: پیش‌بینی عوارض جانبی ناخواسته

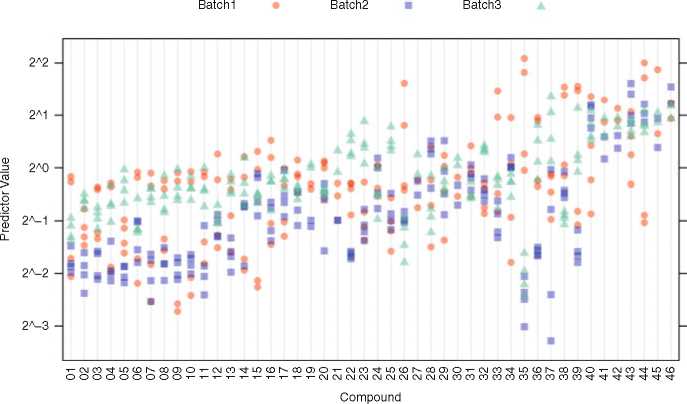
شرکت‌های داروسازی دارای بخش‌هایی برای *استخراج* ترکیبات هستند، یعنی تلاش می‌کنند تشخیص دهند که آیا داروی کاندید دارای عوارض جانبی مضر است یا خیر.



ارزش پیش‌بینی اندازه‌گیری شده

شکل 20. 3: یک موج سینوسی  *شبیه‌سازی* شده و دو مدل زمانی که مقادیر متفاوتی مطابقت دارند

سر و صدا به مقادیر پیش‌بینی سمیت در انسان اضافه می‌شود. یکی از تکنیک‌های تشخیص این مسائل، ایجاد سنجش‌های مبتنی بر سلول است که سیگنال‌هایی را در مورد خطرناک بودن ترکیب می‌دهند. در ارتباط با این نتایج آزمایشگاهی، مجموعه‌ای از ترکیبات با مسائل شناخته شده و همچنین مجموعه‌ای از ترکیبات "پاک" بدون هیچ مسأله شناخته شده‌ای شناسایی می‌شوند. سنجش‌های بیولوژیکی ایجاد شده و به‌عنوان ورودی برای مدل‌های پیش‌بینی استفاده می‌شود. بسیار شبیه مثال قبلی که حلالیت از ­ساختارهای شیمیایی پیش‌بینی می‌شد، این مدل‌ها از اندازه‌گیری نحوه واکنش خطوط سلولی به ترکیبات برای پیش‌بینی مسائل ایمنی بالقوه استفاده می‌کنند.



شکل 20. 4: نتایج یک آزمایش مورد استفاده برای مشخص کردن منابع نویز ناخواسته در مجموعه‌ای از سنجش‌های بیولوژیکی

در مثال ارائه شده در اینجا، هدف توسعه یک مدل پیش‌بینی برای یک سمیت خاص بود. مجموعه‌ای از حدود ده سنجش ایجاد شد که میزان رونویسی RNA (که در غیر این صورت به‌عنوان بیان ژن شناخته می‌شود) را در یک نوع خاص از سلول اندازه‌گیری می‌کرد. حدود 250 ترکیب (اعم از سمی یا تمیز) مورد سنجش قرار گرفت و از این نتایج برای آموزش مجموعه‌ای از مدل‌ها استفاده شد. مدل‌ها در حدود 80 درصد حساسیت و ویژگی‌ها را نشان دادند.

با این حال، هنگامی که مجموعه جدیدی از ترکیبات آزمایش شد (به همراه چندین کنترل از مجموعه آموزشی اصلی)، نتایج بهتر از ­حدس زدن تصادفی نبود. پس از بررسی فرآیند ساخت مدل برای خطاهای روش شناختی، کیفیت سنجش‌ها مورد بررسی قرار گرفت.

آزمایشی اجرا شد که در آن زیرمجموعه‌ای از 46 ترکیب در سه دسته مجزا (در طی روزهای غیر متوالی) اندازه‌گیری شد. در هر دسته، چندین تکرار از هر ترکیب وجود داشت. تجسم ­نقاط داده جداگانه برای یکی از سنجش‌ها در شکل 1 نشان داده شده است.  [20. 4 ،](#bookmark933) که در آن ترکیبات بر اساس مقدار متوسط آنها مرتب شده اند. واضح است که اثرات دسته به دسته در داده‌ها وجود دارد، به خصوص برای برخی از ترکیبات. روندها یکنواخت نیستند. در بسیاری از موارد، دسته سه بیشترین مقادیر را داشتند اما نه برای هر ­ترکیب. به‌علاوه، در یک دسته، مقادیر پیش‌بینی می‌توانند به‌طور چشمگیری متغیر باشند - گاهی اوقات در دو گزارش. به‌عنوان مثال، ترکیب 35 کل محدوده داده‌های پیش‌بینی را در بر می‌گیرد.

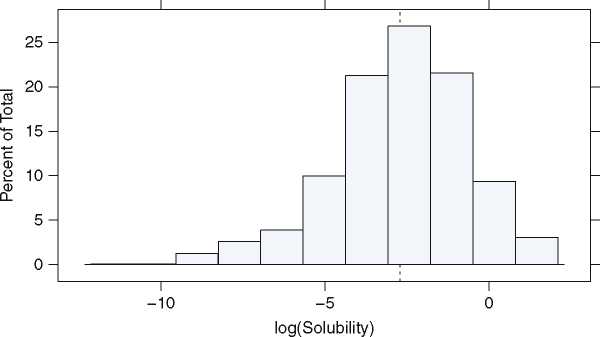
روش‌های آماری به نام مدل‌های مؤلفه‌های واریانس (گاهی اوقات به نام روش‌های *تکرارپذیری و تکرارپذیری سنج* [( مونتگومری و رانگر)](#bookmark1022) [1993](#bookmark1022) )) منبع نویز در داده‌ها را کمی کنید. در این طرح آزمایشی، بهترین حالت ممکن این است که ترکیب به ترکیب متغیر باشد­

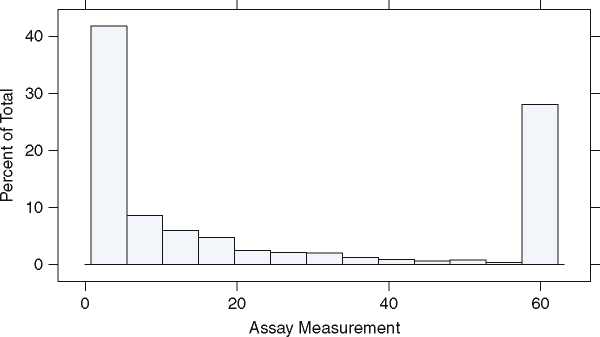
100 درصد نویز در داده‌ها را تشکیل می‌دهد (یعنی هیچ نویز دسته به دسته یا درون دسته وجود نخواهد داشت). با استفاده از تحلیل مؤلفه‌های واریانس، ترکیبات حدود 38 درصد از نویز در داده‌ها را به خود اختصاص دادند. تغییرات دسته به دسته 13 درصد از کل و تغییرات درون دسته‌ای 49 درصد بود. واضح است که این وضعیت بهینه نیست زیرا اکثریت زیادی از تغییرات در داده‌ها نویز ناخواسته است. [[68]](#footnote-68) از آنجایی که برخی از این ترکیبات سمی هستند و برخی دیگر سمی نیستند، بعید است که این اندازه گیری‌ها به ­تمایز ترکیبات سمی و تمیز کمک کند. سنجش‌های دیگر نتایج مشابهی داشتند. لازم به ذکر است که این سنجش‌ها بر اساس داده‌های آزمایش‌های دیگر انتخاب شدند که منطق بیولوژیکی مناسبی برای استفاده از آنها در این زمینه ایجاد کردند. بنابراین، ما معتقدیم که در این پیش‌بینی‌کننده‌ها سیگنالی وجود دارد، اما با نویز سامانه اندازه‌گیری در حال غرق شدن است.

گسسته‌سازی نتایج مستمر

در بسیاری از زمینه‌ها، ممکن است مطلوب باشد که با یک پاسخ طبقه‌بندی شده کار شود، حتی اگر پاسخ اصلی در مقیاس پیوسته باشد. [[69]](#footnote-69) این می‌تواند به دلیل این واقعیت باشد که توزیع اساسی پاسخ واقعاً دووجهی است. شکل ­\_ [20. 5](#bookmark936) که دو هیستوگرام از نتایج عددی از مجموعه داده‌های مختلف را نشان می‌دهد. هیستوگرام بالایی داده انحلال پذیری است که تاکنون در این فصل مورد بحث قرار گرفته است در حالی که هیستوگرام در پایین شکل نشان دهنده توزیع یک اندازه‌گیری برای مجموعه داده دیگری است. در حالی که توزیع حلالیت متقارن است، توزیع پایین به وضوح دووجهی است که در آن بیشتر داده‌ها در هر دو انتهای پاسخ یافت می‌شوند، با نسبتاً کمی در میانی. تلاش برای دسته‌بندی داده‌ها در توزیع بالا دشوار خواهد بود، زیرا هیچ تمایز طبقه‌بندی طبیعی وجود ندارد. با این حال، دسته‌بندی داده‌ها در توزیع پایین طبیعی تر است.

در موقعیت‌های دیگر، تمایل به کار با پاسخ در ­مقیاس طبقه‌بندی ممکن است به دلایل عملی باشد. برای داده‌هایی که تاکنون بررسی کرده‌ایم، حلالیت ممکن است یکی از بسیاری از ویژگی‌هایی باشد که ­دانشمندان برای پیشبرد روند کشف دارو ارزیابی می‌کنند. برای ساده کردن این مسئله بهینه‌سازی چند بعدی، تصمیم گیرندگان ممکن است ترجیح دهند بدانند که آیا یک ترکیب به *اندازه کافی محلول است* یا نه به جای مقدار حلالیت ورود به سامانه پیش‌بینی شده ترکیب. انتخاب مجموعه‌ای از ترکیبات بهینه را می‌توان در یک چک لیست در میان شرایط مورد علاقه که در آن ترکیبات ترجیحی آنهایی هستند که بیشترین خواص مورد علاقه را برآورده می‌کنند، ساده کرد. در حالی که این فرآیند می‌تواند به‌طور کلی امیدوار کننده‌ترین ترکیبات را برای تحقیقات بیشتر شناسایی کند، تفاوت‌های ظریف تر





شکل 20. 5: مقایسه توزیع بین دو مجموعه داده. *بالا* : اندازه‌گیری‌های حلالیت در ترکیباتی که قبلاً در این فصل مورد بحث قرار گرفت. خط عمودی نقطه چین میانگین مقدار حلالیت ورود به سامانه است که برای دسته‌بندی این داده‌ها استفاده می‌شود. *پایین* : مقادیر نتیجه برای مجموعه داده دیگری. این توزیع به وضوح دو وجهی است و دسته‌بندی داده‌ها به دو دسته طبیعی تر خواهد بود

ترکیبات بین ترکیبات از بین می‌روند زیرا اطلاعات موجود در نتیجه دور ریخته می‌شود.

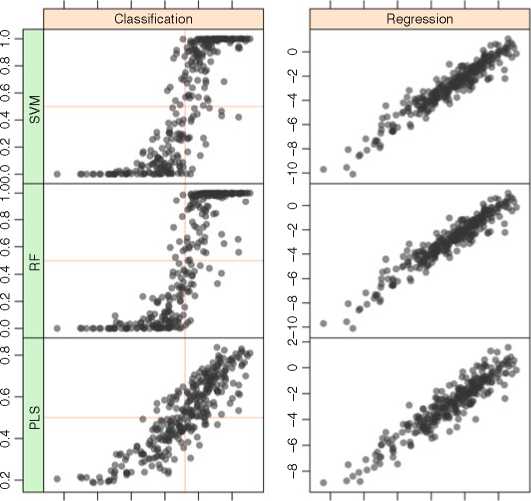
هنگامی که پاسخ دووجهی (یا چندوجهی) است، دسته‌بندی پاسخ مناسب است. با این حال، اگر پاسخ از یک توزیع پیوسته پیروی کند، همانطور که داده‌های حلالیت انجام می‌دهند، طبقه‌بندی پاسخ قبل از مدل‌سازی باعث از دست رفتن اطلاعات می‌شود که کاربرد کلی مدل را ضعیف می‌کند.

برای نشان دادن این از دست دادن اطلاعات، ما به کار با ­داده‌های sol ubility ادامه خواهیم داد. برای این تصویر، ما پاسخ را به‌عنوان بالاتر یا پایین تر از میانگین (-2. 72 واحد ورود به سامانه‌) طبقه‌بندی کرده ایم و هر مدل طبقه‌بندی را با چند اصلاح لازم تنظیم کرده ایم. رگرسیون خطی با رگرسیون لجستیک جایگزین شده است، MARS با FDA جایگزین شده است و LASSO با g lmnet جایگزین شده است که در آن هر جایگزینی ­تکنیک طبقه‌بندی موازی است. پس از آموزش هر مدل، احتمال هر ترکیب در مجموعه آزمایشی را علاوه بر مقدار کاپا برای مجموعه آزمایشی پیش‌بینی می‌کنیم. برای هر ترکیب مجموعه آزمایشی، ما همچنین پاسخ پیوسته پیش‌بینی‌شده را داریم که می‌توانیم با احتمال پیش‌بینی‌شده محلول بودن یک ترکیب هماهنگ کنیم. سپس می‌توانیم نمودارهای پراکندگی پیش‌بینی پیوسته را با پیش‌بینی احتمال در بین ترکیبات مقایسه کنیم.

شکل [20. 6](#bookmark940) نتایج مجموعه آزمون را برای ­رویکردهای رگرسیون و طبقه‌بندی برای PLS، SVM و جنگل‌های تصادفی نشان می‌دهد (نتایج سایر مدل‌ها مشابه هستند و در تمرین ارائه شده اند.  [20. 1 )](#bookmark953) . برای هر یک از ­مدل‌های رگرسیون، مقادیر log (حلالیت) مشاهده‌شده و پیش‌بینی‌شده از یک خط توافق با مقادیر پیش‌بینی‌شده پیروی می‌کنند که در حدود چهار log (حلالیت) واحد از مقادیر واقعی در محدوده پاسخ قرار می‌گیرند. با نگاهی ­به مرکز توزیع، یک مقدار log (حلالیت) پیش‌بینی‌شده *-* 4 به مقادیر مشاهده‌شده بازمی‌گردد که بین تقریباً *-* 6 و *-* 2 واحد ورود به سامانه در سراسر مدل‌ها متغیر است. از سوی دیگر، یک احتمال پیش‌بینی‌شده 0. 5 برای مدل‌های طبقه‌بندی به مقادیر واقعی بازمی‌گردد که بین تقریباً *-* 6and0forPLSand *-* 4 و 0 برای SVM و جنگل‌های تصادفی است. دامنه پیش‌بینی‌کننده‌ها در انتهای مدل‌های طبقه‌بندی حتی گسترده‌تر است. برای مثال، یک احتمال پیش‌بینی‌شده نزدیک به صفر برای SVM مربوط به مقادیر log (حلالیت) واقعی در محدوده *-* 10 تا *-* 2 است و ­توانایی‌های احتمالی پیش‌بینی‌شده نزدیک به یک با مقادیر واقعی بین *-* 3 مطابقت دارد. 5 و 2 (به‌طور مشابه برای جنگل‌های تصادفی). در این مثال، کار با داده‌ها در مقیاس اصلی، پیش‌بینی‌کننده‌های دقیق‌تری را در محدوده پاسخ برای همه مدل‌ها فراهم می‌کند. مقایسه‌های بیشتر نتایج مدل برای این داده‌ها در تمرین ارائه شده است [20. 1](#bookmark953) .

دومین دلیل رایج برای دسته‌بندی پاسخ پیوسته این است که دانشمند ممکن است بر این باور باشد که پاسخ پیوسته دارای درجه بالایی از خطا است، به‌طوری که تنها مقادیر پاسخ در هر دو انتهای توزیع احتمالاً به درستی طبقه‌بندی می‌شوند. در این صورت، داده‌ها را می‌توان به سه دسته تقسیم کرد که در آن داده‌ها در هر دو حالت به‌طور کلی به‌عنوان مثبت و منفی طبقه‌بندی می‌شوند، در حالی که داده‌های میان‌رده به‌عنوان ناشناخته یا نامعین طبقه‌بندی می‌شوند. دسته میانی را می‌توان در یک مدل گنجاند (یا به‌طور خاص از فرآیند تنظیم مدل حذف شد) تا به مدل کمک کند تا به راحتی بین دو دسته تمایز قائل شود.

-10 -8 -6 -4 -2 0



-10 -8 -6 -4 -2 0

مشاهده شده

شکل 20. 6: مقایسه عملکرد مجموعه آزمایشی مدل‌های حلالیت زمانی که پاسخ به‌عنوان یک مقدار پیوسته و طبقه‌بندی شده مدل می‌شود.

چه زمانی باید به پیش‌بینی مدل خود اعتماد کنید؟

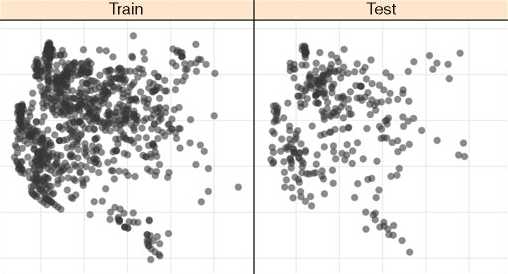
ما می‌توانیم این بخش را به‌عنوان "چه زمانی نباید *به پیش‌بینی* مدل خود اعتماد کنید ­؟" فرآیند مدل‌سازی پیش‌بینی فرض می‌کند که مکانیسم زیربنایی ­که داده‌های فعلی و موجود را هم برای پیش‌بینی‌کننده‌ها و هم برای پاسخ تولید می‌کند، به تولید داده‌ها از همان مکانیسم ادامه می‌دهد. برای مثال ساده‌ای از مکانیزم تولید داده، تجارت تجاری تولید مواد غذایی را در نظر بگیرید. شرکت‌های غذایی تجاری تلاش می‌کنند محصول مشابهی را در طول زمان ایجاد کنند تا مشتریان بتوانند به‌طور مداوم تجربه مزه غذا را داشته باشند. بنابراین شرکت‌ها دستور غذا، مواد تشکیل دهنده و فرآیند آماده‌سازی را در طول زمان یکسان نگه می‌دارند. اگر درصد رطوبت شیرینی‌های شکلاتی را از یک نانوایی تجاری مدل‌سازی می‌کردیم، انتظار می‌رفتیم که پیش‌بینی‌کننده‌های اندازه‌گیری شده روی دسته جدیدی از کوکی‌ها احتمالاً در محدوده پیش‌بینی‌کننده‌های مجموعه آموزشی قرار بگیرند (جمع‌آوری‌شده در برخی از مقاطع گذشته در زمان). ). این بدان معنی است که داده‌های جدید دارای ویژگی‌های مشابه هستند و قسمت‌های مشابهی از فضای پیش‌بینی را به‌عنوان داده‌هایی که مدل بر روی آن ساخته شده است اشغال می‌کند.

برای مثال‌های مورد استفاده در این کتاب، ­گام‌های مناسبی برای ایجاد مجموعه‌های آزمایشی برداشته‌ایم که دارای ویژگی‌های مشابه در فضای پیش‌بینی مانند مجموعه آموزشی بودند. اصول زیربنای این فرآیند در بخش توضیح داده شد.  [4. 3 .](#bookmark204) اگر داده‌های جدید توسط همان ­مکانیزم تولید داده مانند مجموعه آموزشی تولید شوند، می‌توانیم این اطمینان را داشته باشیم که مدل پیش‌بینی معقولی برای داده‌های جدید خواهد داشت. با این حال، اگر داده‌های جدید با مکانیزم یکسان تولید نشوند، یا اگر مجموعه آموزشی آنقدر کوچک یا پراکنده بود که به اندازه کافی محدوده فضایی را که معمولاً توسط مکانیسم تولید داده پوشش می‌داد پوشش دهد، پیش‌بینی‌کننده‌های مدل ممکن است قابل اعتماد نباشد. *برون یابی* معمولاً به‌عنوان استفاده از یک مدل برای پیش‌بینی نمونه‌هایی که خارج از محدوده داده‌های آموزشی هستند تعریف می‌شود ( آرمیتاژ و بری 1994 ). مهم است که برای داده‌های با ابعاد بالا تشخیص داده شود که ممکن است مناطقی در محدوده پیش‌بینی‌کننده‌ها وجود داشته باشد که هیچ داده آموزشی وجود نداشته باشد. بنابراین، ما باید مفهوم برون یابی خود را گسترش دهیم تا این مناطق داخلی خالی از فضای پیش‌بینی را نیز در بر گیرد. پیش‌بینی‌کننده‌های برون‌یابی، چه خارج از محدوده پیش‌بینی‌کننده‌ها یا در مناطق خالی فضا، ممکن است قابل اعتماد نباشند و منجر به تصمیم‌گیری ضعیف شوند.

تشخیص اینکه چه زمانی یک مدل در حال تولید یک پیش‌بینی برون‌یابی است، می‌تواند دشوار باشد و اولین دفاع تمرین‌کننده، درک عمیق ­مکانیزمی است که برای تولید مجموعه آموزشی و مجموعه جدیدی از نمونه‌ها که برای آنها پیش‌بینی ایجاد می‌شود، استفاده می‌شود. دوباره فرآیند تولید کوکی‌های چیپس کولات تجاری را در نظر بگیرید و فرض کنید که سازنده دستور غذا، مواد تشکیل دهنده و فرآیند پخت را تغییر داده است. استفاده از مدلی که قبلاً توسعه داده شده برای پیش‌بینی رطوبت کوکی‌ها از فرآیند فعلی ممکن است پیش‌بینی‌کننده‌های نادرستی به همراه داشته باشد، زیرا پیش‌بینی‌کننده‌های داده‌های جدید احتمالاً در بخش متفاوتی از فضا با داده‌های آموزشی هستند. اگر تمرین‌کننده این اطلاعات را در مورد داده‌های موجود بداند، می‌تواند در اعمال مدل و تفسیر پیش‌بینی‌کننده‌های حاصل از احتیاط مناسب استفاده کند.

با این حال، اغلب اوقات، متخصص نمی‌داند که آیا مکانیسم تولید داده برای داده‌های جدید مشابه داده‌های آموزشی است یا خیر. درک مجموعه‌های داده با پیش‌بینی‌کننده‌های کم بسیار آسان‌تر است و روابط بین پیش‌بینی‌کننده‌ها در مجموعه آموزشی و مجموعه جدید نمونه‌ها را می‌توان از طریق نمودارهای پراکندگی ساده و مقایسه توزیع‌های هر پیش‌بینی بررسی کرد. اما با افزایش ابعاد فضا، بررسی نمودارهای پراکندگی و توزیع‌ها بسیار ناکارآمد است و ممکن است به درک صحیح فضای پیش‌بینی بین داده‌های آموزشی و داده‌های جدید منجر نشود. در این شرایط چند ابزار وجود دارد که می‌توان برای درک بهتر شباهت داده‌های جدید با داده‌های آموزشی استفاده کرد.

حوزه *کاربرد* یک مدل، منطقه‌ای از فضای پیش‌بینی است که در آن «مدل با یک قابلیت اطمینان معین پیش‌بینی می‌کند» [( Netzeva et al. 2005](#bookmark1023) ). انواع مختلفی از این تعریف وجود دارد، اما ساده‌ترین آنها تعریف دامنه از نظر شباهت به داده‌های مجموعه آموزشی است. اگر داده‌های جدید پیش‌بینی شده به اندازه کافی مشابه مجموعه آموزشی باشد، فرض بر این است که این نقاط به‌طور متوسط دارای قابلیت اطمینان هستند که مشخص می‌شود.



10

5

0

-5

10

15

-5 0

5

10

15

-5 0

5

10

15

جزء 1

PCA برای داده‌های حلالیت رسم می‌کند

توسط برآوردهای عملکرد مدل (به‌عنوان مثال، نرخ دقت به دست آمده از یک مجموعه تست).

مقایسه کلی فضای تحت پوشش پیش‌بینی‌کننده‌ها از مجموعه آموزشی و مجموعه جدید را می‌توان با استفاده از تکنیک‌های کاهش ابعاد معمولی مانند تحلیل مؤلفه‌های اصلی (بخش [3. 3 )](#bookmark147) یا مقیاس‌بندی چند بعدی [( دیویسون 1983](#bookmark1014) ). اگر داده‌های آموزشی و داده‌های جدید از یک مکانیسم تولید شوند، طرح این داده‌ها در نمودار پراکندگی همپوشانی دارند. با این حال، اگر داده‌های آموزشی و داده‌های جدید بخش‌های مختلفی از نمودار پراکندگی را اشغال کنند، ممکن است داده‌ها با مکانیسم یکسانی تولید نشوند و پیش‌بینی‌کننده‌های داده‌های جدید باید با احتیاط مورد استفاده قرار گیرند. شکل [20. 7](#bookmark942) طرح ریزی مجموعه‌های آموزشی و آزمایشی برای داده‌های حلالیت را بر روی دو جزء اصلی داده‌های آموزشی نمایش می‌دهد. در این مورد، به نظر می‌رسد که داده‌های آموزش و آزمایش همان فضایی را اشغال می‌کنند که توسط این مؤلفه‌ها تعیین می‌شود. البته، از آنجایی که مجموعه آموزشی و مجموعه تست به‌طور تصادفی از مجموعه داده‌های اصلی تقسیم شدند، این نتیجه قابل انتظار است. اما بررسی بیشتر شباهت بین نمونه‌های آموزش و مجموعه تست در بخش توسعه داده شده است.  [20. 7](#bookmark950) و ورزش [20. 3](#bookmark954) .

هنگامی که بسیاری از پیش‌بینی‌کننده‌ها را در دو بعد مطرح می‌کنیم، روابط پیچیده پیش‌بینی و همچنین فضاهای پراکنده و متراکم را می‌توان پنهان کرد. این به این معنی است که در حالی که ممکن است داده‌های آموزشی و داده‌های جدید در طرح پیش‌بینی همپوشانی داشته باشند، ممکن است مناطقی در فضای پیش‌بینی وجود داشته باشد که مدل به‌طور کافی نمونه‌های جدید را پیش‌بینی نمی‌کند.

برای رفع این مسأله، [هستی و همکاران ( 2008](#bookmark1018) ) رویکردی را برای تعیین کمیت احتمال اینکه یک نمونه جدید عضوی از داده‌های آموزشی باشد، توصیف می‌کند. در این رویکرد، پیش‌بینی‌کننده‌های آموزشی انتخاب می‌شوند. سپس یک نمونه یکنواخت چند متغیره تصادفی بر اساس محدوده پیش‌بینی‌کننده‌های آموزشی تهیه می‌شود

اهمیت متغیر را برای مدل اصلی محاسبه کنید و 20 پیش‌بینی برتر را شناسایی کنید

به‌طور تصادفی این پیش‌بینی‌کننده‌ها را از مجموعه آموزشی جابجا کنید

به‌صورت ردیفی، پیش‌بینی‌کننده‌های برتر مجموعه آموزشی اصلی و نسخه‌ای که به‌طور تصادفی از این پیش‌بینی‌کننده‌ها جابه‌جا شده است را به هم متصل کنید.

یک بردار طبقه‌بندی ایجاد کنید که ردیف‌های مجموعه آموزشی اصلی و ردیف‌های مجموعه آموزشی تغییر یافته را مشخص کند.

یک مدل طبقه‌بندی بر روی داده‌های تازه ایجاد شده آموزش دهید

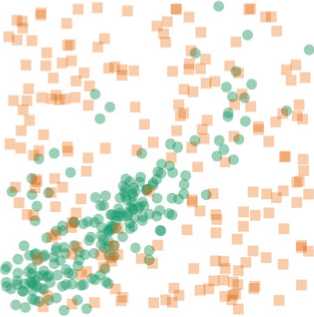
از مدل طبقه‌بندی برای پیش‌بینی احتمال وجود داده‌های جدید در کلاس مجموعه آموزشی استفاده کنید.

الگوریتم 20. 1: الگوریتم تعیین شباهت به ­مجموعه آموزشی

به گونه‌ای تولید می‌شود که تعداد نمونه‌های مشابه داده‌های آموزشی را داشته باشد. بردار پاسخ با داده‌های آموزشی اصلی که به‌عنوان مجموعه آموزشی و نمونه یکنواخت تصادفی به‌عنوان مجموعه تصادفی شناسایی شده است طبقه‌بندی می‌شود. سپس می‌توان هر مدل طبقه‌بندی را بر روی این مجموعه ساخت و از مدل به‌دست‌آمده می‌توان برای پیش‌بینی احتمال عضویت یک نمونه جدید از مجموعه آموزشی استفاده کرد.

ما دو تغییر جزئی در این روش پیشنهاد می‌کنیم تا داده‌های واقعی را بهتر نشان دهیم. اول، با توجه به اینکه بسیاری از مجموعه‌های داده حاوی انواع مختلفی از پیش‌بینی‌کننده‌ها هستند، توصیه می‌کنیم به جای نمونه‌گیری از توزیع یکنواخت در طیف وسیعی از پیش‌بینی‌کننده‌ها، پیش‌بینی‌کننده‌ها را به‌طور تصادفی تغییر دهید. با انجام این کار، ما به یک توزیع تصادفی در کل فضا دست می‌یابیم، در حالی که در ­مقادیر غیر قابل اشتباه هر پیش‌بینی قرار می‌گیریم. بنابراین پیش‌بینی‌کننده‌های طبقه‌بندی فقط مقادیر مقوله‌ای مناسب را دریافت خواهند کرد. دوم، با توجه به اینکه داده‌ها هم از نظر پیش‌بینی‌کننده‌ها و هم از نظر ابعاد بزرگ هستند، توصیه می‌کنیم مدل طبقه‌بندی را بر روی زیرمجموعه‌ای از پیش‌بینی‌کننده‌های اصلی بسازید. به‌طور خاص، پیشنهاد می‌کنیم ۲۰ پیش‌بینی مهم (یا برخی کسر معقول بر اساس زمینه مسئله ­و مدل) را از مدل اصلی داده‌های آموزشی انتخاب کنید. این امر زمان ساخت مدل را تا حد زیادی کاهش می‌دهد، در حالی که همچنان مدلی تولید می‌کند که می‌تواند به ارزیابی اینکه آیا مدل یک پیش‌بینی برون‌یابی شده را تولید می‌کند یا خیر. مراحل این فرآیند در الگوریتم ذکر شده است [20. 1](#bookmark943) .

