

یادگیری ماشین قابل تفسیر

راهنمای ساخت مدل‌های جعبه سیاه قابل توضیح

Second
Edition

Interpretable Machine Learning

A Guide for Making
Black Box Models Explainable



Christoph Molnar

فهرست مطالب

۹	خلاصه
۱۰	فصل ۱ پیشگفتار نویسنده
۱۳	فصل ۲ مقدمه
۱۵	۲.۱ زمان داستان
۲۲	۲.۲ یادگیری ماشین چیست؟
۲۵	۲.۳ اصطلاحات
۲۹	فصل ۳ تفسیرپذیری
۳۰	۳.۱ اهمیت تفسیرپذیری
۳۷	۳.۲ طبقه‌بندی روش‌های تفسیرپذیری
۴۰	۳.۳ حوزه تفسیرپذیری
۴۰	۳.۳.۱ شفافیت الگوریتم
۴۰	۳.۳.۲ تفسیرپذیری مدل کل نگر، کلی
۴۱	۳.۳.۳ تفسیرپذیری مدل جهانی در سطح مدولار
۴۱	۳.۳.۴ تفسیر محلی برای یک پیش‌بینی واحد
۴۲	۳.۳.۵ تفسیر محلی برای گروهی از پیش‌بینی‌ها
۴۳	۳.۴ ارزیابی تفسیرپذیری
۴۴	۳.۵ خواص توضیحات
۴۸	۳.۶ توضیحات انسان پسند
۴۸	۳.۶.۱ توضیح چیست؟
۴۹	۳.۶.۲ یک توضیح خوب چیست؟
۵۴	فصل ۴ مجموعه داده‌ها
۵۵	۴.۱ اجاره دوچرخه (رگرسیون)
۵۷	۴.۲ نظرات هرزنامه YouTube (طبقه‌بندی متن)
۵۸	۴.۳ عوامل خطر برای سرطان دهانه رحم (طبقه‌بندی)
۶۰	فصل ۵ مدل‌های قابل تفسیر
۶۲	۵.۱ رگرسیون خطی

۶۴ ۵.۱.۱ تفسیر
۶۷ ۵.۱.۲ مثال
۶۸ ۵.۱.۳ تفسیر بصری
۷۱ ۵.۱.۴ پیش‌بینی‌های فردی را توضیح دهید
۷۳ ۵.۱.۵ رمزگذاری ویژگی‌های طبقه‌بندی
۷۵ ۵.۱.۶ آیا مدل‌های خطی توضیحات خوبی ایجاد می‌کنند؟
۷۶ ۵.۱.۷ مدل‌های خطی محدود
۷۹ ۵.۱.۸ مزایا
۸۰ ۵.۱.۹ معایب
۸۱ ۵.۲ رگرسیون لجستیک
۸۱ ۵.۲.۱ استفاده از رگرسیون خطی برای طبقه‌بندی چه اشکالی دارد؟
۸۲ ۵.۲.۲ تئوری
۸۴ ۵.۲.۳ تفسیر
۸۶ ۵.۲.۴ مثال
۸۷ ۵.۲.۵ مزایا و معایب
۸۷ ۵.۲.۶ نرم افزار
۸۸ ۵.۳ GAM و GLM و موارد دیگر
۹۰ ۵.۳.۱ خروجی غیر گاوی (GLMs)
۹۶ ۵.۳.۲ تعاملات
۱۰۰ ۵.۳.۳ تاثیرات غیر خطی GAM
۱۰۵ ۵.۳.۴ مزایا
۱۰۶ ۵.۳.۵ معایب
۱۰۷ ۵.۳.۶ نرم افزار
۱۰۷ ۵.۳.۷ تعمیمات بیشتر
۱۰۹ ۵.۴ درخت تصمیم
۱۱۱ ۵.۴.۱ تفسیر
۱۱۲ ۵.۴.۲ مثال

۱۱۴	۵.۴.۳ مزایا
۱۱۵	۵.۴.۴ معایب
۱۱۶	۵.۴.۵ نرم افزار
۱۱۷	۵.۵ قوانین تصمیم
۱۱۹	۵.۵.۱ یادگیری قوانین از یک ویژگی واحد (OneR)
۱۲۳	۵.۵.۲ پوشش ترقیبی
۱۲۸	۵.۵.۳ لیستهای قانون بیزی
۱۳۴	۵.۵.۴ مزایا
۱۳۵	۵.۵.۵ معایب
۱۳۶	۵.۵.۶ نرم افزار و جایگزین ها
۱۳۷	RuleFit ۵.۶
۱۳۸	۵.۶.۱ تفسیر و مثال
۱۴۰	۵.۶.۲ تئوری
۱۴۳	۵.۶.۳ مزایا
۱۴۴	۵.۶.۴ معایب
۱۴۴	۵.۶.۵ نرم افزار و جایگزین
۱۴۶	۵.۷ سایر مدل های قابل تفسیر
۱۴۶	۵.۷.۱ طبقه بند بیز ساده
۱۴۶	۵.۷.۲ k نزدیک ترین همسایه ها
۱۴۸	فصل ۶ روش های آگنوستیک مدل
۱۵۱	فصل ۷ توضیحات مبتنی بر مثال
۱۵۳	فصل ۸ روش های آگنوستیک مدل کلی
۱۵۴	۸.۱ نمودار وابستگی جزئی (PDP)
۱۵۵	۸.۱.۱ اهمیت ویژگی مبتنی بر PDP
۱۵۶	۸.۱.۲ مثال ها
۱۵۹	۸.۱.۳ مزایا
۱۶۰	۸.۱.۴ معایب

۱۶۱	۸.۱.۵ نرم افزار و جایگزین
۱۶۲	۸.۲ نمودار اثرات محلی انباشته. (ALE)
۱۶۲	۸.۲.۱ انگیزه و شهود
۱۶۶	۸.۲.۲ تئوری
۱۶۷	۸.۲.۳ برآورد
۱۷۱	۸.۲.۴ مثال‌ها
۱۸۱	۸.۲.۵ مزايا
۱۸۲	۸.۲.۶ معایب
۱۸۴	۸.۲.۷ پیاده‌سازی و جایگزین‌ها
۱۸۵	۸.۳ تعامل ویژگی‌ها
۱۸۵	۸.۳.۱ تعامل ویژگی؟
۱۸۶	۸.۳.۲ تئوری: آماره H فریدمن
۱۸۷	۸.۳.۳ مثال‌ها
۱۹۰	۸.۳.۴ مزايا
۱۹۱	۸.۳.۵ معایب
۱۹۲	۸.۳.۶ پیاده سازی‌ها
۱۹۲	۸.۳.۷ گزینه‌های جایگزین
۱۹۳	۸.۴ تجزیه تابعی
۱۹۵	۸.۴.۱ چگونه مولفه‌ها را محاسبه نکنیم I
۱۹۶	۸.۴.۲ تجزیه تابعی
۱۹۷	۸.۴.۳ چگونه مولفه‌ها را محاسبه نکنیم II
۱۹۷	۸.۴.۴ ANOVA تابعی
۱۹۹	۸.۴.۵ ANOVA تابعی تعمیم یافته برای ویژگی‌های وابسته
۲۰۰	۸.۴.۶ نمودارهای اثر محلی انباشته
۲۰۱	۸.۴.۷ مدل‌های رگرسیون آماری
۲۰۲	۸.۴.۸ طرح وابستگی جزئی
۲۰۲	۸.۴.۹ مزايا

۲۰۳	۸.۴.۱۰ معايip
۲۰۴	۸.۵ اهميت ويزگي جايگشتی
۲۰۴	۸.۵.۱ تئوري
۲۰۵	۸.۵.۲ آيا باید اهمیت را روی داده های آموزش را محاسبه کنم یا تست؟
۲۰۸	۸.۵.۳ مثال و تفسير
۲۱۰	۸.۵.۴ مزايا
۲۱۱	۸.۵.۵ معايip
۲۱۳	۸.۵.۶ گزينه های جايگزين
۲۱۳	۸.۵.۷ نرم افزار
۲۱۴	۸.۶ جايگزين کلی
۲۱۴	۸.۶.۱ تئوري
۲۱۶	۸.۶.۲ مثال
۲۱۸	۸.۶.۳ مزايا
۲۱۸	۸.۶.۴ معايip
۲۱۹	۸.۶.۵ نرم افزار
۲۲۰	۸.۷ نمونه های اولیه و انتقادات
۲۲۲	۸.۷.۱ تئوري
۲۲۸	۸.۷.۲ مثال ها
۲۲۸	۸.۷.۳ مزايا
۲۲۹	۸.۷.۴ معايip
۲۳۰	۸.۷.۵ کد و جايگرین
۲۳۱	فصل ۹ مدل محلی-روش های آگنوستیک
۲۳۲	۹-۱-۱ انتظار شرطی فردی (ICE)
۲۳۲	۹-۱-۱-۱ مثال ها
۲۳۷	۹-۱-۱-۲ مزايا
۲۳۷	۹-۱-۱-۳ معايip
۲۳۷	۹-۱-۱-۴ نرم افزار و جايگزين

۲۳۸	جایگزین محلی (LIME) ۹.۲
۲۳۹	LIME ۹.۲.۱ برای داده‌های جدولی
۲۴۲	۹.۲.۱.۱ مثال
۲۴۳	LIME ۹.۲.۲ برای متن
۲۴۵	LIME ۹.۲.۳ برای تصاویر
۲۴۶	۹.۲.۴ مزایا
۲۴۷	۹.۲.۵ معایب
۲۴۹	۹.۳ توضیحات خلاف واقع
۲۵۲	۹.۳.۱ ایجاد توضیحات خلاف واقع
۲۵۷	۹.۳.۲ مثال
۲۵۸	۹.۳.۳ مزایا
۲۵۹	۹.۳.۴ معایب
۲۵۹	۹.۳.۵ نرم افزار و جایگزین
۲۶۱	۹.۴ قوانین محدوده (لنگرها)
۲۶۳	۹.۴.۱ یافتن لنگرها
۲۶۶	۹.۴.۲ پیچیدگی و زمان اجرا
۲۶۶	۹.۴.۳ مثال داده‌های جدولی
۲۷۰	۹.۴.۴ مزایا
۲۷۱	۹.۴.۵ معایب
۲۷۱	۹.۴.۶ نرم افزار و جایگزین
۲۷۳	۹.۵ مقدار Shapley
۲۷۳	۹.۵.۱ ایده کلی
۲۷۷	۹.۵.۲ مثال‌ها و تفسیر
۲۷۸	۹.۵.۳ جزئیات مقدار Shapley
۲۷۹	۹.۵.۳.۱ مقدار Shapley
۲۸۱	۹.۵.۳.۲ شهود
۲۸۱	۹.۵.۳.۳ برآورد ارزش Shapley

۲۸۳ مزایا ۹.۵.۴
۲۸۳ معایب ۹.۵.۵
۲۸۵ نرم افزار و جایگزین ۹.۵.۶
۲۸۶ SHAP ۹.۶
۲۸۶ تعریف ۹.۶.۱
۲۸۸ KernelSHAP ۹.۶.۲
۲۹۲ TreeSHAP ۹.۶.۳
۲۹۳ مثال‌ها ۹.۶.۴
۲۹۴ SHAP ۹.۶.۵ اهمیت ویژگی
۲۹۶ SHAP ۹.۶.۶ نمودار خلاصه
۲۹۷ SHAP ۹.۶.۷ نمودار وابستگی
۲۹۸ SHAP ۹.۶.۸ مقادیر تعامل
۲۹۹ Shapley ۹.۶.۹ خوشبندی مقادیر
۲۹۹ مزایا ۹.۶.۱۰
۳۰۰ معایب ۹.۶.۱۱
۳۰۱ نرم افزار ۹.۶.۱۲
۳۰۲	فصل ۱۱ نگاهی به آینده یادگیری ماشین قابل تفسیر
۳۰۴ آینده یادگیری ماشینی ۱۱.۱
۳۰۷ آینده تفسیرپذیری ۱۱.۲
۳۱۰	فصل ۱۲ با استناد به این کتاب
۳۱۱ ترجمه‌ها ۱۴
۳۱۲ فصل ۱۵ سپاسگزاری‌ها
۳۱۳ منابع

خلاصه

یادگیری ماشین پتانسیل زیادی برای بهبود محصولات، فرایندها و تحقیقات دارد. اما رایانه‌ها معمولاً پیش‌بینی‌های خود را توضیح نمی‌دهند که مانعی برای مقبولیت یادگیری ماشین است. این کتاب درباره تفسیر مدل‌های یادگیری ماشین و تصمیمات آن‌هاست.

پس از بررسی مفاهیم تفسیرپذیری، با مدل‌های ساده و قابل تفسیری مانند درخت تصمیم، قوانین تصمیم‌گیری و رگرسیون خطی آشنا خواهید شد. تمرکز کتاب بر روی روش‌های آگنوتیک مدل برای تفسیر مدل‌های جعبه سیاه مانند اهمیت ویژگی و اثرات محلی انباسته و توضیح پیش‌بینی‌های فردی با مقادیر Shapley و LIME است.

همه روش‌های تفسیر به طور عمیق توضیح داده شده و به صورت انتقادی مورد بحث قرار می‌گیرند. چگونه زیر روپوش کار می‌کنند؟ نقاط قوت و ضعف آنها در چیست؟ چگونه می‌توان خروجی‌های آنها را تفسیر کرد؟ این کتاب شما را قادر می‌سازد تا روش تفسیری را که برای پروژه یادگیری ماشین شما مناسب‌تر است، انتخاب و به درستی اعمال کنید. خواندن این کتاب برای یادگیران یادگیری ماشین، دانشمندان داده، آماردانان و هر کسی که علاقه‌مند به تفسیرپذیر ساختن مدل‌های یادگیری ماشین است، توصیه می‌شود.

درباره من: نام من Christoph Molnar است، من یک آماردان و یک متخصص یادگیری ماشین هستم. هدف من این است که یادگیری ماشین را قابل تفسیر کنم.

من را در توییتر دنبال کنید! @ChristophMolnar

جلد توسط @YvonneDoinel

همچنین کتاب دوم من Modeling Mindsets را مشاهده کنید.

این کتاب تحت مجوز Creative Commons Attribution-NonCommercial-ShareAlike 4.0 بین‌المللی مجوز دارد.

فصل ۱ پیشگفتار نویسنده

این کتاب زمانی که من به عنوان آمارگیر در تحقیقات بالینی کار می‌کردم به عنوان یک پروژه جانبی شروع شد. چهار روز در هفته کار می‌کردم و در روزهای بیکاری روی پروژه‌های جانبی کار می‌کردم. در نهایت، یادگیری ماشین قابل تفسیر به یکی از پروژه‌های جانبی من تبدیل شد. در ابتدا قصد نوشتمن کتاب نداشتیم. فقط علاقه‌مند به یافتن اطلاعات بیشتر در مورد یادگیری ماشین قابل تفسیر بودم و به دنبال منابع خوبی برای آموختن بودم. با توجه به موفقیت یادگیری ماشین و اهمیت تفسیرپذیری، من انتظار داشتم که تعداد زیادی کتاب و آموزش در مورد این موضوع وجود داشته باشد. اما من فقط چند مقاله تحقیقاتی و چند پست وبلاگ پراکنده در سراسر اینترنت را پیدا کردم و هیچ منبع جامعی پیدا نکردم. نه کتاب، نه آموزش، نه مقاله مروری، نه هیچ چیز دیگری. این خلاً باعث شد من شروع به نوشتمن این کتاب کنم. در نهایت شروع به نوشتمن کتابی کردم که آرزو داشتم زمانی که مطالعه خود را در مورد یادگیری ماشین قابل تفسیر شروع کردم، وجود داشته باشد. قصد من از این کتاب دو چیز بود: برای خودم یاد بگیرم و این دانش جدید را با دیگران به اشتراک بگذارم.

من مدرک لیسانس و فوق لیسانس خود را در رشته آمار در LMU مونیخ آلمان دریافت کردم. بیشتر دانش من در مورد یادگیری ماشین به صورت خودآموز و شرکت در دوره‌های آنلاین، مسابقات، پروژه‌های جانبی و فعالیت‌های حرفه‌ای است. پیشینه آماری من مهارت بسیار خوبی برای ورود به یادگیری ماشین و بهویژه برای تفسیرپذیری بود. در آمار، تمرکز عمدی بر ساخت مدل‌های رگرسیون قابل تفسیر است. بعد از اینکه فوق لیسانس آمار را تمام کردم تصمیم گرفتم به مقطع دکتری نرم‌ware، چون از نوشتمن پایان‌نامه فوق لیسانس لذت نبردم. نوشتمن خیلی به من استرس وارد می‌کرد؛ بنابراین به عنوان دانشمند داده در استارت‌آپ Fintech و به عنوان آماردان در تحقیقات بالینی مشغول به کار شدم. بعد از سه سال کار در صنعت، نوشتمن این کتاب را شروع کردم و چند ماه بعد، دکتراً خود را در زمینه یادگیری ماشین تفسیرپذیر شروع کردم. در حین کار بر روی این کتاب، لذت نوشتمن را دوباره کشف کردم و به من کمک کرد تا اشتیاقم به تحقیق را زیادتر کنم.

این کتاب بسیاری از تکنیک‌های یادگیری ماشین قابل تفسیر را پوشش می‌دهد. در فصل اول، مفهوم تفسیرپذیری را معرفی می‌کنم و انگیزه لازم را برای تفسیرپذیری بیان می‌کنم. چند داستان کوتاه برای درک بهتر این موضوع آورده شده است! این کتاب در مورد ویژگی‌های مختلف توضیحات و آنچه که انسان فکر می‌کند توضیح خوبی است، بحث می‌کند. سپس مدل‌های یادگیری ماشین که ذاتاً قابل تفسیر هستند، مانند مدل‌های رگرسیون و درخت‌های تصمیم موردنبحث قرار می‌دهیم. تمرکز اصلی این کتاب بر روی روش‌های تفسیرپذیری آگنوستیک مدل است. آگنوستیک مدل به این معنی است که این روش‌ها را می‌توان برای هر مدل یادگیری ماشین اعمال کرد و پس از آموزش مدل اعمال می‌شود. این استقلال از مدل، روش‌های آگنوستیک مدل را بسیار انعطاف‌پذیر و قدرتمند می‌کند. برخی از تکنیک‌ها چگونگی پیش‌بینی‌های فردی را توضیح می‌دهند، مانند

توضیحات - آگنوستیک مدل محلی قابل تفسیر (LIME) و مقادیر Shapley. سایر تکنیک‌ها میانگین رفتار مدل را در یک مجموعه‌داده توصیف می‌کنند. در اینجا با نمودار وابستگی جزئی، اثرات محلی انباشته، اهمیت ویژگی جای‌گشت و بسیاری از روش‌های دیگر آشنا می‌شویم. یک دسته خاص، روش‌های مبتنی بر مثال است که نقاط داده را به عنوان توضیحات تولید می‌کند. توضیحات خلاف واقع، نمونه‌های اولیه، نمونه‌های تأثیرگذار و مثال‌های مختصّ روش‌های مبتنی بر مثال هستند که در این کتاب مورد بحث قرار گرفته‌اند. این کتاب با برخی تأملات در مورد آینده یادگیری ماشین قابل تفسیر به پایان می‌رسد.

شما مجبور نیستید کتاب را از ابتدا تا انتهای بخوانید، می‌توانید به جلو و عقب بروید و روی تکنیک‌هایی تمرکز کنید که بیشتر مورد علاقه شما هستند. من فقط توصیه می‌کنم که از مقدمه و فصل تفسیرپذیری شروع کنید. اکثر بخش‌ها از ساختار مشابهی پیروی می‌کنند و بر یک روش تفسیری تمرکز می‌کنند. پاراگراف اول روش را خلاصه می‌کند. سپس سعی می‌کنم بدون اتكا به فرمول‌های ریاضی، روش را به صورت شهودی توضیح دهم. سپس به تئوری روش می‌پردازم تا درک عمیقی از نحوه عملکرد آن به دست آوریم. این قسمت حاوی فرمول‌هایی خواهد بود. من معتقدم که یک روش جدید با استفاده از مثال‌ها به بهترین وجه قابل درک است. بنابراین، هر روش برای داده‌های واقعی اعمال می‌شود. برخی افراد می‌گویند که آماردانان افراد بسیار منتقدی هستند. این موضوع برای من صدق می‌کند، زیرا هر فصل شامل بحث‌های انتقادی در مورد مزايا و معایب روش تفسیر مربوطه است. این کتاب تبلیغی برای روش‌ها نیست، اما باید به شما کمک کند تصمیم بگیرید که آیا این روش برای تحقیق شما خوب است یا خیر. در بخش آخر هر بخش، نرم‌افزارهای پیاده‌سازی موجود آورده شده است.

یادگیری ماشین موردنوجه بسیاری از افراد در تحقیقات و صنعت قرار گرفته است. گاهی اوقات یادگیری ماشین بیش از حد در رسانه‌ها مطرح می‌شود، در حالیه کاربردی واقعی و تأثیرگذار معینی وجود دارد. یادگیری ماشین یک فناوری قدرتمند برای محصولات، تحقیقات و اتوماسیون است. به عنوان مثال، امروزه از یادگیری ماشین در موارد زیر استفاده می‌شود: برای شناسایی تراکنش‌های مالی تقلیبی، توصیه فیلم‌ها و طبقه‌بندی تصاویر. اغلب مهم است که مدل‌های یادگیری ماشین قابل تفسیر باشند. تفسیرپذیری به توسعه‌دهندگان در رفع اشکال و بهبودها کمک می‌کند، اعتماد به مدل ایجاد می‌کند، پیش‌بینی‌های مدل را توجیه می‌کند و به بینش‌های جدید منجر می‌شود. افزایش نیاز به تفسیرپذیری یادگیری ماشین نتیجه طبیعی افزایش استفاده از یادگیری ماشین است. این کتاب منبعی ارزشمند برای بسیاری از افراد می‌تواند باشد. مربیان آموزشی می‌توانند از این کتاب برای معرفی دانشجویان خود با مفاهیم یادگیری ماشین قابل تفسیر استفاده می‌کنند. من از چندین دانشجوی کارشناسی ارشد کتری ایمیل دریافت کرده‌ام. دانشجویانی که به من گفتند این کتاب نقطه شروع و مهم‌ترین مرجع پایان نامه‌های آنها بوده است. این کتاب به محققان کاربردی در زمینه‌های بوم‌شناسی، مالی، روان‌شناسی

و غیره که از یادگیری ماشین برای درک داده‌های خود استفاده می‌کنند کمک کرده است. دانشمندان داده که در صنعت کار می‌کنند به من گفتند که از کتاب "یادگیری ماشین قابل تفسیر" برای کار خود استفاده می‌کنند و آن را به همکاران خود توصیه می‌کنند. خوشحالم که افراد زیادی از این کتاب بهره برند و در تفسیر مدل متخصص شدند.

من این کتاب را به علاقمندانی توصیه می‌کنم که می‌خواهند مروری بر تکنیک‌های تفسیرپذیر تر کردن مدل‌های یادگیری ماشین خود داشته باشند. همچنین برای دانشجویان و محققین (و هر کس دیگری) که به موضوع علاوه‌مند است، مفید خواهد بود. برای استفاده حداکثری از این کتاب، باید درک اولیه‌ای از یادگیری ماشین داشته باشید. همچنین باید درک درستی از ریاضیات پایه دانشگاهی داشته باشید تا بتوانید تئوری و فرمول‌های این کتاب را دنبال کنید. با این حال، درک توصیف شهودی روش در ابتدای هر فصل بدون ریاضیات نیز باید امکان‌پذیر باشد.

امیدوارم از کتاب لذت ببرید!

فصل ۲ مقدمه

این کتاب به شما توضیح می‌دهد که چگونه می‌توانید مدل‌های یادگیری ماشین (با نظارت) را قابل تفسیر کنید. بخش‌ها حاوی برخی فرمول‌های ریاضی هستند، اما شما باید بتوانید ایده‌های پشت روش‌ها را حتی بدون فرمول‌ها درک کنید. این کتاب برای افرادی نیست که سعی می‌کنند یادگیری ماشین را از ابتدا یاد بگیرند. اگر در یادگیری ماشین تازه‌کار هستید، کتاب‌ها و منابع دیگری برای یادگیری اصول اولیه وجود دارد. من کتاب «عناصر یادگیری آماری» اثر Hastie et al. (۲۰۰۹) و دوره آنلاین «یادگیری ماشین» Andrew Ng در پلتفرم یادگیری آنلاین^۱ coursera.com را برای شروع با یادگیری ماشین توصیه می‌کنم. هم کتاب و هم دوره رایگان در دسترس هستند!

روش‌های جدید برای تفسیر مدل‌های یادگیری ماشین با سرعتی سراسام‌آور منتشر می‌شوند. همگام‌شدن با هر آنچه منتشر می‌شود غیرممکن است. به همین دلیل است که در این کتاب جدیدترین و فانتزی‌ترین روش‌ها را پیدا نمی‌کنید، بلکه روش‌های تثبیت شده و مفاهیم اساسی تفسیرپذیری یادگیری ماشین را پیدا خواهید کرد. این اصول شما را برای ساختن مدل‌های یادگیری ماشین قابل تفسیر آماده می‌کند. درک مفاهیم اساسی به شما این امکان را می‌دهد که هر مقاله جدیدی در مورد تفسیرپذیری منتشر شده در arxiv.org در ۵ دقیقه گذشته از زمان شروع خواندن این کتاب را بهتر درک و ارزیابی کنید (ممکن است در میزان انتشار اغراق کنم).

این کتاب با چند داستان کوتاه (کابوس وار) شروع می‌شود که برای درک کتاب موردنیاز نیست، اما امیدوارم شما را سرگرم کند و به فکر فرببرد. سپس این کتاب مفاهیم تفسیرپذیری یادگیری ماشین را بررسی می‌کند. ما در مورد اینکه تفسیرپذیری مهم است و انواع مختلف توضیحاتی که وجود دارد بحث خواهیم کرد. اصطلاحات استفاده شده در سراسر کتاب را می‌توان در بخش اصطلاحات جستجو کرد. بیشتر مدل‌ها و روش‌های توضیح داده شده، با استفاده از نمونه‌های داده واقعی ارائه شده‌اند که در فصل داده‌ها شرح داده شده است. یکی از راه‌های قابل تفسیر کردن یادگیری ماشین، استفاده از مدل‌های قابل تفسیر، مانند مدل‌های خطی یا درخت‌های تصمیم‌گیری است. گزینه دیگر استفاده از ابزارهای تفسیر آگنوستیک مدل که می‌توانند برای هر مدل یادگیری ماشین نظارت شده‌ای اعمال شوند. روش‌های آگنوستیک مدل را می‌توان به روش‌های کلی که رفتار میانگین مدل را توصیف می‌کنند و روش‌های محلی که پیش‌بینی‌های فردی را توضیح می‌دهند، تقسیم کرد. فصل روش‌های آگنوستیک مدل به روش‌هایی مانند نمودارهای وابستگی جزئی^۲ و اهمیت ویژگی^۳ می‌پردازد. روش‌های آگنوستیک مدل با تغییر ورودی مدل یادگیری ماشین و اندازه‌گیری تغییرات در خروجی کار می‌کنند. این کتاب با یک چشم‌انداز خوش‌بینانه در مورد آینده یادگیری ماشین قابل تفسیر به پایان می‌رسد.

¹ <https://www.coursera.org/learn/machine-learning>

² Partial dependence plots

³ Feature importance

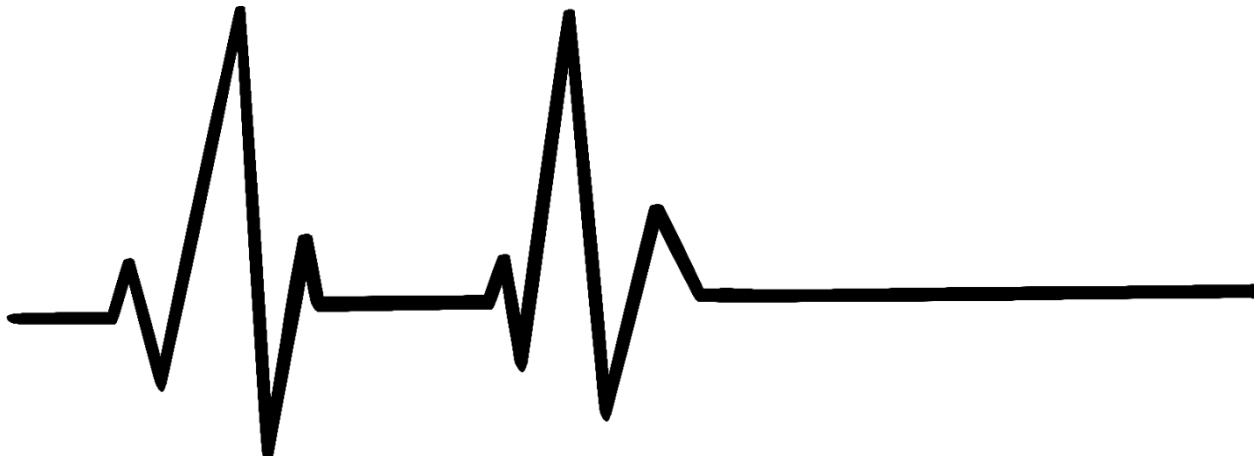
می‌توانید کتاب را از ابتدا تا انتهای بخوانید یا مستقیماً به روش‌های مورد علاقه خود بروید.
امیدوارم از خواندن لذت ببرید!

۲.۱ زمان داستان

با چند داستان کوتاه شروع می‌کنیم. هر داستان یک فراخوان اغراق‌آمیز برای یادگیری ماشین قابل تفسیر است. اگر عجله دارید، می‌توانید از داستان‌ها صرف‌نظر کنید. اگر می‌خواهید سرگرم شوید و انگیزه ندارید، ادامه مطلب را بخوانید!

این قالب از داستان‌های فناوری Import AI Newsletter Jack Clark در خبرنامه او الهام گرفته شده است. اگر این نوع داستان‌ها را دوست دارید یا اگر به هوش مصنوعی علاقه‌مند هستید، توصیه می‌کنم در این خبرنامه ثبت‌نام کنید.

رعدوبرق هرگز دو بار نمی‌زند ۲۰۳۰ : یک آزمایشگاه پزشکی در سوئیس



تام گفت: "قطعاً این بدترین راه برای مردن نیست!" و سعی کرد چیز مثبتی در تراژدی پیدا کند. او پمپ را از قطب داخل وریدی خارج کرد.

لنا افزود: "او فقط به دلایل اشتباه مرد."

تام در حالی که پیچ پشتی پمپ را باز می‌کرد، شکوه کنان گفت: "و مطمئناً با پمپ مرفین خراب! فقط کار برای ما زیاد کرد!". بعد از برداشتن تمام پیچ‌ها، صفحه را بلند کرد و کنار گذاشت. او یک کابل را به درگاه تشخیص وصل کرد.

لنا لبخند تمخرآمیزی به او زد و گفت: "شما فقط از داشتن شغل شکایت نکردید، نه؟"

تام با لحنی کنایه‌آمیز داد زد: "البته که نه. هرگزا" و کامپیوتر پمپ را بوت کرد.

لنا سر دیگر کابل را به تبلتش وصل کرد و گفت: "بسیار خوب، تشخیص در حال انجام است. من واقعاً کنجکاو هستم که ببینم چه اشتباهی رخداده است".

تام گفت: "مطمئناً از طرف ناشناسی به مریض شلیک شده است. غلظت بالای این مواد مورفین. این اولین بار است، درسته؟ معمولاً یک پمپ شکسته اصلاً موادی نمی‌دهد یا مقدار بسیار کمی را تولید می‌کند. اما هرگز، این مقدار زیاد تولید نمی‌کند".

لنا تبلتش را بالا آورد و گفت: "این را نگاه کن. این پیک را اینجا می‌بینی؟ این قدرت ترکیب مسکن‌ها است. نگاه کن این خط سطح مرجع را نشان می‌دهد. بیچاره مخلوطی از مسکن در سیستم خونش داشت که می‌توانست ۱۷ بار او را بکشد. توسط پمپ ما در اینجا تزریق می‌شود. و در اینجا..." او با انگشت خود تند تند گفت: "در اینجا می‌توانید لحظه مرگ بیمار را ببینید".

تام از سرپرستش پرسید: "پس، آیا می‌دانید چه اتفاقی افتاده است، رئیس؟"

لنا گفت: "حسگرها به نظر سالم هستند. ضربان قلب، سطح اکسیژن، گلوکز، و ... داده‌ها همان‌طور که انتظار می‌رفت جمع‌آوری شد. برخی از مقادیر ازدست‌رفته در داده‌های اکسیژن خون وجود دارند، اما این غیرعادی نیست. اینجا را نگاه کن. سنسورها همچنین کاهش ضربان قلب بیمار و سطوح بسیار پایین کورتیزول ناشی از مشتقات مورفین و سایر عوامل مسدود‌کننده درد را تشخیص داده‌اند." او همچنان بهمراه گزارش تشخیصی ادامه داد.

تام مجذوب صفحه‌نمایش شده بود. این اولین تحقیق او در مورد خرابی واقعی دستگاه بود.

لنا به تام گفت: "خوب، اینجا اولین قطعه از پازل ماست. این سیستم در ارسال اخطار به کانال ارتباطی بیمارستان ناموفق بود. هشدار ایجاد شد، اما در سطح پروتکل رد شد. ممکن است تقصیر ما باشد، اما ممکن است تقصیر بیمارستان نیز باشد".

تام در حالی که چشمانت همچنان به صفحه‌نمایش خیره شده بود سر تکان داد.

لنا ادامه داد: "عجیب است. این هشدار همچنین باید باعث خاموش شدن پمپ شود. اما مشخص است که موفق به انجام این کار نشده است. این باید یک اشکال باشد. چیزی که تیم با کیفیت از دست داد. یه چیز واقعاً بد. شاید به مشکل پروتکل مربوط باشد".

تام با تعجب گفت: "بنابراین، سیستم اورژانسی پمپ به نوعی خراب شد، اما چرا پمپ پر شد و این همه مسکن به جان دو تزریق کرد؟"

لنا توضیح داد: "سؤال خوبی بود. حق با شمامست. جدا از خرابی اضطراری پروتکل، پمپ اصلاً نباید آن مقدار دارو را تجویز می‌کرد. با توجه به سطح پایین کورتیزول و سایر علائم هشدار، الگوریتم باید خیلی زودتر به خودی خود متوقف می‌شد".

تام پرسید: "شاید یک بدشานسی، مانند یک در میلیون، مانند برخورد با رعدوبرق؟"

لنا توضیح داد: "نه، تام. اگر اسنادی را که برایتان فرستادم خوانده بودید، می‌دانستید که پمپ ابتدا در آزمایش‌های حیوانی و سپس روی انسان‌ها آموزش داده شد تا بر اساس ورودی حسی، مقدار مناسبی از مسکن‌ها را تزریق کند. الگوریتم پمپ ممکن است مبهم و پیچیده باشد، اما تصادفی نیست. این بدان معناست که در شرایط مشابه، پمپ دوباره دقیقاً به همان روش عمل می‌کند. بیمار ما دوباره می‌میرد. ترکیبی یا اثر متقابل نامطلوب ورودی‌های حسی باید باعث رفتار اشتباه پمپ شده باشد. به همین دلیل است که ما باید عمیق‌تر بگردیم و بفهمیم اینجا چه اتفاقی افتاده است."

تام که در فکر فرو رفته بود پاسخ داد: "می‌بینم. آیا به هر حال بیمار به زودی نخواهد مرد؟ به خاطر سرطان یا مريضى مشابه آن؟"

لنا درحالی که گزارش تحلیل را می‌خواند سر تکان داد.

تام بلند شد و به سمت پنجره رفت. به بیرون نگاه کرد، چشمانش به نقطه‌ای در دوردست دوخته شد. "شاید دستگاه به او لطفی کرده است که او را از درد رهایی بخشد. دیگر رنجی نیست. شاید کار درست را انجام داده است. مثل یک رعدوبرق، اما، می‌دانید، یک رعدوبرق خوب. منظورم مثل قرعه کشی است، اما نه تصادفی. اما به دلیلی دیگر، اگر من جای پمپ بودم، همین کار را می‌کردم".

بالاخره لنا سرش را بلند کرد و به او نگاه کرد.

تام مدام به چیزی بیرون نگاه می‌کرد.

هر دو برای چند لحظه سکوت کردند.

لنا دوباره سرش را پایین انداخت و به تحلیل ادامه داد. "نه، تام. این یک اشکال است... فقط یک باگ لعنتی".

به افتادن اعتماد کن

۲۰۵۰: یک ایستگاه مترو در سنگاپور



با عجله به سمت ایستگاه متروی بیشان رفت. با افکارش از قبل سر کار بود. آزمایشات برای معماری عصبی جدید باید تا الان کامل شده باشد. او بازطراحی «سیستم پیش‌بینی وابستگی مالیاتی برای اشخاص حقیقی» را مدیریت می‌کرد که پیش‌بینی می‌کند آیا شخص پول را از اداره مالیات پنهان می‌کند یا خیر. تیم او یک قطعه مهندسی طریف را ارائه کرده است. در صورت موفقیت، این سیستم نه تنها به اداره مالیات خدمت می‌کند، بلکه به سیستم‌های دیگر مانند سیستم هشدار ضد تروریسم و ثبت تجاری نیز وارد می‌شود. یک روز، دولت حتی می‌تواند پیش‌بینی‌ها را در امتیاز اعتماد مدنی ادغام کند. امتیاز اعتماد مدنی تخمین می‌زند که یک فرد چقدر قابل اعتماد است. این تخمین بر هر بخش از زندگی روزمره شما تأثیر می‌گذارد، مانند دریافت وام یا مدت زمانی که باید برای پاسپورت جدید صبر کنید. وقتی از پله برقی پایین می‌آمد، او تصور کرد که ادغام سیستم تیمش، در سیستم امتیاز اعتماد مدنی چگونه خواهد بود.

او به طور معمول دست خود را روی دستگاه RFID خوان بدون کاهش سرعت راه رفتنش کشید. ذهن او درگیر بود، اما ناهمانگی انتظارات حسی و واقعیت زنگ خطر را در مغزش به صدا درآورد. خیلی دیر.

دماغ ابتدا وارد دروازه ورودی مترو شد و با پشت به زمین افتاد. قرار بود در باید باز می‌شد، اما باز نشد. مات و مبهوت از جایش بلند شد و به صفحه‌نمایش کنار ورودی نگاه کرد. یک شکلک دوستانه روی صفحه پیشنهاد کرد: «لطفاً یک بار دیگر امتحان کنید». شخصی از آنجا گذشت و بی توجه به او دستش را از روی صفحه گذراند. در باز شد و او رفت. در دوباره بسته شد. بینی اش را پاک کرد. درد داشت ولی حداقل خونریزی نداشت. سعی کرد در را باز کند، اما دوباره در باز نشد. عجیب بود. شاید حساب حمل و نقل عمومی او توکن کافی نداشته باشد. او برای بررسی موجودی حساب به ساعت هوشمند خود نگاه کرد.

ساعتش به او اعلام کرد: "ورود رد شد. لطفاً با دفتر مشاوره شهروندان خود تماس بگیرید!".

احساس تهوع مثل مشت به شکمش خورد. او مشکوک بود که چه اتفاقی افتاده است. برای تایید نظریه خود، او بازی موبایل "Sniper Guild" را شروع کرد که یک مسابقه تیراندازی بود. برنامه به طور خودکار بسته شد، که نظریه او را تایید کرد. گیج شد و دوباره روی زمین نشست.

تنها یک توضیح ممکن وجود داشت: امتیاز اعتماد مدنی او بطور قابل ملاحظه‌ای کاهش یافته بود. یک افت کوچک به معنای محرومیت‌های جزئی بود، مانند عدم دریافت پروازهای درجه یک یا نیاز به کمی بیشتر صبر کردن برای استناد رسمی. نمره اعتماد پایین نادر بود و به این معنی بود که شما به عنوان یک تهدید برای جامعه طبقه‌بندی می‌شوید. یکی از اقدامات در برخورد با این افراد دور نگه داشتن آنها از مکان‌های عمومی مانند مترو بود. دولت تراکنش‌های مالی افراد دارای امتیاز اعتماد مدنی پایین را محدود کرد. آنها همچنین شروع به نظارت فعالانه بر رفتار شما در رسانه‌های اجتماعی کردند و حتی تا آنجا پیش رفتند که محتوای خاصی مانند بازی‌های

خشونت آمیز را محدود کردند. افزایش امتیاز اعتماد مدنی هر چه کمتر بود به طور تصاعدی دشوارتر می شد. امتیاز افراد با نمره بسیار پایین معمولاً هرگز بهبود نمی یابند.

او نمی توانست به هیچ دلیلی فکر کند که چرا نمره او باید پایین می آمد. امتیاز بر اساس یادگیری ماشین بود. سیستم امتیاز اعتماد مدنی مانند موتور روغن کاری شدهای عمل می کرد که جامعه را اداره می کرد. عملکرد سیستم امتیاز اعتماد همیشه به دقت نظارت می شد. یادگیری ماشین از ابتدای قرن بسیار بهتر شده بود. آنقدر کارآمد شده بود که تصمیمات اتخاذ شده توسط سیستم امتیاز اعتماد دیگر قابل بحث نبود. یک نظام خطاپذیر. او با نامیدی خنده دید. نظام معصوم. این سیستم به ندرت شکست خورده است. اما شکست خورد. او باید یکی از آن موارد خاص باشد. خطای سیستم؛ از این به بعد یک طرد شده. هیچ کس جرات نداشت سیستم را زیر سوال ببرد. آنقدر در دولت، در خود جامعه ادغام شده بود که نمی توان آن را زیر سوال برد. در محدود کشورهای دموکراتیک باقیمانده، تشکیل جنبش‌های ضد دموکراتیک ممنوع بود، نه به این دلیل که ذاتاً بدخواهانه بودند، بلکه به این دلیل که سیستم فعلی را بی ثبات می کردند. همین منطق در مورد الگوکراسی‌های رایج‌تر هم اعمال می شود. نقد در الگوریتم‌ها به دلیل خطر برای وضع موجود ممنوع بود.

اعتماد الگوریتمی تار و پود نظم اجتماعی بود. برای منافع عمومی، اشتباهات نادر به طور ضمنی پذیرفته شد. صدھا سیستم پیش‌بینی و پایگاه داده دیگر به امتیاز وارد شده‌اند و نمی توان مشخص کرد چه چیزی باعث افت امتیاز او شده است. او احساس می کرد که یک سوراخ تاریک بزرگ در ذهن او باز شده است. با وحشت به فضای خالی نگاه کرد.

سیستم وابستگی مالیاتی او در نهایت در سیستم امتیاز اعتماد مدنی ادغام شد، اما او هرگز با آن را شناخت.

گیره‌های فرمی

۶۱۲ سال پس از استقرار مریخ: موزه‌ای در مریخ



زولا با دوستش زمزمه کرد: «تاریخ کسل کننده است». زولا، دختری با موهای آبی، با تنبلی یکی از پهپادهای پروژکتوری را که در اتاق زمزمه می‌کرد، با دست چپش تعقیب می‌کرد. معلم با صدایی ناراحت گفت: "تاریخ مهم است." زولا سرخ شد. او انتظار نداشت معلمش او را بشنود.

معلم از او پرسید: "زولا، چه چیزی یاد گرفتی؟" او با دقت گفت: "اینکه مردم باستان تمام منابع سیاره زمین را مصرف کردند و سپس مردند؟". دختر دیگری به نام لین افزواد: "نه. کسانی که آب و هوا را گرم کردند، مردم نبودند. کامپیوتر و ماشین بودند. و این سیاره زمین است، نه سیاره زمین". زولا سری به تایید تکان داد. معلم با احساس غرور لبخندی زد و سری تکان داد. «حق با هر دوی شماست. میدونی چرا اینطوری شد؟" زولا پرسید: "چون مردم کوته فکر و حریص بودند؟" لین گفت: "مردم نمی‌توانند ماشین‌های خود را متوقف کنند!".

معلم گفت: «باز هم، هر دوی شما درست می‌گویید، اما موضوع بسیار پیچیده‌تر از این است. بیشتر مردم در آن زمان از آنچه در حال رخ دادن بود آگاه نبودند. برخی تغییرات شدید را دیدند، اما نتوانستند آنها را معکوس کنند. مشهورترین قطعه از این دوره شعری از نویسنده ناشناس است که به بهترین شکل آنچه را که در آن زمان اتفاق افتاد به تصویر می‌کشد. با دقت گوش کنید!»

معلم شعر را شروع کرد. تعداد زیادی از پهپادهای کوچک در مقابل کودکان قرار گرفتند و شروع به پخش ویدیویی در چشمان آنها کردند. ویدیو، فردی را با کت و شلوار نشان می‌داد که در جنگلی ایستاده بود که در آن فقط کنده‌های درخت باقی مانده بود. شروع کرد به صحبت کردن:

ماشین‌ها محاسبه می‌کنند.
ماشین‌ها پیش‌بینی می‌کنند
جوری که بخشی از آن هستیم آن‌ها را دنبال می‌کنیم.
ما به عنوان یک آموزش‌دهنده به دنبال یک بهینه هستیم.
بهینه یک بعدی، محلی و بدون محدودیت است.

سیلیکون و گوشت، در تعقیب تصاعدی.

رشد ذهنیت ماست.

وقتی همه جوایز جمع‌آوری شد،
و عوارض جانبی نادیده گرفته شد.

وقتی تمام سکه‌ها استخراج می‌شوند،
و طبیعت عقب افتاده است.

دچار مشکل خواهیم شد،
به هر حال، رشد تصاعدی یک حباب است.
تراژدی عوام آشکار می‌شود،

منفجر شدن،
جلوی چشمان ما
محاسبات سرد و طمع سرد،
زمین را از گرما پر کنید.
همه چیز در حال مرگ است،
و ما رعایت می کنیم.

ما مانند اسبهایی با چشم بند در مسابقه خلقت خود مسابقه می دهیم،
به سوی فیلتر بزرگ تمدن.

و بنابراین ما بی امان تعقیب می کنیم.
همان طور که ما بخشی از ماشین هستیم.
آنتروپی را در بر می گیرد.

علم برای شکستن سکوت اتاق گفت: "یک خاطره تاریک. در کتابخانه شما آپلود خواهد شد. تکلیف شما این است که آن را تا هفته آینده آن را حفظ کنید." زولا آهی کشید. او موفق شد یکی از پهپادهای کوچک را بگیرد. پهپاد از CPU و موتورها گرم بود. زولا دوست داشت چون دستان او را گرم می کرد.

۲.۲ یادگیری ماشین چیست؟

یادگیری ماشین مجموعه‌ای از روش‌هایی است که رایانه‌ها برای انجام و بهبود پیش‌بینی‌ها یا رفتارها، بر اساس داده‌ها، از آن‌ها استفاده می‌کنند.

برای مثال، برای پیش‌بینی ارزش یک خانه، کامپیوتر، الگوهایی از فروش خانه‌های قبلی، یاد می‌گیرد. این کتاب بر یادگیری ماشین نظارت شده^۱ تمرکز دارد، که همه مسائل پیش‌بینی را پوشش می‌دهد. در این نوع، مجموعه داده‌ای داریم که در آن نتیجه^۲ مورد نظر را می‌دانیم (مثلاً قیمت‌های قبلی خانه) و می‌خواهیم یاد بگیریم که چگونه نتیجه را برای داده‌های جدید پیش‌بینی کنیم. وظایف^۳ خوبه‌بندی^۴ (= یادگیری بدون نظارت^۵) که در آن ما یک نتیجه خاص نداریم، اما می‌خواهیم خوبه‌هایی از نقاط داده را پیدا کنیم، جزء یادگیری تحت نظارت قرار نمی‌گیرد. همچنین مواردی مانند یادگیری تقویتی^۶، که در آن یک عامل^۷ یاد می‌گیرد با انجام دادن یک عمل در محیط پاداش خاصی را بهینه کند (مثلاً رایانه‌ای که Tetris بازی می‌کند)، مستثنی می‌شوند. هدف یادگیری تحت نظارت، یادگیری یک مدل پیش‌بینی است که ویژگی‌های داده‌ها (به عنوان مثال اندازه خانه، مکان، نوع طبقه، و ...) را به یک خروجی (مثلاً قیمت خانه) نگاشت می‌کند. اگر خروجی طبقه‌ای^۸ باشد، کار را طبقه‌بندی و اگر عددی باشد، رگرسیون نامیده می‌شود. الگوریتم یادگیری ماشین یک مدل را با تخمین^۹ پارامترها (مانند وزن‌ها) یا ساختارهای یادگیری (مانند درختان) یاد می‌گیرد. الگوریتم توسط یک امتیاز یا تابع خطای^{۱۰} کنترل می‌شود تا بتواند آن را به حداقل برساند. در مثال ارزش خانه، یادگیری ماشین، تفاوت بین قیمت تخمینی خانه و قیمت پیش‌بینی شده را به حداقل می‌رساند. سپس می‌توان از یک مدل یادگیری ماشین کاملاً آموزش دیده برای پیش‌بینی نمونه‌های^{۱۱} جدید استفاده کرد.

تخمین قیمت خانه، پیشنهادهای محصول، تشخیص تابلوهای خیابان، پیش‌بینی اولیه اعتبار و تشخیص تقلب: همه این مثال‌ها، وجه اشتراکی دارند. این وجه مشترک آن است که می‌توان آن‌ها را با یادگیری ماشین حل کرد. وظایف متفاوت است، اما رویکرد یکسان است:

¹ supervised machine learning / supervised machine learning

² Outcome

³ tasks

⁴ Clustering

⁵ unsupervised learning

⁶ reinforcement learning

⁷ agent

⁸ categorical

⁹ estimating

¹⁰ loss function

¹¹ instances

مرحله ۱: جمع‌آوری داده‌ها^۱. هرچقدر بیشتر بهتر، داده‌ها باید حاوی نتیجه‌های باشد که می‌خواهید پیش‌بینی کنید و اطلاعات اضافی که با استفاده از آن می‌توان پیش‌بینی کرد. برای آشکارساز علائم خیابان ("آیا تابلوی خیابان در تصویر وجود دارد؟")، تصاویر خیابان را جمع‌آوری می‌کنید و برچسب می‌زنید که آیا تابلوی خیابان قابل مشاهده است یا خیر. برای یک پیش‌بینی‌کننده اولیه اعتبار، به داده‌های گذشته در مورد وام‌های واقعی، اطلاعاتی در مورد اینکه آیا مشتریان وام‌های خود را نکول کرده‌اند، و داده‌هایی که به شما در انجام پیش‌بینی‌ها کمک می‌کنند، مانند درآمد، اعتبارات گذشته اولیه و غیره نیاز دارید. برای یک برنامه تخمین زن خودکار ارزش خانه، می‌توانید داده‌ها را از فروش‌های قبلی خانه و اطلاعات مربوط به املاک مانند اندازه، مکان و غیره جمع‌آوری کنید.

مرحله ۲: این اطلاعات را در یک الگوریتم یادگیری ماشین وارد کنید که یک مدل آشکارساز علامت، یک مدل رتبه‌بندی اعتبار یا یک تخمین‌گر ارزش خانه ایجاد می‌کند.

مرحله ۳: از مدل با داده‌های جدید استفاده کنید. مدل را در یک محصول یا فرآیند ادغام کنید، مانند ماشین خودران، فرآیند درخواست اعتبار یا وب سایت بازار املاک.

ماشین‌ها در بسیاری از کارها، مانند بازی شطرنج (یا اخیراً G0) یا پیش‌بینی آب و هوا از انسان‌ها پیشی می‌گیرند. حتی اگر ماشین به خوبی یک انسان باشد یا در یک وظیفه کمی‌بدتر باشد، مزایای زیادی از نظر سرعت، تکرارپذیری و مقیاس‌پذیری وجود دارد. یک مدل یادگیری ماشین پیاده‌سازی شده، می‌تواند یک کار را بسیار سریع‌تر از انسان‌ها انجام دهد، نتایج قابل اعتمادی را ارائه می‌دهد و می‌تواند بینهایت بار کپی شود. تکرار یک مدل یادگیری ماشین در ماشین دیگر سریع و ارزان است. آموزش یک انسان برای انجام یک وظیفه می‌تواند چندین ده طول بکشد (مخصوصاً در جوانی) و بسیار پرهزینه است. یک عیب عمده استفاده از یادگیری ماشین این است که بینش در مورد داده‌ها و وظیفه‌ای که ماشین حل می‌کند در مدل‌های پیچیده، پنهان است. برای توصیف یک شبکه عصبی عمیق^۲ به میلیون‌ها عدد نیاز دارید، و هیچ راهی برای درک کامل مدل وجود ندارد. مدل‌های دیگر، مانند جنگل تصادفی^۳، از صدها درخت تصمیم تشکیل شده‌اند که به پیش‌بینی‌ها رأی^۴ می‌دهند. برای درک چگونگی تصمیم گیری، باید به آرا و ساختار هر یک از صدها درخت نگاه کنید. این کار، صرف‌نظر از اینکه چقدر باهوش هستید یا حافظه شما چقدر خوب کار می‌کند، ممکن نیست. بهترین مدل‌ها اغلب ترکیبی از چندین مدل (معمولًا ترکیبی^۵ نامیده می‌شوند) هستند که قابل تفسیر نیستند، حتی اگر هر مدل منفرد

¹ Data collection

² Deep neural network

³ Random forest

⁴ Vote

⁵ Ensemble

قابل تفسیر باشد. اگر فقط بر روی بهبود عملکرد تمرکز کنید، به طور خودکار مدل‌های غیرشفاف‌تری^۱ خواهید داشت. مدل‌های برنده در مسابقات یادگیری ماشین اغلب ترکیبی از مدل‌ها یا مدل‌های بسیار پیچیده مانند درختان تقویت شده^۲ یا شبکه‌های عصبی عمیق هستند.

¹ Opaque

² Boosted trees

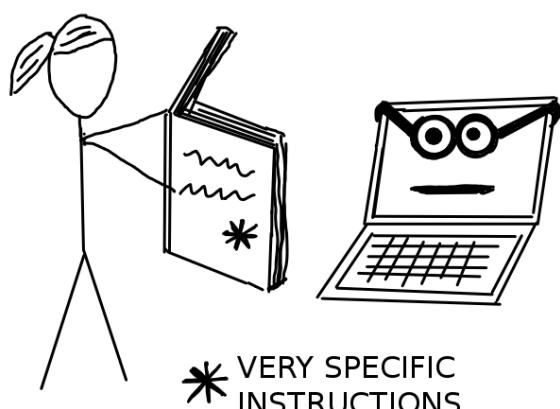
۲.۳ اصطلاحات

برای جلوگیری از سردرگمی به دلیل ابهام، در اینجا چند تعریف از اصطلاحات استفاده شده در این کتاب آورده شده است:

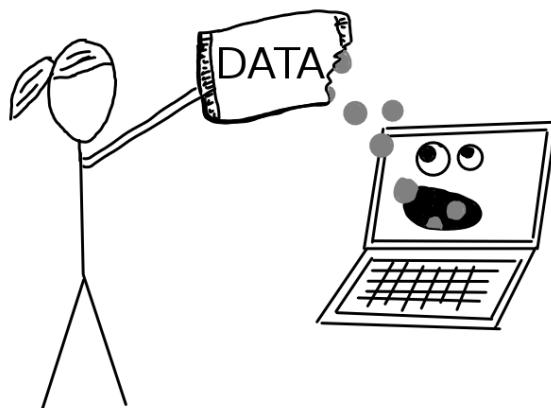
الگوریتم^۱ مجموعه‌ای از قوانین است که یک ماشین برای رسیدن به یک منظور خاص از آنها پیروی می‌کند (*Definition of Algorithm.*, 2017). یک الگوریتم را می‌توان به عنوان دستور العملی در نظر گرفت که ورودی‌ها، خروجی‌ها و تمام مراحل موردنیاز برای رسیدن از ورودی‌ها به خروجی را تعریف می‌کند. دستور العمل‌های آشپزی، الگوریتم‌هایی هستند در آن‌ها مواد اولیه آشپزی ورودی، غذای پخته شده خروجی و مراحل آماده سازی و پخت دستور العمل‌های الگوریتم هستند.

یادگیری ماشین^۲ مجموعه‌ای از روش‌هایی است که به رایانه‌ها اجازه می‌دهد از داده‌ها برای انجام و بهبود پیش‌بینی‌ها (مثلًاً تشخیص سرطان، فروش هفتگی، اعتبار اولیه) یاد بگیرند. یادگیری ماشین یک تغییر پارادایم از «برنامه‌نویسی عادی» است که در آن تمام دستور العمل‌ها باید به صراحت به رایانه داده شود، به «برنامه‌نویسی غیرمستقیم» که داده‌ها به رایانه ارائه می‌شود.

Without Machine Learning



With Machine Learning



یادگیرنده^۳ یا الگوریتم یادگیری ماشین^۴ برنامه‌ای است که برای یادگیری مدل یادگیری ماشین از داده‌ها استفاده می‌شود. نام دیگر "القاگر"^۵ است (به عنوان مثال "القاگر درخت").

¹ Algorithm

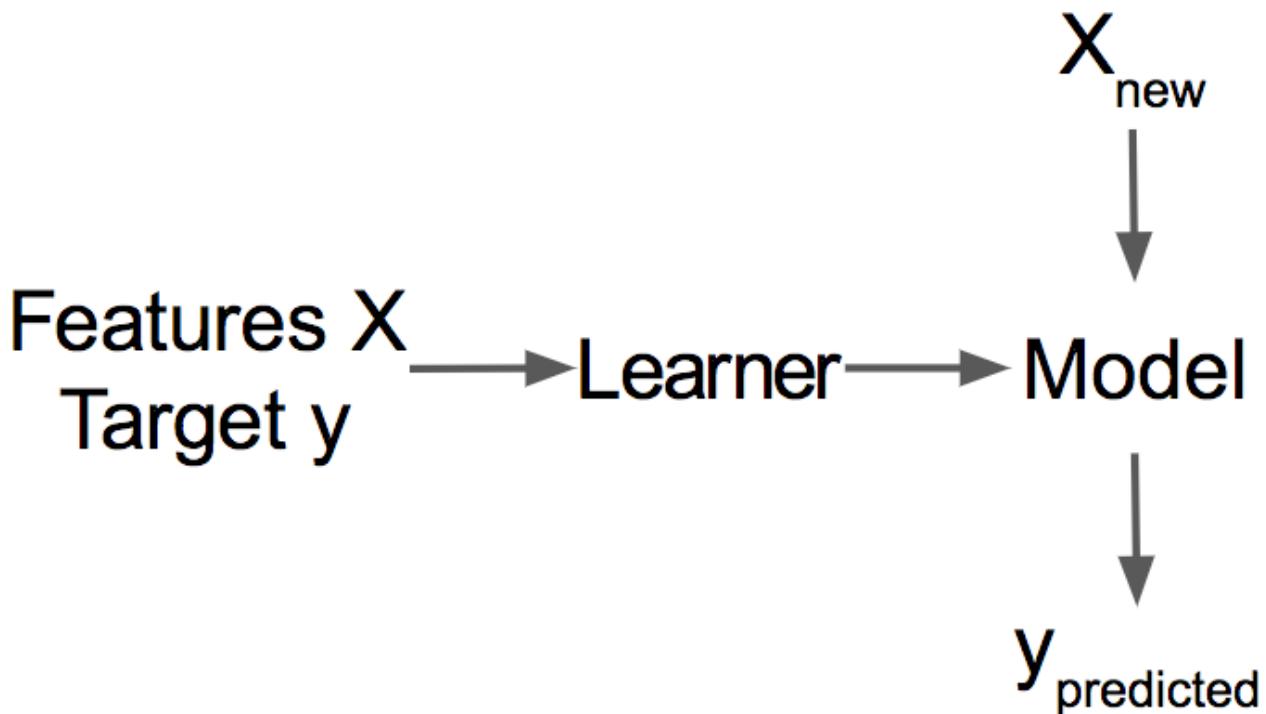
² Machine Learning

³ Learner

⁴ Machine Learning Algorithm

⁵ Inducer

مدل یادگیری ماشین^۱ برنامه‌ای است که ورودی‌ها را به پیش‌بینی‌ها نگاشت می‌کند. این مدل می‌تواند مجموعه‌ای از وزن‌ها برای یک مدل خطی یا شبکه عصبی باشد. نام‌های دیگر کلمه "مدل"، "پیش‌بینی‌کننده"^۲ یا - بسته به کار - "طبقه‌بند"^۳ یا "مدل رگرسیون"^۴ است. در فرمول‌ها، مدل یادگیری ماشین آموزش‌دیده \hat{f} یا $f(x)$ نامیده می‌شود.



شکل ۲.۱: یک یادگیرنده، مدلی را از داده‌های آموزشی برچسب گذاری شده یاد می‌گیرد. این مدل برای پیش‌بینی استفاده می‌شود.

مدل جعبه سیاه^۵ سیستمی است که مکانیسم‌های داخلی خود را آشکار نمی‌کند. در یادگیری ماشین، "جعبه سیاه" به مدل‌هایی اطلاق می‌شود که نمی‌توان با نگاه‌کردن به پارامترهای آنها، درکشان کرد (مثلاً یک شبکه عصبی). نقطه مقابل جعبه سیاه گاهی اوقات به عنوان جعبه سفید^۶ شناخته می‌شود که در این کتاب به عنوان مدل قابل تفسیر از آن یاد می‌شود. روش‌های آگنوستیک مدل برای تفسیرپذیری، مدل‌های یادگیری ماشین را به عنوان جعبه‌های سیاه در نظر می‌گیرند، حتی اگر چنین نباشند.

¹ Machine Learning Model

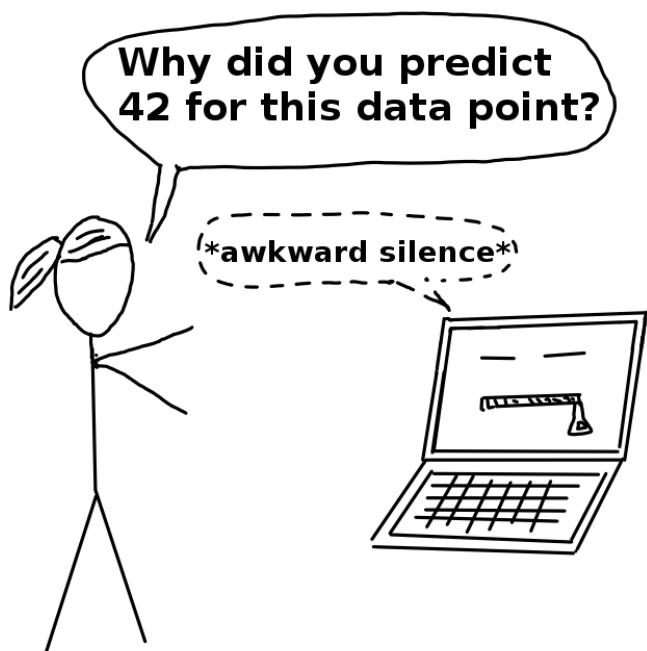
² Predictor

³ Classifier

⁴ Regression model

⁵ Black box model

⁶ White box



یادگیری ماشین قابل تفسیر^۱ به روش‌ها و مدل‌هایی اشاره دارد که رفتار و پیش‌بینی‌های سیستم‌های یادگیری ماشین را برای انسان قابل درک می‌کنند.

مجموعه‌داده^۲، جدولی از داده‌هاست که ماشین از آن یاد می‌گیرد. مجموعه‌داده شامل ویژگی‌ها و هدف پیش‌بینی است. مجموعه‌داده، هنگامی که برای آموزش یک مدل استفاده می‌شود، داده آموزشی^۳ نامیده می‌شود. نمونه^۴، یک ردیف در مجموعه‌داده است. نام‌های دیگر «نمونه» عبارت‌اند از: (داده) نقطه^۵، مثال^۶، مشاهده^۷. یک نمونه از مقادیر ویژگی⁽ⁱ⁾ x و مقدار معلوم y_i تشکیل شده است.

ویژگی‌ها^۸، ورودی‌هایی هستند که برای پیش‌بینی یا طبقه‌بندی استفاده می‌شوند. یک ویژگی یک ستون در مجموعه‌داده است. در سرتاسر کتاب، ویژگی‌ها قابل تفسیر فرض می‌شوند، به این معنی که درک معنای آنها آسان است، مانند دمای یک روز معین یا قد یک فرد. تفسیرپذیری ویژگی‌ها یک فرض بزرگ است. اگر درک ویژگی‌های ورودی سخت باشد، درک اینکه مدل چه کاری انجام می‌دهد بسیار سخت‌تر است. ماتریس تمام ویژگی‌ها X نامیده می‌شود و $x^{(i)}$ برای یک نمونه. بردار یک ویژگی واحد برای همه نمونه‌ها x_j است و مقدار ویژگی j برای نمونه i ، $x_j^{(i)}$ می‌باشد.

¹ Interpretable Machine Learning

² Dataset

³ Training

⁴ Instance

⁵ Point

⁶ Example

⁷ Observation

⁸ Features

هدف^۱، اطلاعاتی است که ماشین یاد می‌گیرد تا آن را پیش‌بینی کند. در فرمول‌های ریاضی معمولاً هدف y یا \hat{y} نامیده می‌شود.

وظیفه یادگیری ماشین^۲ ترکیب مجموعه ویژگی‌ها با یک هدف است. بسته به نوع هدف، وظیفه می‌تواند به عنوان مثال طبقه‌بندی، رگرسیون، تجزیه و تحلیل بقا^۳، خوش‌بندی^۴، یا تشخیص داده پرت^۵ باشد. پیش‌بینی^۶ مقدار هدفی است که مدل یادگیری ماشین بر اساس ویژگی‌های داده شده، «حدس می‌زند^۷». در این کتاب، پیش‌بینی مدل با $(x^{(i)})\hat{f}$ یا \hat{y} نشان داده شده است.

¹ Target

² Machine Learning Task

³ survival analysis

⁴ clustering

⁵ outlier detection

⁶ Prediction

⁷ guesses

فصل ۳ تفسیرپذیری

تعریف تفسیرپذیری (از نظر ریاضی) دشوار است. من تعریف (غیر ریاضی) تفسیرپذیری که توسط Miller (2019) ارائه شده است را دوست دارم. این تعریف عبارت این‌گونه است: تفسیرپذیری درجه‌ای است که یک انسان می‌تواند علت یک تصمیم را درک کند. یکی تعریف دیگر این است: تفسیرپذیری درجه‌ای است که یک انسان می‌تواند به طور مداوم نتیجه مدل را پیش‌بینی کند (Kim et al., 2016). هرچه قابلیت تفسیر یک مدل یادگیری ماشین بالاتر باشد، درک اینکه چرا تصمیم‌ها یا پیش‌بینی‌های خاصی گرفته شده‌اند، برای فردی آسان‌تر است. یک مدل از مدل دیگر قابل تفسیرتر است اگر درک تصمیمات آن برای انسان آسان‌تر از تصمیمات مدل دیگر باشد. من از هر دو اصطلاح قابل تفسیر^۱ و قابل توضیح^۲ به جای یکدیگر استفاده خواهم کرد. مانند (Miller, 2019) من فکر می‌کنم منطقی است که بین اصطلاحات تفسیرپذیری / توضیح پذیری و توضیح تفاوت قائل شویم. من از "توضیح"^۳ برای توضیح پیش‌بینی‌های فردی استفاده خواهم کرد. برای یادگیری آنچه که ما انسان‌ها به عنوان یک توضیح خوب می‌دانیم، بخش توضیحات انسان پسند را مطالعه فرمایید.

یادگیری ماشین قابل تفسیر یک اصطلاح مفید است که "استخراج دانش از یک مدل یادگیری ماشین درباره روابط موجود در داده‌ها یا یادگیری شده توسط مدل" را نشان می‌دهد (Murdoch et al., 2019).

¹ Interpretable

² Explainable

³ Explanation

۳.۱ اهمیت تفسیرپذیری

اگر یک مدل یادگیری ماشین عملکرد خوبی دارد، چرا فقط به مدل اعتماد نمی‌کنیم و از چرایی تصمیم خاصی صرفنظر نمی‌کنیم؟ مشکل این است که یک معیار واحد، مانند دقت طبقه‌بندی، توصیف ناقصی از اکثر وظایف دنیای واقعی است (Doshi-Velez & Kim, 2017).

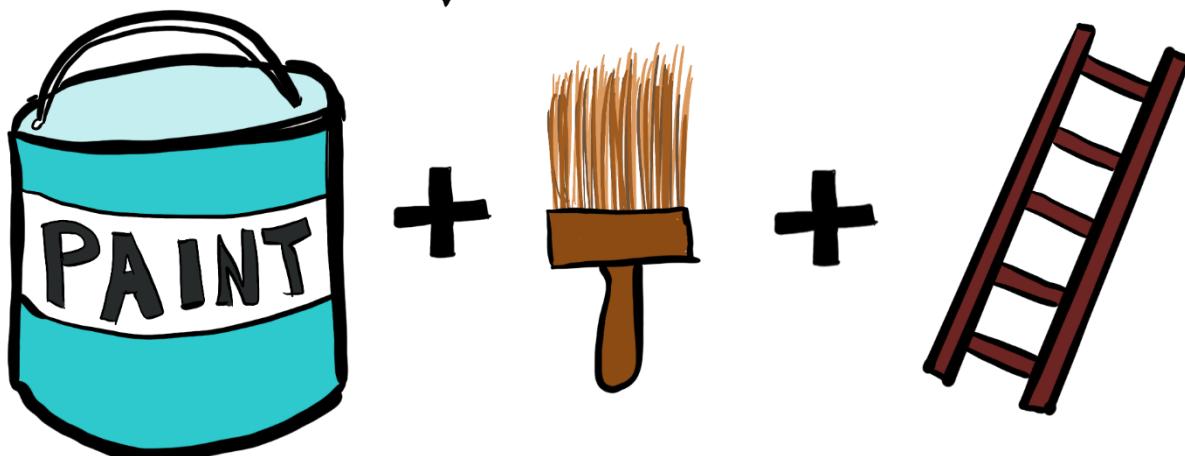
اجازه دهید به دلایلی که چرا تفسیرپذیری بسیار مهم است، عمیق‌تر بپردازیم. وقتی نوبت مدل‌سازی پیش‌بینی فرا می‌رسد، باید یک مبادله انجام دهید: آیا فقط می‌خواهید بدانید چه چیزی پیش‌بینی می‌شود؟ به عنوان مثال، احتمال اینکه یک مشتری سرگردان شود یا اینکه برخی از داروها چقدر برای بیمار موثر است. یا می‌خواهید بدانید چرا پیش‌بینی انجام شد و احتمالاً برای تفسیرپذیری بهتر، چقدر کاهش عملکرد پیش‌بینی‌کننده را تحمل می‌کنید؟ در برخی موارد، برای شما مهم نیست که چرا تصمیم گرفته شده است، کافی است بدانید که عملکرد پیش‌بینی‌کننده روی مجموعه‌داده آزمایشی خوب بوده است. اما در موارد دیگر، دانستن «چرایی» می‌تواند به شما کمک کند تا درباره مساله، داده‌ها و دلیل شکست یک مدل بیشتر بدانید. برخی از مدل‌ها ممکن است نیازی به توضیح نداشته باشند زیرا در یک محیط کم خطر استفاده می‌شوند، به این معنی که یک اشتباہ عواقب جدی در پی نخواهد داشت (مثلاً سیستم پیشنهاد دهنده فیلم) یا روشی که قبل‌اً به طور گسترده مورد مطالعه و ارزیابی قرار گرفته است (مثلاً تشخیص کاراکتر نوری). نیاز به تفسیرپذیری از نقصان در فرمول بندی مسئله ناشی می‌شود (Doshi-Velez & Kim, 2017)، به این معنی که برای مشکلات یا وظایف خاص، پیش‌بینی (چه چیزی) کافی نیست. مدل علاوه بر پیش‌بینی، باید توضیح دهد که چگونه به پیش‌بینی رسید (چرا)، زیرا یک پیش‌بینی صحیح فقط تا حدی مشکل اصلی شما را حل می‌کند. دلایل زیر باعث نیاز برای تفسیرپذیری و توضیح می‌شود (Doshi-Velez & Kim, 2017).

کنجکاوی و یادگیری انسان: انسان‌ها یک مدل ذهنی از محیط خود دارند که زمانی که اتفاق غیرمنتظره ای رخ می‌دهد به روز می‌شود. این به روز رسانی با یافتن توضیحی برای رویداد غیرمنتظره انجام می‌شود. به عنوان مثال، یک انسان به طور غیرمنتظره ای احساس بیماری می‌کند و می‌پرسد: "چرا اینقدر احساس بیماری می‌کنم؟". او یاد می‌گیرد که هر بار که توت قرمز را می‌خورد بیمار می‌شود. او مدل ذهنی خود را به روز می‌کند و به این نتیجه می‌رسد که توت‌ها باعث بیماری شده‌اند و بنابراین باید از آن‌ها اجتناب شود. هنگامی که از مدل‌های یادگیری ماشین غیرشفاف در تحقیقات استفاده می‌شود، اگر مدل فقط پیش‌بینی‌هایی را بدون توضیح ارائه دهد، یافته‌های علمی کاملاً پنهان می‌مانند. برای تسهیل یادگیری و ارضای کنجکاوی در مورد اینکه چرا برخی پیش‌بینی‌ها یا رفتارها توسط ماشین‌ها ایجاد می‌شوند، تفسیرپذیری و توضیحات بسیار مهم است. البته انسان‌ها برای هر اتفاقی که می‌افتد نیازی به توضیح ندارند. برای اکثر مردم اشکالی ندارد که ندانند کامپیوتر

چگونه کار می‌کند. اتفاقات غیرمنتظره ما را کنجکاو می‌کند. به عنوان مثال: چرا کامپیوتر من به طور غیرمنتظره ای خاموش می‌شود؟

میل انسان به یافتن معنا در جهان ارتباط نزدیک با یادگیری دارد. ما می‌خواهیم تضادها یا ناسازگاری‌ها را با عناصر ساختارهای دانش خود هماهنگ کنیم. "چرا سگم مرا گاز گرفت با توجه به اینکه قبلاً این کار را نکرده بود؟" ممکن است یک انسان بپرسد بین آگاهی از رفتار گذشته سگ و تجربه ناخوشایند گاز تازه گرفته شده تنافض وجود دارد. توضیحات دامپزشک تضاد صاحب سگ را برطرف می‌کند: "سگ تحت استرس بود و گاز گرفته بود." هر چه تصمیم یک ماشین بیشتر بر زندگی یک فرد تأثیر بگذارد، توضیح رفتار ماشین اهمیت بیشتری دارد. فرض کنید یک مدل یادگیری ماشین به صورت غیرمنتظره درخواست وام متقاضیان را رد کند. این ناسازگاری بین انتظار متقاضیان و واقعیت را فقط با نوعی توضیح می‌توان آشتبانی داد. در عمل، توضیحات نباید شرایط را به طور کامل توضیح دهند، بلکه باید به یک علت اصلی بپردازنند. مثال دیگر پیشنهاد محصول الگوریتمی است. من شخصاً همیشه به این فکر می‌کنم که چرا محصولات یا فیلم‌های خاصی به صورت الگوریتمی به من توصیه شده‌اند. اغلب کاملاً واضح است: تبلیغات، من را در اینترنت دنبال می‌کند. چون اخیراً یک ماشین لباسشویی خریدم، می‌دانم که در روزهای آینده تبلیغات ماشین لباسشویی دریافت خواهم کرد. بله، اگر در خرید قبل کلاه زمستانی در سبد خریدم وجود داشته باشد، پیشنهاد دستکش منطقی است. الگوریتم این فیلم را پیشنهاد می‌کند، زیرا کاربرانی که فیلمی که من دوست داشتم، را دوست داشته‌اند، از فیلم پیشنهادی لذت برده‌اند. شرکت‌های اینترنتی به طور فزاینده‌ای توضیحاتی را به توصیه‌های خود اضافه می‌کنند. یک مثال خوب، توصیه‌های محصول بر اساس ترکیبات محصولاتی است که اغلب با یکدیگر خریداری می‌شوند:

Frequently Bought Together



شکل ۳.۱: محصولات پیشنهادی که اغلب با هم خریداری می‌شوند.

در بسیاری از رشته‌های علمی تغییری از روش‌های کیفی به کمی (به عنوان مثال جامعه شناسی، روانشناسی)، و همچنین به سمت یادگیری ماشین (زمیست شناسی، ژنومیک) وجود دارد. هدف علم کسب دانش است، اما بسیاری از مشکلات با مجموعه داده‌های بزرگ و مدل‌های یادگیری ماشین جعبه سیاه حل می‌شوند. گاهی خود مدل به جای داده، به منبع دانش تبدیل می‌شود. تفسیرپذیری امکان استخراج این دانش اضافی، که توسط مدل ایجاد شده است را ممکن می‌سازد.

مدل‌های یادگیری ماشین وظایف دنیای واقعی را انجام می‌دهند که به اقدامات ایمنی و تست نیاز دارند. تصور کنید یک ماشین خودران به طور خودکار دوچرخه سواران را بر اساس یک سیستم یادگیری عمیق، شناسایی می‌کند. شما می‌خواهید ۱۰۰٪ مطمئن باشید که انتزاعی که سیستم یادگرفته است بدون خطأ است، زیرا زیزگرفتن دوچرخه سواران بسیار ناگوار است. یک توضیح ممکن است نشان دهد که مهم‌ترین ویژگی یادگرفته شده، تشخیص دو چرخ دوچرخه است، و این توضیح به شما کمک می‌کند در مورد، شرایطی که این چرخ‌ها را پوشش می‌دهند مانند دوچرخه با کیسه‌های جانبی که تا حدی چرخ‌ها را می‌پوشانند فکر کنید.

به طور پیش‌فرض، مدل‌های یادگیری ماشین، سوگیری‌ها را از داده‌های آموزشی دریافت می‌کنند. این می‌تواند مدل‌های یادگیری ماشین شما را به ماشین سوگیری تبدیل کند که علیه گروه‌های دارای نمایندگی کمتر، تبعیض قائل شوند. تفسیرپذیری یک ابزار اشکال زدایی مفید برای تشخیص سوگیری است. در مدل‌های یادگیری ماشین ممکن است این اتفاق بیفتد که مدل یادگیری ماشین که برای تأیید یا رد خودکار درخواست‌های اعتباری آموزش داده‌اید، علیه اقلیتی که از لحاظ تاریخی از حق امتیاز محروم شده‌اند تبعیض قائل شود. هدف اصلی شما اعطای وام فقط به افرادی است که در نهایت آن وام‌ها را بازپرداخت خواهند کرد. ناقص بودن فرمول مساله در این مورد، در این واقعیت نهفته است که شما نه تنها می‌خواهید نکول وام را به حداقل برسانید، بلکه موظف هستید بر اساس جمعیت‌شناسی خاص تبعیض قائل نشوید. این یک محدودیت اضافی است که بخشی از فرمول مساله شما است (اعطای وام به روشی کم خطر و قابل قبول) که توسط تابع خطایی که مدل یادگیری ماشین برای آن بهینه سازی شده است، در نظر گرفته نمی‌شود.

فرآیند ادغام ماشین‌ها و الگوریتم‌ها در زندگی روزمره ما نیازمند تفسیرپذیری برای افزایش پذیرش اجتماعی است. مردم، باورها، امیال، نیات و غیره را به اشیا نسبت می‌دهند. در یک آزمایش معروف، Heider and Simmel (1944) به شرکت کنندگان در آزمایش، فیلم‌هایی از اشکال نشان داد که در این فیلم‌ها، یک دایره، یک "در" را برای ورود به یک "اتاق" (که یک مستطیل ساده بود) باز می‌کرد. شرکت‌کنندگان اعمال شکل‌ها را همانند یک عامل انسانی توصیف می‌کردند. نیات و حتی احساسات و ویژگی‌های شخصیتی، به اشکال نسبت می‌دادند. ربات‌ها مثال خوبی هستند. من نام جاروبرقی ام را Doge گذاشته ام. اگر Doge گیر کند، فکر می‌کنم: "Doge می‌خواهد به تمیز کردن ادامه دهد، اما از من کمک می‌خواهد چون گیر کرده است." وقتی Doge تمیز

کردن را تمام می‌کند و پریز خانه را برای شارژ مجدد جستجو می‌کند، فکر می‌کنم: "Doge میل به شارژ مجدد دارد و قصد دارد پریز خانه را پیدا کند." همچنین ویژگی‌های شخصیتی را به Doge نسبت می‌دهم: "Doge کمی خنگ است، اما زیباست." اینها افکار من است، به خصوص وقتی متوجه می‌شوم که Doge در حالی که خانه را با جاروبرقی، جارو می‌کشد، گیاهی را له کرده است. ماشین یا الگوریتمی که پیش‌بینی‌های خود را توضیح می‌دهد، مقبولیت بیشتری پیدا می‌کند. در بخش توضیحات انسان پسند را گفته می‌شود، که استدلال می‌کند که توضیحات، یک فرآیند اجتماعی هستند.

از توضیحات برای مدیریت تعاملات اجتماعی استفاده می‌شود. توضیح دهنده با ایجاد معنای مشترک از چیزی، بر اعمال، احساسات و باورهای گیرنده توضیح تأثیر می‌گذارد. برای اینکه یک ماشین با ما تعامل داشته باشد، ممکن است نیاز داشته باشد که احساسات و باورهای ما را شکل دهد. ماشین‌ها باید ما را مقاعد کنند تا بتوانند به هدف مورد نظر خود برسند. اگر ربات جاروبرقی، خود را تا حدی توضیح نمی‌داد، به طور کامل نمی‌پذیرفتند. جاروبرقی با توضیح اینکه گیر کرده است، به عنوان مثال، یک «حادثه» (مانند گیر کردن دویاره روی فرش حمام...) به جای توقف کار بدون هیچ توضیحی، معنای مشترکی ایجاد می‌کند. جالب اینجاست که ممکن است بین هدف ماشین توضیح دهنده (ایجاد اعتماد) و هدف گیرنده (درک پیش‌بینی یا رفتار) ناهماهنگی وجود داشته باشد. شاید توضیح کامل برای اینکه چرا Doge گیر کرده می‌تواند این باشد که شارژ باتری، بسیار کم است، یکی از چرخ‌ها به درستی کار نمی‌کند و یک اشکال وجود دارد که باعث می‌شود ربات، بارها و بارها به نقطه‌ای برود که مانع وجود دارد. این دلایل (و چند مورد دیگر) باعث شد ربات گیر کند، اما فقط توضیح داد که چیزی مانع است و همین برای من کافی بود تا به رفتار آن اعتماد کنم و معنای مشترک آن تصادف را دریافت کنم. به هر حال، Doge دویاره در حمام گیر کرد. هربار که قصدارم Doge را در حالت جاروبرقی بگذارم باید فرش را بردارم. اما فقط توضیح داد که مانع در راه است و همین برای من کافی بود تا به رفتار آن اعتماد کنم و معنای مشترکی از آن حادثه پیدا کنم.



شکل ۳.۲: Doge، جاروبرقی ما، گیر کرده است. به عنوان توضیحی برای تصادف، Doge به ما گفت که باید روی سطح صاف باشد.

مدل‌های یادگیری ماشین را فقط زمانی می‌توان اشکال زدایی و بازرسی کرد، که بتوان آنها را تفسیر نمود. حتی در محیط‌های کم خطر، مانند توصیه‌های فیلم، توانایی تفسیر، هم در مرحله تحقیق و توسعه و هم پس از توسعه، ارزشمند است. وقتی از یک مدل در یک محصول استفاده می‌شود، ممکن است همه چیز اشتباه شود. تفسیر یک پیش‌بینی اشتباه، به درک علت خطا کمک می‌کند. این موضوع یک جهت گیری برای نحوه تعمیر سیستم ارائه می‌دهد. به عنوان مثال، یک طبقه‌بندی کننده هاسکی از گرگ را در نظر بگیرید که برخی هاسکی‌ها را به اشتباه به عنوان گرگ طبقه‌بندی می‌کند. با استفاده از روش‌های یادگیری ماشین قابل تفسیر، متوجه می‌شوید که طبقه‌بندی اشتباه به دلیل برف در تصویر است. طبقه‌بندی کننده یاد گرفته است که از برف به عنوان ویژگی برای یافتن تصاویر گرگ استفاده کند. استفاده از این ویژگی ممکن است برای تفکیک گرگ‌ها از هاسکی در مجموعه داده‌های آموزشی منطقی باشد، اما در دنیای واقعی این چنین نیست.

اگر می‌توانید اطمینان حاصل کنید که مدل یادگیری ماشین می‌تواند تصمیماتش را توضیح دهد، می‌توانید موارد زیر را نیز راحت‌تر بررسی کنید (Doshi-Velez & Kim, 2017):

انصار^۱: حصول اطمینان از اینکه پیش‌بینی‌ها بی‌طرفانه هستند و به طور ضمنی یا صریح علیه گروه‌های مهجور تبعیض قائل نمی‌شوند. یک مدل قابل تفسیر می‌تواند به شما بگوید که چرا تصمیم گرفته است

^۱ Fairness

که یک فرد خاص نباید وام دریافت کند، و قضاوت در مورد اینکه آیا این تصمیم بر اساس یک سوگیری جمعیت شناختی (مثلاً نژادی) است یا خیر، برای یک انسان آسان‌تر می‌شود.

حریم خصوصی^۱: اطمینان از این که از اطلاعات حساس در داده‌ها محافظت می‌شود.
قابلیت اطمینان^۲ یا استحکام^۳: اطمینان از اینکه تغییرات کوچک در ورودی منجر به تغییرات بزرگ در پیش‌بینی نمی‌شود.

علیت^۴: بررسی این موضوع که فقط روابط علی انتخاب شده باشند.
اعتماد^۵: در مقایسه با جعبه سیاه، اعتماد به سیستمی که تصمیمات خود را توضیح می‌دهد برای انسان آسان‌تر است.

زمانی که نیازی به تفسیرپذیری نداریم.

سناریوهای زیر نشان می‌دهند که چه زمانی نیازی به تفسیرپذیری مدل‌های یادگیری ماشین نداریم یا حتی نمی‌خواهیم.

اگر مدل تاثیر قابل توجهی نداشته باشد، نیازی به تفسیرپذیری نیست. تصور کنید شخصی به نام مایک روی یک پروژه جانبی یادگیری ماشین کار می‌کند تا بر اساس داده‌های فیس بوک پیش‌بینی کند که دوستانش برای تعطیلات بعدی خود کجا خواهند رفت. مایک فقط دوست دارد دوستانش را با حدسهای صحیح که آنها در تعطیلات کجا خواهند رفت، غافلگیر کند. اگر مدل اشتباه باشد مشکلی جدی به وجود نمی‌آید (در بدترین حالت فقط کمی باعث خجالت مایک است) و اگر مایک نتواند خروجی مدل خود را توضیح دهد نیز مشکلی وجود ندارد. اگر مایک شروع به ایجاد یک کسب و کار در مورد پیش‌بینی‌های مقصد تعطیلات کند، وضعیت تغییر می‌کند. اگر مدل اشتباه باشد، کسبوکار ممکن است ضرر کند، یا این که مدل ممکن است به دلیل تعصبات نژادی یادگیری‌شده برای برخی افراد بدتر عمل کند. به محض اینکه مدل تاثیر قابل توجهی، خواه مالی یا اجتماعی داشته باشد، قابلیت تفسیر، مهم می‌شود.

وقتی مسئله به خوبی مطالعه شده باشد، نیازی به تفسیرپذیری نیست. برخی از کاربردها به اندازه کافی مورد مطالعه قرار گرفته اند، به طوری که تجربه عملی خوبی از مدل وجود دارد و مشکلات مدل در طول زمان حل شده است. یک مثال خوب یک مدل یادگیری ماشین برای تشخیص کاراکترهای نوری است که تصاویر را از

¹ Privacy

² Reliability

³ Robustness

⁴ Causality

⁵ Trust

پاکت‌های نامه پردازش می‌کند و آدرس‌ها را استخراج می‌کند. سال‌ها تجربه با این سیستم‌ها وجود دارد و مشخص است که خوب کار می‌کنند. علاوه بر این، ما واقعاً علاقه‌ای به کسب بیانش اضافی نداریم.

تفسیر پذیری ممکن است افراد یا برنامه‌ها را قادر به **دستکاری سیستم** کند. مشکلات با کاربرانی که یک سیستم را فریب می‌دهند ناشی از عدم تطابق بین اهداف سازنده و کاربر یک مدل است. امتیازدهی اعتباری چنین سیستمی است زیرا بانک‌ها می‌خواهند اطمینان حاصل کنند که وام‌ها فقط به مقاضیانی داده می‌شود که احتمالاً آن‌ها را پس می‌دهند و مقاضیان قصد دارند وام را دریافت کنند حتی اگر بانک نخواهد به آنها وام بدهد. این عدم تطابق بین اهداف، انگیزه‌هایی را برای مقاضیان ایجاد می‌کند تا با سیستم بازی کنند تا شناس خود را برای دریافت وام افزایش دهند. اگر مقاضی بدانند که داشتن بیش از دو کارت اعتباری بر امتیاز او تأثیر منفی می‌گذارد، به سادگی سومین کارت اعتباری خود را برای بهبود امتیاز خود باطل می‌کند و پس از تأیید وام، کارت جدیدی اخذ می‌کند. در حالی که امتیاز او بهبود یافت، احتمال واقعی بازپرداخت وام بدون تغییر باقی ماند. سیستم را تنها در صورتی می‌توان بازی داد که ورودی‌ها یک ویژگی علی، وکالتی^۱ باشند و این ویژگی وکالتی تاثیر علی واقعی ندارد. در صورت امکان، باید از ویژگی‌های وکالتی اجتناب شود، زیرا آنها مدل‌ها را قابل بازی می‌کنند. به عنوان مثال، گوگل سیستمی به نام Google Flu Trends برای پیش‌بینی شیوع آنفلونزا ایجاد کرد. این سیستم جستجوهای گوگل را با شیوع آنفلونزا همبسته می‌کرد و عملکرد ضعیفی داشته است. توزیع پرس و جوهای جستجو تغییر کرد و Google Flu Trends بسیاری از شیوع آنفلونزا را از دست داد. جستجوی گوگل باعث آنفلونزا نمی‌شود. هنگامی که افراد علائمی مانند "تب" را جستجو می‌کنند، صرفاً این جستجو یک همبستگی با شیوع واقعی آنفلونزا دارد. به صورت ایده‌آل، مدل فقط باید از ویژگی‌های علی استفاده کند که قابلیت بازی کردن با آن‌ها وجود نداشته باشد.

¹ Proxy

۳.۲ طبقه‌بندی روش‌های تفسیرپذیری

روش‌های تفسیرپذیری یادگیری ماشین را می‌توان بر اساس معیارهای مختلف طبقه‌بندی کرد.

ذاتی^۱ یا تعقیبی^۲ این معیار تشخیص می‌دهد که آیا تفسیرپذیری، با محدود کردن پیچیدگی مدل یادگیری ماشین (ذاتی) به دست می‌آید یا با استفاده از روش‌هایی که مدل را پس از آموزش (تعقیبی) تجزیه و تحلیل می‌کند. تفسیرپذیری ذاتی به مدل‌هایی از یادگیری ماشین اشاره دارد که به دلیل ساختار ساده‌شان قابل تفسیر می‌باشند، مانند درخت‌های تصمیم کوتاه یا مدل‌های خطی محدود. تفسیرپذیری تعقیبی به روش‌های تفسیری اطلاق می‌شود که پس از آموزش مدل، استفاده می‌شوند. به عنوان مثال، اهمیت ویژگی جایگشت، یک روش تفسیر تعقیبی است. روش‌های تعقیبی می‌توانند برای مدل‌های قابل تفسیر ذاتی نیز اعمال شوند. به عنوان مثال، اهمیت ویژگی جایگشت را می‌توان برای درختان تصمیم نیز محاسبه کرد. سازماندهی فصول در این کتاب با تمایز بین مدل‌های ذاتی قابل تفسیر و روش‌های تفسیر تعقیبی (و آگنوستیک مدل) تعیین می‌شود.

نتیجه روش تفسیر: روش‌های مختلف تفسیر را می‌توان به طور تقریبی با توجه به نتایج آنها متمایز کرد.

- آماره خلاصه ویژگی^۳: بسیاری از روش‌های تفسیر، برای هر ویژگی آماره‌های خلاصه ارائه می‌دهند. برخی از روش‌ها یک عدد واحد را برای هر ویژگی برمی‌گردانند، مانند اهمیت ویژگی، یا یک نتیجه پیچیده‌تر، مانند pairwise feature interaction strengths که برای هر جفت ویژگی، یک عدد ارائه می‌دهد.

- تجسم خلاصه ویژگی^۴: بیشتر آماره‌های خلاصه ویژگی قابل ترسیم نیز هستند. برخی از خلاصه‌های ویژگی‌ها فقط زمانی قابل فهم هستند که ترسیم شوند و استفاده از جدول برای ارائه آن‌ها اشتباه است. وابستگی جزئی^۵ یک ویژگی، چنین موردی است. نمودارهای وابستگی جزئی منحنی‌هایی هستند که یک ویژگی و میانگین خروجی پیش‌بینی شده را نشان می‌دهند. بهترین راه برای ارائه وابستگی‌های جزئی، رسم منحنی به جای چاپ مختصات است.

- محتويات داخلی مدل^۶ (مثلاً وزن‌های آموزش داده شده): تفسیر مدل‌های قابل تفسیر ذاتی در این دسته قرار می‌گیرد. به عنوان مثال می‌توان به وزن در مدل‌های خطی یا ساختار درختی آموزش داده شده در درخت‌های تصمیم (ویژگی‌ها و آستانه‌های مورد استفاده برای تقسیم‌ها) اشاره کرد. مرز بین محتويات داخلی مدل و آماره خلاصه ویژگی، در مدل‌های خطی از بین می‌رود، زیرا وزن‌ها هم زمان هم

¹ Intrinsic

² Post hoc

³ Feature summary statistics

⁴ Feature summary visualization

⁵ Partial dependence

⁶ Model internals

محتویات داخلی مدل و هم آماره خلاصه ویژگی‌ها هستند. Another method that outputs model internals is the visualization of feature detectors learned in convolutional neural networks. Interpretability methods that output model internals are by definition model-specific (see next criterion). روش دیگری که مدل‌های داخلی را خروجی می‌دهد، تجسم آشکارسازهای ویژگی است که در شبکه‌های عصبی کانولوشن آموخته شده‌اند. روش‌های تفسیرپذیری که داخلی‌های مدل خروجی را به‌دست می‌آورند، طبق تعریف، مختص مدل هستند (معیار بعدی را ببینید).

- نقطه داده: این دسته شامل تمام روش‌هایی است که نقاط داده (از قبل موجود یا تازه ایجاد شده) را برای قابل‌تفسیر کردن یک مدل برمی‌گرداند. یکی از روش‌ها توضیحات خلاف واقع نامیده می‌شود. برای توضیح پیش‌بینی یک نمونه داده، این روش با تغییر برخی از ویژگی‌هایی که نتیجه پیش‌بینی شده به روши مرتبط تغییر می‌کند، یک نقطه داده مشابه پیدا می‌کند (مثلاً یک تلنگر در کلاس پیش‌بینی شده). مثال دیگر شناسایی نمونه‌های اولیه کلاس‌های پیش‌بینی شده است. برای مفید بودن، روش‌های تفسیری که نقاط داده جدید را تولید می‌کنند، مستلزم آن هستند که خود نقاط داده قابل‌تفسیر باشند. این برای تصاویر و متون به خوبی کار می‌کند، اما برای داده‌های جدولی با صدها ویژگی کمتر مفید است.
 - مدل قابل‌تفسیر ذاتی: یک راه حل برای تفسیر مدل‌های جعبه سیاه، تقریب آنها (به صورت کلی^۱ یا محلی^۲) با یک مدل قابل‌تفسیر است. خود مدل قابل‌تفسیر با نگاه کردن به پارامترهای مدل داخلی یا آماره خلاصه ویژگی تفسیر می‌شود.
- خاص مدل^۳ یا آگنوستیک مدل^۴ ابزارهای تفسیر خاص مدل به کلاس‌های خاصی از مدل‌ها، محدود می‌شوند. تفسیر وزن‌های رگرسیون در یک مدل خطی یک تفسیر خاص مدل است، زیرا - طبق تعریف - تفسیر مدل‌های ذاتی قابل‌تفسیر، همیشه خاص مدل است. ابزارهایی که به طور مثال فقط برای تفسیر شبکه‌های عصبی کار می‌کنند، خاص مدل هستند. ابزارهای آگنوستیک مدل را می‌توان برای هر مدل یادگیری ماشین استفاده کرد و پس از آموزش مدل (post hoc) بکار برده می‌شود. روش‌های آگنوستیک مدل معمولاً با تجزیه و تحلیل جفت‌های ویژگی ورودی و خروجی کار می‌کنند. طبق تعریف، این روش‌ها نمی‌توانند به اجزای داخلی مدل مانند وزن یا اطلاعات ساختاری دسترسی داشته باشند.

¹ Globally

² Locally

³ Model-specific

⁴ Model-agnostic

محلی^۱ یا کلی^۲؟ آیا روش تفسیر، یک پیش‌بینی فردی یا کل رفتار مدل را توضیح می‌دهد؟ یا حوزه‌ی عملکردش مابین این دو حالت است؟ برای اطلاعات بیشتر در مورد معیار حوزه تفسیر بخش حوزه تفسیرپذیری را مطالعه کنید.

¹ Local

² Global

۳.۳ حوزه^۱ تفسیرپذیری

یک الگوریتم، مدلی را آموزش می‌دهد که پیش‌بینی‌ها را تولید می‌کند. هر مرحله را می‌توان از نظر شفافیت یا تفسیر پذیری ارزیابی کرد.

۳.۳.۱ شفافیت الگوریتم^۲

الگوریتم چگونه مدل را ایجاد می‌کند؟

شفافیت الگوریتم در مورد چگونگی یادگیری الگوریتم از داده‌ها و نوع روابطی که می‌تواند یاد بگیرد، بحث می‌کند. اگر از شبکه‌های عصبی کانولوشن برای طبقه‌بندی تصاویر استفاده می‌کنید، می‌توانید توضیح دهید که الگوریتم، آشکارسازهای لبه و فیلترها را در پایین‌ترین لایه‌ها یاد می‌گیرد. این موضوع، درکی از نحوه عملکرد الگوریتم است. اما این موضوع برای مدل خاصی که در پایان آموزش داده می‌شود و برای چگونگی پیش‌بینی‌های فردی درکی ایجاد نمی‌کند. شفافیت الگوریتم فقط به دانش الگوریتم نیاز دارد و نیازی به داده‌ها یا مدل‌های آموزش داده شده ندارد. این کتاب بر تفسیرپذیری مدل تمرکز دارد و نه شفافیت الگوریتم. الگوریتم‌هایی مانند روش حداقل مربعات برای مدل‌های خطی به خوبی مطالعه و درک شده‌اند. این الگوریتم‌ها به عنوان شفافیت بالا شناخته می‌شوند. رویکردهای یادگیری عمیق (اعمال کردن یک گرادیان از طریق شبکه‌ای با میلیون‌ها وزن) کمتر درک شده‌اند و تحقیق در مورد کارکردهای درونی در مرکز توجه محققان می‌باشد. این موارد به عنوان شفافیت کم شناخته می‌شوند.

۳.۳.۲ تفسیرپذیری مدل کل نگر^۳، کلی

مدل آموزش‌دیده چگونه پیش‌بینی می‌کند؟

اگر بتوانید کل مدل را یکجا درک کنید، می‌توانید آن مدل را قابل تفسیر توصیف کنید (Lipton, 2018). برای توضیح خروجی مدل کلی، به مدل آموزش‌دیده، دانش الگوریتم و داده‌ها نیاز دارید. این سطح از تفسیرپذیری در مورد درک چگونگی تصمیم‌گیری مدل است که بر اساس یک دیدگاه کل نگر از ویژگی‌های آن و هر یک از اجزای آموزش دیده مانند وزن‌ها، پارامترهای دیگر، و ساختار، بنا شده است. کدام ویژگی‌ها مهم هستند و چه نوع تعاملاتی بین آنها وجود دارد؟ تفسیرپذیری مدل کلی به شما کمک می‌کند تا درکی از توزیع نتیجه هدف تان بر اساس ویژگی‌ها داشته باشید. دستیابی به تفسیرپذیری مدل جهانی در عمل بسیار دشوار است. هر مدلی که بیش از تعداد انگشت شماری پارامتر یا وزن داشته باشد، بعيد است قبل درک در حافظه کوتاه مدت یک انسان معمولی باشد. من فکر می‌کنم که شما نتوانید یک مدل خطی با ۵ ویژگی را تصور کنید، زیرا این تصور به معنای ترسیم ابر صفحه تخمینی به صورت ذهنی در یک فضای ۵ بعدی است. هر فضای ویژگی با بیش از ۳ بعد

¹ Scope

² Algorithm transparency

³ Holistic

به سادگی برای انسان قابل تصور نیست. معمولاً هنگامی که افراد سعی در درک یک مدل دارند، فقط بخش‌هایی از آن را در نظر می‌گیرند، مانند وزن‌ها در مدل‌های خطی.

۳.۳.۳ تفسیرپذیری مدل جهانی در سطح مدولار^۱

چگونه بخش‌هایی از مدل بر پیش‌بینی‌ها تأثیر می‌گذارد؟

یک مدل Naïve Bayes با صدھا ویژگی بزرگ‌تر از آن است که من و شما بتوانیم آن را در حافظه کاری خود نگه داریم. و حتی اگر بتوانیم تمام وزن‌ها را به خاطر بسپاریم، نمی‌توانیم به سرعت برای نقاط داده جدید پیش‌بینی کنیم. علاوه بر این، شما باید توزیع توانم^۲ همه ویژگی‌ها را در ذهن خود داشته باشید تا بتوانید اهمیت هر ویژگی را تخمین بزنید و یا اینکه بتوانید تخمین بزنید ویژگی‌ها به طور متوسط چگونه بر پیش‌بینی‌ها تأثیر می‌گذارند. این کار، غیر ممکن است. اما شما به راحتی می‌توانید یک وزن را درک کنید. درحالی که تفسیرپذیری مدل کلی معمولاً دور از دسترس است، شناس خوبی برای درک حداقل برخی از مدل‌ها در سطح مدولار وجود دارد. همه مدل‌ها در سطح پارامتر قابل تفسیر نیستند. برای مدل‌های خطی، بخش‌های قابل تفسیر، وزن‌ها هستند، برای درخت‌ها تقسیم‌ها (ویژگی‌های انتخابی به اضافه نقاط برش) و پیش‌بینی‌های گره برگ است. به عنوان مثال، به نظر می‌رسد که مدل‌های خطی را می‌توان به طور کامل در یک سطح مدولار تفسیر کرد، اما باید توجه داشت که تفسیر یک وزن منفرد، با تمام وزن‌های دیگر در هم تنبیده است. تفسیر وزن منفرد همیشه با این پاورقی همراه می‌شود که سایر ویژگی‌های ورودی در همان مقدار باقی می‌مانند، که در مورد بسیاری از شرایط واقعی صدق نمی‌کند. یک مدل خطی که ارزش یک خانه را پیش‌بینی می‌کند، که هم اندازه خانه و هم تعداد اتاق‌ها را در نظر می‌گیرد، می‌تواند وزن منفی برای ویژگی تعداد اتاق داشته باشد. دلیل این امر این است که بین این ویژگی و ویژگی اندازه خانه همبستگی شدید وجود دارد. در بازاری که مردم اتاق‌های بزرگ‌تر را ترجیح می‌دهند، یک خانه با اتاق‌های کمتر می‌تواند ارزش بیشتری نسبت به خانه‌هایی با اتاق‌های بیشتر داشته باشد، اگر هر دو خانه دارای اندازه یکسان باشند. وزن‌ها فقط در چارچوب سایر ویژگی‌های مدل معنا می‌یابند.

۳.۳.۴ تفسیر محلی برای یک پیش‌بینی واحد^۳

چرا مدل برای یک نمونه، پیش‌بینی خاصی انجام داد؟

می‌توانید روی یک نمونه تمرکز کنید و آنچه را که مدل برای این ورودی پیش‌بینی می‌کند بررسی کنید و توضیح دهید که چرا. اگر به یک پیش‌بینی خاص نگاه کنید، رفتار یک مدل پیچیده ممکن است قابل فهم تر باشد. به طور محلی، پیش‌بینی ممکن است به جای وابستگی پیچیده به برخی ویژگی‌ها، فقط به صورت خطی یا یکنواخت به آن‌ها بستگی داشته باشد. به عنوان مثال، ارزش یک خانه ممکن است به طور غیرخطی به اندازه آن

¹ Modular level

² Joint distribution

³ Local interpretability for a single prediction

بستگی داشته باشد. اما اگر فقط به یک خانه ۱۰۰ متر مربعی خاص نگاه می‌کنید، این احتمال وجود دارد که برای آن زیر مجموعه‌داده، پیش‌بینی مدل شما به صورت خطی به اندازه بستگی دارد. شما می‌توانید با شبیه سازی نحوه تغییر قیمت پیش‌بینی شده با افزایش یا کاهش اندازه ۱۰ متر مربع به این موضوع پی ببرید. بنابراین توضیحات محلی می‌توانند دقیق‌تر از توضیحات کلی باشند. این کتاب روش‌هایی را ارائه می‌کند که می‌توانند پیش‌بینی‌های فردی قابل تفسیرتری ایجاد کنند.

۳.۳.۵ تفسیر محلی برای گروهی از پیش‌بینی‌ها

چرا مدل پیش‌بینی‌های خاصی را برای گروهی از نمونه‌ها انجام داد؟

پیش‌بینی‌های مدل برای گروهی از نمونه‌ها را می‌توان با روش‌های تفسیر مدل کلی (در سطح مدولار) یا با توضیح نمونه‌های جداگانه توضیح داد. روش‌های کلی را می‌توان این گونه اعمال کرد: در نظر گرفتن قسمتی از نمونه‌ها، رفتار با آن‌ها به عنوان کل مجموعه‌داده و استفاده از روش‌های کلی برای این زیرمجموعه. روش‌های توضیح فردی را می‌توان در هر نمونه استفاده کرد و سپس برای کل گروه فهرست یا تجمعی کرد.

۳.۴ ارزیابی تفسیرپذیری

هیچ اتفاق نظر مشخصی در مورد اینکه تفسیرپذیری در یادگیری ماشین چیست، وجود ندارد. همچنین نحوه اندازه گیری آن مشخص نیست. اما برخی تحقیقات اولیه در این مورد و تلاش هایی برای تدوین برخی رویکردها برای ارزیابی، همان طور که در بخش بعدی توضیح داده شده است، وجود دارد.

(Doshi-Velez and Kim 2017) سه سطح اصلی را برای ارزیابی تفسیرپذیری پیشنهاد می کنند:

ارزیابی سطح کاربردی (وظیفه واقعی): توضیحات را در محصول قرار دهید و آن را توسط کاربر نهایی آزمایش کنید. نرم افزار تشخیص شکستگی را با یک قطعه یادگیری ماشین در نظر بگیرید که شکستگی ها را با استفاده از اشعه ایکس مکان یابی و علامت گذاری می کند. در سطح کاربردی، رادیولوژیست ها نرم افزار تشخیص شکستگی را مستقیماً برای ارزیابی مدل تست می کنند. این تست، نیاز به یک چیدمان تجربی خوب و درک چگونگی اندازه گیری کیفیت دارد. یک مبنای خوب برای ارزیابی، می تواند این باشد که یک انسان چقدر در توضیح همان تصمیم خوب است.

ارزیابی سطح انسانی (وظیفه ساده) یک ارزیابی سطح کاربردی ساده شده، است. تفاوت این است که این آزمایش ها به جای متخصصان آن حوزه با افراد عادی انجام می شود. این کار، آزمایش ها را ارزان تر می کند (مخصوصاً اگر متخصصان حوزه رادیولوژیست باشند) و یافتن تست کنندگان بیشتر، آسان تر می شود. به عنوان مثال، توضیحات متفاوتی به کاربر نشان داده می شود و می تواند بهترین را انتخاب می کند.

ارزیابی سطح عملکرد (وظیفه کالی) به انسان نیاز ندارد. این ارزیابی وقتی بهترین حالت را دارد که کلاس مدل مورد استفاده، قبلًا توسط شخص دیگری، در ارزیابی سطح انسانی، ارزیابی شده باشد. به عنوان مثال، ممکن است تعیین شود که کاربران نهایی درخت تصمیم را درک می کنند. در این حالت، یک جایگزین برای کیفیت توضیح، ممکن است عمق درخت باشد. درختان کوتاه تر، نمره توضیح پذیری بهتری دریافت می کنند. البته اضافه کردن این محدودیت زمانی منطقی است که عملکرد پیش بینی درخت، خوب باقی بماند و در مقایسه با درخت بزرگ تر خیلی کاهش نیابد.

بخش بعدی بر ارزیابی توضیحات برای پیش بینی های فردی در سطح عملکرد مرکز دارد. مشخصات مرتبط با توضیحاتی که برای ارزیابی آنها در نظر می گیریم، چیست؟

۳.۵ خواص توضیحات^۱

ما می‌خواهیم پیش‌بینی‌های یک مدل یادگیری ماشین را توضیح دهیم. برای رسیدن به این هدف، به برخی از روش‌های توضیحی تکیه می‌کنیم که الگوریتم‌هایی هستند که توضیحات را ایجاد می‌کند. یک توضیح عموماًً مقادیر ویژگی یک نمونه را به روشنی قابل درک برای انسان به پیش‌بینی مدل برای آن ویژگی مرتبط می‌کند. انواع دیگر توضیحات شامل مجموعه‌ای از نمونه‌های داده می‌باشد (مثلاً در مورد مدل k نزدیک‌ترین همسایه^۲). برای مثال، می‌توانیم خطر سلطان را با استفاده از یک ماشین بردار پشتیبان پیش‌بینی کنیم و پیش‌بینی‌ها را با استفاده از روش جایگزین محلی (local surrogate method) توضیح دهیم، که این روش جایگزین محلی، درخت‌های تصمیم را به عنوان توضیح، تولید می‌کند. یا می‌توانیم به جای ماشین بردار پشتیبان از مدل رگرسیون خطی استفاده کنیم. مدل رگرسیون خطی، مجهز به یک روش توضیحی (تفسیر وزن‌ها) می‌باشد.

ما نگاهی دقیق‌تر به خواص روش‌های توضیحات و توضیحات می‌اندازیم (Robnik-Šikonja & Bohanec, 2018). از این خواص می‌توان برای قضاوت در مورد خوب بودن روش توضیح یا توضیح استفاده کرد. البته مشخص نیست که این ویژگی‌ها چگونه به درستی اندازه گیری می‌شوند و در نتیجه یکی از چالش‌ها، نحوه محاسبه آنها است.

خواص روش‌های توضیح^۳

- قدرت بیان^۴، «زبان» یا ساختار توضیحاتی است که روش توضیحی موردنظر، قادر به ایجاد آن است. قدرت بیان یک روش توضیحی، می‌تواند قوانین IF-THEN، درخت‌های تصمیم، جمع وزنی، زبان طبیعی یا موارد دیگر باشد.
- ماتی‌شفافی^۵ یک روش بیان می‌کند که روش توضیح مذکور تا چه حد به درونیات مدل یادگیری ماشین، مانند پارامترهای آن، متکی است. برای مثال، روش‌های توضیح متکی بر مدل‌های قابل تفسیر ذاتی، مانند مدل رگرسیون خطی (خاص مدل) ماتی‌شفافی بالایی دارند. روش‌هایی که تنها به دستکاری ورودی‌ها و مشاهده تغییرات پیش‌بینی‌ها تکیه می‌کنند، ماتی‌شفافی صفر دارند. بسته به سناریو، سطوح مختلف ماتی‌شفافی ممکن است مطلوب باشد. مزیت ماتی‌شفافی بالا این است که روش

¹ Properties of explanations

² K-nearest neighbor method

³ Properties of explanation methods

⁴ Expressive power

⁵ Translucency

می‌تواند به اطلاعات بیشتری برای تولید توضیحات تکیه کند. مزیت ماتی‌شفافی کم‌این است که روش توضیح قابل حمل تر است.

- **قابلیت حمل**^۱ یک روش توضیح، بیانگر طیفی از مدل‌های یادگیری ماشین است که می‌توان آن روش توضیح را برای آن‌ها به کار برد.. روش‌هایی با ماتی‌شفافی پایین، قابلیت حمل بالاتری دارند، زیرا با مدل یادگیری ماشین مانند یک جعبه سیاه رفتار می‌کنند. Surrogate models احتمالاً بالاترین قابلیت Methods that only work for e.g. recurrent neural networks have low حمل را دارا هستند. portability. روش‌هایی که فقط برای شبکه‌های عصبی مکرر کار می‌کنند، قابلیت حمل کمی‌دارند.
- **پیچیدگی الگوریتمی**^۲، پیچیدگی محاسباتی روش توضیح را توصیف می‌کند. در مواردی که زمان تولید توضیحات یک گلوگاه^۳ باشد، مهم است که این ویژگی، در نظر گرفته شود.

خواص توضیحات تکی^۴

- **دقت**^۵: یک توضیح چقدر داده‌های دیده نشده را خوب پیش‌بینی می‌کند؟ دقت بالا به ویژه در صورتی مهم است که توضیح برای پیش‌بینی‌ها به جای مدل یادگیری ماشین استفاده شود. دقت پایین می‌تواند قابل قبول باشد اگر دقت مدل یادگیری ماشین نیز پایین باشد، و اگر هدف توضیح این باشد که مدل جعبه سیاه چه کاری انجام می‌دهد. در این مورد، فقط وفاداری مهم است.
- **وفاداری**^۶: توضیح چقدر به پیش‌بینی مدل جعبه سیاه نزدیک است؟ وفاداری بالا، یکی از مهم‌ترین ویژگی‌های توضیح است، زیرا استفاده از توضیحی که وفاداری پایینی دارد، برای توضیح مدل یادگیری ماشین بی فایده است. دقت و وفاداری ارتباط نزدیکی با هم دارند. اگر مدل جعبه سیاه، دقت بالایی داشته باشد و توضیحات دارای وفاداری بالا باشد، توضیحات نیز از دقت بالایی برخوردار خواهد بود. برخی از توضیحات فقط دارای وفاداری محلی هستند، به این معنی که توضیح، فقط برای زیر مجموعه از داده‌ها به خوبی به پیش‌بینی مدل برای زیرمجموعه‌ای از داده‌ها (مثالاً مدل‌های جایگزین محلی)^۷ یا حتی برای یک نمونه داده منفرد (مثالاً مقادیر Shapley) تقریب می‌زند.
- **سازگاری**^۸: یک توضیح، چقدر بین مدل‌هایی که برای یک وظیفه مشخص آموزش‌دیده اند و پیش‌بینی‌های مشابهی تولید می‌کنند، تفاوت دارد؟ برای مثال، من یک ماشین بردار پشتیبان و یک

¹ Portability

² Algorithm complexity

³ bottleneck

⁴ Properties of individual explanation

⁵ Accuracy

⁶ Fidelity

⁷ Local surrogate model

⁸ Consistency

مدل رگرسیون خطی را برای یک وظیفه آموزش می‌دهم و هر دو پیش‌بینی‌های بسیار مشابهی را تولید می‌کنند. من توضیحات را با استفاده از روشی که انتخاب می‌کنم محاسبه می‌کنم و تفاوت‌های توضیحات را تجزیه و تحلیل می‌کنم. اگر توضیحات بسیار شبیه به هم باشند، توضیحات بسیار سازگار هستند. من این خاصیت را تا حدودی گمراه کننده یافتم، زیرا این دو مدل می‌توانند از ویژگی‌های متفاوتی استفاده کنند، اما پیش‌بینی‌های مشابهی ایجاد کنند (این موضوع «اثر راشومون^۱» نامیده می‌شود). در این حالت، سازگاری بالا مطلوب نیست زیرا توضیحات باید بسیار متفاوت باشند. اگر مدل‌ها واقعاً بر روابط مشابه بنا شده باشند، سازگاری بالا، مطلوب است.

- **پایداری^۲:** توضیحات برای نمونه‌های مشابه، چقدر شبیه است؟ در حالی که سازگاری توضیحات بین مدل‌ها را مقایسه می‌کند، پایداری، توضیحات بین نمونه‌های مشابه را، برای یک مدل ثابت، مقایسه می‌کند. پایداری بالا به این معنی است که تغییرات جزئی در ویژگی‌های یک نمونه، توضیح را، اساسی تغییر نمی‌دهد (مگر اینکه این تغییرات جزئی، پیش‌بینی را نیز به شدت تغییر دهد). عدم پایداری می‌تواند نتیجه، واریانس زیاد روش توضیح باشد. به عبارت دیگر، روش توضیح به شدت تحت تأثیر تغییرات جزئی مقادیر ویژگی نمونه‌ای است که قرار است توضیح داده شود. فقدان پایداری همچنین می‌تواند ناشی از مؤلفه‌های غیرقطعی^۳ روش توضیح، مانند مرحله نمونه‌گیری داده‌ها، در روش جایگزین محلی باشد. پایداری بالا همیشه مطلوب است.

- **قابلیت درک^۴:** انسان‌ها چقدر توضیحات را درک می‌کنند؟ در ابتدا به نظر می‌رسد این خاصیت هم یک مورد مانند سایر خواص می‌باشد اما موضوعه بسیار حساس می‌باشد. تعریف و اندازه گیری این خاصیت دشوار اما بسیار مهم است. بسیاری از مردم قبول دارند که قابلیت درک بودن به مخاطب بستگی دارد. ایده‌هایی برای اندازه گیری قابلیت درک، شامل اندازه توضیح (تعداد ویژگی‌ها با وزن غیر صفر در یک مدل خطی، تعداد قوانین تصمیم‌گیری، و ...) یا تست اینکه افراد چقدر با استفاده از توضیحات می‌توانند رفتار مدل یادگیری ماشین را پیش‌بینی کنند. قابلیت درک ویژگی‌های استفاده شده در توضیح نیز باید در نظر گرفته شود. تبدیل^۵ پیچیده ویژگی‌ها ممکن است باعث شود، به نسبت ویژگی‌های اصلی بدن تبدیل، قابلیت قابل درک کمتر شود.

¹ Rashomon effect

² Stability

³ Non-deterministic

⁴ Comprehensibility

⁵ Transformation

- **قطعیت^۱**: آیا توضیح، قطعیت مدل یادگیری ماشین را منعکس می‌کند؟ بسیاری از مدل‌های یادگیری ماشین فقط پیش‌بینی می‌کنند بدون اینکه اظهارنظری در مورد اطمینان درست بودن پیش‌بینی مدل‌ها بکنند. اگر مدل، احتمال ^۴ درصدی سلطان، برای یک بیمار پیش‌بینی کند، آیا قطعیت دقیقاً برابر بیمار دیگری است که با مقادیر ویژگی‌های متفاوت احتمال ^۴ درصد دریافت می‌کند؟ توضیحی که شامل قطعیت مدل باشد، بسیار مفید است.
- **درجه اهمیت^۲**: توضیح چقدر اهمیت ویژگی‌ها یا اهمیت بخش‌هایی از توضیح را منعکس می‌کند؟ به عنوان مثال، اگر یک قانون تصمیم، به عنوان توضیحی برای یک پیش‌بینی فردی ایجاد شود، آیا مشخص است که کدام یک از شرایط قانون، مهم‌ترین بوده است؟
- **نوآوری^۳**: آیا توضیح نشان می‌دهد که نمونه داده ای که باید توضیح داده شود از یک منطقه "جدید" به دور از توزیع داده‌های آموزشی آمده است؟ در چنین مواردی، مدل ممکن است غیردقیق باشد و توضیح ممکن است بی فایده باشد. مفهوم نوآوری با مفهوم قطعیت مرتبط است. هر چه نوآوری بالاتر باشد، احتمال اینکه مدل به دلیل کمبود داده، از قطعیت کمتری برخوردار باشد، بیشتر است.
- **نمایندگی^۴**: توضیح چند نمونه را پوشش می‌دهد؟ توضیحات می‌توانند کل مدل را پوشش دهند (مثل تفسیر وزن‌ها در مدل رگرسیون خطی) یا فقط یک پیش‌بینی فردی را نمایندگی کنند (مثل مقادیر Shapley).

¹ Certainty

² Degree of importance

³ Novelty

⁴ Representativeness

۳.۶ توضیحات انسان پسند^۱

بیایید عمیق‌تر آنچه را که ما انسان‌ها به عنوان توضیحات «خوب» می‌پسندیم، بررسی کنیم و پیامدهای آن را برای یادگیری ماشین قابل تفسیر بیابیم. تحقیقات علوم انسانی می‌تواند به ما در یافتن این موضوع کمک کند. (Miller, 2019) تحقیقات مفصلی درباره توضیحات انجام داده است و این بخش، خلاصه‌ای بر اساس تحقیقات ایشان می‌باشد.

در این بخش، می‌خواهم شما را با موارد زیر آشنا کنم: به عنوان توضیحی برای یک رویداد، انسان‌ها توضیحات کوتاه (فقط ۱ یا ۲ علت) را ترجیح می‌دهند تا موقعیت فعلی را با موقعیتی که در آن رویداد رخ نمی‌داد، مقایسه کنند. به خصوص آوردن علل غیرعادی، توضیحات خوبی محسوب می‌شوند. توضیحات، تعاملات اجتماعی بین توضیح دهنده و گیرنده توضیح هستند و بنابراین زمینه اجتماعی تأثیر زیادی بر محتوای واقعی توضیح دارد. وقتی برای یک پیش‌بینی یا رفتار خاص به توضیحاتی با همه عوامل نیاز دارید، به دنبال توضیحی انسان‌پسند نیستید، بلکه یک توصیف علی کامل می‌خواهید. اگر از نظر قانونی ملزم به تعیین همه ویژگی‌های تأثیرگذار هستید یا اگر مدل یادگیری ماشین را اشکال‌زدایی می‌کنید، احتمالاً می‌خواهید یک توصیف علی داشته باشد. در این صورت به نکات زیر توجه نکنید. در سایر موارد که افراد غیرمتخصص یا افراد با کم وقت، دریافت کننده توضیحات هستند، بخش‌های زیر باید برای شما جالب باشد.

۳.۶.۱ توضیح چیست؟

توضیح پاسخ به یک سوال چرایی است (Miller, 2019).

- چرا درمان روی بیمار جواب نداد؟
- چرا درخواست وام من رد شد؟
- چرا هنوز زندگی فرازمنی با ما تماس نگرفته اند؟

دو سؤال اول را می‌توان با توضیح «روزمره» پاسخ داد، درحالی‌که سؤال سوم از دسته‌ی «پدیده‌های کلی‌تر علمی و سؤال‌های فلسفی» می‌باشد. ما روی توضیحات نوع «روزانه» تمرکز می‌کنیم، زیرا این توضیحات مربوط به یادگیری ماشین قابل تفسیر است. سؤالاتی که با «چگونه» شروع می‌شوند معمولاً می‌توانند به صورت سؤالات «چرا» بازنویسی شوند: «چگونه درخواست وام من رد شد؟» را می‌توان به «چرا درخواست وام من رد شد؟» تبدیل کرد.

اصطلاح «توضیح» به فرآیند اجتماعی و شناختی توضیح دادن و همچنین محصول این فرآیندها اشاره دارد. توضیح دهنده می‌تواند یک انسان یا یک ماشین باشد.

¹ Human-friend explanations

۳۶.۲ یک توضیح خوب چیست؟

این بخش، مطالب ارائه شده توسط Miller (2019) را در مورد توضیحات "خوب" به صورت فشرده بیان می‌کند و شامل مفاهیم قطعی برای یادگیری ماشین قابل تفسیر می‌باشد.

۱- توضیحات مقابله‌ای^۱ هستند (Lipton, 1990). انسان‌ها معمولاً نمی‌پرسند چرا یک پیش‌بینی خاص انجام شده است، بلکه می‌پرسند چرا این پیش‌بینی به جای پیش‌بینی دیگری انجام شده است. ما تمایل داریم در موارد حالات خلاف واقع^۲ فکر کنیم، به عنوان مثال "اگر ورودی X متفاوت بود، پیش‌بینی چگونه بود؟". برای پیش‌بینی قیمت مسکن، صاحب خانه ممکن است علاقه‌مند باشد که چرا قیمت پیش‌بینی شده در مقایسه با قیمتی که انتظار داشته است، بالاتر است. اگر درخواست وام من رد شود، اهمیتی برای شنیدن همه عواملی که باعث رد شدن وام شدند، ندارم. من مشتاق به شنیدن عواملی هستم که برای دریافت وام، باید تغییر کنم. من می‌خواهم تفاوت بین درخواست من و نسخه مورد پذیرش درخواستم را بدانم. تشخیص اینکه توضیحات مقایسه‌ای اهمیت دارند، یافته مهمی برای یادگیری ماشین قابل توضیح، است. از اکثر مدل‌های قابل تفسیر، می‌توانید توضیحی را استخراج کنید که به طور ضمنی پیش‌بینی یک نمونه را با پیش‌بینی یک نمونه داده مصنوعی یا میانگین نمونه‌ها مقایسه کند. پزشکان ممکن است بپرسند: "چرا دارو برای بیمار من جواب نمی‌دهد؟" و ممکن است توضیحی بخواهند که بیمارشان را با بیماری که دارو برای او موثر بوده و مشابه بیمار بدون پاسخ است، مقایسه کند. درک توضیحات مقابله‌ای آسانتر از توضیحات کامل است. توضیح کاملی در مورد این سوال پزشک، که چرا دارو کار نمی‌کند، ممکن است شامل موارد زیر باشد: بیمار به مدت ۱۰ سال به این بیماری مبتلا بوده است، ۱۱ ژن over-expressed هستند، بدن بیمار در تجزیه دارو به مواد شیمیایی بی اثر بسیار سریع عمل می‌کند ...the patients body is very quick in breaking the drug down into ineffective chemicals توضیح مقابله‌ای ممکن است بسیار ساده‌تر باشد: برخلاف بیمار پاسخ دهنده به دارو، بیمار بدون پاسخ دارای ترکیب خاصی از ژن‌ها است که اثربخشی دارو را کاهش می‌دهد. بهترین توضیح، توضیحی است که بیشترین تفاوت را، بین شرایط مورد نظر و شرایط مرجع، برجسته کند.

معنی آن برای یادگیری ماشین قابل تفسیر: انسان‌ها توضیح کاملی برای یک پیش‌بینی نمی‌خواهند، اما می‌خواهند تفاوت‌ها را با پیش‌بینی نمونه دیگری (که می‌تواند مصنوعی باشد) مقایسه کنند. ایجاد توضیحات مقابله‌ای وابسته به کاربرد است زیرا به نقطه مرجع برای مقایسه نیاز دارد. این توضیح ممکن است به نقطه داده ای که باید توضیح داده شود و به کاربر دریافت کننده توضیح، بستگی داشته باشد. یک کاربر وبسایت پیش‌بینی قیمت خانه، ممکن است بخواهد توضیحی در مورد پیش‌بینی قیمت خانه در مقایسه با خانه خود یا

¹ Contrastive

² Counterfactual

شاید خانه دیگری در وبسایت یا شاید با یک خانه متوسط در همسایگی اش داشته باشد. راه حل برای ایجاد خودکار توضیحات مقابله ای، ممکن است شامل یافتن نمونه‌های اولیه (prototypes) یا کهن الگوها (archetypes) در داده‌ها باشد.

۲- توضیحات انتخاب شده است. مردم انتظار توضیحی ندارند که فهرست واقعی و کاملی از علل یک رویداد را ارائه دهد. ما عادت کرده ایم که یک یا دو علت را از میان انواع علل احتمالی، به عنوان توضیح انتخاب کنیم. به عنوان مدرک اثبات این موضوع، تلویزیون را روش کنید و به این خبر گوش دهید: "کاهش قیمت سهام شرکت به دلیل واکنش منفی شدید دراثر مشکلات آخرین به روزرسانی نرم افزار محصول آن شرکت می‌باشد". سوباسا و تیمش به دلیل دفاع ضعیف بازی را باختند: آنها به حریفان خود فضای زیادی برای اجرای استراتژی خود دادند.

بی اعتمادی فزاینده به نهادهای مستقر و دولت ما، عوامل اصلی کاهش مشارکت رای دهنده‌گان است. این واقعیت که یک رویداد را می‌توان با علل مختلف توضیح داد، اثر را شومون نامیده می‌شود. را شومون یک فیلم ژاپنی است که داستان‌ها (توضیحات) مقایسه ای و جایگزین درباره مرگ یک سامورایی را روایت می‌کند. برای مدل‌های یادگیری ماشین مطلوبست، اگر بتوان یک پیش‌بینی خوب با استفاده از ویژگی‌های مختلف انجام داد. روش‌های ترکیبی (Ensemble) که چندین مدل را با ویژگی‌های مختلف (توضیحات مختلف) ترکیب می‌کند عموماً عملکرد خوبی دارند، زیرا میانگین‌گیری از آن «داستان‌ها» پیش‌بینی‌ها را مقاوم^۱تر و دقیق‌تر می‌کند. علاوه بر این، بیش از یک توضیح انتخابی وجود دارد که چرا یک پیش‌بینی خاص انجام شده است.

معنی آن برای یادگیری ماشین قابل تفسیر: توضیح را خیلی کوتاه بیان کنید، فقط ۱ تا ۳ دلیل بیاورید، حتی اگر دنیا پیچیده‌تر باشد. روش LIME کار کرد خوبی در این زمینه دارد.

۳- توضیحات اجتماعی هستند. توضیحات بخشی از مکالمه یا تعامل بین توضیح دهنده و گیرنده توضیح هستند. بافت اجتماعی، محتوا و ماهیت توضیحات را تعیین می‌کند. اگر بخواهم به یک فرد فنی، توضیح دهم که چرا ارزهای دیجیتال اینقدر ارزش دارند، مواردی از این قبیل می‌گوییم: «دفتر غیرمت مرکز، توزیع شده، مبتنی بر بلاک چین، که توسط یک نهاد مرکزی قابل کنترل نیست، افرادی که می‌خواهند امنیت داشته باشند را به خرید تشویق می‌کند و در نتیجه قیمت بالا می‌رود». اما اگر قرار بود به مادر بزرگم توضیح دهم، می‌گفتم: "بین، مادر بزرگ: ارزهای دیجیتال کمی شبیه طلای رایانه‌ای هستند. مردم طلا را دوست دارند و برای آن پول زیادی می‌پردازنند و جوانان نیز طلای کامپیوتری را دوست دارند و هزینه زیادی برای آن می‌پردازنند".

معنی آن برای یادگیری ماشین قابل تفسیر چیست: به محیط اجتماعی، برنامه یادگیری ماشین خود و مخاطبان هدف توجه کنید. دریافت درست بخش اجتماعی مدل یادگیری ماشین کاملاً به برنامه خاص شما

¹ Robust

بستگی دارد. متخصصانی از علوم انسانی (به عنوان مثال روانشناسان و جامعه شناسان) را پیدا کنید تا به شما کمک کنند.

۴- توضیحات بر موارد غیرعادی تمرکز دارند. مردم برای توضیح رویدادها بیشتر بر علل غیرعادی تمرکز می‌کنند (Kahneman & Tversky, 1981). این موارد، علی‌هستند که احتمال کمی‌داشتند، اما با این وجود اتفاق افتادند. حذف این علل غیرطبیعی نتیجه را تا حد زیادی تغییر می‌داد (توضیح خلاف واقع). انسان‌ها این نوع علل «غیرعادی» را به عنوان توضیحات خوبی در نظر می‌گیرند. مثالی از Štrumbelj and Kononenko (2011) بیان می‌کند: فرض کنید مجموعه‌داده‌ای از موقعیت‌های آزمون بین معلمان و دانش آموزان داریم. دانش آموزان در یک دوره آموزشی شرکت می‌کنند و پس از یک ارائه موفقیت آمیز می‌توانند مستقیماً قبول شوند. معلم این گزینه را دارد از دانش آموزان سوالاتی بپرسد تا دانش آنها را محک بزنند. دانش آموزانی که نتوانند به این سوالات پاسخ دهند، در دوره مردود خواهند شد. دانش آموزان می‌توانند سطوح آمادگی متفاوتی داشته باشند، که به معنی احتمالات متفاوتی برای پاسخ صحیح به سوالات معلم می‌باشد (اگر تصمیم به تست دانش آموز داشته باشند). می‌خواهیم پیش‌بینی کنیم که آیا یک دانش آموز این دوره را قبول می‌شود و پیش‌بینی خود را توضیح دهیم. در صورتی که معلم هیچ سوال اضافی نپرسد، شанс قبولی ۱۰۰٪ است. اگر معلم سوال بپرسد، احتمال قبولی بستگی به سطح آمادگی دانش آموز و احتمال پاسخگویی صحیح به سوالات، دارد.

سناریوی ۱: معلم معمولاً از دانش آموزان سوالات اضافی می‌پرسد (مثلاً ۹۵ بار از ۱۰۰ بار). دانش آموزی که درس نخوانده است (۱۰٪ شанс قبولی در بخش سوال) جزو افراد خوش شانس نبوده و سوالات اضافی دریافت می‌کند و نمی‌تواند به درستی پاسخ دهد. چرا دانش آموز در درس مردود شد؟ می‌گوییم تقصیر دانشجو بود که درس نخواند.

سناریوی ۲: معلم به ندرت سوالات اضافی می‌پرسد (مثلاً ۲ بار از ۱۰۰ بار). برای دانش آموزی که برای سوالات مطالعه نکرده است، احتمال گذراندن دوره را زیاد پیش‌بینی می‌کنیم، زیرا احتمال سؤال پرسیدن خیلی کم است. البته یکی از دانش آموزان برای سوالات آماده نشد که ۱۰ درصد شانس قبولی در سوالات را به او می‌دهم. او بدشانس است و معلم سوالات اضافی می‌پرسد و دانش آموز نمی‌تواند به آنها پاسخ دهد و در درس مردود می‌شود. دلیل رد شدن چیست؟ من استدلال می‌کنم که اکنون، توضیح بهتر این است که "چون معلم دانش آموز را امتحان کرد". بعید بود که معلم امتحان بگیرد، بنابراین معلم رفتار غیرعادی داشت.

معنی آن برای یادگیری ماشین قابل تفسیر چیست: اگر یکی از ویژگی‌های ورودی برای یک پیش‌بینی به هر معنا غیرعادی بود (مانند یک دسته نادر از یک ویژگی طبقه‌بندی شده) و این ویژگی بر پیش‌بینی تأثیر گذاشت، باید در توضیح گنجانده شود، حتی اگر سایر ویژگی‌های «عادی» تاثیر مشابهی با ویژگی غیرعادی بر

پیش‌بینی داشته باشد. یک ویژگی غیرعادی در مثال پیش‌بینی قیمت خانه ما، ممکن است این باشد که یک خانه نسبتاً گران دو بالکن دارد. بهترین توضیح برای گرانی این خانه ارائه ویژگی غیر عادی دو بلکن است، حتی اگر برعی از روش‌ها نشان دهنده که داشتن دو بالکن به همان مقدار قیمت را تغییر می‌دهد که اندازه خانه بالاتر از متوسط، محله خوب یا نوسازی خانه، باعث افزایش قیمت شده است.

۵- توضیحات صادق^۱ است. توضیحات خوب در واقعیت (یعنی در موقعیت‌های دیگر) صادق هستند. اما به طرز نگران‌کننده‌ای، این مهم‌ترین عامل برای توضیح «خوب» نیست. به عنوان مثال، به نظر می‌رسد انتخابی بودن، مهم‌تر از صداقت است. توضیحی که فقط یک یا دو علت احتمالی را انتخاب می‌کند، به ندرت کل فهرست علل مرتبط را پوشش می‌دهد. انتخاب بخشی از حقیقت را حذف می‌کند. این درست نیست که مثلاً فقط، یک یا دو عامل باعث سقوط بورس شده است. در عوض حقیقت این است که میلیون‌ها علت وجود دارد که میلیون‌ها نفر را تحت تأثیر قرار می‌دهد تا به گونه‌ای عمل کنند که در نهایت باعث سقوط بورس شود.

معنی آن برای یادگیری ماشین قابل تفسیر: توضیح باید رویداد را تا حد امکان صادقانه پیش‌بینی کند، که در یادگیری ماشین گاهی اوقات به آن وفاداری می‌گویند. بنابراین اگر بگوییم که بالکن دوم قیمت یک خانه را افزایش می‌دهد، باید برای خانه‌های دیگر (یا حداقل برای خانه‌های مشابه) نیز صدق کند. برای انسان‌ها، وفاداری یک توضیح، به اندازه انتخابی بودن، مقایسه‌ای بودن و جنبه اجتماعی آن مهم نیست.

۶- توضیحات خوب با باورهای قبلی^۲ توضیح دهنده مطابقت دارد. انسان‌ها تمایل دارند اطلاعاتی را نادیده بگیرند که با باورهای قبلی آنها همخوانی ندارد. این اثر سوگیری تایید^۳ نامیده می‌شود (Nickerson, 1998). توضیحات از این نوع سوگیری، در امان نیستند. مردم تمایل دارند توضیحاتی را که با عقاید آنها همخوانی ندارد بی ارزش بدانند یا نادیده بگیرند. مجموعه باورها، از فردی به فرد دیگر، متفاوت است، اما باورهای گروهی مانند جهان‌بینی سیاسی نیز وجود دارد.