برای نشان دادن این روش، به اصل مثال دو بعدی که ­در شکل 1 نشان داده شده است باز می‌گردیم.  [4. 1](#bookmark196) . البته، در اینجا انتخاب متغیر را حذف می‌کنیم و به مرحله 2 می‌رویم. داده‌های اصلی را با مقادیر جابجایی تصادفی پیش‌بینی‌کننده‌های اصلی افزایش دادیم (شکل 1).  [20. 8 )](#bookmark946) و یک درخت طبقه‌بندی کیسه‌ای ساخت. سپس 50 نمونه تصادفی تولید کردیم و آنها را در مراحل متوالی از داده‌های اصلی دورتر کردیم و از درخت طبقه‌بندی بسته‌بندی شده برای پیش‌بینی احتمال اینکه نمونه‌ها از داده‌های آموزشی هستند استفاده کردیم.



0.0-

Training Set • Random

0.6-

0.4-

0.2-

0.2 0.4 0.6

Predictor A

0.0

داده‌های یکنواخت تصادفی به داده‌های اصلی که ­در شکل 1 نشان داده شده است اضافه شده است.  [4. 1](#bookmark196) به منظور پوشش محدوده فضای دو پیش‌بینی

محل قرارگیری نمونه‌های جدید و همچنین احتمال عضویت نمونه‌ها در مجموعه آموزشی در شکل 1 نشان داده شده است.  [20. 9](#bookmark947) . واضح است که با ­دورتر شدن نمونه‌ها از داده‌های آموزشی اصلی، احتمال عضویت در مجموعه آموزشی کاهش می‌یابد.

20. 6 تأثیر یک نمونه بزرگ

تمرکز ما بر عملکرد تا کنون از طریق دریچه پیش‌بینی‌کننده‌ها و پاسخ بوده است. اکنون به تأثیر تعداد نمونه‌ها بر عملکرد مدل می‌پردازیم. یک فرض اساسی این است که هر چه نمونه‌های بیشتری داشته باشیم، مدل بهتری می‌توانیم تولید کنیم. این فرض مسلماً از توانایی ما برای دستیابی آسان به هر تعداد نمونه که می‌خواهیم تقویت می‌شود. به‌عنوان نمونه‌ای از سهولت دستیابی به داده‌ها، آیرس ( 2007 ) را در نظر بگیرید که فرآیندی را که برای نامگذاری کتاب خود به نام *Super Crunchers دنبال کرد، شرح داد.* برای درک نظر جمعیت خوانندگان در مورد انتخاب عناوین خود، او از چندین نوع از تبلیغات هدفمند گوگل استفاده کرد که هر کدام نام نامزد متفاوتی برای کتاب داشتند. پس از مدت کوتاهی، او یک چهارم میلیون نمونه را جمع‌آوری کرد که بر روی آن آگهی‌ها بیشتر کلیک شده است. از آنجایی که تبلیغات بود

0. 6 -

0. 4-

0. 2 -I

0. 0

0. 5

1. 0

پیش‌بینی A

0. 6

0. 4

0. 2

0. 0 0. 5 1. 0

پیش‌بینی A

0. 0 0. 2 0. 4 0. 6 0. 8 1. 0

شباهت به مجموعه آموزشی

0. 6 -

0. 4-

0. 2 -I

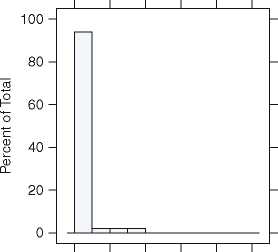
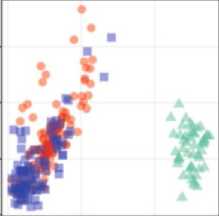
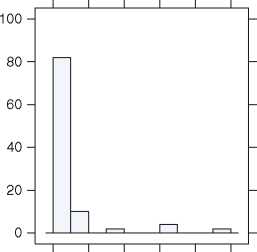
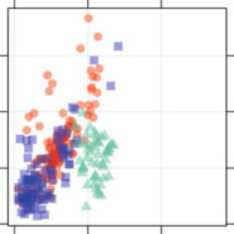
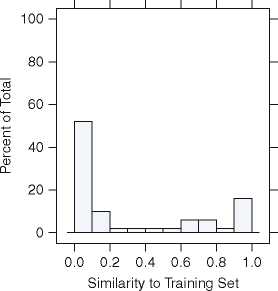
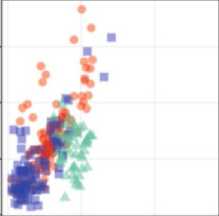
0. 0 0. 5 1. 0

پیش‌بینی A

0. 0 0. 2 0. 4 0. 6 0. 8 1. 0

شباهت به مجموعه آموزشی

شکل 20. 9: تصویری برای تشخیص برون یابی مدل. ستون *سمت چپ* شکل‌ها سه سناریو مجموعه آزمایشی را با فاصله از نزدیک به داده‌های آموزشی ( *بالا* ) تا دور از داده‌های آموزشی ( *پایین* ) نشان می‌دهد. ستون *سمت راست* شکل‌ها احتمال این را نشان می‌دهد که نمونه‌های مجموعه آزمون اعضای داده‌های آموزشی اصلی هستند



این آزمون در مقیاس بزرگ که به‌طور تصادفی ارائه شد، شواهدی قوی ارائه کرد که نشان می‌دهد جمعیت خوانندگان کدام نام کتاب را بیشتر دوست دارند.

این رویکرد ساده و هوشمندانه برای درک یک جمعیت، مثال مثبتی از چگونگی استفاده مؤثر از تعداد زیادی از نمونه‌ها ارائه می‌دهد. اما همانطور که در بخش‌های قبلی دیدیم، نویز در ­پیش‌دیکتورها یا پاسخ می‌تواند هر گونه مزیتی را که ممکن است با افزایش تعداد نمونه‌ها به همراه داشته باشد، به حداقل برساند. علاوه بر این، افزایش تعداد نمونه‌ها می‌تواند پیامدهای مثبت کمتری داشته باشد. اول، بسیاری از مدل‌های پیش‌بینی، با افزایش تعداد نمونه‌ها (و پیش‌بینی‌کننده‌ها) بار محاسباتی قابل‌توجهی دارند. به‌عنوان مثال، یک درخت طبقه‌بندی منفرد جستجوهای جامع بسیاری را در بین نمونه‌ها و پیش‌بینی‌کننده‌ها انجام می‌دهد تا تقسیم بهینه داده‌ها را بیابد. با افزایش ابعاد داده‌ها، زمان محاسبه نیز افزایش می‌یابد. علاوه بر این، بار محاسباتی برای مجموعه‌های درختان حتی بیشتر است و اغلب به سخت‌افزار گران‌تر و/یا پیاده‌سازی‌های ویژه ­مدل‌هایی نیاز دارند که محاسبات را امکان‌پذیر می‌کنند. دوم، بازدهی کاهشی برای افزودن بیشتر *داده‌های مشابه* از یک مجموعه ­جمع‌آوری شده وجود دارد. از آنجایی که مدل‌ها با تعداد نمونه‌های به اندازه کافی تثبیت می‌شوند، جمع‌آوری نمونه‌های بیشتر احتمال کمتری دارد که برازش مدل را تغییر دهد.

به‌عنوان مثال، فناوری جستجوی وب در ابتدا از محتوای یک وب سایت برای رتبه‌بندی نتایج جستجو استفاده می‌کرد. اگر کسی روی عبارتی مانند " مدل‌سازی پیش‌بینی" جستجو کند، الگوریتم جستجو بر روی عبارات موجود در مجموعه وب سایت‌های مجموعه خود تمرکز می‌کند و از آنها برای پیش‌بینی اینکه کدام صفحات وب مرتبط هستند استفاده می‌کند. برای این عبارت جستجوی خاص، وب‌سایت‌های مرتبط با « یادگیری ماشین »، « تشخیص الگو »، « داده کاوی » و موارد دیگر نیز ممکن است مرتبط باشند و الگوریتم باید آن را بفهمد. برای این الگوریتم‌ها، اضافه کردن وب‌سایت‌های بیشتر به مجموعه آنها بعید است که نتایج جستجو را به‌طور قابل‌توجهی بهبود بخشد، اما افزودن داده‌های متفاوت ممکن است. یک پورتال جستجوی وب مانند گوگل نیز می‌تواند تعامل کاربر را ردیابی کند. این اهرم اطلاعات مستقیم تر و با کیفیت بالاتر برای هر جستجو است. برای مثال، اگر بخش زیادی از کاربرانی که « مدل‌سازی پیش‌بینی » را جستجو کرده‌اند، روی وب‌سایت *A کلیک کنند،* این احتمال مرتبط بودن آن را افزایش می‌دهد. در این مثال، افزودن داده‌های *مختلف* مؤثرتر از افزودن درک بیشتر از ویژگی‌های داده مشابه است. به‌طور خلاصه، " *P بزرگ* " معمولا بیشتر از "Big *n* " کمک می‌کند. مواردی وجود دارد که افزودن نمونه‌های بیشتر ممکن است کیفیت پیش‌بینی‌کننده‌ها را بهبود بخشد، اما باید به خاطر داشت که *داده‌های بزرگ* ممکن است به معنای *داده‌های بهتر نباشد.*

به‌طور خلاصه، تعداد زیادی نمونه می‌تواند مفید باشد، به خصوص اگر نمونه‌ها حاوی اطلاعات در سراسر فضای پیش‌بینی باشند، نویز در پیش‌بینی‌کننده‌ها و پاسخ را می‌توان به حداقل رساند، یا محتوای جدیدی اضافه می‌شود. در عین حال، هزینه این نمونه‌ها بار محاسباتی را افزایش می‌دهد.

20. 7 محاسبات

جزئیات محاسباتی برای مدل‌های آموزشی مورد بحث در این فصل را می‌توان در بخش‌های قبلی کتاب یافت. یکی از موضوعات محاسباتی جدید ارائه شده در اینجا به اجرای الگوریتم می‌پردازد [20. 1](#bookmark943) . برای نشان دادن این روش، به بسته R caret ارجاع داده خواهد شد.

برای نشان دادن اجرای الگوریتم شباهت، ابتدا داده‌های حلالیت را بارگذاری کنید و ساختار کنترل را برای آموزش تعریف کنید. در اینجا ما از ساختار آموزشی استفاده شده در سراسر متن برای این داده‌ها استفاده خواهیم کرد.

*کتابخانه (AppliedPredictiveModeling)*

*داده (حلالیت)*

*set. seed (100)*

*indx <- createFolds(solTrainY, returnTrain = TRUE)*

*ctrl <- trainControl(روش = "cv"، index = indx)*

در مرحله بعد، مدل مورد نظر را تنظیم کنید و اهمیت متغیر را محاسبه کنید، زیرا الگوریتم شباهت را می‌توان با کار با مهمترین پیش‌بینی‌کننده‌ها کارآمدتر کرد. در اینجا ما یک مدل جنگل‌های تصادفی را تنظیم می‌کنیم و زیرمجموعه‌ای از داده‌های آموزشی و آزمایشی را با استفاده از 20 پیش‌بینی برتر (مرحله 1) برای گنجاندن در الگوریتم شباهت ایجاد می‌کنیم:

*set. seed (100)*

*mtryVals <- طبقه (seq(10، ncol(solTrainXtrans)، طول = 10))*

*mtryGrid <- data. frame(. mtry = mtryVals)*

*rfTune <- train(x = solTrainXtrans، y = solTrainY،*

*+ روش = "rf"،*

*+ tuneGrid = mtryGrid،*

*+ ntree = 1000،*

*+ اهمیت = درست،*

*+ trControl = ctrl)*

*ImportanceOrder <- order(rfTune$finalModel$importance[,1],*

*+ کاهش = TRUE)*

*top20 <- rownames(rfTune$finalModel$importance[ImportanceOrder,])[1:20]*

*solTrainXimp <- زیر مجموعه (solTrainX، انتخاب = top20)*

*solTestXimp <- زیر مجموعه (solTestX، انتخاب = top20)*

سپس زیرمجموعه پیش‌بینی‌کننده‌ها برای ایجاد مجموعه تصادفی جابجا می‌شوند. راه‌های زیادی برای جابجایی داده‌ها در R وجود دارد. یک راه ساده و مستقیم استفاده از توابع کاربردی و نمونه با هم است. سپس زیرمجموعه اصلی داده و مجموعه تغییر یافته با هم ترکیب می‌شوند و یک متغیر طبقه‌بندی جدید برای شناسایی عضویت هر ردیف ایجاد می‌شود. این مراحل 2-4 الگوریتم را تعریف می‌کند که می‌تواند به صورت زیر پیاده‌سازی شود:

*permutesolTrainXimp <- اعمال (solTrainXimp، 2، تابع (x) نمونه (x))*

*solSimX <- rbind(solTrainXimp، permutesolTrainXimp)*

*groupVals <- c("آموزش"، "تصادفی")*

*groupY <- فاکتور(rep(گروهVals، هر = nrow(solTrainX)))*

در نهایت، مدلی را روی داده‌های طبقه‌بندی جدید ایجاد می‌کنیم و از مدل برای پیش‌بینی احتمال عضویت در مجموعه آموزشی استفاده می‌کنیم.

*> rfSolClass <- train(x = solSimX، y = groupY،*

*+ روش = "rf"،*

*+ تنظیم طول = 5،*

*+ ntree = 1000،*

*+ کنترل = TrainControl (روش = "LGOCV"))*

*> solTestGroupProbs <- پیش‌بینی (rfSolClass، solTestXimp، نوع = "prob")*

تمرینات

شکل 20. 10 مقایسه‌ای را بین چندین مدل ارائه می‌دهد، زمانی که پاسخ به‌عنوان یک مقدار پیوسته و مقوله‌ای مدل می‌شود. محور *x* مقدار حلالیت مجموعه آزمایشی مشاهده شده را نشان می‌دهد و محور *y* مقدار حلالیت پیش‌بینی شده یا احتمال حلالیت پیش‌بینی شده را نشان می‌دهد.

بر اساس شکل، کدام مدل پیوسته بهترین (و بدترین) را در مجموعه تست دارد؟ کدام مدل طبقه‌بندی بهترین (و بدترین) را دارد؟

نتایج حاصل از مدل‌های شبکه عصبی را بررسی کنید. اگر احتمال پیش‌بینی‌شده ­برای یک ترکیب نزدیک به صفر باشد، محدوده مقادیر حلالیت واقعی برای مجموعه آزمایش چقدر است؟ اگر مقدار حلالیت پیش‌بینی شده (از ­مدل پیوسته) نزدیک به *-* 10 باشد، محدوده مقادیر واقعی انحلال چقدر است؟ کدام مدل (مقوله‌ای یا پیوسته) نتایج دقیق‌تری در این حد ارائه می‌دهد؟

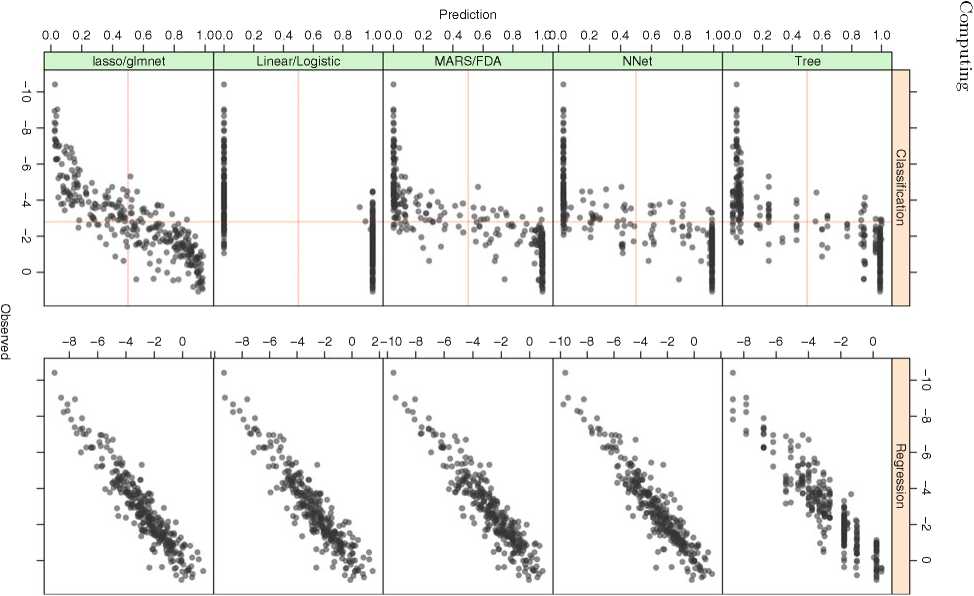
آیا هیچ یک از مدل‌های طبقه‌بندی بهتر از مدل‌های رگرسیونی مربوطه هستند؟ اگر بله، پس توضیح دهید.

همانطور که در بخش بحث شد.  [20. 4](#bookmark49) ، دسته‌بندی یک نتیجه پیوسته می‌تواند تأثیرات مخربی بر عملکرد مدل داشته باشد، به‌ویژه زمانی که توزیع دارای گروه‌بندی مجزا نباشد. با این حال، گاهی اوقات ممکن است یک دلیل ­منطقی برای ترکیب داده‌های پیوسته وجود داشته باشد. اگر اینطور باشد، اما داده‌ها دارای توزیعی مانند داده‌های حلالیت هستند (شکل 2).  [20. 5 )](#bookmark936) ، سپس یک گزینه ممکن برای مدلساز این است که پاسخی بسازد که داده‌ها را به سه گروه تقسیم کند که برای آنها اطمینان بالاتری به داده‌ها در نهایت وجود دارد. برای مثال، داده‌های حلالیت را می‌توان به گروه‌هایی مانند «Insol ­uble»، «Soluble» و «Indeterminate» تقسیم کرد که در آن گروه نامشخص، داده‌هایی است که حول میانگین پاسخ متمرکز شده‌اند.

داده‌های حلالیت را بارگذاری کنید و دو مجموعه آموزشی ایجاد کنید. برای اولین مجموعه، داده‌ها را در میانگین پاسخ تقسیم کنید. برای مجموعه دوم، اگر ترکیبات در یک انحراف استاندارد از میانگین پاسخ باشند، تعریف کنید که نامشخص باشند. این کار با کد زیر قابل انجام است:

*> کتابخانه (AppliedPredictiveModeling)*

*> داده (انحلال پذیری)*



*> trainData <- solTrainXtrans*

*> lowcut <- mean(solTrainY) - sd(solTrainY)*

*> highcut <- mean(solTrainY) + sd(solTrainY)*

*> نقاط شکست <- c(min(solTrainY)، lowcut، highcut، max(solTrainY))*

*> نام‌های گروه <- c("نامحلول"، "محدوده متوسط"، "محلول")*

*> solTrainY3bin <- cut(solTrainY,*

*+ شکست = نقاط شکست،*

*+ include. lowest = TRUE،*

*+ برچسب‌ها = نام گروه)*

*> solTestY3bin <- cut(solTestY,*

*+ شکست = نقاط شکست،*

*+ include. lowest = TRUE،*

*+ برچسب‌ها = نام گروه)*

یک مدل خطی، یک مدل غیرخطی و یک مدل مبتنی بر درخت را به داده‌های سه بانی برازش دهید. به‌عنوان مثال، کد زیر می‌تواند برای تولید یک مدل پارتیشن‌بندی بازگشتی استفاده شود:

*> set. seed(100)*

*> indx3bin <- createFolds(solTrainY3bin، returnTrain = TRUE)*

*> ctrl3bin <- trainControl(method = "cv",*

|  |  |
| --- | --- |
| *+*  *+* | *index = indx3bin، classProbs = TRUE،* |
| *+* | *savePredictions = TRUE)* |

*> Rpart3bin <- train(x = trainXfiltered, y = solTrainY3bin*

|  |  |
| --- | --- |
| *+*  *+* | *روش = "rpart"، معیار = "کاپا"،* |
| *+* | *طول کوک = 30،* |
| *+* | *trControl = ctrl3bin)* |

*>*

عملکرد مجموعه تست را با استفاده از معیار عملکرد انتخابی خود برای هر یک از مدل‌های (الف) پیش‌بینی کنید. کدام مدل برای داده‌های سه بانی بهترین عملکرد را دارد؟

اکنون داده‌های «MidRange» را از مجموعه آموزشی حذف کنید، هر مدل را بازسازی کنید و مجموعه آزمایشی را پیش‌بینی کنید. این را می‌توان با پارتیشن‌بندی بازگشتی ­به صورت زیر انجام داد:

*> trainXfiltered2bin <- trainXfiltered[solTrainY3bin != "MidRange"]*

*> solTrainY2bin <- solTrainY3bin[solTrainY3bin != "MidRange"]*

*> testXfiltered2bin <- testXfiltered[solTestY3bin != "MidRange"]*

*> solTestY2bin <- solTestY3bin[solTestY3bin != "MidRange"]*

*> set. seed(100)*

*> indx2bin <- createFolds(solTrainY2bin، returnTrain = TRUE)*

*> ctrl2bin <- trainControl(method = "cv",*

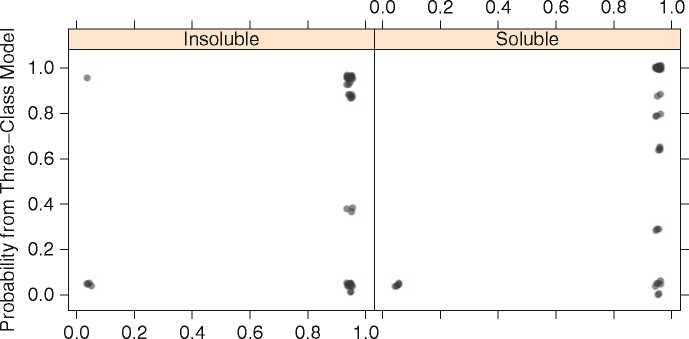
*+ index = indx2bin،*

*+ classProbs = TRUE،*

*+ savePredictions = TRUE)*

*> Rpart2bin <- train(x = trainXfiltered2bin، y = solTrainY2bin،*

*+ روش = "rpart"،*



**احتمال از مدل دو کلاسه**

شکل 20. 11: احتمالات پیش‌بینی شده از مدل‌های دو و سه کلاسه برای مجموعه آزمایشی از مدل پارتیشن‌بندی بازگشتی. به منظور مشاهده تعداد نسبی نمونه، احتمالات کمی مختل شده است

*+ معیار = "کاپا"،*

*+ آهنگ طول = 30،*

*+ trControl = ctrl2bin)*

*> Rpart2binPred <- پیش‌بینی (Rpart2bin، newdata = testXfiltered)*

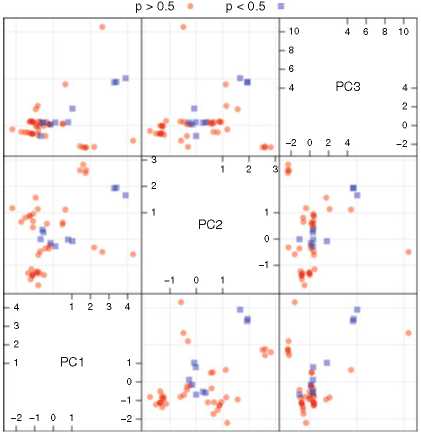
*> Rpart2binCM <- confusionMatrix (Rpart2binPred، solTestY3bin)*

چگونه حساسیت و ویژگی برای کلاس‌های نامحلول و محلول برای رویکردهای binning در (b) و (c) در هر مدل و بین مدل‌ها مقایسه می‌شود؟

شکل [20. 11](#bookmark954) احتمالات کلاس پیش‌بینی‌شده را از مدل‌های دو و سه بانی برای هر دو ترکیب مجموعه آزمایشی نامحلول و محلول مقایسه ­می‌کند. بر اساس احتمالات کلاس، آیا مدل دو یا سه کلاسه مزایای پیش‌بینی مشخصی برای ترکیبات نامحلول و/یا محلول ارائه می‌دهد؟ احتمالات طبقاتی پیش‌بینی‌شده در گروه‌های نامحلول و محلول چگونه با مدل‌های دیگری که توسعه داده‌اید مقایسه می‌شود؟

جزئیات محاسبه برای الگوریتم [20. 1](#bookmark943) اعمال شده برای داده‌های حلالیت در بخش ارائه شده است.  [20. 7 .](#bookmark950) این کد را اجرا کنید و توزیع احتمال عضویت مجموعه آموزشی نمونه‌های مجموعه تست را رسم کنید. بعید است که چند نمونه از مجموعه تست بخشی از توزیع مجموعه آموزشی بوده باشند؟

ورزش [4. 4](#bookmark7) مجموعه داده‌ای را توصیف می‌کند که در آن روغن‌های غذایی برای محتوای هفت نوع اسید چرب مورد تحلیل قرار گرفتند. داده‌ها را بارگذاری کنید و یک مجموعه آموزشی و آزمایشی به شرح زیر ایجاد کنید:



ماتریس طرح پراکندگی

شکل 20. 12: طرح آزمایش روغن بر روی سه جزء اصلی اول تنظیم می‌شود. نمونه‌ها بر اساس احتمال عضویت در مجموعه آموزشی رنگ و شکل می‌گیرند

*> داده (روغن)*

*> set. seed(314)*

*> SampleRows <- sample. int(nrow(fattyAcids)، اندازه = 0. 5\*nrow(fattyAcids))*

*> fattyAcidsTrain <- fattyAcids[sampleRows,]*

*> fattyAcidsTest <- fattyAcids[-sampleRows,]*

اجرای الگوریتم [20. 1](#bookmark943) با استفاده از مجموعه آموزشی. کدام نمونه از مجموعه آزمایشی احتمالاً عضو مجموعه آموزشی نیستند؟ چرا این نمونه‌ها احتمالاً عضو مجموعه آموزشی نیستند؟

شکل [20. 12](#bookmark955) پیش‌بینی داده‌های مجموعه آزمایشی را بر روی سه جزء اول PCA که توسط مجموعه آموزشی تعیین شده است، ارائه می‌کند. نمونه‌ها بر اساس احتمال عضویت در مجموعه آموزشی رنگ و شکل می‌گیرند. این طرح در مورد مکان نمونه‌هایی که احتمالاً عضو مجموعه آموزشی نیستند، چه چیزی را نشان می‌دهد؟

چه اقداماتی می‌تواند انجام شود تا اطمینان حاصل شود که مجموعه‌های آموزشی و آزمایشی همان منطقه فضای پیش‌بینی را پوشش می‌دهند؟

ضمیمه

پیوست اول

خلاصه‌ای از مدل‌های مختلف

جدول A. 1 خلاصه‌ای کوتاه از چندین ویژگی مدل‌های مورد بحث در اینجا را نشان می‌دهد. این ویژگی‌ها به‌طور کلی پابرجا هستند، اما همیشه برای هر مسأله‌ای صادق نیستند. به‌عنوان مثال، مدل‌های تحلیل متمایز خطی، انتخاب ویژگی‌های فرم را انجام نمی‌دهند، اما نسخه‌های تخصصی مدل وجود دارد که از منظم‌سازی برای حذف پیش‌بینی‌کننده‌ها استفاده می‌کند. همچنین، تفسیرپذیری یک مدل ذهنی است. یک درخت ممکن است قابل درک باشد اگر بیش از حد بزرگ نباشد و شکاف‌ها شامل تعداد زیادی دسته نباشند.

همانطور که در فصل بیان شد.  [2](#bookmark107) ، هیچ مدلی به‌طور یکنواخت بهتر از بقیه نیست. کاربرد یک تکنیک به نوع داده‌های مورد تحلیل، نیازهای مدل ساز و زمینه نحوه استفاده از مدل بستگی دارد.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| مدل | اجازه می‌دهد تا *n < p* | پیش پردازش | انتخاب ویژگی خودکار قابل تفسیر | | # تنظیم پارامترها | مقاوم در برابر نویزهای پیش‌بینی | زمان محاسبه |
| رگرسیون خطی | ایکس | CS، NZV، Corr | / | ایکس | 0 | ایکس | / |
| حداقل مربعات جزئی | / | CS | / | o | 1 | ایکس | / |
| رگرسیون ریج | ایکس | CS، NZV | / | ایکس | 1 | ایکس | / |
| شبکه الاستیک / ریج | ایکس | CS، NZV | / | / | 1-2 | ایکس | / |
| شبکه‌های عصبی | / | CS، NZV، Corr | ایکس | ایکس | 2 | ایکس | ایکس |
| ماشین‌های بردار پشتیبانی می‌کند | / | CS | ایکس | ایکس | 1-3 | ایکس | ایکس |
| مریخ / FDA | / |  | o | / | 1-2 | o | o |
| K-نزدیک‌ترین همسایگان | / | CS، NZV | ایکس | ایکس | 1 | o | / |
| تک درختان | / |  | o | / | 1 | / | / |
| درختان/قوانین مدل^ | / |  | o | / | 1-2 | / | / |
| درختان کیسه ای | / |  | ایکس | / | 0 | / | o |
| جنگل تصادفی | / |  | ایکس | o | 0-1 | / | ایکس |
| درختان تقویت شده | / |  | ایکس | / | 3 | / | ایکس |
| کوبیست تی | / |  | ایکس | o | 2 | / | ایکس |
| رگرسیون لجستیک\* | ایکس | CS، NZV، Corr | / | ایکس | 0 | ایکس | / |
| {LQRM}DA\* | ایکس | NZV | o | ایکس | 0-2 | ایکس | / |
| نزدیکترین مرکزهای کوچک شده\* | / | NZV | o | / | 1 | ایکس | / |
| بیز ساده\* | / | NZV | ایکس | ایکس | 0-1 | o | o |
| C5. 0\* | / |  | o | / | 0-3 | / | ایکس |

1 فقط رگرسیون \* فقط طبقه بندی

نمادها نشان دهنده مثبت (A)، منفی (x) و جایی در بین (o) هستند.

550 خلاصه‌ای از مدل‌های مختلف

Table A.l: A summary of models and some of their characteristics

ضمیمه B

مقدمه‌ای بر R

زبان R [(](#bookmark1019) Ihaka [and Gentleman 1996](#bookmark1019) ; [تیم اصلی توسعه R](#bookmark1024) [2010](#bookmark1024) ) بستری برای محاسبات ریاضی و آماری است. از دو جهت رایگان است. اول، R را می‌توان رایگان به دست آورد (اگر چه نسخه‌های تجاری وجود دارد). دوم، هر کسی می‌تواند کد منبع را بررسی یا تغییر دهد. R تحت *مجوز عمومی عمومی منتشر شده است* [( بنیاد نرم‌افزار آزاد](#bookmark1016) ژوئن 2007 ) که نحوه توزیع مجدد برنامه را تشریح می‌کند.

R به دلایل متعدد در این کتاب به‌طور گسترده استفاده شده است. همانطور که ذکر شد، هر کسی می‌تواند این برنامه را دانلود و استفاده کند. دوم، R یک ابزار بسیار قدرتمند ­و انعطاف‌پذیر برای تحلیل داده‌ها است و دارای قابلیت‌های گسترده‌ای برای مدل‌سازی پیش‌بینی است.