معنی آن برای یادگیری ماشین قابل تفسیر چیست: توضیحات خوب با باورهای قبلی سازگار است. ادغام این موضوع، با یادگیری ماشین دشوار است و احتمالاً عملکرد پیش‌بینی را به شدت به خطر می‌اندازد. اعتقاد قبلی ما برای تأثیر اندازه خانه بر قیمت پیش‌بینی شده این است که هر چه خانه بزرگ‌تر باشد، قیمت بالاتر است. اجزاً دهید فرض کنیم که یک مدل اثر منفی اندازه خانه، بر قیمت پیش‌بینی شده را برای چند خانه نشان می‌دهد. مدل این را یاد گرفته است زیرا عملکرد پیش‌بینی را بهبود می‌بخشد (به دلیل برعی از تعاملات پیچیده)، اما این رفتار به شدت با باورهای قبلی ما در تضاد است. می‌توانید محدودیت‌های یکنواختی^۴ (یک

¹ Truthful

² Prior beliefs

³ confirmation bias

⁴ Monotonicity constraint

ویژگی فقط می‌تواند در یک جهت بر پیش‌بینی تأثیر بگذارد) را اعمال کنید، یا از مدلی مانند یک مدل خطی استفاده کنید که این خاصیت را دارد.

۷- **توضیحات خوب کلی و محتمل است.** علتی که بتواند بسیاری از رویدادها را توضیح دهد، توضیح بسیار کلی است و توضیح خوبی محسوب می‌شود. توجه داشته باشید که این موضوع، با این ادعا که علل غیرعادی، توضیحات خوبی هستند، تنافض دارد. همان‌طور که می‌بینم، علل غیرعادی بر علل عمومی‌غلبه می‌کنند. علل غیرعادی بنا به تعریف، نادر هستند. در صورت عدم وجود یک رویداد غیرعادی، یک توضیح کلی توضیح خوبی در نظر گرفته می‌شود. همچنین به یاد داشته باشید که مردم مستعد هستند، احتمالات رویدادهای توأم را اشتباه ارزیابی کنند. (جو یک کتابدار است. آیا او بیشتر یک فرد خجالتی است یا یک فرد خجالتی است که دوست دارد کتاب بخواند؟) یک مثال خوب این است که "خانه گران است چون بزرگ است" که توضیح بسیار کلی و خوب برای گرانی یا ارزانی خانه‌هاست.

معنی آن برای یادگیری ماشین قابل تفسیر چیست: کلیت را به راحتی می‌توان با پشتیبانی^۱ ویژگی اندازه‌گیری کرد، که تعداد نمونه‌هایی است که توضیح برای آن‌ها برقرار است تقسیم بر تعداد کل نمونه‌ها.

¹ Support

فصل ۴ مجموعه داده‌ها

در سراسر کتاب، تمام مدل‌ها و تکنیک‌ها بر روی مجموعه داده‌های واقعی که به‌طور رایگان و آنلاین در دسترس هستند، اعمال می‌شوند. ما از مجموعه داده‌های مختلف برای وظایف مختلف استفاده خواهیم کرد: طبقه‌بندی، رگرسیون و طبقه‌بندی متن.

۴. اجاره دوچرخه (رگرسیون)

این مجموعه‌داده شامل تعداد روزانه دوچرخه‌های کرایه شده از شرکت اجاره دوچرخه^۱ در Capital-Bikeshare در واشنگتن دی سی به همراه اطلاعات آب و هوا و فصلی است. داده‌ها با سخاوت توسط Capital-Bikeshare در دسترس قرار گرفته است. (Fanaee-T and Gama (2014)

هدف این است که پیش‌بینی کنید بسته به آب و هوا و روز چند دوچرخه اجاره می‌شود. داده‌ها را می‌توان از UCI Machine Learning Repository² دانلود کرد.

ویژگی‌های جدیدی به مجموعه‌داده اضافه شد و همه ویژگی‌های اصلی برای مثال‌های این کتاب استفاده نشده است. در اینجا لیستی از ویژگی‌هایی مورد استفاده، آورده شده است:

- تعداد دوچرخه‌های اجاره داده شده به کاربران عادی و ثبت‌نام شده. تعداد دوچرخه‌های اجاره داده شده، به عنوان هدف در وظیفه رگرسیون استفاده می‌شود.
- فصل، شامل بهار، تابستان، پاییز یا زمستان.
- نشان می‌دهد که آیا روز تعطیل بود یا نه.
- سال، ۲۰۱۱ یا ۲۰۱۲.
- تعداد روزهای پس از ۰۰.۱۰.۱۱ (اولین روز در مجموعه‌داده). این ویژگی برای در نظر گرفتن روند در طول زمان اضافه شد.
- نشان می‌دهد که آیا روز یک روز کاری یا آخر هفته بوده است.
- وضعیت آب و هوا در آن روز. یکی از:
 - صاف، کمی ابر، نیمه ابری، ابری
 - مه + ابر، مه + ابرهای کم، مه + چند تکه ابر، مه
 - برف خفیف، باران خفیف + رعدوبرق + ابرهای پراکنده، باران خفیف + ابرهای پراکنده
 - باران شدید + پالت‌های یخ + رعدوبرق + مه، برف + غبار
 - دما بر حسب درجه سانتیگراد.
- رطوبت نسبی بر حسب درصد (۰ تا ۱۰۰).
- سرعت باد بر حسب کیلومتر در ساعت.

برای مثال‌های این کتاب، داده‌ها کمی پردازش شده است. می‌توانید R-script پردازشی را در مخزن GitHub کتاب^۳ به همراه فایل RData نهایی^۱ پیدا کنید.

¹ <https://www.capitalbikeshare.com/>

² <http://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset>

³ <https://github.com/christophM/interpretable-ml-book/blob/master/R/get-bike-sharing-dataset.R>

¹ <https://github.com/christophM/interpretable-ml-book/blob/master/data/bike.RData>

۴.۲ نظرات هرزنامه YouTube (طبقه‌بندی متن)

به عنوان نمونه ای برای طبقه‌بندی متن، ما با ۱۹۵۶ نظر از ۵ ویدیوی مختلف YouTube کار می‌کنیم. خوشبختانه، نویسنده‌گانی که از این مجموعه داده در مقاله‌ای در مورد طبقه‌بندی هرزنامه استفاده کردند، داده‌ها را به صورت رایگان در دسترس قرار دادند (Alberto et al., 2015).

نظرات از طریق API YouTube از پنج ویدیو از ده ویدیوی پر بازدید YouTube در نیمه اول سال ۲۰۱۵ جمع‌آوری شد. هر ۵ مورد، ویدیو موزیک هستند. یکی از آنها Gangnam Style اثر هنرمند کره‌ای Psy است. هنرمندان دیگر Eminem، LMFAO، Katy Perry و Shakira بودند.

برخی از نظرات را برسی کنید. نظرات به صورت دستی به عنوان هرزنامه یا قانونی برچسب گذاری شدند. هرزنامه با "۱" و نظرات قانونی با "۰" کدگذاری شد.

جدول ۴.۱: نمونه نظرات از مجموعه داده‌های هرزنامه YouTube

کلاس	محتوا
۱	Huh, anyway check out this you [tube] channel: kobyoshi02
۱	Hey guys check out my new channel and our first vid THIS IS US THE MONKEYS!!! I'm the monkey in the white shirt,please leave a like comment and please subscribe!!!! just for test I have to say murdev.com
۱	me shaking on my channel enjoy ^_^
۱	watch?v=vtaRGgvGtWQ Check this out .
۱	Hey, check out my new website!! This site is about kids stuff. kidsmediausa . com
۱	Subscribe to my channel
۰	i turned it on mute as soon as i came on i just wanted to check the views...
۱	You should check my channel for Funny VIDEOS!!
۱	and u should.d check my channel and tell me what I should do next!

همچنین می‌توانید به یوتیوب بروید و به بخش نظرات نگاهی بیندازید. اما لطفاً در جهنم یوتیوب گرفتار نشوید و در نهایت به تماشای ویدئوهایی از میمون‌ها که در حال دزدیدن و نوشیدن کوکتل از گردشگران در ساحل هستند، بنشینید. آشکارساز هرزنامه گوگل نیز احتمالاً از سال ۲۰۱۵ تغییرات زیادی کرده است.

ویدیوی رکورددشتی "Gangnam Style" را در اینجا تماشا کنید.

اگر می‌خواهید با داده‌ها بازی کنید، می‌توانید فایل RData را به همراه اسکریپت R با برخی عملکردهای راحت در مخزن GitHub کتاب پیدا کنید.

۴.۳ عوامل خطر برای سرطان دهانه رحم (طبقه‌بندی)

مجموعه داده‌های سرطان دهانه رحم شامل شاخص‌ها و عوامل خطر برای پیش‌بینی اینکه آیا یک زن به سرطان دهانه رحم مبتلا می‌شود یا خیر. این ویژگی‌ها شامل داده‌های جمعیت شناختی^۱ (مانند سن)، سبک زندگی و سابقه پزشکی است. داده‌ها را می‌توان از مخزن یادگیری ماشین^۲ UCI² دانلود کرد و توسط Fernandes et al (۲۰۱۷) ارائه شده است.

زیرمجموعه از ویژگی‌های مجموعه داده که در مثال‌های کتاب استفاده شده‌اند، عبارت‌اند از:

- سن بر حسب سال
- تعداد شرکای جنسی
- اولین رابطه جنسی (سن بر حسب سال)
- تعداد حاملگی‌ها
- سیگار کشیدن بله یا خیر
- سیگار کشیدن (بر حسب سال)
- داروهای ضد بارداری هورمونی بله یا خیر
- داروهای ضد بارداری هورمونی (بر حسب سال)
- دستگاه داخل رحمی بله یا خیر (IUD)
- تعداد سال‌های استفاده از دستگاه داخل رحمی (IUD)
- آیا بیمار تا به حال بیماری مقابله‌یافته (STD) داشته است بله یا خیر
- تعداد تشخیص‌های STD
- زمان از اولین تشخیص STD
- زمان از آخرین تشخیص STD
- نتیجه نمونه برداری "سالم" یا "سرطان" است. خروجی هدف.

نمونه برداری به عنوان استاندارد طلایی برای تشخیص سرطان دهانه رحم عمل می‌کند. برای مثال‌های این کتاب، نتیجه نمونه برداری به عنوان هدف مورد استفاده قرار گرفت. مقادیر گم شده برای هر ستون با حالت (متداول‌ترین مقدار) پر شده‌اند، که احتمالاً راه حل بدی است، زیرا پاسخ واقعی می‌تواند با احتمال گم شدن یک مقدار همبستگی شود. احتمالاً سوگیری وجود دارد زیرا سوالات ماهیت بسیار خصوصی دارند. اما این کتابی در مورد پرکردن داده‌های از گم شده^۳ نیست، بنابراین حالت پرکردن، باید برای مثال‌ها کافی باشد.

¹ Demographic data

² <https://archive.ics.uci.edu/ml/datasets/Cervical+cancer+%28Risk+Factors%29>

³ missing data imputation

برای باز تولید مثال های این کتاب با این مجموعه داده، پیش پردازش R-script¹ و فایل RData نهایی² را در مخزن GitHub کتاب پیدا کنید.

¹ <https://github.com/christophM/interpretable-ml-book/blob/master/R/get-cervical-cancer-dataset.R>

² <https://github.com/christophM/interpretable-ml-book/blob/master/data/cervical.RData>

فصل ۵ مدل‌های قابل تفسیر

ساده‌ترین راه برای دستیابی به تفسیرپذیری، استفاده از زیر مجموعه‌ای از الگوریتم‌هایی است که مدل‌های قابل تفسیر می‌سازند. مدل‌های قابل تفسیر، متداول عبارتند از: رگرسیون خطی، رگرسیون لجستیک و درخت تصمیم.

در بخش‌های بعدی در مورد این مدل‌ها صحبت خواهیم کرد. نه به صورت مفصل بلکه فقط اصول اولیه، زیرا در حال حاضر تعداد زیادی کتاب، فیلم، مقالات و منابع دیگر برای آموزش این مدل‌ها در دسترس است. ما بر نحوه تفسیر مدل‌ها تمرکز خواهیم کرد. این کتاب رگرسیون خطی، رگرسیون لجستیک، دیگر پسوندهای رگرسیون خطی، درختان تصمیم، قوانین تصمیم گیری و الگوریتم RuleFit را با جزئیات بیشتری مورد بحث قرار می‌دهد. همچنین سایر مدل‌های قابل تفسیر را فهرست می‌کند.

تمام مدل‌های تفسیرپذیر توضیح داده شده در این کتاب در سطح مدولار قابل تفسیر هستند، به استثنای روش k-nearest همسایه. جدول زیر یک نمای کلی از انواع مدل‌های قابل تفسیر و خواص آنها ارائه می‌دهد. یک مدل خطی است اگر ارتباط بین ویژگی‌ها و هدف به صورت خطی مدل شود. یک مدل با محدودیت‌های یکنواختی تضمین می‌کند که رابطه بین یک ویژگی و نتیجه هدف همیشه در یک جهت در کل محدوده ویژگی پیش می‌رود: افزایش در مقدار ویژگی یا همیشه منجر به افزایش یا همیشه به کاهش هدف منجر می‌شود. نتیجه یکنواختی برای تفسیر یک مدل مفید است زیرا درک یک رابطه را آسان‌تر می‌کند. برخی از مدل‌ها می‌توانند به طور خودکار شامل تعامل بین ویژگی‌ها برای پیش‌بینی هدف هستند. می‌توانید با ایجاد دستی ویژگی‌های تعاملی، تعاملات را در هر مدلی اضافه کنید. تعاملات می‌توانند عملکرد پیش‌بینی را بهبود بخشدند، اما تعاملات زیاد یا بسیار پیچیده می‌توانند به تفسیرپذیری آسیب برسانند. برخی از مدل‌ها فقط رگرسیون، برخی فقط طبقه‌بندی و برخی دیگر هر دو وظیفه را انجام می‌دهند.

از این جدول، می‌توانید یک مدل قابل تفسیر مناسب برای وظیفه خود (رگرسیون یا طبقه‌بندی) انتخاب کنید.

الگوریتم	خطی	یکنواخت	اثر متقابل	وظیفه
رگرسیون خطی	بله	بله	خیر	رگرسیون
رگرسیون لجستیک	خیر	بله	خیر	طبقه بندی
درختان تصمیم	خیر	مقداری	بله	رگرسیون، طبقه بندی
RuleFit	بله	خیر	بله	رگرسیون، طبقه بندی
Naive Bayes	خیر	بله	خیر	طبقه بندی

k نزدیک ترین همسایگان

طبقه
رگرسیون،
بندی

خیر

خیر

خیر

شما می‌توانید استدلال کنید که هم رگرسیون لجستیک و هم Naive Bayes، توضیحات خطی دارند. اما باید در مظر داشت که خطی بودن، فقط برای لگاریتم هدف، برقرار است: افزایش یک ویژگی به اندازه یک، لگاریتم احتمال هدف، را به میزان معینی افزایش می‌دهد، با فرض ثابت ماندن همه ویژگی‌های دیگر.

۵.۱ رگرسیون خطی

یک مدل رگرسیون خطی، هدف را با استفاده از مجموع وزنی ویژگی های ورودی پیش‌بینی می‌کند. خطی بودن رابطه آموزش داده شده، تفسیر را آسان می‌کند. مدل‌های رگرسیون خطی سال هاست که توسط آماردانان، دانشمندان کامپیوتر و سایر افرادی که به مسائل کمی رسیدگی می‌کنند، استفاده می‌شود.

مدل‌های خطی می‌توانند برای مدل سازی وابستگی یک هدف رگرسیونی y به تعدادی از ویژگی‌های x استفاده شوند. روابط آموزش داده شده، خطی هستند و می‌توان آنها را برای نمونه \hat{y} به صورت زیر نوشت:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \epsilon$$

نتیجه پیش‌بینی شده یک نمونه، مجموع وزنی از p ویژگی آن است. بتاهای (β_i) اوزان یا ضرایب ویژگی‌های آموزش داده شده را نشان می‌دهد. وزن اول (β_0) عرض از مبدأ^۱ نامیده می‌شود و با هیچ یک از ویژگی‌ها ضرب نمی‌شود. اپسیلون (ϵ) خطایی است که پیش‌بینی ما، دارد، یعنی تفاوت بین پیش‌بینی و نتیجه واقعی. فرض بر این است که این خطاهای از یک توزیع گاووسی^۲ پیروی می‌کنند، به این معنی که ما در هر دو جهت منفی و مثبت خطای داریم و تعداد زیادی خطای کوچک و تعداد کمی خطاهای بزرگ وجود دارد.

برای تخمین وزن بهینه، می‌توان از روش‌های مختلفی استفاده کرد. معمولاً از روش حداقل مربعات معمولی^۳، برای یافتن وزن‌هایی استفاده می‌شود، که مجبور اختلاف بین نتایج واقعی و تخمینی را حداقل می‌کند:

$$\beta = \arg \min_{\beta_0, \dots, \beta_p} \sum_{i=1}^n \left(y^{(i)} - \left(\beta_0 + \sum_{j=1}^p \beta_j x_j^{(i)} \right) \right)^2$$

ما در مورد چگونگی یافتن وزن‌های بهینه به تفصیل بحث نخواهیم کرد، اما اگر علاقه‌مند هستید، می‌توانید بخش ۳.۲ کتاب "عناصر یادگیری آماری" (Hastie, 2009) یا یکی دیگر از کتاب‌های آنلاین در مورد مدل‌های رگرسیون خطی را مطالعه کنید.

بزرگ‌ترین مزیت مدل‌های رگرسیون خطی، خطی بودن است: این روش تخمین را ساده می‌کند و مهم‌تر از همه، این معادلات خطی تفسیر آسانی در سطح مدولار (یعنی وزن‌ها) دارند. این موضوع، یکی از دلایل اصلی گسترش استفاده از مدل خطی و همه مدل‌های مشابه در زمینه‌های دانشگاهی مانند پزشکی، جامعه‌شناسی، روان‌شناسی و بسیاری دیگر از زمینه‌های تحقیقاتی کمی است. به عنوان مثال، در زمینه پزشکی، نه تنها پیش‌بینی نتیجه بالینی یک بیمار مهم است، بلکه تعیین کمیت تأثیر دارو و در عین حال در نظر گرفتن جنسیت، سن و سایر ویژگی‌ها به روشی قابل تفسیر، نیز اهمیت دارد.

¹ Intercept

² Gaussian distribution

³ ordinary least squares method

وزن‌های تخمینی با فواصل اطمینان بیان می‌گردد. فاصله اطمینان، محدوده‌ای برای تخمین وزن است که وزن «واقعی» را با اطمینان خاصی دربر می‌گیرد. به عنوان مثال، فاصله اطمینان ۹۵٪ برای وزن ۲ می‌تواند از ۱ تا ۳ متغیر باشد. تفسیر این فاصله به این صورت خواهد بود: اگر تخمین را ۱۰۰ بار با داده‌های نمونه گیری جدید تکرار کنیم، فاصله اطمینان در ۹۵ مورد از ۱۰۰ مورد شامل وزن صحیح می‌شود، به شرط اینکه مدل رگرسیون خطی، مدل درست برای داده‌ها باشد.

اینکه آیا مدل، مدل «درست» است بستگی به این دارد که آیا روابط موجود در داده‌ها مفروضات خاصی را برآورده می‌کنند یا نه. این مفروضات عبارت‌اند از خطی بودن، نرمال بودن، همسانی، استقلال، ویژگی‌های ثابت و عدم وجود چند خطی.

خطی بودن^۱

مدل رگرسیون خطی، پیش‌بینی را مجبور می‌کند که ترکیبی خطی از ویژگی‌ها باشد، که هم بزرگ‌ترین مزیت و هم بزرگ‌ترین محدودیت آن است. خطی بودن منجر به مدل‌های قابل تفسیر می‌شود. اثرات خطی به راحتی کمی و توصیف می‌شوند. این اثرات تجمعی هستند، بنابراین به راحتی می‌توان اثرات را از هم تفکیک کرد. اگر شما مشکوک به اثرات تعاملات ویژگی یا رابطه غیرخطی یک ویژگی هستید، می‌توانید عبارات تعاملی را اضافه کنید یا از اسپیلین‌های رگرسیونی^۲ استفاده کنید.

نرمال بودن^۳

فرض می‌شود که خروجی هدف به شرط ویژگی‌ها، از توزیع نرمال پیروی می‌کند. اگر این فرض نقض شود، فواصل اطمینان تخمینی برای اوزان ویژگی‌ها، نامعتبر است.

هم واریانسی^۴ (واریانس ثابت)

واریانس عبارات خط، در کل فضای ویژگی، ثابت فرض می‌شود. فرض کنید می‌خواهید ارزش یک خانه را به شرط مساحت نشیمن بر حسب متر مربع پیش‌بینی کنید. شما یک مدل خطی را تخمین می‌زنید که فرض می‌کند، صرف نظر از اندازه خانه، خط در اطراف پاسخ پیش‌بینی شده واریانس یکسانی دارد. این فرض اغلب در واقعیت نقض می‌شود. در مثال خانه، احتمالاً واریانس عبارت خط در اطراف قیمت پیش‌بینی شده برای خانه‌های بزرگ‌تر، بیشتر باشد، زیرا قیمت‌ها بالاتر هستند و فضای بیشتری برای نوسانات قیمت وجود دارد. فرض کنید میانگین خط (تفاوت بین قیمت پیش‌بینی شده و واقعی) در مدل رگرسیون خطی شما ۵۰۰۰۰ یورو باشد. اگر

¹ Linearity

² Regression splines

³ Normality

⁴ Homoscedasticity

هم واریانسی بودن را فرض کنید، به این معناست که فرض کردید که میانگین خطای ۵۰۰۰۰ برای خانه‌هایی که ۱ میلیون قیمت دارند و برای خانه‌هایی که فقط ۴۰۰۰۰ قیمت دارند، یکسان است.

مستقل بودن^۱

فرض بر این است که هر نمونه مستقل از نمونه‌های دیگر است. اگر اندازه‌گیری‌های مکرر^۲ انجام دهید، مانند آزمایش‌های خون متعدد برای هر بیمار، نقاط داده مستقل نیستند. برای داده‌های وابسته، به مدل‌های رگرسیون خطی خاص، مانند مدل‌های اثر مختلط^۳ یا GEEs نیاز دارید. اگر از مدل رگرسیون خطی "عادی" استفاده کنید، ممکن است نتیجه گیری اشتباہی از مدل دریافت کنید.

ویژگی‌های ثابت^۴

ویژگی‌های ورودی "ثابت" در نظر گرفته می‌شوند. ثابت به این معنی است که ویژگی‌ها به عنوان "به شرط ثابت" و نه به عنوان متغیرهای آماری در نظر گرفته می‌شوند. این فرض ایجاب می‌کند که ویژگی‌ها، فاقد خطاهای اندازه‌گیری هستند. این یک فرض، عموماً غیر واقعی است. با این حال، بدون این فرض، شما باید مدل‌های خطای اندازه‌گیری بسیار پیچیده‌ای را که خطاهای اندازه‌گیری ویژگی‌های ورودی شما را محاسبه می‌کنند، برازش دهید. و معمولاً شما نمی‌خواهید این کار را انجام دهید.

فقدان هم‌راستایی چندگانه^۵

شما ویژگی‌های قوی همبسته را نمی‌خواهید، زیرا این موضوع، تخمین وزن‌ها را به هم می‌زند. در شرایطی که دو ویژگی به شدت همبستگی داشته باشند، تخمین وزن‌ها مشکل می‌شود، زیرا اثرات ویژگی تجمعی هستند و تخمین این وزن‌ها، غیرقابل تعیین می‌شود، چون مشخص نیست این اثرات به کدام یک از ویژگی‌های همبسته نسبت داده شود.

۵.۱.۱ تفسیر

تفسیر وزن در مدل رگرسیون خطی به نوع ویژگی مربوطه بستگی دارد.

- ویژگی عددی؟ افزایش ویژگی عددی به اندازه یک واحد، نتیجه تخمینی را به اندازه وزن آن ویژگی

تغییر می‌دهد. یک مثال از یک ویژگی عددی اندازه خانه است.

- ویژگی باینری^۶: ویژگی که یکی از دو مقدار ممکن را برای هر نمونه می‌گیرد. به عنوان مثال ویژگی

"خانه همراه با یک باغ" یک ویژگی باینری است. یکی از مقادیر، به عنوان دسته مرجع (در برخی از

¹ Independence

² Repeated measurements

³ Mixed effect models

⁴ Fixed features

⁵ Multicollinearity

⁶ Numerical

زبان‌های برنامه نویسی با صفر کدگذاری شده‌اند) به حساب می‌آید، مانند "بدون باغ". تغییر ویژگی از دسته مرجع به دسته دیگر، نتیجه تخمینی را بر اساس وزن ویژگی تغییر می‌دهد.

- ویژگی طبقه‌بندی با طبقه‌های متعدد^۱: ویژگی با تعداد ثابتی از مقادیر ممکن. به عنوان مثال ویژگی «نوع کف» با طبقه‌های احتمالی «فرش»، «لمینت» و «پارکت» است. یک راه حل برای مواجه با طبقه‌های متعدد، one-hot-encoding است، به این معنی که هر طبقه دارای ستون باینری خاص خود است. برای یک ویژگی طبقه‌بندی با L طبقه، شما فقط به $L-1$ طبقه نیاز دارید، زیرا طبقه L اضافی است (به عنوان مثال وقتی ستون‌های ۱ تا $L-1$ ، همه دارای مقدار صفر برای یک نمونه هستند، می‌دانیم که ویژگی طبقه‌بندی این نمونه در طبقه L قرار می‌گیرد). تفسیر برای هر طبقه مانند تفسیر ویژگی‌های باینری است. برخی از زبان‌ها، مانند R ، به شما امکان می‌دهند تا ویژگی‌های طبقه‌بندی را به روش‌های مختلف رمزگذاری کنید، همان‌طور که در ادامه این بخش توضیح داده شده است.

- عرض از مبدا β_0 : عرض از مبدا، وزن ویژگی برای "ویژگی ثابت" است که همیشه برای همه موارد ۱ است. اکثر پکیج‌های نرم افزاری به طور خودکار یک ویژگی^۲^۱ برای تخمین عرض از مبدا اضافه می‌کنند. تفسیر این است: برای مثال، با تمام مقادیر ویژگی‌های عددی در صفر و مقادیر ویژگی‌های طبقه‌بندی شده در طبقه‌های مرتع، پیش‌بینی مدل، وزن عرض از مبدا است. تفسیر عرض از مبدا معمولاً مهم نیست، زیرا نمونه‌هایی با مقادیر همه ویژگی‌ها در صفر اغلب معنی ندارند. تفسیر عرض از مبدا، تنها زمانی معنادار است که ویژگی‌ها استاندارد شده باشند (میانگین صفر، انحراف معیار یک). در این حالت، عرض از مبدا، نتیجه پیش‌بینی شده نمونه‌ای را منعکس می‌کند که در آن همه ویژگی‌ها در مقدار میانگین خود هستند.

تفسیر ویژگی‌ها در مدل رگرسیون خطی را می‌توان با استفاده از الگوهای متنه زیر خودکار کرد.

تفسیر یک ویژگی عددی

افزایش ویژگی x_k به اندازه یک واحد، پیش‌بینی y را β_k واحد افزایش می‌دهد، زمانی که تمام مقادیر ویژگی‌های دیگر ثابت باقی بمانند.

تفسیر یک ویژگی طبقه‌بندی شده

تغییر ویژگی x_k از طبقه مرجع به طبقه دیگر، پیش‌بینی y را β_k واحد افزایش می‌دهد زمانی که تمام ویژگی‌های دیگر ثابت بمانند.

¹ Binary

² Categorical feature with multiple categories

اندازه گیری مهم دیگر برای تفسیر مدل‌های خطی، اندازه گیری R-squared است. به شما می‌گوید که چه مقدار از واریانس کل نتیجه هدف شما توسط مدل توضیح داده شده است. هرچه R-squared بالاتر باشد، مدل شما، داده‌ها را بهتر توضیح می‌دهد. فرمول محاسبه R-squared به صورت زیر است:

$$R^2 = 1 - \frac{SSE}{SST}$$

SSE مجموع مجذورات خط‌^۱ است:

$$SSE = \sum_{i=1}^n (y^{(i)} - \hat{y}^{(i)})^2$$

SST مجموع مجذورات واریانس^۲ داده است:

$$SST = \sum_{i=1}^n (y^{(i)} - \bar{y})^2$$

SSE به شما می‌گوید که پس از برازش مدل خطی، چقدر واریانس باقی می‌ماند. این واریانس با مجذور اختلافات بین مقادیر هدف پیش‌بینی شده و واقعی اندازه گیری می‌شود. SST، واریانس کل خروجی هدف است. R-squared به شما می‌گوید که چه مقدار از واریانس شما را می‌توان با مدل خطی توضیح داد. عموماً بین صفر برای مدل‌هایی که مدل، اصلًاً داده‌ها را توضیح نمی‌دهد و ۱ برای مدل‌هایی که تمام واریانس داده‌های شما را مدل توضیح می‌دهند، متغیر است. همچنین ممکن است R-squared، بدون نقض قوانین ریاضی، یک مقدار منفی به خود بگیرد. این حالت زمانی اتفاق می‌افتد که SSE بزرگ‌تر از SST باشد، به این معنی که یک مدل نمی‌تواند، روند داده‌ها را درک کند و استفاده از آن برای پیش‌بینی بدتر از استفاده از مقدار ثابت میانگین هدف، است.

R-squared با افزایش تعداد ویژگی‌های مدل افزایش می‌یابد، حتی اگر اصلًاً حاوی اطلاعاتی در مورد مقدار هدف نباشند. بنابراین، بهتر است از R-squared تعديل شده^۳ استفاده کنید که تعداد ویژگی‌های استفاده شده در مدل را به حساب می‌آورد. محاسبه آن به صورت زیر است:

$$R^2 = 1 - (1 - R^2) \frac{n - 1}{n - p - 1}$$

که در آن p تعداد ویژگی‌ها و n تعداد نمونه‌ها است.

تفسیر یک مدل با R-squared (یا تعديل شده) بسیار کم، بی معنی است، زیرا چنین مدلی اساساً واریانس زیادی را توضیح نمی‌دهد و درنتیجه هر گونه تفسیری از اوزان معنادار نخواهد بود.

اهمیت ویژگی

¹ Squared sum of the error

² Squared sum of variance

³ Adjusted R-squared

اهمیت یک ویژگی در مدل رگرسیون خطی را می‌توان با قدر مطلق آماره t^1 اندازه گیری کرد. آماره t ، وزن تخمینی مقیاس شده با خطای استاندارد آن است.

$$t_{\beta_j} = \frac{\hat{\beta}_j}{SE(\hat{\beta}_j)}$$

اجازه دهید بررسی کنیم که این فرمول به ما چه می‌گوید: اهمیت یک ویژگی با افزایش وزن افزایش می‌یابد. این منطقی است. هر چه وزن تخمینی، واریانس بیشتری داشته باشد ($=$ هر چه نسبت به مقدار صحیح اطمینان کمتری داشته باشیم)، اهمیت ویژگی کمتر است. این نیز منطقی است.

مثال ۵.۱.۲

در این مثال، ما از مدل رگرسیون خطی برای پیش‌بینی تعداد دوچرخه‌های اجاره‌ای در یک روز خاص، به شرط اطلاعات آب و هوا و تقویم استفاده می‌کنیم. برای تفسیر، وزن‌های رگرسیون تخمینی را بررسی می‌کنیم. ویژگی‌ها شامل ویژگی‌های عددی و طبقه‌ای است. برای هر ویژگی، جدول وزن تخمینی، خطای استاندارد تخمین (SE) و قدر مطلق آماره $t(|t|)$ را نشان می‌دهد.

	Wei ght	SE	$ t $
(Intercept)	2399 .4	23 8.3	1 0.1
seasonSPRING	899. 3	12 2.3	7. 4
seasonSUMMER	138. 2	16 1.7	0. 9
seasonFALL	425. 6	11 0.8	3. 8
holidayHOLIDAY	- 686.1	20 3.3	3. 4
workingdayWORKING DAY	124. 9	73. 3	1. 7
weathersitMISTY	- 379.4	87. 6	4. 3
weathersitRAIN/SNOW/STO	-	22	8.
RM	1901.5	3.6	5
temp	110. 7	7 5.7	1
hum	-17.4	3.2	5. 5
windspeed	-42.5	6.9	6. 2
days_since_2011	4.9	0.2	2

¹ T-statistics

تفسیر یک ویژگی عددی (دما): افزایش دما به میزان ۱ درجه سانتیگراد، تعداد پیش‌بینی شده دوچرخه‌ها را تا به اندازه ۱۱۰.۷ افزایش می‌دهد، زمانی که سایر ویژگی‌ها ثابت می‌مانند.

تفسیر یک ویژگی طبقه‌بندی شده ("Weathersit"): تعداد تخمینی دوچرخه‌ها در هنگام باران، برف یا طوفان ۱۹۰.۵ کمتر از آب و هوای خوب است و دوباره با فرض اینکه همه ویژگی‌های دیگر تغییر نمی‌کنند. وقتی هوا مه آلود است، به شرط ثابت ماندن سایر ویژگی‌ها، تعداد دوچرخه‌های پیش‌بینی شده ۳۷۹.۴ در مقایسه با هوای خوب، کمتر است.

همه تفاسیر همیشه با ذکر این نکته همراه می‌شوند که "همه ویژگی‌های دیگر ثابت می‌مانند". این به دلیل ماهیت مدل‌های رگرسیون خطی است. هدف پیش‌بینی شده، ترکیبی خطی از ویژگی‌های وزنی است. معادله خطی برآورده شده یک ابر صفحه در فضای ویژگی/هدف است (یک خط در حالت یک ویژگی واحد). وزن‌ها شبیه (گرادیان) ابر صفحه را در هر جهت مشخص می‌کنند. جنبه خوب این موضوع است که تجمعی بودن، تفسیر یک ویژگی مشخص را از سایر ویژگی‌ها محافظت می‌کند. این امکان‌پذیر است این امر به این دلیل است که تمام اثرات ویژگی (= وزن ضربدر مقدار ویژگی) در معادله با یک علامت جمع، ترکیب می‌شوند. جنبه بد این موضوع، تفسیر توزیع توام ویژگی‌ها را در نظر نمی‌گیرد. افزایش یک ویژگی، بدون تغییر ویژگی دیگر، می‌تواند منجر به نقاط داده غیر واقعی یا حداقل بعيد شود. به عنوان مثال افزایش تعداد اتاق‌ها بدون افزایش اندازه خانه ممکن است غیرواقعی باشد.

۵.۱.۳ تفسیر بصری^۱

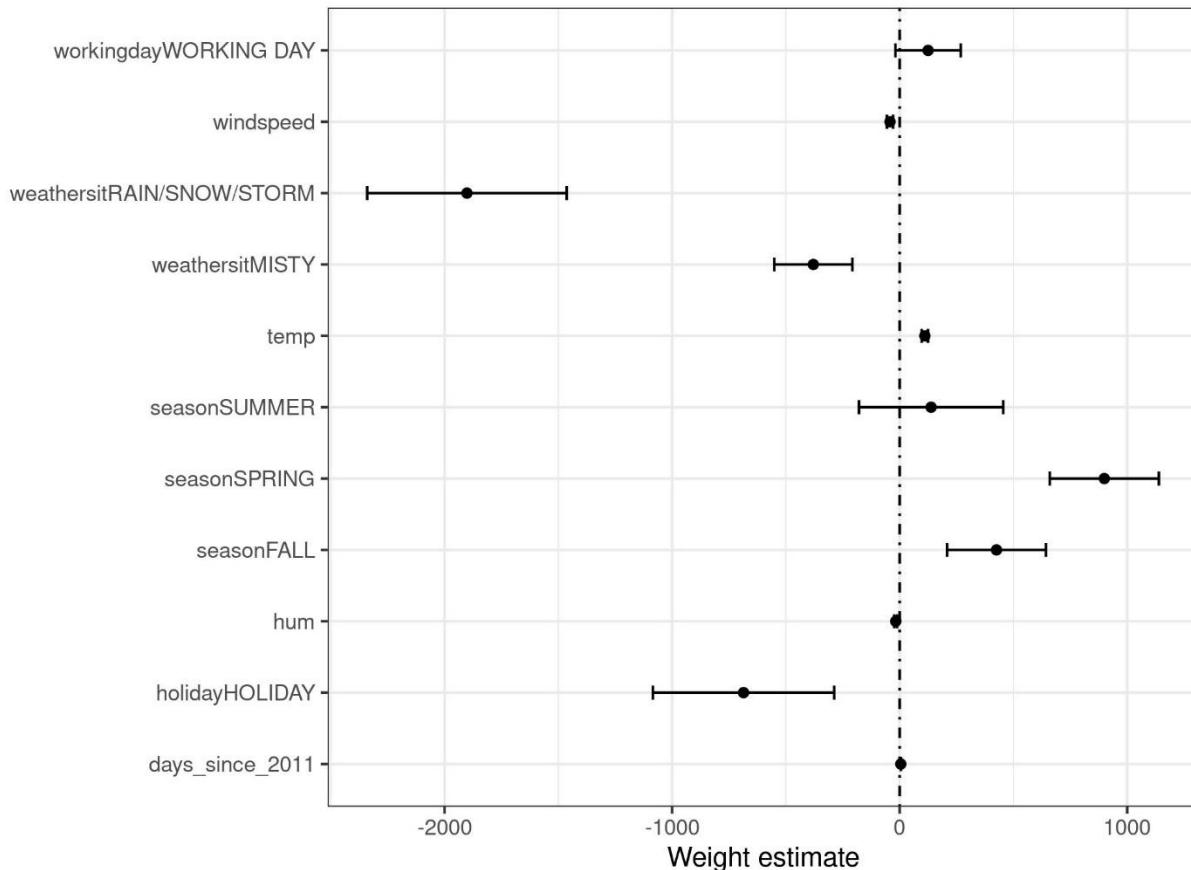
بصری سازی‌های مختلف، مدل رگرسیون خطی را برای انسان آسان و درک آن را سریع می‌کند.

۵.۱.۳.۱ نمودار وزن^۲

اطلاعات جدول وزنی (تخمین وزن و واریانس) را می‌توان در نمودار وزنی بصری نمود. نمودار زیر نتایج حاصل از مدل رگرسیون خطی قبلی را نشان می‌دهد.

¹ Visual interpretation

² Weight plot



شکل ۵.۱: وزن‌ها به صورت نقاط و فاصله‌های اطمینان ۹۵ درصد به صورت خطوط نمایش داده شده‌اند. نمودار وزن نشان می‌دهد که هوای بارانی/برفی/طوفانی (rainy/snowy/stormy) تأثیر منفی قوی بر تعداد پیش‌بینی شده دوچرخه دارد. وزن ویژگی روز کاری (working day) نزدیک به صفر است و صفر در بازه ۹۵٪ قرار دارد، که به این معنی است که اثر از نظر آماری معنی دار نیست.^۱ برخی از فواصل اطمینان بسیار کوچک و تخمین‌ها نزدیک به صفر هستند، با این حال اثرات ویژگی از نظر آماری معنی‌دار است. دما یکی از این ویژگی‌ها است. مشکل نمودار وزن این است که ویژگی‌ها در مقیاس‌های مختلف اندازه گیری می‌شوند. در حالی که برای آب و هوای وزن تخمینی تفاوت بین هوای خوب و بارانی/برفی/طوفانی را نشان می‌دهد، برای دما فقط افزایش ۱ درجه سانتی‌گراد را نشان می‌دهد. قبل از برآش مدل خطی، می‌توانید وزن‌های تخمینی را با مقیاس بندی ویژگی‌ها (میانگین صفر و انحراف استاندارد یک) قابل مقایسه تر کنید.

^۱ not statistically significant

۵.۱.۳.۲ نمودار اثر^۱

وقتی وزن‌های مدل رگرسیون خطی زمانی در مقادیر واقعی ضرب شوند می‌توانند به طور معنی دارتری تحلیل شوند. وزن‌ها به مقیاس ویژگی‌ها بستگی دارند. مثلاً فرض کنید که ویژگی‌ای دارید که قد یک فرد را اندازه‌گیری می‌کند و واحد اندازه‌گیری را از متر به سانتی‌متر تغییر می‌دهید. با این تغییر اندازه‌گیری، وزن‌ها تغییر خواهند کرد، اما اثرات واقعی در داده‌های شما تغییر نخواهد کرد. از سوی دیگر، دانستن توزیع ویژگی در داده‌ها مهم است، زیرا اگر واریانس بسیار پایینی دارید، به این معنی است که تقریباً همه نمونه‌ها سهم مشابهی از این ویژگی دارند. نمودار اثر می‌تواند به شما کمک کند تا بهفهمید که ترکیب وزن و ویژگی، چقدر به پیش‌بینی‌ها کمک می‌کند. با محاسبه اثرات شروع کنید، که وزن‌های ویژگی ضربر مقدار ویژگی یک نمونه است:

$$effect_j^{(i)} = w_j x_j^{(i)}$$

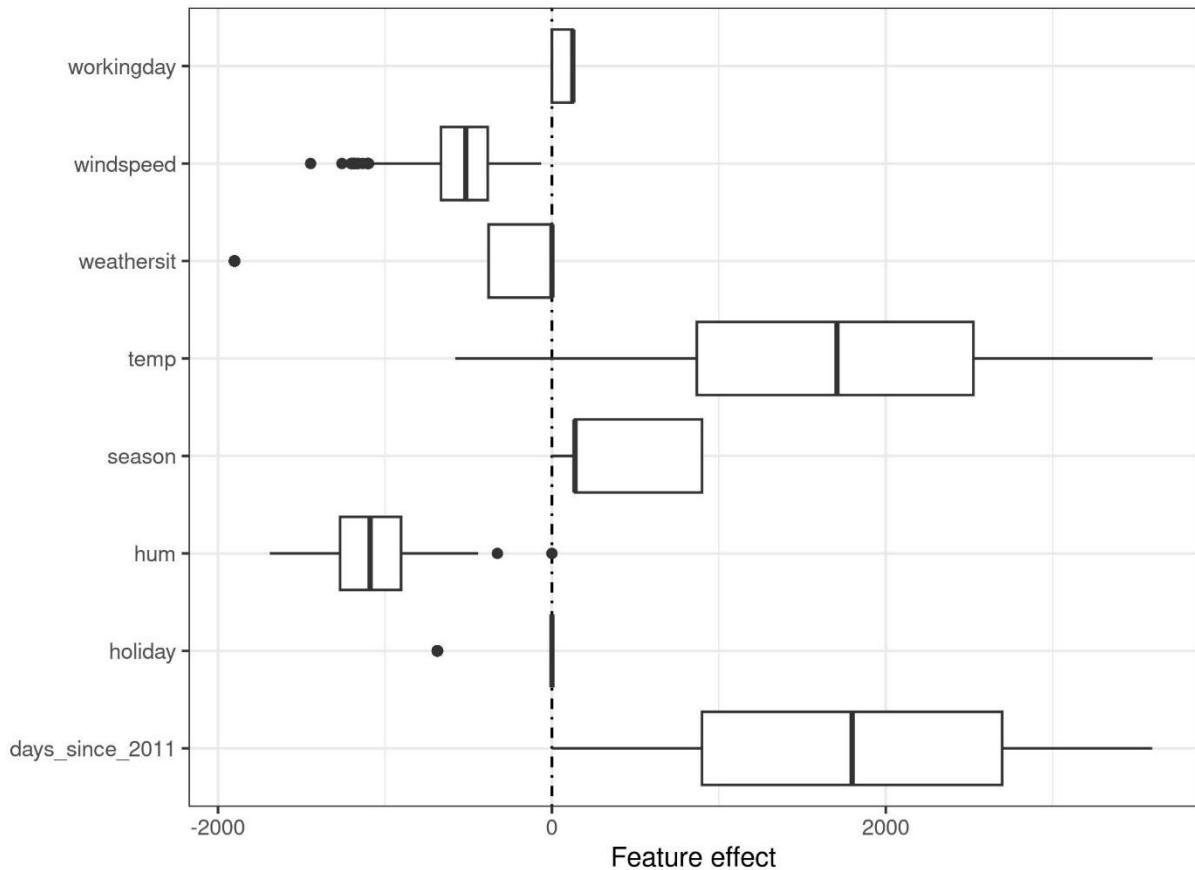
اثرات را می‌توان با نمودارهای باکس‌پلات مستطیلی بصری کرد. مستطیل در یک باکس‌پلات شامل محدوده اثر برای نیمی از داده‌ها (چندک‌های اثر ۰٪ تا ۷۵٪). خط عمودی در مستطیل، اثر میانه است، یعنی ۵۰ درصد از نمونه‌ها تأثیر کمتر و نیمی‌دیگر، تأثیر بیشتری بر پیش‌بینی دارند. نقاط پرت با این تعریف مشخص می‌شوند: نقاطی که بیش از ۱.۵ برابر دامنه بین چارکی^۲ (یعنی تفاوت بین ربع اول و سوم) بالای ربع سوم، یا کمتر از ۱.۵ برابر دامنه بین چارکی زیر چارک اول، قرار دارند. دو خط افقی که جاروبهای^۳ پایینی و بالایی نامیده می‌شوند، نقاط زیر چارک اول و بالای چارک سوم را که پرت نیستند به هم متصل می‌کنند. اگر نقاط پرت وجود نداشته باشد، جاروبهای تا مقادیر حداقل و حداکثر گسترش می‌یابند.

اثرات تمام طبقه‌های یک ویژگی‌ها طبقه‌ای را می‌توان در یک باکس‌پلات با یکدیگر خلاصه کرد، برخلاف نمودار وزن، که در آن هر طبقه ردیف خاص خود را نیاز داشت.

¹ Effect plot

² interquartile range

³ whiskers



شکل ۵.۲: نمودار اثر ویژگی که توزیع اثرات (= مقدار ویژگی ضربدر وزن ویژگی) را در بین دادهها در هر ویژگی نشان می‌دهد.

بیشترین سهم در تعداد مورد انتظار دوچرخه‌های اجاره‌ای، مربوط به ویژگی دما و ویژگی روزها (ویژگی روزها، روند اجاره دوچرخه را در طول زمان نشان می‌دهد) است. دما، دامنه وسیعی از مشارکت در پیش‌بینی دارد. ویژگی روند روز، از صفر به مقادیر مثبت بزرگ رو به افزایش است، زیرا اولین روز در مجموعه‌داده (۰۱۰.۱۰۱۱) تأثیر روند بسیار کمی دارد و وزن تخمینی برای این ویژگی مثبت است (۴.۹۳). این به این معنی است که اثر با هر افزایش هر روز بیشتر می‌شود و برای آخرین روز در مجموعه‌داده (۳۱.۱۲۰۱۲) دارای بالاترین میزان است. توجه داشته باشید که برای اثراتی با وزن منفی، نمونه‌هایی با اثر مثبت آنهاست هستند که دارای ارزش ویژگی منفی هستند. به عنوان مثال، روزهایی که سرعت باد دارای اثر منفی زیاد است، روزهایی هستند که سرعت باد زیاد است.

۵.۱.۴ پیش‌بینی‌های فردی را توضیح دهید

هر یک از ویژگی‌های یک نمونه چقدر در پیش‌بینی مشارکت کرده است؟ این مقدار را می‌توان با محاسبه اثرات برای این نمونه پاسخ داد. تفسیر اثرات خاص نمونه فقط در مقایسه با توزیع اثر برای هر ویژگی منطقی است.

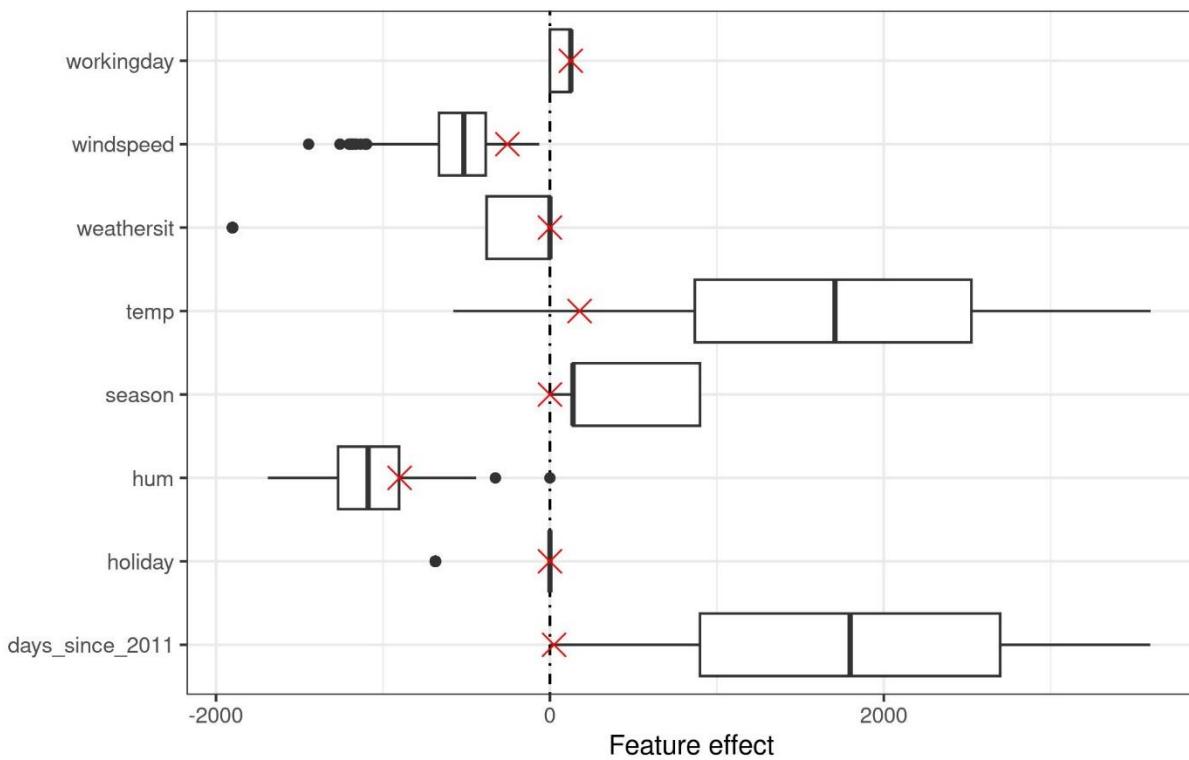
می خواهیم پیش بینی مدل خطی را برای نمونه ششم از مجموعه داده دوچرخه توضیح دهیم. نمونه دارای مقادیر ویژگی زیر است.

جدول ۵.۱: مقادیر ویژگی برای نمونه ۶

Feature	Value
season	WINTER
yr	2011
mnth	JAN
holiday	NO HOLIDAY
weekday	THU
workingday	WORKING DAY
weathersit	GOOD
temp	1.604356
hum	51.8261
windspeed	6.000868
cnt	1606
days_since_2	5
011	

برای به دست آوردن اثرات ویژگی این نمونه، باید مقادیر ویژگی آن را در وزن های آن ویژگی در مدل رگرسیون خطی ضرب کنیم. ویژگی "روز کاری" مقدار "روز کاری" دارد و اثرش ۱۲۴.۹ است. برای دمای ۱.۶ درجه سانتیگراد، اثر ۱۷۷.۶ است. ما این اثرات را به صورت علامت ضربدر به نمودار اثر اضافه می کنیم، که توزیع اثرات در داده ها را به ما نشان می دهد. این کار به ما اجازه می دهد تا اثرات فردی را با توزیع اثرات در داده ها مقایسه کنیم.

Predicted value for instance: 1571
 Average predicted value: 4504
 Actual value: 1606



شکل ۵.۳: نمودار اثر برای یک نمونه، نمودار توزیع اثر را نشان می دهد و اثرات نمونه مورد نظر را برجسته می کند.

اگر ما از پیش‌بینی نمونه‌های داده‌های آموزشی میانگین بگیریم، عدد ۴۵۰۴ به دست می‌آید. در مقایسه با این میانگین، پیش‌بینی نمونه ششم کوچک است، زیرا تنها ۱۵۷۱ کرايه دوچرخه پیش‌بینی شده است. نمودار اثر دلیل آن را نشان می‌دهد. باکس پلات‌ها، اثرات را برای همه نمونه‌های مجموعه‌داده نشان می‌دهند، علامت‌های ضربدر اثرات را برای نمونه ۶ نشان می‌دهند. نمونه ششم تأثیر دمای پایینی دارد زیرا در این روز دما ۲ درجه بوده است که در مقایسه با اکثر روزهای دیگر پایین است (و به یاد داشته باشید که وزن ویژگی دما مثبت است). همچنین تأثیر ویژگی روند «days_since_2011» در مقایسه با سایر نمونه‌های مجموعه داده کم است، زیرا این نمونه مربوط به اوایل سال ۲۰۱۱ (روز ۵) است و ویژگی روند نیز وزنی مثبتی دارد.

۵.۱.۵ رمزگذاری^۱ ویژگی‌های طبقه‌بندی

راههای مختلفی برای رمزگذاری یک ویژگی طبقه‌بندی وجود دارد و انتخاب هر حالت، بر تفسیر وزن‌ها موثر است.

¹ Encoding

استاندارد در مدل‌های رگرسیون خطی، کدگذاری تیمار است که در اکثر موارد، مناسب است. استفاده از رمزگذاری‌های، یک ستون یکسان ویژگی طبقه بندی را به تبدیل به ماتریس‌های مختلف (طراحی) از می‌کند. در این بخش، سه کدگذاری مختلف ارائه می‌شود، اما موارد بیشتری وجود دارد. مثال مورد استفاده دارای شش نمونه و یک ویژگی طبقه‌بندی شده با سه طبقه است. برای دو نمونه اول، ویژگی طبقه A دارد، برای نمونه سه و چهار، طبقه B، و برای دو مورد آخر، طبقه C.

کدگذاری تیمار^۱

در کدگذاری تیمار، وزن هر طبقه، تفاوت تخمینی پیش‌بینی، بین طبقه مربوطه و طبقه مرجع است. عرض از مبدا مدل خطی، میانگین دسته مرجع است (زمانی که سایر ویژگی‌ها ثابت می‌مانند). ستون اول ماتریس طراحی، عرض از مبدا است که همیشه ۱ است. ستون دو نشان می‌دهد که آیا نمونه i در طبقه B قرار دارد یا خیر، ستون سه نشان می‌دهد که آیا در طبقه C قرار دارد یا خیر. برای دسته A نیازی به ستون اضافی نیست، زیرا در این حالت، معادله خطی بیش از حد مقید^۲ می‌شود و هیچ راه حل منحصر به فردی برای وزن‌ها نمی‌توان یافت. کافی است بدانیم که یک نمونه در طبقه B یا C نیست.

ماتریس ویژگی:

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

کدگذاری اثر^۳

وزن هر طبقه، تفاوت تخمینی y ، از طبقه مربوطه نسبت به میانگین کلی (به شرطی که همه ویژگی‌های دیگر صفر یا دسته مرجع هستند) می‌باشد. ستون اول برای تخمین عرض از مبدا، استفاده می‌شود. وزن β_0 عرض از مبدا، نشان دهنده میانگین کلی و β_1 ، وزن طبقه دو، تفاوت بین میانگین کلی و دسته B است. اثر کل دسته B برابر با $\beta_1 + \beta_0$ می‌باشد. تفسیر دسته C مشابه است. برای رده مرجع A، مقدار $(\beta_1 + \beta_2)$ ، تفاوت میانگین کلی و $(\beta_1 + \beta_2) - \beta_0$ می‌باشد.

ماتریس ویژگی:

¹ Treatment

² Overspecified

³ Effect

$$\begin{pmatrix} 1 & -1 & -1 \\ 1 & -1 & -1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

رمزگذاری مجازی^۱

β هر طبقه، مقدار میانگین تخمینی \bar{x} برای هر طبقه است (به شرطی که همه ویژگی‌های دیگر صفر یا دسته مرجع هستند). توجه داشته باشید که عرض از مبدا در اینجا حذف شده است تا بتوان یک جواب منحصر به فرد برای وزن‌های مدل خطی پیدا کرد. راه دیگر برای غلبه بر این مشکل هم راستایی چندگانه، کنار گذاشتن یکی از طبقه هاست.

ماتریس ویژگی:

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

اگر می‌خواهید کمی عمیق‌تر در رمزگذاری‌های مختلف برای ویژگی‌های طبقه‌بندی شده مطالعه کنید، این صفحه وب (<https://stats.oarc.ucla.edu/r/library/r-library-contrast-coding-systems-for-categorical-variables/>) و این پست وبلاگ (<http://heidiseibold.github.io/page7/>) را بررسی کنید.

۵.۱.۶ آیا مدل‌های خطی توضیحات خوبی ایجاد می‌کنند؟

با قضاوت بر اساس شرایط توضیح خوب، همان‌طور که در فصل توضیحات انسان پسند ارائه شده است، مدل‌های خطی بهترین توضیحات را ایجاد نمی‌کنند. مدل‌های خطی مقابله‌ای هستند، اما نمونه مرجع، یک نقطه داده است که در آن همه ویژگی‌های عددی صفر هستند و ویژگی‌های طبقه‌بندی در دسته‌های مرجع خود قرار دارند. این نمونف، معمولاً یک نمونه مصنوعی و بی معنی است که بعيد است در داده‌های شما یا واقعیت رخ دهد. یک استثنا وجود دارد: اگر همه ویژگی‌های عددی در مرکز میانگین باشند (ویژگی منهای میانگین ویژگی) و همه ویژگی‌های طبقه‌بندی با رمزگذاری اثر، بیان شده باشند، نمونه مرجع نقطه داده‌ای است که در آن همه ویژگی‌ها مقدار میانگین ویژگی را دارا می‌باشند. در این حالت نیز ممکن است یک این نمونه، وجود نداشته باشد، اما حداقل، محتمل‌تر یا معنادارتر است. در این حالت، وزن‌ها ضربدر مقادیر ویژگی (اثرات ویژگی) سهم تویزگی را در نتیجه پیش‌بینی‌شده در مقایسه با «نمونه میانگین» توضیح می‌دهند. یکی دیگر از جنبه‌های توضیح خوب، انتخاب پذیری است که در مدل‌های خطی با استفاده از ویژگی‌های کمتر یا با آموزش مدل‌های

¹ Dummy

خطی محدود می‌توان به آن دست یافت. اما به طور پیش فرض، مدل‌های خطی توضیحات انتخابی ایجاد نمی‌کنند. مدل‌های خطی توضیحات صادق ایجاد می‌کنند، تا زمانی که معادله خطی، مدل مناسبی برای رابطه بین ویژگی‌ها و خروجی باشد. هر چه رفتارهای غیر خطی و تعاملات بیشتر باشد، دقت مدل خطی کمتر خواهد بود و توضیحات کمتر صادق می‌باشند. خطی بودن توضیحات را کلی‌تر و ساده‌تر می‌کند. من معتقدم ماهیت خطی مدل، عامل اصلی استفاده از مدل‌های خطی برای توضیح روابط است.

۵.۱.۷ مدل‌های خطی محدود^۱

مثالی که در این بخش آورده شده است، تعداد ویژگی‌های کمی دارد. اما در واقعیت ممکن است شما فقط چند ویژگی نداشته باشید، بلکه صدھا یا هزاران ویژگی وجود داشته باشند. در این حالت مدل‌های رگرسیون خطی چه طور رفتار خواهند کرد؟ مطمئناً تفسیرپذیری به دچار مشکلات جدی خواهد شد. حتی ممکن است در مساله‌ای، تعداد ویژگی‌ها از تعداد نمونه‌ها بیشتر باشد و نتوانید یک مدل خطی استاندارد برازش دهید. خبر خوب این است که راههایی برای معرفی محدودیت (= تعداد کمی ویژگی) در مدل‌های خطی وجود دارد.

Lasso ۵.۱.۷.۱

یک راه خودکار و راحت برای اعمال محدودیت به مدل رگرسیون خطی است. Lasso (Least Absolute Shrinkage and Selection Operator) است و هنگامی که در یک مدل رگرسیون خطی اعمال می‌شود، وظیفه انتخاب ویژگی و تنظیم وزن ویژگی‌های انتخاب شده را دارد. اجازه دهید مساله کمینه‌سازی، که وزن‌ها را بهینه می‌کنند در نظر بگیریم:

$$\min_{\beta} \left(\frac{1}{n} \sum_{i=1}^n (y^{(i)} - x_i^T \beta)^2 \right)$$

یک عبارت به این مسئله بهینه سازی اضافه می‌کند.

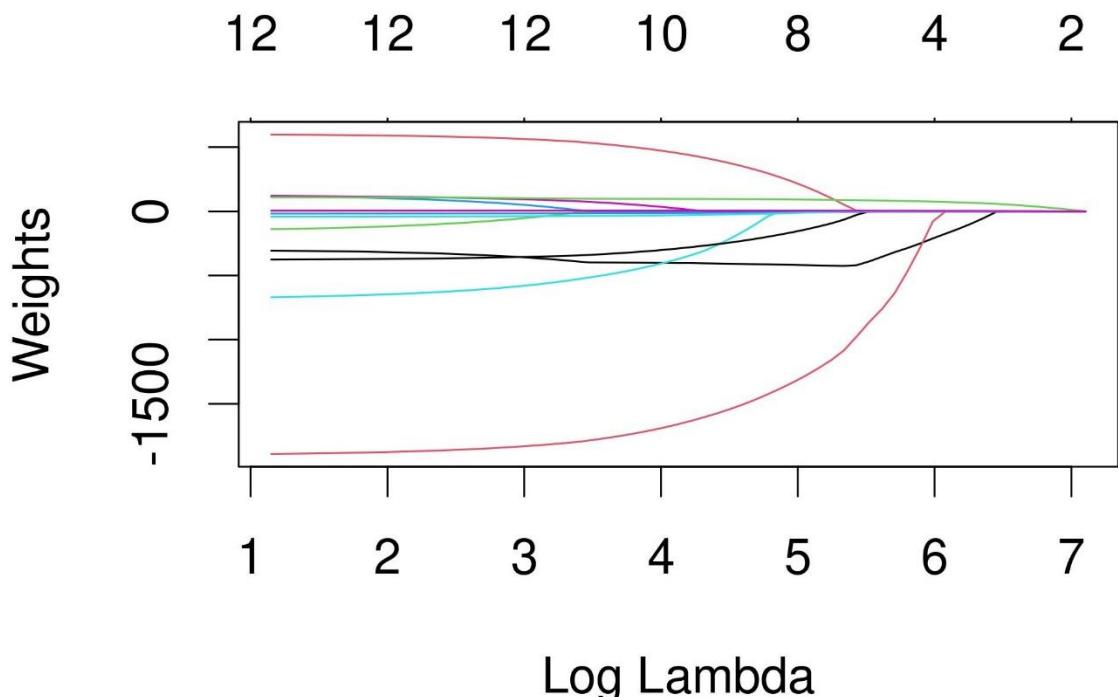
$$\min_{\beta} \left(\frac{1}{n} \sum_{i=1}^n (y^{(i)} - x_i^T \beta)^2 + \lambda \|\beta\|_1 \right)$$

عبارت $\|\beta\|_1$ ، نرم^۲ L1 بودار ویژگی است و منجر به جریمه وزن‌های بزرگ می‌شود. از آنجایی که ازنرم L1 استفاده شده است، تخمین بسیاری از وزن‌ها صفر می‌شود و بقیه کوچک می‌شوند. پارامتر لامبدا (λ) قدرت اثر تنظیم را کنترل می‌کند و معمولاً با اعتبارسنجی متقطع^۳ تنظیم می‌شود. هرچه لامبدا بزرگ‌تر باشد، بسیاری از وزن‌ها، مقدار صفر دریافت می‌کنند. وزن هر ویژگی با یک منحنی در شکل زیر نشان داده شده است.

¹ Spare linear model

² Norm

³ Cross-validation



شکل ۴.۵: با افزایش جریمه وزن‌ها، ویژگی‌های کمتر و کمتری تخمین وزن غیر صفر دریافت می‌کنند. به این منحنی‌ها مسیرهای تنظیم نیز می‌گویند. عدد بالای نمودار تعداد وزن‌های غیر صفر است. چه مقداری را برای لامبدا انتخاب کنیم؟ اگر عبارت جریمه را به عنوان یک پارامتر تنظیم در نظر بگیرید، می‌توانید با اعتبارسنجی متقطع لامبادایی را پیدا کنید که خطای مدل را به حداقل می‌رساند. همچنین می‌توانید لامبدا را به عنوان پارامتری برای کنترل تفسیرپذیری مدل در نظر بگیرید. هر چه جریمه بزرگ‌تر باشد، ویژگی‌های کمتری در مدل وجود دارد (زیرا مابقی وزن صفر دارند) و بهتر می‌توان مدل را تفسیر کرد.

مثال با Lasso

مثال اجاره دوچرخه را با استفاده از Lasso پیش‌بینی می‌کنیم. تعداد ویژگی‌هایی که می‌خواهیم در مدل داشته باشیم را از قبل تعیین می‌کنیم. اجازه دهید ابتدا عدد را روی ۲ ویژگی تنظیم کنیم:

	Weight
seasonWINTER	0
seasonSPRING	0
seasonSUMMER	0
seasonFALL	0
holidayHOLIDAY	0
workingdayWORKING DAY	0

weathersitMISTY	0
weathersitRAIN/SNOW/STORM	0
temp	52.33
hum	0
windspeed	0
days_since_2011	2.15

دو ویژگی اول با وزن‌های غیرصفر در مسیر Lasso، دما ("temp") و روند زمانی ("days_since_2011") هستند.

اگر ۵ ویژگی انتخاب کنیم:

	Weight
seasonWINTER	-389.99
seasonSPRING	0
seasonSUMMER	0
seasonFALL	0
holidayHOLIDAY	0
workingdayWORKING DAY	0
weathersitMISTY	0
weathersitRAIN/SNOW/STORM	-862.27
temp	85.58
hum	-3.04
windspeed	0
days_since_2011	3.82

توجه داشته باشید که وزن‌های "temp" و "days_since_2011" با وزن‌های مدل با دو ویژگی متفاوت است. دلیل این امر این است که با کاهش لامبда، حتی ویژگی‌هایی که قبلاً در مدل هستند، کمتر جرمیه می‌شوند و ممکن است وزن مطلق بیشتری به دست آورند. تفسیر وزن‌های Lasso با تفسیر وزن‌ها در مدل رگرسیون خطی مطابقت دارد. فقط باید به استاندارد بودن یا نبودن ویژگی‌ها توجه کنید، زیرا این موضوع روی وزن‌ها تأثیر می‌گذارد. در این مثال، ویژگی‌ها توسط نرم‌افزار، استاندارد شده بودند. اما وزن‌ها به طور خودکار به برگشت داده شدند تا با مقیاس‌های ویژگی اصلی تطابق داشته باشند.

روش‌های دیگر برای محدودیت در مدل‌های خطی

طیف گسترده‌ای از روش‌ها را می‌توان برای کاهش تعداد ویژگی‌ها، در یک مدل خطی استفاده کرد.

روش‌های پیش پردازش^۱:

¹ Pre-processing methods

- انتخاب ویژگی‌های به صورت دستی: همیشه می‌توانید از دانش متخصصان، برای انتخاب یا حذف برخی از ویژگی‌ها استفاده کنید. اشکال بزرگ این است که نمی‌توان آن را خودکار کرد و شما باید به کسی دسترسی داشته باشید که داده‌ها را درک کند.
 - انتخاب تک متغیره^۱: یک مثال ضریب همبستگی^۲ است. شما فقط ویژگی‌هایی را در نظر می‌گیرید که از آستانه مشخصی از همبستگی بین ویژگی و هدف فراتر می‌روند. نقطه ضعف روش، این است که ویژگی‌ها را فقط به صورت جداگانه در نظر می‌گیرد. برخی از ویژگی‌ها ممکن است تا زمانی که مدل خطی برخی ویژگی‌های دیگر را در نظر نگرفته باشد، همبستگی نشان ندهند. در این وضعیت، این ویژگی‌ها را با روش‌های انتخاب تک متغیره از دست خواهید داد.
 - روش‌های گام به گام^۳:
 - انتخاب رو به جلو^۴: مدل خطی را با یک ویژگی برازش دهید. این کار را با سایر ویژگی‌ها نیز انجام دهید. مدلی را انتخاب کنید که بهترین عملکرد را دارد (مثلاً بالاترین R-squared). اکنون دوباره، ویژگی‌های باقی‌مانده را یک به یک به مدل تک ویژگی قبلی اضافه کنید و بهترین ویژگی بعدی را بیابید. این کار را تا رسیدن به معیاری مانند حداکثر تعداد ویژگی‌های مدل ادامه دهید.
 - انتخاب رو به عقب^۵: مشابه انتخاب رو به جلو می‌باشد. اما به جای افزودن ویژگی‌ها، با مدلی شروع کنید که شامل همه ویژگی‌ها است و سعی کنید یک ویژگی را برای حذف بیابید که مدل بدون ویژگی، بالاترین عملکرد را داشته باشید. این کار را تا رسیدن به معیار توقف تکرار کنید.
- توصیه می‌کنم از Lasso استفاده کنید، زیرا خودکار است، همه ویژگی‌ها را به طور همزمان در نظر می‌گیرد و از طریق لامبда قابل کنترل می‌باشد. علاوه بر مدل خطی، برای مدل رگرسیون لجستیک (طبقه‌بندی) نیز کار می‌کند.

۵.۱.۸ مزايا

مدل‌سازی پیش‌بینی‌ها با استفاده از یک **تجمیع وزنی**، نحوه تولید پیش‌بینی‌ها را شفاف می‌کند. و با Lasoo می‌توانیم اطمینان حاصل کنیم که تعداد ویژگی‌های مورد استفاده، کم باقی می‌ماند. بسیاری از افراد از مدل‌های رگرسیون خطی استفاده می‌کنند. این بدان معناست که در بسیاری از موارد، برای مدل سازی پیش‌بینی و انجام استنتاج پذیرفته شده است. سطح بالایی از تجربه و تخصص جمعی، از

¹ Univariate selection

² Correlation coefficient

³ Step-wise methods

⁴ Forward selection

⁵ Backward selection

جمله مطالب آموزشی در مورد مدل‌های رگرسیون خطی و پیاده‌سازی آن در نرم افزار وجود دارد. پیاده‌سازی رگرسیون خطی را می‌توان در نرم‌افزارهای R، Python، Scala، Julia، Java، Javascript و ... یافت.
از نظر ریاضی، تخمین وزن‌ها ساده است و شما تصمیمی برای یافتن وزن‌های بهینه دارید (به شرطی که داده‌ها تمام مفروضات مدل رگرسیون خطی را دارا باشند).
همراه با وزن‌ها، فواصل اطمینان، تست‌ها و تئوری آماری قوی را ارائه می‌شود. توسعه‌های زیادی هم برای مدل رگرسیون خطی وجود دارد (بخش GAM و موارد دیگر را مطالعه کنید).

۵.۱.۹ معایب

مدل‌های رگرسیون خطی فقط می‌توانند روابط خطی را مدل کنند، یعنی مجموع وزنی ویژگی‌های ورودی. هرگونه رابطه غیرخطی یا تعامل باید به صورت دستی اعمال شود و به عنوان یک ویژگی ورودی به صورت صریح به مدل وارد شود.

همچنین اغلب، عملکرد پیش‌بینی مدل‌های خطی خوب نیست، زیرا روابطی را که می‌توانند بیاموزند، بسیار محدود است و معمولاً پیچیدگی واقعیت را بیش از حد ساده می‌کنند.

تفسیر یک وزن می‌تواند غیر شهودی^۱ باشد زیرا وزن آن ویژگی به تمام ویژگی‌های دیگر بستگی دارد. یک ویژگی با همبستگی مثبت بالا با خروجی y و یک ویژگی دیگر در ورودی ممکن است وزن منفی دریافت کند. دلیل مقدار منفی وزن ویژگی داشتن همبستگی با y در فضای با ابعاد بالا می‌باشد. ویژگی‌های کاملاً همبسته حتی ممکن است یافتن یک راه حل منحصر به فرد را برای معادله خطی، غیرممکن کنند. یک مثال: شما یک مدل برای پیش‌بینی ارزش یک خانه دارید و دارای ویژگی‌هایی مانند تعداد اتاق‌ها و اندازه خانه هستید. اندازه خانه و تعداد اتاق‌ها به شدت مرتبط هستند: هر چه خانه بزرگ‌تر باشد، اتاق‌هایی بیشتری دارد. اگر هر دو ویژگی را در یک مدل خطی قرار دهید، ممکن است این اتفاق بیفتد که اندازه خانه پیش‌بینی کننده بهتری باشد و وزن مثبت زیادی دریافت کند. تعداد اتاق‌ها ممکن است وزن منفی داشته باشد، زیرا در یک خانه دارای اندازه ثابت، افزایش تعداد اتاق‌ها می‌تواند ارزش آن را کاهش دهد. زمانی که همبستگی خیلی قوی باشد، پایداری معادله خطی کم می‌شود.

¹ Unintuitive

۵.۲ رگرسیون لجستیک^۱

رگرسیون لجستیک احتمالات را برای مسائل طبقه‌بندی با دو خروجی ممکن، مدل‌سازی می‌کند. این مدل، توسعه مدل رگرسیون خطی برای مسائل طبقه‌بندی است.

۵.۲.۱ استفاده از رگرسیون خطی برای طبقه‌بندی چه اشکالی دارد؟

مدل رگرسیون خطی می‌تواند برای رگرسیون خوب کار کند، اما برای طبقه‌بندی شکست می‌خورد. چرا اینطور است؟ در صورت وجود دو کلاس، می‌توانید یکی از کلاس‌ها را با صفر و دیگری را با یک برچسب گذاری کنید و از رگرسیون خطی استفاده کنید. از نظر فنی کار می‌کند و اکثر برنامه‌های مدل خطی، وزن‌ها را برای شما محاسبه می‌کنند. اما این روش چند مشکل دارد:

یک مدل خطی، احتمالات را به عنوان خروجی محاسبه نمی‌کند. این مدل، کلاً سهای را به عنوان اعداد (۰ و ۱) در نظر می‌گیرد و با بهترین ابر صفحه (که برای تک ویژگی، یک خط است)، فاصله بین نقاط و ابر صفحه را به حداقل می‌رساند. این کار یک درونیابی^۲ بین نقاط است و نمی‌توانید خروجی آن را به عنوان احتمال تفسیر کنید. یک مدل خطی برونویابی^۳ نیز می‌کند و خروجی زیر صفر و بالای یک نیز دارد. این مشکل، نشانه خوبی است که ممکن است رویکرد هوشمندانه‌تری برای طبقه‌بندی وجود داشته باشد.

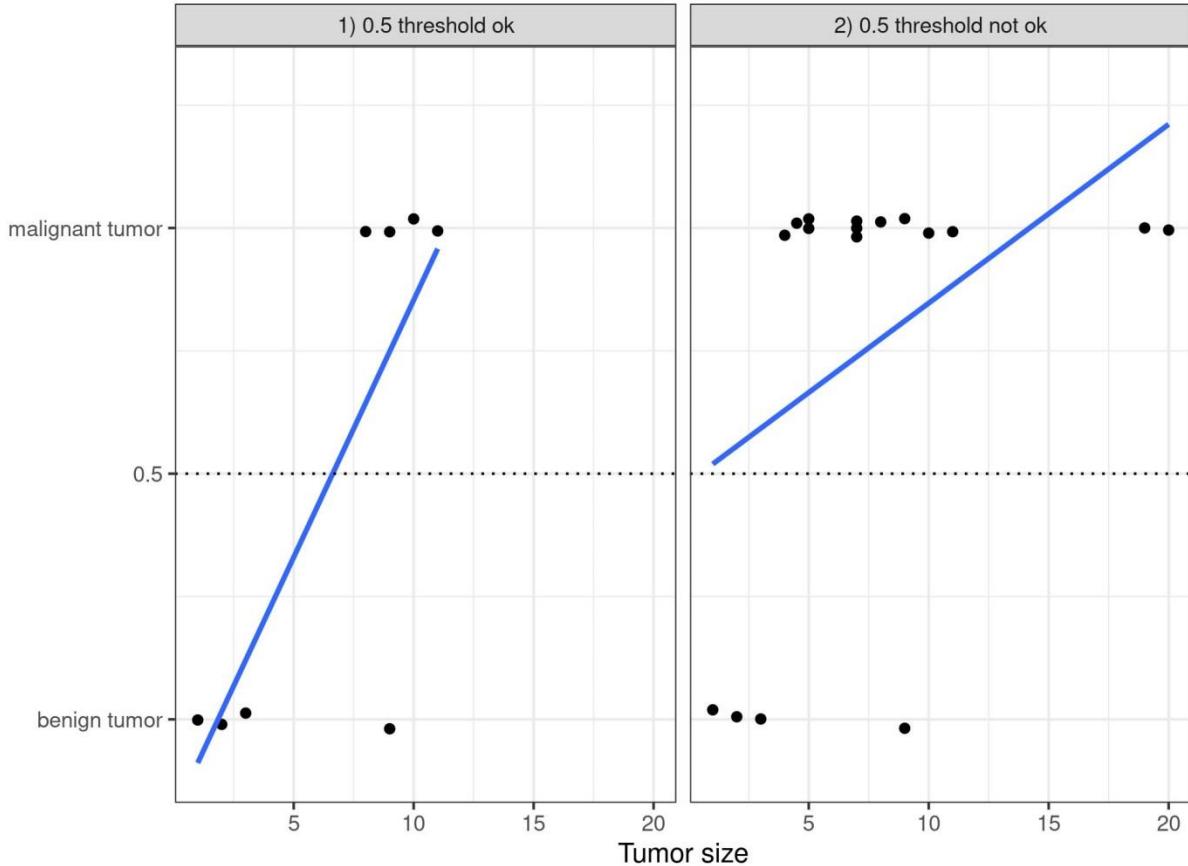
از آنجایی که نتیجه پیش‌بینی شده یک احتمال نیست، بلکه یک درونیابی خطی بین نقاط است، هیچ آستانه معنی‌داری وجود ندارد که در آن بتوانید یک کلاس را از کلاس دیگر تشخیص دهید. تصویر خوبی از این موضوع Stackoverflow(<https://stats.stackexchange.com/questions/22381/why-not-approach-classification-through-regression>) در

مدل‌های خطی به مسائل طبقه‌بندی با کلاس‌های متعدد تعمیم نمی‌یابند. در این مسائل، باید شروع به برچسب زدن کلاس بعدی با ۲، سپس ۳ و غیره کنید. کلاس‌ها ممکن است ترتیب معنی‌داری نداشته باشند، اما مدل خطی، ساختار عجیبی را در ارتباط بین ویژگی‌ها و پیش‌بینی‌های کلاس شما ایجاد می‌کند. هر چه ارزش یک ویژگی با وزن مثبت بیشتر باشد، بیشتر به پیش‌بینی کلاسی با عدد بالاتر کمک می‌کند، حتی اگر کلاس‌هایی که اتفاقاً عدد مشابهی به دست می‌آورند از کلاس‌های دیگر نزدیک‌تر نباشند.

¹ Logistic regression

² Interpolate

³ Extrapolate



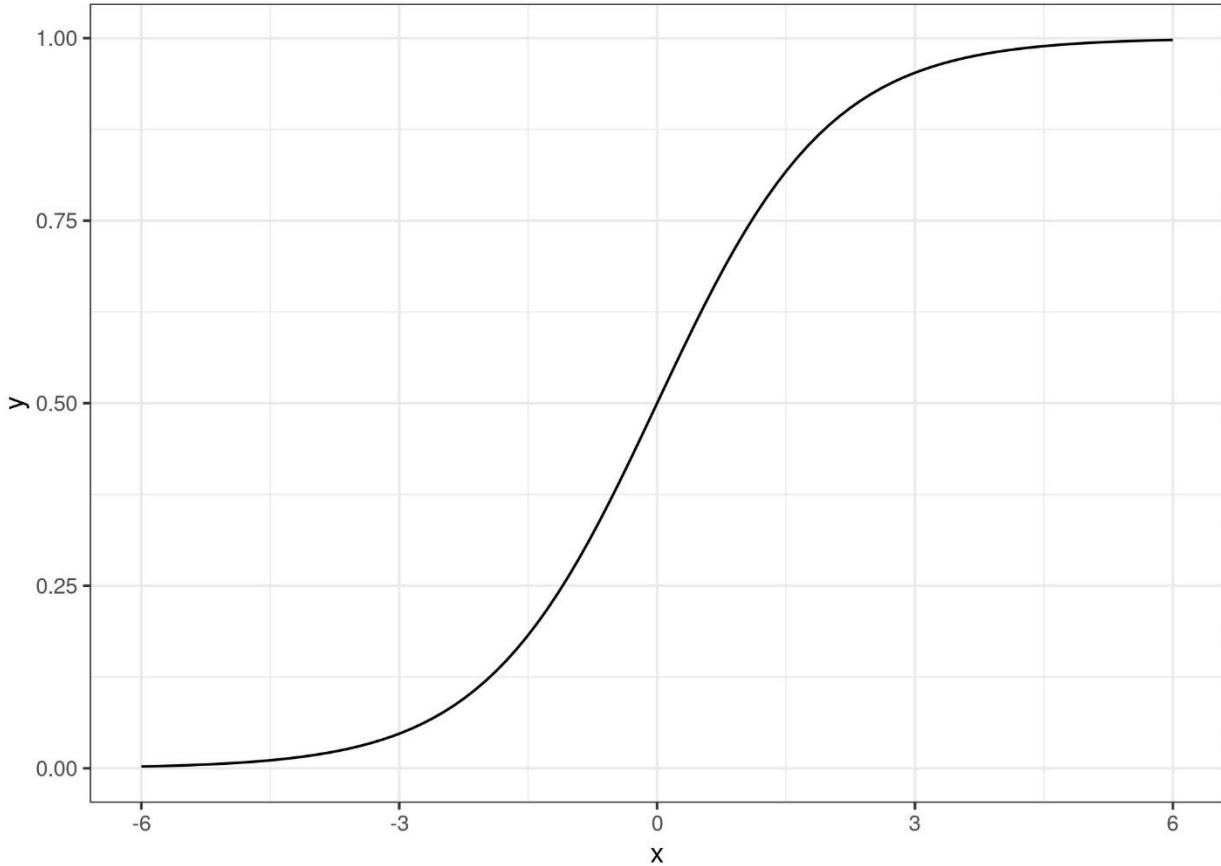
شکل ۵.۵: یک مدل خطی، تومورها را به شرط اندازه آنها، به عنوان بدخیم (۱) یا خوش خیم (۰) طبقه‌بندی می‌کند. خطوط، پیش‌بینی مدل خطی را نشان می‌دهد. برای داده‌های سمت چپ، می‌توانیم از $0/5$ به عنوان آستانه طبقه‌بندی استفاده کنیم. پس از معرفی چند مورد تومور بدخیم دیگر، خط رگرسیون تغییر می‌کند و آستانه $0/5$ دیگر کلاس‌ها را از هم جدا نمی‌کند. نقاط کمی جابجا شده اند تا در نمودار روی یکدیگر نیفتدند.

۵.۲.۲ تئوری

یک راه حل برای مسائل طبقه‌بندی، رگرسیون لجستیک است. مدل رگرسیون لجستیک به جای برازش یک خط مستقیم یا ابر صفحه، از تابع لجستیک برای فشرده کردن خروجی یک معادله خطی بین 0 و 1 استفاده می‌کند. تابع لجستیک به صورت زیر تعریف می‌شود:

$$\text{logistic}(\eta) = \frac{1}{1 + \exp(-\eta)}$$

و بدین شکل است:



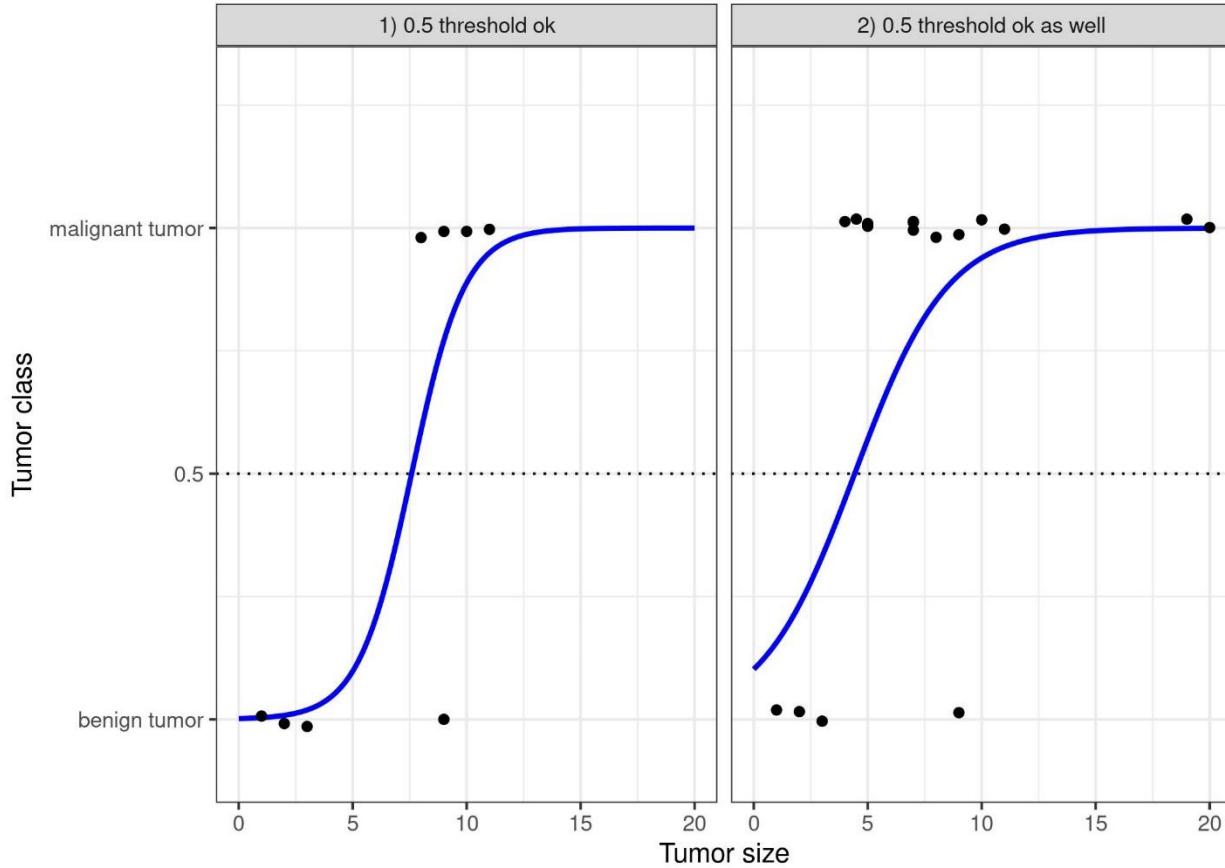
شکل ۵.۶: تابع لجستیک. خروجی اعداد بین ۰ و ۱ در ورودی ۰، خروجی ۵/۰ است. رسیدن از رگرسیون خطی به رگرسیون لجستیک ساده است. در مدل رگرسیون خطی، ما رابطه بین نتیجه و ویژگی‌ها را با یک معادله خطی مدل کرده ایم:

$$\hat{y}^{(i)} = \beta_0 + \beta_1 x_1^{(i)} + \dots + \beta_p x_p^{(i)}$$

برای طبقه‌بندی، احتمالات بین ۰ و ۱ را ترجیح می‌دهیم، بنابراین سمت راست معادله را در تابع لجستیک قرار می‌دهیم. این امر خروجی را مجبور می‌کند که فقط مقادیر بین ۰ و ۱ داشته باشد.

$$P(y^{(i)} = 1) = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 x_1^{(i)} + \dots + \beta_p x_p^{(i)}))}$$

اجازه دهید دوباره مثال اندازه تومور را بررسی کنیم. اما به جای مدل رگرسیون خطی، از مدل لجستیک استفاده کنیم:



شکل ۵.۷: مدل رگرسیون لجستیک مرز تصمیم‌گیری صحیح بین بدخیم و خوش‌خیم را بسته به اندازه تومور پیدا می‌کند. مرز، تابع لجستیکی است که برای برازش داده‌ها، جابجا و فشرده شده است.

با رگرسیون لجستیک بهتر می‌توان طبقه‌بندی کرد و در هر دو مورد می‌توانیم از $0/5$ به عنوان آستانه استفاده کنیم. اضافه کردن نقاط جدید تاثیر زیادی بر منحنی برازش شده نمی‌گذارد.

۵.۲.۳ تفسیر

تفسیر وزن‌ها در رگرسیون لجستیک با تفسیر وزن‌ها در رگرسیون خطی متفاوت است، زیرا خروجی در رگرسیون لجستیک احتمالی بین 0 و 1 است. وزن‌ها دیگر به صورت خطی، احتمال را نشان نمی‌دهند. مجموع وزنی توسط تابع لجستیک به یک احتمال تبدیل می‌شود. بنابراین ما باید معادله را برای تفسیر دوباره فرمول بندی کنیم به نحوی که فقط عبارت خطی در سمت راست فرمول باشد.

$$\ln\left(\frac{P(y=1)}{1-P(y=1)}\right) = \log\left(\frac{P(y=1)}{P(y=0)}\right) = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p$$

ما عبارت موجود در تابع "ln" را "شانس"^۱ می‌نامیم (احتمال رویداد تقسیم بر احتمال عدم رویداد) و وقتی لگاریتم آن محاسبه می‌شود به آن لگاریتم احتمالات گفته می‌شود.

این فرمول نشان می‌دهد که مدل رگرسیون لجستیک یک مدل خطی از لگاریتم احتمالات است. با کمی تحلیل، می‌توانید بفهمید که هنگامی یکی از ویژگی‌ها x_j به اندازه یک واحد تغییر می‌کند، پیش‌بینی چقدر تغییر می‌کند. برای انجام این کار، ابتدا می‌توانیم تابع \exp را در دو طرف معادله اعمال کنیم:

$$\frac{P(y = 1)}{1 - P(y = 1)} = odds = \exp(\beta_0 + \beta_1 x_1 + \dots + \beta_p x_p)$$

بنابراین آنچه را پس از تغییر یک واحد در ویژگی اتفاق می‌افتد، بررسی می‌نماییم. اما به جای بررسی تفاوت، به نسبت دو پیش‌بینی نگاه می‌کنیم:

$$\frac{odds_{x_j+1}}{odds} = \frac{\exp(\beta_0 + \beta_1 x_1 + \dots + \beta_j(x_j + 1) + \dots + \beta_p x_p)}{\exp(\beta_0 + \beta_1 x_1 + \dots + \beta_j x_j + \dots + \beta_p x_p)}$$

قانون زیر را اعمال می‌کنیم:

$$\frac{\exp(a)}{\exp(b)} = \exp(a - b)$$

و بسیاری از عبارات را حذف می‌کنیم:

$$\frac{odds_{x_j+1}}{odds} = \exp(\beta_j(x_j + 1) - \beta_j x_j) = \exp(\beta_j)$$

در پایان، عبارتی به سادگی \exp از وزن ویژگی داریم. تغییر در یک ویژگی به اندازه یک واحد، نسبت شانس (ضریب)^۲ را با ضریب (β_j) تغییر می‌دهد. همچنین می‌توانیم این‌گونه نیز تفسیر کنیم: تغییر در x_j به اندازه یک واحد، نسبت لگاریتم شانس را به اندازه مقدار وزن مربوطه افزایش می‌دهد. اکثر مردم نسبت شانس را تفسیر می‌کنند زیرا فکر کردن در مورد \ln یک عبارت، برای انسان سخت است. تفسیر نسبت شانس از قبل نیاز به تمرین دارد. به عنوان مثال، اگر شما شانس ۲ دارید، به این معنی است که احتمال $y=1$ دو برابر $y=0$ است. اگر وزن (نسبت لگاریتم شانس) $0.7/7$ دارید، آنگاه افزایش یک واحدی ویژگی مربوطه، شانس را در $\exp(0.7)$ ضرب می‌کند (تقریباً ۲) و شانس به ۴ تغییر می‌کند. اما معمولاً شما با شانس کار نمی‌کنید و وزن‌ها را فقط به عنوان نسبت‌های شانس تفسیر می‌کنید. زیرا برای محاسبه واقعی شانس باید یک مقدار برای هر ویژگی تعیین کنید، که تنها زمانی منطقی است که بخواهید یک نمونه خاص از مجموعه‌داده را در نظر بگیرید.

در اینجا تفاسیر مدل رگرسیون لجستیک با انواع مختلف ویژگی‌ها آورده شده است:

- ویژگی عددی: اگر مقدار ویژگی x_j را یک واحد افزایش دهید، شانس تخمینی با ضریب β تغییر می‌کند.

¹ Odds

² multiplicative

- ویژگی طبقه‌ای باینری: یکی از دو مقدار ویژگی، دسته مرجع است (در برخی زبان‌ها، مقداری که با β رمزگذاری می‌شود). با تغییر ویژگی β از طبقه مرجع به طبقه دیگر، شанс تخمینی، با ضریب β تغییر می‌کند.
- ویژگی طبقه‌ای با بیش از دو طبقه: یک راه حل برای با برخورد با مسائل چندین طبقه، one-hot-encoding است. بدین صورت که هر طبقه ستون خاص خود را دارد. شما فقط به $L-1$ ستون برای یک L ویژگی دسته‌ای نیاز دارید، در غیر این صورت بیش از حد پارامتر تعریف شده است. طبقه L ام، طبقه مرجع است. شما می‌توانید از هر رمزگذاری دیگری که می‌تواند در رگرسیون خطی به کار برد شود، استفاده کنید. بعد از این کار، تفسیر برای هر طبقه معادل تفسیر ویژگی‌های باینری است.
- عرض از مبدا β_0 : وقتی همه ویژگی‌های عددی صفر هستند و ویژگی‌های طبقه‌بندی در دسته مرجع قرار دارند، شанс تخمینی $\exp(\beta_0)$ می‌باشد. تفسیر عرض از مبدا معمولاً انجام نمی‌شود.

۵.۲.۴ مثال

ما از مدل رگرسیون لجستیک برای پیش‌بینی سرطان دهانه رحم بر اساس برخی عوامل خطر استفاده می‌کنیم. جدول زیر وزن‌های تخمینی، نسبت‌های شанс مربوطه و خطای استاندارد تخمین‌ها را نشان می‌دهد.

جدول ۵.۲: نتایج برآش یک مدل رگرسیون لجستیک بر روی مجموعه‌داده سرطان دهانه رحم. ویژگی‌های مورد استفاده در مدل، وزن‌های تخمینی و نسبت‌های شанс مربوطه و خطاهای استاندارد وزن‌های تخمینی آورده شده است.

	Weight	Odds ratio	Std. Error
Intercept	-2.91	0.05	0.32
Hormonal contraceptives y/n	-0.12	0.89	0.30
Smokes y/n	0.26	1.30	0.37
Num. of pregnancies	0.04	1.04	0.10
Num. of diagnosed STDs	0.82	2.27	0.33
Intrauterine device y/n	0.62	1.86	0.40

تفسیر یک ویژگی عددی STD های تشخیص داده شده (Num. of diagnosed STDs): افزایش تعداد STD های تشخیص داده شده (بیماری‌های مقابله‌یابی) شанс ابتلا به سرطان در مقایسه با عدم وجود سرطان را با ضریب $2/27$ تغییر می‌دهد (افزایش می‌دهد) در حالی که همه ویژگی‌های دیگر ثابت باقی بمانند. به خاطر داشته باشید که همبستگی، علیت را ایجاب نمی‌کند.

تفسیر یک ویژگی طبقه‌بندی شده ("داروهای ضد بارداری هورمونی بله یا خیر"): برای زنانی که از داروهای ضد بارداری هورمونی استفاده می‌کنند نسبت به زنان بدون ضد بارداری هورمونی، شанс ابتلا به سرطان در مقایسه با بدون سرطان 0.89 کمتر است، به شرطی که سایر ویژگی‌ها یکسان باقی بمانند.

مانند مدل خطی، تفاسیر همیشه با این بند آمده است که "همه ویژگی‌های دیگر ثابت می‌مانند".

۵.۲.۵ مزایا و معایب

بسیاری از مزایا و معایب مدل رگرسیون خطی در مورد مدل رگرسیون لجستیک نیز صدق می‌کند. رگرسیون لجستیک به طور گسترده توسط افراد مختلف مورد استفاده قرار گرفته است، اما با توجه به عبارات محدود خود (مثلًاً تعاملات باید به صورت دستی اضافه شوند) مشکلاتی دارد و مدل‌های دیگر ممکن است عملکرد پیش‌بینی بهتری داشته باشند.

یکی دیگر از معایب مدل رگرسیون لجستیک این است که تفسیر دشوارتری دارد، زیرا تفسیر اوزان ضربی است و افزایشی نیست.

رگرسیون لجستیک در موارد تفکیک کامل^۱ مشکل دارد. اگر ویژگی ای وجود داشته باشد که این دو کلاس را کاملاً از هم جدا کند، مدل رگرسیون لجستیک دیگر قابل آموزش نیست. این موضوع، به این دلیل است که وزن آن ویژگی همگرا^۲ نمی‌شود، زیرا وزن بهینه بی نهایت خواهد بود. این امر، واقعًا نامیدکننده است، زیرا این ویژگی واقعًا مفید است. اما اگر قانون ساده‌ای دارید که هر دو کلاس را از هم جدا می‌کند، نیازی به یادگیری ماشین ندارید. مشکل تفکیک کامل را می‌توان با معرفی جریمه وزن‌ها یا تعریف یک توزیع احتمال اولیه از وزن‌ها^۳ حل کرد.

از طرفی، مدل رگرسیون لجستیک نه تنها یک مدل طبقه‌بندی است، بلکه احتمالات را نیز محاسبه می‌دهد. این یک مزیت بزرگ نسبت به مدل‌هایی است که فقط می‌توانند طبقه‌بندی نهایی را ارائه دهند. دانستن اینکه یک نمونه برای یک کلاس ۹۹ درصد احتمال دارد در مقایسه با حالتی که ۵۱ درصد احتمال دارد، تفاوت بزرگی ایجاد می‌کند.

رگرسیون لجستیک همچنین می‌تواند از طبقه‌بندی باینری به طبقه‌بندی چند کلاسه گسترش یابد. در این حالت به آن رگرسیون چند جمله‌ای^۴ می‌گویند.

۵.۲.۶ نرم افزار

من از تابع `glm` در R برای همه مثال‌ها استفاده کردم. شما می‌توانید رگرسیون لجستیک را در هر زبان برنامه نویسی که می‌تواند برای انجام تجزیه و تحلیل داده‌ها استفاده شود، پیدا کنید، مانند `Python`, `Java`, `Stata`, `Matlab` و ...

¹ Complete separation

² converge

³ prior probability distribution of weights

⁴ Multinomial regression

۵.۳ GAM²، GLM¹ و موارد دیگر

بزرگ‌ترین نقطه قوت و همچنین بزرگ‌ترین نقطه ضعف مدل رگرسیون خطی این است که پیش‌بینی به عنوان مجموع وزنی ویژگی‌ها، مدل می‌شود. علاوه بر این، مدل خطی با بسیاری از مفروضات دیگر همراه است. خبر بد این است این است که همه این فرضیات اغلب در واقعیت نقض می‌شوند: خروجی به شرط ویژگی‌ها ممکن است توزیع غیر گاووسی داشته باشد، ویژگی‌ها ممکن است تعامل داشته باشند و رابطه بین ویژگی‌ها و خروجی ممکن است غیرخطی باشد. خبر خوب این است که جامعه آمار گونه‌های مختلفی را ایجاد کرده است که مدل رگرسیون خطی را از یک ابزار ساده به یک ابزاری قدرتمند تبدیل می‌کند.

این بخش قطعاً یک راهنمای جامع برای توسعه مدل‌های خطی نیست. بلکه به عنوان یک نمای کلی از برنامه‌های تعمیمی مانند مدل‌های خطی تعمیم یافته (GLMs) و مدل‌های افزودنی تعمیم یافته (GAMs) عمل می‌کند و کمی شهود به شما می‌دهد. پس از مطالعه، باید یک دید کلی از نحوه تعمیم مدل‌های خطی داشته باشید. اگر می‌خواهید ابتدا درباره مدل رگرسیون خطی بیشتر بدانید، پیشنهاد می‌کنم بخش مدل‌های رگرسیون خطی را مطالعه کنید، اگر قبلًاً این کار را نکرده‌اید.

بیایید فرمول یک مدل رگرسیون خطی را به خاطر بیاوریم:

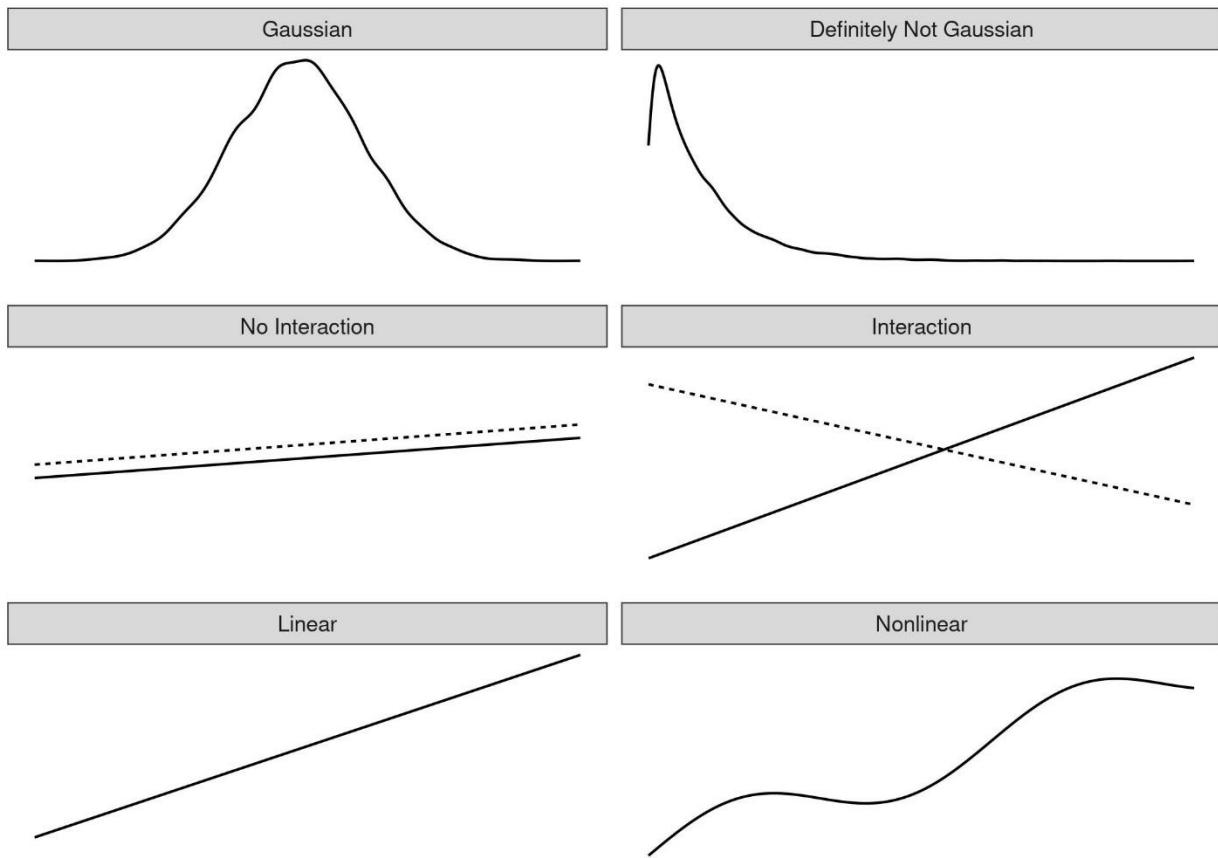
$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \epsilon$$

مدل رگرسیون خطی فرض می‌کند که نتیجه y یک نمونه را می‌توان با مجموع وزنی از p ویژگی‌های آن، با یک خطای مشخص ϵ بیان کرد، که این خطای از توزیع گاووسی پیروی می‌کند. با وارد کردن داده‌ها به این فرمول، قابلیت تفسیر خوبی به دست می‌آوریم. اثرات ویژگی تجمعی هستند، به این معنی که هیچ تعاملی وجود ندارد. رابطه خطی است، به این معنی که افزایش یک ویژگی به اندازه یک واحد می‌تواند مستقیماً به افزایش/کاهش نتیجه پیش‌بینی شده منجر شود. مدل خطی به ما اجازه می‌دهد تا رابطه بین یک ویژگی و نتیجه مورد انتظار را در یک عدد واحد، یعنی وزن تخمینی، فشرده کنیم.

اما یک جمع وزنی ساده، برای بسیاری از مسائل پیش‌بینی دنیای واقعی بسیار محدود کننده است. در این بخش با سه مشکل موجود در مدل رگرسیون خطی کلاسیک و نحوه حل آنها آشنا خواهیم شد. مسائل بسیاری وجود دارند که فرضیات در نظر گرفته شده در مدل رگرسیون خطی را نقض می‌کنند، اما ما بر روی سه مورد نشان داده شده در شکل زیر تمرکز خواهیم کرد:

¹ Generalized Linear Model

² Generalized Additive Model



شکل ۵.۸: سه فرض مدل خطی (سمت چپ): توزیع گاووسی خروجی به شرط ویژگی‌ها، تجمیعی بودن (= بدون تعامل) و رابطه خطی. در واقعیت معمولاً این مفروضات نقض می‌شوند (سمت راست): نتایج ممکن است دارای توزیع‌های غیر گاووسی باشند، ویژگی‌ها ممکن است تعامل داشته باشند و رابطه ممکن است غیرخطی باشد.
برای همه این مشکلات، راه حلی وجود دارد:

مشکل: خروجی هدف y به شرط ویژگی‌ها از توزیع گاووسی پیروی نمی‌کند.
مثال: فرض کنید می‌خواهم پیش‌بینی کنم که در یک روز معین چند دقیقه دوچرخه‌سواری خواهم کرد. به عنوان ویژگی من نوع روز، آب و هوا و غیره را دارم. اگر از یک مدل خطی استفاده کنم، می‌تواند دقیقه‌های منفی را نیز پیش‌بینی کند، زیرا توزیع را گاووسی فرض کرده ایم و این توزیع در دقیقه ۰ متوقف نمی‌شود. همچنین اگر بخواهم احتمالات را با یک مدل خطی پیش‌بینی کنم، می‌توانم احتمالات منفی یا بزرگ‌تر از ۱ را به دست بیاورم.

راه حل: مدل‌های خطی تعمیم‌یافته (GLMs)
مشکل: ویژگی‌ها با هم تعامل دارند.

مثال: به طور متوسط، باران ملایم تأثیر منفی جزئی بر تمایل من به دوچرخه سواری دارد. اما در تابستان، در ساعات شلوغی، از باران استقبال می‌کنم، زیرا در این صورت تمام دوچرخه‌سوارانی هوای مطبوع در خانه می‌مانند و من مسیرهای دوچرخه را برای خودم دارم! این یک تعامل بین زمان و آب و هوا است که با یک مدل صرفاً تجمعی قابل درک نیست.

راه حل: افزودن تعاملات به صورت دستی.

مشکل: رابطه واقعی بین ویژگی‌ها و y خطی نیست.

مثال: بین 0° تا 25° درجه سانتیگراد، تأثیر دما بر تمایل من به دوچرخه سواری می‌تواند خطی باشد، به این معنی که افزایش از 0° به 1° درجه باعث افزایش همان افزایش میل دوچرخه سواری با افزایش از 20° به 21° می‌شود. اما در دماهای بالاتر انگیزه من برای دوچرخه سواری کاهش می‌یابد و حتی کاهش می‌یابد - من دوست ندارم وقتی هوا خیلی گرم است دوچرخه سواری کنم.

راه حل‌ها: مدل‌های افزایشی تعمیم یافته (GAMs)، تبدیل ویژگی‌ها.

راه حل‌های این سه مشکل در این بخش ارائه شده است. بسیاری از تعمیمات دیگر مدل خطی حذف شده‌اند. اگر بخواهم همه موارد را در اینجا بیاورم، این بخش به سرعت تبدیل به کتابی درباره این موضوع می‌شود که قبلًا در بسیاری از کتاب‌های دیگر پوشش داده شده است. اما از آنجایی که شما در حال حاضر اینجا هستید، من یک بررسی اجمالی مشکل، به اضافه یک راه حل برای تعمیم مدل خطی آورده ام که می‌توانید در انتهای بخش آن را مشاهده کنید. نام راه حل به عنوان نقطه شروع برای جستجو است.

۵.۳.۱ خروجی غیر گاوی (GLMs)

مدل رگرسیون خطی فرض می‌کند که خروجی به شرط ویژگی‌های ورودی از یک توزیع گاوی پیروی می‌کند. این فرض شامل حال بسیاری از موارد نمی‌شود: خروجی می‌تواند یک طبقه (سرطان در مقابل سالم)، شمارش (تعداد فرزندان)، زمان وقوع یک رویداد (زمان تا خرابی یک دستگاه) یا یک نتیجه دارای چولگی^۱ زیاد با تعداد کمی مقادیر بسیار بالا (درآمد خانوار) باشد. مدل رگرسیون خطی را می‌توان برای مدل سازی همه این نوع خروجی‌ها گسترش داد. این تعمیم، مدل‌های خطی تعمیم یافته یا به اختصار GLM نامیده می‌شود. در طول این بخش، من از نام GLM هم برای چارچوب کلی و هم برای مدل‌های خاص این چارچوب استفاده خواهم کرد. مفهوم اصلی هر GLM این است: جمع وزنی ویژگی‌ها را حفظ کنید، اما توزیع‌های نتیجه غیر گاوی را مجاز کنید و میانگین مورد انتظار این توزیع و مجموع وزنی را از طریق یکتابع احتمالاً غیرخطی به هم مرتبط کنید. به عنوان مثال، مدل رگرسیون لجستیک توزیع برنولی را برای خروجی فرض می‌کند و میانگین مورد انتظار و مجموع وزنی را با استفاده از تابع لجستیک به هم مرتبط می‌کند.

¹ Skewness

GLM به صورت ریاضی، جمع وزنی ویژگی‌ها را با مقدار میانگین توزیع فرضی با استفاده از تابع اتصال^۱ g متصل می‌کند، که این تابع اتصال بسته به نوع خروجی می‌تواند به طور انعطاف پذیر انتخاب شود.

$$g(E_Y(y|x)) = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p$$

GLM‌ها از سه جزء تشکیل شده‌اند: تابع اتصال g ، مجموع وزنی $\beta^T X$ (گاهی اوقات پیش‌بینی کننده خطی نامیده می‌شود) و یک توزیع احتمال از خانواده نمایی^۲ که با E_Y تعریف می‌شود.

خانواده نمایی مجموعه‌ای از توزیع‌هایی است که می‌توان با همان فرمول (پارامتری شده) نوشت که شامل یک توان^۳، میانگین و واریانس توزیع و برخی پارامترهای دیگر است. من وارد جزئیات ریاضی نمی‌شوم زیرا این جهان بسیار بزرگی است که من نمی‌خواهم وارد آن شوم. ویکی‌پدیا فهرست دقیقی از توزیع‌ها از خانواده نمایی دارد. هر توزیعی از این لیست می‌تواند برای GLM انتخاب شود. بر اساس نوع خروجی که می‌خواهید پیش‌بینی کنید، توزیع مناسبی را انتخاب کنید. آیا خروجی، شمارش چیزی است (مثلًاً تعداد کودکانی که در یک خانواده زندگی می‌کنند)? در این حالت، توزیع پواسون^۴ می‌تواند انتخاب خوبی باشد. آیا نتیجه همیشه مثبت است (مثلًاً زمان بین دو رویداد)? در این حالت، توزیع نمایی^۵ می‌تواند انتخاب خوبی باشد.

اجازه دهید مدل خطی کلاسیک را به عنوان یک مورد خاص از یک GLM در نظر بگیریم. تابع اتصال برای توزیع گاووسی در مدل خطی کلاسیک به سادگی تابع همانی^۶ است. توزیع گاووسی با پارامترهای میانگین و واریانس، پارامتری می‌شود. میانگین مقداری را که به طور متوسط انتظار داریم و واریانس نشان می‌دهد که مقادیر در حدود این میانگین چقدر تغییر می‌کنند. در مدل خطی، تابع اتصال، مجموع وزنی ویژگی‌ها را به میانگین توزیع گاووسی متصل می‌کند.

در چارچوب GLM، این مفهوم، به هر توزیع (از خانواده نمایی) و توابع اتصال دلخواه تعمیم می‌یابد. اگر y شمارشی از چیزی باشد، مانند تعداد قهقهه‌هایی که فرد در یک روز خاص می‌نوشد، می‌توانیم آن را با GLM با توزیع پواسون و لگاریتم طبیعی به عنوان تابع اتصال مدل‌سازی کنیم:

$$\ln(E_Y(y|x)) = X^T \beta$$

مدل رگرسیون لجستیک نیز یک GLM است که توزیع را برخلاف فرض می‌کند و از تابع logit به عنوان تابع اتصال استفاده می‌کند. میانگین توزیع دوچشمی ای مورد استفاده در رگرسیون لجستیک احتمال از y با مقدار ۱ می‌باشد.

¹ Link function

² Exponential

³ Exponent

⁴ Poisson distribution

⁵ exponential distribution

⁶ Identity function

$$X^T \beta = \ln \left(\frac{E_Y(y|x)}{1 - E_Y(y|x)} \right) = \ln \left(\frac{P(y=1|x)}{1 - P(y=1|x)} \right)$$

و اگر این معادله را بنحوی حل کنیم که در یک طرف $P(y=1)$ باشد، فرمول رگرسیون لجستیک به دست می‌آید:

$$P(y=1) = \frac{1}{1 + \exp(-x^T \beta)}$$

هر توزیع از خانواده نمایی دارای یکتابع اتصال متعارف^۱ است که می‌تواند به صورت ریاضی از توزیع استخراج شود. چارچوب GLM امکان انتخاب تابع اتصال را مستقل از توزیع فراهم می‌کند. چگونه تابع اتصال مناسب را انتخاب کنیم؟ هیچ دستور العمل کاملی وجود ندارد. شما دانش خود را در مورد توزیع هدف در نظر می‌گیرید، اما ملاحظات نظری و اینکه مدل چقدر با داده‌های واقعی شما مطابقت دارد را نیز در نظر داشته باشید. برای برخی از توزیع‌ها، تابع اتصال متعارف می‌تواند به مقادیری منجر شود که برای آن توزیع نامعتبر هستند. در مورد توزیع نمایی، تابع اتصال متعارف، معکوس منفی^۲ است که می‌تواند منجر به پیش‌بینی‌های منفی شود که خارج از دامنه توزیع نمایی هستند. از آنجایی که می‌توانید هر تابع اتصالی را انتخاب کنید، راه حل ساده این است که تابع دیگری را انتخاب کنید که با دامنه توزیع هماهنگ باشد.

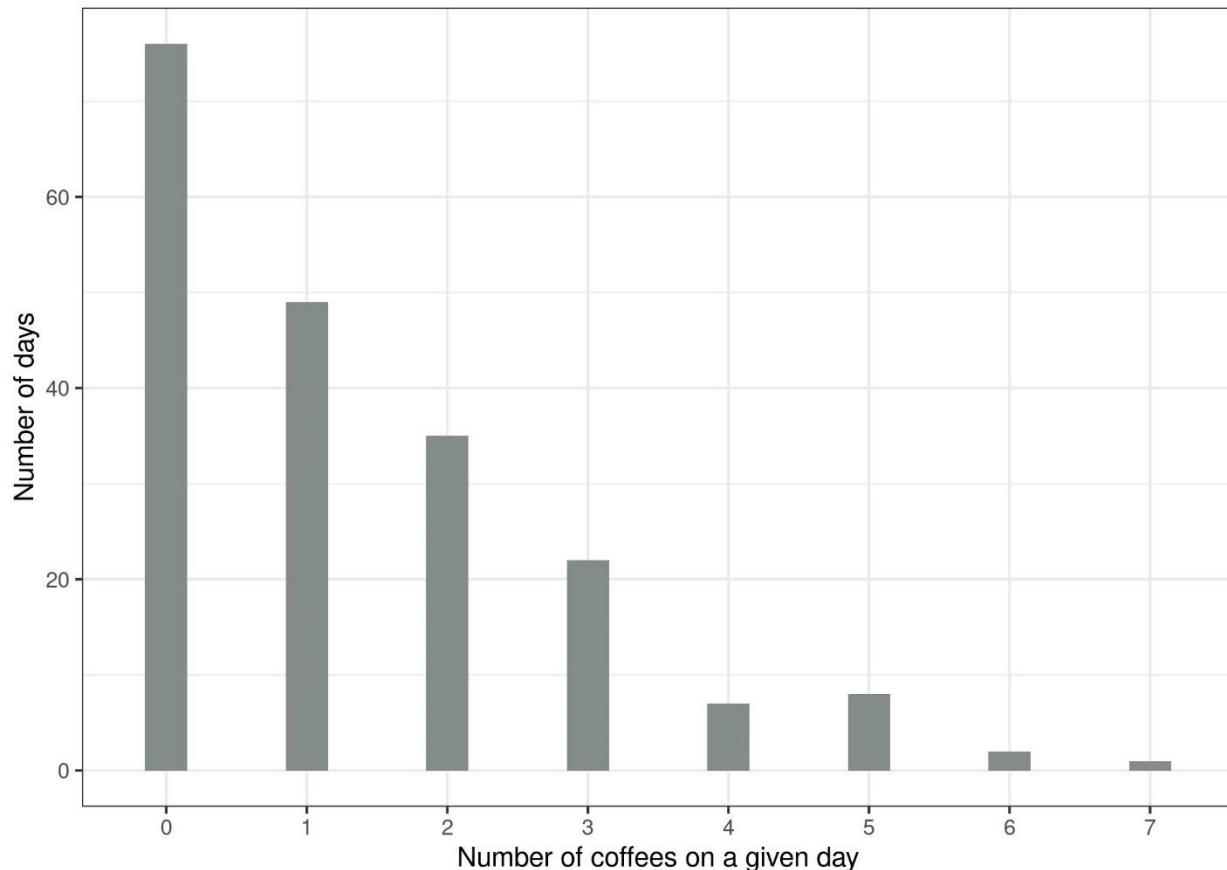
مثال‌ها

من مجموعه‌داده‌ای را در مورد رفتار نوشیدن قهوه شبیه سازی کرده ام تا نیاز به GLM‌ها را برجسته کنم. فرض کنید اطلاعاتی در مورد رفتار نوشیدن قهوه روزانه خود جمع‌آوری کرده اید. اگر قهوه دوست ندارید، در مورد چای این کار را انجام دهد. همراه با تعداد فنجان‌ها، سطح استرس فعلی خود را در مقیاس ۱ تا ۱۰ ثبت می‌کنید، شب قبل چقدر خوب خوابیده اید در مقیاس ۱ تا ۱۰ و اینکه آیا باید در آن روز کار کنید یا خیر. هدف پیش‌بینی تعداد قهوه‌ها به شرط ویژگی‌های استرس، خواب و کار است. من داده‌ها را برای ۲۰۰ روز شبیه سازی کردم. استرس و خواب به طور یکنواخت بین ۱ تا ۱۰ و بله/نه کار با شанс ۵۰/۵۰ در نظر گرفته شد. سپس برای هر روز، تعداد قهوه‌ها از توزیع پواسون گرفته شد و مدل‌سازی شدت λ (که مقدار مورد انتظار توزیع پواسون است) به عنوان تابعی از ویژگی‌های خواب، استرس و کار انجام شد. می‌توانید حدس بزنید که این داستان به کجا ختم می‌شود: "اجازه دهید این داده‌ها را با یک مدل خطی مدل‌سازی کنیم. متسافانه مدل خطی کار نمی‌کند. حالا اجازه دهید یک GLM با توزیع پواسون را امتحان کنیم. حالا کار می‌کند!".

بیایید به توزیع متغیر هدف، تعداد قهوه در یک روز معین نگاه کنیم:

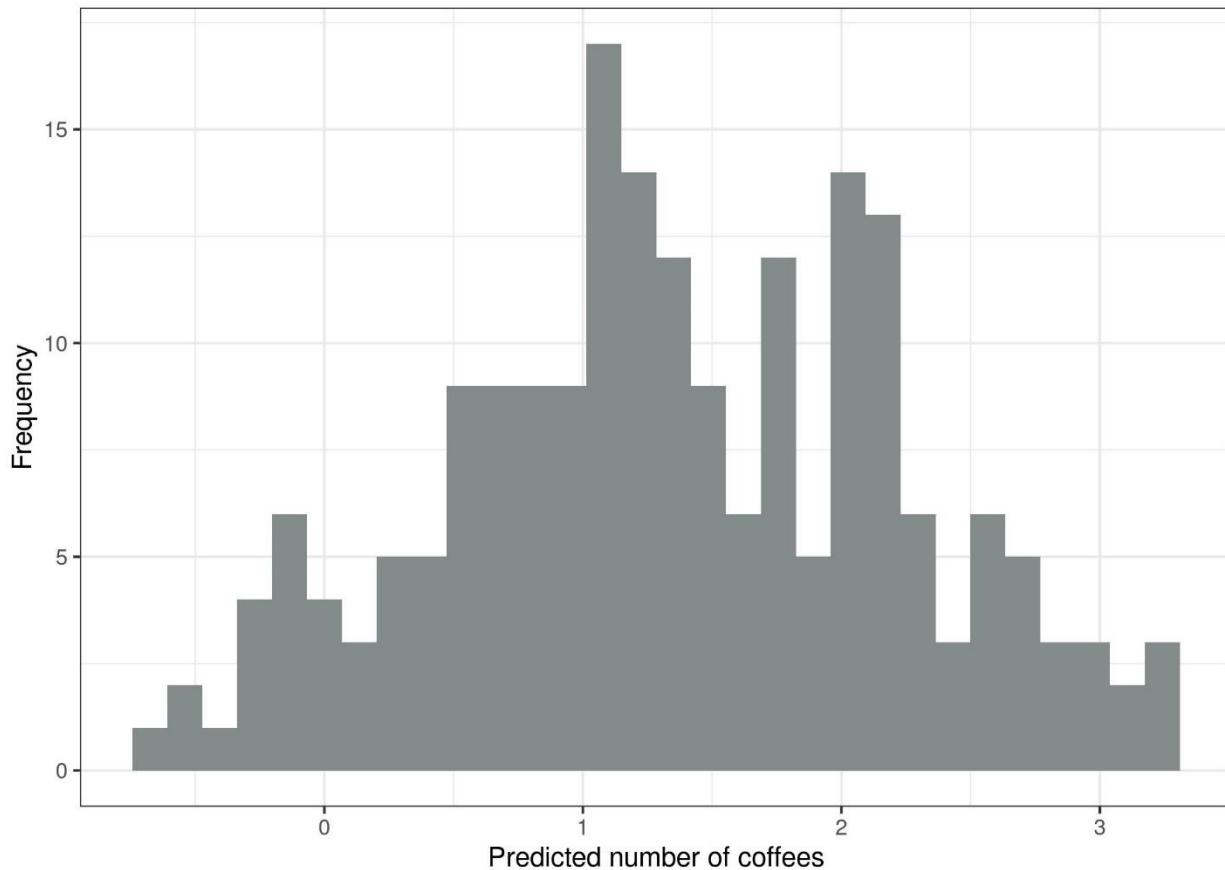
¹ canonical link function

² Negative inverse



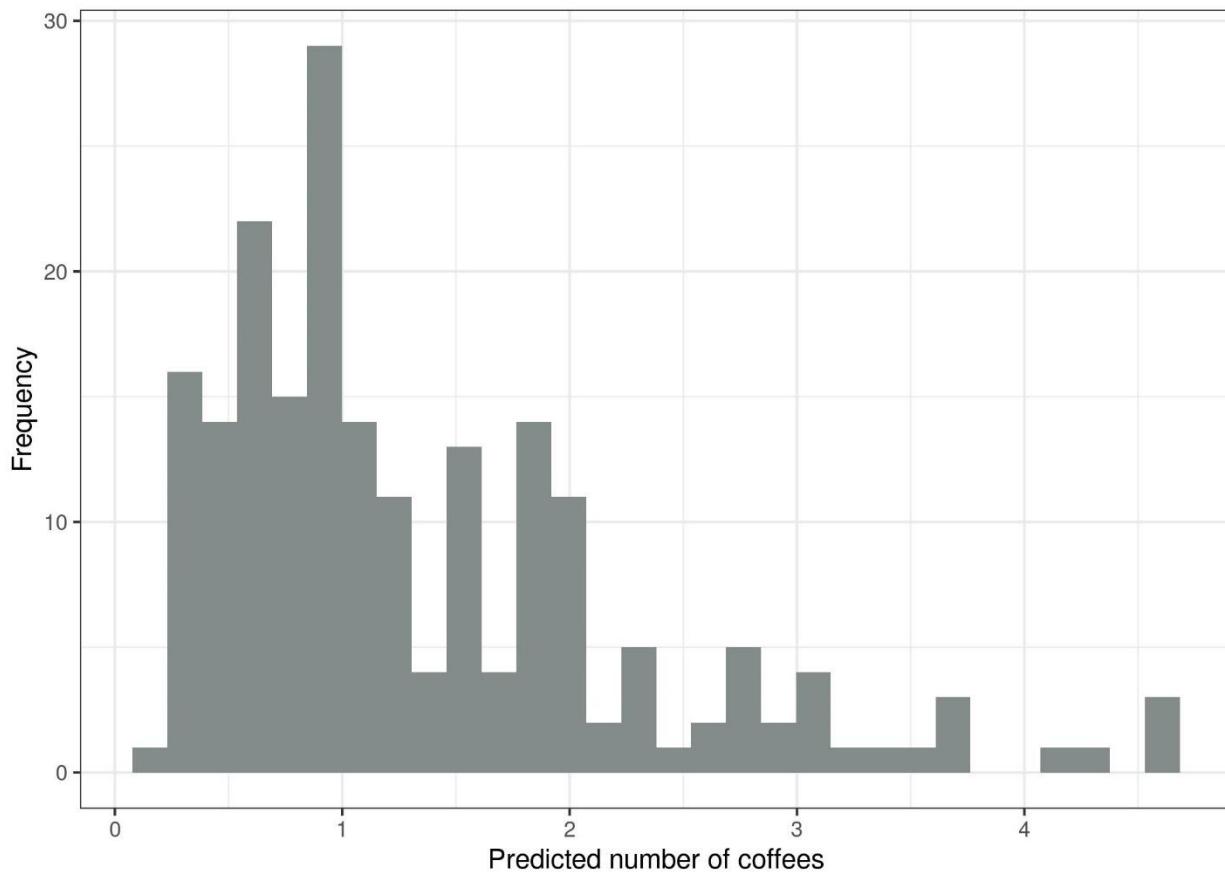
شکل ۵.۹: توزیع شبیه سازی شده تعداد قهوه های روزانه برای ۲۰۰ روز.

در ۷۶ روز از ۲۰۰ روز، اصلاً قهوه نخوردید و در شدیدترین روز، ۷ قهوه خوردید. اجازه دهید ساده لوحانه از یک مدل خطی برای پیش بینی تعداد قهوه ها با استفاده از سطح خواب، سطح استرس و کار بله / خیر به عنوان ویژگی ها استفاده کنیم. وقتی به اشتباه توزیع گاووسی را فرض می کنیم مرتبک چه اشتباہی شده ایم؟ یک فرض اشتباه می تواند منجر به نامعتبر شدن تخمین ها، به ویژه فواصل اطمینان وزن ها، شود. مشکل واضح تر این است که پیش بینی ها، با دامنه «مجاز» خروجی واقعی مطابقت ندارند، همان طور که شکل زیر نشان می دهد.



شکل ۵.۱۰: تعداد قهوه‌های پیش‌بینی شده با استفاده از ویژگی‌های استرس، خواب و کار. مدل خطی مقادیر منفی را پیش‌بینی می‌کند.

مدل خطی منطقی نیست، زیرا تعداد قهوه‌های منفی را پیش‌بینی می‌کند. این مشکل را می‌توان با مدل‌های خطی تعمیم یافته (GLM) حل کرد. ما می‌توانیم تابع اتصال و توزیع فرضی را تغییر دهیم. یکی از امکان‌ها حفظ توزیع گاوی و استفاده از تابع اتصالی است که همیشه منجر به پیش‌بینی‌های مثبت می‌شود مانند تابع log-link (معکوس تابع \exp است) به جای تابع همانی. حتی بهتر: ما توزیعی را انتخاب می‌کنیم که با فرآیند تولید داده و یک تابع اتصال مناسب مطابقت دارد. از آنجایی که نتیجه یک شمارش است، توزیع پواسون یک انتخاب مناسب به همراه لگاریتم به عنوان تابع اتصال است. در این مورد بخصوص، که داده‌ها با توزیع پواسون تولید شده‌اند، GLM پواسون، انتخابی عالی است. GLM پواسون، برآش داده شده منجر به توزیع زیر از مقادیر پیش‌بینی شده می‌شود:



شکل ۵.۱۱: تعداد قهوه‌های پیش‌بینی شده با توجه به استرس، خواب و کار. GLM با فرض توزیع پواسون و تابع اتصال \log مدل مناسبی برای این مجموعه‌داده است. بدون مقادیر منفی قهوه، اکنون مدل بسیار بهتر به نظر می‌رسد.

تفسیر وزن‌های GLM

توزیع مفروض همراه با تابع اتصال تعیین می‌کند که وزن ویژگی برآورد شده، چگونه تفسیر می‌شود. در مثال شمارش قهوه، من از یک GLM با توزیع پواسون و اتصال لگاریتمی استفاده کردم که رابطه زیر بین خروجی مورد انتظار و ویژگی‌های استرس (str)، خواب (slp) و کار (wrk) را ایجاد می‌کند.

$$\ln(E(\text{coffee}|\text{str} - \text{slp} - \text{wrk})) = \beta_0 + \beta_{\text{str}}x_{\text{str}} + \beta_{\text{slp}}x_{\text{slp}} + \beta_{\text{wrk}}x_{\text{wrk}}$$

برای تفسیر وزن‌ها، تابع اتصال را معکوس می‌کنیم تا بتوانیم تأثیر ویژگی‌ها را بر خروجی مورد انتظار تفسیر کنیم و نه بر لگاریتم خروجی مورد انتظار.

$$E(\text{coffee}|\text{str} - \text{slp} - \text{wrk}) = \exp(\beta_0 + \beta_{\text{str}}x_{\text{str}} + \beta_{\text{slp}}x_{\text{slp}} + \beta_{\text{wrk}}x_{\text{wrk}})$$

از آنجایی که همه وزن‌ها در تابع نمایی هستند، تفسیر اثر تجمعی نیست، بلکه ضربی است، زیرا $\exp(a + b)$ برابر با $\exp(a) \exp(b)$ است. آخرین جزء برای تفسیر، وزن‌های واقعی داده‌های ساختگی است. جدول زیر وزن‌های تخمینی و $\exp(\text{weight})$ را همراه با فاصله اطمینان ۹۵ درصد فهرست می‌کند:

جدول ۵.۳: وزن‌ها در مدل پواسون

	weight	$\exp(\text{weight}) [2.5\%, 97.5\%]$
(Intercept)	-0.16	0.85 [0.54, 1.32]
stress	0.12	1.12 [1.07, 1.18]
sleep	-0.15	0.86 [0.82, 0.90]
workYES	0.80	2.23 [1.72, 2.93]

افزایش سطح استرس به اندازه یک واحد، تعداد قهوه مورد انتظار را در ضریب ۱/۱۲ ضرب می‌کند. افزایش کیفیت خواب به اندازه یک واحد، تعداد قهوه مورد انتظار را در ضریب ۰/۸۶ ضرب می‌کند. تعداد قهوه‌های پیش‌بینی شده در یک روز کاری به طور متوسط ۲/۲۳ برابر تعداد قهوه‌های یک روز تعطیل است. به طور خلاصه، هر چه استرس بیشتر، خواب کمتر و کار بیشتر باشد، قهوه بیشتری مصرف می‌شود.

در این بخش شما کمی در مورد مدل‌های خطی تعمیم یافته یاد گرفتید و دانستید زمانی مفید هستند که خروجی از توزیع گاوی پیروی نمی‌کند. در مرحله بعد، به نحوه ادغام تعاملات بین دو ویژگی در مدل رگرسیون خطی می‌پردازیم.

۵.۳.۲ تعاملات

مدل رگرسیون خطی فرض می‌کند که تأثیر یک ویژگی بدون توجه به مقادیر سایر ویژگی‌ها یکسان است (= بدون تعامل). اما اغلب در داده‌ها تعاملاتی وجود دارد. برای پیش‌بینی تعداد دوچرخه‌های کرایه شده، ممکن است بین دما و اینکه روز کاری است یا نه، تعاملی وجود داشته باشد. شاید وقتی مردم مجبور به کار هستند، دما زیاد روی تعداد دوچرخه‌های اجاره‌ای تأثیر نمی‌گذارد، زیرا مردم هر اتفاقی بیفتند با دوچرخه اجاره‌ای به محل کار خود می‌روند. در روزهای تعطیل، بسیاری از مردم برای تفریح دوچرخه سواری می‌کنند به شرطی که هوا به اندازه کافی گرم باشد. وقتی صحبت از دوچرخه‌های کرایه ای می‌شود، ممکن است انتظار تعامل بین دما و روز کاری را داشته باشد.

چگونه می‌توانیم مدل خطی را شامل تعاملات کنیم؟ قبل از اینکه مدل خطی را برازش کنید، یک ستون به ماتریس ویژگی اضافه کنید که نشان دهنده تعامل بین ویژگی‌ها است و در ادامه مطابق معمول مدل را برازش دهید. این راه حل جوری انتخاب شده است که نیاز به هیچ تغییری در مدل خطی ندارد، فقط به ستون‌های اضافی در داده‌ها نیاز دارد. در مثال روز کاری و دما، یک ویژگی جدید اضافه می‌کنیم که برای روزهای غیرکاری

صفر است، و برای روزهای کاری مقدار ویژگی دما را دارد (با فرض اینکه روز کاری طبقه مرجع باشد). فرض کنید داده‌های ما به این شکل است:

Work	temp
Y	25
N	12
N	30
Y	5

ماتریس داده استفاده شده توسط مدل خطی کمی متفاوت با مجموعه داده اصلی می‌باشد. جدول زیر نشان می‌دهد که اگر هیچ گونه تعاملی را اضافه نکنیم، داده‌های تهیه شده برای مدل چگونه هستند. به طور معمول، این تبدیل به طور خودکار توسط هر نرم افزار آماری انجام می‌شود.

Intercept	workY	temp
1	1	25
1	0	12
1	0	30
1	1	5

ستون اول عبارت عرض از مبدا است. ستون دوم ویژگی طبقه‌بندی را با ۰ برای دسته مرجع و ۱ برای دسته دیگر رمزگذاری می‌کند. ستون سوم شامل دما است.

اگر بخواهیم مدل خطی تعامل بین دما و ویژگی روز کاری را در نظر بگیرد، باید یک ستون برای تعاملات اضافه کنیم:

Intercept	workY	temp	workY.temp
1	1	25	25
1	0	12	0
1	0	30	0
1	1	5	5

ستون جدید "workY.temp" تعامل بین ویژگی‌های روز کاری (work) و دما (temp) را نشان می‌دهد. برای مثال اگر ویژگی کار در رده مرجع ("N" برای روز غیرکاری) باشد، این ستون ویژگی جدید صفر است، در غیر این صورت مقادیر ویژگی برابر با دمای نمونه‌ها را می‌باشد. با این نوع رمزگذاری، مدل خطی می‌تواند یک اثر خطی متفاوت دما را برای هر دو نوع روز یاد بگیرد. این اثر متقابل بین دو ویژگی است. بدون یک عبارت تعاملی، اثر ترکیبی یک ویژگی دسته‌ای و عددی را می‌توان با خطی توصیف کرد که برای دسته‌های مختلف به صورت

عمودی جابجا شده است. اگر تعامل را لحاظ کنیم، اجازه می‌دهیم اثر ویژگی‌های عددی (شیب) در هر طبقه، مقدار متفاوتی داشته باشد.

برای تعامل دو ویژگی طبقه‌ای به طور مشابه عمل می‌شود. ما ویژگی‌هایی اضافی ایجاد می‌کنیم که ترکیبی از طبقه‌ها را نشان می‌دهد. در اینجا برخی از داده‌های مصنوعی حاوی روز کاری (work) و یک ویژگی طبقه‌ای آب و هوا (wthr) آمده است:

work	wthr
Y	2
N	0
N	1
Y	2

در مرحله بعد، ما عبارات تعامل را اعمال می‌کنیم:

Intercept	workY	wthr1	wthr2	workY.wthr1	workY.wthr2
1	1	0	1	0	1
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	1	0	1

ستون اول برای تخمین عرض از مبدا است. ستون دوم ویژگی کار رمزگذاری شده است. ستون‌های سه و چهار برای ویژگی آب و هوا هستند که به دو ستون نیاز دارند زیرا برای ثبت تاثیر برای سه دسته نیاز به دو وزن دارد که یکی از آن‌ها دسته مرجع است. بقیه ستون‌ها تعاملات را نشان می‌دهند. برای هر دسته از هر دو ویژگی (به جز دسته‌های مرجع)، یک ستون ویژگی جدید ایجاد می‌کنیم که اگر مقدار هر دو ویژگی ۱ باشد، مقدار ۱ می‌گیرد و در غیر این صورت ۰ است.

برای دو ویژگی عددی، ساخت ستون تعامل آسان‌تر است: ما به سادگی هر دو ویژگی عددی را ضرب می‌کنیم. رویکردهایی برای شناسایی خودکار و افزودن عبارات تعاملی وجود دارد. یکی از آنها را می‌توان در فصل RuleFit یافت. الگوریتم RuleFit ابتدا عبارات تعامل را استخراج می‌کند و سپس یک مدل رگرسیون خطی شامل تعاملات را تخمین می‌زند.

مثال

اجازه دهید به مساله پیش‌بینی اجاره دوچرخه که قبلاً در بخش مدل خطی، مدل‌سازی کرده‌ایم بازگردیم. این بار، علاوه بر این موارد قبلی، تعامل بین ویژگی‌های دما و روز کاری را در اضافه می‌کنیم. این کار، منجر به وزن‌های تخمینی و فواصل اطمینان زیر می‌شود.