وب سایت Comprehensive R Archive Network (CRAN) حاوی کد منبع برنامه و همچنین نسخه‌های کامپایل شده‌ای است که آماده استفاده هستند:

[http://cran. r-project. org/](http://cran.r-project.org/)

این ضمیمه در نظر گرفته شده است تا یک دوره خراب در مفاهیم اولیه و نحو برای R باشد. راهنماهای عمیق تر برای مبانی زبان هستند [اسپکتور](#bookmark1025) [( 2008](#bookmark1025) ) و [جنتلمن](#bookmark1016) [( 2008](#bookmark1016) ). چرخه عمر توسعه نرم‌افزار در R Development [Core](#bookmark1024) Team [( 2008](#bookmark1024) ) به تفصیل آمده است.

B. 1 راه اندازی و دریافت کمک

CRAN شامل نسخه‌های از پیش کامپایل شده R برای Microsoft Windows، Apple OS X و چندین نسخه لینوکس است. برای ویندوز و OS X، این برنامه دارای یک رابط کاربری گرافیکی (GUI) است. هنگام نصب نسخه‌های سازگار R برای این دو سامانه عامل، نماد R بر روی رایانه نصب می‌شود. برای شروع یک جلسه تعاملی، برنامه را با استفاده از نماد راه اندازی کنید. از طرف دیگر، R را می‌توان در خط فرمان با تایپ R شروع کرد.

پس از شروع برنامه، تابع q (برای خروج) جلسه را پایان می‌دهد.

*> # نظرات بعد از نمادهای " # " رخ می‌دهند و* اجرا نمی‌شوند

*> # برای خروج از این دستور استفاده کنید*

*> q()*

هنگام ترک، از کاربر گزینه‌هایی برای ذخیره کار فعلی خود خواسته می‌شود. توجه داشته باشید که زبان به حروف کوچک و بزرگ حساس است: Q نمی‌تواند برای خروج از جلسه استفاده شود.

برای دریافت راهنمایی در مورد یک موضوع خاص، مانند یک تابع، یک علامت سوال قبل از تابع قرار دهید و اینتر را فشار دهید:

*> # در مورد عملکرد Sweave کمک بگیرید*

*> ?سوو*

با این کار صفحه راهنمای Sweave باز می‌شود. یکی از چالش‌های رایج با R، یافتن یک تابع مناسب است. برای جستجو در تمام توابع R محلی در رایانه خود، apropos یک کلمه کلیدی را با توابع موجود مطابقت می‌دهد:

*> apropos ("prop")*

[1] "مناسب"

"getProperties"

"power.prop.test"

"prop.test"

"reconcilePropertiesAndPrototype

[3] "pairwise. prop. test"

[5] "prop. table"

[7] "prop. trend. test"

، تابع RSiteSearch جستجوی آنلاین همه عملکردها، راهنماها، اسناد کمکی، گروه خبری R-Help و سایر منابع را برای یک کلمه کلیدی انجام می‌دهد. به‌عنوان مثال، برای جستجوی روش‌های مختلف برای تولید منحنی‌های ROC،

*> RSiteSearch ("roc")*

یک مرورگر وب را باز می‌کند و مسابقات را نشان می‌دهد. آرگومان محدود این تابع جستجو را گسترده می‌کند ( برای جزئیات بیشتر به ?RSiteSearch مراجعه کنید).

B. 2 بسته ها

*Base R* سامانه اسمی است که ویژگی‌های زبان اصلی را در بر می‌گیرد (به‌عنوان مثال، برنامه اجرایی، چارچوب برنامه نویسی اساسی). بیشتر کد R واقعی در ماژول‌های مجزایی به نام *بسته‌ها موجود* است. هنگامی که R نصب می‌شود، مجموعه کوچکی از بسته‌های هسته اصلی نیز نصب می‌شود ( [برای](#bookmark1024) لیست قطعی، تیم اصلی R De ­velopment [( 2008 ) را ببینید).](#bookmark1024) با این حال، تعداد زیادی بسته خارج از این مجموعه وجود دارد. وب سایت CRAN حاوی بیش از 4150 بسته برای دانلود است در حالی که پروژه Bioconductor [( Gentleman et al.](#bookmark1016)  [2004](#bookmark1016) )، یک سامانه مبتنی بر R برای زیست شناسی محاسباتی، شامل بیش از 600 بسته R است.

برای بارگذاری یک بسته، از تابع کتابخانه استفاده می‌شود:

*# بسته جنگل‌های تصادفی را بارگیری کنید*

*کتابخانه (جنگل تصادفی)*

*> # لیست بسته‌های بارگذاری شده فعلی و سایر اطلاعات را نشان دهید*

*> sessionInfo()*

R نسخه 2. 15. 2 (26-10-2012)

پلتفرم: x86\_64-apple-darwin9. 8. 0/x86\_64 (64 بیت)

محل:

[1] سی

|  |  |  |  |
| --- | --- | --- | --- |
| بسته‌های پایه پیوست شده:  ابزارهای splines  [8] پایه روش | گرافیک آماری | grDevices Utils | مجموعه داده ها |
| سایر بسته‌های پیوست شده  [1] randomForest\_4. 6-7  [4] foreach\_1. 4. 0  [7] plyr\_1. 7. 1  [10] survival\_2. 36-14  [13] digest\_0. 6. 0 | BiocInstaller\_1. 8. cluster\_1. 14. 3 lattice\_0. 20-10 weaver\_1. 24. 0 | 3 caret\_5. 15-045 reshape\_0. 8. 4 Hmisc\_3. 10-1 codetools\_0. 2-8 |  |
| بارگیری شده از طریق فضای نام (و پیوست نشده):  [1] grid\_2. 15. 2 iterators\_1. 0. 6 | |  |  |

تابع install. packages را می‌توان برای نصب ماژول‌های اضافی استفاده کرد. به‌عنوان مثال، برای نصب بسته rpart برای درختان طبقه‌بندی و رگرسیون که در Sects بحث شده است.  [8. 1](#bookmark392) و [14. 1](#bookmark686) ، کد

*install. packages ("rpart")*

میتواند مورد استفاده قرار گیرد. از طرف دیگر، وب‌سایت CRAN شامل «نماهای وظیفه» است که بسته‌های مشابه را با هم گروه‌بندی می‌کند. به‌عنوان مثال، نمای کار برای "آموزش ماشین" مجموعه‌ای از بسته‌های مدل‌سازی پیش‌بینی را نصب می‌کند:

*# ابتدا بسته task view را نصب کنید*

*install. packages ("ctv")*

*# کتابخانه را قبل از اولین استفاده بارگیری کنید*

*کتابخانه (ctv)*

*install. views ("MachineLearning")*

برخی از بسته‌ها به بسته‌های دیگر (یا نسخه‌های خاص) بستگی دارند. توابع ­install. packages و install. views نیازمندی‌های اضافی بسته را تعیین می‌کنند و وابستگی‌های لازم را نصب می‌کنند.

ب. 3 ایجاد اشیا

هر چیزی که در R ایجاد شود یک *شی است.* با استفاده از " <- " می‌توان به اشیا مقادیر اختصاص داد. مثلا:

*صفحات <- 97*

*شهر <- "ریچموند"*

*## Equals نیز کار می‌کند، اما بخش B. 9 را در زیر ببینید*

برای دیدن مقدار یک Object کافی است آن را تایپ کرده و اینتر را بزنید. همچنین، می‌توانید صریحاً به R بگویید که مقدار شی را چاپ کند

*> صفحات*

[1] 97

*> چاپ (شهر)*

[1] "ریچموند"

یکی دیگر از عملکردهای مفید برای درک محتوای یک شی str (برای ساختار) است. به‌عنوان مثال، R به‌طور خودکار با یک شی همراه می‌شود که حاوی نام اختصاری ماه است.

*> month. abb*

[1] "ژان" "فوریه" "مار" "آوریل" "مه" "ژوئن" "ژوئیه" "اوت" "سپتامبر" "اکتبر" "نوامبر"

[12] "دسامبر"

*str (month. abb)*

chr [1:12] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul". . .

این نشان می‌دهد که month. abb یک موضوع شخصیت با دوازده عنصر است. ما همچنین می‌توانیم ساختار اشیایی را که حاوی داده نیستند، مانند تابع چاپی که قبلاً بحث شد، تعیین کنیم:

*خیابان (چاپ)*

تابع (x،. . . )

*str(sessionInfo)*

تابع (بسته = NULL)

این برای جستجوی نام آرگومان‌های تابع مفید است. توابع در زیر با جزئیات بیشتر مورد بحث قرار خواهند گرفت.

B. 4 انواع داده‌ها و ساختارهای اساسی

انواع مختلفی از داده‌های اصلی در R وجود دارد. انواع مربوطه عبارتند از انواع عددی، کاراکتری، عاملی و منطقی. داده‌های منطقی می‌توانند مقدار TRUE یا FALSE را بگیرند. به‌عنوان مثال، این مقادیر را می‌توان برای مقایسه استفاده کرد یا می‌توان به یک شی اختصاص داد:

*if(3 > 2) print("بیشتر") else print("کمتر")* [1] "بزرگتر"

*isGreater <- 3 > 2*

*بزرگتر است*

درست است، واقعی

*> is. logical(isGreater)*

[1] درست است

داده‌های عددی شامل اعداد صحیح و اعداد با دقت مضاعف (یعنی اعشاری با ارزش اعشاری) است. برای اختصاص یک مقدار عددی واحد به یک شی R :

*>x<-3. 6*

*> is. numeric(x)*

[1] درست است

*> is. integer(x)*

[1] نادرست

*> is. double(x)*

[1] درست است

*> typeof(x)*

[1] "دو"

رشته‌های کاراکتر را می‌توان با قرار دادن متن در داخل نقل قول ایجاد کرد:

*y <- "تبلیغ شما در اینجا"*

*نوع (y)*

[1] "شخصیت"

*> z <- "شما می‌توانید متن را نیز* ' *نقل قول* ' *کنید" >z*

[1] "شما همچنین می‌توانید متن را "نقل قول" کنید"

توجه داشته باشید که R طول رشته کاراکترها را محدود نمی‌کند.

چندین تابع مفید وجود دارد که روی رشته‌ها کار می‌کنند. ابتدا کاراکتر تعداد کاراکترها را می‌شمارد:

*> nchar(y)*

[1] 12

*> nchar(z)*

[1] 29

تابع grep می‌تواند برای تعیین اینکه آیا رشته فرعی در ­رشته کاراکتر کاراکتر وجود دارد یا خیر استفاده می‌شود

*> grep("ad"، y)*

[1] 1

*> grep("my"، y)*

عدد صحیح (0)

*# اگر رشته وجود دارد، کل مقدار را برگردانید*

*grep ("too"، z، value = TRUE)*

[1] "شما همچنین می‌توانید متن را "نقل قول" کنید"

تاکنون، اشیاء R نشان داده شده دارای یک مقدار یا عنصر واحد هستند. اساسی‌ترین ساختار داده برای نگهداری مقادیر چندگانه از یک نوع داده یک بردار است. ابتدایی‌ترین روش ایجاد بردار استفاده از تابع c (برای *ترکیب* ) است. برای ایجاد بردار داده‌های عددی:

*وزن‌ها <- c(90، 150، 111، 123)*

*is. vector(وزن ها)*

درست است، واقعی

*> نوع (وزن)*

[1] "دو"

*> طول (وزن)*

[1] 4

*> وزن + 0. 25*

[1] 90. 25 150. 25 111. 25 123. 25

توجه داشته باشید که دستور آخر نمونه‌ای از *عملیات برداری* است. به جای حلقه زدن بر روی عناصر بردار، عملیات بردار عملیات مختصرتر و کارآمدتر است.

بسیاری از توابع روی بردارها کار می‌کنند:

*> میانگین (وزن)*

[1] 118. 5

*رنگ‌ها <- c("سبز"، "قرمز"، "آبی"، "قرمز"، "سفید")*

*grep ("قرمز"، رنگ ها)*

[1] 2 4

*> nchar (رنگ ها)*

[1] 5 3 4 3 5

یک روش جایگزین برای ذخیره داده‌های کاراکتر در یک بردار استفاده از *فاکتورها* است. فاکتورها با تعیین تمام مقادیر منحصر به فرد در داده‌ها که سطوح عامل نامیده می‌شوند، داده‌های کاراکتر را ذخیره می‌کنند. سپس داده‌های کاراکتر به‌عنوان اعداد صحیح ذخیره می‌شوند که با سطوح فاکتور مطابقت دارند:

*> color2 <- as. factor(colors) > color2*

[1] سبز قرمز آبی قرمز سفید

سطوح: آبی سبز قرمز سفید

*> سطوح (colors2)*

[1] "آبی" "سبز" "قرمز" "سفید"

*> as. numeric(colors2)*

[1] 2 3 1 3 4

چند مزیت برای ذخیره داده‌ها در فاکتورها وجود دارد. اول، حافظه کمتری برای ذخیره مقادیر مورد نیاز است، زیرا رشته‌های کاراکتر طولانی بالقوه فقط یک بار (در سطوح) ذخیره می‌شوند و وقوع آنها به‌عنوان بردار ذخیره می‌شوند. دوم، بردار عامل تمام مقادیر ممکن را "به خاطر می‌آورد". فرض کنید بردار عامل را با حذف اولین مقدار با استفاده از یک عدد صحیح منفی زیر مجموعه کنیم:

*> رنگ2[-1]*

[1] قرمز آبی قرمز سفید

سطوح: آبی سبز قرمز سفید

حتی اگر عنصر با مقدار "سبز" حذف شد، عامل همچنان همان سطوح را حفظ می‌کند. فاکتورها ابزار اولیه ذخیره متغیرهای گسسته در R هستند و بسیاری از مدل‌های طبقه‌بندی از آنها برای مشخص کردن داده‌های نتیجه استفاده می‌کنند.

برای کار با زیرمجموعه‌ای از یک بردار، تک براکت‌ها را می‌توان به روش‌های مختلفی استفاده کرد:

*> وزنه ها*

[1] 90 150 111 123

*# اعداد صحیح مثبت نشان می‌دهد که کدام عناصر را نگه دارید*

*وزن [c(1، 4)]*

90 123

*# عدد صحیح منفی با عناصری که باید حذف شوند مطابقت دارند*

*وزن [-c(1, 4)]*

150 111

*# بردار مقادیر منطقی را نیز می‌توان استفاده کرد اما باید وجود داشته باشد*

*# به اندازه عناصر مقدار منطقی باشد*

*وزن [c(TRUE، TRUE، FALSE، TRUE)]*

90 150 123

بردارها باید همان نوع داده را ذخیره کنند. یک جایگزین یک لیست است. این یک نوع برداری است که می‌تواند اشیاء از هر نوع را به‌عنوان عناصر ذخیره کند:

*هر دو <- لیست (رنگ = رنگ2، وزن = وزن)*

*is. vector (هر دو)*

[1] درست است

*> is. list (هر دو)*

[1] درست است

*> طول (هر دو)*

[1] 2

*> نام‌ها (هر دو)*

[1] "رنگ ها" "وزن"

لیست‌ها را می‌توان به روشی مشابه بردارها فیلتر کرد. با این حال، دو براکت فقط عنصر را برمی‌گرداند، در حالی که تک براکت‌ها فهرست دیگری را برمی‌گردانند:

*> هر دو[[1]]*

[1] سبز قرمز آبی قرمز سفید

سطوح: آبی سبز قرمز سفید

*> is. list (هر دو[[1]])*

[1] نادرست

*> هر دو[1]*

$ رنگ

[1] سبز قرمز آبی قرمز سفید

سطوح: آبی سبز قرمز سفید

*> is. list (هر دو[1])*

درست است، واقعی

*> # ما همچنین می‌توانیم با استفاده از نام لیست زیر مجموعه ها*

*> هر دو[["رنگ"]]*

سبز قرمز آبی قرمز سفید

سطوح: آبی سبز قرمز سفید

مقادیر از دست رفته در R به‌عنوان مقادیر NA کدگذاری می‌شوند :

*احتمالات <- c(. 05، 0. 67، NA، 0. 32، 0. 90)*

*is. na (احتمالات)*

FALSE FALSE TRUE FALSE FALSE

*# NA به‌عنوان یک رشته کاراکتر در نظر گرفته نمی‌شود*

*احتمالات == "NA"*

FALSE FALSE NA FALSE FALSE

*# اکثر توابع مقادیر گم شده را منتشر می‌کنند. . .*

*میانگین (احتمالات)*

NA

*#. . . مگر اینکه خلاف آن گفته شود*

*میانگین (احتمالات، na. rm = TRUE)*

[1] 0. 485

B. 5 کار با مجموعه داده‌های مستطیلی

مجموعه داده‌های مستطیلی معمولاً به موقعیت‌هایی اشاره می‌کنند که در آن نمونه‌ها در ردیف‌هایی از یک مجموعه داده قرار دارند در حالی که ستون‌ها با متغیرها مطابقت دارند (در برخی حوزه‌ها، این قرارداد برعکس است). دو ساختار اصلی برای داده‌های مستطیلی وجود دارد: ماتریس‌ها و فریم‌های داده. تفاوت اصلی بین این دو نوع شی در نوع داده‌ای است که می‌توان در آنها ذخیره کرد. یک ماتریس فقط می‌تواند حاوی داده‌هایی از همان نوع باشد (مثلاً کاراکتر یا عددی) در حالی که فریم‌های داده باید شامل ستون‌هایی از همان نوع داده باشند. ماتریس‌ها از نظر محاسباتی ­کارآمدتر هستند اما به وضوح محدود هستند.

ما می‌توانیم با استفاده از تابع ماتریس یک ماتریس ایجاد کنیم. در اینجا یک بردار عددی از اعداد صحیح از یک تا دوازده ایجاد می‌کنیم و از سه ردیف و چهار ستون استفاده می‌کنیم:

*> mat <- matrix(1:12، nrow = 3) > mat*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | [,1] | [,2] | [,3] | [,4] |
| [1،] | 1 | 4 | 7 | 10 |
| [2،] | 2 | 5 | 8 | 11 |
| [3،] | 3 | 6 | 9 | 12 |

سطرها و ستون‌ها را می‌توان نام داد:

*> rownames(mat) <- c("ردیف 1"، "ردیف 2"، "ردیف 3")*

*> colnames(mat) <- c("col1"، "col2"، "col3"، "col4") > mat*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | col1 | col2 | col3 | col4 |
| ردیف | 1 | 1 | 4 | 7 | 10 |
| ردیف | 2 | 2 | 5 | 8 | 11 |
| ردیف | 3 | 3 | 6 | 9 | 12 |

*> نام ردیف ها*

[1] "ردیف 1" "ردیف 2" "ردیف 3"

ماتریس‌ها را می‌توان با استفاده از روش‌هایی مشابه بردارها زیرمجموعه قرار داد، اما ردیف‌ها و ستون‌ها را می‌توان به‌طور جداگانه زیر مجموعه قرار داد:

*> mat [1, 2:3]*

col2 col3

47

*> mat["ردیف 1"، "col3"]*

[1] 7

*تشک[1،]*

col1 col2 col3 col4

14710

یکی از مسائل زیر مجموعه ماتریس‌ها این است که می‌توان ابعاد را *حذف کرد.* اگر یک سطر یا یک ستون با زیر مجموعه یک ماتریس تولید شود، یک بردار نتیجه است:

*is. matrix(mat[1,])*

نادرست

*> is. vector(mat[1,])*

[1] درست است

یک روش برای جلوگیری از این امر این است که هنگام زیرمجموعه، گزینه drop را به ماتریس منتقل کنید:

*تشک[1،]*

col1 col2 col3 col4

14710

*mat[1,, drop = FALSE]*

col1 col2 col3 col4

ردیف 114710

*is. matrix(mat[1,,drop = FALSE])*

درست است، واقعی

*> is. vector(mat[1,,drop = FALSE])*

نادرست

فریم‌های داده را می‌توان با استفاده از تابع data. frame ایجاد کرد :

*df <- data. frame (colors = color2, + time = 1:5)*

|  |  |  |
| --- | --- | --- |
| *df* | رنگ ها | زمان |
| 1 | سبز | 1 |
| 2 | قرمز | 2 |
| 3 | آبی | 3 |
| 4 | قرمز | 4 |
| 5 | سفید | 5 |

*کم نور (df)*

[1] 5 2

*> colnames(df)*

[1] "رنگ ها" "زمان"

*> نام ردیف (df)*

[1] "1" "2" "3" "4" "5"

علاوه بر تکنیک‌های زیرمجموعه‌ای که قبلاً برای ماتریس‌ها نشان داده شده بود، عملگر $ می‌تواند برای برگرداندن ستون‌های منفرد استفاده شود، در حالی که تابع زیر مجموعه می‌تواند برای برگرداندن زیرمجموعه‌های پیچیده‌تر از ردیف‌ها استفاده شود:

*> df$colors*

سبز قرمز آبی قرمز سفید

سطوح: آبی سبز قرمز سفید

*> زیر مجموعه (df، رنگ‌ها %in% c("قرمز"، "سبز") و زمان <= 2)*

زمان رنگ ها

1 سبز 1

2قرمز2

یک تابع مفید برای تعیین اینکه آیا مقادیری از دست رفته در ردیفی از یک ماتریس یا فریم داده وجود دارد یا خیر، تابع full. cases است که در صورت وجود هیچ مقادیری، TRUE را برمی گرداند :

*df2 <- df*

*# مقادیر گم شده را به قاب داده اضافه کنید*

*df2[1, 1] <- NA*

*df2[5, 2] <- NA*

*df2*

|  |  |  |
| --- | --- | --- |
|  | رنگ ها | زمان |
| 1 | <NA> | 1 |
| 2 | قرمز | 2 |
| 3 | آبی | 3 |
| 4 | قرمز | 4 |
| 5 | سفید | NA |

*full. cases(df2)*

[1] FALSE TRUE TRUE TRUE FALSE

ب. 6 اشیاء و کلاسها

هر شی حداقل یک نوع یا *کلاس* مرتبط با آن دارد. کلاس یک شی مشخص می‌کند که چیست (به‌عنوان مثال، یک رشته کاراکتر، مدل خطی، URL وب سایت). کلاس ساختار یک شی (یعنی نحوه ذخیره آن) و عملیات ممکن مرتبط با این نوع شی (که *متدهای* کلاس نامیده می‌شود) را تعریف می‌کند. به‌عنوان مثال، اگر نوعی شی مدل ایجاد شود، ممکن است مورد توجه قرار گیرد:

برای درک جزئیات مدل را *چاپ کنید*

*مدل* را برای تجسم ترسیم کنید، یا

نمونه‌های جدید را *پیش‌بینی کنید*

در این مورد، چاپ، رسم و پیش‌بینی برخی از روش‌های ممکن برای آن نوع خاص از مدل هستند (که توسط کلاس آن تعیین می‌شود). این پارادایم برنامه نویسی شی گرا نامیده می‌شود.

ما می‌توانیم به سرعت کلاس اشیاء قبلی را تعیین کنیم:

*> صفحات*

[1] 97

*> کلاس (صفحات)*

[1] "عددی"

*> شهر*

[1] "ریچموند" *> کلاس (شهر)*

"شخصیت"

هنگامی که کاربر R را برای انجام برخی عملیات، مانند ایجاد ­پیش‌بینییی از یک شی مدل، هدایت می‌کند، کلاس کد خاصی را برای معادله پیش‌بینی تعیین می‌کند. به این *روش ارسال روش می‌گویند.* دو تکنیک اصلی ­برای برنامه نویسی شی گرا در R وجود دارد : کلاس‌های S3 و کلاس‌های S4. رویکرد S3 ساده تر از S4 است و توسط بسیاری از بسته‌ها استفاده می‌شود. روش‌های S4 قوی‌تر از روش‌های S3 هستند، اما بسیار پیچیده‌تر از آن هستند که به‌طور کامل در این بررسی کلی توضیح داده شوند.  [چمبرز ( 2008](#bookmark1013) ) این تکنیک‌ها را با جزئیات بیشتری توصیف می‌کند.

در روش S3، قرارداد نامگذاری استفاده از نقطه برای جداسازی کلاس‌ها و متدها است. برای مثال، summary. lm تابعی است که برای محاسبه مقادیر خلاصه برای یک شی که دارای کلاس lm است استفاده می‌شود (این کلاس برای برازش مدل‌های خطی، مانند تحلیل رگرسیون خطی است). فرض کنید کاربری با استفاده از تابع lm یک Ob ject به نام myModel ایجاد کرده است. فرمان

modelSummary <- summary(myModel)

آمار توصیفی رایج برای مدل را محاسبه می‌کند. R می‌بیند که myModel دارای کلاس lm است، بنابراین کد را در تابع summary. lm اجرا می‌کند.

برای این متن، درک مفهوم اشیا، کلاس‌ها و متدها مهم است. با این حال، این مفاهیم در سطح بالایی استفاده خواهند شد. کد موجود در کتاب به ندرت به جزئیات فنی "زیر کاپوت" می‌پردازد. به‌عنوان مثال، تابع پیش‌بینی به‌طور گسترده مورد استفاده قرار می‌گیرد، اما استفاده از آن برای دانستن اینکه کدام روش خاص اجرا شده است، مورد نیاز نخواهد بود.

توابع R

در R، قطعات کد مدولار را می‌توان در توابع جمع‌آوری کرد. بسیاری از توابع قبلاً در این بخش استفاده شده اند، مانند تابع کتابخانه که بسته‌ها را بارگیری می‌کند. توابع دارای *آرگومان هستند* : اسلات‌های خاصی که برای ارسال اشیا به تابع استفاده می‌شوند. در R، آرگومان‌ها نامگذاری می‌شوند (برخلاف سایر زبان‌ها، مانند matlab ). به‌عنوان مثال، تابع برای خواندن داده‌های ذخیره شده در فرمت محدود شده با کاما (CSV) در یک شی R دارای این آرگومان‌ها است:

*> str(read. csv)*

تابع (پرونده، هدر = TRUE، sep = "،"، نقل قول = ""، dec = ". "، fill = TRUE، comment. char = ""،. . . )

جایی که فایل یک رشته کاراکتر است که به فایل CSV اشاره می‌کند و هدر نشان می‌دهد ­که ردیف اولیه با نام متغیرها مطابقت دارد یا خیر. آرگومان فایل هیچ مقدار پیش‌فرضی ­ندارد و اگر نام فایلی مشخص نشود، این تابع با خطا مواجه می‌شود. از آنجایی که این توابع نامگذاری شده اند، می‌توان آنها را به چند روش مختلف فراخوانی کرد:

*> read. csv("data. csv")*

*> read. csv (سرصفحه = FALSE، فایل = "data. csv")*

توجه داشته باشید که تابع read. csv در پایان یک آرگومان با سه نقطه دارد. این بدان معنی است که آرگومان‌های دیگری را می‌توان به فراخوانی تابع read. csv اضافه کرد که به یک تابع خاص در کد read. csv ارسال می‌شوند. در این حالت کد از تابع دیگری به نام read. table استفاده می‌کند که عمومیت بیشتری دارد. read. table حاوی آرگومانی به نام na. strings است که در read. csv وجود ندارد. این آرگومان به R می‌گوید که کدام مقادیر کاراکتر یک مقدار گم شده را در فایل نشان می‌دهد. استفاده كردن

*> read. csv("data. csv", na. strings = "?")*

اثر ارسال آرگومان na. strings = "?" از تابع read. csv تا تابع read. table. توجه داشته باشید که اگر قرار است این آرگومان عبور داده شود باید نامگذاری شود. این سه نقطه به‌طور گسترده در بخش‌های محاسباتی هر فصل استفاده می‌شود.

سه چهره =

تا کنون، نماد = در چندین زمینه مختلف استفاده شده است:

ایجاد اشیا، مانند x=3

تست برای هم ارزی: x==4

تعیین مقادیر برای آرگومان‌های تابع: read. csv(header = FALSE)

این می‌تواند برای تازه واردان گیج کننده باشد. مثلا:

*> جدید = زیر مجموعه (قدیمی، زیر مجموعه = ارزش == "آبی"، قطره = نادرست)*

در هر سه حالت چهار بار از نماد استفاده می‌کند. یک روش برای جلوگیری از سردرگمی استفاده از <- به‌عنوان عملگر انتساب است.

B. 9 بسته AppliedPredictiveModeling

این بسته به‌عنوان یک همراه برای کتاب عمل می‌کند و شامل بسیاری از مجموعه داده‌های مورد استفاده در اینجا است که قبلاً در سایر بسته‌های R موجود نیستند. همچنین شامل کد R مورد استفاده در سر فصل‌ها و توابع R می‌باشد. بسته در CRAN موجود است.

جدول B. 1: بررسی دستورات برای تولید احتمالات کلاس در ­بسته‌های مختلف

|  |  |  |
| --- | --- | --- |
| کلاس شیء | بسته | پیش‌بینی نحو تابع |
| lda | جرم | پیش‌بینی (شیء) (بدون گزینه مورد نیاز) |
| glm | آمار | پیش‌بینی (شیء، نوع = "پاسخ") |
| gbm | gbm | پیش‌بینی (شیء، نوع = "پاسخ"، n. درخت) |
| mda | mda | پیش‌بینی (شیء، نوع = "پسین") |
| بخش | بخش | پیش‌بینی (شیء، نوع = "مسأله") |
| Weka\_classifier | RWeka | پیش‌بینی (شیء، نوع = "احتمال") |
| LogitBoost | caTools | پیش‌بینی (شیء، نوع = "خام"، nIter) |

تابع آموزش در بسته caret از یک نحو رایج پیش‌بینی (object, type = "prob") استفاده می‌کند.

B. 10 بسته گاریت

بسته caret (مخفف C lassification A nd REgression T raining ) برای ساده‌سازی فرآیند ساخت و ارزیابی مدل‌های پیش‌بینی ایجاد ­شد. با استفاده از بسته، یک متخصص می‌تواند به سرعت انواع مختلف مدل‌ها را ارزیابی کند تا ابزار مناسب‌تری برای داده‌های خود بیابد.

زیبایی R این است که مجموعه بزرگ و متنوعی از ­سنین بسته مدلینگ را ارائه می‌دهد. با این حال، از آنجایی که این بسته‌ها توسط افراد مختلف در طول زمان ایجاد می‌شوند، حداقل مجموعه‌ای از قراردادها وجود دارد که برای هر مدل مشترک است. به‌عنوان مثال، جدول [B. 1](#bookmark995) نحوی را برای محاسبه احتمالات کلاس برای چندین نوع مختلف مدل طبقه‌بندی نشان می‌دهد. به خاطر سپردن تغییرات نحوی می‌تواند دشوار باشد و این امر کاربران را از ارزیابی انواع مدل‌ها منصرف می‌کند. یکی از روش‌های کاهش این پیچیدگی، ارائه یک سطح یکپارچه ­به توابع برای ساخت و پیش‌بینی مدل است. caret چنین رابطی را برای طیف گسترده‌ای از مدل‌ها (بیش از 140) فراهم می‌کند. این بسته همچنین ­گزینه‌های بسیاری را برای پیش‌پردازش داده‌ها و تکنیک‌های تنظیم پارامتر مبتنی بر نمونه‌برداری مجدد ارائه می‌کند (فصل‌های.  [3](#bookmark131) و [4 )](#bookmark192) .