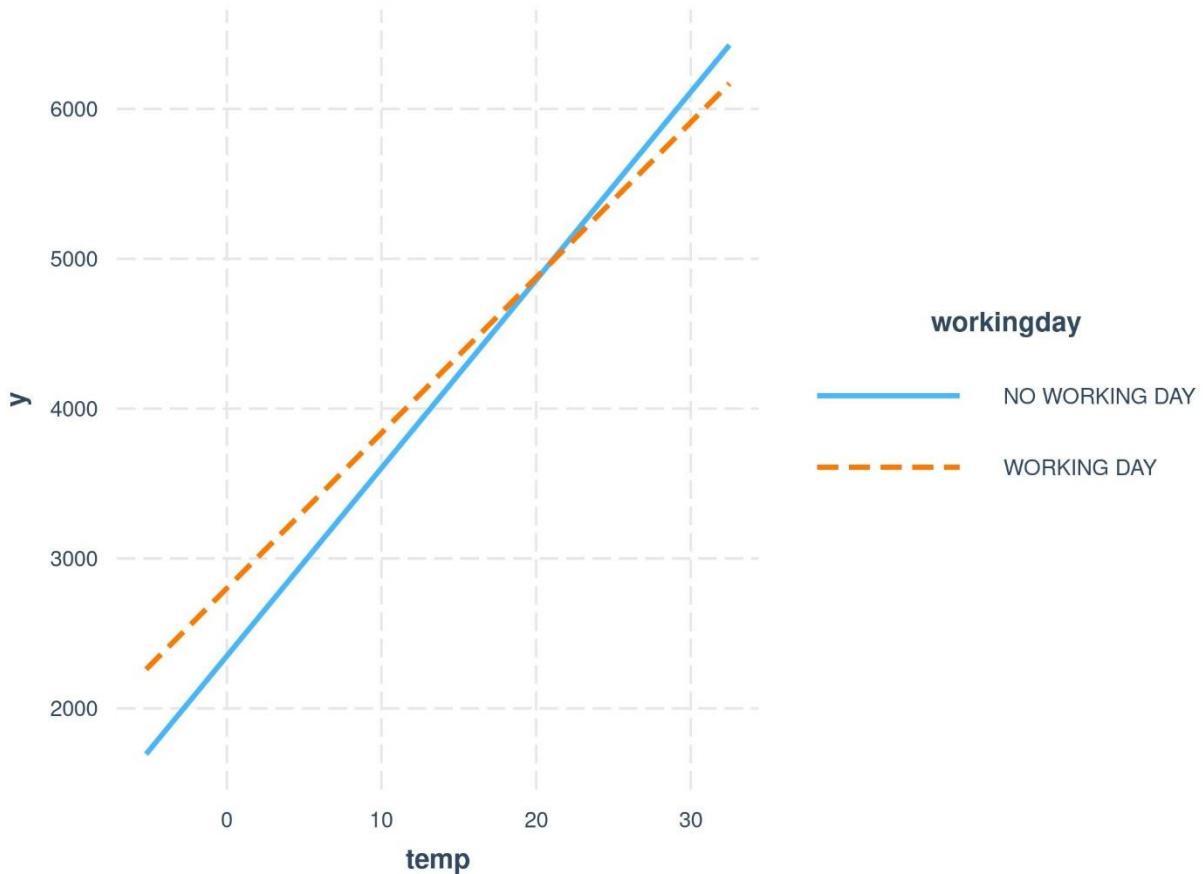
	Weight	Std. Error	2.5%	97.5%
(Intercept)	2185.8	250.2	1694.6	2677.1
seasonSPRING	893.8	121.8	654.7	1132.9
seasonSUMMER	137.1	161.0	-179.0	453.2
seasonFALL	426.5	110.3	209.9	643.2

holiday	HOLIDAY	-674.4	202.5	-1071.9	-276.9
workingday	WORKING DAY	451.9	141.7	173.7	730.1
weathersit	MISTY	-382.1	87.2	-553.3	-211.0
weathersit	RAIN/...	-1898.2	222.7	-2335.4	-1461.0
temp		125.4	8.9	108.0	142.9
hum		-17.5	3.2	-23.7	-11.3
windspeed		-42.1	6.9	-55.5	-28.6
days_since_2011		4.9	0.2	4.6	5.3
workingday	WORKING DAY:temp	-21.8	8.1	-37.7	-5.9

اثر تعامل اضافی منفی است (-۲۱/۸) و تفاوت معناداری با صفر دارد (همان‌طور که مشاهده می‌شود فاصله اطمینان ۹۵٪ شامل صفر نمی‌شود). به هر حال، داده‌ها^۱ iid (مستقل با توزیع یکسان) نیستند، زیرا روزهای نزدیک به یکدیگر، مستقل از یکدیگر نیستند. فواصل اطمینان ممکن است گمراه کننده باشد، در نتیجه زیاد آن را جدی نگیرید. عبارت تعامل، تفسیر وزن‌های ویژگی‌های درگیر در آن را تغییر می‌دهد. به شرطی که روز کاری باشد، آیا دما تأثیر منفی دارد؟ حتی یک کاربر آموزش ندیده هم می‌داند که جواب منفی است. ما نمی‌توانیم وزن تعامل «workingdayWORKING DAY:temp» را به صورت مجزا تفسیر کنیم، زیرا این تفسیر به این صورت خواهد بود: درحالی که همه مقادیر ویژگی‌های دیگر بدون تغییر باقی بماند، افزایش اثر تعامل دما برای روز کاری، تعداد پیش‌بینی‌شده دوچرخه‌ها را کاهش می‌دهد. اما اثر تعامل فقط به اثر اصلی دما اضافه می‌شود. فرض کنید یک روز کاری است و می‌خواهیم بدانیم اگر امروز، دمای هوا یک درجه گرمتر بود چه اتفاقی می‌افتد. پس باید هر دو وزن "temp" و "workingday WORKING DAY:temp" را جمع کنیم تا تعیین کنیم تخمین چقدر افزایش می‌یابد.

درک تعامل به صورت بصری آسان‌تر است. با معرفی یک عبارت تعاملی بین یک ویژگی طبقه‌ای و عددی، به جای یک شب، دو شب برای دما به دست می‌آوریم. شب دما برای روزهایی که افراد مجبور به کار نیستند ("NO WORKING DAY") مستقیماً از جدول (۱۲۵/۴) قابل خواندن است. شب دما برای روزهایی که افراد باید در آن کار کنند ("روز کاری") مجموع هر دو وزن دما ($10\frac{3}{6} + 21\frac{1}{8} = 125\frac{1}{8}$) است. عرض از مبدا خط 'NO WORKING DAY' در دمای صفر توسط عرض از مبدا مدل خطی ($2185\frac{1}{8}$) تعیین می‌شود. عرض از مبدا خط "روز کاری" در دمای صفر با عرض از مبدا مدل خطی + اثر روز کاری ($2185\frac{1}{8} + 451\frac{9}{6} = 2637\frac{7}{7}$) تعیین می‌شود.

¹ Independent and identically distributed



شکل ۵.۱۲: تأثیر (شامل تعامل) دما و روز کاری بر تعداد پیش‌بینی شده دوچرخه‌ها با استفاده از یک مدل خطی. به طور موثر، ما دو شیب برای دما داریم، برای هر طبقه از ویژگی روز کاری، یک شیب.

۵.۳.۳ تأثیرات غیر خطی GAM

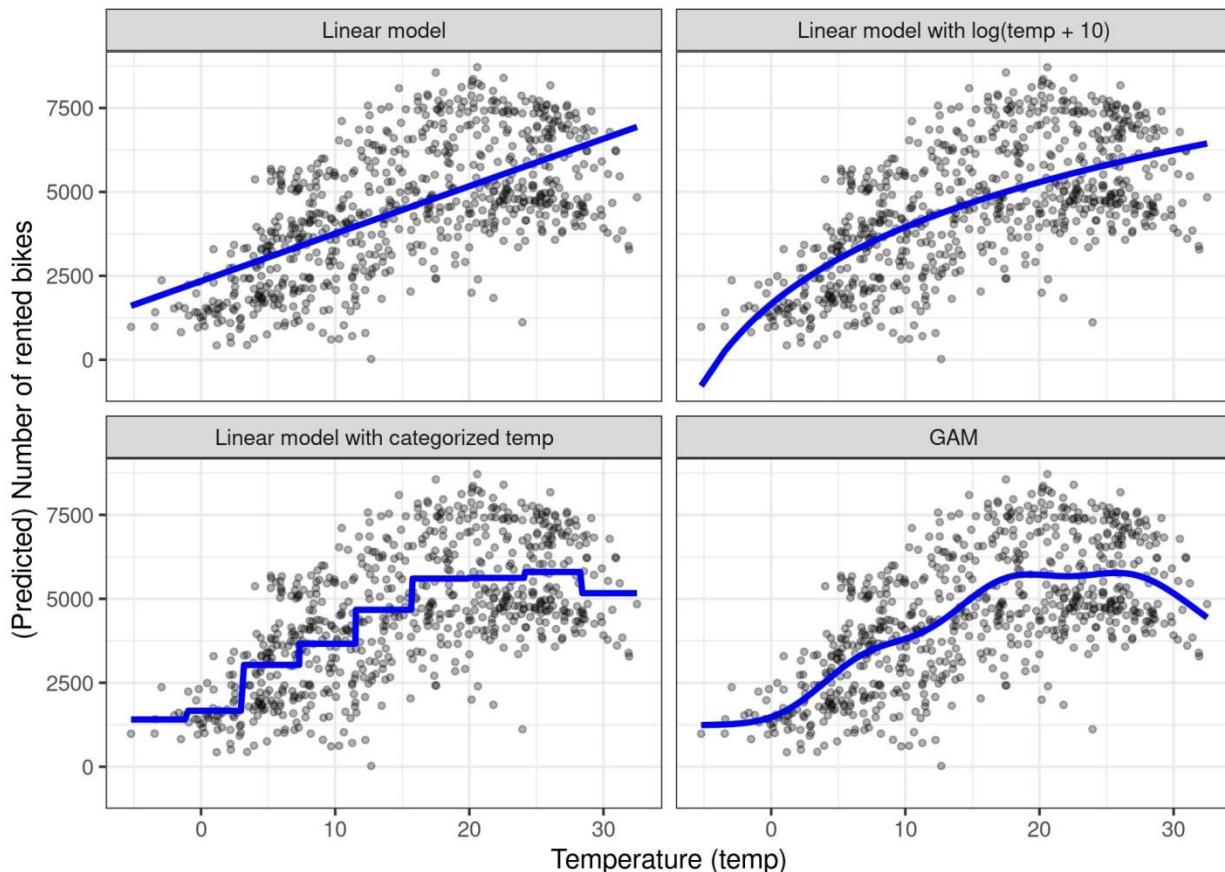
دنیا خطی نیست. خطی بودن در مدل‌های خطی به این معنی است که صرفنظر از مقداری که یک نمونه در یک ویژگی خاص داشته باشد، افزایش مقدار به اندازه یک واحد همیشه همان اثر را بر خروجی پیش‌بینی شده دارد. آیا منطقی است که فرض کنیم افزایش یک درجه دما در ۱۰ درجه سانتیگراد همان تأثیری را بر تعداد دوچرخه‌های اجاره‌ای دارد که افزایش دما در ۴۰ درجه دارد؟ به طور شهودی، انتظار می‌رود که افزایش دما از ۱۰ به ۱۱ درجه سانتیگراد تأثیر مثبتی بر اجاره دوچرخه و از ۴۰ به ۴۱ تأثیر منفی داشته باشد. این موضوع در بسیاری از نمونه‌ها دیده می‌شود. ویژگی دما تأثیر خطی و مثبتی بر تعداد دوچرخه‌های اجاره‌ای دارد، اما در برخی مواقع بی تأثیر است و حتی در دماهای بالا تأثیر منفی می‌گذارد. مدل خطی به این موضوع، اهمیتی نمی‌دهد و با وظیفه‌شناسی بهترین صفحه خطی را (با به حداقل رساندن فاصله اقلیدسی) پیدا می‌کند.

می‌توانید روابط غیرخطی را با استفاده از یکی از تکنیک‌های زیر مدل سازی کنید:

- تبدیل ساده ویژگی (مثلًاً لگاریتم)

- طبقه بندی ویژگی
- مدل های افزایشی تعمیم یافته (GAMs)

قبل از اینکه به جزئیات هر روش بپردازم، اجازه دهید با مثالی شروع کنیم که هر سه روش را نشان می دهد. من مجموعه داده اجاره دوچرخه را در نظر گرفتم و یک مدل خطی با فقط ویژگی دما برای پیش‌بینی تعداد دوچرخه‌های اجاره‌ای آموزش دادم. شکل زیر شبی برآورد شده را نشان می دهد: مدل خطی استاندارد، مدل خطی با دمای تبدیل شده (لگاریتم)، مدل خطی با دما به عنوان ویژگی طبقه‌بندی شده و با استفاده از اسپلاین‌های^۱ رگرسیون (GAM).



شکل ۵.۱۳: پیش‌بینی تعداد دوچرخه‌های اجاره‌ای تنها با استفاده از ویژگی دما. یک مدل خطی (بالا سمت چپ) به خوبی با داده‌ها مطابقت ندارد. راه حل این است که ویژگی را با لگاریتم ویژگی (بالا سمت راست) جایگزین کنید، یا ویژگی را دسته بندی کنید (پایین سمت چپ)، که معمولاً یک تصمیم اشتباه است، یا استفاده از مدل‌های افزودنی تعمیم یافته که می‌تواند به طور خودکار یک منحنی نرم را برای دما تنظیم کند (پایین سمت راست).

¹ Splines

تبديل ويزگي

أغلب از لگاریتم ویژگی به عنوان تبدیل استفاده می‌شود. استفاده از لگاریتم نشان می‌دهد که هر 10 برابر افزایش دما تأثیر خطی یکسانی بر تعداد دوچرخه‌ها دارد، بنابراین تغییر از 1 به 10 درجه سانتیگراد همان تأثیر تغییر از 1 به 1 درجه سانتیگراد را دارد (که به نظر اشتباه می‌رسد). مثال‌های دیگر برای تبدیل ویژگی عبارت‌اند از: جذر^۱، تابع مربع^۲ و تابع نمایی^۳. استفاده از تبدیل ویژگی به این معنی است که ستون این ویژگی را در داده‌ها با تابعی از ویژگی مانند لگاریتم ویژگی جایگزین می‌کنید و طبق معمول مدل خطی را برازش می‌کنید. برخی از برنامه‌های آماری به شما اجازه می‌دهند که تبدیل‌ها را در فراخوانی مدل خطی مشخص کنید. وقتی ویژگی را تغییر می‌دهید می‌توانید ابتکار داشته باشید. تفسیر ویژگی با توجه‌به تبدیل انتخاب شده تغییر می‌کند. اگر از تبدیل \log استفاده می‌کنید، تفسیر در یک مدل خطی به این صورت می‌شود: "اگر لگاریتم ویژگی یک واحد افزایش یابد، پیش‌بینی به اندازه وزن مربوطه افزایش می‌یابد." وقتی از GLM با تابع اتصال استفاده می‌کنید که تابع همانی نیست، تفسیر پیچیده‌تر می‌شود، زیرا باید هر دو تبدیل را در تفسیر بگنجانید (به جز زمانی که یکدیگر را خنثی می‌کنند، مانند \log و \exp ، که در این حالت تفسیر راحت‌تر می‌شود).

دسته بندی ویژگی‌ها

امکان دیگر برای مواجه با یک اثر غیرخطی، گسسته کردن ویژگی است. این کار، ویژگی را به یک ویژگی طبقه‌بندی تبدیل کنید. به عنوان مثال، می‌توانید ویژگی دما را به 20 بازه با سطوح $[10, -5)$ ، $[5, 0)$ ، و ... تقسیم کنید. هنگامی که شما از دمای طبقه‌بندی شده به جای دمای پیوسته استفاده می‌کنید، مدل خطی یک تابع پله‌ای را تخمین می‌زند زیرا هر سطح تخمین خاص خود را دارد. مشکل این رویکرد این است که به داده‌های بیشتری نیاز دارد، احتمال بیشتری وجود دارد که بیش برازش^۴ رخ دهد و مشخص نیست که چگونه ویژگی را به طور صحیحی گسسته کنیم (فاصله‌های مساوی یا چندک؟ چه تعداد بازه؟). من فقط در صورتی از گسسته سازی استفاده می‌کنم که یک دلیل بسیار قوی برای آن وجود داشته باشد. به عنوان مثال، برای این که بتوان مدل را با مطالعه دیگری مقایسه نمود.

مدل‌های افزایشی تعمیم یافته (GAM)

چرا به مدل خطی (تعمیم یافته) اجازه نمی‌دهیم روابط غیرخطی را یاد بگیرد؟ این انگیزه پشت GAM‌ها است. GAM‌ها این محدودیت را کاهش می‌دهند که رابطه باید یک جمع وزنی ساده باشد، و در عوض فرض می‌کنند

¹ Square root

² Square function

³ Exponential function

⁴ Overfit

که نتیجه می‌تواند با مجموع توابع دلخواه هر ویژگی مدل شود. از نظر ریاضی، رابطه در یک GAM به شکل زیر است:

$$g(E_Y(y|x)) = \beta_0 + f_1(x_1) + f_2(x_2) + \dots + f_p(x_p)$$

فرمول مشابه فرمول GLM است با این تفاوت که عبارت خطی $\beta_j x_j$ با یکتابع انعطاف پذیرتر ($f_j(x_j)$) جایگزین می‌شود. هسته یک GAM هنوز مجموع اثرات ویژگی است، اما شما این گزینه را دارید که اجازه دهد روابط غیرخطی بین برخی ویژگی‌ها و خروجی وجود داشته باشد. اثرات خطی نیز توسط این چارچوب پوشش داده می‌شوند، برای اینکه ویژگی‌ها به صورت خطی مدیریت شوند، می‌توانید تابع $(f_j(x_j))$ را به شکل $\beta_j x_j$ محدود کنید.

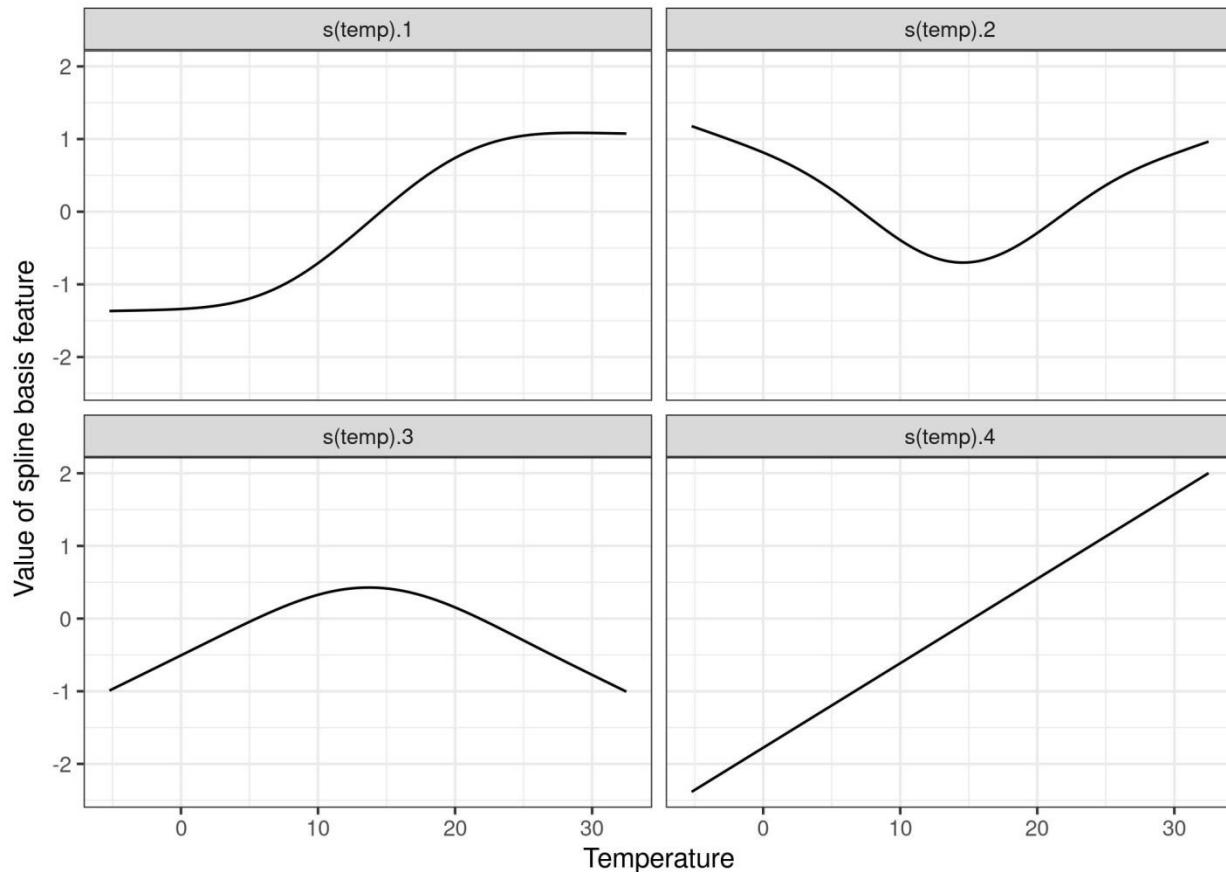
سوال بزرگ این است که چگونه توابع غیرخطی را یاد بگیریم. جواب استفاده از «اسپلاین» یا «توابع اسپلاین» است. اسپلاین‌ها توابعی هستند که از توابع پایه ای ساده‌تر ساخته می‌شوند. اسپلاین‌ها را می‌توان برای تقریب سایر توابع پیچیده‌تر استفاده کرد. کمی شبیه چیدن آجرهای لگو برای ساختن چیزی پیچیده‌تر. راههای گیج کننده ای برای تعریف این توابع پایه اسپلاین وجود دارد. اگر علاقه‌مند به کسب اطلاعات بیشتر در مورد تمام روش‌های تعریف توابع پایه ای هستید، برای شما در سفرтан آرزوی موفقیت می‌کنم. من قصد ندارم در اینجا وارد جزئیات شوم، من فقط قصد دارم یک شهود بسازم. چیزی که شخصاً بیشترین کمک را به من برای درک اسپلاین کرد، تجسم توابع پایه تکی و بررسی چگونگی اصلاح ماتریس داده بود. به عنوان مثال، برای مدل‌سازی دما با اسپلاین، ما ویژگی دما را از داده‌ها حذف می‌کنیم و مثلًاً ۴ ستون را جایگزین آن می‌کنیم که هر کدام یک تابع پایه اسپلاین را نشان می‌دهند. معمولاً توابع پایه ای اسپلاین بیشتری خواهید داشت، من فقط برای تجسم بهتر، تعداد را کاهش دادم. مقدار هر نمونه از این ویژگی‌های جدید پایه اسپلاین به مقادیر دمای نمونه‌ها بستگی دارد. GAM علاوه بر تمام اثرات خطی، وزن‌های اسپلاین را نیز تخمین می‌زنند. GAM‌ها همچنین برای اوزان یک عبارت جریمه در نظر می‌گیرند تا آنها را نزدیک به صفر نگه دارند. این کار به طور موثر، انعطاف پذیری اسپلاین‌ها و برآش بیش از حد را کاهش می‌دهد. سپس یک پارامتر صافی که معمولاً برای کنترل انعطاف پذیری منحنی استفاده می‌شود، از طریق اعتبارسنجی متقطع تنظیم می‌شود. بدون درنظر گرفتن عبارت جریمه، مدل‌سازی غیرخطی با اسپلاین، مهندسی ویژگی‌های فرضی است.

در مثالی که ما تعداد دوچرخه‌ها را با GAM فقط با استفاده از دما پیش‌بینی می‌کنیم، ماتریس ویژگی مدل به این صورت است:

(Intercept)	s(temp).1	s(temp).2	s(temp).3	s(temp).4
1	-0.93	-0.14	0.21	-0.83
1	-0.83	-0.27	0.27	-0.72
1	-1.32	0.71	-0.39	-1.63
1	-1.32	0.70	-0.38	-1.61

1	-1.29	0.58	-0.26	-1.47
1	-1.32	0.68	-0.36	-1.59

هر ردیف نشان دهنده یک نمونه از داده‌ها (یک روز) است. هر ستون پایه اسپلاین حاوی مقدار تابع پایه اسپلاین در مقادیر دمایی خاص است. شکل زیر نشان می‌دهد که این توابع پایه اسپلاین چگونه هستند:

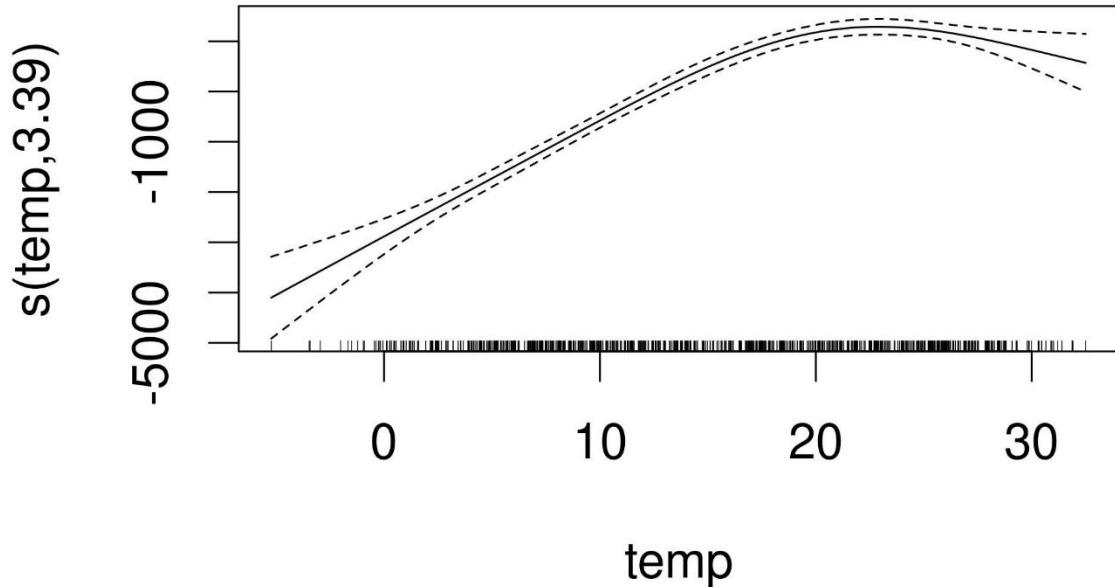


شکل ۵.۱۴: برای مدل سازی هموار اثر دما، از ۴ تابع پایه اسپلاین استفاده می‌کنیم. هر مقدار دما به ۴ مقدار پایه اسپلاین (در اینجا) نگاشت می‌شود. اگر دمای یک نمونه ۳۰ درجه سانتیگراد باشد، مقدار اولین ویژگی پایه اسپلاین ۱- می‌شود، برای دومی $7/0$ ، برای سومی -8 و برای چهارمین $1/7$.

GAM وزن‌هایی را به هر ویژگی اسپلاین پایه‌ای دما اختصاص می‌دهند:

	weight
(Intercept)	4504.35
s(temp).1	989.34
s(temp).2	740.08
s(temp).3	2309.84
s(temp).4	558.27

و منحنی اسپلاین واقعی که از مجموع وزن دار توابع پایه اسپلاین با وزن های تخمینی حاصل می شود، به شکل زیر است:



شکل ۵.۱۵: اثر ویژگی GAM دما برای پیش‌بینی تعداد دوچرخه‌های کرایه شده (دماهی به عنوان تنها ویژگی استفاده شده است).

تفسیر اثرات هموار نیاز به بررسی بصری منحنی برازش داده شده دارد. اسپلاین‌ها معمولاً حول میانگین پیش‌بینی مرکز می‌شوند، بنابراین یک نقطه در منحنی، مقدار تفاوت با پیش‌بینی میانگین است. به عنوان مثال، در دماهی صفر درجه سانتیگراد، تعداد دوچرخه‌های پیش‌بینی شده ۳۰۰۰ واحد کمتر از میانگین پیش‌بینی است.

۵.۳.۴ مزایا

هر تعمیمی از مدل خطی، مقداری جهان را توصیف می‌کنند. هرگاه در مدل‌های خطی با مشکلی مواجه می‌شوید، احتمالاً تعمیمی خواهید یافت که آن را برطرف می‌کند.

بیشتر روش‌ها برای چندین دهه مورد استفاده قرار گرفته اند. به عنوان مثال، GAM‌ها تقریباً ۳۰ سال از عمر شان می‌گذرد. بسیاری از محققان و دست اندرکاران صنعت در استفاده از مدل‌های خطی بسیار با تجربه هستند و این روش‌ها در بسیاری از انجمن‌ها به عنوان وضعیت موجود برای مدل‌سازی پذیرفته شده است.

علاوه بر پیش‌بینی، می‌توانید از مدل‌ها برای استنتاج، نتیجه‌گیری در مورد داده‌ها استفاده کنید (به شرطی که مفروضات مدل نقض نشود). شما در این مدل‌ها، فواصل اطمینان برای وزن‌ها، تست‌های معناداری، فواصل پیش‌بینی و موارد دیگر را دریافت می‌کنید.

نرم‌افزارهای آماری معمولاً دارای رابطه‌های واقعاً خوبی برای برازش با GLM، GAM و مدل‌های خطی خاص‌تر هستند.

مشکل بسیاری از مدل‌های یادگیری ماشین ناشی از این موارد است: ۱) عدم محدودیت^۱، به این معنی که تعداد زیادی از ویژگی‌ها برای پیش‌بینی استفاده می‌شوند، ۲) با ویژگی‌ها به صورت غیرخطی رفتار می‌شوند، به این معنی که برای توصیف اثر به بیش از یک وزن نیاز دارید، ۳) مدل سازی تعاملات بین ویژگی‌ها. با توجه به این که مدل‌های خطی بسیار قابل تفسیر هستند، اما اغلب با واقعیت مناسب نیستند، تعمیمات شرح داده شده در این بخش راه خوبی برای دستیابی به یک انتقال هموار به مدل‌های انعطاف‌پذیرتر می‌باشد تا بتوان همچنان برخی از قابلیت تفسیر را حفظ کرد.

۵.۳.۵ معايب

به عنوان مزیت گفتم که مدل‌های خطی در جهان خودشان زندگی می‌کنند. تعداد روش‌هایی که می‌توانید برای مدل خطی ساده تعمیم یابند بسیار زیاد است. در عمل، جهان‌های موازی متعددی وجود دارد، زیرا بسیاری از انجمن‌های محققان و پزشکان نامهای خاص خود را برای روش‌هایی دارند که کم و بیش یک کار را انجام می‌دهند. این موضوع می‌تواند بسیار گیج‌کننده باشد.

اکثر اصلاحات مدل خطی باعث می‌شود که مدل کمتر قابل تفسیر باشد. هرتابع اتصال (در GLM) که تابع همانی نباشد، تفسیر را پیچیده می‌کند. تعاملات ویژگی‌ها نیز تفسیر را پیچیده می‌کند. تاثیرات ویژگی غیرخطی یا کمتر بصری هستند (مانند تبدیل \log) یا دیگر نمی‌توان آنها را در یک عدد خلاصه کرد (مثلاً توابع اسپیلاین).

GAM‌ها و غیره بر فرضیات مربوط به فرآیند تولید داده تکیه دارند. اگر آنها نقض شوند، دیگر تفسیر اوزان معتبر نیست.

¹ sparseness

عملکرد مدل های ترکیبی درخت پایه مانند درخت گرادیان تقویت شده (gradient tree boosting) در بسیاری موارد بهتر از پیچیده ترین مدل های خطی است. این نکته، قسمتی از تجربه شخصی من و قسمتی از مشاهده کردن مدل های برنده در پلتفرم هایی مانند kaggle.com است.

۵.۳.۶ نرم افزار

تمام مثال های این فصل با استفاده از زبان R پیاده سازی شده اند. برای GAMها از پکیج gam استفاده شد، اما پکیج های متعدد دیگری نیز موجود می باشد. R دارای تعداد بسیار زیادی پکیج برای گسترش مدل های رگرسیون خطی است. بدون پیشی گرفتن از هر زبان تجزیه و تحلیل دیگری، R اولین گزینه برای هر تعمیمی از مدل رگرسیون خطی است. شما پیاده سازی هایی از GAMها را در Python پیدا خواهید کرد (مانند pyGAM)، اما این پیاده سازی ها کامل نیستند.

۵.۳.۷ تعمیمات بیشتر

همان طور که وعده داده بودم، در اینجا لیستی از مشکلاتی که ممکن است در مدل های خطی با آنها مواجه شوید، همراه با نام راه حلی که برای این مشکل وجود دارد، آورده شده است. می توانید نام روش را کپی و در موتور جستجوی مورد علاقه خود فراخوانی کنید.

داده های من فرض مستقل بودن و توزیع یکسان (iid) را نقض می کند.
به عنوان مثال، اندازه گیری های مکرر روی یک بیمار.

جستجوی کنید برای generalized estimating equations یا mixed models
مدل من دارای خطاهای ناهم واریانس^۱ است.

به عنوان مثال، هنگام پیش بینی ارزش یک خانه، خطاهای مدل معمولاً در خانه های گران قیمت بیشتر است، که فرض هم واریانسی مدل خطی را نقض می کند.

جستجوی کنید برای robust regression
نقاط پرتی دارم که به شدت بر مدل من تأثیر می گذارد.

جستجوی کنید برای robust regression
من می خواهم زمان رخ دادن یک رویداد را پیش بینی کنم.

داده های زمان تا رویداد معمولاً با اندازه گیری های سانسور شده ارائه می شوند، به این معنی که در برخی موارد زمان کافی برای مشاهده رویداد وجود نداشت. به عنوان مثال، یک شرکت می خواهد خرابی ماشین های خود را پیش بینی کند، اما فقط اطلاعات دو سال را در اختیار دارد. برخی از ماشین ها پس از دو سال هنوز سالم هستند، اما ممکن است بعداً از کار بیفتدند.

¹ heteroscedastic

جستجوی کنید برای survival analysis و cox regression ،parametric survival models نتیجه من برای پیش‌بینی یک طبقه است.

اگر خروجی دارای دو طبقه است از مدل رگرسیون لجستیک استفاده کنید که احتمال را برای طبقه ها مدل می‌کند.

اگر طبقه‌های بیشتری دارید، جستجو کنید برای multinomial regression رگرسیون لجستیک و رگرسیون چند جمله‌ای هر دو GLM هستند. من می‌خواهم دسته بندی‌های مرتب شده را پیش‌بینی کنم. به عنوان مثال نمرات مدرسه.

جستجوی کنید برای proportional odds model نتیجه من یک شمارش است (مانند تعداد فرزندان در یک خانواده).

جستجوی کنید برای Poisson regression مدل پواسون نیز GLM است. همچنین ممکن است این مشکل را داشته باشید که مقادیر شمارش شده + بسیار زیاد است.

جستجوی کنید برای hurdle model ،(zero-inflated Poisson regression model) من مطمئن نیستم که چه ویژگی‌هایی باید در مدل گنجانده شود تا نتیجه گیری‌های علی درست انجام شود. به عنوان مثال، می‌خواهم بدانم اثر یک دارو بر فشار خون چیست. این دارو بر یکی از شاخص‌های خونی تأثیر مستقیم دارد و این شاخص بر خروجی تأثیر می‌گذارد. آیا باید شاخص خون را در مدل رگرسیون لحاظ کنم؟

جستجوی کنید برای mediation analysis ،causal inference من داده‌های گم شده دارم

جستجوی کنید برای multiple imputation من می‌خواهم دانش قبلی¹ را در مدل‌های خود ادغام کنم.

جستجوی کنید برای Bayesian inference

¹ Prior knowledge

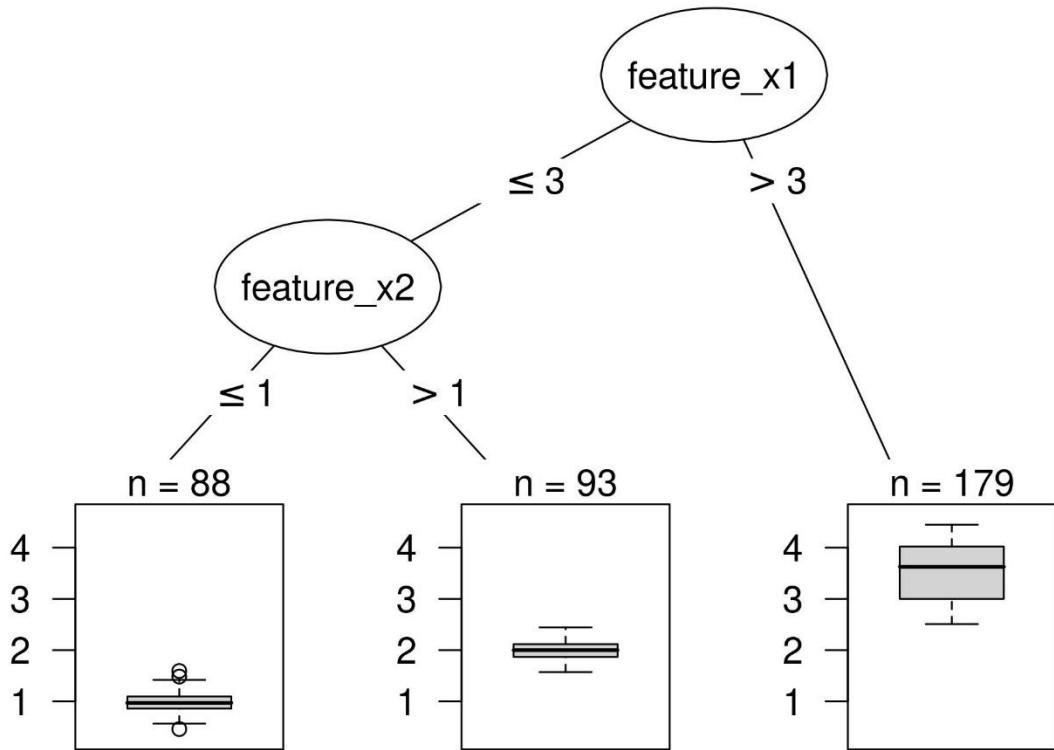
۵.۴ درخت تصمیم

مدل‌های رگرسیون خطی و رگرسیون لجستیک در شرایطی که رابطه بین ویژگی‌ها و خروجی غیرخطی است یا جایی که ویژگی‌ها با یکدیگر تعامل دارند، شکست می‌خورند. زمان درخشش درخت تصمیم است! مدل‌های مبتنی بر درخت، داده‌ها را چندین بار بر اساس مقادیر برش^۱ مشخص در ویژگی‌ها تقسیم می‌کنند. از طریق تقسیم، زیرمجموعه‌های مختلفی از مجموعه‌داده ایجاد می‌شود که هر نمونه متعلق به یکی از این زیرمجموعه‌ها می‌باشد. زیرمجموعه‌های نهایی را گره‌های پایانی یا برگ و زیرمجموعه‌های میانی را گره‌های داخلی یا گره‌های تقسیم می‌نامند. برای پیش‌بینی خروجی در هر گره برگ، از میانگین خروجی داده‌های آموزشی در این گره استفاده می‌شود. درختان را می‌توان برای طبقه‌بندی و رگرسیون استفاده کرد.

الگوریتم‌های مختلفی وجود دارند که می‌توانند یک درخت را گسترش دهند. این الگوریتم‌ها در ساختار ممکن درخت (به عنوان مثال تعداد تقسیم‌ها در هر گره)، معیارهای چگونگی یافتن تقسیم‌ها، زمان توقف تقسیم و نحوه تخمین مدل‌های ساده در گره‌های برگ، متفاوت عمل می‌کنند. الگوریتم طبقه‌بندی و رگرسیون درختان (CART²) احتمالاً محبوب‌ترین الگوریتم برای القای درخت است. ما بر روی CART تمرکز خواهیم کرد، اما تفسیر برای اکثر انواع درختان مشابه است. کتاب "عناصر یادگیری آماری" (Hastie, 2009) را برای معرفی دقیق‌تر CART توصیه می‌کنم.

¹ Cutoff

² classification and regression trees



شکل ۵.۱۶: درخت تصمیم با داده‌های مصنوعی. نمونه‌هایی با مقدار بیشتر از ۳ برای ویژگی x_1 به گره ۵ ختم می‌شوند. نمونه‌های دیگر بسته به اینکه مقادیر ویژگی x_2 از ۱ بیشتر باشد به گره ۳ یا گره ۴ اختصاص داده می‌شوند.

فرمول زیر رابطه بین خروجی y و ویژگی‌های x را توصیف می‌کند.

$$\hat{y} = \hat{f}(x) = \sum_{m=1}^M c_m I\{x \in R_m\}$$

هر نمونه دقیقاً در یک گره برگ ($=$ زیر مجموعه R_m) قرار می‌گیرد. $I\{x \in R_m\}$ تابع همانی است که ۱ بر می‌گرداند اگر x عضو زیرمجموعه R_m باشد و ۰ در غیر این صورت. اگر یک نمونه در یک گره برگ R_l بیفت، خروجی پیش‌بینی شده $C_l = \hat{y}$ است. که در اینجا، C_l میانگین تمام نمونه‌های آموزشی در گره برگ است.

اما زیر مجموعه‌ها از کجا می‌آیند؟ این بسیار ساده است: CART یک ویژگی را می‌گیرد و تعیین می‌کند که کدام نقطه برش واریانس y را برای یک کار رگرسیونی یا شاخص جینی^۱ توزیع کلاس y برای وظایف طبقه‌بندی را به حداقل می‌رساند. واریانس به ما می‌گوید که مقادیر y در یک گره چقدر در اطراف مقدار میانگین خود

^۱ Gini index

پخش می‌شوند. شاخص جینی به ما می‌گوید که یک گره چقدر "ناخالص^۱" است، به عنوان مثال اگر همه کلاس‌ها فراوانی یکسانی داشته باشند، گره ناخالص است، اگر فقط یک کلاس وجود داشته باشد، خالص کامل است. زمانی که نقاط داده در گره‌ها مقادیر بسیار مشابهی برای y داشته باشند، واریانس و شاخص جینی به حداقل می‌رسد. در نتیجه، بهترین نقطه برش، دو زیرمجموعه حاصل را تا حد ممکن با توجه به خروجی هدف متفاوت می‌کند. برای ویژگی‌های طبقه‌بندی، الگوریتم سعی می‌کند با گروه‌بندی‌های مختلف طبقه‌ها، زیرمجموعه‌هایی ایجاد کند. پس از تعیین بهترین برش برای هر ویژگی، الگوریتم ویژگی را برای تقسیم که منجر به بهترین تقسیم از نظر واریانس یا شاخص جینی می‌شود انتخاب می‌کند و این تقسیم را به درخت اضافه می‌کند. الگوریتم این جستجو و تقسیم^۲ را به صورت بازگشتی^۳ در هر دو گره جدید تا رسیدن به یک معیار توقف^۴ ادامه می‌دهد. معیارهای ممکن عبارت‌اند از: حداقل تعداد نمونه‌هایی که باید قبل از تقسیم در یک گره باشند، یا حداقل تعداد نمونه‌هایی که باید در یک گره پایانی باشند.

۵.۴.۱ تفسیر

تفسیر ساده است: با شروع از گره ریشه، به گره‌های بعدی می‌روید و لبه‌ها به شما می‌گویند که به کدام زیر مجموعه‌ها نگاه می‌کنید. هنگامی که به گره برگ رسیدید، گره خروجی پیش‌بینی شده را به شما می‌گوید. تمام لبه‌ها با "AND" به هم متصل می‌شوند.

الگو: اگر ویژگی x (کوچک‌تر/بزرگ‌تر) از آستانه c و ... باشد، نتیجه پیش‌بینی شده، میانگین مقدار y نمونه‌های آن گره است.

اهمیت ویژگی

اهمیت کلی یک ویژگی در یک درخت تصمیم را می‌توان به روش زیر محاسبه کرد: تمام تقسیم‌بندی‌هایی را که این ویژگی برای آنها استفاده شده است را بررسی کنید و اندازه بگیرید که چقدر واریانس یا شاخص جینی را در مقایسه با گره والد کاهش داده است. مجموع همه اهمیت‌ها به ۱۰۰ مقياس می‌شود. این بدان معنی است که هر اهمیت را می‌توان به عنوان سهمی از اهمیت کلی مدل تفسیر کرد.

تجزیه درخت^۵

پیش‌بینی‌های فردی یک درخت تصمیم را می‌توان با تجزیه مسیر تصمیم به یک جزء در هر ویژگی توضیح داد. می‌توانیم یک تصمیم را از طریق درخت ردیابی کنیم و یک پیش‌بینی را با مشارکت‌های اضافه شده در هر گره تصمیم توضیح دهیم.

¹ Impure

² Search-and-split

³ recursively

⁴ stop criterion

⁵ Tree decomposition

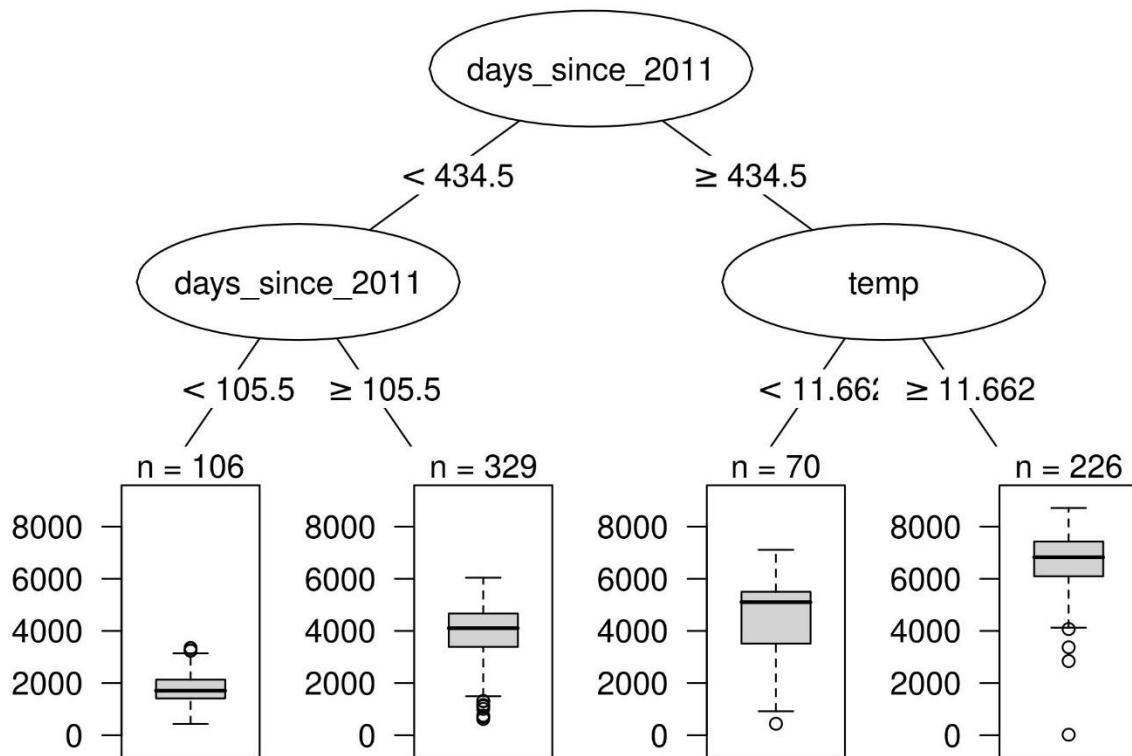
گره ریشه در درخت تصمیم نقطه شروع ما است. اگر بخواهیم از گره ریشه برای پیش‌بینی استفاده کنیم، میانگین نتیجه داده‌های آموزشی را پیش‌بینی می‌کند. با تقسیم بعدی، بسته به گره بعدی در مسیر، یا کم می‌کنیم یا یک جمله به این جمع اضافه می‌کنیم. برای رسیدن به پیش‌بینی نهایی، باید مسیر نمونه داده‌ای را که می‌خواهیم توضیح دهیم دنبال کنیم و مدام به فرمول اضافه کنیم.

$$\hat{f}(x) = \bar{y} + \sum_{d=1}^D split \cdot contrib(d, x) = \bar{y} + \sum_{j=1}^p feat \cdot contrib(j, x)$$

پیش‌بینی یک نمونه بخصوص، میانگین خروجی هدف به اضافه مجموع تمام مشارکت‌های تقسیم‌های D است که بین گره ریشه و گره پایانی که در آن نمونه به پایان می‌رسد، رخ می‌دهد. اگرچه ما به مشارکت‌های تقسیم‌شده علاقه‌ای نداریم، بلکه به مشارکت‌های ویژگی علاقه‌مندیم. یک ویژگی ممکن است برای بیش از یک تقسیم استفاده شود یا اصلاً استفاده نشود. می‌توانیم مشارکت‌ها را برای هر یک از ویژگی‌های p اضافه کنیم و تفسیری از میزان مشارکت هر ویژگی در یک پیش‌بینی دریافت کنیم.

۵.۴.۲ مثال

اجازه دهید نگاهی دیگر به داده‌های اجاره دوچرخه داشته باشیم. می‌خواهیم تعداد دوچرخه‌های کرايه شده در یک روز خاص را با درخت تصمیم پیش‌بینی کنیم. درخت آموخته شده به شکل زیر است:

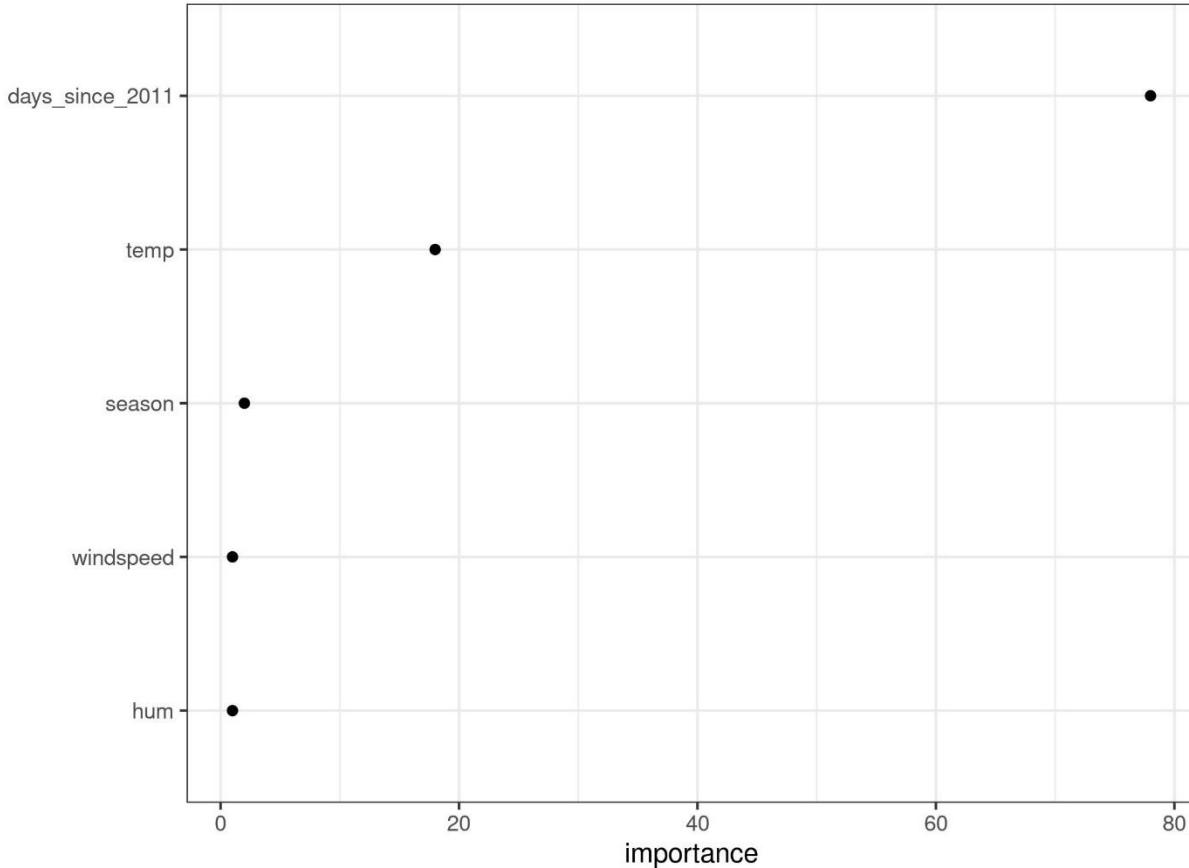


شکل ۵.۱۷: درخت رگرسیون بر روی داده‌های اجاره دوچرخه بازنشر شده است. حداقل عمق مجذب برای درخت روی ۲ تنظیم شد. ویژگی‌های روند (temp) و دما (days since 2011) برای تقسیمات انتخاب شده‌اند. نمودارهای باکس پلات توزیع تعداد دوچرخه را در گره پایانی نشان می‌دهد.

تقسیم اول و یکی از دو تقسیم دوم با ویژگی روند انجام شد که از زمان شروع جمع‌آوری داده‌ها، روزها را شمارش می‌کند و نشان می‌دهد که سرویس اجاره دوچرخه در طول زمان محبوب‌تر شده است. برای روزهای قبل از روز ۱۰۵، تعداد دوچرخه‌های پیش‌بینی شده حدود ۱۸۰۰ دوچرخه است، بین روزهای ۱۰۶ و ۴۳۰ حدود ۳۹۰۰ است. برای روزهای بعد از روز ۴۳۰، پیش‌بینی ۴۶۰۰ (اگر دما زیر ۱۲ درجه باشد) یا ۶۶۰۰ (اگر دما بالای ۱۲ درجه باشد) است.

اهمیت ویژگی به ما می‌گوید که یک ویژگی چقدر به بهبود خلوص همه گره‌ها کمک کرده است. در اینجا، واریانس استفاده می‌شود، زیرا پیش‌بینی اجاره دوچرخه یک کار رگرسیونی است.

درخت بصری شده، نشان می‌دهد که هم دما و هم روند زمانی برای شکاف‌ها استفاده شده است، اما مشخص نمی‌کند که کدام ویژگی مهم‌تر است. اندازه گیری اهمیت ویژگی نشان می‌دهد که روند زمانی بسیار مهم‌تر از دما است.



شکل ۵.۱۸: اهمیت ویژگی‌ها اندازه گیری می کند که چه مقدار خلوص گره به طور متوسط بهبود یافته است.

۵.۴.۳ مزايا

ساختار درختی برای ثبت تعاملات بین ویژگی‌ها در داده‌ها ایده آل است.

داده‌ها به **گروه‌های مجزا** ختم می‌شوند که درک آن‌ها اغلب آسان‌تر از نقاط روی یک ابر صفحه چند بعدی مانند آن چه در رگرسیون خطی اتفاق می‌افتد، می‌باشد. مسلماً تفسیر بسیار ساده است.
ساختار درختی با گره‌ها و لبه‌هایش **تجسم طبیعی** نیز دارد.

درختان **توضیحات خوبی** را همان‌طور که در فصل "توضیحات انسان پسند" تعریف شده است، ایجاد می‌کنند. ساختار درختی به‌طور خودکار باعث می‌شود تا در مورد مقادیر پیش‌بینی‌شده برای نمونه‌های فردی به‌عنوان خلاف واقع فکر کنیم: «اگر یک ویژگی بزرگ‌تر/اکوچک‌تر از نقطه تقسیم بود، پیش‌بینی به جای y_2 ، y_1 بود». توضیحات درخت مقابله‌ای هستند، زیرا همیشه می‌توانید پیش‌بینی یک نمونه را با ستاریوهای مربوط به «چه می‌شود» (همان‌طور که توسط درخت تعریف می‌شود) که این چه می‌شود همان سایر گره‌های برگ درخت هستند، مقایسه کنید. اگر درخت کوتاه باشد، مانند عمق یک تا سه تقسیم، توضیحات حاصل انتخابی است. درختی با عمق سه به حداقل سه ویژگی و نقطه تقسیم نیاز دارد تا توضیحی برای پیش‌بینی یک نمونه فردی

ایجاد کند. صادق بودن پیش‌بینی به عملکرد پیش‌بینی درخت بستگی دارد. توضیحات مربوط به درختان کوتاه بسیار ساده و کلی است.

نیازی به تغییر ویژگی‌ها نیست. در مدل‌های خطی، گاهی اوقات لازم است لگاریتم یک ویژگی را در نظر بگیرید. یک درخت تصمیم با هر تبدیل یکنواخت یک ویژگی به یک اندازه خوب کار می‌کند.

۵.۴.۴ معایب

درختان در مواجه با روابط خطی شکست می‌خورند. هر رابطه خطی بین یک ویژگی ورودی و خروجی باید با تقسیم‌بندی تقریب زده شود و یکتابع پله‌ای ایجاد کرد. این کار مناسب روابط خطی نیست.

این کار باعث، پله‌پله شدن و عدم نرمی است. تغییرات جزئی در ویژگی ورودی می‌تواند تأثیر زیادی بر خروجی پیش‌بینی شده داشته باشد که معمولاً مطلوب نیست. درختی را تصور کنید که ارزش یک خانه را پیش‌بینی می‌کند و درخت از اندازه خانه به عنوان یکی از ویژگی‌های تقسیم استفاده می‌کند. تقسیم در $100/5$ متر مربع رخ می‌دهد. تصور کنید که کاربر یک تخمین گر قیمت خانه، از مدل درخت تصمیم شما استفاده می‌کند: کاربر خانه خود را اندازه می‌گیرد، به این نتیجه می‌رسند که خانه ۹۹ متر مربع است، آن را وارد ماشین حساب قیمت می‌کنند و ۲۰۰۰۰ یورو را پیش‌بینی می‌کنند. کاربر متوجه می‌شود که فراموش کرده است یک انبار کوچک ۲ متر مربعی را اندازه گیری کنند. اتاق انبار دارای یک دیوار شیبدار است، بنابراین آنها مطمئن نیستند که آیا می‌توانند تمام مساحت را اضافه کنند یا فقط نیمی از آن را. بنابراین آنها تصمیم می‌گیرند هر دو ۱۰۰.۰ و ۱۰۱.۰ متر مربع را امتحان کنند. نتایج: ماشین حساب قیمت خروجی ۲۰۰۰۰ یورو و ۲۰۵۰۰ یورو دارد.

درختان نیز کاملاً ناپایدار هستند. چند تغییر در مجموعه‌داده آموزشی می‌تواند درختی کاملاً متفاوت ایجاد کند. این به این دلیل است که هر تقسیم به تقسیم والد بستگی دارد. و اگر یک ویژگی متفاوت به عنوان اولین ویژگی تقسیم انتخاب شود، کل ساختار درخت تغییر می‌کند. اگر ساختار به این راحتی تغییر کند، اطمینانی در مدل ایجاد نمی‌کند.

درختان تصمیم بسیار قابل تفسیر هستند تا زمانی که کوتاه باشند. تعداد گره‌های پایانی به سرعت با عمق افزایش می‌یابد. هر چه گره‌های پایانی بیشتر و درخت عمیق‌تر باشد، درک قوانین تصمیم گیری درخت دشوارتر می‌شود. عمق ۱ به معنای ۲ گره پایانی است. عمق ۲ به معنای حداقل ۴ گره است. عمق ۳ به معنای حداقل ۸ است. حداقل تعداد گره‌های پایانی در یک درخت به توان ۲ عمق است.

۵.۴.۵ نرم افزار

برای مثال‌های این بخش، از پکیج rpart نرم افزار R استفاده کردم که CART را پیاده‌سازی می‌کند. CART در بسیاری از زبان‌های برنامه نویسی از جمله Python پیاده‌سازی شده است. مسلماً CART یک الگوریتم بسیار قدیمی و تا حدودی منسخ شده است و الگوریتم‌های جدید و جالبی برای برآش درختان وجود دارد. می‌توانید خلاصه‌ای از پکیج‌های نرم افزار R برای درخت‌های تصمیم‌گیری را در Machine Learning and Statistical Learning CRAN Task View¹ در زیر کلمه کلیدی "Recursive Partitioning" بباید. در Python، پکیج imodels الگوریتم‌های مختلفی را برای رشد درخت‌های تصمیم‌گیری (به عنوان مثال برآش حریصانه^۲ در مقابل بهینه^۳، هرس^۴ درختان و تنظیم کردن^۵ درختان ارائه می‌کند.

¹ <https://cran.r-project.org/web/views/MachineLearning.html>

² greedy

³ optimal

⁴ pruning

⁵ regularizing

۵.۵ قوانین تصمیم

قانون تصمیم گیری یک دستور IF-THEN ساده است که از یک شرط (مقدم^۱ نیز نامیده می‌شود) و یک پیش‌بینی تشکیل شده است. به عنوان مثال: اگر امروز باران باراد و اگر آوریل باشد (شرط)، پس فردا باران خواهد آمد (پیش‌بینی). یک قانون تصمیم می‌تواند منفرد یا ترکیبی از چندین قانون، می‌تواند برای پیش‌بینی استفاده شود.

قوانين تصمیم گیری از یک ساختار کلی پیروی می‌کنند: اگر شرایط برآورده شد، پیش‌بینی خاصی انجام می‌شود. قوانین تصمیم گیری احتمالاً قابل تفسیرترین مدل‌های پیش‌بینی هستند. ساختار IF-THEN آنها از نظر معنایی شبیه زبان طبیعی و طرز تفکر ما است، مشروط بر اینکه شرط از ویژگی‌های قابل فهم ساخته شده باشد، طول شرط کوتاه باشد (تعداد کمی از جفت‌ها feature=value با یک AND ترکیب شده باشد) و قوانین زیادی وجود نداشته باشند. در برنامه نویسی، نوشتن قوانین IF-THEN بسیار متداول است. موضوع جدید در یادگیری ماشین این است که قوانین تصمیم گیری از طریق یک الگوریتم، آموزش داده می‌شوند.

تصور کنید از یک الگوریتم برای یادگیری قوانین تصمیم برای پیش‌بینی ارزش یک خانه (low, medium or high) استفاده می‌کنید. یک قانون تصمیم که توسط این مدل آموزش داده می‌شود می‌تواند این باشد: اگر خانه ای بزرگ‌تر از ۱۰۰ متر مربع باشد و دارای باغ باشد، ارزش آن زیاد است. به صورت رسمی‌تر

$$\text{IF size} > 100 \text{ AND garden} = 1 \text{ THEN value} = \text{high}$$

جازه دهید قانون تصمیم را بشکنیم:

- شرط اول در بخش IF است.
- شرط دوم در قسمت IF است.
- این دو شرط با یک "AND" متصل می‌شوند تا یک شرط جدید ایجاد کنند. برای اعمال قانون، هر دو باید برقرار باشند.
- خروجی پیش‌بینی شده (THEN-part) $\text{value} = \text{high}$ است.

یک قانون تصمیم حداقل از یک عبارت $\text{feature} = \text{value}$ در شرط استفاده می‌کند. بدون هیچ محدودیتی می‌توان چند عبارت دیگر را با «AND» اضافه کرد. یک استثنای قانون پیش‌فرض (default rule) است که بخش IF صریح ندارد و زمانی اعمال می‌شود که هیچ قانون دیگری اعمال نشود، اما بعداً در مورد آن بیشتر توضیح خواهیم داد.

میزان سودمندی یک قانون تصمیم معمولاً در دو عدد خلاصه می‌شود: پشتیبانی (Support) و دقت (accuracy).

¹ antecedent

پشتیبانی یا پوشش (coverage) یک قانون: درصد نمونه‌هایی که شرط یک قانون در مورد آنها اعمال می‌شود، پشتیبانی نامیده می‌شود. به عنوان مثال قانون $size = big \text{ AND } location = good \text{ THEN } value = high$ در پیش‌بینی ارزش خانه را در نظر بگیرید. فرض کنید ۱۰۰ خانه از ۱۰۰۰ خانه بزرگ و در مکان مناسبی هستند، پس پشتیبانی قانون ۱۰٪ است. پیش‌بینی (THEN-part) برای محاسبه پشتیبانی مهم نیست.

دقت یا اطمینان (confidence) یک قانون: دقت یک قانون، معیاری است از میزان دقت قانون در پیش‌بینی کلاس صحیح برای نمونه‌هایی که شرط قانون در مورد آنها اعمال می‌شود. به عنوان مثال: فرض کنید از ۱۰۰ خانه که در آن قانون $size = big \text{ AND } location = good \text{ THEN } value = high$ صدق می‌کند، ۸۵٪ خانه $value = low$ و ۱۴٪ خانه $value = medium$ دارند، سپس دقت قانون ۸۵٪ است.

معمولًاً بین دقت و پشتیبانی تعادل وجود دارد: با افزودن ویژگی‌های بیشتر به شرایط، می‌توانیم به دقت بالاتری دست پیدا کنیم، اما در این حالت پشتیبانی کم می‌شود.

برای ایجاد یک طبقه‌بند خوب برای پیش‌بینی ارزش یک خانه، ممکن است لازم باشد نه تنها یک قانون، بلکه ۱۰ یا ۲۰ قانون را یاد بگیرید. پس همه چیز پیچیده‌تر می‌شود و می‌توانید با یکی از مشکلات زیر مواجه شوید:

- قوانین می‌توانند همپوشانی داشته باشند: اگر بخواهم ارزش یک خانه را پیش‌بینی کنم و دو یا چند قانون برقرار باشند و پیش‌بینی‌های متناقضی به من بدهند، چه کنم؟
- هیچ قانون ای برقرار نمی‌شود: اگر بخواهم ارزش یک خانه را پیش‌بینی کنم و هیچ یک از قوانین برقرار نشود، چه کنم؟

دو استراتژی اصلی برای ترکیب قوانین چندگانه وجود دارد: لیست های تصمیم^۱ (مرتب شده) و مجموعه های تصمیم^۲ (غیر مرتب شده). هر استراتژی راه حل متفاوتی برای مشکل همپوشانی قوانین ایجاد می‌کند. فهرست تصمیم گیری، ترتیبی برای قوانین تصمیم گیری ارائه می‌کند. اگر شرط قانون اول برای نمونه ای درست باشد، از پیش‌بینی قانون اول استفاده می‌کنیم. اگر نه، به قانون بعدی می‌رویم و بررسی می‌کنیم که آیا برقرار است و غیره. لیست‌های تصمیم مشکل همپوشانی قوانین را تنها با برگرداندن اولین پیش‌بینی در لیست که قانون در مورد آن برقرار است، حل می‌کند.

یک مجموعه تصمیم شبیه دموکراسی قوانین است، با این تفاوت که برخی از قوانین ممکن است قدرت را بالاتری داشته باشند. در یک مجموعه، قوانین یا منحصر به فرد هستند، یا یک استراتژی برای حل تعارض وجود

¹ Decision lists

² Decision sets

دارد، مانند رأی اکثریت^۱، که ممکن است باعث وزن دهی براساس دقت قانون، یا هر اندازه کیفی دیگری شود. تفسیر پذیری هنگامی که چندین قانون برقرار باشد، مستعد آسیب پذیری می باشد.

هم لیست‌های و هم مجموعه‌های تصمیم می‌توانند از این مشکل رنج ببرند که هیچ قانونی برای یک نمونه خاص برقرار نباشد. این مشکل را می‌توان با معرفی یک قانون پیش فرض حل کرد. قانون پیش فرض، زمانی اعمال می‌شود که هیچ قانون دیگری برقرار نیست. پیش‌بینی قانون پیش‌فرض اغلب رایج‌ترین کلاس از نقاط داده است که توسط قوانین دیگر پوشش داده نمی‌شوند. اگر مجموعه یا فهرستی از قوانین کل فضای ویژگی را پوشش دهد، آن را اشباع^۲ می‌نامیم. با افزودن یک قانون پیش فرض، یک مجموعه یا فهرست به طور خودکار اشباع می‌شود.

راه‌های زیادی برای یادگیری قوانین از داده‌ها وجود دارد و این کتاب به دنبال پوشش همه آنها نیست. این فصل سه مورد از آنها را برای شما ارائه می‌دهد. الگوریتم‌های انتخاب شده‌اند تا پوشش طیف وسیعی از ایده‌های کلی برای یادگیری قوانین را نشان دهند، بنابراین هر مورد، رویکردی بسیار متفاوتی را بیان می‌کنند.

۱- **OneR** قوانین را از یک ویژگی می‌آموزد. OneR به سادگی قابل تفسیر بودنش شناخته می‌شود و از آن به عنوان یک بنج مارک استفاده می‌شود.

۲- **پوشش ترتیبی**^۳ یک روش کلی است که به طور مکرر قوانین را یاد می‌گیرد و نقاط داده ای را که توسط قانون جدید پوشش داده می‌شوند، حذف می‌کند. این روش توسط بسیاری از الگوریتم‌های یادگیری قوانین استفاده می‌شود.

۳- **لیست‌های قانون بیزی**^۴ الگوهای مکرر از پیش استخراج شده^۵ را با استفاده از آمار بیزی در یک لیست تصمیمی، ترکیب می‌کند. استفاده از الگوهای از پیش استخراج شده، یک رویکرد رایج است که توسط بسیاری از الگوریتم‌های یادگیری قوانین استفاده می‌شود. بیایید با ساده‌ترین رویکرد شروع کنیم: استفاده از بهترین ویژگی برای یادگیری قوانین.