در این متن، نمونه‌گیری مجدد رویکرد اولیه برای بهینه‌سازی مدل‌های پیش‌بینی با پارامترهای تنظیم است. برای انجام این کار، بسیاری از نسخه‌های جایگزین مجموعه آموزشی برای آموزش مدل و پیش‌بینی مجموعه نگهدارنده استفاده می‌شوند. این ­فرآیند بارها تکرار می‌شود تا تخمین‌های عملکردی به دست آید که به مجموعه داده‌های جدید تعمیم می‌یابد. هر یک از مجموعه داده‌های نمونه برداری مجدد مستقل از سایرین است، بنابراین هیچ الزام رسمی وجود ندارد که مدل‌ها باید به صورت متوالی اجرا شوند. اگر کامپیوتری با چندین پردازنده یا هسته در دسترس باشد، محاسبات ­را می‌توان در سراسر این "کارگران" پخش کرد تا کارایی محاسباتی افزایش یابد. caret از یکی از چارچوب‌های پردازش موازی در R برای انجام این کار استفاده می‌کند. بسته foreach اجازه می‌دهد تا کد R به صورت متوالی یا موازی با استفاده از چندین فناوری مختلف مانند بسته‌های چند هسته‌ای یا Rmpi اجرا شود (نگاه کنید به [اشمیدبرگر و همکاران ( 2009](#bookmark1025) ) برای خلاصه و توصیف گزینه‌های موجود). چندین بسته R وجود دارد که با foreach برای پیاده‌سازی این تکنیک‌ها کار می‌کنند، مانند doMC (برای چند هسته‌ای ) یا doMPI (برای Rmpi ).

برای تنظیم یک مدل پیش‌بینی با استفاده از چندین کارگر، نحو در توابع بسته caret (به‌عنوان مثال، آموزش، rfe یا sbf ) تغییر نمی‌کند. یک تابع جداگانه برای "ثبت" تکنیک پردازش موازی و تعیین تعداد کارگران مورد استفاده استفاده می‌شود. به‌عنوان مثال، برای استفاده از بسته چند هسته‌ای (در ویندوز موجود نیست) با پنج هسته در یک دستگاه، بسته بارگیری می‌شود و سپس ثبت می‌شود:

*> کتابخانه (doMC)*

*registerDoMC (هسته = 5)*

*## تمام مدل‌های بعدی به صورت موازی اجرا می‌شوند*

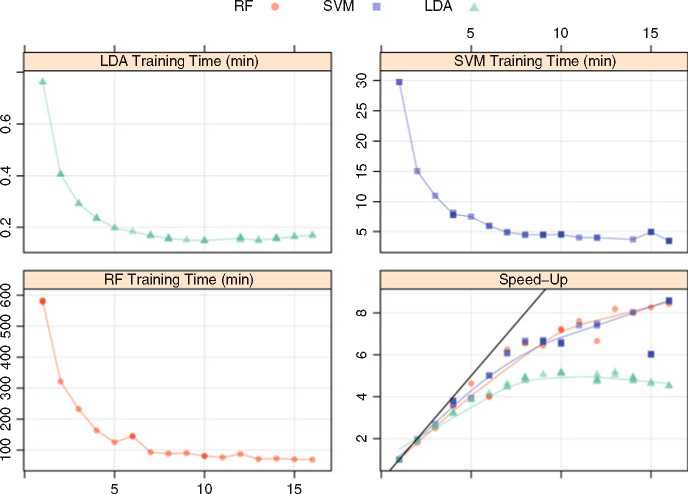
*مدل <- train(y ~. ، داده = آموزش، روش = "rf")*

نحو برای بسته‌های دیگر مرتبط با foreach بسیار مشابه است. توجه داشته باشید که با افزایش تعداد کارگران، حافظه مورد نیاز نیز افزایش می‌یابد. به‌عنوان مثال، استفاده از پنج کارگر در مجموع شش نسخه از داده‌ها را در حافظه نگه می‌دارد. اگر داده‌ها بزرگ باشند یا مدل محاسباتی نیاز داشته باشد، اگر مقدار حافظه مورد نیاز از مقدار فیزیکی موجود بیشتر شود، عملکرد می‌تواند تحت تأثیر قرار گیرد.

آیا این به کاهش زمان مناسب برای مدل‌ها کمک می‌کند؟ داده‌های زمان‌بندی کار (فصل.  [17 )](#bookmark795) چندین بار با تعداد کارگران مختلف برای چندین مدل مدلسازی شد. جنگل تصادفی با 2000 درخت استفاده شد و بیش از 10 مقدار *متر تلاش کوک شد.* محاسبات اهمیت متغیر نیز در طول هر برازش مدل انجام شد. تحلیل تفکیک خطی نیز اجرا شد، همانطور که یک ­ماشین بردار پشتیبان تابع پایه شعاعی حساس به هزینه (که بیش از 15 مقدار هزینه تنظیم شده بود) اجرا شد. همه مدل‌ها با استفاده از پنج تکرار 10 برابر اعتبار متقاطع تنظیم شدند. نتایج در شکل نمایش داده شده اند.  [ب. 1](#bookmark999) . محور ­*y با کل* زمان اجرا (شامل تنظیم مدل و برازش مدل نهایی) در مقابل تعداد کارگران مطابقت دارد. جنگل تصادفی به وضوح طولانی‌ترین زمان را برای آموزش نیاز داشت و مدل‌های LDA از نظر محاسباتی بسیار کارآمد بودند. کل زمان (بر حسب دقیقه) با افزایش تعداد کارگران کاهش یافت اما در حدود هفت کارگر تثبیت شد. داده‌های این نمودار به‌صورت تصادفی‌سازی شده تولید شد تا در ترتیب اجرا هیچ گونه بایاس وجود نداشته باشد. پانل پایین سمت راست سرعت افزایش را نشان می‌دهد *که* زمان متوالی تقسیم بر زمان موازی است. به‌عنوان مثال، افزایش سرعت سه نشان می‌دهد که نسخه موازی سه برابر سریعتر از نسخه متوالی بوده است. در بهترین حالت، موازی‌سازی می‌تواند به افزایش سرعت خطی دست یابد. یعنی برای کارگران *M،* زمان موازی 1 */M* است. برای این مدل‌ها، تا زمانی که از چهار یا پنج کارگر استفاده شود، افزایش سرعت نزدیک به خطی است. پس از این، یک بهبود کوچک در عملکرد وجود دارد. از آنجایی که LDA در حال حاضر از نظر محاسباتی ­کارآمد است، سرعت افزایش سریع‌تر از مدل‌های دیگر کاهش می‌یابد. اگرچه خطی نیست، کاهش زمان اجرا مفید است - برازش مدل تقریباً 10 ساعته به حدود 90 دقیقه کاهش یافت.

توجه داشته باشید که برخی از مدل‌ها، به ویژه آنهایی که از پکیج RWeka استفاده می‌کنند، ممکن است به دلیل ساختار کد زیرین، نتوانند به صورت موازی اجرا شوند.

یکی از ترفندهای اضافی که آموزش برای افزایش کارایی محاسباتی از ­آن بهره می‌برد، استفاده از مدل‌های فرعی است. یک برازش مدل واحد می‌تواند پیش‌بینی‌کننده‌هایی برای



کارگران

شکل B. 1: سه مدل با استفاده از تعداد کارگران مختلف اجرا می‌شوند. محور *y یا* زمان اجرا بر حسب دقیقه یا افزایش سرعت (در *پانل پایین سمت راست* ) پارامترهای تنظیم چندگانه است. برای مثال، در اکثر پیاده‌سازی‌های مدل‌های تقویت‌شده، یک مدل آموزش‌دیده بر روی تکرارهای تقویت‌کننده *B* می‌تواند پیش‌بینی‌کننده‌هایی را برای مدل‌هایی برای تکرارهای کمتر از *B* ایجاد کند. برای داده‌های گرنت، یک مدل gbm مناسب بود که 200 ترکیب متمایز از سه پارامتر تنظیم را ارزیابی می‌کرد (شکل 2 را ببینید).  [14. 10 )](#bookmark716) . در واقعیت، Train فقط برای 40 مدل اشیاء ایجاد کرد و سایر پیش‌بینی‌کننده‌ها را از این اشیاء استخراج کرد.

جزئیات بیشتر در مورد بسته caret را می‌توانید در اینجا بیابید [کوهن](#bookmark1020) [( 2008](#bookmark1020) ) یا چهار کتابچه راهنمای توسعه یافته (به نام "وینیت") در وب سایت بسته [( کوهن 2010](#bookmark1020) ).

B. 11 نرم‌افزار مورد استفاده در این متن

تابع Sweave عالی [( Leisch 2002a , b](#bookmark1020) ) در R باعث شد کد تحلیل داده‌ها در محتوای این متن یکپارچه شود. این تابع کد R را اجرا کرد و کد را با موارد تولید شده توسط کد مانند متن، شکل‌ها و جداول جایگزین کرد. تمامی نرم‌افزارها و داده‌های مورد استفاده در اینجا در زمان نگارش این مقاله در دسترس عموم هستند. بسته‌های R AppliedPredictiveModeling و caret شامل بسیاری از مجموعه داده‌ها می‌شود. برای داده‌هایی که نمی‌توانند گنجانده شوند، بسته AppliedPredictiveModeling شامل کد R برای ایجاد مجدد مجموعه داده‌های استفاده شده در این متن است. فهرست گسترده‌ای از بسته‌ها و توابع در R مربوط به تحقیقات تکرارپذیر را می‌توان در CRAN یافت:

[http://cran. r-project. org/web/views/ReproducibleResearch. html](http://cran.r-project.org/web/views/ReproducibleResearch.html)

نسخه 2. 15. 2 (26-10-2012) R همراه با نسخه‌های بسته زیر استفاده شد: AppliedPredictiveModeling (1. 01)، قوانین ( 1. 0-12 )، C50 (0. 1. 0 ­013)، caret (5. 15-045)، coin (1. 0-21)، CORElearn (0. 9. 40)، corrplot (0. 70)، ctv (0. 7-4)، Cubist (0. 0. 12)، مطلوبیت (1. 05)، DMwR (0. 2. 3)، doBy (4. 5-5) )، doMC (1. 2. 5)، DWD (0. 10)، e1071 (1. 6-1)، زمین (3. 2-3)، الاستیکنت (1. 1)، بیضی (0. 3 ­7)، gbm (1. 6-3. 2)، glmnet (1. 8- 2)، Hmisc (3. 10-1)، ipred (0. 9-1)، kernlab (0. 9 ­15)، klaR (0. 6-7)، lars (1. 1)، latticeExtra (0. 6-24)، شبکه (0. 20-10)، MASS (7. 3 ­22)، mda (0. 4-2)، minerva (1. 2)، mlbench (2. 1-1)، nnet (7. 3-5)، pamr (1. 54)، partykit (0. 1-4)، party (1. 0-3)، pls (2. 3-0)، plyr (1. 7. 1)، pROC (1. 5. 4)، پروکسی (0. 4 ­9)، QSARdata (1. 02)، randomForest (4. 6-7)، RColorBrewer (1. 0-5)، reshape2 (1. 2. 1)، تغییر شکل (0. 8. 4)، rms (3. 6-0)، rpart (4. 0-3)، RWeka (0. 4-12 )، SparseLDA ( 0. 1-6)، انتخاب فرعی ( 0. 12-2 )، svmpath (0. 952) و جدول (0. 12). برخی از این بسته‌ها مستقیماً با مدل‌سازی پیش‌بینی مرتبط نیستند، اما برای جمع‌آوری یا قالب‌بندی محتوا یا برای تجسم استفاده می‌شوند.

پیوست ج

وب سایت‌های جالب

نرم‌افزار

[http://www. r-project. org](http://www.r-project.org)

این وب سایت اصلی R با پیوندهایی به اطلاعیه‌ها، راهنماها، کتاب‌ها، کنفرانس‌ها و اطلاعات دیگر است.

[http://cran. r-project. org](http://cran.r-project.org)

CRAN، یا شبکه جامع آرشیو R، مخزن اصلی برای R و بسته‌های الحاقی متعدد است.

[http://cran. r-project. org/web/views/MachineLearning. html](http://cran.r-project.org/web/views/MachineLearning.html)

نمای کار یادگیری ماشینی فهرستی از بسیاری از ­سنین بسته مدل‌سازی پیش‌بینی در R است.

[http://caret. r-forge. r-project. org](http://caret.r-forge.r-project.org)

بسته caret در اینجا میزبانی می‌شود.

[http://www. rulequest. com](http://www.rulequest.com)

RuleQuest نسخه‌های تجاری و منبع باز Cubist و C5. 0 را منتشر می‌کند.

[http://rattle. togaware. com](http://rattle.togaware.com)

Rattle [( ویلیامز 2011](#bookmark1027) ) یک رابط کاربری گرافیکی برای مدل‌های پیش‌بینی R است.

[http://www. cs. waikato. ac. nz/ml/weka/](http://www.cs.waikato.ac.nz/ml/weka/)

Weka مجموعه‌ای از برنامه‌های جاوا برای داده کاوی است.

[http://orange. biolab. si](http://orange.biolab.si)

Orange یک رابط کاربری گرافیکی متن باز و چند پلتفرمی برای بسیاری از ابزارهای یادگیری ماشین است. رابط یک "خط لوله" است که در آن کاربران اجزایی را برای ایجاد یک گردش کار کنار هم قرار می‌دهند.

[http://www. knime. org](http://www.knime.org)

KNIME (Konstanz Information Miner) یک پلتفرم یکپارچه سازی، پردازش، تحلیل و اکتشاف داده‌های منبع باز کاربرپسند و جامع است. [http://www. spss. com/software/modeler](http://www.spss.com/software/modeler)

IBM SPSS Modeler که قبلاً Clemintine نامیده می‌شد، یک پلت فرم بصری برای ساخت مدل است.

[http://www. sas. com/technologies/analytics/datamining/miner](http://www.sas.com/technologies/analytics/datamining/miner)

یک محصول SAS برای داده کاوی.

سایر برنامه‌ها در فهرست شده اند [http://www. kdnuggets. com/software/suites.](http://www.kdnuggets.com/software/suites.html)  [html \_](http://www.kdnuggets.com/software/suites.html)

مسابقات

[http://www. kaggle. com](http://www.kaggle.com) [http://tunedit. org](http://tunedit.org)

مجموعه داده ها

[http://archive. ics. uci. edu/ml](http://archive.ics.uci.edu/ml)

دانشگاه کالیفرنیا (ایروین) یک مکان شناخته شده برای طبقه‌بندی و مجموعه داده‌های رگرسیون است.

[http://www. kdnuggets. com/datasets](http://www.kdnuggets.com/datasets)

انجمن ماشین‌های محاسباتی (ACM) یک گروه علاقه‌مند ویژه به کشف دانش در داده‌ها (KDD) دارد. گروه KDD مسابقات سالانه یادگیری ماشین را سازماندهی می‌کند.

[http://fueleconomy. gov](http://fueleconomy.gov)

وب سایتی که توسط دفتر کارایی انرژی و انرژی‌های تجدیدپذیر وزارت انرژی ایالات متحده و آژانس حفاظت از محیط زیست ایالات متحده اداره می‌شود و تخمین‌های متفاوتی از مصرف سوخت برای خودروهای سواری و کامیون‌ها را فهرست می‌کند.

[http://www. cheminformatics. org](http://www.cheminformatics.org)

این وب سایت حاوی نمونه‌های زیادی از مجموعه داده‌های شیمی محاسباتی است.

[http://www. ncbi. nlm. nih. gov/geo](http://www.ncbi.nlm.nih.gov/geo)

وب‌سایت NCBI GEO «یک مخزن عمومی است که ریزآرایه‌ها، توالی‌یابی نسل بعدی و دیگر اشکال داده‌های ژنومی عملکردی پرتوان ارائه شده توسط جامعه علمی را بایگانی و توزیع می‌کند».

منابع

عبدی اچ، ویلیامز ال (2010). "تحلیل مولفه اصلی. " *Wiley Interdis ­ciplinary Reviews: Computational Statistics ,* 2 (4)، 433-459.

Agresti A (2002). تحلیل *داده‌های طبقه‌بندی شده.* Wiley-Interscience.

Ahdesmaki M، Strimmer K (2010). "انتخاب ویژگی در مسائل پیش‌بینی Omics با استفاده از امتیازات CAT و کنترل نرخ عدم کشف نادرست. " *The Annals of Applied Statistics* , 4 (1)، 503-519.

آلین آ (2009). "مقایسه الگوریتم‌های PLS زمانی که تعداد اشیاء بسیار بزرگتر از تعداد متغیرها است. " *مقالات آماری* , 50 , 711-720.

آلتمن دی، بلند جی (1994). "تست‌های تشخیصی 3: نمودارهای مشخصه عامل گیرنده. " *مجله متخصصی بریتانیا،* 309 (6948)، 188.

Ambroise C، McLachlan G (2002). "بایاس انتخاب در استخراج ژن بر اساس داده‌های بیان ژن ریزآرایه. " *مجموعه مقالات آکادمی ملی علوم،* 99 (10)، 6562-6566.

Amit Y، Geman D (1997). "کوانتیزه‌سازی و تشخیص شکل با ­درختان ران domized. " *محاسبات عصبی،* 9، 1545-1588.

آرمیتاژ پی، بری جی (1994). روش‌ها*ی آماری در تحقیقات متخصصی.* انتشارات علمی بلک چاه، آکسفورد، ویرایش سوم. ­

آرتیس ام، آیوسو ام، گیلن ام (2002). "تشخیص تقلب در بیمه خودرو با مدل‌های انتخاب گسسته و مطالبات طبقه‌بندی شده اشتباه. " *مجله ریسک و بیمه،* 69 (3)، 325-340.

آستین پی، برونر ال (2004). "تورم نرخ خطای نوع I هنگامی که یک متغیر مخدوش کننده پیوسته در تحلیل‌های رگرسیون لجستیک طبقه‌بندی می‌شود. " *آمار در متخصصی،* 23 (7)، 1159-1178.

آیرس I (2007). *Super Crunchers: چرا فکر کردن بر اساس اعداد راهی جدید برای هوشمند بودن* است. بانتام.

بارکر ام، راینز دبلیو (2003). "حداقل مربعات جزئی برای تبعیض. " *مجله Chemometrics، ­*17 ( 3)، 166-173.

باتیستا جی، پراتی آر، مونارد ام (2004). "مطالعه رفتار چند روش برای متعادل کردن داده‌های آموزشی یادگیری ماشین. " *خبرنامه اکتشافات ACM SIGKDD،* 6 (1)، 20-29.

Bauer E, Kohavi R (1999). "مقایسه تجربی ­الگوریتم‌های طبقه‌بندی رای: بسته بندی، تقویت و انواع. " *یادگیری ماشینی،* 36، 105-142.

بکتون دیکنسون و شرکت (1991). *ProbeTec ET Chlamydia trachomatis و Neisseria gonorrhoeae Amplified DNA Assas (Package Insert).*

بن دور آ، برون ال، فریدمن ن، ناچمن آی، شومر ام، یاخینی ز (2000). "طبقه‌بندی بافت با نمایه‌های بیان ژن. " *مجله زیست شناسی محاسباتی* , 7 (3), 559-583.

بنتلی جی (1975). درخت‌های جستجوی دودویی چند بعدی که برای جستجوی انجمنی استفاده می‌شوند. *ارتباطات ACM،* 18 (9)، 509-517.

Berglund A, Kettaneh N, Uppgard L, Wold S, DR NB, Cameron (2001). "رویکرد GIFI به مدلسازی غیر خطی PLS. " *مجله شیمی سنجی،* 15، 321-336.

Berglund A، Wold S (1997). "INLR، رگرسیون متغیر نهفته غیرخطی ضمنی. " *مجله شیمی سنجی،* 11، 141-156.

برگمیر سی، بنیتز جی ام (2012). "شبکه‌های عصبی در R با استفاده از شبیه ساز شبکه عصبی اشتوتگارت: RSNNS. " *مجله نرم‌افزار آماری* , 46 (7), 1-26.

برگسترا جی، کاساگراند ن، ارهان دی، اک دی، کگل بی (2006). "ویژگی‌های مجموع و AdaBoost برای طبقه‌بندی موسیقی. " *یادگیری ماشینی،* 65، 473-484.

برنتسون پی، ولد اس (1986). "مقایسه بین داده‌های کریستالوگرافی اشعه ایکس و پارامترهای فیزیکوشیمیایی با توجه به اطلاعات آنها ­در مورد فعالیت آنتاگونیست کانال کلسیم 4-فنیل-1،4- دی هیدروپیریدین ها. " *روابط کمّی ساختار-فعالیت،* 5، 45-50.

بانو بی، لین وای (2003). "انتخاب ویژگی مبتنی بر الگوریتم ژنتیک برای ­تشخیص تار در تصاویر SAR. " *محاسبات تصویر و دید،* 21، 591-608.

اسقف سی (1995). *شبکه‌های عصبی برای تشخیص الگو.* انتشارات دانشگاه آکسفورد، آکسفورد.

اسقف سی (2006). تشخیص الگو و *یادگیری ماشینی* اسپرینگر.

بلاند جی، آلتمن دی (1995). "یادداشت‌های آماری: آزمون‌های معناداری چندگانه: روش بونفرونی. " *مجله متخصصی بریتانیا،* 310 (6973)، 170-170.

بلاند جی، آلتمن دی (2000). "نسبت شانس. " *مجله متخصصی بریتانیا،* 320 (7247)، 1468.

بوهاچفسکی، جانسون ام، استین ام (1986). "بازپخت شبیه‌سازی شده عمومی ­برای بهینه‌سازی عملکرد. " *فن سنجی،* 28 (3)، 209-217.

Bone R، Balk R، Cerra F، Dellinger R، Fein A، Knaus W، Schein R، Sibbald W (1992). "تعریف سپسیس و نارسایی اندام و دستورالعمل‌های استفاده از درمان‌های نوآورانه در سپسیس. " *سینه،* 101 (6)، 1644-1655.

Boser B, Guyon I, Vapnik V (1992). "الگوریتم آموزشی برای ­طبقه‌بندی کننده‌های مارجین بهینه. " در «مجموعه مقالات پنجمین کارگاه سالانه ­نظریه یادگیری محاسباتی»، ص 144-152.

Boulesteix A, Strobl C (2009). "انتخاب طبقه‌بندی‌کننده بهینه و بایاس منفی در تخمین نرخ خطا: یک مطالعه تجربی در مورد پیش‌بینی با ابعاد بالا. " *روش تحقیق متخصصی BMC،* 9 (1)، 85.

جعبه جی، کاکس دی (1964). "تحلیلی از تحولات. " *مجله انجمن آمار سلطنتی. سری ب (روش شناختی)،* صص 211-252.

جعبه جی، هانتر دبلیو، هانتر جی (1978). *آمار برای* آزمایشگران ویلی، نیویورک

جعبه جی، تیدول پی (1962). "تغییر متغیرهای مستقل. " فن سنجی، 4 (4)، 531-550.

Breiman L (1996a). «پیش‌بینی‌کنندگان کیف». *یادگیری ماشینی،* 24 (2)، 123-140.

Breiman L (1996b). "ابتکارات بی ثباتی و پایداری در انتخاب مدل. " *The Annals of Statistics* , 24 (6)، 2350-2383.

Breiman L (1996c). "نکته فنی: برخی از ویژگی‌های ضوابط تقسیم. " *یادگیری ماشینی،* 24 (1)، 41-47.

بریمن ال (1998). "طبقه بندی‌های قوس الکتریکی. " *The Annals of Statistics* , 26 , 123-140.

بریمن ال (2000). "تصادفی‌سازی خروجی‌ها برای افزایش دقت پیش‌بینی. " *ماخ فرا گرفتن. ،* 40، 229-242. ISSN 0885-6125.

بریمن ال (2001). "جنگل‌های تصادفی. " *یادگیری ماشینی،* 45، 5-32.

بریمن ال، فریدمن جی، اولشن آر، استون سی (1984). *طبقه‌بندی و ­رگرسیون درختان.* چپمن و هال، نیویورک.

بریدل جی (1990). "تفسیر احتمالی ­خروجی‌های شبکه طبقه‌بندی پیشخور، با روابط با تشخیص الگوی آماری. " در «رایانش عصبی: الگوریتم‌ها، معماری‌ها و کاربردها»، صفحات 227-236. Springer-Verlag.

برلینگر دی (2004). "برخی از تحلیل داده‌ها با استفاده از اطلاعات متقابل. " *مجله احتمالات و آمار برزیل،* 18 (6)، 163-183.

Brodnjak-Vonina D، Kodba Z، Novi M (2005). تحلیل داده‌های چند متغیره در طبقه‌بندی روغن‌های گیاهی با محتوای اسیدهای چرب مشخص می‌شود. *شیمی سنجی و سامانه‌های آزمایشگاهی هوشمند،* 75 (1)، 31-43.

براون سی، دیویس اچ (2006). منحنی‌های مشخصه‌های عملیاتی گیرنده و اقدامات تصمیم‌گیری مرتبط: یک آموزش. شیمی سنجی و *سامانه‌های سخنوری آزمایشگاهی هوشمند،* 80 (1)، 24-38.

Bu G (2009). آپولیپوپروتئین E و گیرنده‌های آن در بیماری آلزایمر ­: مسیرها، پاتوژنز و درمان. *Nature Reviews Neuroscience،* 10 (5)، 333-344.

Buckheit J, Donoho DL (1995). "WaveLab و تحقیقات تکرارشونده. " در A Antoniadis، G Oppenheim (ویرایش)، "Wavelets in Statistics"، صفحات 55-82. Springer-Verlag، نیویورک.

بورز جی، ون دن پول دی (2009). مدیریت عدم تعادل کلاس در پیش‌بینی ریزش مشتری. *سامانه‌های خبره با برنامه‌های کاربردی،* 36 (3)، 4626-4636.

Cancedda N، Gaussier E، Goutte C، Renders J (2003). "توالی کلمات Ker ­nels. " *مجله تحقیقات یادگیری ماشین،* 3، 1059-1082.

Caputo B، Sim K، Furesjo F، Smola A (2002). "تشخیص شیء مبتنی بر ظاهر با استفاده از SVM: از کدام هسته باید استفاده کنم؟" در "مجموعه مقالات کارگاه آموزشی NIPS در مورد روش‌های آماری برای آزمایش‌های محاسباتی در پردازش بصری و بینایی کامپیوتری"،.

Carolin C، Boulesteix A، Augustin T (2007). "انتخاب تقسیم بی طرفانه برای درختان طبقه‌بندی بر اساس شاخص جینی. " *آمار محاسباتی و تحلیل داده‌ها،* 52 (1)، 483-501.

Castaldi P، Dahabreh I, Ioannidis J (2011). "ارزیابی تجربی شیوه‌های اعتبارسنجی برای طبقه‌بندی کننده‌های مولکولی. " *توضیحات در بیوانفورماتیک،* 12 (3)، 189-202.

چمبرز جی (2008). *نرم‌افزار برای تحلیل داده ها: برنامه نویسی* با *R.* اسپرینگر.

چان کی، لوه دبلیو (2004). "لوتوس: الگوریتمی برای ساخت درختان رگرسیون لجستیک دقیق و قابل درک. " *مجله آمار محاسباتی و گرافیکی* , 13 (4), 826-852.

چانگ سی سی، لین سی جی (2011). "LIBSVM: کتابخانه‌ای برای پشتیبانی از بردار ­چینی ها. " *ACM Transactions on Intelligent Systems and Technology،* 2، 27: 1-27:27.

Chawla N، Bowyer K، Hall L، Kegelmeyer W (2002). "SMOTE: روش نمونه برداری بیش از حد اقلیت مصنوعی. " *Journal of Artificial Intelligence Re ­search* , 16 (1), 321-357.

چون اچ، کلس اس (2010). "رگرسیون حداقل مربعات جزئی پراکنده برای ­کاهش ابعاد چندگانه Si و انتخاب متغیر. " *مجله انجمن آماری سلطنتی: سری B (روش شناسی آماری)،* 72 (1)، 3-25.

چانگ دی، کلس اس (2010). "طبقه‌بندی حداقل مربعات جزئی پراکنده برای داده‌های با ابعاد بالا. " *کاربردهای آماری در ژنتیک و ­زیست شناسی مولکولی،* 9 (1)، 17.

کلارک آر (1997). "OptiSim: روش انتخاب عدم تشابه گسترده برای یافتن زیرمجموعه‌های نماینده متنوع. " *مجله اطلاعات شیمی و علوم کامپیوتر* , 37 (6), 1181-1188.

کلارک تی (2004). «آیا مقایسه‌های پیش‌بینی خارج از نمونه می‌تواند به جلوگیری از برازش بیش از حد کمک کند؟» *مجله پیش‌بینی* , 23 (2), 115-139.

کلمنسن ال، هیستی تی، ویتن دی، ارسبول بی (2011). "تحلیل تفکیک پراکنده. " فن سنجی، 53 (4)، 406-413.

کلیولند دبلیو (1979). "رگرسیون محلی وزن دار قوی و نمودارهای پراکندگی هموار. " *مجله انجمن آماری آمریکا،* 74 (368)، 829-836.

کلیولند دبلیو، دولین اس (1988). "رگرسیون وزنی محلی: رویکردی به تحلیل رگرسیون با برازش محلی. " *مجله انجمن آماری آمریکا،* صفحات 596-610.

کوهن جی، هیلاریو ام، پلگرینی سی، گیس بوهلر A (2005). مدلسازی SVM از طریق یک استراتژی ژنتیکی ترکیبی. یک برنامه مراقبت‌های بهداشتی. ” در R Engelbrecht، AGC Lovis (eds. )، "Connecting Medical Informatics and Bio-Informatics"، صفحات 193-198. IOS Press.

کوهن جی (1960). "ضریب توافق برای داده‌های اسمی. " *سنجش تربیتی و روانشناختی،* 20، 37-46.

Cohn D، Atlas L، Ladner R (1994). "بهبود تعمیم با یادگیری فعال. " *یادگیری ماشینی،* 15 (2)، 201-221.

کرنل جی (2002). *آزمایش‌ها با مخلوط‌ها: طرح‌ها، مدل‌ها و ­تحلیل مقعدی داده‌های مخلوط.* وایلی، نیویورک، نیویورک

Cortes C، Vapnik V (1995). "شبکه‌های بردار پشتیبانی. " *یادگیری ماشینی،* 20 (3)، 273-297.

Costa N، Lourenco J، Pereira Z (2011). "رویکرد عملکرد مطلوب: بررسی و ارزیابی عملکرد در شرایط نامطلوب. " *شیمی‌سنجی ­و سامانه‌‌های آزمایشگاهی هوشمند،* 107 (2)، 234-244.