۵.۵ یادگیری قوانین از یک ویژگی واحد (OneR)

الگوریتم OneR پیشنهاد شده توسط Holte (1993) یکی از ساده‌ترین الگوریتم‌های القای قانون است. این الگوریتم از بین تمام ویژگی‌ها، یکی را انتخاب می‌کند که بیشترین اطلاعات را در مورد خروجی مورد نظر دارد و قوانین تصمیم گیری را از این ویژگی ایجاد می‌کند.

¹ majority voting

² exhaustive

³ Sequential covering

⁴ Bayesian Rule Lists

⁵ pre-mined frequent patterns

علیرغم نام OneR که مخفف "One Rule" است، الگوریتم بیش از یک قانون تولید می‌کند: این نامگذاری به این علت است که این الگوریتم یک قانون برای هر مقدار ویژگی منحصر به فرد از بهترین ویژگی انتخاب شده، دارد. نام بهتر OneFeatureRules است.

الگوریتم ساده و سریع است:

۱- با انتخاب فواصل مناسب، ویژگی‌های پیوسته را گسسته کنید.

۲- برای هر ویژگی:

- یک جدول متقاطع بین مقادیر ویژگی و خروجی (طبقه ای) ایجاد کنید.

برای هر مقدار از ویژگی، یک قانون ایجاد کنید که پرتکرارترین کلاس نمونه‌هایی را که دارای

این مقدار ویژگی خاص هستند، پیش‌بینی می‌کند (از جدول متقاطع قابل خواندن است).

- خطای کل قوانین مربوط به ویژگی را محاسبه کنید.

۳- ویژگی با کمترین خطای کل را انتخاب کنید.

همیشه تمام نمونه‌های مجموعه‌داده را تحت پوشش قرار می‌دهد، زیرا از تمام سطوح ویژگی انتخاب شده استفاده می‌کند. مقادیر گم شده را می‌توان به عنوان یک مقدار ویژگی اضافی در نظر گرفت یا از قبل آنها را پر کرد.

یک مدل OneR یک درخت تصمیم است که تنها یک تقسیم دارد. تقسیم مانند CART باینری نیست، بلکه به تعداد مقادیر ویژگی منحصر به فرد می‌باشد.

اجازه دهید با ذکر مثالی چگونگی انتخاب بهترین ویژگی توسط OneR توضیح داده شود. جدول زیر مجموعه داده‌های مصنوعی در مورد خانه‌ها، با اطلاعاتی در مورد ارزش، مکان، اندازه و اینکه آیا حیوانات خانگی مجاز هستند را نشان می‌دهد. ما علاقه‌مند به یادگیری یک مدل ساده برای پیش‌بینی ارزش یک خانه هستیم.

location	size	pets	value
good	small	yes	high
good	big	no	high
good	big	no	high
bad	medium	no	medium
good	medium	only cats	medium
good	small	only cats	medium
bad	medium	yes	medium
bad	small	yes	low
bad	medium	yes	low
bad	small	no	low

جدوال متقابل را بین هر ویژگی و نتیجه ایجاد می‌کند:

	value=low	value=medium	value=high
location=bad	3	2	0
location=good	0	2	3

	value=low	value=medium	value=high
size=big	0	0	2
size=medium	1	3	0
size=small	2	1	1

	value=low	value=medium	value=high
pets=no	1	1	2
pets=only cats	0	2	0
pets=yes	2	1	1

برای هر ویژگی، ردیف به ردیف جدول را مرور می‌کنیم: هر مقدار ویژگی، قسمت IF یک قانون است. رایج‌ترین کلاس برای نمونه‌هایی با این مقدار ویژگی، پیش‌بینی، قسمت THEN از قانون است. به عنوان مثال، ویژگی اندازه با سطوح small، medium و big منجر به سه قانون می‌شود. برای هر ویژگی، نرخ کل خطای قوانین تولید شده را محاسبه می‌کنیم که مجموع خطاهای است. ویژگی مکان دارای مقادیر ممکن bad و good است. بیشترین مقدار برای خانه‌هایی که در مکان‌های بد قرار دارند low است و وقتی از low به عنوان پیش‌بینی استفاده می‌کنیم، در دو نمونه اشتباه مرتکب می‌شویم، زیرا دو خانه دارای ارزش medium هستند. ارزش پیش‌بینی شده خانه‌ها در مکان‌های خوب، high است و دوباره ما اشتباه دو اشتباه می‌شویم، زیرا دو خانه دارای ارزش medium هستند. خطای ما با استفاده از ویژگی مکان $4/10$ ، برای ویژگی اندازه $3/10$ و برای ویژگی حیوان خانگی $4/10$ است. ویژگی اندازه، قوانینی با کمترین خطا تولید می‌کند و برای مدل نهایی OneR استفاده می‌شود:

IF size = small THEN value = low

IF size = medium THEN value = medium

IF size = big THEN value = high

OneR ویژگی‌هایی را با سطوح ممکن زیاد ترجیح می‌دهد، زیرا این ویژگی‌ها می‌توانند راحت‌تر به هدف، بیش برآش داده شوند. مجموعه‌داده ای را تصور کنید که فقط شامل نویز است و سیگنالی وجود ندارد، به این معنی که همه ویژگی‌ها مقادیر تصادفی می‌گیرند و هیچ ارزش پیش‌بینی کننده ای برای هدف ندارند. برخی از ویژگی‌ها سطوح بیشتری نسبت به سایر ویژگی‌ها دارند. در این شرائط، ویژگی‌های با سطوح بیشتر، می‌توانند به راحتی باعث بیش برآش شوند. یک ویژگی که یک سطح جداگانه برای هر نمونه از داده‌ها دارد، می‌تواند کل

مجموعه‌داده آموزشی را کاملاً پیش‌بینی کند. یک راه حل این است که داده‌ها را به مجموعه‌های آموزشی^۱ و ارزیابی^۲ تقسیم کنید، یادگیری قوانین با استفاده از داده‌های آموزشی و خطای کل برای انتخاب ویژگی با استفاده از مجموعه ارزیابی انجام شود.

مشکل دیگر، تساوی‌ها^۳ است، یعنی زمانی که دو ویژگی منجر به یک خطای کل یکسان شوند. OneR مشکل تساوی‌ها را یا با استفاده از اولین ویژگی با کمترین خطا یا ویژگی با کمترین p -value در یک تست chi-squared حل می‌کند.

مثال

اجازه دهید OneR را با داده‌های واقعی امتحان کنیم. ما از کار طبقه‌بندی سرطان دهانه رحم برای آزمایش الگوریتم OneR استفاده می‌کنیم. همه ویژگی‌های ورودی پیوسته در ۵ کمیت خود گستته شدند. قوانین زیر ایجاد می‌شود:

Age	prediction
(12.9,27.2]	Healthy
(27.2,41.4]	Healthy
(41.4,55.6]	Healthy
(55.6,69.8]	Healthy
(69.8,84.1]	Healthy

ویژگی سن توسط OneR به عنوان بهترین ویژگی پیش‌بینی انتخاب شد. از آنجایی که سرطان نادر است، برای هر قانون، طبقه اکثریت و بنا براین برچسب پیش‌بینی شده همیشه سالم است، که مفید نیست. استفاده از پیش‌بینی برچسب در این مورد نامتعادل منطقی نیست. جدول متقارع بین فواصل سنی و سرطان/سالم همراه با درصد زنان مبتلا به سرطان گویاتر تر است:

	# Cancer	# Healthy	P(Cancer)
Age=(12.9,27.2]	26	477	0.05
Age=(27.2,41.4]	25	290	0.08
Age=(41.4,55.6]	4	31	0.11
Age=(55.6,69.8]	0	1	0.00
Age=(69.8,84.1]	0	4	0.00

اما قبل از شروع تفسیر هر چیزی: از آنجایی که پیش‌بینی هر ویژگی و هر مقدار سالم است، میزان خطای کل برای همه ویژگی‌ها یکسان است. مشکل تساوی‌ها در خطای کل، به طور پیش‌فرض، با استفاده از اولین ویژگی

¹ Training

² Evaluation

³ Ties

از ویژگی‌هایی که کمترین میزان خطا را دارند (در اینجا، همه ویژگی‌ها دارای خطای ۵۵/۸۵۸ هستند)، که ویژگی Age است، حل می‌شود.

برای موارد رگرسیونی کار نمی‌کند. اما می‌توانیم یک وظیفه رگرسیونی را با گستته کردن خروجی پیوسته به فواصل به یک وظیفه طبقه‌بندی تبدیل کنیم. ما از این ترفند برای پیش‌بینی تعداد دوچرخه‌های اجاره‌ای با OneR استفاده می‌کنیم و تعداد دوچرخه‌ها را به چهار چارک (۰-۲۵٪، ۲۵-۵۰٪، ۵۰-۷۵٪ و ۷۵-۱۰۰٪) تقسیم می‌کنیم. جدول زیر ویژگی انتخاب شده را پس از برآش مدل OneR نشان می‌دهد:

mnth	prediction
JAN	[22,3152]
FEB	[22,3152]
MAR	[22,3152]
APR	(3152,4548]
MAY	(5956,8714]
JUN	(4548,5956]
JUL	(5956,8714]
AUG	(5956,8714]
SEP	(5956,8714]
OCT	(5956,8714]
NOV	(3152,4548]
DEC	[22,3152]

ویژگی انتخاب شده ماه است. ویژگی ماه دارای ۱۲ سطح ویژگی است که بیشتر از سایر ویژگی‌ها می‌باشد. بنابراین خطر بیش برآش وجود دارد. از جنبه خوش‌بینانه‌تر: ویژگی ماه می‌تواند روند فصلی را کنترل کند (مثلاً دوچرخه‌های اجاره‌ای کمتر در زمستان) و پیش‌بینی‌ها معقول به نظر می‌رسند.

اکنون از الگوریتم ساده OneR به رویه‌ای پیچیده‌تر با استفاده از قوانین با شرایط پیچیده‌تر متشکل از چندین ویژگی حرکت می‌کنیم: پوشش متوالی.

۵.۵.۲ پوشش ترتیبی^۱

پوشش ترتیبی یک روش کلی است که به طور مکرر یک قانون واحد را برای ایجاد یک لیست (یا مجموعه) تصمیم می‌آموزد که این لیست (یا مجموعه) کل مجموعه داده‌ها را پوشش می‌دهد. بسیاری از الگوریتم‌های یادگیری قوانین، گونه‌هایی از الگوریتم پوشش متوالی هستند. در این بخش روند کلی و RIPPER معرفی می‌گردد که نوعی از الگوریتم پوشش ترتیبی است که برای مثال‌ها استفاده می‌شوند.

ایده این روش، ساده است: ابتدا یک قانون خوب پیدا کنید که برای برخی از نقاط داده صادق است. تمام نقاط داده ای که توسط قانون پوشش داده شده‌اند را حذف کنید. یک نقطه داده زمانی پوشش داده می‌شود که

¹ Sequential Covering

شرايط برای آن برقرار باشد، صرفنظر از اينکه آيا نقاط به درستی طبقه‌بندی شده‌اند یا خير. آموزش قوانين و حذف نقاط تحت پوشش را با نقاط باقیمانده تکرار کنيد تا زمانی که نقطه دیگری باقی نماند یا شرط توقف دیگری برآورده شود. نتيجه، يك لیست تصمیم است. این رویکرد يادگیری مکرر قوانین و حذف نقاط داده تحت پوشش "دسته بندی و حل^۱" نامیده می‌شود.

فرض کنيد الگوريتم داريم که می‌تواند يك قانون واحد ايجاد کند که بخشی از داده‌ها را پوشش می‌دهد. الگوريتم پوشش ترتیبی برای دو کلاس (يکی مثبت، يکی منفی) بهصورت زیر عمل می‌کند:

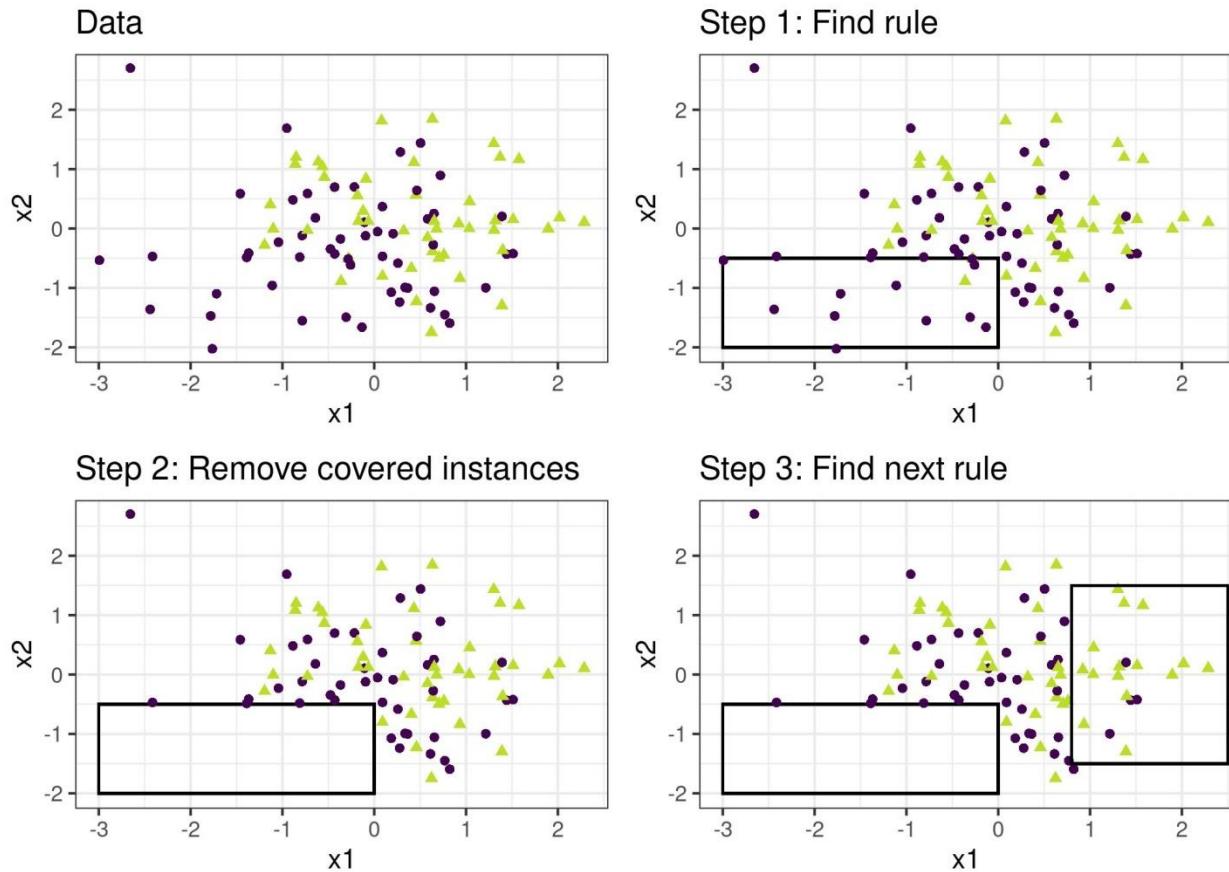
- با يك لیست خالی از قوانین (rlist) شروع کنيد.
- يك قانون را ياد بگيريد.

تا هنگامی که لیست قوانین، کيفيتی کمتر از يك آستانه کيفيت مشخص دارد (يا نمونه‌های مثبت هنوز پوشش داده نشده اند):

- قانون r را به rlist اضافه کنيد.
- تمام نقاط داده تحت پوشش قانون r را حذف کنيد.
- قانون دیگري را در مورد داده‌های باقی مانده بیاموزيد.

- لیست تصمیم را ارائه دهيد.

¹ separate-and-conquer



شکل ۵.۱۹: الگوریتم پوشش با پوشش ترتیبی فضای ویژگی با قوانین واحد و حذف نقاط داده ای که قبلاً توسط آن قوانین پوشش داده شده‌اند، کار می‌کند. به منظور بصری سازی، ویژگی‌های x_1 و x_2 پیوسته هستند، اما اکثر الگوریتم‌های یادگیری قوانین به ویژگی‌های طبقه ای نیاز دارند.

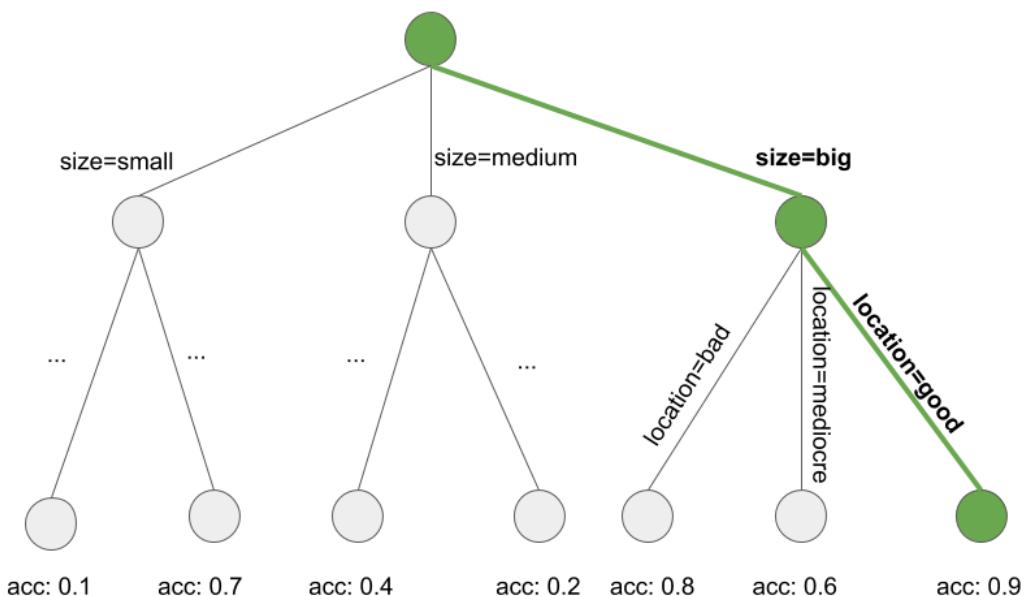
به عنوان مثال: ما یک مجموعه داده برای پیش‌بینی مقادیر خانه‌ها از اندازه، مکان و اینکه آیا حیوانات خانگی مجاز هستند یا خیر داریم. قانون اول را به صورت زیر می‌گیریم:
 If size = big and location = good, then value = high
 با داده‌های باقی مانده، قانون بعدی را می‌گیریم. شاید: If location = good, then value = medium.
 توجه داشته باشید که این قانون روی داده‌ها بدون خانه‌های بزرگ در مکان‌های خوب آموزش داده می‌شود و تنها خانه‌های متوسط و کوچک در مکان‌های خوب باقی می‌مانند.

برای تنظیمات چند کلاسه، رویکرد باید اصلاح شود. اول، طبقات بر اساس بیشترین فراوانی مرتب می‌شوند. الگوریتم پوشش ترتیبی با کلاسی شروع می‌شود که کمترین فراوانی را دارد، یک قانون برای آن می‌آموزد، تمام نمونه‌های تحت پوشش حذف می‌شود، سپس به سراغ دومین کلاس کم فراوان و ... می‌رود. کلاس فعلی همیشه

به عنوان کلاس مثبت در نظر گرفته می‌شود و تمام کلاس‌ها با فراوانی بالاتر در کلاس منفی قرار می‌گیرند. آخرین کلاس، قانون پیش فرض است. در طبقه‌بندی به این استراتژی یک در مقابل همه^۱ نیز گفته می‌شود. چگونه یک قانون واحد را یاد بگیریم؟ الگوریتم OneR در اینجا بی فایده خواهد بود، زیرا همیشه کل فضای ویژگی را پوشش می‌دهد. اما بسیاری از روش‌های دیگر وجود دارد. یک امکان یادگیری یک قانون واحد از درخت تصمیم با جستجوی ستونی^۲ به شرح زیر است:

- یک درخت تصمیم را آموزش دهید (با CART یا هر الگوریتم یادگیری درختی دیگر).
- از گره ریشه شروع کنید و به صورت بازگشتی خالص‌ترین گره را انتخاب کنید (مثالاً با کمترین نرخ طبقه‌بندی اشتباه).
- کلاس اکثریت گره پایانی به عنوان قانون پیش‌بینی استفاده می‌شود. مسیر منتهی به آن گره به عنوان شرط قانون استفاده می‌شود.

شكل زیر جستجوی ستونی در یک درخت را نشان می‌دهد:



شكل ۵.۲۰: یادگیری یک قانون با جستجوی یک مسیر از طریق درخت تصمیم. یک درخت تصمیم برای پیش‌بینی هدف مورد نظر ایجاد می‌گردد. ما از گره ریشه شروع می‌کنیم، حریصانه و مکرر مسیری را دنبال می‌کنیم که به صورت محلی خالص‌ترین زیرمجموعه را تولید می‌کند (به عنوان مثال بالاترین دقت) و تمام مقادیر

¹ one-versus-all

² beam search

If location = good and size = big, then value = high

یادگیری یک قانون یک مساله جستجو است، جایی که فضای جستجو فضای همه قوانین ممکن است. هدف از جستجو یافتن بهترین قانون بر اساس برخی معیارها است. استراتژی‌های جستجوی مختلفی وجود دارند: تپه‌نورده^۱، جستجوی ستونی، جستجوی جامع^۲، جستجوی ابتدا بهترین^۳، جستجوی سفارشی^۴، جستجوی تصادفی^۵، جستجوی بالا به پایین^۶، جستجوی پایین به بالا^۷، ...

RIPPER^۸ ارائه شده توسط Cohen (1995) گونه‌ای از الگوریتم پوشش ترتیبی است. RIPPER کمی پیچیده تر است و از یک مرحله پس پردازش (هرس قانون) برای بهینه سازی لیست (یا مجموعه) تصمیم استفاده می‌کند. RIPPER می‌تواند در حالت مرتب یا نامرتب اجرا شود و یک لیست تصمیم یا مجموعه تصمیم ایجاد کند.

مثال‌ها

ما از RIPPER برای حل مثال‌ها استفاده خواهیم کرد.

الگوریتم RIPPER هیچ قانونی را در طبقه‌بندی سلطان دهانه رحم پیدا نمی‌کند.

هنگامی که از RIPPER در کار رگرسیون برای پیش‌بینی تعداد دوچرخه استفاده می‌کنیم، قوانینی پیدا می‌شوند. از آنجایی که RIPPER فقط برای طبقه‌بندی کار می‌کند، شمارش دوچرخه باید تبدیل به یک خروجی طبقه‌بندی شود. من با چارک بندی شمارش دوچرخه به این امر دست یافتم. به عنوان مثال (۵۹۵۶، ۴۵۴۸) فاصله‌ای است که تعداد دوچرخه‌های پیش‌بینی شده بین ۴۵۴۸ و ۵۹۵۶ را می‌باشد. جدول زیر لیست تصمیم قوانین آموخته شده را نشان می‌دهد.

rules

(temp >= 16) and (days_since_2011 <= 437) and (weathersit = GOOD) and (temp <= 24)	and (days_since_2011 >= 131) => cnt=(4548,5956]
(temp <= 13) and (days_since_2011 <= 111) => cnt= [22,3152]	
(temp <= 4) and (workingday = NO WORKING DAY) => cnt= [22,3152]	
(season = WINTER) and (days_since_2011 <= 368) => cnt= [22,3152]	
(hum >= 72) and (windspeed >= 16) and (days_since_2011 <= 381) and (temp <= 17) =>	cnt= [22,3152]
(temp <= 6) and (weathersit = MISTY) => cnt= [22,3152]	

¹ hill-climbing

² exhaustive search

³ best-first search

⁴ ordered search

⁵ stochastic search

⁶ top-down search

⁷ bottom-up

⁸ Repeated Incremental Pruning to Produce Error Reduction

(hum >= 91) => cnt= [22,3152]

(mnth = NOV) and (days_since_2011 >= 327) => cnt= [22,3152]

(days_since_2011 >= 438) and (weathersit = GOOD) and (hum >= 51) => cnt=(5956,8714]

(days_since_2011 >= 441) and (hum <= 73) and (temp >= 15) => cnt=(5956,8714]

(days_since_2011 >= 441) and (windspeed <= 10) => cnt=(5956,8714]

(days_since_2011 >= 455) and (hum <= 40) => cnt=(5956,8714]

=> cnt=(3152,4548]

تفسیر ساده است: اگر شرایط برقرار باشد، فاصله سمت راست را برای تعداد دوچرخه‌ها پیش‌بینی می‌کنیم. آخرین قانون، قانون پیش‌فرض است که زمانی اعمال می‌شود که هیچ یک از قوانین دیگر در مورد یک نمونه برقرار نباشد. برای پیش‌بینی یک نمونه جدید، از بالای لیست شروع کنید و بررسی کنید که آیا یک قانون اعمال می‌شود یا خیر. هنگامی که یک شرط مطابقت دارد، سمت راست قانون پیش‌بینی این نمونه است. قانون پیش‌فرض تضمین می‌کند که همیشه یک پیش‌بینی وجود دارد.

۵.۵.۳ لیست‌های قانون بیزی^۱

در این بخش، روش دیگری را برای یادگیری یک لیست تصمیم بیان می‌کنم که از این دستور تقریبی پیروی می‌کند:

- الگوهای مکرر پیش استخراج شده می‌توانند به عنوان شرایط قوانین تصمیم استفاده شوند.
- از مجموعه‌ای از قوانین از پیش استخراج شده، یک لیست تصمیم بیاموزید.

یک رویکرد خاص با استفاده از این دستور، لیست قانون بیزی (Letham et al., 2015) یا به اختصار BRL نامیده می‌شود. BRL از آمار بیزی برای یادگیری لیست‌های تصمیم از الگوهای مکرر استفاده می‌کند که از قبل با الگوریتم FP-tree استخراج شده اند (Borgelt, 2005). اما اجازه دهید به آرامی با اولین مرحله BRL شروع کنیم.

پیش استخراج الگوهای مکرر

یک الگوی مکرر، وقوع مکرر (همزمان) مقادیر ویژگی است. در مرحله پیش پردازش برای الگوریتم BRL، از ویژگی‌ها استفاده می‌کنیم (در این مرحله به خروجی هدف نیاز نداریم) و الگوهای متداول را از آنها استخراج می‌کنیم. یک الگو می‌تواند یک مقدار ویژگی منفرد مانند size = medium یا ترکیبی از مقادیر ویژگی مانند size = medium AND location = bad باشد.

فراوانی یک الگو با پشتیبانی آن در مجموعه داده اندازه گیری می‌شود:

¹ Bayesian Rule Lists

$$Support(x_j = A) = \frac{1}{n} \sum_{i=1}^n I(x_j^{(i)} = A)$$

که در آن A مقدار ویژگی، n تعداد نقاط داده در مجموعه داده و I تابع نشانگر^۱ است که ۱ برمی گرداند اگر ویژگی x_j از نمونه A داشته باشد و در غیر این صورت ۰ است. در مجموعه داده ای از مقادیر خانه، اگر ۲۰٪ خانه ها فاقد بالکن و ۸۰٪ دارای یک یا چند بالکن باشند، آنگاه پشتیبانی الگوی $\text{balcony} = 0$ مقدار است. پشتیبانی همچنین می تواند برای ترکیبی از مقادیر ویژگی اندازه گیری شود، به عنوان مثال برای $\text{balcony} = 0 \text{ AND } \text{pets} = \text{allowed}$

الگوریتم های زیادی برای یافتن چنین الگوهای مکرر وجود دارد، به عنوان مثال FP-Growth یا Apriori. این که شما از آن کدام استفاده می کنید اهمیت زیادی ندارد، فقط سرعت یافتن الگوها متفاوت است، اما الگوهای حاصل همیشه یکسان هستند.

من یک ایده تقریبی از نحوه عملکرد الگوریتم Apriori (به معنی "از علت به معلول رسیده") برای یافتن الگوهای مکرر به شما ارائه خواهم داد. در واقع الگوریتم Apriori از دو بخش تشکیل شده است که بخش اول الگوهای مکرر را پیدا می کند و بخش دوم قوانین وابستگی^۲ را از آنها می سازد. برای الگوریتم BRL، ما فقط به الگوهای مکرر علاقه مندیم که در قسمت اول Apriori تولید می شوند.

در مرحله اول، الگوریتم Apriori با تمام مقادیر ویژگی که پشتیبانی بیشتر از حداقل پشتیبانی تعریف شده توسط کاربر دارند، شروع می شود. اگر کاربر بگویید که حداقل پشتیبانی باید ۱۰٪ باشد و فقط ۵٪ از خانه ها $\text{size} = \text{big}$ دارند، ما آن مقدار ویژگی را حذف می کنیم و فقط $\text{size} = \text{medium}$ و $\text{size} = \text{small}$ به عنوان عبارت $\text{size} = \text{big}$ می داریم. این به این معنی نیست که این خانه ها از داده ها حذف می شوند، فقط به این معنی است که $\text{size} = \text{big}$ به عنوان الگوی مکرر برگردانده نمی شوند. بر اساس الگوهای مکرر با یک مقدار ویژگی واحد، الگوریتم Apriori به طور مکرر سعی می کند ترکیبی از مقادیر ویژگی با مرتبه بالاتر را بیابد. الگوها با ترکیب $\text{feature} = \text{value}$ می شوند، به عنوان مثال: $\text{size} = \text{medium} \text{ AND } \text{location} = \text{bad}$. الگوهای تولید شده با پشتیبانی زیر حداقل پشتیبانی حذف می شوند. در پایان ما همه الگوهای مکرر را داریم. هر زیر مجموعه ای از بنده ای الگوی مکرر دوباره مکرر است که به آن خاصیت Apriori می گویند. به طور شهودی منطقی است: با حذف یک شرط از یک الگو، الگوی کاهش یافته می تواند تعداد بیشتری یا حداقل همان تعداد نقاط داده را پوشش دهد، اما نه کمتر. به عنوان مثال، اگر ۲۰٪ از خانه ها هستند $\text{size} = \text{medium}$ and $\text{location} = \text{good}$ که فقط

¹ indicator function

² association rules

medium هستند ۲۰٪ یا بیشتر است. خاصیت Apriori برای کاهش تعداد الگوهای مورد بازرسی استفاده می‌شود. فقط در مورد الگوهای مکرر باید الگوهای مرتبه بالاتر را بررسی کنیم.

اکنون ما با شرایط پیش استخراج برای الگوریتم فهرست قوانین بیزی به آشنا شده ایم. اما قبل از اینکه به مرحله دوم BRL برویم، می‌خواهم به روش دیگری برای یادگیری قوانین بر اساس الگوهای از پیش استخراج شده اشاره کنم. روش‌های دیگر پیشنهاد می‌کنند که نتیجه مورد علاقه را نیز در فرآیند الگوکاوی مکرر و همچنین اجرای بخش دوم الگوریتم Apriori که قوانین IF-THEN را ایجاد می‌کند، وارد کنیم. از آنجایی که الگوریتم نظارت نشده است، قسمت THEN نیز حاوی مقادیر ویژگی است که ما به آنها علاقه‌ای نداریم. اما می‌توانیم قوانینی را که تنها خروجی مورد علاقه را در قسمت THEN دارند، فیلتر کنیم. این قوانین یک مجموعه تصمیم را تشکیل می‌دهند، اما تنظیم، هرس، حذف یا ترکیب مجدد قوانین نیز امکان‌پذیر است. با این حال، در رویکرد BRL، ما با الگوهای مکرر کار می‌کنیم و قسمت THEN و نحوه چیدمان الگوها را در لیست تصمیم با استفاده از آمار بیزی یاد می‌گیریم.

یادگیری لیست‌های قانون بیزی

هدف الگوریتم BRL، یادگیری یک لیست تصمیم دقیق با استفاده از منتخبی از شرایط از پیش استخراج شده می‌باشد، در عین حالی که اولویت با لیست‌هایی است که قوانین کم و شرایط کوتاه دارند. BRL، با تعریف توزیعی از لیست‌های تصمیم با توزیع‌های پیشینی¹ برای طول شرایط (ترجمیحاً قوانین کوتاه‌تر) و تعداد قوانین (ترجمیحاً یک لیست کوتاه‌تر) به این هدف می‌پردازد.

توزیع احتمال پسینی² لیست‌ها این امکان را فراهم می‌کند که گفته شود که یک لیست تصمیم چقدر محتمل است به شرط مفروضات کوتاه بودن و چقدر خوب لیست‌ها با داده‌ها برازش دارند. هدف ما یافتن لیستی است که این احتمال پسین را به حداقل می‌رساند. از آنجایی که یافتن دقیق بهترین لیست مستقیماً از توزیع لیست‌ها امکان‌پذیر نیست، BRL دستورالعمل زیر را پیشنهاد می‌کند:

- ۱- یک لیست تصمیم اولیه ایجاد کنید که به طور تصادفی از توزیع پیشینی گرفته می‌شود.
- ۲- به طور مکرر لیست را با افزودن، جابجایی یا حذف قوانین تغییر دهید، و اطمینان حاصل کنید که لیست‌های حاصل از توزیع پسین لیست‌ها پیروی می‌کنند.

۳- لیست تصمیم را از لیست‌های نمونه برداری شده با بیشترین احتمال با توجه به توزیع پسینی انتخاب کنید. اجازه دهید الگوریتم را دقیق‌تر بررسی کنیم: الگوریتم با الگوهای مقدار ویژگی پیش استخراج با الگوریتم FP-Growth شروع می‌شود. BRL چندین فرض را در مورد هدف و توزیع پارامترهایی که توزیع هدف را تعریف

¹ prior distributions

² posteriori

می‌کند، ایجاد می‌کند. (این آمار بیزی است). اگر با آمار بیزی آشنایی ندارید، زیاد در گیر توضیحات نباشد. دانستن این نکته مهم است که رویکرد بیزی، روشی برای ترکیب دانش یا الزامات موجود (به اصطلاح توزیع‌های پیشینی) در برآش داده‌ها می‌باشد. در مورد لیست‌های تصمیم، رویکرد بیزی منطقی است، زیرا مفروضات قبلی فهرست‌های تصمیم را به کوتاهی (با قوانین کوتاه) ترغیب می‌کند.

هدف، لیست‌های تصمیم نمونه d از توزیع پسینی است:

$$\underbrace{p(d|x.y.A.\alpha.\lambda.\eta)}_{posteriori} \propto \underbrace{p(y|x.d.\alpha)}_{likelihood} \cdot \underbrace{p(d|A.\lambda.\eta)}_{priori}$$

که در آن d یک لیست تصمیم، x ویژگی‌ها، y هدف، A مجموعه شرایط پیش استخراج شده، λ طول مورد انتظار پیشینی از لیست‌های تصمیم، η تعداد مورد پیشینی شرائط در یک قانون، α شبه شمارش پیشینی برای کلاس‌های مثبت و منفی که با بهترین مقدار در $(1, 1)$ ثابت شده است، می‌باشد.

$$p(d|x.y.A.\alpha.\lambda.\eta)$$

رابطه بالا محاسبه می‌کند که به شرط داده‌های مشاهده شده و مفروضات پیشینی، فهرست تصمیم چقدر محتمل است. این مقدار با بخت^۱ متناسب است با خروجی y به شرط لیست تصمیم و داده‌ها ضربدر احتمال لیست به شرط مفروضات پسینی و شرایط پیش استخراج.

$$p(y|x.d.\alpha)$$

رابطه بالا بخت مشاهده y به شرط لیست تصمیم و داده‌ها می‌باشد. BRL فرض می‌کند که y توسط توزیع دیریکله-چند جمله‌ای^۲ تولید می‌شود. هرچه لیست تصمیم d داده‌ها را بهتر توضیح دهد، بخت بالاتری دارد.

$$p(d|A.\lambda.\eta)$$

رابطه بالا توزیع پسینی لیست‌های تصمیم است. این رابطه به طور ضریبی یک توزیع پواسون بریده شده^۳ (پارامتر λ) برای تعداد قوانین موجود در لیست و یک توزیع پواسون بریده شده (پارامتر η) برای تعداد مقادیر ویژگی در شرایط قوانین را ترکیب می‌کند.

یک لیست تصمیم اگر خروجی y را به خوبی توضیح دهد و همچنین بر اساس مفروضات پیشینی محتمل باشد، احتمال پسینی بالایی دارد.

تخمین‌ها در آمار بیزی همیشه کمی گمراه کننده است، زیرا ما معمولاً نمی‌توانیم پاسخ صحیح را مستقیماً محاسبه کنیم، اما باید کاندیدها را ترسیم کنیم، آنها را ارزیابی کنیم و تخمین‌های پسینی خود را با استفاده از روش زنجیره مارکف مونت کارلو^۴ به روز کنیم. برای لیست‌های تصمیم، این کار دشوارتر نیز می‌باشد، زیرا ما باید از توزیع لیست‌های تصمیم استفاده کنیم. نویسنده‌گان BRL پیشنهاد می‌کنند که ابتدا یک لیست تصمیم اولیه

¹ likelihood

² Dirichlet-Multinomial

³ truncated Poisson distribution

⁴ Markov chain Monte Carlo

ترسیم شود و سپس به طور مکرر آن را اصلاح کنند تا نمونه‌هایی از لیست‌های تصمیم از توزیع پسینی لیست‌ها (یک زنجیره مارکوف از لیست‌های تصمیم) تولید شود. نتایج به طور بالقوه به لیست تصمیم اولیه بستگی دارد، بنابراین توصیه می‌شود این روش را تکرار کنید تا از تنوع زیاد لیست‌ها، اطمینان حاصل شود. پیش فرض در پیاده‌سازی نرم افزار ۱۰ بار است. دستور العمل زیر به ما می‌گوید که چگونه یک لیست تصمیم اولیه ترسیم کنیم:

- الگوهای پیش استخراج با FP-Growth.
- پارامتر طول لیست m را از توزیع پواسون بریده شده نمونه بگیرید.
- برای قانون پیش فرض: پارامتر توزیع دیریکله-چندجمله‌ای θ_0 را برای مقدار هدف، نمونه بگیرید.
(یعنی زمانی که هیچ چیز دیگری اعمال نشود، این قانون اعمال شود).
- برای لیست تصمیم $m = 1, \dots, m$ انجام دهید:
 - از پارامتر طول قانون ۱ (تعداد شرایط) برای قانون زنمونه بگیرید.
 - یک شرط طول r_1 از شرایط از پیش استخراج نمونه بگیرید.
 - از پارامتر توزیع دیریکله-چندجمله‌ای برای قسمت THEN نمونه بگیرید (یعنی برای توزیع خروجی هدف به شرط قانون)
- برای هر مشاهده در مجموعه داده:
 - قانونی را از لیست تصمیم بباید که اول اعمال می‌شود (بالا به پایین).
 - خروجی پیش‌بینی‌شده از توزیع احتمال (دوجمله‌ای) که توسط قانون اعمالی، پیشنهاد می‌شود، را ترسیم کنید.

گام بعدی تولید تعداد زیادی از لیست‌های جدید است که از این نمونه اولیه شروع می‌شود تا نمونه‌های زیادی از توزیع پسینی لیست‌های تصمیم گیری به دست آید.

لیست‌های تصمیم گیری جدید با لیست اولیه شروع می‌شوند و سپس به طور تصادفی به یکی از این حالت‌هایی که در ادامه بیان می‌شود، نمونه برداری می‌شوند: انتقال یک قانون به موقعیت دیگری در لیست یا اضافه کردن یک قانون به لیست تصمیم فعلی از شرایط پیش استخراج یا حذف یک قانون از لیست تصمیم. این که کدام یک از قوانین تغییر، اضافه یا حذف می‌شود به طور تصادفی انتخاب می‌شود. در هر مرحله، الگوریتم احتمال پسینی لیست تصمیم (ترکیبی از دقت و کوتاهی) را ارزیابی می‌کند. الگوریتم متروپلیس-هیستینگز^۱ تضمین می‌کند که ما از لیست‌های تصمیمی نمونه برداری می‌کنیم که احتمال بالایی دارند. این روش نمونه‌های

^۱ Metropolis Hastings

زیادی از توزیع لیست‌های تصمیم در اختیار ما قرار می‌دهد. الگوریتم BRL لیست تصمیم از نمونه‌هایی را با بیشترین احتمال پسینی انتخاب می‌کند.

مثال‌ها

تا اینجا توضیح تئوری بود، اکنون بباید روش BRL را در عمل ببینیم. مثال‌ها از نوع سریع‌تری از BRL به نام فهرست‌های قوانین بیزی مقیاس‌پذیر (SBRL¹) ارائه شده توسط Yang et al (۲۰۱۷) استفاده می‌کنند. ما از الگوریتم SBRL برای پیش‌بینی خطر ابتلا به سرطان دهانه رحم استفاده می‌کنیم. ابتدا مجبور شدم تمام ویژگی‌های ورودی را برای کارکرد الگوریتم SBRL گسسته کنم. برای این منظور، من ویژگی‌های پیوسته را بر اساس فراوانی مقادیر بر حسب چند ک قسمت بندی کردم.

ما قوانین زیر را به دست می‌آوریم:

rules

If (STDs=1) (rule [259]) then positive probability = 0.16049383

else if (Hormonal.Contraceptives..years.= [0,10)) (rule [82]) then positive probability = 0.04685408

else (default rule) then positive probability = 0.27777778

توجه داشته باشید که ما قوانین معقولی دریافت می‌کنیم، زیرا پیش‌بینی در قسمت THEN نتیجه کلاس نیست، بلکه احتمال پیش‌بینی شده برای سرطان است.

شرایط از الگوهایی که از قبل با الگوریتم FP-Growth پیش استخراج شده بودند انتخاب شدند. جدول زیر مجموعه شرایطی را نشان می‌دهد که الگوریتم SBRL می‌تواند از بین آنها برای ساخت یک لیست تصمیم انتخاب کند. حداقل تعداد مقادیر ویژگی، که من به عنوان کاربر اجازه دادم دو عدد بود. در اینجا نمونه‌ای از ده الگو آورده شده است:

pre-mined conditions

Num.of.pregnancies= [3.67, 7.33)

IUD=0, STDs=1

Number.of.sexual.partners= [1, 10), STDs..Time.since.last.diagnosis= [1, 8)

First.sexual.intercourse= [10, 17.3), STDs=0

Smokes=1, IUD..years.= [0, 6.33)

Hormonal.Contraceptives..years.= [10, 20), STDs..Number.of.diagnosis= [0, 1)

Age= [13, 36.7)

Hormonal.Contraceptives=1, STDs..Number.of.diagnosis= [0, 1)

Number.of.sexual.partners= [1, 10), STDs..number.= [0, 1.33)

STDs..number.= [1.33, 2.67), STDs..Time.since.first.diagnosis= [1, 8)

¹ Scalable Bayesian Rule Lists

سپس، الگوریتم SBRL را برای پیش‌بینی اجاره دوچرخه به کار بردیم. این الگوریتم تنها در صورتی کار می‌کند که مساله رگرسیون پیش‌بینی تعداد دوچرخه به یک مساله طبقه‌بندی باینری تبدیل شود. من به طور خودسرانه با ایجاد یک برچسب که ۱ است اگر تعداد دوچرخه‌ها از ۴۰۰۰ دوچرخه در روز بیشتر شود و در غیر این صورت ۰، یک مساله طبقه‌بندی ایجاد کرده‌ام.

لیست زیر توسط SBRL آموزش داده شده است:

rules

```
If (yr=2011,temp= [-5.22,7.35)) (rule [718]) then positive probability = 0.01041667
else if (yr=2012,temp= [7.35,19.9)) (rule [823]) then positive probability = 0.88125000
else if (yr=2012,temp= [19.9,32.5]) (rule [816]) then positive probability = 0.99253731
else if (season=SPRING) (rule [351]) then positive probability = 0.06410256
else if (temp= [7.35,19.9)) (rule [489]) then positive probability = 0.44444444
else (default rule) then positive probability = 0.79746835
```

اجازه دهید پیش‌بینی کنیم که تعداد دوچرخه‌ها برای یک روز در سال ۲۰۱۲ با دمای ۱۷ درجه سانتیگراد از ۴۰۰۰ عبور می‌کند. قانون اول برقرار نمی‌شود، زیرا فقط برای روزهای سال ۲۰۱۱ اعمال می‌شود. قانون دوم برقرار می‌شود، زیرا روز در سال ۲۰۱۲ است و ۱۷ درجه در بازه [۷.۳۵, ۱۹.۹] قرار دارد. پیش‌بینی ما برای احتمال اجاره بیش از ۴۰۰۰ دوچرخه ۸۸ درصد است.

۵.۵.۴ مزايا

این بخش به طور کلی مزایای قوانین IF-THEN را مورد بحث قرار می‌دهد. قوانین IF-TEN به راحتی قابل تفسیر هستند. آنها احتمالاً قابل تفسیرترین مدل‌های قابل تفسیر هستند. این موضوع فقط در صورتی برقرار است که تعداد قوانین کم باشد، شرایط قوانین کوتاه (به نظر من حداقل^۳) باشند و اگر قوانین در یک لیست تصمیم یا مجموعه تصمیم غیر متقاطع^۱ سازماندهی شده باشند. قوانین تصمیم گیری می‌توانند به اندازه درخت تصمیم گویا باشند، در حالی که فشرده‌تر هستند. درخت‌های تصمیم اغلب مشکل درخت‌های فرعی تکراری دارند، یعنی زمانی که تقسیم‌ها در گره فرزند چپ و راست دارای ساختار یکسانی است.

پیش‌بینی با قوانین IF-THEN سریع است، زیرا برای تعیین اینکه کدام قوانین برقرار است، فقط باید چند عبارت باینری بررسی شوند.

¹ non-overlapping

قوانين تصمیم گیری در برابر تبدیل یکنواخت ویژگی‌های ورودی مقاوم هستند، زیرا فقط آستانه در شرایط تغییر می‌کند. آنها همچنین در برابر موارد پرت مقاوم هستند، زیرا فقط مهم است که یک شرط اعمال می‌شود یا نه.

قوانين IF-THEN معمولاً مدل‌های محدود ایجاد می‌کنند، به این معنی که ویژگی‌های زیادی به کار گرفته نمی‌شود. این مدل‌ها فقط ویژگی‌های مرتبط^۱ را برای مدل انتخاب می‌کنند. به عنوان مثال، یک مدل خطی به طور پیش فرض وزنی را به تمام ویژگی‌های ورودی اختصاص می‌دهد. ویژگی‌هایی که نامرتبط هستند را می‌توان به سادگی با قوانین IF-THEN نادیده گرفت.

قوانين ساده مانند OneR را می‌توان به عنوان پایه برای الگوریتم‌های پیچیده‌تر استفاده کرد.

۵.۵.۵ معایب

این بخش به طور کلی به معایب قوانین IF-THEN می‌پردازد. تحقیقات و ادبیات قوانین IF-THEN بر طبقه‌بندی تمرکز دارد و تقریباً به طور کامل از رگرسیون غلت می‌کند. درست است که همیشه می‌توانید یک هدف پیوسته را به فواصل تقسیم کنید و مساله رگرسیون را به یک مساله طبقه‌بندی تبدیل کنید، اما همیشه این کار باعث می‌شود اطلاعات را از دست می‌دهید. به طور کلی، رویکردها در صورتی جذاب‌تر هستند که بتوان از آنها برای رگرسیون و طبقه‌بندی استفاده کرد.

اغلب ویژگی‌ها نیز باید طبقه‌بندی شوند. این بدان معناست که اگر می‌خواهید از ویژگی‌های عددی استفاده کنید، باید طبقه‌بندی شوند. راه‌های زیادی برای شکستن یک ویژگی پیوسته به فواصل مختلف وجود دارد. انتخاب روش مناسب بسیار مهم است و با سوالات بسیاری بدون پاسخ روش همراه است. ویژگی باید به چند بازه تقسیم شود؟ معیار تقسیم چیست: طول بازه‌های ثابت، چندک یا چیز دیگر؟ طبقه‌بندی ویژگی‌های پیوسته موضوعی مهم است که اغلب نادیده گرفته می‌شود و افراد فقط از بهترین روش استفاده می‌کنند (روشی که من در حل مثال‌ها استفاده کردم).

بسیاری از الگوریتم‌های قدیمی‌تر یادگیری قوانین مستعد بیش برآذش هستند. الگوریتم‌های ارائه شده در اینجا، همگی حداقل برخی از حفاظت‌ها برای جلوگیری از بیش برآذش را دارند: OneR محدود است، زیرا فقط می‌تواند از یک ویژگی استفاده کند (فقط اگر ویژگی دارای سطوح بیش از حد باشد یا اگر ویژگی‌های زیادی وجود داشته باشد، که معادل با مشکل آزمایش چندگانه^۲ است، RIPPER هرس انجام می‌دهد و لیست های قانون بیزی توزیع پسینی را در لیست‌های تصمیم اعمال می‌کند).

¹ relevant

² multiple testing problem

قوانين تصمیم در توصیف روابط خطی بین ویژگی‌ها و خروجی بد هستند. این مشکلی است که این قوانین مشترکاً با درختان تصمیم دارند. درخت‌ها و قوانین تصمیم فقط می‌توانند توابع پیش‌بینی پله‌ای تولید کنند. به همین علت تغییرات در پیش‌بینی همیشه گستته هستند و هرگز منحنی‌های نرمی ارائه نمی‌کنند. دلیل این امر، مربوط به این موضوع است که ورودی‌ها باید طبقه‌بندی شوند. در درخت‌های تصمیم، با تقسیم آنها طبقه‌بندی می‌شوند.

۵.۵.۶ نرم افزار و جایگزین‌ها

OneR در پکیج OneR^۱ نرم افزار R پیاده‌سازی شده است که برای مثال‌های این کتاب استفاده گردید. همچنین در کتابخانه یادگیری ماشین Weka^۲ پیاده‌سازی شده است و به همین ترتیب در Java و R و RWeka^۳ موجود است. RIPPER در Weka نیز پیاده‌سازی شده است. برای مثال، از پیاده‌سازی JRIP در پکیج SBRL به صورت پکیج^۴ R (که من برای مثال استفاده کردم)، در Python^۵ و C^۶ پیاده‌سازی شده است. علاوه بر این، من پکیج^۷ imodels را توصیه می‌کنم که مدل‌های مبتنی بر قانون مانند لیست‌های قانون بیزی، OneR، CORELS، لیست‌های قوانین حریصانه و موارد دیگر را در یک پکیج Python با یک رابط یادگیری unified scikit-learn پیاده‌سازی می‌کند.

من حتی سعی نکردم همه جایگزین‌ها را برای یادگیری مجموعه‌ها و لیست‌های قوانین تصمیم فهرست کنم، اما به برخی از کارهای به صورت خلاصه اشاره خواهم کرد. من کتاب "مبانی یادگیری قوانین" ارائه شده توسط Fürnkranz et al (۲۰۱۲) را توصیه می‌کنم. این یک کتاب مبسوط در مورد یادگیری قوانین بحث می‌کند و برای کسانی است که می‌خواهند عمیق‌تر به موضوع بپردازنند. در این کتاب، یک چارچوب جامع برای تفکر در مورد قوانین یادگیری ارائه می‌شود و بسیاری از الگوریتم‌های یادگیری قوانین را بیان می‌کند. همچنین توصیه می‌کنم^۸ Weka rule learners را که PART، M5Rules، RIPPER، OneR و بسیاری موارد دیگر را پیاده‌سازی می‌کنند، بررسی کنید. قوانین IF-THEN را می‌توان در مدل‌های خطی همان‌طور که در بخش مربوط به الگوریتم RuleFit این کتاب توضیح داده شده است، استفاده کرد.

^۱ <https://cran.r-project.org/web/packages/OneR/>

^۲ https://www.eecs.yorku.ca/tdb/_doc.php/userg/sw/weka/doc/weka/classifiers/rules/package-summary.html

^۳ <https://cran.r-project.org/web/packages/RWeka/index.html>

^۴ <https://cran.r-project.org/web/packages/sbrl/index.html>

^۵ <https://github.com/datascienceinc/Skater>

^۶ <https://github.com/Hongyuy/sbrlmod>

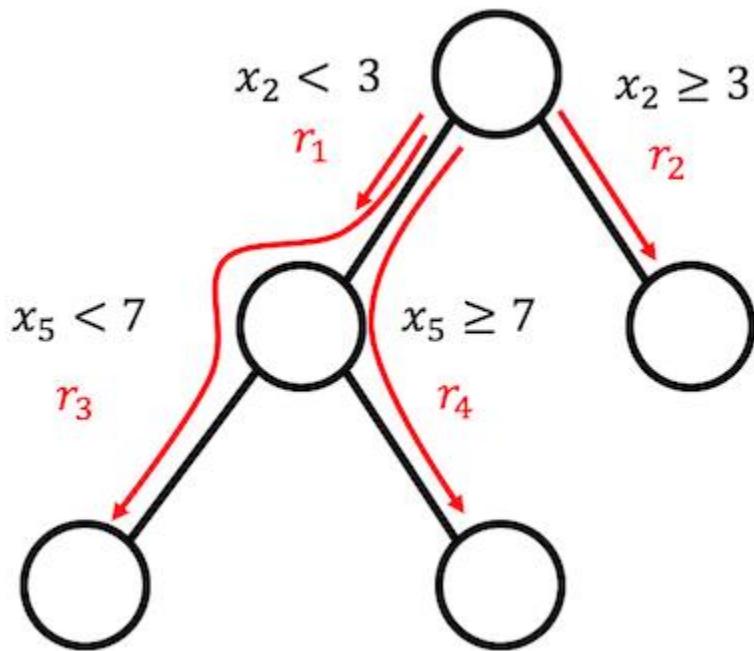
^۷ <https://github.com/csinva/imodels>

^۸ <http://weka.sourceforge.net/doc.dev/weka/classifiers/rules/package-summary.html>

RuleFit ۵.۶

الگوریتم RuleFit ارائه شده توسط Friedman and Popescu (2008) مدل‌های خطی محدودی را یاد می‌گیرد که شامل اثرات تعامل خودکار شناسایی شده، در قالب قوانین تصمیم است.

مدل رگرسیون خطی تعامل بین ویژگی‌ها را در نظر نمی‌گیرد. آیا داشتن مدلی که به اندازه مدل‌های خطی، ساده و قابل تفسیر باشد، اما تعاملات ویژگی‌ها را نیز در نظر بگیرد، راحت است؟ RuleFit این خلاصه را پر می‌کند. RuleFit یک مدل خطی محدود را با استفاده از ویژگی‌های اصلی و همچنین تعدادی ویژگی جدید که قوانین RuleFit تصمیم هستند، یاد می‌گیرد. این ویژگی‌های جدید، تعامل بین ویژگی‌های اصلی را به تصویر می‌کشد. به طور خودکار این ویژگی‌ها را از درخت‌های تصمیم تولید می‌کند. هر مسیر یک درخت می‌تواند با ترکیب تصمیمات تقسیم به یک قانون، به یک قانون تصمیم تبدیل شود. پیش‌بینی‌های گره کنار گذاشته می‌شوند و فقط تقسیم‌ها در قوانین تصمیم‌گیری استفاده می‌شوند:



شکل ۵.۲۱: ۴ قوانین را می‌توان از یک درخت با ۳ گره پایانی تولید کرد.

آن درختان تصمیم از کجا می‌آیند؟ درختان برای پیش‌بینی خروجی مورد علاقه آموزش می‌بینند. این کار تصمیم می‌کند که تقسیم‌ها برای کار پیش‌بینی معنی دار هستند. هر الگوریتمی که تعداد زیادی درخت تولید کند، می‌تواند برای RuleFit استفاده شود، برای مثال یک جنگل تصادفی. هر درخت به قوانین تصمیم تجزیه می‌شود که به عنوان ویژگی‌های اضافی در یک مدل رگرسیون خطی محدود (Lasso) استفاده می‌شود.

مقاله RuleFit از داده‌های مسکن بوستون برای نشان دادن این موضوع استفاده می‌کند: هدف پیش‌بینی میانگین ارزش خانه یک محله بوستون است. یکی از قوانین ایجاد شده توسط RuleFit این است

.IF number of rooms > 6.64 AND concentration of nitric oxide < 0.67 THEN 1 ELSE 0

همچنین یک اندازه گیری اهمیت ویژگی انجام می‌دهد که به شناسایی عبارات خطی و قوانینی که برای پیش‌بینی‌ها مهم هستند، کمک می‌کند. اهمیت ویژگی از وزن مدل رگرسیون محاسبه می‌شود. اندازه اهمیت را می‌توان برای ویژگی‌های اصلی (که هم به شکل خام و هم احتمالاً در بسیاری از قوانین تصمیم استفاده می‌شود) تجمعیع کرد.

همچنین نمودارهای وابستگی جزئی را معرفی می‌کند که این نمودارها میانگین تغییر در پیش‌بینی را در اثر تغییر یک ویژگی نشان می‌دهند. نمودار وابستگی جزئی یک روش آگنوتیک مدل است که می‌تواند با هر مدلی استفاده شود و در بخش نمودارهای وابستگی جزئی توضیح داده می‌شود.

۵.۶.۱ تفسیر و مثال

از آنجایی که RuleFit یک مدل خطی را در پایان تخمین می‌زند، تفسیر مشابه مدل‌های خطی "عادی" است. تنها تفاوت این است که مدل دارای ویژگی‌های جدیدی است که از قوانین تصمیم به دست آمده است. قوانین تصمیم، ویژگی‌های باینری هستند: مقدار ۱ به این معنی است که همه شرایط قانون برآورده می‌شود، در غیر این صورت مقدار ۰ است. برای عبارت‌های خطی در RuleFit، تفسیر مانند مدل‌های رگرسیون خطی است: اگر ویژگی یک واحد افزایش یابد، خروجی پیش‌بینی شده با وزن ویژگی مربوطه تغییر می‌کند.

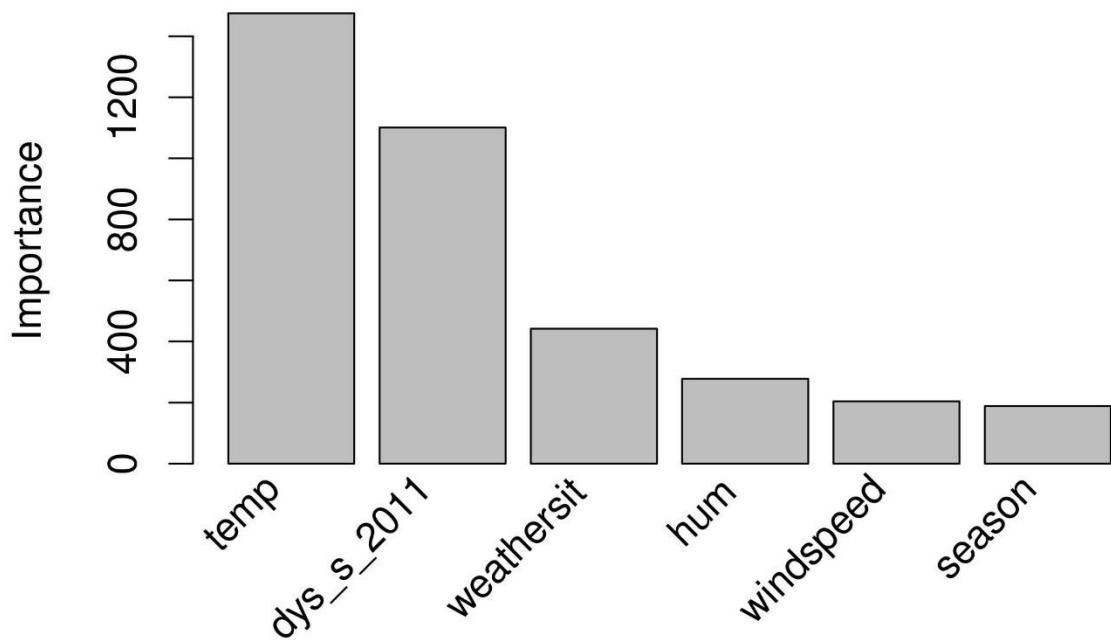
در این مثال، از RuleFit برای پیش‌بینی تعداد دوچرخه‌های اجاره‌ای در یک روز معین استفاده می‌کنیم. جدول پنج مورد از قوانینی را که توسط RuleFit ایجاد شده است به همراه وزن Lasso و اهمیت آن‌ها نشان می‌دهد. نحوه محاسبه در ادامه توضیح داده می‌شود.

Description	Weight	Importance
days_since_2011 > 111 & weathersit in ("GOOD", "MISTY")	795	303
37.25 <= hum <= 90	-20	278
temp > 13 & days_since_2011 > 554	676	239
4 <= windspeed <= 24	-41	204
days_since_2011 > 428 & temp > 5	356	174

مهمترین قانون این است: days_since_2011 > 111 & weathersit in ("GOOD", "MISTY"). وزن مربوطه ۷۹۵ است. تفسیر این است. ۷۹۵ افزایش ۱۱۱ & weathersit in ("GOOD", "MISTY") به میزان ۷۹۵ افزایش

می‌یابد، زمانی که سایر مقادیر ویژگی ثابت باقی بمانند. در مجموع، ۲۷۸ قانون از این ۸ ویژگی اصلی ایجاد شد. خیلی زیاد! اما به لطف Lasso، تنها ۵۹ مورد از ۲۷۸ وزن متفاوت از ۰ دارند. محاسبه اهمیت ویژگی‌های کلی نشان می‌دهد که دما و روند زمانی مهم‌ترین ویژگی‌ها هستند:

Variable importances



شکل ۵.۲۲: اندازه‌های اهمیت ویژگی برای مدل RuleFit که تعداد دوچرخه را پیش‌بینی می‌کند. مهم‌ترین ویژگی برای پیش‌بینی دما و روند زمانی بود.

اندازه‌گیری اهمیت ویژگی شامل اهمیت عبارت ویژگی خام و تمام قوانین تصمیم است که ویژگی در آنها ظاهر می‌شود.

الگوی تفسیر

تفسیر مشابه مدل‌های خطی است: خروجی پیش‌بینی شده به اندازه β_j تغییر می‌کند اگر ویژگی x_j به اندازه یک واحد تغییر کند، مشروط بر اینکه سایر ویژگی‌ها بدون تغییر باقی بمانند. تفسیر وزن یک قانون تصمیم در یک مورد خاص ای گونه است: اگر همه شرایط یک قانون تصمیم r_k اعمال شود، خروجی پیش‌بینی شده به اندازه α_k (وزن یادگیری شده قانون r_k در مدل خطی) تغییر می‌کند.

برای طبقه‌بندی (استفاده از رگرسیون لجستیک به جای رگرسیون خطی): اگر همه شرایط قانون تصمیم r_k برقرار شود، شناس رویداد در برابر عدم رویداد با ضریب α_k تغییر می‌کند.

۵.۶.۲ تئوری

اجازه دهید عمیق‌تر به جزئیات تکنیکی الگوریتم RuleFit از دو بخش تشکیل شده است: بخش اول «قوانين» را از درخت‌های تصمیم تولید می‌کند و بخش دوم یک مدل خطی را با ویژگی‌های اصلی و قوانین جدید به عنوان ورودی برازش می‌دهد (از این رو «RuleFit» نامیده می‌شود).

مرحله ۱: تولید قانون^۱

یک قانون چگونه به نظر می‌رسد؟ قوانین تولید شده توسط الگوریتم شکل ساده‌ای دارند. به عنوان مثال $IF x_2 < 3 \text{ AND } x_5 < 7 \text{ THEN } 1 \text{ ELSE } 0$. قوانین با تجزیه درختان تصمیم ساخته می‌شوند: هر مسیری منتهی به یک گره در یک درخت، می‌تواند به یک قانون تصمیم تبدیل شود. درختان مورد استفاده برای قوانین برای پیش‌بینی خروجی هدف برازش داده شده‌اند. بنابراین تقسیم‌ها و قوانین حاصل برای پیش‌بینی نتیجه مورد علاقه شما بهینه شده‌اند. مطلوب است که بسیاری از قوانین متنوع و معنادار ایجاد شود. از تقویت گرادیان^۲ برای برازش ترکیبی از درختان تصمیم برای رگرسیون یا طبقه‌بندی y با ویژگی‌های اصلی X استفاده می‌شود. هر درخت به دست آمده به چندین قانون تبدیل می‌شود. نه تنها درختان تقویت شده، بلکه از هر الگوریتم ترکیبی درختی می‌توان برای تولید درختان برای RuleFit استفاده کرد. یک ترکیب درختی را می‌توان با این فرمول کلی توصیف کرد:

$$\hat{f}(x) = a_0 + \sum_{m=1}^M a_m \hat{f}_m(X)$$

M تعداد درختان و $\hat{f}_m(X)$ تابع پیش‌بینی درخت m -ام است. a_m وزن‌ها هستند. Bagged ensembles، جنگل تصادفی، AdaBoost و MART ترکیبات درختی تولید می‌کنند و می‌توانند برای RuleFit استفاده شوند.

ما قوانین را از تمام درختان گروه ایجاد می‌کنیم. هر قانون r_m به شکل زیر بیان می‌شود:

$$r_m(x) = \prod_{j \in T_m} I(x_j \in s_{jm})$$

در این رابطه T_m مجموعه‌ای از ویژگی‌های مورد استفاده در درخت m -ام است، I تابع نشانگر است که ۱ است هنگامی که ویژگی x_j در زیر مجموعه مشخص شده از مقادیر s برای ویژگی j -ام باشد (همان‌طور که توسط تقسیم درخت مشخص شده است) و در غیر این صورت ۰. برای ویژگی‌های عددی، s_{jm} بازه‌ای در محدوده مقدار ویژگی است. بازه یکی از این دو حالت زیر است:

¹ Rule generation

² Gradient boosting

$$x_{s_{jm}.lower} < x_j \\ x_j < x_{s_{jm}.upper}$$

تقسیمات بیشتر در آن ویژگی احتمالاً به بازه‌های پیچیده‌تر منجر می‌شود. برای ویژگی‌های طبقه‌بندی، زیرمجموعه s شامل طبقه‌های خاصی از ویژگی است. یک مثال ساخته شده برای مجموعه‌داده اجاره دوچرخه:

$$r_{17}(x) = I(x_{temp} < 15) \cdot I(x_{weather} \in \{good, cloudy\}) \cdot I(10 \leq x_{windspeed} < 20)$$

اگر هر سه شرط برآورده شوند، این قانون ۱ را برمی‌گرداند، در غیر این صورت .. RuleFit همه قوانین ممکن از یک درخت را استخراج می‌کند، نه فقط از گره‌های برگ. بنابراین قانون دیگری که ایجاد می‌شود این است:

$$r_{18}(x) = I(x_{temp} < 15) \cdot I(x_{weather} \in \{good, cloudy\})$$

در مجموع، تعداد قوانین ایجاد شده از مجموعه‌ای با M درخت و t_m گره پایانی عبارت‌اند از:

$$K = \sum_{m=1}^M 2(t_m - 1)$$

ترفندی که توسط نویسندهای RuleFit معرفی شده است، یادگیری درختان با عمق تصادفی است در نتیجه تعداد زیادی از قوانین متنوع با طول‌های مختلف تولید می‌شوند. توجه داشته باشید که مقدار پیش‌بینی شده را در هر گره کنار می‌گذاریم و فقط شرایطی را حفظ می‌کنیم که ما را به یک گره هدایت می‌کند و سپس از آن یک قانون ایجاد می‌کنیم. وزن دهی قوانین تصمیم‌گیری در مرحله ۲، RuleFit انجام می‌شود.

روش دیگری برای انجام مرحله ۱: RuleFit مجموعه جدیدی از ویژگی‌ها را با استفاده از ویژگی‌های اصلی ایجاد می‌کند. این ویژگی‌ها باینری هستند و می‌توانند نشان دهنده تعاملات کاملاً پیچیده، ویژگی‌های اصلی باشند. قوانین برای به حداقل رساندن وظیفه پیش‌بینی انتخاب می‌شوند. قوانین به طور خودکار از ماتریس کوواریانس X تولید می‌شوند. شما به سادگی می‌توانید قوانین را به عنوان ویژگی‌های جدید بر اساس ویژگی‌های اصلی خود ببینید.

مرحله ۲: مدل خطی محدود

شما قوانین بسیاری را در مرحله ۱ به دست خواهید آورد. از آنجایی که مرحله اول را می‌توان تنها به عنوان یک تبدیل ویژگی در نظر گرفت، هنوز کار برآش یک مدل تمام نشده است. همچنین، می‌خواهید تعداد قوانین را کاهش دهید. علاوه بر قوانین، تمام ویژگی‌های «خام» مجموعه‌داده اصلی، نیز در مدل خطی پراکنده استفاده خواهند شد. هر قانون و هر ویژگی اصلی به یک ویژگی در مدل خطی تبدیل می‌شود و یک وزن تخمینی می‌گیرد. ویژگی‌های خام اولیه اضافه می‌شوند زیرا درخت‌ها در نمایش روابط خطی ساده بین y و x شکست

می خورند. قبل از این که یک مدل خطی محدود را آموزش دهیم، ویژگی های اصلی را حاشیه کوب می کنیم^۱ تا در برابر داده های پرت مقاوم تر باشند:

$$l_j^*(x_j) = \min(\delta_j^+ \cdot \max(\delta_j^- \cdot x_j))$$

در این رابطه، δ_j^- و δ_j^+ چندک های δ تایی از توزیع داده در ویژگی x_j هستند. انتخاب 0.5% برای δ به این معنی است که هر مقدار از ویژگی x_j که در 0.5% کمترین یا 0.5% بالاترین مقدارین باشد، به ترتیب به مقدارهای 0.5% یا 0.95% تنظیم می شود. به عنوان یک قانون کلی، شما می توانید $\delta = 0.025$ را انتخاب کنید. علاوه بر این، عبارات خطی باید به گونه ای نرمال شوند که اهمیت پسینی به اندازه یک قانون تصمیم گیری معمولی داشته باشند:

$$l_j(x_j) = 0.4 \cdot l_j^*(x_j) / \text{std}(l_j^*(x_j))$$

مقدار 0.4 میانگین انحراف استاندارد قوانین با توزیع پشتیبانی یکنواخت $sk \sim U(0.1)$ است.