جلد TM، توماس JA (2006). *عناصر نظریه اطلاعات.* وایلی- Interscience.

کریگ شاپیرو آر، کوهن ام، شیونگ سی، پیکرینگ ای، لیو جی، میسکو تی پی، پرین آر، بیلز کی، سوآرس اچ، فاگان آ، هولتزمن دی (2011). پانل مونواسی Im Multiplexed بیومارکرهای جدید CSF را برای تشخیص و پیش آگهی بیماری آلزایمر شناسایی می‌کند. ­*PLoS ONE،* 6 (4)، e18850.

Cruz-Monteagudo M، Borges F، Cordeiro MND (2011). مدیریت مشترک قدرت و سمیت پپتیدومیمتیک‌های ضد میکروبی با قوانین ساده از نظریه مطلوبیت و شیمی انفورماتیک. *Journal of Chemical In ­formation and Modeling* , 51 (12), 3060-3077.

دیویسون ام (1983). *مقیاس‌بندی چند* بعدی جان وایلی و پسران، شرکت

دیال بی، مک گرگور جی (1997). "الگوریتم‌های PLS بهبود یافته. " *مجله شیمی سنجی،* 11، 73-85.

د یونگ اس (1993). SIMPLS: رویکردی جایگزین برای رگرسیون جزئی حداقل مربعات. *شیمی سنجی و سامانه‌های آزمایشگاهی هوشمند،* 18، 251-263.

دی جونگ اس، تر براک سی (1994). "ارتباطات کوتاه: نظرات الگوریتم هسته PLS. " *مجله شیمی سنجی،* 8، 169-174.

د لئون ام، کلانک دبلیو (2006). نشانگرهای زیستی برای تشخیص زودهنگام بیماری آلزایمر. *The Lancet Neurology،* 5 (3)، 198-199.

دفرنز ام، کمسلی‌ای (1997). "استفاده و استفاده نادرست از شیمی سنجی برای درمان مسائل طبقه بندی. " *TrAC Trends in Analytical Chemistry* , 16 (4)، 216-221.

DeLong E، DeLong D، Clarke-Pearson D (1988). "مقایسه مناطق تحت ­دو یا بیشتر منحنی مشخصه عملکرد گیرنده همبسته: یک رویکرد ناپارامعیار. " *بیومعیار،* 44 (3)، 837-45.

Derksen S، Keselman H (1992). « الگوریتم‌های انتخاب زیرمجموعه خودکار به عقب، جلو و گام به گام : فرکانس به دست آوردن متغیرهای معتبر و نویز». ­*مجله انگلیسی ریاضی و آماری روانشناسی،* 45 (2)، 265-282.

دررینگر جی، سوئیچ آر (1980). "بهینه‌سازی همزمان چندین متغیر پاسخ. " *مجله فناوری کیفیت* , 12 (4), 214-219.

دیتریش تی (2000). "مقایسه تجربی سه روش برای ساخت مجموعه‌های درختان تصمیم‌گیری: کیسه‌بندی، تقویت و ­دانومیزاسیون". *یادگیری ماشینی،* 40، 139-158.

دیلون دبلیو، گلدشتاین ام (1984). *تحلیل چند متغیره: روش‌ها و ­کاربردها.* ویلی، نیویورک

دابسون A (2002). *مقدمه‌ای بر مدل‌های خطی تعمیم یافته.* چپمن و هال/CRC.

Drucker H، Burges C، Kaufman L، Smola A، Vapnik V (1997). "پشتیبانی از ماشین‌های رگرسیون برداری. " *پیشرفت‌ها در سامانه‌‌های پردازش اطلاعات عصبی،* صفحات 155-161.

Drummond C، Holte R (2000). "نمایش صریح هزینه مورد انتظار: جایگزینی برای نمایندگی ROC. " در "مجموعه مقالات ششمین کنفرانس بین المللی ACM SIGKDD در مورد کشف دانش و داده ­کاوی"، صفحات 198-207.

Duan K، Keerthi S (2005). بهترین روش SVM چند کلاسه کدام است؟ یک مطالعه تجربی. » *سامانه‌های طبقه‌بندی چندگانه،* صص 278-285.

Dudoit S, Fridlyand J, Speed T (2002). "مقایسه روش‌های تشخیص برای طبقه‌بندی تومورها با استفاده از داده‌های بیان ژن. " *مجله انجمن آماری آمریکا،* 97 (457)، 77-87.

داهیگ سی (2012). "شرکت‌ها چگونه اسرار شما را یاد می‌گیرند. " *نیویورک تایمز.* آدرس  [http://www. nytimes. com/2012/02/19/magazine/](http://www.nytimes.com/2012/02/19/magazine/shopping-habits.html)

[shopping-habits. html](http://www.nytimes.com/2012/02/19/magazine/shopping-habits.html) .

دان دبلیو، ولد اس (1990). "تکنیک‌های تشخیص الگو در طراحی دارو. " در سی هانش، پی سامز، جی تیلور (ویرایش‌ها)، "مطالعه جامع شیمی دارویی ­"، صفحات 691-714. چاپ پرگامون، آکسفورد.

دوایر دی (2005). "نمونه‌هایی از بیش برازش در هنگام ساخت ­مدل‌های پیش فرض پیش فرض شرکت خصوصی. " *گزارش فنی،* مودی KMV.

افرون بی (1983). برآورد میزان خطای یک قانون پیش‌بینی: بهبود ­اعتبارسنجی متقاطع. *مجله انجمن آماری آمریکا، صفحات ­316-331.*

افرون بی، هستی تی، جانستون آی، تیبشیرانی آر (2004). "سیون رگرسیون حداقل زاویه. " *The Annals of Statistics* , 32 (2), 407-499.

افرون ب، تبشیرانی آر (1986). "روش‌های بوت استرپ برای خطاهای استاندارد، ­فواصل اطمینان و سایر معیارهای دقت آماری. " *علوم آماری،* ص 54-75.

افرون بی، تبشیرانی آر (1997). "بهبودهای اعتبارسنجی متقابل: روش 632+ Bootstrap. " *مجله انجمن آماری آمریکا،* 92 (438)، 548-560.

Eilers P، Boer J، Van Ommen G، Van Houwelingen H (2001). "طبقه‌بندی داده‌های ریزآرایه با رگرسیون لجستیک مجازات". در مجموعه مقالات SPIE، جلد 4266، ص. 187.

Eugster M، Hothorn T، Leisch F (2008). "تحلیل اکتشافی و استنباطی آزمایش‌های معیار. " *Ludwigs-Maximilians-Universitat Munchen، گروه آمار، فناوری. نماینده* , 30.

Everitt B، Landau S، Leese M، Stahl D (2011). تحلیل *خوشه‌ای* وایلی.

اوالد بی (2006). "انتخاب پست فوری نقاط برش، بایاس را در ­تحقیقات تشخیصی معرفی کرد. " *مجله اپیدمیولوژی بالینی،* 59 (8)، 798-801.

Fanning K، Cogger K (1998). "تشخیص کلاهبرداری مدیریت شبکه با استفاده از داده‌های مالی منتشر شده. " *مجله بین المللی ­سامانه‌های هوشمند در حسابداری، مالی و مدیریت،* 7 (1)، 21-41.

Faraway J (2005). *مدل‌های خطی با* R. چپمن و هال/CRC، بوکا راتون. فاست تی (2006). "مقدمه‌ای بر تحلیل ROC. " *نامه‌های تشخیص الگو،* *27 (8)،* 861-874.

فیشر آر (1936). "استفاده از اندازه گیری‌های چندگانه در مسائل طبقه‌بندی. " *Annals of Eugenics* , *7 (*2), 179-188.

Forina M، Casale M، Oliveri P، Lanteri S (2009). "برادران کایمن: خانواده‌ای ­از تکنیک‌های قدرتمند طبقه‌بندی و مدل‌سازی کلاس. " *کمومت ریکس ­و سامانه‌‌های آزمایشگاهی هوشمند،* *96 (*2)، 239-245.

فرانک ای، وانگ ای، اینگلیس اس، هولمز جی (1998). "استفاده از درختان مدل برای طبقه‌بندی. " *یادگیری ماشینی.*

فرانک ای، ویتن من (1998). "تولید مجموعه قوانین دقیق بدون بهینه‌سازی جهانی. " *مجموعه مقالات پانزدهمین کنفرانس بین المللی یادگیری ماشینی،* صفحات 144-151.

بنیاد نرم‌افزار آزاد (ژوئن 2007). مجوز عمومی عمومی *گنو*

فروند Y (1995). "تقویت الگوریتم ضعیف یادگیری توسط اکثریت. " اطلاعات *و محاسبات،* *121، 256-285.*

فروند Y، Schapire R (1996). "آزمایش با یک الگوریتم تقویت جدید. " *یادگیری ماشینی: مجموعه مقالات سیزدهمین کنفرانس بین المللی،* صفحات 148-156.

فریدمن جی (1989). "تحلیل متمایز منظم. " *مجله انجمن آماری آمریکا،* *84 (*405)، 165-175.

فریدمن جی (1991). "Splines رگرسیون تطبیقی چند متغیره. " *The Annals of Statistics* , *19 (*1), 1-141.

فریدمن جی (2001). "تقریبی تابع حریص: ماشین تقویت کننده گرادیان. " *Annals of Statistics* , *29 (*5), 1189-1232.

فریدمن جی (2002). "تقویت گرادیان تصادفی. " *آمار محاسباتی و تحلیل داده‌ها،* *38 (*4)، 367-378.

فریدمن جی، هستی تی، تبشیرانی آر (2000). "رگرسیون لجستیک افزودنی: دیدگاهی آماری از تقویت. " *Annals of Statistics* , *38* , 337-374.

فریدمن جی، هستی تی، تبشیرانی آر (2010). "مسیرهای منظم‌سازی برای ­مدل‌های خطی عمومی از طریق نزول مختصات. " *مجله نرم‌افزارهای آماری ­,* 33 (1), 1-22.

گیسر اس (1993). *استنتاج پیش‌بینی شده: مقدمه.* چپمن و هال

گلادی پ، کوالسکی ب (1986). "رگرسیون جزئی حداقل مربعات: یک آموزش. " *Analytica Chimica Acta،* *185،* 1-17.

گلادی پی، مانلی ام، لستاندر تی (2003). "نقشه پراکندگی در تحلیل داده‌های چند متغیره. " *Journal of Chemometrics* , *17 (*8-9), 503-511.

جنتلمن آر (2008). برنامه نویسی *R برای بیوانفورماتیک* مطبوعات CRC.

جنتلمن آر، کری وی، بیتس دی، بولستاد بی، دتلینگ ام، دودویت اس، الیس بی، گوتیه ال، جی ای، جنتری جی، هورنیک کی، هاتورن تی، هوبر ام، ایاکوس اس، آیریزاری آر، لیش اف، لی سی، ماچلر ام، روسینی ای، ساویتزکی جی، اسمیت سی، اسمیت جی، تیرنی ال، یانگ جی، ژانگ جی (2004). Bioconductor: توسعه نرم‌افزار باز برای زیست شناسی محاسباتی و بیوانفورماتیک. *ژنوم بیولوژی،* *5 (*10)، R80.

جولیانو کی، دبیاسیو آر، دانلی آر، گوف آ، ولوسکی جی، زوک جی، پاولاکیس جی، تیلور دی (1997). "غربالگری با محتوای بالا: رویکردی جدید برای کاهش تنگناهای کلیدی در فرآیند کشف دارو. " *مجله غربالگری بیومولکولی،* 2 (4)، 249-259.

گلدبرگ دی (1989). *الگوریتم‌های ژنتیک در جستجو، بهینه‌سازی و ­یادگیری کامپیوتری.* ادیسون-وسلی، بوستون

گلوب جی، هیث ام، وهبا جی (1979). "تعمیم اعتبار متقاطع به‌عنوان روشی برای انتخاب پارامتر ریج خوب. " فن سنجی، 21 (2)، 215-223.

خوب P (2000). *آزمون‌های جایگشت: راهنمای عملی برای روش‌های نمونه‌گیری مجدد برای آزمایش فرضیه‌ها.* اسپرینگر.

Gowen A، Downey G، Esquerre C، O'Donnell C (2010). «جلوگیری از ­برازش در مدل‌های کالیبراسیون PLS داده‌های طیف‌سنجی فروسرخ نزدیک (NIR) با استفاده از ضرایب رگرسیون». *مجله شیمی سنجی،* 25، 375-381.

Graybill F (1976). *تئوری و کاربرد مدل خطی.* Wadsworth & Brooks، Pacific Grove، CA.

Guo Y, Hastie T, Tibshirani R (2007). "تحلیل متمایز خطی منظم و کاربرد آن در ریزآرایه ها. " *آمار زیستی،* 8 (1)، 86-100.

گوپتا اس، هانسنس دی، هاردی بی، کان دبلیو، کومار وی، لین ان، راویشانکر ان، سریرم اس (2006). "مدل‌سازی ارزش طول عمر مشتری. " *مجله پژوهش خدمات* , 9 (2), 139-155.

Guyon I، Elisseeff A (2003). "مقدمه‌ای بر انتخاب متغیر و ویژگی. " *مجله تحقیقات یادگیری ماشین،* 3، 1157-1182.

Guyon I، Weston J، Barnhill S، Vapnik V (2002). "انتخاب ژن برای طبقه‌بندی سرطان با استفاده از ماشین‌های ناقل پشتیبان. " *یادگیری ماشینی،* 46 (1)، 389-422.

هال ام، اسمیت ال (1997). "انتخاب زیر مجموعه ویژگی: رویکرد فیلتر مبتنی بر همبستگی. " ­*کنفرانس بین المللی پردازش اطلاعات عصبی و سامانه‌های اطلاعاتی هوشمند،* صفحات 855-858.

هال پی، هیندمن آر، فن وای (2004). "فاصله‌های اطمینان ناپارامتری برای منحنی‌های مشخصه عملیاتی گیرنده. " *Biometrika،* 91، 743-750.

همپل اچ، فرانک آر، برویچ کی، تیپل اس، کاتز آر، هاردی جی، هرهولز کی، بوکده آ، جسن اف، هوسلر وای (2010). نشانگرهای زیستی برای بیماری آلزایمر: ­دیدگاه‌های آکادمیک، صنعت و نظارتی. *Nature Reviews Drug Dis ­covery* , 9 (7)، 560-574.

دست D، تا R (2001). "یک تعمیم ساده از ناحیه زیر منحنی ROC برای مسائل طبقه‌بندی چند طبقه. " *یادگیری ماشینی،* 45 (2)، 171-186.

هانلی جی، مک نیل بی (1982). "معنا و استفاده از ناحیه تحت یک مشخصه منحنی گیرنده (ROC)". *رادیولوژی،* 143 (1)، 29-36.

Hardle W، Werwatz A، Muller M، Sperlich S، Hardle W، Werwatz A، Muller M، Sperlich S (2004). "تخمین چگالی ناپارامعیار. " در «مدل‌های غیرپارامتری ­و نیمه‌پارامتری»، ص 39-83. اسپرینگر برلین هایدلبرگ.

هارل اف (2001). *استراتژی‌های مدل‌سازی رگرسیون: با کاربرد در مدل‌های خطی، رگرسیون لجستیک و تحلیل بقا.* اسپرینگر، نیویورک

هیستی تی، پرگیبون دی (1990). "کوچک شدن درختان. " *گزارش فنی،* گزارش فنی آزمایشگاه‌های AT&T Bell.

هستی تی، تبشیرانی ر (1990). *مدل‌های افزایشی تعمیم* یافته چپمن و هال/CRC.

هستی تی، تبشیرانی ر (1996). "تحلیل تمایز توسط مخلوط‌های گاوسی. " *مجله انجمن آمار سلطنتی. سری ب،* صص 155-176.

هستی تی، تبشیرانی ر، بوجا ا (1994). "تحلیل تفکیک انعطاف‌پذیر با امتیازدهی بهینه. " *مجله انجمن آماری آمریکا،* 89 (428)، 1255-1270.

هستی تی، تبشیرانی آر، فریدمن جی (2008). *عناصر یادگیری آماری ­: داده کاوی، استنتاج و پیش‌بینی.* اسپرینگر، نسخه 2.

هاوکینز دی (2004). "مسأله بیش برازش. " *مجله اطلاعات شیمی ­و علوم کامپیوتر* , 44 (1), 1-12.

هاوکینز دی، بساک اس، میلز دی (2003). "ارزیابی برازش مدل با اعتبارسنجی متقاطع. " ­*مجله اطلاعات شیمی و علوم کامپیوتر* , 43 (2), 579-586.

هندرسون H، Velleman P (1981). "ساخت مدل‌های رگرسیون چندگانه به ­صورت تعاملی. " *بیومعیار،* صص 391-411.

Hesterberg T، Choi N، Meier L، Fraley C (2008). "کمترین زاویه و رگرسیون جریمه شده ­*L* 1 : یک بررسی. " بررسی‌های *آماری،* 2، 61-93.

هیمن آر، اسلپ آ (2001). "خطرات پیش‌بینی طلاق بدون تایید متقابل. " *مجله ازدواج و خانواده،* 63 (2)، 473.

Hill A، LaPan P، Li Y، Haney S (2007). "تأثیر تقسیم‌بندی تصویر بر کیفیت داده‌های غربالگری محتوای بالا برای سلول‌های SK-BR-3. " *BMC Bioinfor ­matics،* 8 (1)، 340.

هو تی (1998). "روش زیرفضای تصادفی برای ساخت جنگل‌های تصمیم گیری. " *معاملات IEEE در تحلیل الگو و هوش ماشینی،* 13، 340-354.

Hoerl A (1970). "رگرسیون برآمدگی: تخمین بایاس برای مسائل غیر متعامد. " *فن سنجی،* 12 (1)، 55-67.

هالند جی (1975). *سازگاری در سامانه‌های طبیعی و مصنوعی.* انتشارات دانشگاه میشیگان، آن آربور، MI.

هالند جی (1992). *سازگاری در سامانه‌های طبیعی و مصنوعی.* MIT Press، کمبریج، MA.

هولمز جی، هال ام، فرانک‌ای (1993). "تولید مجموعه قوانین از درختان مدل. " در "کنفرانس مشترک استرالیا در مورد هوش مصنوعی"،.

Hothorn T، Hornik K، Zeileis A (2006). "پارتیشن‌بندی بازگشتی بی طرفانه: یک چارچوب استنتاج مشروط. " مجله *آمار محاسباتی و نموداری ­*, 15 (3), 651-674.

Hothorn T، Leisch F، Zeileis A، Hornik K (2005). «طراحی و تحلیل ­آزمایش‌های معیار». *مجله آمار محاسباتی و گرافیکی* , 14 (3), 675-699.

Hsieh W, Tang B (1998). "کاربرد مدل‌های شبکه عصبی برای پیش‌بینی و تحلیل داده‌ها در هواشناسی و اقیانوس شناسی. " *Bulletin of the Ameri ­can Meteorological Society،* 79 (9)، 1855-1870.

Hsu C، Lin C (2002). "مقایسه روش‌ها برای ­ماشین‌های بردار پورت پشتیبانی چند کلاسه. " *معاملات IEEE در شبکه‌های عصبی،* 13 (2)، 415-425.

هوانگ سی، چانگ بی، چنگ دی، چانگ سی (2012). "انتخاب ویژگی و ­بهینه‌سازی پارامترهای یک مدل انتخاب سهام مبتنی بر فازی با استفاده از الگوریتم‌های ژنتیک. " *مجله بین المللی سامانه‌های فازی،* 14 (1)، 65-75.

Huuskonen J (2000). "تخمین حلالیت آبی برای مجموعه متنوعی از ترکیبات آلی بر اساس توپولوژی مولکولی. " *مجله اطلاعات شیمی و علوم کامپیوتر* , 40 (3), 773-777.

ایهاکا آر، جنتلمن آر (1996). R : زبانی برای تحلیل داده‌ها و ­نمودارهای گرافیکی. *مجله آمار محاسباتی و گرافیکی* , 5 (3), 299-314.

Jeatrakul P، Wong K، Fung C (2010). "طبقه‌بندی داده‌های نامتعادل با ترکیب شبکه عصبی مکمل و الگوریتم SMOTE. " *پردازش اطلاعات عصبی مدل‌ها و کاربردها،* صص 152-159.

جرز جی، مولینا آی، گارسیا-لانسینا پی، آلبا آر، ریبلس ان، مارتین ام، فرانکو ال (2010). "تقریبا داده‌های گمشده با استفاده از روش‌های آماری و یادگیری ماشینی در یک مسأله واقعی سرطان سینه. " *هوش مصنوعی در متخصصی،* 50، 105-115.

جان جی، کوهاوی آر، فلگر ک (1994). "ویژگی‌های نامربوط و مسأله انتخاب زیر مجموعه. " *مجموعه مقالات یازدهمین کنفرانس بین المللی یادگیری ماشینی،* 129، 121-129.

جانسون کی، راینز دبلیو (2007). "روش‌های طبقه‌بندی مدرن برای پوشش دیس دارو. " در A Dmitrienko، C Chuang-Stein، R D'Agostino (ویرایشگران)، "Pharma ­ceutical Statistics Using SAS : A Practical Guide"، صفحات 7-43. کری، NC: موسسه SAS Inc.

جانسون آر، ویچرن دی (2001). *تحلیل آماری چند متغیره* کاربردی سالن پرنتیس

Jolliffe I, Trendafilov N, Uddin M (2003). "تکنیک اصلاح شده اجزای اصلی بر اساس ریج. " *مجله آمار محاسباتی و گرافیکی* , 12 (3), 531-547.

Kansy M، Senner F، Gubernator K (1998). غربالگری با توان بالای فیزیکوشیمیایی: سنجش نفوذ غشایی مصنوعی موازی در توصیف ­فرآیندهای جذب غیرفعال. *مجله شیمی دارویی،* 41، 1007-1010.

Karatzoglou A, Smola A, Hornik K, Zeileis A (2004). kernlab -An Package S4 for Kernel Methods در R. *مجله نرم‌افزارهای آماری* , 11 (9), 1-20.

کرنز ام، والیانت ال (1989). "محدودیت‌های رمزنگاری در یادگیری فرمول‌های بولی و خودکارهای محدود. " در مجموعه مقالات بیست و یکم ­سمپوزیوم سالانه ACM در نظریه محاسبات،.

کیم جی، بساک جی، هولتزمن دی (2009). نقش آپولیپوپروتئین E در بیماری آلزایمر. *نورون،* 63 (3)، 287-303.

کیم جی اچ (2009). «تخمین میزان خطای طبقه‌بندی: ­اعتبارسنجی متقاطع مکرر، توقف مکرر و بوت استرپ». *آمار محاسباتی و تحلیل داده‌ها،* 53 (11)، 3735-3745.

کیمبال A (1957). "اشتباهات از نوع سوم در مشاوره آماری. " *مجله انجمن آماری آمریکا،* 52، *133-142.*

کیرا کی، رندل ال (1992). "مسأله انتخاب ویژگی: روش‌های سنتی ­و یک الگوریتم جدید. " *مجموعه مقالات همایش ملی ­هوش مصنوعی،* ص 129-129.

Kline DM، Berardi VL (2005). "بازبینی توابع خطای مربعی و متقاطع آنتروپی برای آموزش طبقه‌بندی کننده‌های شبکه عصبی. " *محاسبات عصبی و برنامه‌های کاربردی،* 14 (4)، 310-318.

کوهوی ر (1374). "مطالعه اعتبارسنجی متقاطع و بوت استرپ برای تخمین دقت و انتخاب مدل. " *کنفرانس بین المللی مشترک ­هوش مصنوعی،* 14، 1137-1145.

کوهوی ر (1375). افزایش دقت طبقه‌بندی‌کننده‌های ساده‌لوحانه بیز: ترکیبی درخت تصمیم. در "مجموعه مقالات دومین کنفرانس بین المللی ­کشف دانش و داده کاوی" جلد 7.

کوهونن تی (1995). نقشه‌های *خودسازماندهی* اسپرینگر.

Kononenko I (1994). "برآورد ویژگی ها: تحلیل و بسط آزادی امداد. ­" در F Bergadano, L De Raedt (eds. ), "Machine Learning: ECML-94" جلد 784, pp. 171-182. اسپرینگر برلین / هایدلبرگ.

کوهن ام (2008). "ساخت مدل‌های پیش‌بینی در R با استفاده از بسته کارت. " *مجله نرم‌افزار آماری* , 28 (5).

کوهن ام (2010). " صفحه اصلی بسته caret. " URL [http://caret. r-forge.](http://caret.r-forge.r-project.org/)  [r-project. org/](http://caret.r-forge.r-project.org/) .

کویپر اس (2008). "مقدمه‌ای بر رگرسیون چندگانه: ارزش ماشین شما چقدر است؟" *مجله آموزش آمار* , 16 (3).

Kvaiseth T (1985). "نکته احتیاطی در مورد *R* 2. " *آماردان آمریکایی،* 39 (4)، 279-285.

Lachiche N، Flach P (2003). "بهبود دقت و هزینه طبقه‌بندی کننده‌های احتمالی دو کلاسه و چند کلاسه با استفاده از منحنی‌های ROC. " در « ­مجموعه مقالات بیستمین کنفرانس بین المللی یادگیری ماشین»، جلد ­20، صفحات 416-424.

لاروس دی (2006). روش‌ها *و مدل‌های داده کاوی.* وایلی.

Lavine B, Davidson C, Moores A (2002). "الگوریتم‌های ژنتیک ابتکاری برای شیمی انفورماتیک. " *شیمی سنجی و سامانه‌های آزمایشگاهی هوشمند،* 60 (1)، 161-171.

لیچ A، ژیلت V (2003). *مقدمه‌ای بر شیمی انفورماتیک.* اسپرینگر.

لیش اف (2002a). Sweave : تولید پویا گزارش‌های آماری با استفاده از تحلیل داده‌های باسواد. در W Hardie، B Ronz (ویرایش‌ها)، "Compstat 2002 - Proceedings in Computational Statistics"، صفحات 575-580. Physica Verlag، هایدلبرگ.

لیش اف (2002b). Sweave , Part I: Mixing R and L A TEX. *R News* , 2 (3), 28-31.

لوی اس (2010). "انقلاب هوش مصنوعی در راه است. " *سیمی.*

لی جی، فاین جی پی (2008). "تحلیل ROC با کلاس‌های چندگانه و تست‌های متعدد: روش شناسی و کاربرد آن در مطالعات ریزآرایه. " *Biostatis ­tics،* 9 (3)، 566-576.

لیندگرن اف، گلادی پی، وولد اس (1993). "الگوریتم هسته برای PLS. " مجله *Chemometrics، 7، 45-59.*

لینگ سی، لی سی (1998). "داده کاوی برای بازاریابی مستقیم: مسائل و راه حل ­ها. " در «مجموعه مقالات چهارمین کنفرانس بین المللی ­کشف لبه دانش و داده کاوی»، صفحات 73-79.

Lipinski C، Lombardo F، Dominy B، Feeney P (1997). "رویکردهای تجربی و محاسباتی برای تخمین حلالیت و نفوذپذیری در تنظیمات کشف و توسعه دارو. " *نماهای Advanced Drug Delivery Re ­*, 23 , 3-25.

لیو بی (2007). *داده کاوی وب.* اسپرینگر برلین / هایدلبرگ.

لیو وای، راینز دبلیو (2007). "PLS و کاهش ابعاد برای طبقه بندی. " *آمار محاسباتی،* ص 189-208.

Lo V (2002). "مدل افزایش واقعی: یک رویکرد داده کاوی جدید برای مدل‌سازی پاسخ در بازاریابی پایگاه داده. " *ACM SIGKDD Explorations Newsletter* , 4 (2), 78-86.

لودی اچ، ساندرز سی، شاوه-تیلور جی، کریستیانینی ن، واتکینز سی (2002). "طبقه‌بندی متن با استفاده از هسته‌های رشته ای. " *The Journal of Machine Learning Re ­search* , 2 , 419-444.

Loh WY (2002). "درختان رگرسیون با انتخاب متغیر بی طرف و تشخیص تعامل. " *Statistica Sinica* , 12 , 361-386.

Loh WY (2010). «طبقه‌بندی‌کننده‌های درختی». *Wiley Interdisciplinary Re ­views: Computational Statistics* , 2 , 364-369.

Loh WY، Shih YS (1997). "روش‌های انتخاب تقسیم برای درختان طبقه بندی. " *Statistica Sinica،* 7، 815-840.

ماهه پی، اوئدا ان، آکوتسو تی، پرت جی، ورت جی (2005). "هسته‌های نمودار برای تحلیل رابطه ساختار مولکولی-فعالیت با ماشین‌های بردار پشتیبان. " *مجله اطلاعات و مدل‌سازی شیمیایی،* 45 (4)، 939-951.

ماهه پی، ورت جی (2009). "هسته‌های نمودار بر اساس الگوهای درختی برای مولکول ها. " *یادگیری ماشینی،* 75 (1)، 3-35.

Maindonald J، Braun J (2007). *تحلیل داده‌ها و گرافیک با استفاده* از *R.* انتشارات دانشگاه کم بریج، چاپ دوم. ­

ماندال A، جانسون کی، وو سی، بورنمایر دی (2007). "شناسایی ترکیبات امیدوارکننده در کشف دارو: الگوریتم‌های ژنتیک و برخی ­تکنیک‌های آماری جدید. " *مجله اطلاعات و مدل‌سازی شیمیایی،* 47 (3)، 981-988.

ماندال A، وو سی، جانسون کی (2006). SELC: حذف متوالی ترکیب‌های سطح با استفاده از الگوریتم‌های ژنتیک اصلاح‌شده. فن سنجی، 48 (2)، 273-283.

مارتین جی، هیرشبرگ دی (1996). "آمار نمونه کوچک برای نرخ خطای طبقه‌بندی I: اندازه‌گیری نرخ خطا. " *گزارش فنی گروه انفورماتیک و علوم کامپیوتر.*

مارتین تی، هارتن پی، یانگ دی، موراتوف ای، گلبریخ آ، ژو اچ، تروپشا آ (2012). «آیا انتخاب منطقی مجموعه‌های آموزشی و آزمایشی نتیجه مدل‌سازی QSAR را بهبود می‌بخشد؟» *Journal of Chemical Information and Mod ­eling ,* 52 (10), 2570-2578.

ماسی دبلیو (1965). "رگرسیون مولفه‌های اصلی در تحقیقات آماری اکتشافی. " ­*مجله انجمن آماری آمریکا،* 60، 234-246.

McCarren P، Springer C، Whitehead L (2011). "بررسی ­داده‌های جهش زایی دارویی مرتبط و تاثیر بر ­پتانسیل پیش‌بینی آمیز. " *مجله شیمی انفورماتیک* , 3 (51).

مک کلیش دی (1989). "تحلیل بخشی از منحنی ROC. " *تصمیم‌گیری متخصصی،* 9، 190-195.

Melssen W، Wehrens R، Buydens L (2006). "شبکه‌های کوهونن تحت نظارت برای مسائل طبقه بندی. " *Chemometrics and Intelligent Laboratory Systems ­*, 83 (2)، 99-113.

منته اس، لومباردو اف (2005). "یک مدل پارتیشن‌بندی بازگشتی برای ­نفوذ سد خونی مغز. " *مجله طراحی مولکولی به کمک کامپیوتر،* 19 (7)، 465-481.

Menze B، Kelm B، Splitthoff D، Koethe U، Hamprecht F (2011). "در جنگل‌های تصادفی مایل. " *یادگیری ماشین و کشف دانش در پایگاه‌های داده،* صفحات 453-469.

Mevik B، Wehrens R (2007). " بسته pls : مولفه اصلی و رگرسیون جزئی حداقل مربعات در R. " *مجله نرم‌افزار آماری* , 18 (2), 1-24.

Michailidis G، de Leeuw J (1998). "سامانه ­گیفی تحلیل توصیفی چند متغیره. " *علوم آماری* , 13 , 307-336.

Milborrow S (2012). *بسته یادداشت‌ها در زمین.* URL [http://cran. r-project.](http://cran.r-project.org/package=earth)  [org/package=earth.](http://cran.r-project.org/package=earth)

مین اس، لی جی، هان‌ای (2006). "الگوریتم‌های ترکیبی ژنتیک و ماشین‌های بردار پشتیبان برای پیش‌بینی ورشکستگی. " *سامانه‌های خبره با برنامه‌های کاربردی* , 31 (3), 652-660.

میچل ام (1998). *مقدمه‌ای بر الگوریتم‌های* ژنتیک مطبوعات MIT.

مولینارو A (2005). "تخمین خطای پیش‌بینی: مقایسه روش‌های Resam Pling. " *بیوانفورماتیک،* 21 (15)، 3301-3307.

Molinaro A، Lostritto K، Van Der Laan M (2010). " partDSA : الگوریتم حذف/جایگزینی/افزودن برای تقسیم‌بندی فضای کمکی در پیش‌بینی. ” *بیوانفورماتیک* , 26 (10), 1357-1363.

مونتگومری دی، رانگر جی (1993). "قابلیت سنج و آزمایش‌های طراحی شده. بخش اول: روش‌های اساسی. *مهندسی کیفیت* , 6 (1), 115-135.

Muenchen R (2009). *R برای کاربران SAS و SPSS.* اسپرینگر.

مایرز آر (1994). *رگرسیون کلاسیک و مدرن با کاربردها.* PWS- KENT Publishing Company، بوستون، MA، چاپ دوم.

مایرز آر، مونتگومری دی (2009). *روش‌شناسی سطح پاسخ: بهینه‌سازی فرآیند و محصول با استفاده از آزمایش‌های طراحی‌شده.* وایلی، نیویورک، نیویورک

نیل آر (1996). *آموزش بیزی برای شبکه‌های* عصبی Springer-Verlag.

نلدر جی، مید آر (1965). "روشی ساده برای کمینه‌سازی عملکرد. " *The Computer Journal ,* 7 (4), 308-313.

نتزوا تی، ورث، آلدنبرگ تی، بنینی آر، کرونین ام، گراماتیکا پی، جاورسکا جی، کان اس، کلوپمن جی، مارچانت سی (2005). «وضعیت فعلی روش‌ها برای تعریف دامنه کاربردی روابط ساختار-فعالیت (کمی)». در کارگاه «گزارش و توصیه‌های مرکز اروپایی اعتبارسنجی روش‌های جایگزین 52» جلد 33، ص 1-19.

Niblett T (1987). "ساخت درختان تصمیم در دامنه‌های پر سر و صدا. " در I Bratko، N Lavrac (ویرایش‌ها)، "پیشرفت در یادگیری ماشین: مجموعه مقالات EWSL-87"، صفحات 67-78. چاپ سیگما، بلد، یوگسلاوی.

اولدن جی، جکسون دی (2000). «داده‌های عذاب‌آور به‌خاطر کلیت: مدل‌های رگرسیون ما چقدر معتبر هستند؟» *Ecoscience،* 7 (4)، 501-510.

اولسون دی، نلسون ال (1975). "روش ساده نلدر مید برای به ­حداقل رساندن عملکرد. " *فن سنجی،* 17 (1)، 45-51.

Osuna E, Freund R, Girosi F (1997). "ماشین‌های بردار پشتیبانی: آموزش و کاربردها. " *گزارش فنی،* آزمایشگاه هوش مصنوعی MIT.

Ozuysal M، Calonder M، Lepetit V، Fua P (2010). "تشخیص سریع نقطه کلید ­با استفاده از سرخس‌های تصادفی. " *معاملات IEEE در تحلیل الگو و هوش ماشینی،* 32 (3)، 448-461.

پارک ام، هیستی تی (2008). "رگرسیون لجستیک مجازات شده برای تشخیص تعاملات ژن. " *آمار زیستی،* 9 (1)، 30.

پپه ام اس، لانگتون جی، جینز اچ (2009). "برآورد و مقایسه ­منحنی‌های مشخصه عملکرد گیرنده. " *Stata Journal* , 9 (1), 1-16.

Perrone M، Cooper L (1993). "وقتی شبکه‌ها موافق نیستند: روش‌های مجموعه‌ای برای شبکه‌های عصبی ترکیبی. " در RJ Mammone (ویرایش)، "شبکه‌های عصبی مصنوعی برای گفتار و بینایی"، صفحات 126-142. چپمن و هال، لندن.

پیرسما A، Genschow E، Verhoef A، Spanjersberg M، Brown N، Brady M، Burns A، Clemann N، Seiler A، Spielmann H (2004). اعتبار سنجی آزمایش کشت کامل جنین موش صحرایی پس از لانه گزینی در مطالعه اعتبارسنجی بین المللی EC-VAM روی سه آزمایش سمیت جنینی آزمایشگاهی. جایگزینی *برای حیوانات آزمایشگاهی،* 32، *275-307.*

پلات جی (2000). «خروجی‌های احتمالی برای ماشین‌های بردار پشتیبان و ­مقایسه با روش‌های درست‌نمایی منظم». در B Bartlett, B Scholkopf, D Schuurmans, A Smola (ویرایش‌ها)، «پیشرفت‌ها در روش‌های هسته از یادگیری برداری پشتیبانی می‌کنند،» صفحات 61-74. کمبریج، MA: مطبوعات MIT.

پروست اف، دومینگوس پی (2003). "القای درختی برای رتبه‌بندی بر اساس احتمال. " *یادگیری ماشینی،* 52 (3)، 199-215.

Provost F, Fawcett T, Kohavi R (1998). "تخمین موردی در برابر دقت ­برای مقایسه الگوریتم‌های القایی. " *مجموعه مقالات پانزدهمین کنفرانس بین المللی یادگیری ماشین،* صفحات 445-453.

کوینلان آر (1987). "ساده‌سازی درختان تصمیم گیری. " *مجله بین المللی مطالعات انسان و ماشین،* 27 (3)، 221-234.

کوینلان آر (1992). "یادگیری با کلاس‌های مداوم. " *مجموعه مقالات پنجمین کنفرانس مشترک استرالیا در زمینه هوش مصنوعی،* صفحات 343-348.

Quinlan R (1993a). "ترکیب یادگیری مبتنی بر نمونه و مبتنی بر مدل. " *مجموعه مقالات دهمین کنفرانس بین المللی یادگیری ماشینی،* صفحات 236-243.

Quinlan R (1993b). *C4. 5 : برنامه‌هایی برای یادگیری ماشینی.* ناشران مورگان کافمن

Quinlan R (1996a). "کیف زدن، تقویت و C4. 5. " در مجموعه مقالات سیزدهمین کنفرانس ملی هوش مصنوعی،.

Quinlan R (1996b). "استفاده بهبود یافته از ویژگی‌های پیوسته در C4. 5. " *مجله تحقیقات هوش مصنوعی،* 4، 77-90.

Quinlan R، Rivest R (1989). "استنتاج درختان تصمیم با استفاده از ­اصل طول توضیحات Mini Mum. " *اطلاعات و محاسبات* , 80 (3), 227-248.

رادکلیف ان، سوری پی (2011). "مدل‌سازی در دنیای واقعی با ­درختان برافراشته بر اساس اهمیت. " *گزارش فنی،* راه حل‌های تصادفی.

Rannar S، Lindgren F، Geladi P، Wold S (1994). یک الگوریتم هسته PLS برای مجموعه‌های داده با متغیرهای زیاد و اشیاء کمتر. بخش 1: نظریه و الگوریتم. *مجله شیمی سنجی،* 8، 111-125.

تیم هسته توسعه R (2008). *R : مسائل مربوط به انطباق با مقررات و اعتبار سند راهنمایی برای استفاده از R در محیط‌های آزمایشی بالینی تنظیم شده.* بنیاد R برای محاسبات آماری، وین، اتریش.

R (2010). *R : زبان و محیطی برای محاسبات آماری.* بنیاد R برای محاسبات آماری، وین، اتریش.

Reshef D، Reshef Y، Finucane H، Grossman S، McVean G، Turnbaugh P، Lander E، Mitzenmacher M، Sabeti P (2011). "تشخیص ارتباطات جدید در مجموعه داده‌های بزرگ. " *علم،* 334 (6062)، 1518-1524.

ریچاردسون ام، دومینوفسکا ای، راگنو آر (2007). "پیش‌بینی کلیک ها: تخمین ­نرخ کلیک برای تبلیغات جدید. " در «مجموعه مقالات شانزدهمین *کنفرانس* بین المللی شبکه جهانی وب»، ص 521-530.

Ridgeway G (2007). "مدل‌های تقویت شده عمومی: راهنمای عصر بسته ­gbm. " URL [http://cran. r-project. org/web/packages/gbm/vignettes/gbm.](http://cran.r-project.org/web/packages/gbm/vignettes/gbm.pdf)  [پی دی اف](http://cran.r-project.org/web/packages/gbm/vignettes/gbm.pdf) .

ریپلی بی (1995). "ایده‌های آماری برای انتخاب معماری شبکه. " *شبکه‌های عصبی: هوش مصنوعی و کاربردهای صنعتی،* ص 183-190.

ریپلی بی (1996). *تشخیص الگو و شبکه‌های عصبی.* انتشارات دانشگاه کمبریج.

رابین ایکس، تورک ان، هاینارد آ، تیبرتی ان، لیزاچک اف، سانچز جی سی، مولر ام (2011). " pROC : یک بسته منبع باز برای R و S+ برای تحلیل و مقایسه منحنی‌های ROC. " *BMC Bioinformatics،* 12 (1)، 77.

Robnik-Sikonja M، Kononenko I (1997). "اقتباسی از امداد برای ­برآورد ادای احترام در رگرسیون. " *مجموعه مقالات چهاردهمین ­کنفرانس بین المللی یادگیری ماشینی،* صفحات 296-304.

رودریگز ام (2011). "شکست مدل‌سازی پیش‌بینی و چرا ما از گله پیروی می‌کنیم. " *گزارش فنی،* Concepcion، Martinez & Bellido.

Ruczinski I، Kooperberg C، Leblanc M (2003). "رگرسیون منطقی. " *مجله آمار محاسباتی و گرافیکی ,* 12 (3), 475-511.

روملهارت دی، هینتون جی، ویلیامز آر (1986). "یادگیری بازنمایی‌های داخلی با انتشار خطا. " انتشارات MIT در کتاب «پردازش موازی توزیع شده: اکتشافات ­در ریزساختار شناخت».

Rzepakowski P، Jaroszewicz S (2012). "مدل‌سازی ارتقا در بازاریابی مستقیم. " *مجله مخابرات و فناوری اطلاعات،* 2، 43-50.

Saar-Tsechansky M، Provost F (2007a). "یادگیری فعال تصمیم محور مدل‌های باینری نتیجه. " *تحقیقات سامانه‌های اطلاعاتی* , 18 (1), 4-22.

Saar-Tsechansky M، Provost F (2007b). "کنترل مقادیر از دست رفته هنگام ­استفاده از مدل‌های طبقه بندی. " *مجله تحقیقات یادگیری ماشین،* 8، 1625-1657.

Saeys Y، Inza I، Larranaga P (2007). "مروری بر تکنیک‌های انتخاب ویژگی ­در بیوانفورماتیک. " *بیوانفورماتیک،* 23 (19)، 2507-2517.

Schapire R (1990). "قدرت یادگیری ضعیف. " *یادگیری ماشینی،* 45، 197-227.

Schapire YFR (1999). "بازی تطبیقی با استفاده از وزن‌های ضربی. " *بازی‌ها و رفتار اقتصادی،* 29، 79-103.

Schmidberger M، Morgan M، Eddelbuettel D، Yu H، Tierney L، Mansmann U (2009). "آخرین روز در محاسبات موازی با R. " *مجله نرم‌افزار آماری* , 31 (1).

Serneels S، Nolf ED، Espen PV (2006). "پیش پردازش علائم فضایی: روشی ساده برای ایجاد استحکام متوسط به برآوردگرهای چند متغیره. " *مجله اطلاعات شیمیایی و مدلسازی* , 46 (3), 1402-1409.

Shachtman N (2011). "نرم‌افزار پیش‌بینی پنتاگون ناآرامی مصر را شناسایی نکرد. " *سیمی.*

شانون سی (1948). "نظریه ریاضی ارتباطات. " *مجله فنی سامانه بل،* 27 (3)، 379-423.

سیگل‌ای (2011). "مدل‌سازی ارتقاء: تحلیل پیش‌بینی نمی‌تواند ­تصمیمات بازاریابی را بدون آن بهینه کند. " *گزارش فنی،* Prediction Impact Inc.

سیمون آر، رادماچر ام، دابین کی، مک شین ال (2003). مسائل در استفاده از داده‌های ریزآرایه DNA برای طبقه‌بندی تشخیصی و پیش آگهی. *مجله موسسه ملی سرطان،* 95 (1)، 14-18.

اسمولا A (1996). "تخمین رگرسیون با بردار پشتیبان یادگیری ما ­چین. " *پایان نامه کارشناسی ارشد، دانشگاه فنی در مونشن.*

اسپکتور پی (2008). *دستکاری داده‌ها* با *R.* اسپرینگر.

Steyerberg E (2010). *مدل‌های پیش‌بینی بالینی: رویکردی عملی برای توسعه، اعتبارسنجی و به‌روزرسانی.* اسپرینگر، چاپ اول. جلد نرم اصل ویرایش نسخه 2009.

استون ام، بروکس آر (1990). "رگرسیون پیوسته: ­پیش‌بینی متقابل ساخته شده متقابل که شامل حداقل مربعات معمولی، حداقل مربعات جزئی و رگرسیون مولفه اصلی است. " *مجله انجمن آمار سلطنتی، سری B،* 52، 237-269.

Strobl C، Boulesteix A، Zeileis A، Hothorn T (2007). "بایاس در معیارهای اهمیت متغیر تصادفی جنگل: تصاویر، منابع و یک راه حل. " *BMC Bioinformatics،* 8 (1)، 25.

Suykens J، Vandewalle J (1999). «دسته‌بندی‌کننده‌های ماشین بردار حداقل مربعات پشتیبانی می‌کند». *حروف پردازش عصبی،* 9 (3)، 293-300.

Tetko I, Tanchuk V, Kasheva T, Villa A (2001). "برآورد انحلال ­پذیری ترکیبات شیمیایی در آب با استفاده از شاخص‌های E-State. " *مجله اطلاعات شیمی و علوم کامپیوتر،* 41 (6)، 1488-1493.

تبشیرانی ر (1375). "انقباض رگرسیون و انتخاب از طریق ریج. " *مجله انجمن آماری سلطنتی سری B (روش شناختی)،* 58 (1)، 267-288.

تیبشیرانی ر، هستی تی، نراسیمهان بی، چو جی (2002). "تشخیص انواع مختلف سرطان توسط مرکزهای کوچک بیان ژن. " *مجموعه مقالات آکادمی ملی علوم،* 99 (10)، 6567-6572.

تیبشیرانی ر، هستی تی، نراسیمهان بی، چو جی (2003). «پیش‌بینی کلاس توسط نزدیک‌ترین مرکز کوچک‌شده، با کاربرد در ریزآرایه‌های DNA». *علوم آماری* , 18 (1), 104-117.

Ting K (2002). "روش وزن دهی نمونه برای القای درختان حساس به هزینه. " *IEEE Transactions on Knowledge and Data Engineering،* 14 (3)، 659-665.

انعام M (2001). "یادگیری پراکنده بیزی و بردار ارتباط ما ­چین. " *مجله تحقیقات یادگیری ماشین،* 1، 211-244.

تیترینگتون ام (2010). "شبکه‌های عصبی. " *بررسی‌های بین رشته‌ای وایلی: آمار محاسباتی،* 2 (1)، 1-8.

Troyanskaya O, Cantor M, Sherlock G, Brown P, Hastie T, Tibshirani R, Botstein D, Altman R (2001). "روش‌های تخمین ارزش گمشده برای ریزآرایه‌های DNA. " *بیوانفورماتیک،* 17 (6)، 520-525.

Tumer K، Ghosh J (1996). "تحلیل مرزهای تصمیم‌گیری در ­طبقه‌بندی کننده‌های عصبی ترکیبی خطی. " *تشخیص الگو،* 29 (2)، 341-348.

کمیسیون معاملات آتی کالای ایالات متحده و کمیسیون بورس و اوراق بهادار ایالات متحده (2010). *یافته‌های مربوط به رویدادهای بازار 6 می‌2010.*

Valiant L (1984). "نظریه‌ای در مورد آموختنی ها. " *ارتباطات ACM،* 27، 1134-1142.

Van Der Putten P، Van Someren M (2004). "تحلیل بایاس-واریانس یک مسأله یادگیری دنیای واقعی: چالش CoIL 2000. " *یادگیری ماشینی،* 57 ( 1)، 177-195.

ون هولس جی، خوش گفتار تی، ناپولیتانو آ (2007). "دیدگاه‌های تجربی ­در مورد یادگیری از داده‌های نامتعادل. " در "مجموعه مقالات بیست و *چهارمین کنفرانس* بین المللی در مورد یادگیری ماشین"، صفحات 935-942.

Vapnik V (2010). *ماهیت نظریه یادگیری آماری.* اسپرینگر.

وارما اس، سایمون آر (2006). "بایاس در تخمین خطا هنگام استفاده از ­اعتبارسنجی متقاطع برای انتخاب مدل. " *BMC Bioinformatics،* 7 (1)، 91.

وارموزا ک، هی پی، نیش ک (2003). "تقویت برای طبقه‌بندی داده‌های طیف جرمی اعمال می‌شود. " *مجله علم داده* , 1 , 391-404.

Venables W، Ripley B (2002). *آمار کاربردی مدرن* با *S.* اسپرینگر.

Venables W، Smith D، تیم هسته توسعه R (2003). *مقدمه‌ای* بر *R. بنیاد* R برای محاسبات آماری، وین، اتریش، نسخه 1. 6. 2 ویرایش. ISBN 3-901167-55-2، URL [http://www. R-pro ject. org.](http://www.R-project.org)

Venkatraman E (2000). "یک تست جایگشت برای مقایسه منحنی‌های مشخصه عملکرد گیرنده. " *بیومعیار،* 56 (4)، 1134-1138.

Veropoulos K، Campbell C، Cristianini N (1999). "کنترل حساسیت ­ماشین‌های بردار پشتیبان. " *مجموعه مقالات کنفرانس مشترک بین المللی هوش مصنوعی،* 1999، 55-60.

ورزانی ج (1381). " simpleR - استفاده از R برای آمار مقدماتی. " URL [http://](http://www.math.csi.cuny.edu/Statistics/R/simpleR) [www. math. csi. cuny. edu/Statistics/R/simpleR](http://www.math.csi.cuny.edu/Statistics/R/simpleR) .

Wager TT، Hou X، Verhoest PR، Villalobos A (2010). "حرکت فراتر از قوانین: توسعه یک ­رویکرد بهینه‌سازی چندپارامتری سامانه عصبی مرکزی (CNS MPO) برای فعال کردن همسویی خواص دارویی. " *ACS Chemical Neuroscience،* 1 (6)، 435-449.

والاس سی (2005). *استنتاج آماری و استقرایی با حداقل طول پیام.* Springer-Verlag.

وانگ سی، ونکاتش اس (1984). توقف بهینه و ترکیب موثر ماشین ­در یادگیری. *پیشرفت‌ها در NIPS،* صفحات 303-310.

وانگ ای، ویتن من (1997). القای درختان مدل برای کلاس‌های پیوسته. *مجموعه مقالات نهمین کنفرانس اروپایی یادگیری ماشینی،* صفحات 128-137.

ویس جی، پروست اف (2001a). "تاثیر توزیع کلاس بر یادگیری طبقه‌بندی کننده: یک مطالعه تجربی. " *گروه علوم کامپیوتر، دانشگاه راتگرز.*

ویس جی، پروست اف (2001b). "تاثیر توزیع کلاس بر یادگیری طبقه‌بندی کننده: یک مطالعه تجربی. " *گزارش فنی ML-TR-44،* گروه علوم کامپیوتر، دانشگاه راتگرز.

ولش بی (1939). "یادداشت در مورد عملکردهای متمایز. " *بیومعیارا،* 31، 218-220.

Westfall P، Young S (1993). *آزمایش چندگانه مبتنی بر نمونه‌برداری مجدد: مثال‌ها و روش‌هایی برای تنظیم P-Value.* وایلی.

وستفال سی (2008). *داده کاوی برای اطلاعات، کلاهبرداری* و *تشخیص جرم ­: تحلیل پیشرفته* و *فناوری‌های به اشتراک گذاری اطلاعات.* مطبوعات CRC.

ویتینگهام ام، استفنز پی، بردبری آر، فرکلتون آر (2006). "چرا ما هنوز از مدل‌سازی گام به گام در اکولوژی و رفتار استفاده می‌کنیم؟" *مجله بوم شناسی حیوانات* , 75 (5), 1182-1189.

ویلت پی (1999). "الگوریتم‌های مبتنی بر عدم تشابه برای انتخاب مجموعه‌های ساختاری متنوع از ترکیبات. " *مجله زیست شناسی محاسباتی* , 6 (3), 447-457.

ویلیامز جی (2011). *داده کاوی با جغجغه و R : هنر کاوش داده‌ها برای کشف دانش.* اسپرینگر.

ویتن دی، تبشیرانی ر (2009). "رگرسیون کوواریانس منظم و طبقه‌بندی برای مسائل با ابعاد بالا. " *مجله انجمن سلطنتی Sta ­tistical. سری B (روش شناسی آماری)،* 71 (3)، 615-636.

ویتن دی، تبشیرانی ر (1390). "طبقه‌بندی مجازات شده با استفاده از تشخیص دهنده گوش فیشر لاین. " *مجله انجمن آمار سلطنتی. سری B ( ­روش آماری)، 73 (5)،* 753-772.

ولد اچ (1966). "برآورد مولفه‌های اصلی و مدل‌های مرتبط با حداقل مربعات تکراری. " در P Krishnaiah (ed. ), "Multivariate Analyses" pp. 391-420. انتشارات آکادمیک، نیویورک.

ولد اچ (1982). "مدل‌سازی نرم: طراحی اساسی و برخی از برنامه‌های افزودنی. " در K Joreskog، H Wold (ویرایشگران)، "سامانه‌های تحت مشاهده غیرمستقیم: علت، ساختار، پیش‌بینی،" pt. 2، ص 1-54. هلند شمالی، آمستردام

ولد اس (1995). "PLS برای مدل‌سازی خطی چند متغیره. " در H van de Water-beemd (ویرایش)، "روش‌های کمومتری در طراحی مولکولی"، صفحات 195-218. VCH، واینهایم.

ولد اس، یوهانسون ام، کوکی ام (1993). PLS-Partial Least-Squares Pro jections to Latent Structures. در H Kubinyi (ویرایش)، "3D QSAR در طراحی دارو،" جلد 1، صفحات 523-550. ناشران آکادمیک Kluwer، هلند.

ولد اس، مارتنز اچ، وولد اچ (1983). "مسئله کالیبراسیون چند متغیره ­در شیمی با روش PLS حل شد. " در «مجموعه مقالات کنفرانس مدادهای ماتریسی»، Springer-Verlag، هایدلبرگ.

ولپرت دی (1996). «عدم تمایز ­*پیشینی بین یادگیری* الگوریتم‌ها». *محاسبات عصبی* , *8 (*7), 1341-1390.

یه من (1998). مدل‌سازی مقاومت بتن با کارایی بالا با استفاده ­از شبکه‌های عصبی مصنوعی. *تحقیقات سیمان و بتن،* *28 (*12)، 1797-1808.

یه من (2006). «تحلیل مقاومت بتن با استفاده از طراحی آزمایش‌ها ­و شبکه‌های عصبی». *مجله مواد در مهندسی عمران،* *18،* 597-604.

یودن دبلیو (1950). "شاخص برای رتبه‌بندی تست‌های تشخیصی. " *سرطان،* *3 (*1)، 32-35. زادروزنی بی، الکان سی (2001). "به دست آوردن تخمین‌های احتمال کالیبره شده از درختان تصمیم‌گیری و طبقه‌بندی کننده‌های ساده بیزی. " در «مجموعه مقالات هجدهمین کنفرانس بین المللی یادگیری ماشین»، صفحات 609-616. مورگان کافمن.

Zeileis A، Hothorn T، Hornik K (2008). "پارتیشن‌بندی بازگشتی مبتنی بر مدل. " *مجله آمار محاسباتی و گرافیکی* , *17 (*2), 492-514.

ژو جی، هستی تی (2005). رگرسیون لجستیک هسته و ­ماشین بردار وارداتی. *مجله آمار محاسباتی و گرافیکی* , *14 (*1), 185-205.

زو اچ، هیستی تی (2005). "قانونی‌سازی و انتخاب متغیر از طریق شبکه الاستیک. " *مجله انجمن آمار سلطنتی، سری B،* *67 (*2)، 301-320.

زو اچ، هستی تی، تبشیرانی ر (2004). "تحلیل مولفه اصلی پراکنده. " *مجله آمار محاسباتی و گرافیکی،* *15،* 2006.

شاخص ها

محاسبه

قطع‌های جایگزین، [438](#bookmark781) [- 439](#bookmark784)

مقایسه بین مدل، [87](#bookmark252) [- 89](#bookmark255)

زیست رسانا، [552](#bookmark968)

بوت استرپ، 415

مطالعات موردی

بیماری آلزایمر، [511](#bookmark905) [- 518](#bookmark914)

ماشین‌ها، [56](#bookmark127) [- 58](#bookmark5)

تقسیم‌بندی سلولی، [51](#bookmark4) [- 56 ،](#bookmark127) [480](#bookmark851) [- 484](#bookmark859)

اعتبار آلمانی، [85](#bookmark249) [- 89](#bookmark255)

درخواست‌های گرنت، [308](#bookmark585) [- 326 ،](#bookmark605) [359](#bookmark658) [- 366 ،](#bookmark677) [400](#bookmark727) [- 411 ،](#bookmark737) 415 [- 417](#bookmark743)

سیاستهای بیمه، [435](#bookmark778) [- 442](#bookmark43)

برنامه ریزی شغلی، [457](#bookmark812) [- 460](#bookmark814)

انحلال پذیری، [128](#bookmark321) [- 136 ،](#bookmark332) [162](#bookmark370) [- 168 ،](#bookmark381) [213](#bookmark440) [- 218 ،](#bookmark457) [478](#bookmark847) [- 480 ،](#bookmark851) [541](#bookmark950) [- 542](#bookmark953)

احتمالات کلاس، [563](#bookmark995)

خط خطی، [311](#bookmark586)

جامع R (CRAN)، 551 [- 553](#bookmark971)

ماتریس سردرگمی، [51 ،](#bookmark4) [268](#bookmark515)

آموزش حساس به هزینه [440](#bookmark787) [- 442](#bookmark43)

ایجاد متغیرهای ساختگی، [56](#bookmark127) [- 58](#bookmark5)

تقسیم داده‌ها، [81](#bookmark239) [- 82](#bookmark242)

شیوع رویداد، [269](#bookmark18) برون یابی، [541](#bookmark950) [- 542](#bookmark953)

انتخاب ویژگی

به عقب، [511](#bookmark905) [- 513](#bookmark909)

روش‌های فیلتر، [516](#bookmark47) [- 518](#bookmark914)

رو به جلو، [511](#bookmark905) [- 513](#bookmark909)

ویژگی بازگشتی

حذف، [513](#bookmark909) [- 516](#bookmark47)

گام به گام، [511](#bookmark905) [- 513](#bookmark909)

فیلتر کردن

همبستگی بالا، [55](#bookmark184) [- 56](#bookmark127)

واریانس نزدیک به صفر، [55](#bookmark184) تحلیل تفکیک انعطاف پذیر، [362](#bookmark665)

نسبت، [54](#bookmark181)

*K* -نزدیکترین همسایه ها

طبقه بندی، [83 ،](#bookmark245) [364](#bookmark671) [- 365](#bookmark674)

پسرفت، [168](#bookmark381)

تحلیل تشخیص خطی، [318](#bookmark594) [- 320](#bookmark597)

رگرسیون لجستیک، [87](#bookmark252) [- 89 ،](#bookmark255) [312](#bookmark589) [- 318](#bookmark594)

تنظیم مدل، [84](#bookmark248) [- 87](#bookmark252)

خطوط رگرسیون تطبیقی چند متغیره (MARS)

پسرفت، [163](#bookmark373) [- 166](#bookmark376)

بیز ساده لوح، [365](#bookmark674) [- 366](#bookmark677)

پیش‌بینی‌کننده‌های واریانس نزدیک به صفر، [310](#bookmark28)

نزدیکترین مرکزهای کوچک شده، [324](#bookmark30) [- 326](#bookmark605)

ارزش پیش‌بینی شده منفی، [268](#bookmark515) شبکه‌های عصبی

طبقه بندی، [360](#bookmark661) [- 361](#bookmark662)

پسرفت، [162](#bookmark370) [- 163](#bookmark373)

تحلیل تفکیک غیرخطی، [359](#bookmark658) [- 360](#bookmark661)

برنامه نویسی شی گرا، [560](#bookmark984)

رگرسیون حداقل مربعات معمولی، [128](#bookmark321) [- 132](#bookmark324)

حداقل مربعات جزئی

طبقه بندی، [320](#bookmark597) [- 321](#bookmark598)

پسرفت، [133](#bookmark327) [- 134](#bookmark330)

مدل‌های طبقه‌بندی مجازات شده تشخیص دهنده خطی

تحلیل و بررسی، [323](#bookmark29)

رگرسیون لجستیک، [322](#bookmark578) [- 323](#bookmark29) مدل‌های رگرسیون مجازات شده

شبکه الاستیک، [136](#bookmark332)