ما هر دو نوع ویژگی را برای ایجاد یک ماتریس ویژگی جدید ترکیب می کنیم و یک مدل خطی محدود را با استفاده از Lasso با ساختار زیر آموزش می دهیم:

$$\hat{f}(x) = \hat{\beta}_0 + \sum_{k=1}^K \hat{\alpha}_k r_k(x) + \sum_{j=1}^p \hat{\beta}_j l_j(x_j)$$

در این رابطه $\hat{\alpha}$ بردار وزن تخمینی برای قوانین ویژگی ها و $\hat{\beta}$ بردار وزن برای ویژگی های اصلی می باشد. از آنجایی که RuleFit از Lasso استفاده می کند،تابع زیان محدودیت اضافی دارد تا تخمین برخی از وزن ها را صفر کند:

$$(\{\hat{\alpha}\}_1^K \cdot \{\hat{\beta}\}_0^p) = \underset{\{\hat{\alpha}\}_1^K \cdot \{\hat{\beta}\}_0^p}{\text{argmin}} \sum_{i=1}^n L(y^{(i)} \cdot f(x^{(i)})) + \lambda \cdot \left(\sum_{k=1}^K |\alpha_k| + \sum_{j=1}^p |b_k| \right)$$

نتیجه یک مدل خطی است که اثراتی خطی برای همه ویژگی های اصلی و قوانین دارد. تفسیر مشابه مدل های خطی است، تنها تفاوت این است که برخی از ویژگی ها اکنون قوانین باینری هستند.

مرحله ۳ (اختیاری): اهمیت ویژگی

برای عبارات خطی ویژگی های اصلی، اهمیت ویژگی با پیش بینی استاندارد شده، اندازه گیری می شود:

$$I_j = |\hat{\beta}_j| \cdot \text{std}(l_j(x_j))$$

در این رابطه، β وزن مدل Lasso و $std(l_j(x_j))$ انحراف استاندارد عبارت خطی بر روی داده است.

برای عبارات قانون تصمیم، اهمیت با فرمول زیر محاسبه می شود:

$$I_k = |\hat{\alpha}_k| \cdot \sqrt{s_k(1-s_k)}$$

¹ winsorize

در این رابطه $\hat{\alpha}_k$ وزن Lasso مربوطه از قانون تصمیم است و s_k پشتیبان ویژگی در داده است، که پشتیبان درصدی از نقاط داده است که قانون تصمیم در مورد آنها برقرار است (جایی که $1 = r_k(x)$).

$$s_k = \frac{1}{n} \sum_{i=1}^n r_k(x^{(i)})$$

یک ویژگی در یک عبارت خطی وجود دارد و احتمالاً در بسیاری از قوانین تصمیم گیری نیز رخ می‌دهد. چگونه اهمیت کلی یک ویژگی را اندازه گیری کنیم؟ اهمیت $(x)_j$ یک ویژگی را می‌توان برای هر پیش‌بینی منفرد، اندازه گیری کرد:

$$J_j(x) = I_j(x) + \sum_{x_j \in r_k} I_k(x) / m_k$$

در این رابطه I_l اهمیت عبارت خطی و I_k اهمیت قوانین تصمیمی است که x_j در آن‌ها ظاهر می‌شود، و تعداد ویژگی‌های تشکیل دهنده قانون r_k است. تجمعی اهمیت ویژگی همه نمونه‌ها، اهمیت ویژگی کلی را می‌دهد:

$$J_j(X) = \sum_{i=1}^n J_j(x^{(i)})$$

ممکن است زیر مجموعه‌ای از نمونه‌ها انتخاب شود و محاسبه اهمیت ویژگی برای این گروه انجام شود.

۵.۶.۳ مزایا

RuleFit به طور خودکار تعاملات ویژگی را به مدل‌های خطی اضافه می‌کند. بنابراین، مشکل مدل‌های خطی را حل می‌کند که باید عبارات تعامل را به صورت دستی اضافه کنید و کمی به مسئله مدل سازی روابط غیرخطی کمک می‌کند.

RuleFit می‌تواند هم وظایف طبقه‌بندی و هم رگرسیون را انجام دهد. قوانین ایجاد شده به راحتی قابل تفسیر هستند، زیرا آن‌ها قوانین تصمیم بازی‌نری هستند. یا این قانون در مورد یک نمونه اعمال می‌شود یا خیر. تفسیر پذیری خوب تنها زمانی تضمین می‌شود که تعداد شرایط درون یک قانون خیلی زیاد نباشد. قانون با ۱ تا ۳ شرط به نظر من معقول است. این به معنای حداقل عمق ۳ برای درختان در ترکیب درختی است.

حتی اگر قوانین زیادی در مدل وجود داشته باشد، آن‌ها برای هر نمونه اعمال نمی‌شوند. برای یک نمونه خاص فقط تعداد انگشت شماری از قوانین اعمال می‌شود (= وزن غیر صفر دارند). این قابلیت تفسیر محلی را بهبود می‌بخشد.

مجموعه‌ای از ابزارهای تشخیصی مفید را پیشنهاد می‌کند. این ابزارها آگنوستیک مدل هستند، که می‌توانید آنها را در بخش آگنوستیک مدل کتاب بیایید: اهمیت ویژگی، نمودارهای وابستگی جزئی و تعاملات ویژگی.

۵.۶.۴ معایب

گاهی اوقات RuleFit قوانین زیادی ایجاد می‌کند که وزن غیر صفر در مدل Lasso دریافت می‌کند. تفسیرپذیری با افزایش تعداد ویژگی‌ها در مدل کاهش می‌یابد. یک راه حل امیدوارکننده این است که تاثیرات ویژگی را مجبور کنیم که یکنواخت باشند، به این معنی که افزایش یک ویژگی باید منجر به افزایش پیش‌بینی شود. یک تجربه شخصی: مقالات مدعی عملکرد خوب RuleFit هستند - اغلب نزدیک به عملکرد پیش‌بینی جنگل‌های تصادفی! اما در موارد محدودی که من شخصاً آن را امتحان کردم، عملکرد نامیدکننده بود. فقط آن را برای مساله خود امتحان کنید و ببینید چگونه کار می‌کند.

محصول نهایی روبه RuleFit یک مدل خطی با ویژگی‌های فانتزی اضافی (قوانین تصمیم گیری) است. اما از آنجایی که این یک مدل خطی است، تفسیر وزن هنوز غیرشهودی است. همچنان با همان «پاورقی» مدل رگرسیون خطی معمولی ارائه می‌شود: «... به شرطی که همه ویژگی‌ها ثابت هستند». وقتی قوانینی با هم تداخل دارند کمی دشوارتر می‌شود. به عنوان مثال، یک قانون تصمیم (ویژگی) برای پیش‌بینی دوچرخه می‌تواند این باشد: $temp > 10$ و قانون دیگری می‌تواند $weather = 'GOOD'$ باشد. اگر هوا خوب باشد و دما بالای ۱۵ درجه باشد، دما به طور خودکار از ۱۰ بیشتر می‌شود. در مواردی که قانون دوم اعمال می‌شود، قانون اول نیز اعمال می‌شود. تفسیر وزن تخمینی برای قانون دوم این است: با فرض ثابت ماندن تمام ویژگی‌های دیگر، تعداد پیش‌بینی شده دوچرخه‌ها β_2 افزایش می‌یابد، زمانی که هوا خوب و دمای بالای ۱۵ درجه باشد. اما، اکنون واقعاً روش می‌شود که «همه ویژگی‌های دیگر ثابت شده‌اند» مشکل‌ساز است، زیرا اگر قانون ۲ اعمال شود، قانون ۱ نیز اعمال می‌شود و تفسیر بی‌معنی است.

۵.۶.۵ نرم افزار و جایگزین

الگوریتم RuleFit در R توسط Fokkema (2017) پیاده‌سازی شده است و می‌توانید نسخه Python را در GitHub¹ پیدا کنید.

یک چارچوب بسیار مشابه² skope-rules است، که یک ماژول Python که قوانین را از مجموعه‌ها استخراج می‌کند. این ماژول در نحوه یادگیری قوانین نهایی، متفاوت است: اول، skope-rules قوانین با عملکرد پایین را بر اساس فراخوانی و آستانه‌های دقیق حذف می‌کنند. سپس قوانین تکراری و مشابه با انجام یک انتخاب بر

¹ <https://github.com/christophM/rulefit>

² <https://github.com/scikit-learn-contrib/skope-rules>

اساس تنوع اصطلاحات منطقی (متغیر + عملگر بزرگتر/کوچکتر) و عملکرد (F1-score) حذف می‌شوند. این مرحله نهایی به استفاده از Lasso متکی نیست، بلکه فقط به out-of-bag F1-score و عبارات منطقی را که قوانین را تشکیل می‌دهند، بستگی دارد.

پکیج¹ imodels همچنین شامل اجرای مجموعه قوانین دیگر، مانند مجموعه قوانین بیزی، مجموعه قوانین تقویت شده، و مجموعه قوانین SLIPPER به عنوان یک پکیج scikit-learn با یک رابط Python یکپارچه است.

¹ <https://github.com/csinva/imodels>

۵.۷ سایر مدل‌های قابل تفسیر

فهرست مدل‌های قابل تفسیر دائماً در حال افزایش است و تعداد آن‌ها نامشخص است. این فهرست شامل مدل‌های ساده‌ای مانند مدل‌های خطی، درخت‌های تصمیم و بیز ساده است. علاوه بر این فهرست، مدل‌های پیچیده‌تری وجود دارند که مدل‌های یادگیری ماشین غیرقابل تفسیر را ترکیب یا اصلاح می‌کنند تا آن‌ها را قابل تفسیرتر کنند. به‌ویژه مستندات علمی، مربوط به مدل‌های نوع دوم در حال حاضر با نرخ بالا تولید می‌شوند و به سختی می‌توان تمام پیشرفت‌ها را رصد کرد. این کتاب فقط طبقه‌بند بیز ساده و k نزدیک‌ترین همسایه را در این فصل پوشش می‌دهد.

۵.۷.۱ طبقه‌بند بیز ساده^۱

طبقه‌بند بیز ساده از قضیه احتمالات شرطی بیز^۲ استفاده می‌کند. برای هر ویژگی، احتمال یک کلاس را بسته به مقدار ویژگی محاسبه می‌کند. طبقه‌بند بیز ساده، احتمالات کلاس را برای هر ویژگی به طور مستقل محاسبه می‌کند. این مقدار با یک فرض مهم (= ساده لوحانه) استقلال شرطی ویژگی‌ها است. بیز ساده یک مدل احتمال شرطی است و احتمال کلاس C_k را به شرح زیر مدل می‌کند:

$$P(C_k|x) = \frac{1}{Z} P(C_k) \prod_{i=1}^n P(x_i|C_k)$$

عبارت Z یک پارامتر مقیاس‌بندی است که تضمین می‌کند که مجموع احتمالات برای همه کلاس‌ها ۱ است (در غیر این صورت خروجی‌ها، احتمالات نیستند). احتمال شرطی یک کلاس، احتمال کلاس ضربدر احتمال هر ویژگی به شرط کلاس است، که با Z نرمال شده است. این فرمول را می‌توان با استفاده از قضیه بیز به دست آورد.

بیز ساده به دلیل فرض استقلال، یک مدل قابل تفسیر است. می‌توان آن را در سطح مدولار تفسیر کرد. برای هر ویژگی بسیار واضح است که چقدر به پیش‌بینی کلاس خاصی کمک می‌کند، زیرا می‌توانیم احتمال شرطی را تفسیر کنیم.

۵.۷.۲ k نزدیک‌ترین همسایه‌ها

روش k نزدیک‌ترین همسایه را می‌توان برای رگرسیون و طبقه‌بندی استفاده کرد و از نزدیک‌ترین همسایگان یک نقطه داده برای پیش‌بینی استفاده می‌کند. برای طبقه‌بندی، روش k نزدیک‌ترین همسایه، رایج‌ترین کلاس نزدیک‌ترین همسایه‌های یک نمونه را اختصاص می‌دهد. برای رگرسیون، میانگین نتیجه همسایگان را بر می

¹ Naive Bayes Classifier

² Bayes' theorem of conditional probabilities

گرداند. بخش‌های دشوار یافتن تعداد k مناسب و تصمیم درباره نحوه اندازه‌گیری فاصله بین نمونه‌ها است که در نهایت همسایگی را مشخص می‌کند.

مدل k نزدیک‌ترین همسایه با سایر مدل‌های قابل تفسیر ارائه شده در این کتاب متفاوت است زیرا یک الگوریتم یادگیری مبتنی بر نمونه است. چگونه می‌توان k نزدیک‌ترین همسایه‌ها را تفسیر کرد؟ اول از همه، هیچ پارامتری برای یادگیری وجود ندارد، بنابراین هیچ تفسیرپذیری در سطح مدولار وجود ندارد. علاوه بر این، مشکل عدم تفسیرپذیری مدل کلی نیز وجود دارد، زیرا مدل ذاتاً محلی است و هیچ وزن یا ساختار کلی ای ندارد که به صراحت آموخته شده باشد. آیا ممکن است در سطح محلی قابل تفسیر باشد؟ برای توضیح یک پیش‌بینی، همیشه می‌توانید k همسایه را که برای پیش‌بینی استفاده شده‌اند، بازیابی کنید. اینکه آیا مدل قابل تفسیر است یا نه، تنها به این سؤال بستگی دارد که آیا می‌توانید یک نمونه واحد را در مجموعه‌داده «تفسیر» کنید. اگر یک نمونه از صدھا یا هزاران ویژگی تشکیل شده باشد، من استدلال می‌کنم که قابل تفسیر نیست. اما اگر تعداد ویژگی‌های کمی دارید یا راهی برای کاهش تعداد ویژگی‌های نمونه خود به مهم‌ترین ویژگی‌ها دارید، ارائه k نزدیک‌ترین همسایه می‌تواند توضیحات خوبی به شما بدهد.

فصل ۶ روش‌های آگنوتیک مدل

جداسازی توضیحات از مدل یادگیری ماشین (= روش‌های تفسیر آگنوتیک مدل مدل) مزایایی دارد (Ribeiro et al., 2016a). مزیت بزرگ روش‌های تفسیر آگنوتیک مدل نسبت به روش‌های خاص مدل، انعطاف پذیری آنهاست. اگر بتوان روش‌های تفسیر را برای هر مدلی اعمال کرد، آن گاه توسعه‌دهندگان یادگیری ماشین آزادند که از هر مدل یادگیری ماشین که دوست دارند استفاده کنند. هر چیزی که مبتنی بر تفسیر یک مدل یادگیری ماشین باشد، مانند گرافیک یا رابط کاربری، از مدل یادگیری ماشین اصلی نیز مستقل می‌شود. به طور معمول، نه تنها یک، بلکه بسیاری از انواع مدل‌های یادگیری ماشین برای انجام یک وظیفه ارزیابی می‌شوند و هنگام مقایسه مدل‌ها از نظر تفسیرپذیری، کار با توضیحات آگنوتیک مدل آسان‌تر است، زیرا همان توضیح را می‌توان برای هر مدل دیگر نیز استفاده کرد.

یک جایگزین برای روش‌های تفسیر آگنوتیک مدل، استفاده از مدل‌های قابل تفسیر است، که اغلب دارای این عیب بزرگ است که عملکرد پیش‌بینی در مقایسه با سایر مدل‌های یادگیری ماشین کاهش می‌یابد و شما مجبور به استفاده از مدل‌های محدودی هستید. جایگزین دیگر استفاده از روش‌های تفسیر خاص مدل است. عیب این کار این است که شما را به یک نوع مدل محدود می‌کند و تغییر از این مدل به مدل دیگر دشوار خواهد بود.

جنبه‌های مطلوب یک سیستم توضیح آگنوتیک مدل عبارت‌اند از (Ribeiro et al., 2016a):

- **انعطاف‌پذیری مدل^۱:** روش تفسیر می‌تواند با هر مدل یادگیری ماشین مانند جنگل‌های تصادفی و شبکه‌های عصبی عمیق کار کند.
- **انعطاف‌پذیری توضیح^۲:** شما محدود به شکل خاصی از توضیح نیستید. در برخی موارد ممکن است داشتن یک فرمول خطی مفید باشد، در موارد دیگر یک نمودار اهمیت ویژگی.
- **انعطاف‌پذیری ارائه^۳:** سیستم توضیح باید قادر به استفاده از ارائه ویژگی مختلف باشد تا مدل توضیح داده شود.

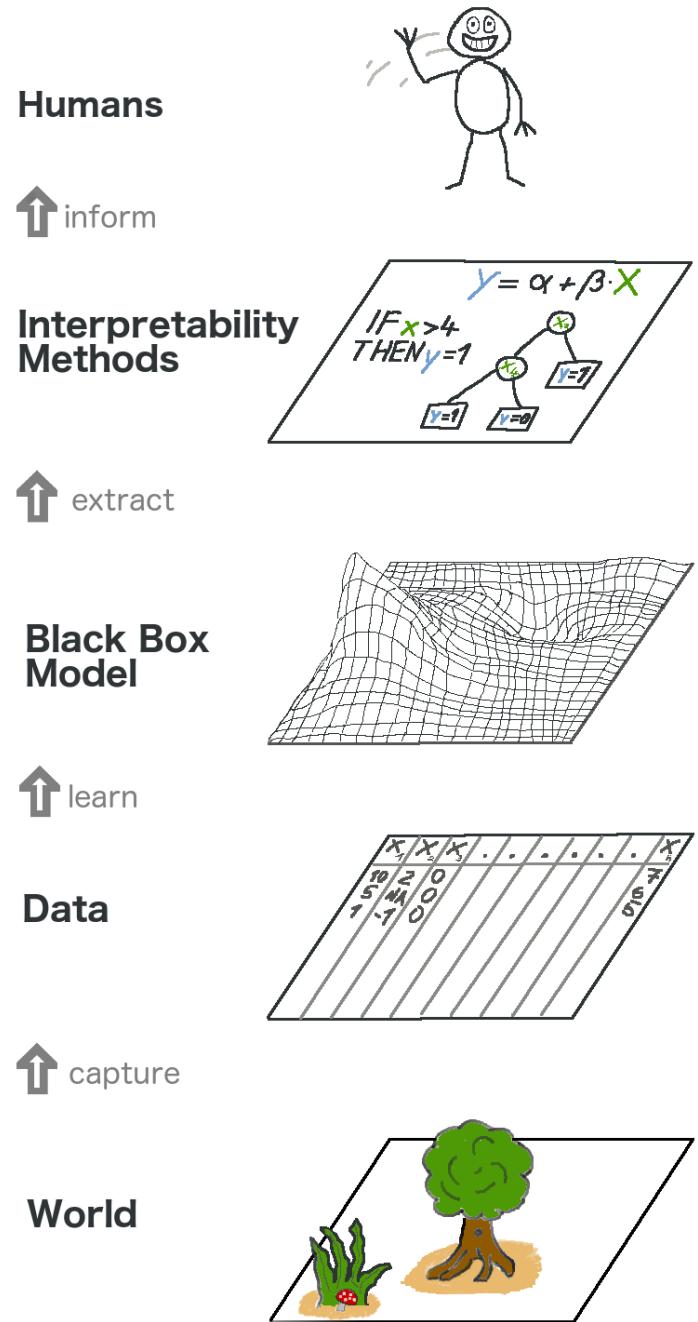
تصویر بزرگ‌تر

اجازه دهید نگاهی در سطح بالا به تفسیرپذیری آگنوتیک مدل بیندازیم. ما با جمع‌آوری داده‌ها، جهان را به تصویر می‌کشیم و با آموزش مدل یادگیری ماشین با استفاده از این داده‌ها برای پیش‌بینی داده‌ها (برای وظیفه) آن را انتزاع می‌کنیم. تفسیرپذیری لایه مجزای دیگری در بالای صفحه است که به فهم انسان کمک می‌کند.

¹ Model flexibility

² Explanation flexibility

³ Representation flexibility



شکل ۱.۶: تصویر بزرگ یادگیری ماشین قابل توضیح. دنیای واقعی قبل از اینکه در قالب توضیحات به دست انسان برسد از لایه‌های زیادی عبور می‌کند. پایین‌ترین لایه جهان است. جهان می‌تواند به معنای واقعی کلمه خود طبیعت باشد، مانند بیولوژی بدن انسان و نحوه واکنش آن به دارو، اما ممکن است چیزهای انتزاعی‌تر مانند بازار املاک و مستغلات نیز باشد. لایه جهان

شامل هر چیزی است که می‌توان مشاهده کرد و مورد توجه است. در نهایت، ما می‌خواهیم چیزی در مورد جهان بیاموزیم و با آن تعامل داشته باشیم.

لایه دوم لایه داده است. ما باید جهان را دیجیتالی کنیم تا بتوانیم آن را برای کامپیوترها پردازش و همچنین اطلاعات را ذخیره نماییم. لایه داده شامل هر چیزی مانند تصاویر، متون، داده‌های جدولی و غیره است.

با برازش مدل‌های یادگیری ماشین بر اساس لایه داده، لایه مدل جعبه سیاه را به دست می‌آوریم. الگوریتم‌های یادگیری ماشین با داده‌های دنیای واقعی آموزش می‌بینند تا پیش‌بینی کنند یا ساختارها را بیابند. در بالای لایه مدل جعبه سیاه لایه **روش‌های تفسیرپذیری** قرار دارد که به ما کمک می‌کند تا با عدم شفافیت مدل‌های یادگیری ماشین مقابله کنیم. مهم‌ترین ویژگی برای یک تشخیص خاص چه بود؟ چرا یک تراکنش مالی به عنوان تقلب طبقه‌بندی شد؟

آخرین لایه توسط یک انسان اشغال شده است. نگاه کن! این قسمت برای شماست زیرا در حال خواندن این کتاب هستید و به ارائه توضیحات بهتر برای مدل‌های جعبه سیاه کمک می‌کنید! انسان‌ها در نهایت مصرف کننده توضیحات هستند.

این انتزاع چند لایه همچنین به درک تفاوت‌های رویکردهای بین آماردانان و متخصصان یادگیری ماشین کمک می‌کند. آماردانان با لایه داده سروکار دارند، مانند برنامه ریزی آزمایشات بالینی یا طراحی نظرسنجی. آنها لایه مدل جعبه سیاه را رد می‌کنند و مستقیماً به لایه روش‌های تفسیرپذیری می‌روند. متخصصان یادگیری ماشین همچنین با لایه داده سروکار دارند، مانند جمع‌آوری نمونه‌های برچسب گذاری شده از تصاویر سلطان پوست یا crawling Wikipedia. سپس آنها یک مدل یادگیری ماشین جعبه سیاه را آموزش می‌دهند. لایه روش‌های تفسیرپذیری نادیده گرفته می‌شود و انسان‌ها مستقیماً با پیش‌بینی‌های مدل جعبه سیاه سروکار دارند. بسیار خوب است که یادگیری ماشین قابل تفسیر، کار آماردانان و متخصصان یادگیری ماشین را ترکیب می‌کند.

البته این گرافیک همه چیز را به تصویر نمی‌کشد: داده‌ها می‌توانند از شبیه‌سازی به دست آیند. مدل‌های جعبه سیاه گاهی پیش‌بینی‌هایی را ارائه می‌دهند که حتی ممکن است به دست انسان هم نرسد، بلکه فقط برای ماشین‌های دیگر و غیره را ایجاد شده‌اند. اما به طور کلی، درک اینکه چگونه تفسیرپذیری به این لایه جدید در بالای مدل‌های یادگیری ماشین تبدیل می‌شود، یک انتزاع مفید است.

روش‌های تفسیر آگنوستیک مدل را می‌توان به روش‌های محلی و کلی تقسیم بندی کرد. کتاب نیز بر اساس این تقسیت‌تنظيم شده است. روش‌های کلی چگونگی تأثیر ویژگی‌ها بر پیش‌بینی را به طور متوسط توصیف می‌کنند. در مقابل، هدف روش‌های محلی، توضیح پیش‌بینی‌های خاص است.

فصل ۷ توضیحات مبتنی بر مثال

روش‌های توضیح مبتنی بر مثال، نمونه‌های خاصی از مجموعه‌داده را برای توضیح رفتار مدل‌های یادگیری ماشین یا توضیح توزیع داده‌های اساسی انتخاب می‌کنند.

توضیح‌های مبتنی بر مثال عمدتاً آگنوستیک مدل هستند، زیرا هر مدل یادگیری ماشینی را قابل تفسیرتر می‌کنند. تفاوت این روش‌ها با روش‌های آگنوستیک مدل در این است که روش‌های مبتنی بر مثال، یک مدل را با انتخاب نمونه‌هایی از مجموعه‌داده توضیح می‌دهند و نه با ایجاد خلاصه‌ای از ویژگی‌ها (مانند اهمیت ویژگی یا وابستگی جزئی). توضیحات مبتنی بر مثال تنها زمانی معنا پیدا می‌کنند که بتوانیم نمونه‌ای از داده‌ها را به روشی قابل فهم برای انسان نمایش دهیم. این روش به خوبی برای تصاویر کار می‌کند، زیرا می‌توانیم آنها را مستقیماً مشاهده کنیم. به طور کلی، روش‌های مبتنی بر مثال اگر مقادیر ویژگی‌های یک نمونه دارای محتوای بیشتری باشند، به خوبی کار می‌کنند، به این معنی که داده‌ها دارای ساختاری هستند، مانند تصاویر یا متون. نمایش داده‌های جدولی به روشنی معنادار چالش برانگیزتر است، زیرا یک نمونه می‌تواند از صدھا یا هزاران ویژگی (کمتر ساختارمند) تشکیل شده باشد. فهرست کردن تمام مقادیر ویژگی برای توصیف یک نمونه معمولاً مفید نیست. این روش اگر فقط تعداد انگشت شماری از ویژگی‌ها وجود داشته باشد یا راهی برای خلاصه کردن یک نمونه داشته باشیم، به خوبی کار می‌کند.

توضیحات مبتنی بر مثال به انسان کمک می‌کند تا مدل‌های ذهنی از مدل یادگیری ماشین و داده‌هایی را که مدل یادگیری ماشین روی آنها آموزش دیده است، بسازد. این به ویژه به درک توزیع داده‌های پیچیده کمک می‌کند. اما منظور من از توضیحات مبتنی بر مثال چیست؟ ما اغلب از آنها در شغل و زندگی روزمره خود استفاده می‌کنیم. اجازه دهید با چند مثال شروع کنیم (Aamodt & Plaza, 1994).

یک پزشک، بیمار را با سرفه غیرمعمول و تب خفیف می‌بیند. علائم بیمار او را به یاد بیمار دیگری می‌اندازد که سال‌ها پیش با علائم مشابه داشت. او مشکوک است که بیمار فعلی اش ممکن است به همان بیماری مبتلا باشد و برای آزمایش این بیماری خاص نمونه خون می‌گیرد.

یک دانشمند داده روی پروژه جدیدی برای یکی از مشتریان خود کار می‌کند: تجزیه و تحلیل عوامل خطر که منجر به خرابی ماشین‌های تولید صفحه کلید می‌شود. دانشمند داده پروژه مشابهی را که روی آن کار کرده بود به خاطر می‌آورد و از بخش‌هایی از کد پروژه قدیمی دوباره استفاده می‌کند زیرا فکر می‌کند مشتری همان تحلیل را می‌خواهد.

بچه گربه‌ای روی طاقچه پنجه خانه‌ای در حال سوختن و خالی از سکنه نشسته است. آتش نشانی سر می‌رسد و یکی از آتش نشان‌ها برای لحظه‌ای فکر می‌کند که آیا می‌تواند برای نجات بچه گربه به داخل ساختمان برود یا خیر. او موارد مشابهی را در زندگی خود به عنوان آتش نشان به یاد می‌آورد: خانه‌های چوبی قدیمی که مدتی

است به آرامی می سوختند، اغلب ناپایدار بودند و در نهایت فرو ریختند. به دلیل شباهت این مورد، تصمیم می گیرد وارد نشود، زیرا خطر ریزش خانه خیلی زیاد است. خوشبختانه، بچه گربه از پنجره به بیرون می پرد، به سلامت فرود می آید و هیچ کس در آتش آسیبی نمی بیند. پایان خوش.

این داستانها نحوه تفکر ما انسانها را در مثالها یا قیاسها نشان می دهد. طرح اولیه توضیحات مبتنی بر مثال این است: چیز B شبیه چیز A است و A باعث Y است، بنابراین من پیش‌بینی می کنم که B نیز باعث Y شود. به طور ضمنی، برخی از رویکردهای یادگیری ماشین مبتنی بر مثال هستند. درختان تصمیم، داده‌ها را بر اساس شباهت نقاط داده در ویژگی‌هایی که برای پیش‌بینی هدف مهم هستند، به گره‌ها تقسیم کنید. یک درخت تصمیم، پیش‌بینی یک نمونه داده جدید را با یافتن نمونه‌های مشابه (= در همان گره پایانی) و برگرداندن میانگین نتایج آن نمونه‌ها به عنوان پیش‌بینی، کار می کند. روش k نزدیکترین همسایه (knn) به صراحت با رویکرد پیش‌بینی‌های مبتنی بر مثال کار می کند. برای مثال جدید، یک مدل knn، k نزدیکترین همسایگان (مثلاً $k=3$ نزدیکترین نمونه) را تعیین می کند و میانگین نتایج آن همسایگان را به عنوان یک پیش‌بینی بر می گرداند. پیش‌بینی یک knn را می توان با برگرداندن k همسایه توضیح داد، که - دوباره - تنها در صورتی معنی دار است که راه خوبی برای نمایش یک نمونه واحد داشته باشیم.

روش‌های تفسیر زیر همگی مبتنی بر مثال هستند:

- توضیحات خلاف واقع به ما می گوید که چگونه یک نمونه باید تغییر کند تا پیش‌بینی آن به طور قابل توجهی تغییر کند. با ایجاد نمونه‌های خلاف واقع، ما می آموزیم که مدل چگونه پیش‌بینی‌های خود را انجام می دهد و می توانیم پیش‌بینی‌های فردی را توضیح دهیم.
- مثال‌های متخصص¹، خلاف واقع‌هایی هستند که برای فریب دادن مدل‌های یادگیری ماشین استفاده می شوند. تاکید بر سبک سنگین کردن² پیش‌بینی است و نه توضیح دادن آن.
- نمونه‌های اولیه³ انتخابی از نمونه‌های نماینده از داده‌ها هستند و انتقادات نمونه‌هایی هستند که به خوبی توسط آن نمونه‌های اولیه نمایندگی نمی شوند (Aamodt & Plaza, 1994).
- نمونه‌های تأثیرگذار، نقاط داده آموزشی هستند که بیشترین تأثیر را برای پارامترهای یک مدل پیش‌بینی یا خود پیش‌بینی‌ها داشтند. شناسایی و تجزیه و تحلیل نمونه‌های نمونه‌های تأثیرگذار به یافتن مشکلات داده‌ها، اشکال زدایی مدل و درک بهتر رفتار مدل کمک می کند.
- مدل k نزدیکترین همسایه: یک مدل یادگیری ماشین (قابل تفسیر) بر اساس مثال‌ها.

¹ Adversarial examples

² flipping

³ Prototypes

فصل ۸ روش‌های آگنوستیک مدل کلی

روش‌های کلی، رفتار متوسط یک مدل یادگیری ماشین را توصیف می‌کنند. نقطه مقابل روش‌های کلی روش‌های محلی هستند. روش‌های کلی اغلب به صورت مقادیر مورد انتظار بر اساس توزیع داده‌ها بیان می‌شوند. به عنوان مثال، نمودار وابستگی جزئی یک نمودار اثر ویژگی است که پیش‌بینی مورد انتظار را نشان می‌دهد. چون روش‌های تفسیر کلی، رفتار متوسط را توصیف می‌کنند، زمانی که سازنده مدل می‌خواهد مکانیسم‌های کلی در داده‌ها را بفهمد یا یک مدل را اشکال‌زدایی کند، بسیار مفید هستند.

در این کتاب، با تکنیک‌های تفسیر کلی آگنوستیک مدل زیر آشنا خواهید شد:

- نمودار وابستگی جزئی یک روش اثر ویژگی است.
- نمودار جلوه‌های محلی انباسته یکی دیگر از روش‌های اثر ویژگی است که در صورت وابسته بودن ویژگی‌ها کار می‌کند.
- تعامل ویژگی (آماره H) کمی می‌کند که پیش‌بینی تا چه حد نتیجه اثرات مشترک ویژگی‌ها است.
- تجزیه تابعی (Functional decomposition)، یک ایده مرکزی از تفسیرپذیری می‌باشد و تکنیکی است که تابع پیش‌بینی پیچیده را به بخش‌های کوچک‌تر تجزیه می‌کند.
- اهمیت ویژگی جایگشتی، اهمیت یک ویژگی را با استفاده از اندازه گیری خطأ در هنگام جایگزینی یک ویژگی محاسبه می‌کند.
- مدل‌های جایگزین کلی مدل اصلی را با مدلی ساده‌تر برای تفسیر جایگزین می‌کند.
- نمونه‌های اولیه و انتقادات نشان‌دهنده نقاط داده نماینده ای هستند که می‌توانند برای افزایش تفسیرپذیری استفاده شوند.

۸.۱ نمودار وابستگی جزئی (PDP¹)

نمودار وابستگی جزئی (به اختصار PDP یا نمودار PD) اثر حاشیه ای^۲ یک یا دو ویژگی بر خروجی پیش‌بینی شده یک مدل یادگیری ماشین را نشان می‌دهد (Friedman, 2001). نمودار وابستگی جزئی می‌تواند نشان دهد که آیا رابطه بین هدف و یک ویژگی خطی، یکنواخت یا پیچیده تر است. به عنوان مثال، هنگامی که به یک مدل رگرسیون خطی اعمال می‌شود، نمودارهای وابستگی جزئی همیشه یک رابطه خطی را نشان می‌دهند.

تابع وابستگی جزئی برای رگرسیون به صورت زیر تعریف می‌شود:

$$\hat{f}_S(x_S) = E_{X_C}[\hat{f}(x_S, X_C)] = \int \hat{f}(x_S, X_C) d\mathbb{P}(X_C)$$

x_S ویژگی‌هایی هستند که تابع وابستگی جزئی باید برای آنها ترسیم شود و X_C دیگر ویژگی‌های مورد استفاده در مدل یادگیری ماشین \hat{f} هستند، که در اینجا به عنوان متغیرهای تصادفی در نظر گرفته می‌شوند. معمولاً فقط یک یا دو ویژگی در مجموعه S وجود دارد. ویژگی‌های (x_S و X_C) در S آنها بی جزئی هستند که می‌خواهیم تأثیر آنها را بر پیش‌بینی بدانیم. ترکیب بردارهای ویژگی x_S و X_C فضای کلی ویژگی x را تشکیل می‌دهند. وابستگی جزئی با تجمعیح حاشیه ای خروجی مدل یادگیری ماشین بر روی توزیع ویژگی‌های مجموعه C عمل می‌کند، به طوری که تابع، رابطه بین مجموعه ویژگی‌های مورد علاقه S و نتیجه پیش‌بینی شده را نشان می‌دهد. با به تجمعیح حاشیه ای سایر ویژگی‌ها، تابعی به دست می‌آوریم که فقط به ویژگی‌های S بستگی دارد، تعامل با سایر ویژگی‌ها نیز گنجانده شده است.

تابع جزئی \hat{f} با محاسبه میانگین‌ها در داده‌های آموزشی تخمین زده می‌شود که این روش به مونت کارلو نیز معروف است:

$$\hat{f}_S(x_S) = \frac{1}{n} \sum_{i=1}^n \hat{f}(x_S, x_C^{(i)})$$

تابع جزئی به ما می‌گوید که برای مقدار (های) معین ویژگی S ، میانگین اثر حاشیه ای در پیش‌بینی چقدر است. در این فرمول، $x_C^{(i)}$ مقادیر واقعی ویژگی از مجموعه داده برای ویژگی‌هایی هستند که ما به آنها علاقه نداریم و n تعداد نمونه‌های موجود در مجموعه داده است. یک فرض PDP این است که ویژگی‌های موجود در C با ویژگی‌های S همبستگی ندارند. اگر این فرض نقض شود، میانگین‌های محاسبه شده برای نمودار وابستگی جزئی شامل نقاط داده‌ای خواهد بود که بسیار بعید یا حتی غیرممکن هستند (معایب را ببینید).

¹ Partial Dependence Plot

² marginal

برای طبقه‌بندی که در آن مدل یادگیری ماشین احتمالات را پیش بینی می‌کند، نمودار وابستگی جزئی احتمال را برای یک کلاس خاص به شرط مقادیر مختلف برای ویژگی(ها) در S نشان می‌دهد. یک راه آسان برای مقابله با چندین کلاس ترسیم یک خط یا نمودار برای هر کلاس است.

نمودار وابستگی جزئی یک روش کلی است: این روش همه نمونه‌ها را در نظر می‌گیرد و بیانیه ای در مورد رابطه کلی یک ویژگی با خروجی پیش‌بینی شده ارائه می‌دهد.

ویژگی‌های طبقه‌ای

تا اینجا فقط ویژگی‌های عددی را در نظر گرفته ایم. برای ویژگی‌های طبقه‌بندی، محاسبه وابستگی جزئی بسیار آسان است. برای هر یک از طبقه‌ها، با وادار کردن همه نمونه‌های داده به داشتن طبقه یکسان، یک تخمین PDP دریافت می‌کنیم. به عنوان مثال، اگر به مجموعه داده‌های اجاره دوچرخه نگاه کنیم و به نمودار وابستگی جزئی برای فصل علاقه‌مند باشیم، چهار عدد به دست می‌آوریم، یکی برای هر فصل. برای محاسبه مقدار "تابستان"، فصل تمام نمونه‌های داده را با "تابستان" جایگزین می‌کنیم و پیش‌بینی‌ها را میانگین می‌کنیم.

۸.۱.۱ اهمیت ویژگی مبتنی بر PDP

Greenwell et al (۲۰۱۸) یک معیار اهمیت ویژگی مبتنی بر وابستگی جزئی را پیشنهاد کردند. ایده اصلی این است که وقتی PDP یک ویژگی بی تغییر است آن ویژگی مهم نیست، و هر چه PDP بیشتر تغییر کند، آن ویژگی مهم‌تر است. برای ویژگی‌های عددی، اهمیت به عنوان انحرافات هر مقدار ویژگی منحصر به فرد از منحنی میانگین تعریف می‌شود:

$$I(x_S) = \sqrt{\frac{1}{K-1} \sum_{k=1}^K \left(\hat{f}_S(x_S^{(k)}) - \frac{1}{K} \sum_{k=1}^K \hat{f}_S(x_S^{(k)}) \right)^2}$$

توجه داشته باشید که در اینجا $x_S^{(k)}$ مقادیر منحصر به فرد K از ویژگی X_S هستند. برای ویژگی‌های دسته بندی داریم:

$$I(x_S) = \left(\max_k \left(\hat{f}_S(x_S^{(k)}) \right) - \min_k \left(\hat{f}_S(x_S^{(k)}) \right) \right) / 4$$

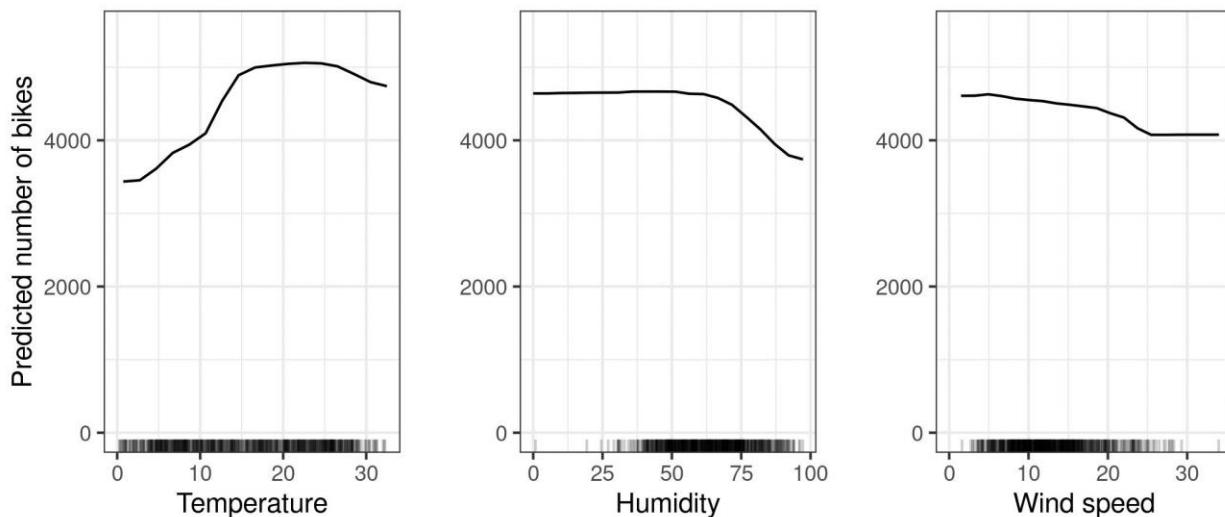
این محدوده مقادیر PDP برای طبقه‌های منحصر به فرد تقسیم بر چهار است. این روش عجیب برای محاسبه انحراف، قانون محدوده^۱ نامیده می‌شود. این قانون، هنگامی که شما فقط محدوده را می‌دانید، برای به دست آوردن یک تخمین تقریبی برای انحراف کمک می‌کند. مخرج چهار از توزیع نرمال استاندارد می‌آید: در توزیع نرمال، ۹۵ درصد داده‌ها در محدوده منهای دو و به علاوه دو انحراف معیار حول میانگین قرار دارند. بنابراین محدوده تقسیم بر چهار یک تخمین تقریبی است که احتمالاً واریانس واقعی را کمتر در نظر می‌گیرد.

¹ range rule

این اهمیت ویژگی مبتنی بر PDP باید با دقت تفسیر شود. این مقدار، فقط اثر اصلی ویژگی را در نظر می‌گیرد و تعاملات احتمالی ویژگی را نادیده می‌گیرد. یک ویژگی می‌تواند بر اساس روش‌های دیگر مانند اهمیت ویژگی جایگشت (permutation feature importance) بسیار مهم باشد، اما PDP آن می‌تواند بی‌تغییر باشد زیرا این ویژگی عمده‌ای از طریق تعامل با سایر ویژگی‌ها بر پیش‌بینی تأثیر می‌گذارد. یکی دیگر از اشکالات این معیار این است که بر روی مقادیر منحصر به فرد تعریف شده است. یک مقدار ویژگی خاص با تنها یک نمونه، وزن یکسانی با مقداری که تعداد زیادی نمونه دارد، در محاسبه اهمیت می‌گیرد.

۸.۱.۲ مثال‌ها

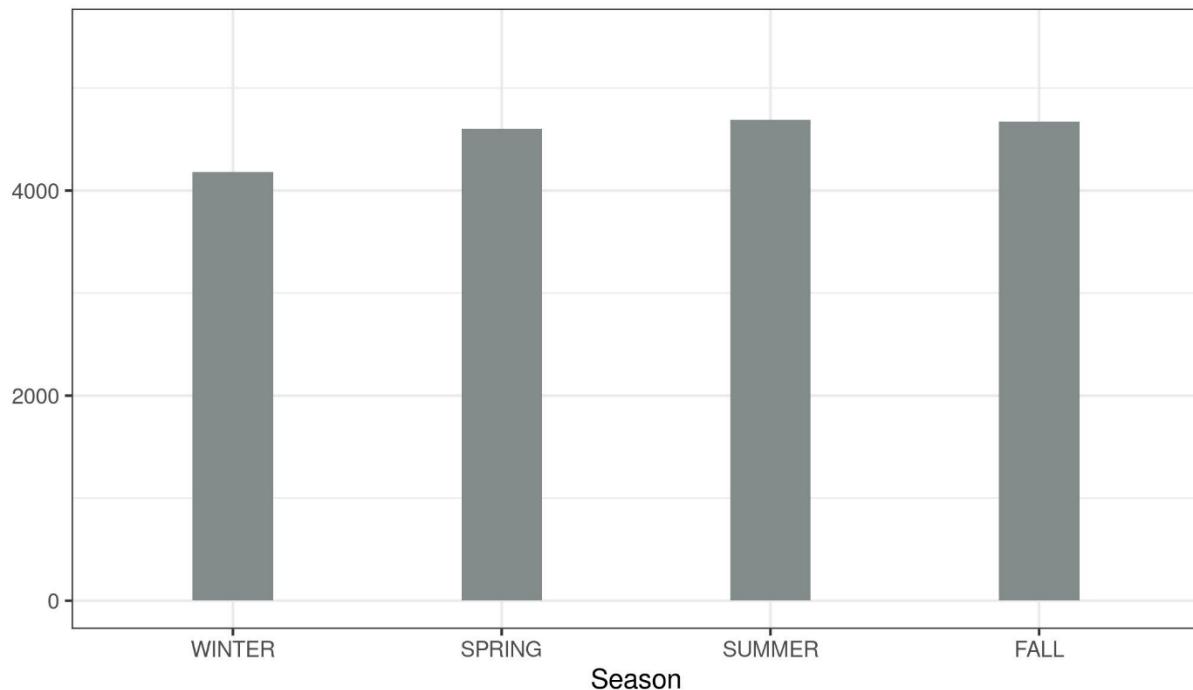
در عمل، مجموعه ویژگی‌های S معمولاً فقط شامل یک یا حداقل دو ویژگی است، زیرا یک ویژگی نمودارهای دوبعدی و دو ویژگی نمودارهای سه بعدی تولید می‌کنند. درک بیش از دو ویژگی آن بسیار مشکل است. حتی درک نمودار سه بعدی روی کاغذ یا مانیتور دو بعدی مشکل است. اجازه دهید به مثال رگرسیون برگردیم، که در آن تعداد دوچرخه‌هایی را که در یک روز معین اجاره می‌شوند، پیش‌بینی می‌کنیم. ابتدا مدل یادگیری ماشین را برآش می‌کنیم، سپس وابستگی‌های جزئی را تجزیه و تحلیل می‌کنیم. در این مورد، ما یک جنگل تصادفی برای پیش‌بینی تعداد دوچرخه‌ها و استفاده از نمودار وابستگی جزئی برای تجسم روابطی که مدل آموخته است، پیاده‌سازی کرده‌ایم. تأثیر ویژگی‌های آب و هوا بر تعداد دوچرخه‌های پیش‌بینی شده در شکل زیر نشان داده شده است.



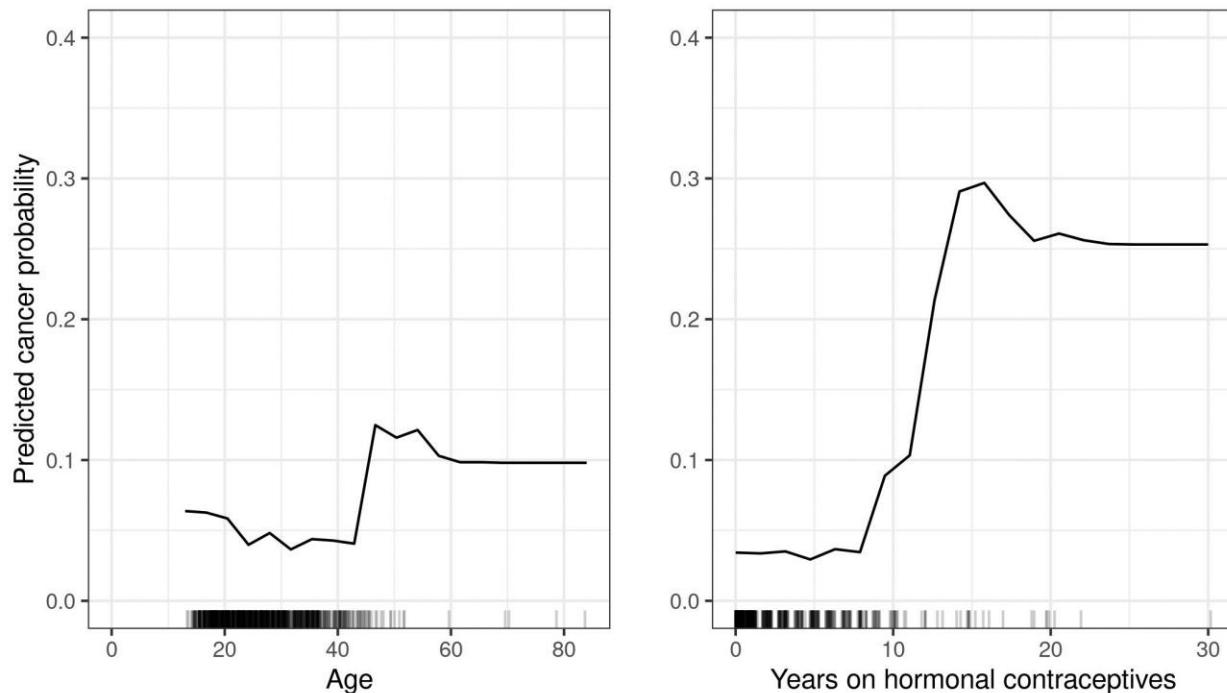
شکل ۸.۱: PDP ها برای مدل پیش‌بینی تعداد دوچرخه و دما، رطوبت و سرعت باد. بیشترین تفاوت را می‌توان در دما مشاهده کرد. هر چه گرم‌تر باشد، دوچرخه‌های بیشتری اجاره می‌شود. این روند تا ۲۰ درجه سانتیگراد بالا می‌رود، سپس صاف می‌شود و در ۳۰ کمی کاهش می‌یابد. علائم روی محور x توزیع داده‌ها را نشان می‌دهد.

برای آب و هوای گرم اما نه خیلی گرم، این مدل به طور متوسط تعداد زیادی دوچرخه کرایه ای را پیش‌بینی می‌کند. دوچرخه‌سواران بالقوه شدیداً از اجاره دوچرخه در زمانی که رطوبت از ۶۰٪ فراتر می‌رود، منصرف می‌شوند. علاوه بر این، هر چه باد بیشتر باشد، افراد کمتری دوست دارند دوچرخه سواری کنند، که منطقی است. جالب اینجاست که وقتی سرعت باد از ۲۵ به ۳۵ کیلومتر در ساعت می‌رسد، تعداد پیش‌بینی شده اجاره دوچرخه کاهش نمی‌یابد، اما داده‌های آموزشی زیادی در این بازه وجود ندارد (به تعداد کم علائم روی محور x دقت کنید)، بنابراین مدل یادگیری ماشین احتمالاً نمی‌تواند آموزش مناسبی برای پیش‌بینی این محدوده دریافت کند. حداقل به طور شهودی، من انتظار دارم با افزایش سرعت باد، تعداد دوچرخه‌ها کاهش یابد، به خصوص زمانی که سرعت باد بسیار زیاد است.

برای نشان دادن یک نمودار وابستگی جزئی با یک ویژگی طبقه‌بندی شده، ما تأثیر ویژگی فصل را بر اجاره دوچرخه پیش‌بینی شده بررسی می‌کنیم.

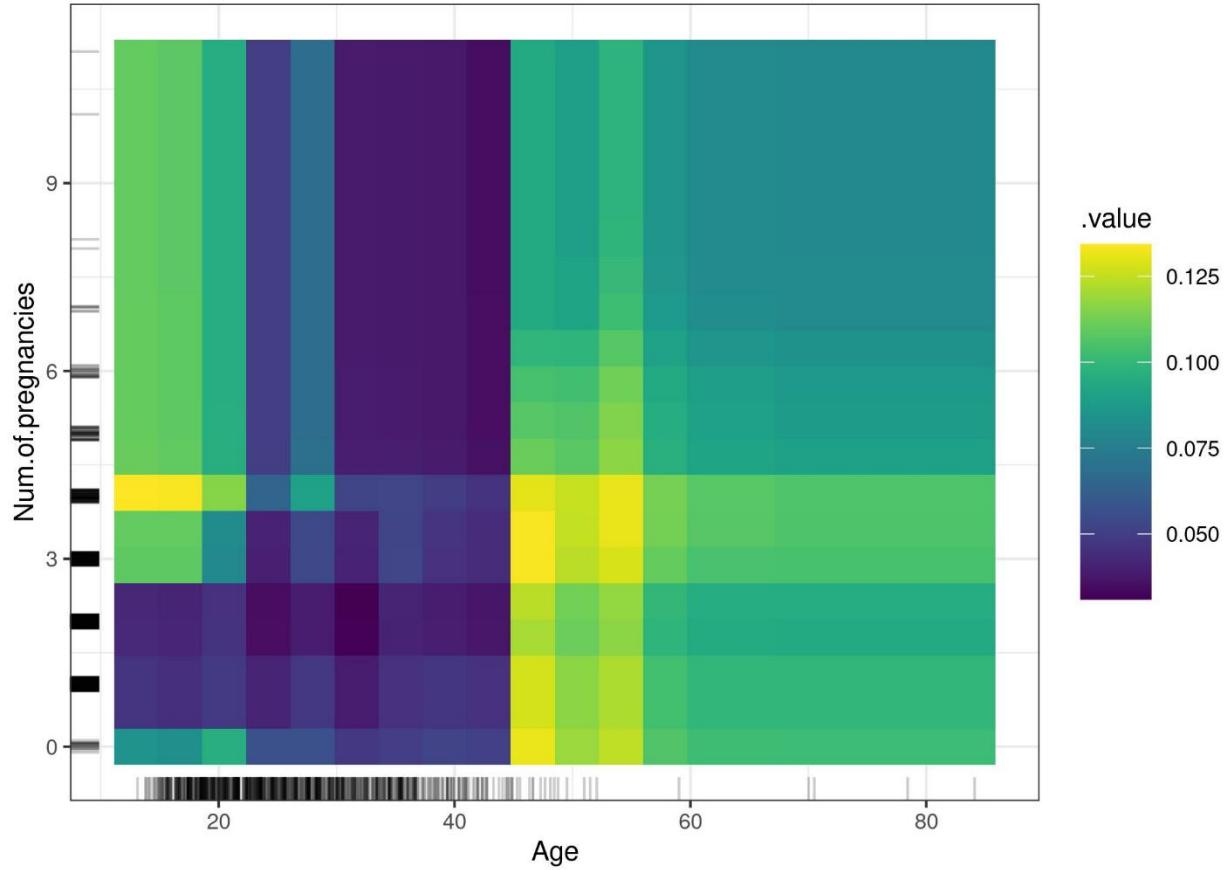


شکل ۸.۲: PDP‌ها برای مدل پیش‌بینی تعداد دوچرخه و فصل. بطور غیرمنتظره تمام فصول اثر مشابهی بر پیش‌بینی‌های مدل نشان می‌دهند، فقط برای زمستان مدل، اجاره دوچرخه کمتری را پیش‌بینی می‌کند. همچنین وابستگی جزئی را برای طبقه‌بندی سرطان دهانه رحم محاسبه می‌کنیم. این بار یک جنگل تصادفی را برای پیش‌بینی اینکه آیا یک زن ممکن است بر اساس عوامل خطر به سرطان دهانه رحم مبتلا شود، آموزش می‌دهیم. ما وابستگی جزئی احتمال سرطان به ویژگی‌های مختلف را برای جنگل تصادفی محاسبه و ترسیم می‌کنیم:



شکل ۸.۳: PDPهای احتمال سرطان بر اساس سن و سال‌های استفاده از داروهای ضد بارداری هورمونی. برای سن، PDP نشان می‌دهد که احتمال تا ۴۰ سالگی کم است و بعد از آن افزایش می‌یابد. هر چه سال‌های استفاده از داروهای ضد بارداری هورمونی بیشتر باشد، خطر سرطان پیش‌بینی شده بیشتر می‌شود، مخصوصاً بعد از ۱۰ سال. برای هر دو ویژگی، برای مقادیر بالا، نقاط داده زیادی در دسترس نبود، بنابراین برآوردهای PD در آن مناطق کمتر قابل اعتماد هستند.

ما همچنین می‌توانیم وابستگی جزئی دو ویژگی را در یک نمودار ترسیم کنیم:



شکل ۸.۴: PDP احتمال سرطان و اثر تعامل سن و تعداد حاملگی. این نمودار افزایش احتمال ابتلا به سرطان را در ۴۵ سالگی نشان می‌دهد. برای سنین زیر ۲۵ سال، زنانی که ۱ یا ۲ بارداری داشتند، در مقایسه با زنانی که ۰ یا بیشتر از ۲ بارداری داشتند، خطر سرطان پیش‌بینی شده کمتری داشتند. اما هنگام نتیجه گیری مراقب باشید: این موضوع، ممکن است فقط یک همبستگی باشد و نه علیت!

۸.۱.۳ مزایا

محاسبه نمودارهای وابستگی جزئی شهودی است: تابع وابستگی جزئی در یک مقدار مشخصه خاص، میانگین پیش‌بینی را نشان می‌دهد اگر همه نقاط داده را مجبور کنیم آن مقدار ویژگی را فرض کنند. بر اساس تجربه من، افراد غیرمتخصص معمولاً ایده PDP ها را به سرعت درک می‌کنند.

اگر مشخصه ای که PDP را برای آن محاسبه کرده اید با ویژگی‌های دیگر همبستگی نداشته باشد، PDP ها به خوبی نشان می‌دهند که چگونه ویژگی به طور متوسط بر پیش‌بینی تأثیر می‌گذارد. در مورد غیر همبسته، تفسیر واضح است: نمودار وابستگی جزئی نشان می‌دهد که چگونه میانگین پیش‌بینی در مجموعه‌داده شما با تغییر ویژگی زام تغییر می‌کند. وقتی ویژگی‌ها با هم همبستگی داشته باشند، اوضاع پیچیده‌تر است، معایب را نیز ببینید.

طرح‌های وابستگی جزئی به راحتی قابل پیاده‌سازی هستند.

محاسبه برای نمودارهای وابستگی جزئی یک تفسیر علی دارد. ما روی یک ویژگی مداخله می‌کنیم و تغییرات پیش‌بینی‌ها را اندازه می‌گیریم. در انجام این کار، ما رابطه علی بین ویژگی و پیش‌بینی را تحلیل می‌کنیم (Zhao & Hastie, 2021). این رابطه برای مدل علی است (زیرا ما صریحاً خروجی را به عنوان تابعی از ویژگی‌ها، مدل می‌کنیم). اما این رابطه علی برای مدل است نه لزوماً برای دنیای واقعی!

۸.۱.۴ معایب

حداکثر واقعی تعداد ویژگی‌ها در یک تابع وابستگی جزئی دو است. این تقصیر PDP‌ها نیست، بلکه از نمایش دو بعدی (کاغذ یا صفحه‌نمایش) و همچنین ناتوانی ما در تصور بیش از ۳ بعد است.

برخی از نمودارهای PD توزیع ویژگی را نشان نمی‌دهند. حذف توزیع ممکن است گمراه کننده باشد، زیرا ممکن است مناطقی را که تقریباً هیچ داده ای ندارند، بیش از حد تفسیر کنید. این مشکل با نشان دادن یک علامت (نقاط داده در محور x) یا هیستوگرام به راحتی حل می‌شود.

فرض استقلال بزرگ‌ترین مشکل نمودارهای PD است. فرض بر این است که ویژگی‌ها (هایی) که وابستگی جزئی برای آنها محاسبه می‌شود با سایر ویژگی‌ها همبستگی ندارند. برای مثال، فرض کنید با توجه به وزن و قد فرد می‌خواهید سرعت راه رفتن یک فرد را پیش‌بینی کنید. برای وابستگی جزئی یکی از ویژگی‌ها، مثلاً قد، فرض می‌کنیم که سایر ویژگی‌ها (وزن) با قد همبستگی ندارند، که بدیهی است یک فرض نادرست است. برای محاسبه PDP در قد معین (مثلاً ۲۰۰ سانتی‌متر)، توزیع حاشیه‌ای وزن را میانگین می‌گیریم، که ممکن است شامل وزن کمتر از ۵۰ کیلوگرم باشد، که برای یک فرد ۲ متری غیر واقعی است. به عبارت دیگر: وقتی ویژگی‌ها با هم همبستگی دارند، ما نقاط داده جدیدی را در مناطقی از توزیع ویژگی ایجاد می‌کنیم که احتمال وقوع بسیار کم دارند (به عنوان مثال بعید است که فردی ۲ متر قد داشته باشد اما وزن آن کمتر از ۵۰ کیلوگرم باشد). یکی از راه حل‌های این مشکل است نمودارهای اثر محلی انباسته^۱ یا نمودارهای کوتاه ALE که با توزیع شرطی^۲ به جای توزیع حاشیه ای کار می‌کنند.

اثرات ناهمگن^۳ ممکن است پنهان باشد زیرا نمودارهای PD فقط اثرات حاشیه ای متوسط را نشان می‌دهند. فرض کنید برای یک ویژگی، نیمی از نقاط داده شما ارتباط مثبتی با پیش‌بینی داشته باشند - هر چه مقدار ویژگی بزرگ‌تر باشد، پیش‌بینی بزرگ‌تر است - و نیمی دیگر دارای ارتباط منفی باشد - هر چه مقدار ویژگی کوچک‌تر باشد، پیش‌بینی بزرگ‌تر است. منحنی PD می‌تواند یک خط افقی شود، زیرا اثرات هر دو نیمه

¹ Accumulated Local Effect plots

² Conditional

³ Heterogeneous

مجموعه‌داده می‌تواند یکدیگر را خنثی کند. سپس نتیجه می‌گیرید که این ویژگی تاثیری بر پیش‌بینی ندارد. با ترسیم منحنی‌های انتظار شرطی فردی^۱ به جای خط تجمعی، می‌توانیم اثرات ناهمگن را کشف کنیم.

۸.۱.۵ نرم افزار و جایگزین

تعدادی پکیج R وجود دارد که PDP‌ها را پیاده‌سازی کرده‌اند. من از پکیج iml برای مثال‌ها استفاده کردم، اما پکیج‌های pdp یا DALEX نیز وجود دارد. در Python، نمودارهای وابستگی جزئی در scikit-learn پیاده‌سازی شده‌اند و می‌توان از دستور PDPBox استفاده نمود. جایگزین‌های PDP ارائه شده در این کتاب نمودارهای ALE و منحنی‌های ICE هستند.

^۱ individual conditional expectation curves

۸.۲ نمودار اثرات محلی انباشته (ALE¹)

اثرات محلی انباشته (Apley & Zhu, 2020) توصیف می‌دهد که چگونه ویژگی‌ها به طور متوسط بر پیش‌بینی یک مدل یادگیری ماشین تأثیر می‌گذارند. نمودارهای ALE یک جایگزین سریعتر و بی طرفانه برای نمودارهای وابستگی جزئی (PDP) هستند.

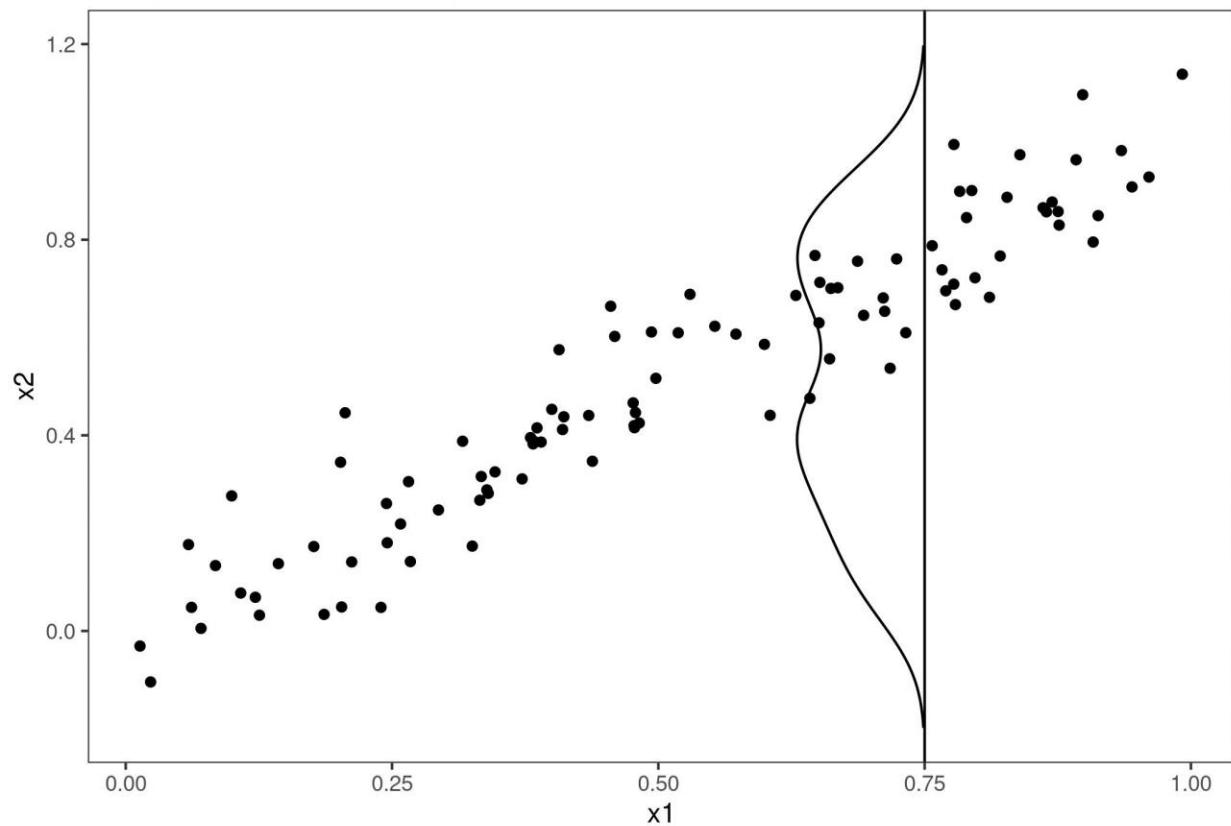
توصیه می‌کنم ابتدا فصل مربوط به نمودارهای وابستگی جزئی را مطالعه کنید، زیرا درک آنها آسان‌تر است و هر دو روش هدف یکسانی دارند: هر دو توضیح می‌دهند که چگونه یک ویژگی به طور متوسط بر پیش‌بینی تأثیر می‌گذارد. در بخش بعدی، می‌خواهم شما را متلاعند کنم که نمودارهای وابستگی جزئی زمانی که ویژگی‌ها همبستگی دارند، یک مشکل جدی دارند.

۸.۲.۱ انگیزه و شهود

اگر ویژگی‌های یک مدل یادگیری ماشین با هم همبستگی داشته باشند، نمی‌توان به نمودار وابستگی جزئی اعتقاد کرد. محاسبه یک نمودار وابستگی جزئی برای یک ویژگی که به شدت با سایر ویژگی‌ها همبستگی دارد، میانگین پیش‌بینی نمونه‌هایی مصنوعی است که در واقعیت بعید هستند. این کار می‌تواند تا حد زیادی اثر ویژگی تخمینی را سودار کند. تصور کنید که نمودارهای وابستگی جزئی را برای یک مدل یادگیری ماشین محاسبه کنید که ارزش یک خانه را بسته به تعداد اتاق‌ها و اندازه‌ی نشیمن پیش‌بینی می‌کند. ما به تأثیر اندازه نشیمن بر مقدار پیش‌بینی شده علاقه مندیم. برای یادآوری، دستورالعمل محاسبه نمودارهای وابستگی جزئی به این صورت است: ۱) ویژگی را انتخاب کنید. ۲) شبکه را تعریف کنید. ۳) در هر مقدار شبکه: (الف) ویژگی را با مقدار شبکه جایگزین کنید و ب) از پیش‌بینی‌ها میانگین بگیرید. ۴) منحنی را رسم کنید. برای محاسبه اولین مقدار شبکه PDP، مثلاً ۳۰ مترمربع، اندازه‌ی نشیمن را برای همه موارد ۳۰ متر مربع جایگزین می‌کنیم، حتی برای خانه‌هایی با ۱۰ اتاق. به نظر من یک خانه بسیار غیر عادی است. نمودار وابستگی جزئی شامل این خانه‌های غیرواقعی در تخمین اثر ویژگی می‌شود و وامنود می‌کند که همه چیز خوب است. شکل زیر دو ویژگی مرتبط را نشان می‌دهد و اینکه چگونه روش نمودار وابستگی جزئی از پیش‌بینی‌های نمونه‌هایی بعید میانگین می‌گیرد.