ریج، [135](#bookmark331) [- 136](#bookmark332)

رگرسیون ریج، [134](#bookmark330) [- 135](#bookmark331)

ارزش پیش‌بینی شده مثبت، [268](#bookmark515)

پیش‌بینی، [84](#bookmark248)

اهمیت پیش‌بینی

نتایج طبقه‌بندی شده، [480](#bookmark851) [- 483](#bookmark856)

امتیازهای اهمیت مبتنی بر مدل،  
 [483](#bookmark856) [- 484](#bookmark859)

نتایج عددی، [478](#bookmark847) [- 480](#bookmark851)

بسته‌های R

AppliedPredictiveModeling، viii، [51 ،](#bookmark4) [81 ،](#bookmark239) [89 ،](#bookmark255) [128 ،](#bookmark321) [236 ،](#bookmark483) [266 ،](#bookmark529) [309 ،](#bookmark27) [481 ،](#bookmark852) [485 ،](#bookmark860) [511 ،](#bookmark905) [562](#bookmark992)

C50، [328 ،](#bookmark606) [404 ،](#bookmark730) [441](#bookmark788)

CORElearn، [481](#bookmark852)

کوبیست، [217](#bookmark454)

DMwR، [439](#bookmark784)

DWD، [435](#bookmark778)

طراحی، [87](#bookmark252)

Hmisc، [237](#bookmark484)

جرم، [52 ،](#bookmark179) [83 ،](#bookmark245) [130 ،](#bookmark322) [134 ،](#bookmark330) [266 ،](#bookmark529) [318 ،](#bookmark594) [359 ،](#bookmark658) [512 ،](#bookmark906) [563](#bookmark995)

RWeka، [214 ،](#bookmark443) [215 ،](#bookmark448) [403 ،](#bookmark38) [406 ،](#bookmark732) [563 ،](#bookmark995) [564](#bookmark996)

caTools، [563](#bookmark995)

گاری، [51 ،](#bookmark4) [53](#bookmark180) [- 56 ،](#bookmark127) [81](#bookmark239) [- 85 ،](#bookmark249) [99 ،](#bookmark274) [130 ،](#bookmark322) [162 ،](#bookmark370) [168 ،](#bookmark381) [237 ،](#bookmark484) [240 ،](#bookmark485) [266 ،](#bookmark529) [268 ،](#bookmark515) [364 ،](#bookmark671) [439 ،](#bookmark784) [479 ،](#bookmark848) [483 ،](#bookmark856) [512 ،](#bookmark906) [513 ،](#bookmark909) [516](#bookmark47) [- 518 ،](#bookmark914) [563](#bookmark995)

خرابکاری، [51 ،](#bookmark4) [55](#bookmark184)

سی تی وی، [553](#bookmark971)

مطلوبیت، [243](#bookmark486)

e1071 \_ [51 ،](#bookmark4) [52 ،](#bookmark179) [84 ،](#bookmark248) [87 ،](#bookmark252) [166 ،](#bookmark376) [365](#bookmark674)

زمین، [163 ،](#bookmark373) [362](#bookmark665)

شبکه الاستیک، [134](#bookmark330)

پیشروی، [563](#bookmark995)

gbm، [216 ،](#bookmark451) [271 ،](#bookmark541) [409 ،](#bookmark734) [563](#bookmark995)

glmnet \_ [135 ،](#bookmark331) [322](#bookmark578)

نسبت دادن، [54](#bookmark181)

آیپرد \_ [83 ،](#bookmark245) [84 ،](#bookmark248) [87 ،](#bookmark252) [215 ،](#bookmark448) [408](#bookmark733)

کرنلب، [166 ،](#bookmark376) [363 ،](#bookmark668) [440](#bookmark787)

klaR، [266 ،](#bookmark529) [273 ،](#bookmark543) [359 ،](#bookmark658) [365 ،](#bookmark674) [512](#bookmark906)

لارس، [135](#bookmark331)

شبکه، [51 ،](#bookmark4) [52 ،](#bookmark179) [271 ،](#bookmark541) [479](#bookmark848)

جهش، [512](#bookmark906)

mda، [359 ،](#bookmark658) [362 ،](#bookmark665) [563](#bookmark995)

مینروا، [479](#bookmark848)

nnet \_ [161 ،](#bookmark367) [162 ،](#bookmark370) [360 ،](#bookmark661) [563](#bookmark995)

PROC، [266 ،](#bookmark529) [269 ،](#bookmark18) [318 ،](#bookmark594) [438 ،](#bookmark781) [481](#bookmark852)

پامر، [324](#bookmark30)

کیت مهمانی، [214 ،](#bookmark443) [404](#bookmark730)

مهمانی، [212 ،](#bookmark439) [215 ،](#bookmark448) [216](#bookmark451)

خواهش میکنم [133 ،](#bookmark327) [320](#bookmark597)

جنگل تصادفی، [215 ،](#bookmark448) [216 ،](#bookmark451) [266 ،](#bookmark529)

[408 ،](#bookmark733) [484](#bookmark859)

rms \_ [314](#bookmark590)

قسمت r [212 ،](#bookmark439) [402 ،](#bookmark729) [441 ،](#bookmark788) [553](#bookmark971)

rrcov \_ [359](#bookmark658)

SparseLDA، [323](#bookmark29)

آمار، [478 ،](#bookmark847) [482 ،](#bookmark853) [511 ،](#bookmark905) [563](#bookmark995)

انتخاب فرعی، [311](#bookmark586)

زبان برنامه نویسی R

روش‌های S3 [561](#bookmark987)

روش‌های S4 [561](#bookmark987)

مقادیر شخصیت، [555](#bookmark975)

کلاس‌ها، [560](#bookmark984)

فریم‌های داده، [559](#bookmark981)

عوامل، [556](#bookmark976)

کارکرد، [561](#bookmark987)

لیست‌ها، [557](#bookmark977)

مقادیر منطقی [554](#bookmark974)

ماتریس ها [558](#bookmark980)

مواد و روش‌ها، [560](#bookmark984)

مقادیر عددی، [554](#bookmark974)

بسته‌ها، [552](#bookmark968)

بردارها، [555](#bookmark975)

ویژگی عملکرد گیرنده

(ROC) منحنی، [269](#bookmark18) [- 270](#bookmark538)

ماشین‌های بردار مرتبط، [168](#bookmark381)

نمونه‌گیری مجدد، [82](#bookmark242) [- 83](#bookmark245)

باقیمانده‌ها، [131 ،](#bookmark323) [132](#bookmark324)

خطوط مکعبی محدود، [314](#bookmark590) [- 315](#bookmark591)

رگرسیون حداقل مربعات قوی، [132](#bookmark324) [- 133](#bookmark327)

RSiteSearch، [51](#bookmark4)

روش‌های نمونه گیری، [439](#bookmark784)

حساسیت، [268](#bookmark515)

اختصاصی، [268](#bookmark515)

مشخص کردن مدل ها

رابط فرمول، [83](#bookmark245)

رابط غیر فرمول، [83](#bookmark245)

ماشین‌های بردار پشتیبانی، [84](#bookmark248) [- 87](#bookmark252)

طبقه بندی، [363](#bookmark668) [- 364](#bookmark671)

پسرفت، [166](#bookmark376) [- 168](#bookmark381)

بردارهای پشتیبانی، [168](#bookmark381)

تحولات

باکس کاکس، [52](#bookmark179) [- 53](#bookmark180)

PCA، [53](#bookmark180) [- 54](#bookmark181)

نشانه فضایی، [54](#bookmark181)

مدل‌های درختی

درختان کیسه ای، [215 ،](#bookmark448) [408](#bookmark733)

درختان تقویت شده، [216](#bookmark451) [- 217 ،](#bookmark454) [409](#bookmark734) [- 411](#bookmark737)

درختان طبقه بندی، [402](#bookmark729) [- 405](#bookmark731)

کوبیست، [217](#bookmark454) [- 218](#bookmark457)

درختان مدل، [214](#bookmark443) [- 215](#bookmark448)

جنگل تصادفی، [215](#bookmark448) [- 216 ،](#bookmark451) [408](#bookmark733) [- 409](#bookmark734)

درختان رگرسیون، [212](#bookmark439) [- 214](#bookmark443)

قوانین، [406](#bookmark732) [- 408](#bookmark733)

عمومی

*e* -رگرسیون غیر حساس، [151](#bookmark356)

یادگیری فعال، [226](#bookmark12)

مدل‌های افزودنی، [148 ،](#bookmark351) [150](#bookmark353) [- 152 ،](#bookmark357) [205 ،](#bookmark427) [285 ،](#bookmark559) [338 ،](#bookmark629) [341 ،](#bookmark632) [342 ،](#bookmark633) [421](#bookmark41)

معیار اطلاعات آکایک، [493](#bookmark44)

بیماری آلزایمر، [502](#bookmark46) [- 504](#bookmark894)

برآورد ظاهری عملکرد، [64 ،](#bookmark199) [65 ،](#bookmark200) [73 ،](#bookmark222) [74 ،](#bookmark6) [143 ،](#bookmark344) [148](#bookmark351)

دامنه کاربردی، [535](#bookmark941)

کوله بری

FDA، [342](#bookmark633) [- 343](#bookmark32)

رگرسیون خطی، [194](#bookmark413) [- 195](#bookmark414)

مریخ، [194](#bookmark413) [- 195](#bookmark414)

درختان، [192](#bookmark411) [- 195 ،](#bookmark414) [206 ،](#bookmark428) 221، [230 ،](#bookmark474) [385](#bookmark699) [- 386 ،](#bookmark702) [453 ،](#bookmark804) [456 ،](#bookmark809) [537](#bookmark943)

قانون بیز، [250 ،](#bookmark502) 287 [- 289 ،](#bookmark563) [300 ،](#bookmark25) [301 ،](#bookmark573) [353](#bookmark648) [- 354](#bookmark649)

Binning داده‌ها، [49](#bookmark173) [- 50 ،](#bookmark174) [122 ،](#bookmark311) [447](#bookmark797) [- 448 ،](#bookmark798) [531](#bookmark49) [- 534](#bookmark940)

توزیع دو جمله ای، [282](#bookmark557) [- 285 ،](#bookmark559) [303 ،](#bookmark26) [333](#bookmark622)

نشانگرهای زیستی، [502](#bookmark46) [- 503](#bookmark893)

تصحیح بونفرونی، [499 ،](#bookmark886) [509](#bookmark899)

افزایش

C5. 0، [396](#bookmark719) [- 397](#bookmark720)

مدل‌های درختی، [79 ،](#bookmark233) [203](#bookmark425) [- 208 ،](#bookmark432) 221، [230 ،](#bookmark474) [389](#bookmark37) [- 392](#bookmark715)

بوت استرپ، [70 ،](#bookmark211) [72](#bookmark219) [- 73 ،](#bookmark222) [76 ،](#bookmark226) [78 ،](#bookmark232) [110 ،](#bookmark290) [197 ،](#bookmark416) [264 ،](#bookmark523) [427 ،](#bookmark767) [428 ،](#bookmark768) [501](#bookmark890)

کوله بری، [192](#bookmark411) [- 194 ،](#bookmark413) [198](#bookmark419)

برآورد عملکرد، [70 ،](#bookmark211) [72](#bookmark219) [- 73 ،](#bookmark222) [76 ،](#bookmark226) [78 ،](#bookmark232) [110 ،](#bookmark290) [501](#bookmark890)

جنگل‌های تصادفی، [198](#bookmark419) [- 199](#bookmark420)

تبدیل جعبه کاکس، [32](#bookmark2) [- 33 ،](#bookmark147) [38 ،](#bookmark152) [111](#bookmark293)

C4. 5، [377](#bookmark689) [- 383](#bookmark695)

C5. 0، [392](#bookmark715) [- 400 ،](#bookmark727) [432 ،](#bookmark774) [434 ،](#bookmark42) [454 ،](#bookmark807) [456](#bookmark809)

طرح کالیبراسیون، [249](#bookmark501)

مطالعات موردی

تقسیم‌بندی سلولی، [28](#bookmark134) [- 33 ،](#bookmark147) [39](#bookmark153) [- 40 ،](#bookmark154) [43 ،](#bookmark161) [47 ،](#bookmark169) [468](#bookmark831) [- 470 ،](#bookmark833) [476](#bookmark841)

اختلال شناختی، [502](#bookmark46) [- 510](#bookmark900)

حلالیت ترکیب، [102](#bookmark281) [- 105 ،](#bookmark286) [144](#bookmark345) [- 146 ،](#bookmark349) [149 ،](#bookmark352) [151 ،](#bookmark356) [157 ،](#bookmark361) [160 ،](#bookmark364) [186](#bookmark404) [- 192 ،](#bookmark411) [211](#bookmark434) [- 212 ،](#bookmark439) 221 [- 223 ،](#bookmark463) [464](#bookmark825) [- 468 ،](#bookmark831) [488](#bookmark867) [- 490 ،](#bookmark871) [525](#bookmark924) [- 527 ،](#bookmark928) [532](#bookmark936) [- 533](#bookmark937)

مقاومت بتن، [196 ،](#bookmark415) 225 [- 243](#bookmark486)

امتیازدهی اعتباری، [73](#bookmark222) [- 76 ،](#bookmark226) [251 ،](#bookmark505) [257 ،](#bookmark514) [262](#bookmark17) [- 264](#bookmark523)

ریزش مشتری، [327](#bookmark31) [- 328 ،](#bookmark606) [411 ،](#bookmark737) [484](#bookmark859)

بازاریابی مستقیم، [260](#bookmark517) [- 262 ،](#bookmark17) [442](#bookmark43) [- 443](#bookmark791)

اقتصاد سوخت، [19](#bookmark107) [- 24](#bookmark115)

درخواست‌های گرنت، [275](#bookmark550) [- 282 ،](#bookmark557) [284](#bookmark558) [- 286 ،](#bookmark560) [293](#bookmark566) [- 294 ،](#bookmark567) [299 ،](#bookmark24) [303](#bookmark26) [- 306 ،](#bookmark581) [308 ،](#bookmark585) [336 ،](#bookmark625) [339](#bookmark630) [- 343 ،](#bookmark32) [349](#bookmark640) [- 350 ،](#bookmark643) [382](#bookmark692) [- 383 ،](#bookmark695) [385 ،](#bookmark699) [470](#bookmark833) [- 472](#bookmark837)

آسیب کبدی، [326](#bookmark605) [- 327 ،](#bookmark31) [366](#bookmark677) [- 367 ،](#bookmark678) [411](#bookmark737) [- 413](#bookmark739)

سطوح درآمد، [442](#bookmark43)

بیمه نامه، 419 [- 425 ،](#bookmark759) 425، [426 ،](#bookmark764) [428](#bookmark768) [- 429 ،](#bookmark771) [431](#bookmark773) [- 432 ،](#bookmark774) 433، 434

برنامه ریزی شغلی، [445](#bookmark795) [- 457](#bookmark812) انواع روغن، [327 ،](#bookmark31) [367 ،](#bookmark678) [484](#bookmark859) عوارض جانبی ناخواسته، [528](#bookmark931) [- 531](#bookmark49)

وزن مورد، [426](#bookmark764) [- 427](#bookmark767)

داده‌های سانسور شده، [41](#bookmark157)

کلاس مرکز، [49 ،](#bookmark173) [306](#bookmark581) [- 308](#bookmark585) عدم تعادل طبقاتی، 419 [- 434](#bookmark42)

احتمالات کلاس، [247](#bookmark497) [- 254](#bookmark512) قطع‌های جایگزین، [423](#bookmark757) [- 426](#bookmark764) منطقه مبهم، [254](#bookmark512) به خوبی کالیبره شده، [249](#bookmark501) [- 252 ،](#bookmark506) [296 ،](#bookmark569) [358](#bookmark655)

طبقه بندی

دقت، [254](#bookmark512)

مرزها، [62](#bookmark195)

درختان طبقه بندی، [370](#bookmark686) [- 383](#bookmark695)

نرخ کلیک، ضریب تعیین 419، 95 [- 97 ،](#bookmark270) [100](#bookmark275)

خط خطی، [98 ،](#bookmark273) [110](#bookmark290) [- 111 ،](#bookmark293) [123 ،](#bookmark312) [125 ،](#bookmark314) [127](#bookmark316)

کمیته‌ها، [210](#bookmark433)

ماتریس سردرگمی، [254 ،](#bookmark512) [456](#bookmark809) [- 457](#bookmark812)

ماتریس همبستگی، [45](#bookmark165)

نمودار همبستگی، [45](#bookmark165)

ویژگی مبتنی بر همبستگی

انتخاب، [493](#bookmark44)

آموزش حساس به هزینه [429](#bookmark771) [- 434 ،](#bookmark42) [452](#bookmark803) [- 456](#bookmark809)

ارزش طول عمر مشتری، [248](#bookmark498)

پیش پردازش داده‌ها، [27](#bookmark131) [- 49](#bookmark173)

مرکز سازی، [30](#bookmark139)

نسبت، [42](#bookmark158) [- 43](#bookmark161)

مقیاس بندی، [30](#bookmark139)

تبدیل نشانه فضایی، [34 ،](#bookmark3) [336](#bookmark625)

کاهش داده‌ها، [35](#bookmark150)

تقسیم داده‌ها، [67](#bookmark204) [- 69 ،](#bookmark210) [279](#bookmark552) [- 282 ،](#bookmark557) [450](#bookmark802)

نمونه‌گیری حداکثر عدم تشابه، [68](#bookmark205) [- 69 ،](#bookmark210) [233](#bookmark13)

مجموعه تست، [67](#bookmark204) [- 69](#bookmark210)

مجموعه آموزشی، [67](#bookmark204)

توابع مطلوبیت، [234](#bookmark479) [- 235](#bookmark480)

متغیرهای ساختگی، [47](#bookmark169) [- 48 ،](#bookmark170) [298 ،](#bookmark23) [300 ،](#bookmark25) [372](#bookmark34) [- 373 ،](#bookmark687) [400](#bookmark727)

شبکه الاستیک، [126](#bookmark315) [- 128 ،](#bookmark321) [303](#bookmark26)

آنتروپی، [333 ،](#bookmark622) [378](#bookmark35)

شیوع رویداد، [255 ،](#bookmark14) [258](#bookmark15) [- 259 ،](#bookmark516) [354](#bookmark649)

طراحی تجربی، 225 [- 227](#bookmark467) مطالعات R&R گیج، [530](#bookmark933) [- 531](#bookmark49)

طرح‌های مخلوط، [226](#bookmark12)

سطح پاسخ

روش شناسی، 225 آزمایش متوالی، 225

برون یابی، [534](#bookmark940) [- 538](#bookmark946)

نرخ مثبت کاذب، [256](#bookmark513)

مهندسی ویژگی، [27](#bookmark131) [- 28 \_](#bookmark134) [276](#bookmark19) [- 277](#bookmark551)

انتخاب ویژگی

انتخاب به عقب، [494](#bookmark45)

C5. 0 (برنده)، [398](#bookmark721) [- 399](#bookmark722)

مبتنی بر همبستگی، [493](#bookmark44) فیلترها، [490 \_](#bookmark871) [499 ،](#bookmark886) [509](#bookmark899) [- 510](#bookmark900) انتخاب رو به جلو، [491](#bookmark874) [- 493](#bookmark44)

الگوریتم‌های ژنتیک، [497](#bookmark882) [- 499](#bookmark886)

پیش‌بینی‌کننده‌های بسیار همبسته، [45](#bookmark165) [- 47 ،](#bookmark169) [277](#bookmark551)

ذاتی، [487 ،](#bookmark864) [489](#bookmark868)

ریج، [125 ،](#bookmark314) [303 ،](#bookmark26) [306](#bookmark581)

واریانس نزدیک به صفر

پیش‌بینی‌ها، [44](#bookmark162) [- 45](#bookmark165) نزدیکترین مرکزهای کوچک شده، [307](#bookmark582) [- 308](#bookmark585)

حذف ویژگی بازگشتی، [494](#bookmark45) [- 495 ،](#bookmark880) [500](#bookmark889) [- 502 ،](#bookmark46) [504](#bookmark894) [- 507](#bookmark897)

بایاس انتخاب، [500](#bookmark889) [- 501](#bookmark890)

بازپخت شبیه‌سازی شده، [495](#bookmark880) [- 497](#bookmark882)

تک درختان، [180](#bookmark396) [- 181](#bookmark397)

پراکنده و نامتعادل

پیش‌بینی‌ها، [44 ،](#bookmark162) [277](#bookmark551) [- 278](#bookmark20)

انتخاب گام به گام، [493 ،](#bookmark44) [494](#bookmark45)

تحت نظارت، [487](#bookmark864) [- 510](#bookmark900)

نظارت نشده، [43](#bookmark161) [- 47 ،](#bookmark169) [278 ،](#bookmark20) [299 ،](#bookmark24) [488](#bookmark867)

با استفاده از مریخ، [149](#bookmark352)

لفاف‌ها، [490](#bookmark871) [- 499](#bookmark886)

تست دقیق فیشر، [471 ،](#bookmark834) [472](#bookmark837)

تحلیل تفکیک انعطاف پذیر

(FDA) [306 ،](#bookmark581) [338](#bookmark629) [- 343 ،](#bookmark32) [420 ،](#bookmark40) [426 ،](#bookmark764) [453 ،](#bookmark804) [454](#bookmark807)

اهمیت متغیر، [343](#bookmark32)

پیش‌بینی تقلب، [248](#bookmark498)

مدل‌های افزودنی تعمیم یافته، [285](#bookmark559)

اعتبار سنجی متقابل تعمیم یافته

(GCV) [148](#bookmark351)

مدل‌های خطی تعمیم یافته، [284](#bookmark558)

الگوریتم ژنتیک، [497](#bookmark882) [- 499](#bookmark886)

ژنوتیپ، [503](#bookmark893) [- 504](#bookmark894)

شاخص جینی، [370](#bookmark686) [- 374 ،](#bookmark688) [433](#bookmark775)

glmnet \_ [303](#bookmark26) [- 305](#bookmark579)

هسته‌های نمودار، [349](#bookmark640)

نقشه حرارت، [251 ،](#bookmark505) [253](#bookmark507)

واحدهای پنهان، [141](#bookmark342) [- 143](#bookmark344)

محاسبات با عملکرد بالا، [445](#bookmark795)

غربالگری با محتوای بالا، [29](#bookmark1)

توابع لولا، [145 ،](#bookmark348) [338](#bookmark629) [- 339](#bookmark630)

تابع هوبر، [109 ،](#bookmark289) [151](#bookmark356)

آزمایش فرضیه، [285 ،](#bookmark559) [466](#bookmark827) [- 468 ،](#bookmark831) [471](#bookmark834) [- 472 ،](#bookmark837) [491](#bookmark874) [- 494 ،](#bookmark45) [499 ،](#bookmark886) [507](#bookmark897)

تقسیم‌بندی تصویر، [29](#bookmark1)

واردات ماشین آلات بردار، [349](#bookmark640)

به دست آوردن اطلاعات، [378 ،](#bookmark35) [379](#bookmark36)

نظریه اطلاعات، [377](#bookmark689) [- 381](#bookmark691)

تصحیحات مبتنی بر نمونه، [210](#bookmark433) [- 211](#bookmark434)

*K* -نزدیکترین همسایه‌ها، [64](#bookmark199)

طبقه بندی، [64](#bookmark199) [- 65 ،](#bookmark200) [350](#bookmark643) [- 352](#bookmark645)

نسبت، [42](#bookmark158) [- 43](#bookmark161)

پسرفت، [159](#bookmark363) [- 161](#bookmark367)

آمار کاپا، [255](#bookmark14) [- 256 ،](#bookmark513) [431](#bookmark773) [- 432 ،](#bookmark774) [434 ،](#bookmark42) [455 ،](#bookmark808) [533](#bookmark937)

توابع هسته، [155](#bookmark359) [- 157 ،](#bookmark361) [347 ،](#bookmark638) [349](#bookmark640)

تصحیح لاپلاس، [357](#bookmark652)

ریج، [124](#bookmark313) [- 126 ،](#bookmark315) [303](#bookmark26) [- 306 ،](#bookmark581) 306

رگرسیون کمترین زاویه، [126](#bookmark315) بردار پشتیبانی حداقل مربعات

ماشین آلات، [349](#bookmark640) ترک-گروه-خارج شدن

تأیید متقابل، به تقسیم‌های آموزشی/آزمون مکرر *مراجعه کنید*

نمودارهای بالابر، [265](#bookmark526) [- 266 ،](#bookmark529) [421](#bookmark41) [- 423](#bookmark757) تحلیل تشخیص خطی، 287 [- 297 ،](#bookmark572) [305](#bookmark579) [- 306 ،](#bookmark581) [453 ،](#bookmark804) [454 ،](#bookmark807) [504 ،](#bookmark894) [505 ،](#bookmark895) [508](#bookmark898) [- 509](#bookmark899)

رگرسیون خطی، [20](#bookmark108) [- 24](#bookmark115) رگرسیون وزنی محلی، [464](#bookmark825) [- 466](#bookmark827)

رگرسیون لجستیک، [79 ،](#bookmark233) [250](#bookmark502) [- 252 ،](#bookmark506) [282](#bookmark557) - 287، [420 ،](#bookmark40) [504](#bookmark894)

مجازات شد، [302](#bookmark576) [- 305](#bookmark579)

لبه، [343](#bookmark32) [- 344](#bookmark636)

حداکثر ضریب اطلاعات (MIC)، [466 ،](#bookmark827) [470 ،](#bookmark833) 476، 477

برآورد حداکثر احتمال، [282](#bookmark557) [- 284](#bookmark558)

خطای اندازه گیری، [524](#bookmark923) [- 531](#bookmark49)

تحلیل سامانه‌های اندازه گیری، [530](#bookmark933) [- 531](#bookmark49)

ارزش از دست رفته، [41](#bookmark157) [- 43 ،](#bookmark161) [277 ،](#bookmark551) [380](#bookmark690) [- 382](#bookmark692)

آموزنده، [41](#bookmark157)

تحلیل متمایز مخلوط (MDA)، [331](#bookmark618) [- 332](#bookmark619)

میانگین‌گیری مدل، [144 ،](#bookmark345) [335](#bookmark624) [- 336](#bookmark625) تنظیم پارامتر مدل، [64](#bookmark199) [- 66](#bookmark201) تأیید متقاطع مونت کارلو، به تقسیم‌های آموزشی/تست مکرر *مراجعه کنید*

طرح موزاییک، [448 ،](#bookmark798) [450](#bookmark802) چند خطی، [45](#bookmark165) [- 47](#bookmark169)

بهینه‌سازی چند پارامتری، [234](#bookmark479)

اصلاحات مقایسه چندگانه، [182](#bookmark398)

خطوط رگرسیون تطبیقی چند متغیره (MARS)، [79 ،](#bookmark233) [489](#bookmark868)

کیسه ای، [194](#bookmark413) [- 195](#bookmark414)

طبقه بندی، [338](#bookmark629) [- 339](#bookmark630)

هرس، [148](#bookmark351)

پسرفت، [22](#bookmark109) [- 24 ،](#bookmark115) [145](#bookmark348) [- 151](#bookmark356)

مدل مرتبه دوم، [148](#bookmark351)

اهمیت متغیر، [150](#bookmark353)

بیز ساده لوح، [79 ،](#bookmark233) [353](#bookmark648) [- 358 ،](#bookmark655) [505](#bookmark895)

اثر دینامیت ناپلئون، [41](#bookmark157)

پیش‌بینی‌کننده‌های واریانس نزدیک به صفر، [44](#bookmark162) [- 45 ،](#bookmark165) [277 ،](#bookmark551) [285 ،](#bookmark559) [293](#bookmark566)

نزدیکترین مرکزهای کوچک شده، [306](#bookmark581) [- 308](#bookmark585)

ارزش پیش‌بینی شده منفی، [258](#bookmark15)

*نایسریا گونوره،* [259](#bookmark516)

روش سیمپلکس نلدر مید، [233](#bookmark13)

شبکه‌های عصبی، [453 ،](#bookmark804) [456 ،](#bookmark809) [489](#bookmark868)

طبقه بندی، [333](#bookmark622) [- 336](#bookmark625)

توقف زودهنگام، [143](#bookmark344)

واحدهای پنهان، [141](#bookmark342) [- 143](#bookmark344)

میانگین‌گیری مدل، [144 ،](#bookmark345) [335](#bookmark624) [- 336](#bookmark625)

پسرفت، [141](#bookmark342) [- 145](#bookmark348)

کاهش وزن، [143](#bookmark344) [- 144 ،](#bookmark345) [303 ،](#bookmark26) [334](#bookmark623) [- 336](#bookmark625)

بدون نرخ اطلاعات، [255](#bookmark14)

پیش‌بینی‌کننده‌های غیر اطلاعاتی، [487](#bookmark864) [- 490](#bookmark871)

شانس، [283](#bookmark21) [- 285](#bookmark559)

نسبت شانس، [470](#bookmark833) [- 472](#bookmark837)

قانون خطای یک استاندارد، [75](#bookmark225)

حداقل مربعات معمولی

پسرفت، [105](#bookmark286) [- 112](#bookmark8)

موارد پرت، [33](#bookmark147) [- 34 ،](#bookmark3) [109 ،](#bookmark289) [151](#bookmark356)

بیش از حد مناسب، [62](#bookmark195) [- 64 ،](#bookmark199) [280 ،](#bookmark553) [335 ،](#bookmark624) [336 ،](#bookmark625) [347 ،](#bookmark638) [352 ،](#bookmark645) [372 ،](#bookmark34) [381 ،](#bookmark691) [490 \_](#bookmark871) [493 ،](#bookmark44) [500 ،](#bookmark889) [501 ،](#bookmark890) [503](#bookmark893)

حداقل مربعات جزئی، [79](#bookmark233) طبقه بندی، [297](#bookmark572) [- 302 ،](#bookmark576) [306 ،](#bookmark581) [453 ،](#bookmark804) [454](#bookmark807)

هسته PLS، [121](#bookmark308)

NIPALS، [120](#bookmark307)

پسرفت، [112](#bookmark8) [- 122](#bookmark311)

ساده، [121](#bookmark308)

تحمل عملکرد، [75](#bookmark225)

توزیع جایگشت، [475](#bookmark840) [- 476](#bookmark841)

ارزش پیش‌بینی شده مثبت، [258](#bookmark15)

احتمال پسین، [258 ،](#bookmark15) 287، [354](#bookmark649)

تحلیل اجزای اصلی (PCA)، [35](#bookmark150) [- 40 ،](#bookmark154) [105 ،](#bookmark286) [107 ،](#bookmark287) [113 ،](#bookmark296) [297 ،](#bookmark572) [536](#bookmark942)

رگرسیون مؤلفه اصلی، [113 ،](#bookmark296) [115](#bookmark298) [- 116 ،](#bookmark301) [118](#bookmark303)

اجزای اصلی، [35](#bookmark150)

احتمال قبلی، [255 ،](#bookmark14) [300 ،](#bookmark25) [354 ،](#bookmark649) [356 \_](#bookmark651) [426](#bookmark764)

هرس

C4. 5، [381](#bookmark691)

کوبیست، 209، 210

مریخ، [148](#bookmark351)

درختان مدل، [186](#bookmark404)

تحلیل متمایز درجه دوم (QDA)، [330](#bookmark615)

*R* 2، 95 [- 97](#bookmark270)

جنگل تصادفی، [79 ،](#bookmark233) [420 ،](#bookmark40) [428 ،](#bookmark768) [453 ،](#bookmark804) [456](#bookmark809) [- 457 ،](#bookmark812) [489 ،](#bookmark868) [504 ،](#bookmark894) [505 ،](#bookmark895) [508](#bookmark898) [- 509](#bookmark899)

همبستگی رتبه، [97 ،](#bookmark270) [100](#bookmark275)

منحنی مشخصه عملیاتی گیرنده (ROC)، [257 ،](#bookmark514) [262](#bookmark17) [- 264 ،](#bookmark523) [421](#bookmark41) [- 425 ،](#bookmark759) [468](#bookmark831) [- 470 ،](#bookmark833) [476](#bookmark841)

حذف ویژگی بازگشتی، [494](#bookmark45) [- 495 ،](#bookmark880) [500](#bookmark889) [- 502 ،](#bookmark46) [504](#bookmark894) [- 508](#bookmark898)

منظم سازی، [122](#bookmark311) [- 128 ،](#bookmark321) [143](#bookmark344) [- 144 ،](#bookmark345) [153 ،](#bookmark9) [302](#bookmark576) [- 306 ،](#bookmark581) [346](#bookmark637) [- 347](#bookmark638)

تحلیل متمایز منظم (RDA)، [330](#bookmark615) [- 331](#bookmark618)

ماشین‌های بردار مرتبط، [157](#bookmark361) [- 159 ،](#bookmark363) [349](#bookmark640)

امتیازات امدادی، [470 \_](#bookmark833) [472](#bookmark837) [- 476](#bookmark841)

نمونه‌گیری مجدد، [69](#bookmark210) [- 73](#bookmark222)

*k* -Fold Cross-Validation، [21 ،](#bookmark0) [69](#bookmark210)

بوت استرپ، [70 ،](#bookmark211) [72](#bookmark219) [- 73 ،](#bookmark222) [76 ،](#bookmark226) [78 ،](#bookmark232) [110 ،](#bookmark290) [501](#bookmark890)

روش بوت استرپ 632، [73](#bookmark222) برای مقایسه مدل‌ها، [79](#bookmark233) [- 80](#bookmark236)

ترک-یک-خارج

اعتبار سنجی متقابل، [70](#bookmark211)

تخمین خارج از کیف، [197 ،](#bookmark416) [200](#bookmark421)

دام‌ها، [500](#bookmark889) [- 501](#bookmark890)

تأیید متقابل مکرر، [70 ،](#bookmark211) [452](#bookmark803)

تقسیم‌های آموزشی/آزمایشی مکرر، [71](#bookmark216)

اعتبار سنجی متقاطع طبقه ای، [70](#bookmark211)

باقیمانده، 95، [97 ،](#bookmark270) 101، [108 ،](#bookmark288) [109 ،](#bookmark289) [112 ،](#bookmark8) [117 ،](#bookmark302) [119 ،](#bookmark304) [143 ،](#bookmark344) [151](#bookmark356) [- 153 ،](#bookmark9) [156 ،](#bookmark360) [204](#bookmark426) [- 206 ،](#bookmark428) [209 ،](#bookmark11) [230 ،](#bookmark474) [232 ،](#bookmark476) [282 ،](#bookmark557) [524](#bookmark923)

خطوط مکعبی محدود، [285](#bookmark559)

رگرسیون ریج، [123](#bookmark312) [- 124 ،](#bookmark313) [303](#bookmark26)

رگرسیون قوی، [109 ،](#bookmark289) [151](#bookmark356) [- 153](#bookmark9)

ریشه میانگین مربعات خطا (RMSE)، 95

مدل‌های مبتنی بر قانون

قوانین C4. 5 [383](#bookmark695) [- 384](#bookmark696)

C5. 0، [395](#bookmark718) [- 396](#bookmark719)

قوانین مدل طبقه بندی، [383](#bookmark695) [- 385](#bookmark699)

کوبیست، [208](#bookmark432) [- 212](#bookmark439)

قسمت، [385](#bookmark699)

قوانین مدل رگرسیون، [190](#bookmark408) [- 192 ،](#bookmark411) [211](#bookmark434) [- 212](#bookmark439)

نمونه برداری، [427](#bookmark767) [- 429](#bookmark771)

نمونه برداری پایین، [427](#bookmark767) [- 429](#bookmark771)

اقلیت مصنوعی

روش نمونه برداری بیش از حد (SMOTE)، [428](#bookmark768) [- 429](#bookmark771)

نمونه برداری بالا، [427 ،](#bookmark767) [429](#bookmark771)

بایاس انتخاب، [149 ،](#bookmark352) [299 ،](#bookmark24) [500](#bookmark889) [- 501](#bookmark890)

حساسیت، [256 ،](#bookmark513) [421 ،](#bookmark41) [423](#bookmark757) [- 425 ،](#bookmark759) [432 ،](#bookmark774) [433](#bookmark775)

انقباض، *به* منظم‌سازی مراجعه کنید

بازپخت شبیه‌سازی شده، [233 ،](#bookmark13) [495](#bookmark880) [- 497](#bookmark882)

چولگی، [31](#bookmark142) [- 33 ،](#bookmark147) [104 ،](#bookmark283) [105 ،](#bookmark286) [111 ،](#bookmark293) [458](#bookmark813)

تبدیل سافت مکس، [248 ،](#bookmark498) [300](#bookmark25)

اختصاصی، [256 ،](#bookmark513) [421 ،](#bookmark41) [423](#bookmark757)

افزایش گرادیان تصادفی، [206 ،](#bookmark428) [391](#bookmark712)

هسته‌های رشته، [349](#bookmark640)

روش‌های نظارت شده، [27 ،](#bookmark131) [115](#bookmark298)

ماشین‌های بردار پشتیبانی، [79 ،](#bookmark233) [280](#bookmark553) [- 281 ،](#bookmark554) [490 \_](#bookmark871) [500 ،](#bookmark889) [505](#bookmark895)

اوزان کلاس، [431](#bookmark773) [- 432 ،](#bookmark774) [453](#bookmark804) [- 456](#bookmark809)

طبقه بندی، [343](#bookmark32) [- 350](#bookmark643) هسته‌ها، [155](#bookmark359) [- 157 ،](#bookmark361) [347 ،](#bookmark638) [349](#bookmark640)

بیش از حد مناسب، [65](#bookmark200)

پسرفت، [151](#bookmark356) [- 157](#bookmark361)

تنظیم، [74](#bookmark6)

بردارهای پشتیبانی، 155، [155 ،](#bookmark359) [345](#bookmark33)

پاسخ التهابی سامانه‌یک

سندرم (SIRS) [49](#bookmark173)

نمودار جدول، [448 ،](#bookmark798) 451 مدل درختی

کیسه ای، [192](#bookmark411) [- 194 ،](#bookmark413) [385](#bookmark699) [- 386 ،](#bookmark702) [453 ،](#bookmark804) [456](#bookmark809)

افزایش، [203](#bookmark425) [- 208 ،](#bookmark432) 221، [230 ،](#bookmark474) [389](#bookmark37) [- 392 ،](#bookmark715) [396](#bookmark719) [- 397](#bookmark720)

C4. 5، [377](#bookmark689) [- 383](#bookmark695)

C5. 0، [394](#bookmark717) [- 395 ،](#bookmark718) 400، [432 ،](#bookmark774) [434 ،](#bookmark42) [454 ،](#bookmark807) [456](#bookmark809)

طبقه بندی، [370](#bookmark686) [- 383](#bookmark695)

درختان طبقه‌بندی و رگرسیون، [370](#bookmark686) [- 377 ،](#bookmark689) [453 ،](#bookmark804) [454 ،](#bookmark807) [456](#bookmark809) [- 457](#bookmark812)

هرس هزینه-پیچیدگی، [177](#bookmark394) [- 178 ،](#bookmark395) [372](#bookmark34)

حساس به هزینه، [432](#bookmark774) [- 434 ،](#bookmark42) [455 ،](#bookmark808) [456](#bookmark809)

تعمیم یافته، بی طرفانه، تشخیص و تخمین تعامل (راهنما)، [182](#bookmark398)

درختان مدل، [184](#bookmark402) [- 190](#bookmark408) قانون خطای یک استاندارد، [178](#bookmark395)

هرس بدبینانه، [381](#bookmark691)

جنگل تصادفی، [198](#bookmark419) [- 203 ،](#bookmark425) [386](#bookmark702) [- 389 ،](#bookmark37) [453 ،](#bookmark804) [456](#bookmark809) [- 457 ،](#bookmark812) [489 ،](#bookmark868) [504 ،](#bookmark894) [505 ،](#bookmark895) [508](#bookmark898) [- 509](#bookmark899)

پسرفت، [175](#bookmark392) [- 183](#bookmark399)

بایاس انتخاب، [182](#bookmark398) [- 183](#bookmark399)

صاف کردن، [185](#bookmark403) [- 186 ،](#bookmark404) [209](#bookmark11)

پارامترهای تنظیم، [22](#bookmark109) [- 23 ،](#bookmark110) [65](#bookmark200) خطای نوع III، [522](#bookmark48) [- 524](#bookmark923)

پاسخ نهایی به پرسش نهایی زندگی، جهان، و

همه چیز، 42

روش‌های بدون نظارت، [27 ،](#bookmark131) [115 ،](#bookmark298) [278 ،](#bookmark20) [299 ،](#bookmark24) [488](#bookmark867)

مدل‌سازی ارتقاء، [522](#bookmark48) [- 524](#bookmark923)

اهمیت متغیر، [463](#bookmark822) [- 477 ،](#bookmark842) [505](#bookmark895)

درختان کیسه ای، [198](#bookmark419)

درختان تقویت شده، [207](#bookmark429) [- 208](#bookmark432)

C5. 0، [398](#bookmark721)

کوبیست، [212](#bookmark439) [- 213](#bookmark440)

رگرسیون لجستیک، [286](#bookmark560)

مریخ، [150](#bookmark353)

حداکثر اطلاعات

ضریب (MIC) [466 ،](#bookmark827) [470 \_](#bookmark833) [476](#bookmark841) [- 477](#bookmark842)

حداقل مربعات جزئی،   
[118](#bookmark303) [- 120 ،](#bookmark307) [302](#bookmark576)

جنگل تصادفی، [201](#bookmark422) [- 203 ،](#bookmark425) [464](#bookmark825)

امتیازات امدادی، [470 \_](#bookmark833) [472](#bookmark837) [- 476](#bookmark841)

تک درختان، [180](#bookmark396) [- 182](#bookmark398)

انتخاب متغیر، به ویژگی *مراجعه کنید*

انتخاب

عوامل تورم واریانس (VIF)، [47](#bookmark169)

مبادله واریانس-بایاس، [97](#bookmark270) [- 98 \_](#bookmark273) [122](#bookmark311) [- 123 ،](#bookmark312) [192 ،](#bookmark411) [194](#bookmark413)

برنده شدن، [398](#bookmark721) [- 399](#bookmark722)

شاخص یودن، [424](#bookmark758)

پیش‌بینی واریانس صفر، [44](#bookmark162)

1. این نوع نمونه گیری شباهت زیادی به *مطالعات مورد شاهدی* در حوزه پزشکی دارد. [↑](#footnote-ref-1)
2. یکی از استادان فارغ التحصیل ما زمانی گفت: "تنها راه برای راحت بودن با داده های خود این است که هرگز به آنها نگاه نکنید." [↑](#footnote-ref-2)
3. نقاط داده فردی را می توان در وب سایت مجله یا در بسته R Applied- PredictiveModeling یافت. به بخش محاسبات در پایان این فصل مراجعه کنید. [↑](#footnote-ref-3)
4. نویسندگان اصلی چندین ویژگی «وضعیت» را شامل می‌شوند که بازنمایی باینری ­از ویژگی‌های دیگر در مجموعه داده‌ها هستند. ما اینها را از تحلیل این فصل حذف کردیم. [↑](#footnote-ref-4)
5. برخی از خوانندگان آشنا با [باکس و کاکس](#bookmark1012) [( 1964](#bookmark1012) ) خواهد دانست که این تبدیل برای داده های *نتیجه توسعه* داده شده است [باکس و تیدول](#bookmark1012) [( 1962](#bookmark1012) ) روش های مشابهی را ­برای تبدیل مجموعه ای از پیش بینی کننده ها در یک مدل خطی توصیف می کند. تجربه ما این است که تبدیل Box-Cox ساده‌تر است، کمتر مستعد مسائل عددی است و به همان اندازه برای تبدیل متغیرهای پیش‌بین فردی مؤثر است. [↑](#footnote-ref-5)
6. بخش [20.5](#bookmark940) *برون یابی* مدل را مورد بحث قرار می دهد - جایی که مدل نمونه هایی را خارج ­از جریان اصلی داده های آموزشی پیش بینی می کند. مفهوم دیگر *دامنه کاربردپذیری* مدل است که جمعیت نمونه هایی است که می تواند به طور موثر ­توسط مدل پیش بینی شود. [↑](#footnote-ref-6)
7. یک نسخه از پیش پردازش شده از این داده ها نیز در بسته caret یافت می شود و در فصل های بعدی استفاده می شود. [↑](#footnote-ref-7)
8. [http://archive.ics.uci.edu/ml/index.html .](http://archive.ics.uci.edu/ml/index.html) [↑](#footnote-ref-8)
9. این مدل از یک هسته تابع پایه شعاعی استفاده می کند که در بخش تعریف شده است. [13.4 .](#bookmark32) اگرچه نه [↑](#footnote-ref-9)
10. در اینجا بررسی شد، ما از رویکرد تحلیلی که بعداً مورد بحث قرار گرفت برای تعیین آن استفاده کردیم [↑](#footnote-ref-10)
11. پارامتر هسته و این مقدار را برای تمام تکنیک های نمونه برداری مجدد ثابت کرد. [↑](#footnote-ref-11)
12. نویسندگان بیان می کنند که 95 نمونه از روغن های شناخته شده وجود دارد. با این حال، ما در جدول 1 آنها 96 را می شماریم (ص 33-35 مقاله). [↑](#footnote-ref-12)
13. در عمل، آستانه همبستگی برای داشتن یک معنادار باید کوچکتر باشد [↑](#footnote-ref-13)
14. تاثیر بر روی هم خطی در این داده ها، متغیرهای مهم را نیز حذف می کند. همچنین، می توان بررسی کرد که چگونه این اصطلاحات در مدل قرار می گیرند. برای مثال، ممکن است بین پیش‌بینی‌کننده‌هایی که مهم هستند، تعاملاتی وجود داشته باشد و تبدیل‌های غیرخطی پیش‌بینی‌کننده‌ها نیز ممکن است مدل را بهبود بخشد. برای این داده‌ها، این مجموعه از فعالیت‌ها با دقت بیشتری در فصل بررسی شده است. [19 .](#bookmark864) [↑](#footnote-ref-14)
15. جریمه در اینجا به عنوان معکوس رگرسیون برآمدگی یا کاهش وزن در شبکه های عصبی نوشته می شود زیرا به باقیمانده ها و نه پارامترها متصل است. [↑](#footnote-ref-15)
16. همچنین، توجه داشته باشید که سه تقسیم اول در اینجا شامل همان پیش بینی کننده های درخت رگرسیون نشان داده شده در شکل 8.4 است (و دو تا از سه مقدار تقسیم یکسان هستند). [↑](#footnote-ref-16)
17. ما مدیون کار کریس کیفر هستیم که به طور گسترده کد منبع کوبیسم را مطالعه کرده است. [↑](#footnote-ref-17)
18. مواردی وجود دارد که از انواع تخصصی طرح های آزمایشی استفاده می شود [↑](#footnote-ref-18)
19. مدل های پیش بینی در زمینه شیمی سنجی، یک طراحی از نوع آرایه متعامد به دنبال ­حذف متوالی الگوریتم ترکیب سطح نشان داده شده است که مدل های QSAR را بهبود می بخشد [( ماندال و همکاران 2006](#bookmark1021) ، [2007](#bookmark1021) ). همچنین زمینه *یادگیری فعال­* [↑](#footnote-ref-19)
20. *افزودن* متوالی نمونه‌ها بر اساس مجموعه آموزشی با استفاده از نتایج مدل پیش‌بینی‌کننده [( Cohn et al. 1994](#bookmark1013) ; [Saar-Tsechansky و Provost 2007a](#bookmark1025) ). [↑](#footnote-ref-20)
21. خواننده همچنین می تواند بازپخت شبیه سازی شده را با استفاده از کد انتهای فصل امتحان کند. [↑](#footnote-ref-21)
22. در اصطلاح پزشکی، این میزان به عنوان شیوع یک بیماری نامیده می شود، در حالی که در آمار بیزی، توزیع قبلی رویداد است. [↑](#footnote-ref-22)
23. این درست است زیرا مدل‌های پیش‌بینی به دنبال یافتن یک رابطه همخوان با حقیقت هستند. یک کاپا منفی بزرگ نشان می‌دهد که بین پیش‌بینی‌کننده‌ها و پاسخ رابطه وجود دارد و مدل پیش‌بینی‌کننده به دنبال یافتن رابطه در جهت درست است. [↑](#footnote-ref-23)
24. در رابطه با آمار بیزی، حساسیت و ویژگی احتمالات مشروط، شیوع اولیه و مقادیر پیش بینی شده مثبت/منفی احتمالات پسین هستند. [↑](#footnote-ref-24)
25. این به چند فرض بستگی دارد که ممکن است درست باشد یا نباشد. بخش [20.1](#bookmark48) این جنبه از مثال را با جزئیات بیشتری در زمینه *مدل سازی بالابر خالص مورد بحث قرار می دهد* . [↑](#footnote-ref-25)
26. در این تحلیل، ما از مجموعه آزمون برای بررسی اثرات آستانه های جایگزین استفاده کرده ایم. به طور کلی، یک آستانه جدید باید از مجموعه داده های جداگانه ای نسبت به آنهایی که برای آموزش مدل یا ارزیابی عملکرد استفاده می شود، مشتق شود. [↑](#footnote-ref-26)
27. R تعدادی بسته دارد که می تواند منحنی ROC را محاسبه کند، از جمله ROCR ، caTools ، PresenceAbsence ، و غیره. [↑](#footnote-ref-27)
28. [http://blog.kaggle.com/ .](http://blog.kaggle.com/) [↑](#footnote-ref-28)
29. [http://www.kaggle.com/c/unimelb .](http://www.kaggle.com/c/unimelb) [↑](#footnote-ref-29)
30. کدهای RFCD را می توان در این آدرس یافت<http://tinyurl.com/25zvts>در حالی که کدهای سئو را می توانید در اینجا پیدا کنید [http://tinyurl.com/8435ae4 .](http://tinyurl.com/8435ae4) [↑](#footnote-ref-30)
31. Research Fields, Courses and Disciplines (RFCD) [↑](#footnote-ref-31)
32. با این حال، چندین روش مبتنی بر درخت وجود دارد که در فصل توضیح داده شده است. [14](#bookmark683) که اگر پیش‌بینی‌کننده‌های طبقه‌بندی به متغیرهای ساختگی تبدیل نشوند، مؤثرتر هستند. در این موارد از مجموعه کامل دسته بندی ها استفاده می شود. [↑](#footnote-ref-32)
33. داده‌های دارای سه یا چند کلاس معمولاً با استفاده از *توزیع چندجمله‌ای مدل‌سازی می‌شوند ­*. برای جزئیات بیشتر به Agresti ( 2002 ) مراجعه کنید. [↑](#footnote-ref-33)
34. قانون بیز با جزئیات بیشتر در بخش بررسی شده است. [13.6 .](#bookmark648) [↑](#footnote-ref-34)
35. این وضعیت دوباره برای مدل های ساده بیز در فصل بعدی مورد بررسی قرار می گیرد. [↑](#footnote-ref-35)
36. از نظر ریاضی، اگر *C -* 1 از متغیرهای ساختگی را بدانیم، مقدار آخرین متغیر ساختگی مستقیماً دلالت می شود. از این رو، فقط می توان از متغیرهای ساختگی *C - 1 استفاده کرد.* [↑](#footnote-ref-36)
37. ساختار کوواریانس آشفته به دلیل محدودیت های بهینه برای ماتریس پاسخ است. بارکر و راینز ( 2003 ) با زیرکی تشخیص دادند که محدودیت بهینه پاسخ در این تنظیمات معنی ندارد، محدودیت را حذف کردند و مشکل را حل کردند. بدون محدودیت فضای پاسخ، راه حل PLS راه حلی است که دقیقاً ماتریس کوواریانس بین گروهی را شامل می شود. [↑](#footnote-ref-37)
38. به عنوان یادآوری، مجموعه پیش بینی کننده ها بر اساس ارتباط آنها ­با نتیجه انتخاب نمی شوند. این انتخاب *بدون نظارت* نباید *سوگیری انتخاب ایجاد کند* ، که موضوعی است که در بخش توضیح داده شده است. [19.5 .](#bookmark889) [↑](#footnote-ref-38)
39. به یاد بیاورید که مدل ها بر اساس داده های قبل از سال 2008 ساخته شده اند و سپس بر اساس مجموعه نگهدارنده سال 2008 تنظیم شده اند. این پیش بینی ها از مدل PLSDA با چهار مؤلفه ایجاد شده تنها با استفاده از داده های قبل از سال 2008 است. [↑](#footnote-ref-39)
40. روش دیگری برای اضافه کردن این جریمه در فصل بعد با استفاده از شبکه های عصبی مورد بحث قرار گرفته است. در این مورد، یک شبکه عصبی با فروپاشی وزن و یک واحد پنهان، یک مدل رگرسیون لجستیک جریمه‌شده را تشکیل می‌دهد. با این حال، شبکه‌های عصبی لزوماً از احتمال دوجمله‌ای هنگام تعیین تخمین‌های پارامتر استفاده نمی‌کنند (به بخش مراجعه کنید). [13.2 )](#bookmark622) . [↑](#footnote-ref-40)
41. [http://www.kaggle.com/c/unimelb .](http://www.kaggle.com/c/unimelb) [↑](#footnote-ref-41)
42. در فصل بعد، چندین مدل مورد بحث قرار می‌گیرند که می‌توانند پیش‌بینی‌کننده‌های طبقه‌بندی را به روش‌های مختلف نشان دهند. به عنوان مثال، درختان می‌توانند از متغیرهای ساختگی در تقسیم‌بندی‌ها استفاده کنند، اما اغلب می‌توانند بر اساس یک یا چند گروه از دسته‌ها تقسیم‌بندی ایجاد کنند. در آن فصل، نسخه‌های عاملی این پیش‌بینی‌کننده‌ها به‌طور مفصل مورد بحث قرار می‌گیرند. [↑](#footnote-ref-42)
43. یکی دیگر از مشکلات نامگذاری تکراری ممکن است در اینجا رخ دهد. تابعی به نام sda در بسته sda (برای تحلیل تفکیک انقباض) ممکن است باعث سردرگمی شود. اگر هر دو بسته بارگذاری شوند، استفاده از sparseLDA:::sda و sda:::sda مشکل را کاهش می دهد. [↑](#footnote-ref-43)
44. در داده‌های ریزآرایه، تعداد پیش‌بینی‌کننده‌ها معمولاً بسیار بیشتر از تعداد نمونه‌ها است. به همین دلیل، و تعداد محدود ستون‌ها در نرم‌افزارهای صفحه‌گسترده محبوب، این قرارداد معکوس می‌شود. [↑](#footnote-ref-44)
45. [http://www.sgi.com/tech/mlc .](http://www.sgi.com/tech/mlc) [↑](#footnote-ref-45)
46. با این حال، مدل‌های MARS و FDA نسبت به مدل‌های مبتنی بر درخت پایدارتر هستند، زیرا از رگرسیون خطی برای تخمین پارامترهای مدل استفاده می‌کنند. [↑](#footnote-ref-46)
47. وضعیت مشابهی را با مدل‌های رگرسیون بردار پشتیبان به یاد بیاورید که در آن تابع پیش‌بینی توسط نمونه‌هایی با بیشترین باقیمانده تعیین شد. [↑](#footnote-ref-47)
48. دیدن [بریمن](#bookmark1012) [( 1996c](#bookmark1012) ) برای بحث در مورد تفاوت های ظریف فنی الگوریتم های تقسیم. [↑](#footnote-ref-48)
49. یک راه جایگزین برای فکر کردن به این موضوع بر حسب *آنتروپی است* ، که معیاری از عدم قطعیت است. وقتی کلاس ها 50/50 متعادل هستند، ما هیچ توانایی واقعی برای حدس زدن نتیجه نداریم: تا حد امکان نامشخص است. با این حال، اگر ده نمونه در کلاس 1 قرار داشته باشند، عدم قطعیت کمتری خواهیم داشت زیرا احتمال بیشتری وجود دارد که یک نقطه داده تصادفی در کلاس 1 باشد. [↑](#footnote-ref-49)
50. همچنین به عنوان آمار *اطلاعات متقابل شناخته می شود.* این آمار دوباره در فصل بحث شده است. [18 .](#bookmark822) [↑](#footnote-ref-50)
51. به طور پیش فرض، C4.5 از تقسیم باینری ساده پیش بینی کننده های پیوسته استفاده می کند. با این حال، [کوینلان](#bookmark1024) [( 1993b](#bookmark1024) ) همچنین تکنیکی به نام *آستانه نرم را توصیف* می کند که مقادیر نزدیک به نقطه تقسیم را به طور متفاوتی بررسی می کند. برای اختصار، در اینجا بیشتر مورد بحث قرار نمی گیرد. [↑](#footnote-ref-51)
52. از آنجایی که از یک طبقه‌بندی ضعیف استفاده می‌شود، مقادیر مرحله اغلب نزدیک به صفر هستند. [↑](#footnote-ref-52)
53. نمونه ای از این نوع استدلال در بخش نشان داده شده است. [16.9](#bookmark778) که در آن rpart با استفاده از هزینه های تفاضلی برای انواع مختلف خطاها مناسب است. [↑](#footnote-ref-53)
54. همانطور که قبلاً اشاره شد، روش‌های آماری رسمی‌تر در بیان گزاره‌های استنباطی درباره اهمیت پیش‌بینی‌کننده‌ها بسیار بهتر از معیارهای اهمیت متغیر هستند. [↑](#footnote-ref-54)
55. مایلیم از پیتر ون در پوتن برای اجازه استفاده از این داده ها تشکر کنیم. [↑](#footnote-ref-55)
56. دادن مقادیر یکسان به همه مشتریان در یک کد پستی برای پیش بینی کننده ها به معنای وجود نویز غیر قابل مشاهده در پیش بینی کننده ها است. مفاهیم نویز مجموعه پیش بینی در بخش بحث شده است. [20.3 .](#bookmark928) [↑](#footnote-ref-56)
57. نتایج فرآیند تنظیم در اینجا خلاصه شده است. جزئیات کامل فرآیند را می‌توانید در دایرکتوری Chapters بسته AppliedPredictiveModeling و در بخش Computing این فصل بیابید. [↑](#footnote-ref-57)
58. یک دیدگاه جایگزین از این معیار این است که مقدمات باید در ارزش هزینه مناسب لحاظ شوند. [↑](#footnote-ref-58)
59. این داده ها ابتدا در [کهوی ( 1996](#bookmark1020) ). شرحی از مجموعه داده‌ها ­و خلاصه‌ای از نتایج مدل‌های قبلی را می‌توانید در مخزن یادگیری ماشین UCI ( [http://archive.ics.uci.edu/ml/machine-learning- databases/adult/adult) بیابید. نام ها](http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.names) ). [↑](#footnote-ref-59)
60. ویژگی دیگر، "fnlwgt"، فقط حاوی اطلاعاتی در مورد فرآیند جمع آوری داده ها است و نباید به عنوان یک پیش بینی استفاده شود. [↑](#footnote-ref-60)
61. <http://www.dataminingconsultant.com/data>. [↑](#footnote-ref-61)
62. اکثر بورس تحصیلی در مورد این مشکل حول محور انتخاب مدل *گام به گام* ، نوعی از انتخاب رو به جلو بوده است. با این حال، نتایج برای هر روش جستجو با استفاده از آزمون فرضیه به این روش اعمال می شود. [↑](#footnote-ref-62)
63. لازم به ذکر است که آمار *AIC* برای مقایسه از پیش برنامه ریزی شده ­بین مدل ها طراحی شده است (بر خلاف مقایسه بسیاری از مدل ها در طول جستجوهای خودکار). [↑](#footnote-ref-63)
64. با این حال، [درکسن و کزلمن](#bookmark1014) [( 1992](#bookmark1014) ) و [هندرسون و ولمن](#bookmark1018) [( 1981](#bookmark1018) ) به طور کلی استدلال هایی را علیه انتخاب ویژگی خودکار ارائه می کند. [↑](#footnote-ref-64)
65. انواع مختلفی از نشانگرهای زیستی وجود دارد. به عنوان مثال، اعتقاد بر این است که سطح کلسترول خون نشان دهنده آمادگی قلبی عروقی است. به این ترتیب، می‌توان آن‌ها را تا سطح ­سلامت بیمار زیر نظر گرفت، اما برای مشخص کردن تأثیر درمان‌ها نیز استفاده می‌شود. در [↑](#footnote-ref-65)
66. <http://caret.r-forge.r-project.org/>. [↑](#footnote-ref-66)
67. در بازاریابی، [رزپاکوفسکی و یاروسزویچ ( 2012](#bookmark1025) ) کلاس‌های زیر از مدل‌های بازاریابی را تعریف کردند: «در مدل‌های تمایل، اطلاعات تاریخی در مورد خرید ­(یا سایر معیارهای موفقیت مانند بازدیدها) استفاده می‌شود، در حالی که در مدل‌های *پاسخ* ، همه مشتریان تحت یک آزمایش آزمایشی قرار گرفته‌اند. پویش." [↑](#footnote-ref-67)
68. توجه داشته باشید که در این حالت نویز در پیش بینی کننده ها *سیستماتیک* و تصادفی است. به عبارت دیگر، می‌توان منشأ تغییر را به یک علت خاص نسبت داد. [↑](#footnote-ref-68)
69. نمونه ای از این داده های زمان بندی کار است که در آن زمان اجرای یک کار به چهار گروه تقسیم می شود. در این مورد، سیستم صف نمی‌تواند از تخمین‌های طول کار استفاده کند، اما می‌تواند از نسخه‌های binned این نتیجه استفاده کند. [↑](#footnote-ref-69)