¹ Accumulated Local Effects

Marginal distribution $P(x_2)$

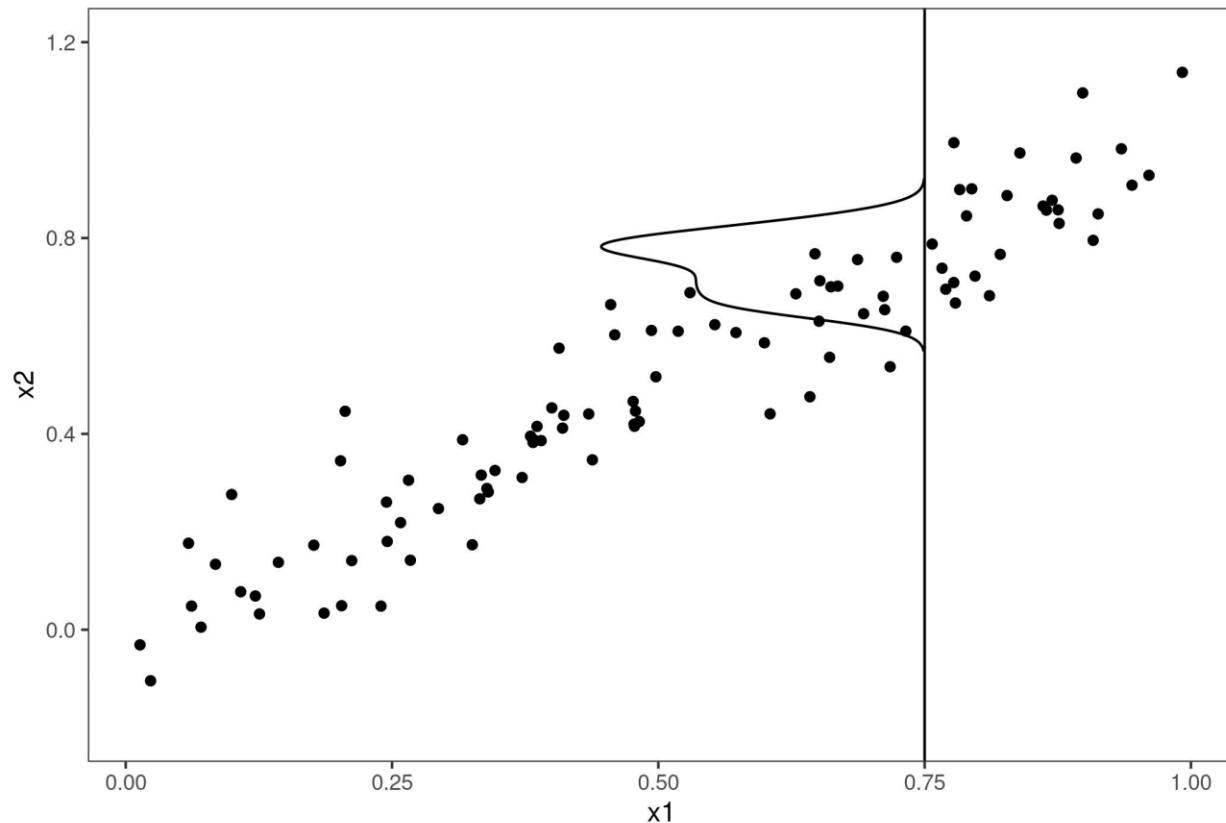


شکل ۸.۵: ویژگی‌های x_1 و x_2 با همبستگی شدید. برای محاسبه اثر ویژگی x_1 در $0/75$ ، x_1 همه نمونه‌ها را با $0/75$ جایگزین می‌کند، به اشتباہ فرض می‌کنیم که توزیع در $0/75 = x_1$ با توزیع حاشیه ای x_2 (خط عمودی) یکسان است. این کار منجر به ترکیبات بعید از x_1 و x_2 می‌شود (به عنوان مثال $0/2 = x_2$ در $= x_1 0/75$) که PDP برای محاسبه اثر میانگین از این ترکیبات استفاده می‌کند.

برای به دست آوردن تخمین اثر ویژگی‌هایی که با یکدیگر همبستگی دارند، چه کاری می‌توانیم انجام دهیم؟ ما می‌توانیم میانگین گیری از توزیع شرطی ویژگی را کنیم، به این معنی که در یک مقدار شبکه ای x_1 پیش‌بینی‌های نمونه‌هایی با مقدار مشابه x_1 را میانگین بگیریم. راه حل برای محاسبه اثرات ویژگی با استفاده از توزیع شرطی، نمودارهای حاشیه ای یا M-Plots نامیده می‌شود (نام گیج کننده است، زیرا بر اساس توزیع شرطی است، نه توزیع حاشیه ای). صبر کن، آیا به شما قول ندادم که در مورد نمودارهای ALE صحبت کنید؟ راه حلی نیست که ما به دنبال آن هستیم. چرا M-Plots مشکل ما را حل نمی‌کند؟ اگر میانگین پیش‌بینی تمام خانه‌ها را در حدود 30 مترمربع به دست آوریم، تاثیر ترکیب اندازه‌ی نشیمن و تعداد اتاق‌ها را تخمین می‌زنیم به این دلیل که آن‌ها با یکدیگر همبستگی دارند. فرض کنید اندازه‌ی نشیمن هیچ تاثیری بر ارزش پیش‌بینی شده یک خانه ندارد، فقط تعداد اتاق‌ها تاثیری دارد. M-Plot همچنان نشان می‌دهد که اندازه‌ی

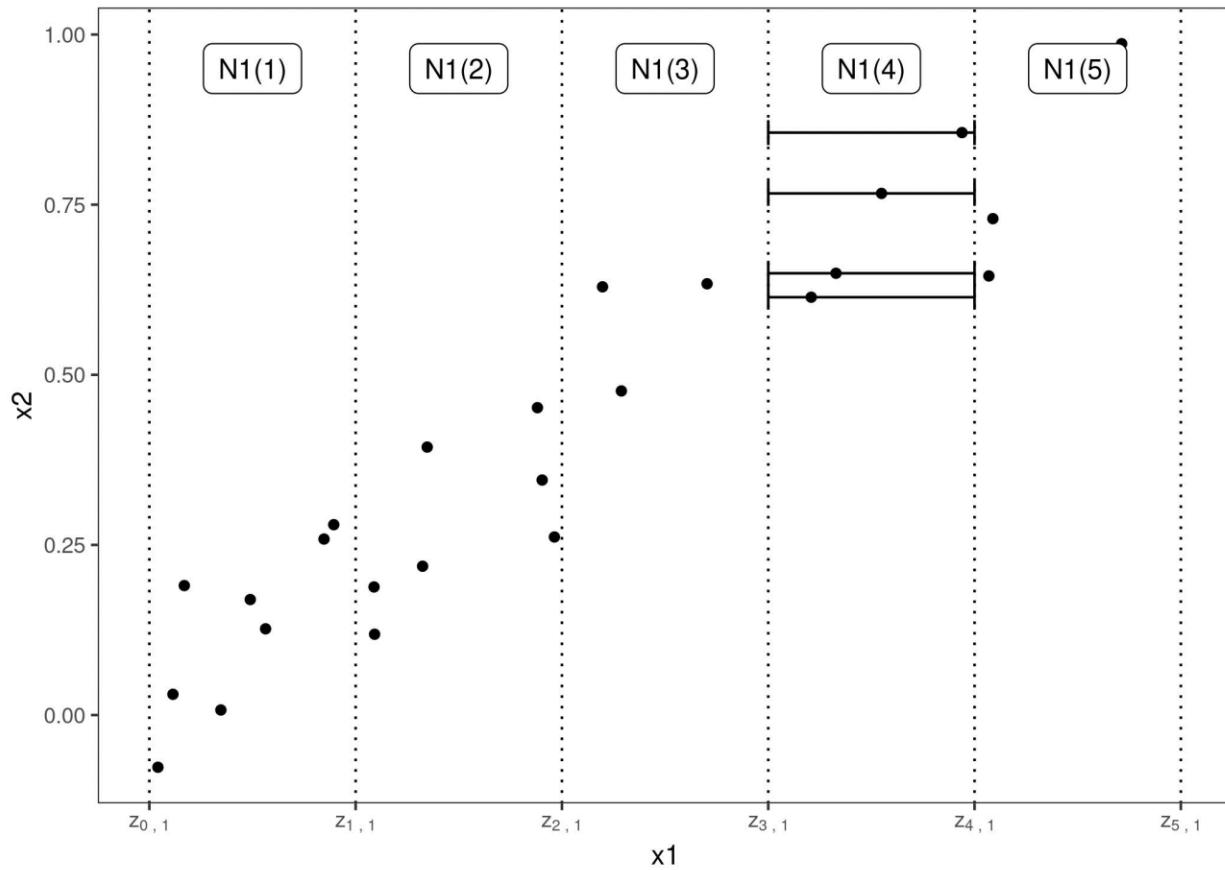
نشیمن مقدار پیش‌بینی شده را افزایش می‌دهد، زیرا تعداد اتاق‌ها با اندازه‌ی نشیمن افزایش می‌یابد. نمودار زیر نحوه کار M-Plots را برای دو ویژگی همبسته نشان می‌دهد.

Conditional distribution $P(x_2|x_1=0.75)$



شکل ۸.۶: ویژگی‌های x_1 و x_2 با همبستگی شدید. M-Plots روی توزیع شرطی میانگین می‌گیرد. در اینجا توزیع شرطی x_2 در $x_1=0.75$ است. میانگین گرفتن از پیش‌بینی‌های محلی منجر به مخلوط شدن اثرات هر دو ویژگی می‌شود.

M-Plots از میانگین پیش‌بینی نمونه‌های داده بعید اجتناب می‌کند، اما آنها اثر یک ویژگی را با اثرات همه ویژگی‌های همبسته مخلوط می‌کنند. نمودارهای ALE این مشکل را با محاسبه بر اساس توزیع شرطی ویژگی‌ها - تفاوت در پیش‌بینی‌ها به جای میانگین‌ها حل می‌کند. برای تأثیر مساحت نشیمن در ۳۰ متر مربع، روش ALE از تمام خانه‌های با مساحت حدود ۳۰ متر مربع استفاده می‌کند، پیش‌بینی‌های مدل را برای خانه‌ها از ۲۹ تا ۳۱ متر مربع به دست می‌آورد. این روش به ما اثر خالص مساحت نشیمن را می‌دهد و اثر ویژگی را با اثرات ویژگی‌های همبسته مخلوط نمی‌کند. استفاده از تفاوت‌ها اثر سایر ویژگی‌ها را مسدود می‌کند. نمودار زیر نحوه محاسبه نمودارهای ALE را نشان می‌دهد.



شکل ۸.۷: محاسبه ALE برای ویژگی x_1 که با x_2 همبستگی دارد. ابتدا ویژگی را به بازه هایی (خطوط عمودی) تقسیم می کنیم. برای نمونه های داده (نقاط) در یک بازه، ویژگی را با حد بالا و حد پایین بازه (خطوط افقی) جایگزین می کنیم، تفاوت پیش بینی را محاسبه می کنیم. این تفاوت ها بعداً انباسته و متمرکز می شوند و منحنی ALE حاصل می شود.

به صورت خلاصه اینکه چگونه هر نوع نمودار (PDP، M و ALE) اثر یک ویژگی را در یک مقدار شبکه ای مشخص ۷ محاسبه می کند، در ادامه آورده شده است:

نمودارهای وابستگی جزئی: اجزا دهید به شما نشان دهم که مدل به طور متوسط وقتی نمونه های داده، مقدار ۷ برای آن ویژگی دارند، چه مقداری را پیش بینی می کند. برای من مهم نیست که آیا مقدار ویژگی ۷ برای همه نمونه ها معنادار هست یا نه.

M-Plots: اجزا دهید به شما نشان دهم که مدل به طور متوسط برای نمونه های داده ای که مقادیر نزدیک به ۷ برای آن ویژگی دارند، چه مقداری را پیش بینی می کند. این تأثیر می تواند به دلیل آن ویژگی باشد، اما ممکن هم هست به دلیل ویژگی های همبسته باشد.

نمودارهای ALE: اجازه دهید به شما نشان دهم که چگونه پیش‌بینی‌های مدل در "پنجره" کوچکی در اطراف ویژگی ۷ برای نمونه‌های داده در آن پنجره، تغییر می‌کند.

۸.۲.۲ تئوری

نمودارهای PD و ALE چه تفاوت‌هایی از نظر ریاضی دارند؟ موضوع مشترک هر سه روش این است که آنها تابع پیش‌بینی پیچیده f را به تابعی کاهش می‌دهند که فقط به یک (یا دو) ویژگی بستگی دارد. هر سه روش با میانگین‌گیری اثرات سایر ویژگی‌ها، تابع را کاهش می‌دهند، اما تفاوت سه روش در این است که میانگین‌پیش‌بینی‌ها یا تفاوت‌ها در پیش‌بینی‌ها محاسبه شود یا این که میانگین بر روی توزیع حاشیه‌ای یا شرطی انجام شود.

نمودارهای وابستگی جزئی از پیش‌بینی‌ها روی توزیع حاشیه‌ای میانگین می‌گیرند.

$$\hat{f}_{S.PDP}(x) = E_{X_C}[\hat{f}(x_S, X_C)] = \int_{X_C} \hat{f}(x_S, X_C) d\mathbb{P}(X_C)$$

این مقدار تابع پیش‌بینی f در مقدار(های) ویژگی x_S است که بر روی همه ویژگی‌ها در X_C (در اینجا به عنوان متغیرهای تصادفی در نظر گرفته می‌شود) میانگین گرفته شده است. میانگین‌گیری به معنای محاسبه امید ریاضی^۱ حاشیه‌ای E بر روی ویژگی‌های مجموعه C است، که انتگرال روی پیش‌بینی‌های وزن دار شده با توزیع احتمال است. غیر عینی به نظر می‌رسد، اما برای محاسبه مقدار امید ریاضی روی توزیع حاشیه‌ای، به سادگی تمام نمونه‌های داده را مجبور می‌کنیم که یک مقدار شبکه ای معین برای ویژگی‌های مجموعه S داشته باشند، و از پیش‌بینی‌های این مجموعه داده دستکاری شده، میانگین می‌گیریم. این فرآیند تضمین می‌کند که ما روی توزیع حاشیه‌ای ویژگی‌ها میانگین گرفته ایم.

نمودارهای M پیش‌بینی‌ها را روی توزیع شرطی میانگین می‌گیرند.

$$\hat{f}_{S.M}(x_S) = E_{X_C|X_S}[\hat{f}(X_S, X_C)|X_S = x_S] = \int_{X_C} \hat{f}(x_S, X_C) d\mathbb{P}(X_C|X_S = x_S)$$

تنها چیزی که در مقایسه با PDP‌ها تغییر می‌کند این است که به جای فرض توزیع حاشیه‌ای در هر مقدار شبکه، از پیش‌بینی‌های شرطی هر مقدار شبکه از مقدار مورد علاقه میانگین گیری می‌کنیم. در عمل، این بدان معنی است که ما باید یک همسایگی تعریف کنیم، به عنوان مثال برای محاسبه اثر ۳۰ متر مربع بر ارزش خانه پیش‌بینی شده، می‌توانیم میانگین پیش‌بینی همه خانه‌ها بین ۲۸ تا ۳۲ متر مربع را محاسبه کنیم.

نمودارهای ALE میانگین تغییرات در پیش‌بینی‌ها را محاسبه می‌کنند و آنها را در شبکه انباسته می‌کنند (در ادامه در مورد محاسبه بیشتر توضیح داده می‌شود).

¹ expectation

$$\begin{aligned}\hat{f}_{S,ALE} &= \int_{z_{0,S}}^{x_S} E_{X_C} |X_S = x_S [\hat{f}^S(X_S, X_C) | (X_S = z_S)] dz_S - constant \\ &= \int_{z_{0,S}}^{x_S} \left(\int_{x_C} \hat{f}^S(z_S, X_C) d\mathbb{P}(X_C | X_S = z_S) d \right) dz_S - constant\end{aligned}$$

این فرمول سه تفاوت را با M-Plots نشان می‌دهد. ابتدا، تغییرات پیش‌بینی‌ها را میانگین می‌گیریم، نه خود پیش‌بینی‌ها. تغییر به عنوان مشتق جزئی تعریف می‌شود (اما در ادامه، برای محاسبات واقعی، با تفاوت در پیش‌بینی‌ها روی یک بازه، جایگزین می‌شود).

$$\hat{f}^S(x_S, x_C) = \frac{\partial \hat{f}(x_S, X_C)}{\partial x_S}$$

تفاوت دوم انتگرال اضافی روی z است. ما مشتقات جزئی محلی را روی محدوده ویژگی‌های مجموعه S انباشته می‌کنیم، که به ما تأثیر ویژگی را بر پیش‌بینی می‌دهد. برای محاسبات واقعی، z ، با شبکه ای از بازه‌ها جایگزین می‌شوند که این بازه‌ها روی جایی است که در آن تغییرات پیش‌بینی را محاسبه می‌کنیم. روش ALE به جای میانگین‌گیری مستقیم پیش‌بینی‌ها، تفاوت‌های شرطی پیش‌بینی در ویژگی‌های S محاسبه می‌کند و از مشتق نسبت به ویژگی‌های S برای محاسبه اثر، انتگرال می‌گیرد. خوب، عاقلانه به به نظر نمی‌رسد. مشتق جزئی و انتگرال گیری معمولاً یکدیگر را خنثی می‌کنند، مانند ابتدا تفریق و سپس جمع کردن همان عدد. اما چرا اینجا منطقی است؟ مشتق (یا تفاوت فاصله) اثر ویژگی مورد علاقه را جدا می‌کند و اثر ویژگی‌های همبسته را مسدود می‌کند.

سومین تفاوت نمودارهای ALE با نمودارهای M این است که یک ثابت را از نتایج کم می‌کنیم. این کار باعث می‌شود تا نمودار ALE را در مرکز قرار گیرند و اثر میانگین روی داده‌ها صفر شود.

یک مشکل باقی می‌ماند: همه مدل‌ها دارای گرادیان نیستند، برای مثال جنگلهای تصادفی گرادیان ندارند. اما همان‌طور که خواهید دید، محاسبات واقعی بدون گرادیان کار می‌کند و از فواصل استفاده می‌کند. اجازه دهید کمی عمیق‌تر به برآورد نمودارهای ALE بپردازیم.

۸.۲.۳ برآورد

ابتدا توصیف خواهیم داد که چگونه نمودارهای ALE برای یک ویژگی عددی واحد، بعداً برای دو ویژگی عددی و برای یک ویژگی طبقه‌بندی واحد تخمین زده می‌شوند. برای برآورد اثرات محلی، ویژگی را به بازه‌های زیادی تقسیم می‌کنیم و تفاوت‌های پیش‌بینی‌ها را محاسبه می‌کنیم. این فرآیند مشتقات را تقریب می‌کند و همچنین برای مدل‌های بدون مشتق کار می‌کند.
ابتدا اثر متمرکزنشده را تخمین می‌زنیم:

$$\hat{f}_{j,ALE}(x) = \sum_{k=1}^{k_j(x)} \frac{1}{n_j(k)} \sum_{i:x_j^{(i)} \in N_j(k)} [\hat{f}(z_{k,j}, x_{-j}^{(i)}) - \hat{f}(z_k, x_{-j}^{(i)})]$$

بیایید از سمت راست شروع به شکستن این فرمول کنیم. نام Accumulated Local Effects به خوبی تمام عبارات این فرمول را منعکس می‌کند. در هسته خود، روش ALE تفاوت‌های پیش‌بینی‌ها را محاسبه می‌کند، که به موجب آن، ویژگی مورد نظر را با مقادیر شبکه Z جایگزین می‌کنیم. تفاوت در پیش‌بینی اثر (Effect) است که ویژگی برای یک نمونه خاص در یک بازه مشخص دارد. مجموع سمت راست اثرات تمام نمونه‌ها را در یک بازه تجمعی می‌کند که در فرمول به صورت همسایگی ($N_j(k)$) ظاهر می‌شود. ما این مجموع را بر تعداد نمونه‌های این بازه تقسیم می‌کنیم تا میانگین اختلاف پیش‌بینی‌های این بازه را به دست آوریم. این میانگین در بازه با عبارت محلی **Local** در نام ALE پوشش داده می‌شود. نماد سیگما سمت چپ به این معنی است که ما اثرات میانگین را در تمام بازه‌ها تجمعی می‌کنیم. ALE (غیر مرکزی) یک مقدار ویژگی که مثلاً در بازه سوم قرار دارد، مجموع تأثیرات بازه‌های اول، دوم و سوم است. کلمه انباسته (Accumulated) در ALE این موضوع را نشان می‌دهد.

این اثر در مرکز قرار می‌گیرد تا اثر میانگین صفر شود.

$$\hat{f}_{j,ALE}(x) = \hat{f}_{j,ALE}(x) - \frac{1}{n} \sum_{i=1}^n \hat{f}_{j,ALE}(x_j^{(i)})$$

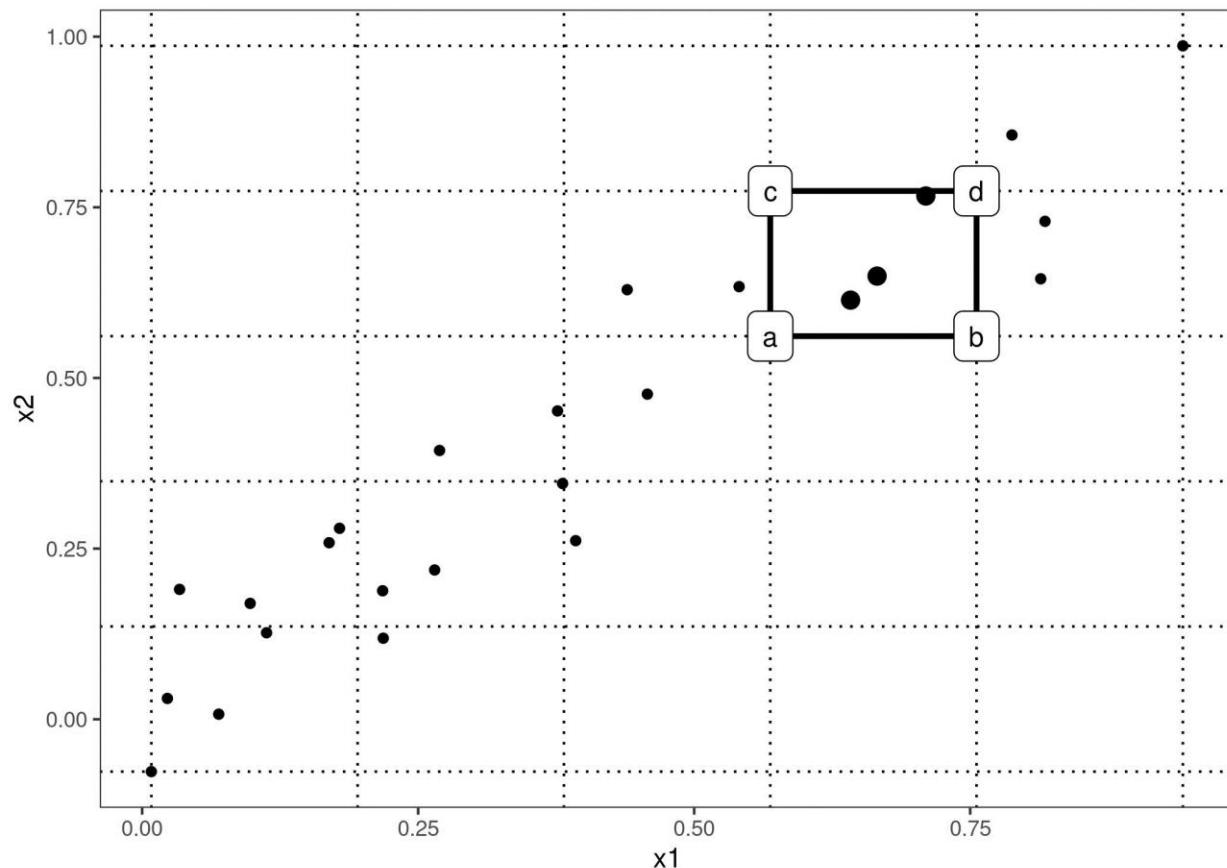
مقدار ALE را می‌توان به عنوان تأثیر اصلی ویژگی در یک مقدار مشخص در مقایسه با میانگین پیش‌بینی داده‌ها تفسیر کرد. به عنوان مثال، برآورد ALE به میزان -2 در $x_j = 3$ به این معنی است که وقتی ویژگی زداری مقدار 3 باشد، پیش‌بینی در مقایسه با پیش‌بینی متوسط 2 واحد کمتر است.

چندک‌های توزیع ویژگی به عنوان شبکه ای که فواصل را تعریف می‌کند استفاده می‌شود. استفاده از چندک‌ها تضمین می‌کند که تعداد نمونه‌های داده موجود در بازه‌ها یکسانند. چندک‌ها این عیب را دارند که بازه‌ها می‌توانند طول‌های بسیار متفاوتی داشته باشند. اگر ویژگی مورد نظر چوگی زیاد داشته باشد، به عنوان مثال تعداد زیادی مقادیر کم و تنها چند مقدار بسیار زیاد، در این حالات نمودارهای ALE عجیب می‌شوند.

نمودارهای ALE برای تعامل دو ویژگی

نمودارهای ALE همچنین می‌توانند اثر تعامل دو ویژگی را نشان دهند. اصول محاسبه مانند یک ویژگی است، اما ما به جای بازه‌ها، با سلول‌های مستطیلی کار می‌کنیم، زیرا باید اثرات را در دو بعد تجمعی کنیم. علاوه بر تنظیم برای اثر میانگین کلی، ما همچنین اثرات اصلی هر دو ویژگی را تنظیم می‌کنیم. این بدان معنی است که ALE برای دو ویژگی، اثر مرتبه دوم را تخمین می‌زند، که شامل اثرات اصلی ویژگی‌ها نمی‌شود. به عبارت دیگر، ALE برای دو ویژگی فقط اثر متقابل اضافی دو ویژگی را نشان می‌دهد. من از فرمول‌های نمودارهای ALE دو

بعدی صرف نظر می‌کنم زیرا خواندن آنها طولانی و ناخوشایند است. اگر به محاسبه علاقه دارید، شما را به مقاله، فرمول (۱۳) - (۱۶) ارجاع می‌دهم. من برای ایجاد شهود در مورد محاسبه ALE مرتبه دوم به تجسم‌ها تکیه خواهم کرد.



شکل ۸.۸: محاسبه 2D-ALE. یک شبکه روی دو ویژگی قرار می‌دهیم. در هر سلول شبکه تفاوت‌های مرتبه دوم را برای همه نمونه‌های درون آن را محاسبه می‌کنیم. ابتدا مقادیر x_1 و x_2 را با مقادیر گوش سلول جایگزین می‌کنیم. اگر a ، b ، c و d پیش‌بینی‌های "گوش" یک نمونه دستکاری شده را نشان دهند (همان‌طور که در نمودار نشان داده شده است)، تفاوت مرتبه دوم $(b - a) - (c - d)$ است. میانگین اختلاف مرتبه دوم در هر سلول روی شبکه تجمعی شده و در مرکز قرار می‌گیرد.

در شکل قبل، بسیاری از سلول‌ها به دلیل همبستگی خالی هستند. در نمودار ALE این را می‌توان با یک کادر خاکستری یا تیره به تصویر کشید. به عنوان یک روش دیگر می‌توانید تخمین ALE در سلول‌های خالی را با تخمین ALE نزدیک ترین سلول غیرخالی جایگزین کنید.

از آنجایی که تخمین‌های ALE برای دو ویژگی فقط اثر مرتبه دوم ویژگی‌ها را نشان می‌دهد، تفسیر نیاز به توجه ویژه دارد. اثر مرتبه دوم، اثر تعامل اضافی ویژگی‌ها است، پس از اینکه اثرات اصلی ویژگی‌ها را در نظر

گرفتیم. فرض کنید دو ویژگی با هم تعامل ندارند، اما هر کدام یک اثر خطی بر خروجی پیش‌بینی شده دارند. در نمودار یک بعدی ALE برای هر ویژگی، ما یک خط را به عنوان منحنی ALE تخمین زده می‌بینیم. اما وقتی تخمین‌های ALE دو بعدی را رسم می‌کنیم، باید نزدیک به صفر باشند، زیرا اثر مرتبه دوم فقط اثر اضافی تعامل است. نمودارهای ALE و نمودارهای PD از این نظر متفاوت هستند: PDP‌ها همیشه اثر کل را نشان می‌دهند، نمودارهای ALE اثر مرتبه اول یا دوم را نشان می‌دهند. اینها تصمیمات طراحی هستند که به ریاضیات پایه آن‌ها بستگی ندارند. می‌توانید تاثیرات مرتبه پایین‌تر را از نمودار وابستگی جزئی کم کنید تا اثرات خالص یا مرتبه دوم را به دست آورید یا می‌توانید با خودداری از تفیریق اثرات مرتبه پایین، تخمینی از کل نمودارهای ALE دریافت کنید.

اثرات محلی انباشته را می‌توان برای مرتبه‌های بالاتر دلخواه (تعامل سه یا چند ویژگی) نیز محاسبه کرد، اما همان‌طور که در فصل PDP بحث شد، فقط محاسبات تا دو ویژگی منطقی است، زیرا تعاملات بالاتر را نمی‌توان تجسم کرد یا حتی به صورت معنی‌دار تفسیر کرد.

ALE برای ویژگی‌های طبقه‌بندی شده

روش اثرات محلی انباشته - طبق تعریف - به مقادیر ویژگی نیاز دارد تا ترتیب داشته باشد، زیرا این روش اثرات را در جهت خاصی تجمعی می‌کند. ویژگی‌های طبقه‌بندی هیچ ترتیب طبیعی ندارند. برای محاسبه نمودار ALE برای یک ویژگی طبقه‌بندی، باید به نحوی یک ترتیب ایجاد یا پیدا کنیم. ترتیب طبقه‌ها بر محاسبه و تفسیر اثرات محلی انباشته تأثیر می‌گذارد.

یک راه حل این است که طبقه‌ها را بر اساس شباهت آنها بر اساس سایر ویژگی‌ها مرتب کنید. فاصله بین دو طبقه مجموع فواصل هر ویژگی است. فاصله از جنبه ویژگی، یا توزیع تجمعی در هر دو طبقه، که به آن فاصله کولموگروف-اسمیرنوف^۱ (برای ویژگی‌های عددی) می‌گویند را مقایسه می‌کند یا جداول فراوانی نسبی (برای ویژگی‌های طبقه‌بندی). هنگامی که فواصل بین همه طبقه‌ها را بدست آوردیم، از مقیاس بندی چند بعدی برای کاهش ماتریس فاصله به یک اندازه فاصله یک بعدی استفاده می‌کنیم. این کار یک ترتیب مبتنی بر شباهت طبقه‌ها ایجاد می‌کند.

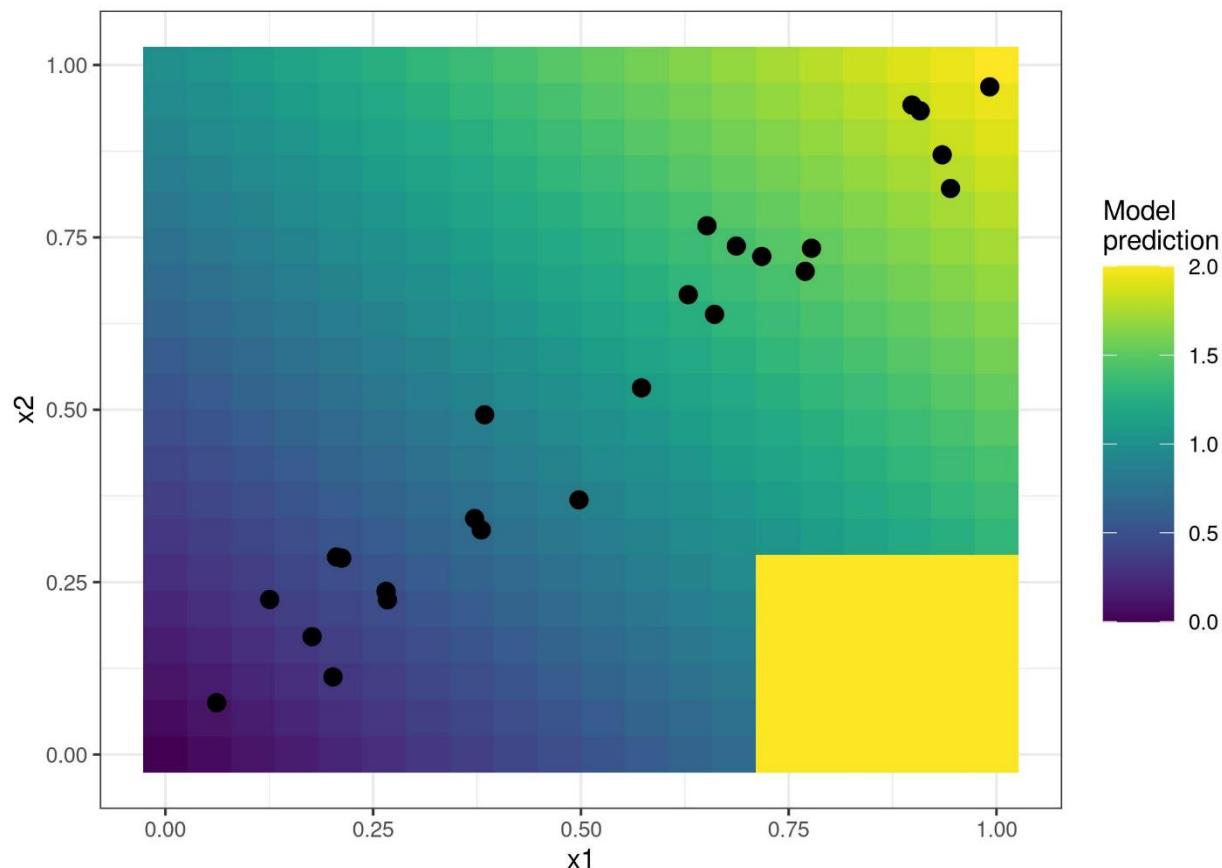
برای اینکه این موضوع کمی‌ واضح‌تر شود، در اینجا یک مثال آورده شده است: فرض کنید دو ویژگی طبقه‌بندی "فصل" و "آب و هوا" و یک ویژگی عددی "دما" را داریم. برای اولین ویژگی طبقه بندی (فصل) می‌خواهیم ALE‌ها را محاسبه کنیم. این ویژگی دارای دسته‌های «بهار»، «تابستان»، «پاییز»، «زمستان» است. ما شروع به محاسبه فاصله بین دسته‌های "بهار" و "تابستان" می‌کنیم. فاصله مجموع فواصل ویژگی‌های دما و آب و هوا است. برای دما، همه نمونه‌ها را با فصل «بهار» می‌گیریم،تابع توزیع تجمعی عملی را محاسبه می‌کنیم و همین

¹ Kolmogorov-Smirnov

کار را برای نمونه‌هایی با فصل «تابستان» انجام می‌دهیم و فاصله آنها را با آماره کولموگروف-اسمیرنوف اندازه‌گیری می‌کنیم. برای ویژگی آب و هوا، ما برای همه موارد "بهار" احتمالات را برای هر نوع آب و هوا محاسبه می‌کنیم. همین کار را برای نمونه‌های "تابستان" انجام دهید و فواصل مطلق را در توزیع احتمال تجمعی کنید. اگر «بهار» و «تابستان» دما و آب و هوای بسیار متفاوتی داشته باشند، فاصله طبقه کل بزرگ می‌شود. ما این روش را با سایر جفت‌های فصلی تکرار می‌کنیم و ماتریس فاصله حاصل را با مقیاس بندی چند بعدی به یک بعد کاهش می‌دهیم.

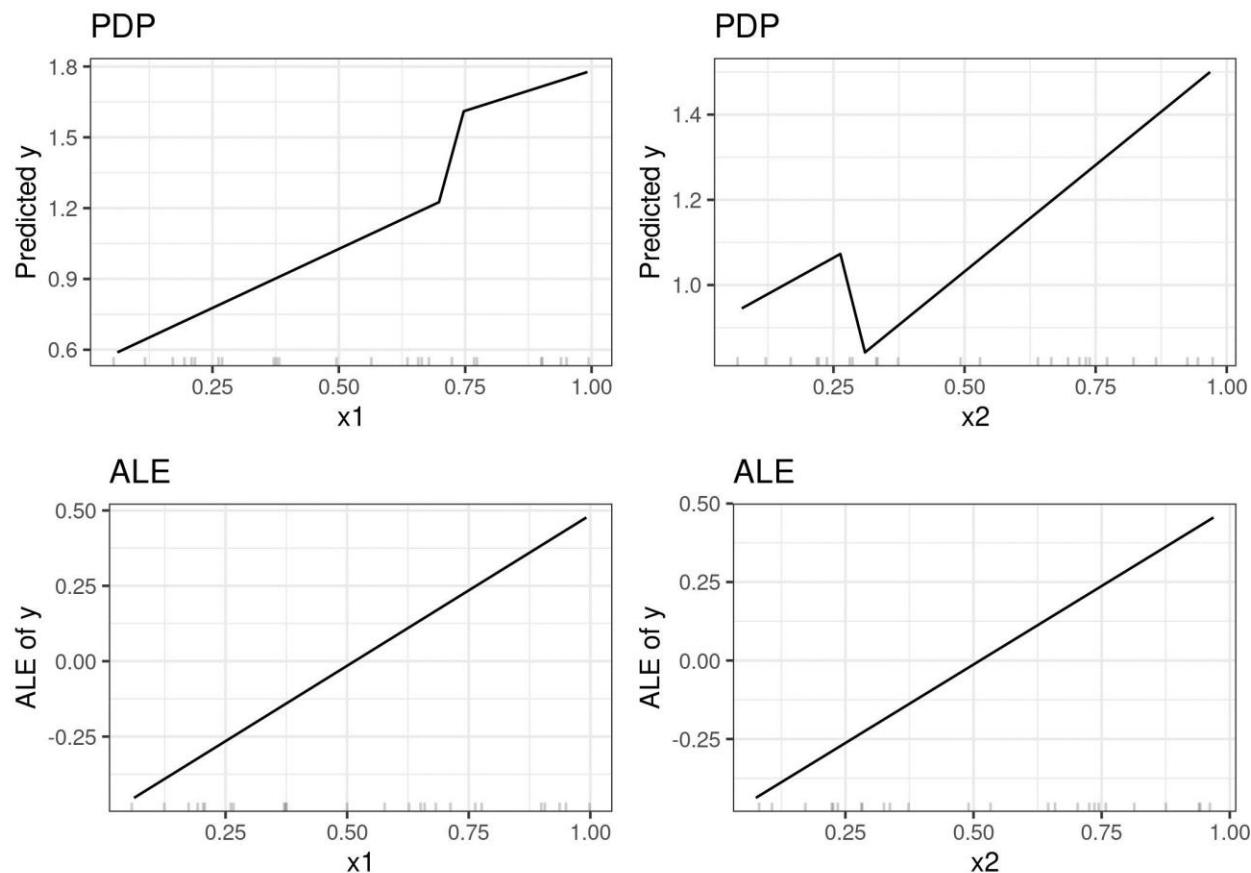
۸.۲.۴ مثال‌ها

اجازه دهید نمودارهای ALE را در عمل ببینیم. من سناریویی ساخته ام که در آن نمودارهای وابستگی جزئی شکست می‌خورند. این سناریو از یک مدل پیش‌بینی و دو ویژگی شدیداً همبسته تشکیل شده است. مدل پیش‌بینی اساساً یک مدل رگرسیون خطی است، اما با ترکیبی از دو ویژگی که ما هرگز نمونه‌هایی را برای آن مشاهده نکرده‌ایم، کار عجیبی انجام می‌دهد.



شکل ۸.۹: دو ویژگی و خروجی پیش‌بینی شده. مدل مجموع دو ویژگی (پس زمینه سایه دار) را پیش‌بینی می‌کند، با این تفاوت که اگر x_1 بزرگ‌تر از $0/7$ و x_2 کمتر از $0/3$ باشد، مدل همیشه ۲ را پیش‌بینی می‌کند. این ناحیه از توزیع داده‌ها (نقاط) فاصله زیادی دارد و بر عملکرد مدل و همچنین تفسیر آن تأثیری ندارد.

آیا اصلاً این یک سناریوی واقعی و مرتبط است؟ هنگامی که یک مدل را آموزش می‌دهید، الگوریتم یادگیری خطای نمونه‌های داده‌های آموزشی موجود را به حداقل می‌رساند. موارد عجیب می‌تواند خارج از توزیع داده‌های آموزشی اتفاق بیفتد، زیرا مدل برای انجام کارهای عجیب و غریب در این زمینه‌ها جریمه نمی‌شود. خروج از توزیع داده برون یابی^۱ نامیده می‌شود، که می‌تواند برای فریب دادن مدل‌های یادگیری ماشین نیز استفاده شود (مثال‌های مختصّ). در مثال کوچک ما ببینید نمودارهای وابستگی جزئی در مقایسه با نمودارهای ALE چگونه رفتار می‌کنند.



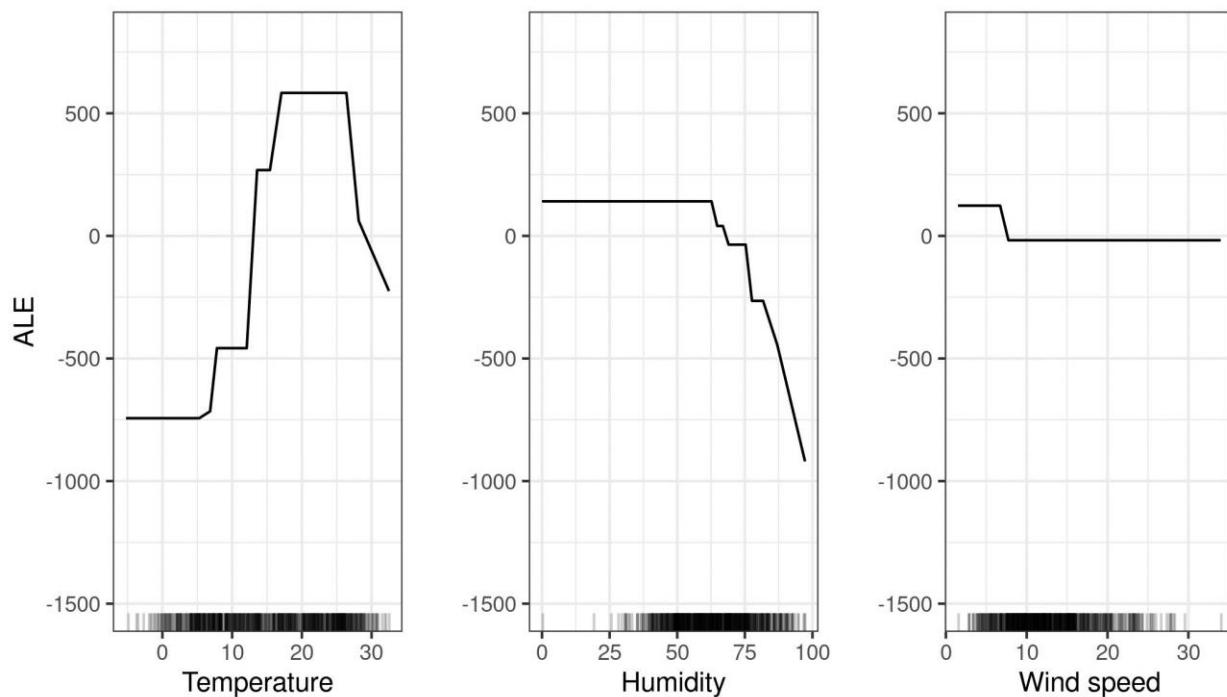
شکل ۸.۱۰: مقایسه اثرات ویژگی محاسبه شده با PDP (ردیف بالا) و ALE (ردیف پایین). تخمین‌های تحت تاثیر رفتار عجیب مدل در خارج از توزیع داده‌ها (پرش‌های تند در نمودارها) قرار دارند. نمودارهای ALE به

^۱ extrapolation

درستی مشخص می کنند که مدل یادگیری ماشین رابطه ای خطی بین ویژگی ها و پیش بینی دارد و مناطق بدون داده را نادیده می گیرد.

اما آیا جالب نیست که ببینیم مدل ما در $x_1 < 0.7$ و $x_2 < 0.3$ رفتار عجیبی دارد؟ خوب، بله و نه. از آنجایی که اینها نمونه های هستند که ممکن است از نظر فیزیکی غیرممکن یا حداقل بسیار بعيد باشند، معمولاً بررسی این نمونه ها بی معنا می باشد. اما اگر مشکوک هستید که توزیع تست شما ممکن است کمی متفاوت باشد و برخی از نمونه های واقعی در آن محدوده هستند، خوب است که این ناحیه را در محاسبه اثرات ویژگی ها لحاظ کنید. اما این تصمیم باید یک تصمیم آگاهانه باشد که آیا اینجا مناطقی هستند که هنوز داده ای را مشاهده نکرده ایم و نه یک تصمیم ناآگاهانه مثلایک اثر سو در اثر انتخابی مانند PDP. اگر مشکوک هستید که مدل بعداً با توزیع داده های متفاوت استفاده می شود، توصیه می کنم از نمودارهای ALE استفاده کنید و توزیع داده هایی را که انتظار دارید، شبیه سازی کنید.

با چرخش به سمت یک مجموعه داده واقعی، اجازه دهید تعداد دوچرخه های اجاره ای را بر اساس آب و هوا و روز پیش بینی کنیم و بررسی کنیم که آیا نقشه های ALE واقعاً به همان اندازه که وعده داده شده کار می کنند یا خیر. ما یک درخت رگرسیون را برای پیش بینی تعداد دوچرخه های اجاره ای در یک روز معین آموخته می دهیم و از نمودارهای ALE برای تجزیه و تحلیل چگونگی تأثیر دما، رطوبت نسبی و سرعت باد بر پیش بینی ها استفاده می کنیم. ببینیم نمودارهای ALE چه می گویند:



شکل ۸.۱۱: نمودارهای ALE برای مدل پیش‌بینی دوچرخه بر اساس دما، رطوبت و سرعت باد. دما تأثیر زیادی در پیش‌بینی دارد. میانگین پیش‌بینی با افزایش دما افزایش می‌یابد، اما وقتی به بالای ۲۵ درجه سانتیگراد می‌رسد، کاهش می‌یابد. رطوبت اثر منفی دارد: وقتی بالای ۶۰ درصد باشد، هر چه رطوبت نسبی بیشتر باشد، پیش‌بینی کمتر است. سرعت باد روی پیش‌بینی‌ها تاثیر زیادی ندارد.

اجازه دهدید به همبستگی بین دما، رطوبت و سرعت باد و سایر ویژگی‌ها نگاه کنیم. از آنجایی که داده‌ها دارای ویژگی‌های طبقه‌بندی هستند، ما نمی‌توانیم از ضریب همبستگی پیرسون^۱ استفاده کنیم که فقط در صورتی کار می‌کند که هر دو ویژگی عددی باشند. در عوض، من یک مدل خطی را آموزش می‌دهم تا مثلاً دما را بر اساس یکی از ویژگی‌های دیگر به عنوان ورودی پیش‌بینی کند. سپس اندازه می‌گیرم چه مقدار از واریانس توسط ویژگی موجود در مدل خطی توضیح داده می‌شود و جذر این مقدار را می‌گیرم. اگر ویژگی درون مدل، عددی بود، نتیجه برابر با قدر مطلق ضریب همبستگی پیرسون استاندارد است. اما این رویکرد مبتنی بر مدل "واریانس توضیح داده شده"^۲ (که همچنین ANAlysis Of VAriance که مخفف ANOVA نیز نامیده می‌شود) را حتی اگر ویژگی از نوع طبقه‌ای نیز باشد، حساب می‌کند. اندازه گیری "واریانس توضیح داده شده" همیشه بین ۰ (بدون ارتباط) و ۱ (دما را می‌توان کاملاً از ویژگی دیگر پیش‌بینی کرد) قرار دارد. ما واریانس توضیح داده شده دما، رطوبت و سرعت باد را با تمام ویژگی‌های دیگر محاسبه می‌کنیم. هر چه واریانس توضیح داده شده (همبستگی) بیشتر باشد، مشکلات (بالقوه) بیشتری در نمودارهای PD وجود دارد. شکل زیر نشان می‌دهد که چقدر ویژگی‌های آب و هوا با سایر ویژگی‌ها همبستگی دارند.

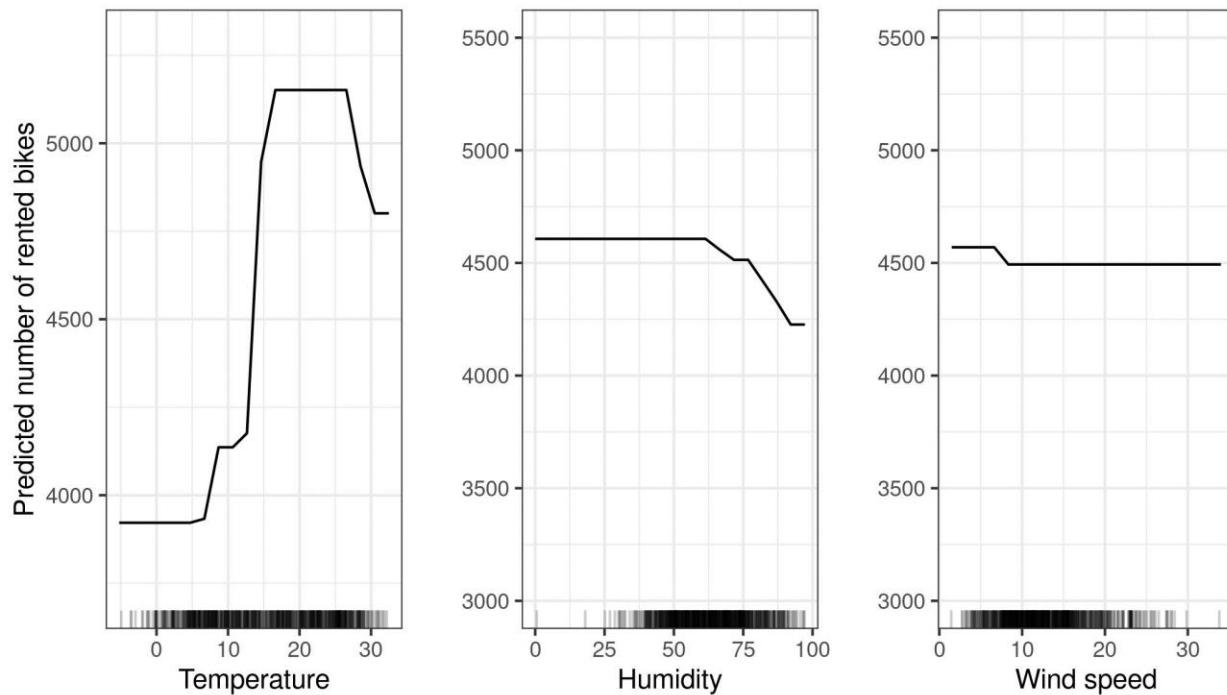
¹ Pearson correlation coefficient

² variance-explained



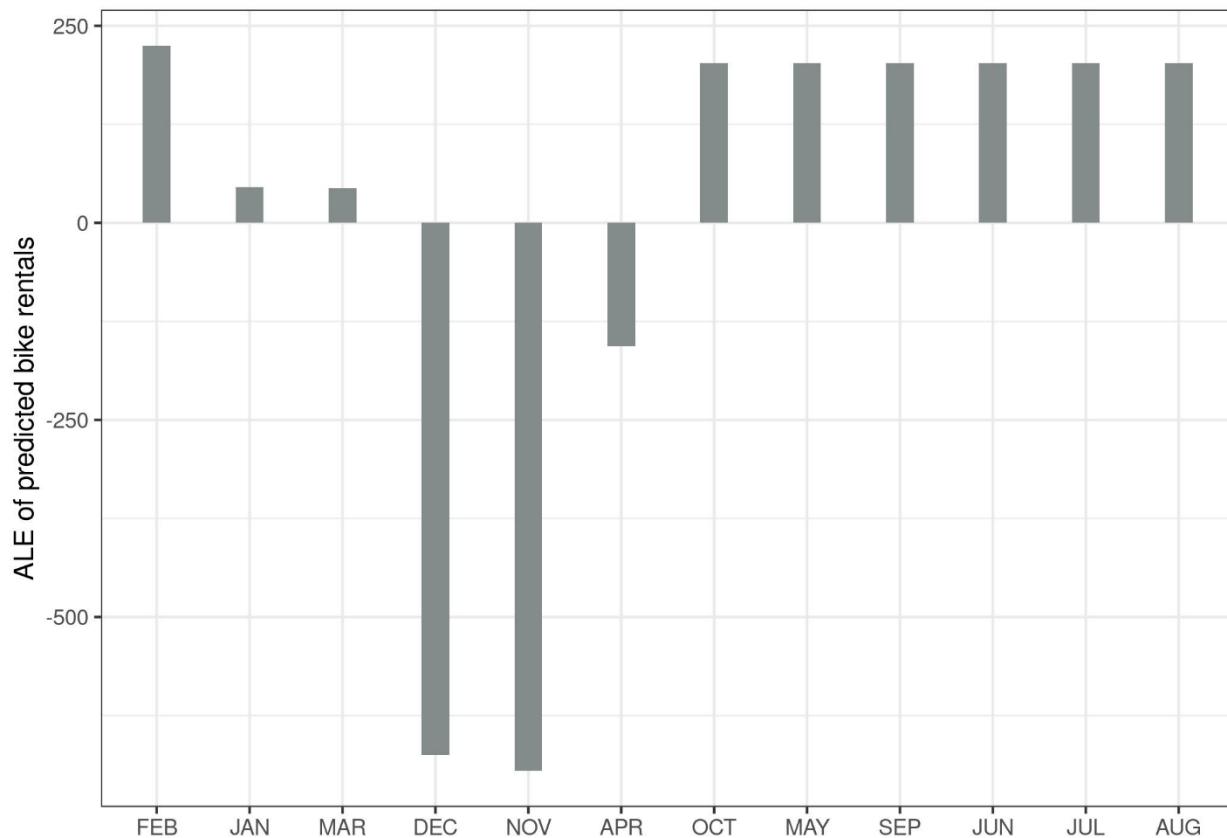
شکل ۸.۱۲: قدرت همبستگی بین دما، رطوبت و سرعت باد با همه ویژگی‌ها، اندازه گیری شده به عنوان مقدار واریانس توضیح داده شده، زمانی که ما یک مدل خطی را به عنوان مثال با دما برای پیش‌بینی ویژگی فصل آموزش می‌دهیم. برای دما، به صورتی که انتظار داریم - همبستگی بالایی با فصل و ماه مشاهده می‌کنیم. رطوبت با وضعیت آب و هوا ارتباط دارد.

این تجزیه و تحلیل همبستگی نشان می‌دهد که ما ممکن است در نمودارهای وابستگی جزئی با مشکلاتی مواجه شویم، به ویژه برای ویژگی دما. مشاهده کنید:



شکل ۸.۱۳: PDP‌ها برای دما، رطوبت و سرعت باد. در مقایسه با نمودارهای ALE، کاهش کمتری در تعداد پیش‌بینی شده دوچرخه‌ها برای دمای بالا یا رطوبت بالا یا نشان می‌دهند. PDP از تمام نمونه‌های داده برای محاسبه اثر دماهای بالا استفاده می‌کند، حتی اگر مثلاً نمونه‌هایی با فصل «زمستان» باشند. نمودارهای ALE قابل اعتمادتر هستند.

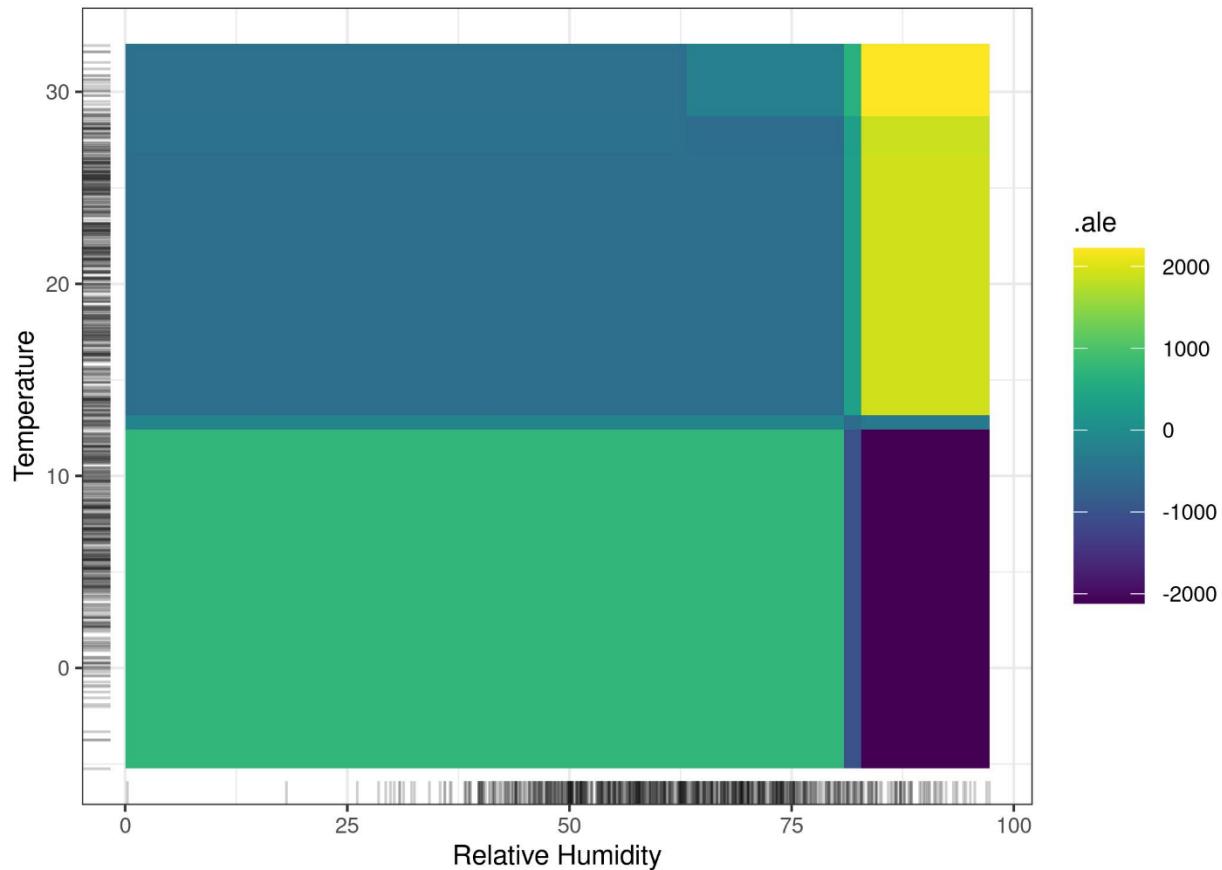
در مرحله بعد، اجازه دهید نمودارهای ALE را در عمل برای یک ویژگی طبقه‌بندی ببینیم. ماه یک ویژگی طبقه‌بندی است که می‌خواهیم تأثیر آن را بر تعداد پیش‌بینی شده دوچرخه‌ها تجزیه و تحلیل کنیم. مسلماً، ماهها از قبل نظم خاصی دارند (زانویه تا دسامبر)، اما اجازه دهید ببینیم اگر ابتدا دسته‌ها را بر اساس شباهت دوباره ترتیب دهیم و سپس اثرات را محاسبه کنیم، چه اتفاقی می‌افتد. ماهها بر اساس شباهت روزهای هر ماه بر اساس ویژگی‌های دیگر مانند دما یا تعطیلی آن مرتب می‌شوند.



شکل ۸.۱۴: نمودار ALE برای ویژگی طبقه‌بندی ماه. ماهها بر اساس شباهت آنها به یکدیگر، بر اساس توزیع ویژگی‌های دیگر غیر از خود ماه، مرتب می‌شوند. مشاهده می‌کنیم که ژانویه، مارس و آوریل، به ویژه دسامبر و نوامبر، در مقایسه با ماههای دیگر اثر کمتری بر تعداد دوچرخه‌های اجاره‌ای پیش‌بینی شده دارند.

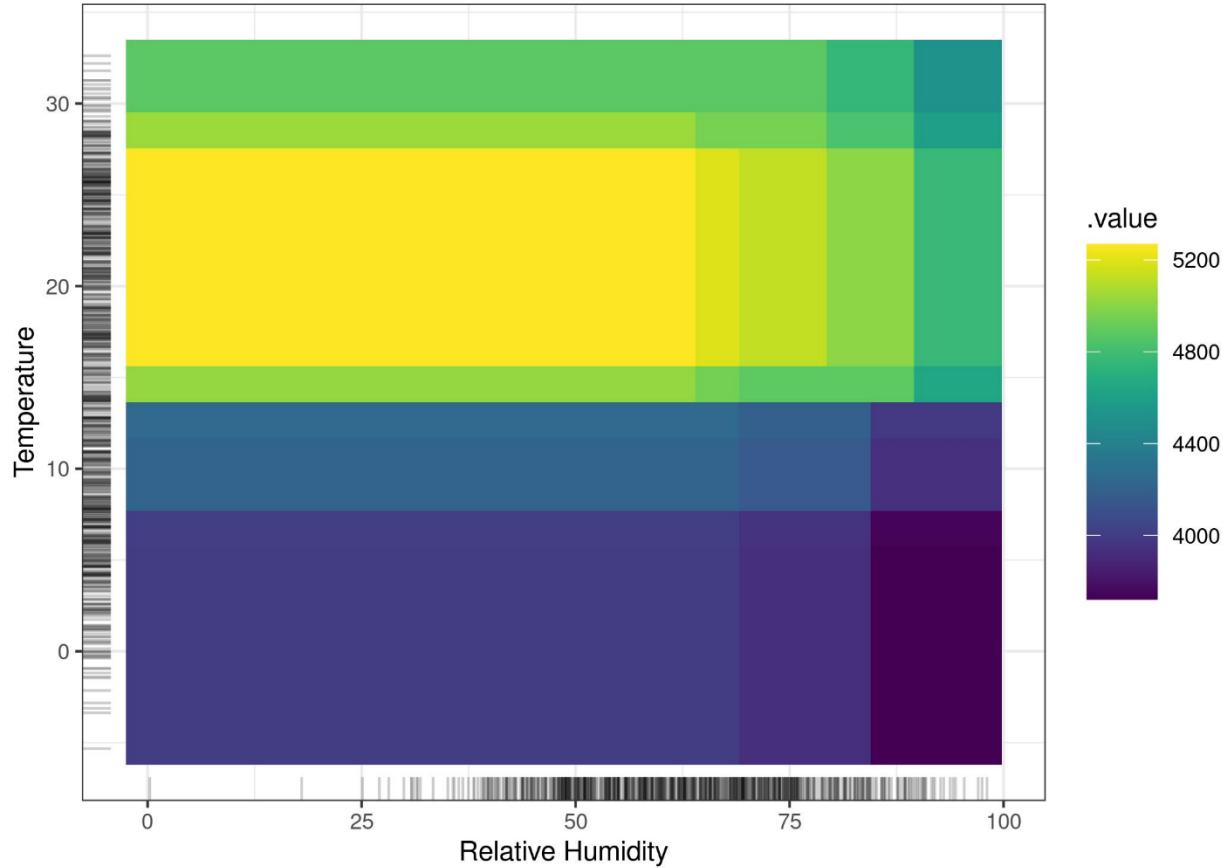
از آنجایی که بسیاری از ویژگی‌ها مربوط به آب و هوا هستند، ترتیب ماهها به شدت نشان دهنده شباهت آب و هوا بین ماهها است. تمام ماههای سردتر در سمت چپ (فوریه تا آوریل) و ماههای گرم‌تر در سمت راست (اکتبر تا آگوست) قرار دارند. به خاطر داشته باشید که ویژگی‌های غیر آب و هوای نیز در محاسبه شباهت لحاظ شده است، به عنوان مثال فراوانی نسبی تعطیلات وزنی برابر با دما برای محاسبه شباهت بین ماهها دارد.

در مرحله بعد، تأثیر درجه دوم رطوبت و دما را بر تعداد پیش‌بینی شده دوچرخه‌ها در نظر می‌گیریم. به یاد داشته باشید که اثرات مرتبه دوم اثر متقابل اضافی دو ویژگی است و اثرات اصلی را شامل نمی‌شود. این بدان معنی است که، برای مثال، شما اثر اصلی که رطوبت بالا منجر به تعداد کمتری از دوچرخه‌های پیش‌بینی شده به طور متوسط می‌شود را در نمودار ALE درجه دوم نخواهید دید.



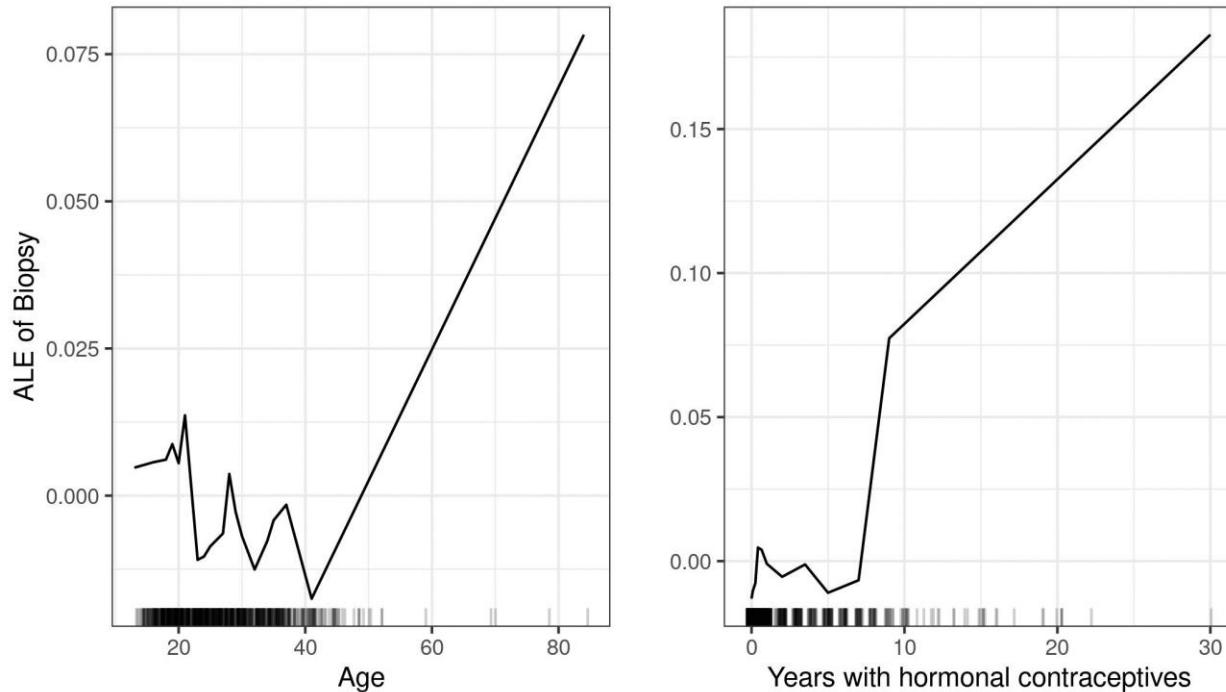
شکل ۸.۱۵: نمودار ALE برای اثر مرتبه دوم رطوبت و دما بر تعداد پیش‌بینی شده دوچرخه‌های اجاره‌ای. سایه‌ی روشن‌تر نشان‌دهنده یک پیش‌بینی بالاتر از حد متوسط و سایه‌ی تیره‌تر یک پیش‌بینی کمتر از میانگین است، زمانی اثرات اصلی قبلًا در نظر گرفته شده‌اند. این نمودار یک تعامل بین دما و رطوبت را نشان می‌دهد: هوای گرم و مرطوب پیش‌بینی را افزایش می‌دهد. در هوای سرد و مرطوب یک اثر منفی اضافی بر تعداد دوچرخه‌های پیش‌بینی شده نشان داده است.

به خاطر داشته باشید که هر دو اثر اصلی رطوبت و دما می‌گویند که تعداد پیش‌بینی شده دوچرخه‌ها در هوای بسیار گرم و مرطوب کاهش می‌یابد. بنابراین در هوای گرم و مرطوب، اثر ترکیبی دما و رطوبت مجموع اثرات اصلی نیست، بلکه بزرگ‌تر از مجموع آن است. برای تأکید بر تفاوت بین اثر مرتبه دوم خالص (نقشه دوبعدی ALE که همین الان دیدید) و اثر کل، اجازه دهید به طرح وابستگی جزئی نگاه کنیم. PDP اثر کل را نشان می‌دهد که پیش‌بینی میانگین، دو اثر اصلی و اثر مرتبه دوم (تعامل) را ترکیب می‌کند.

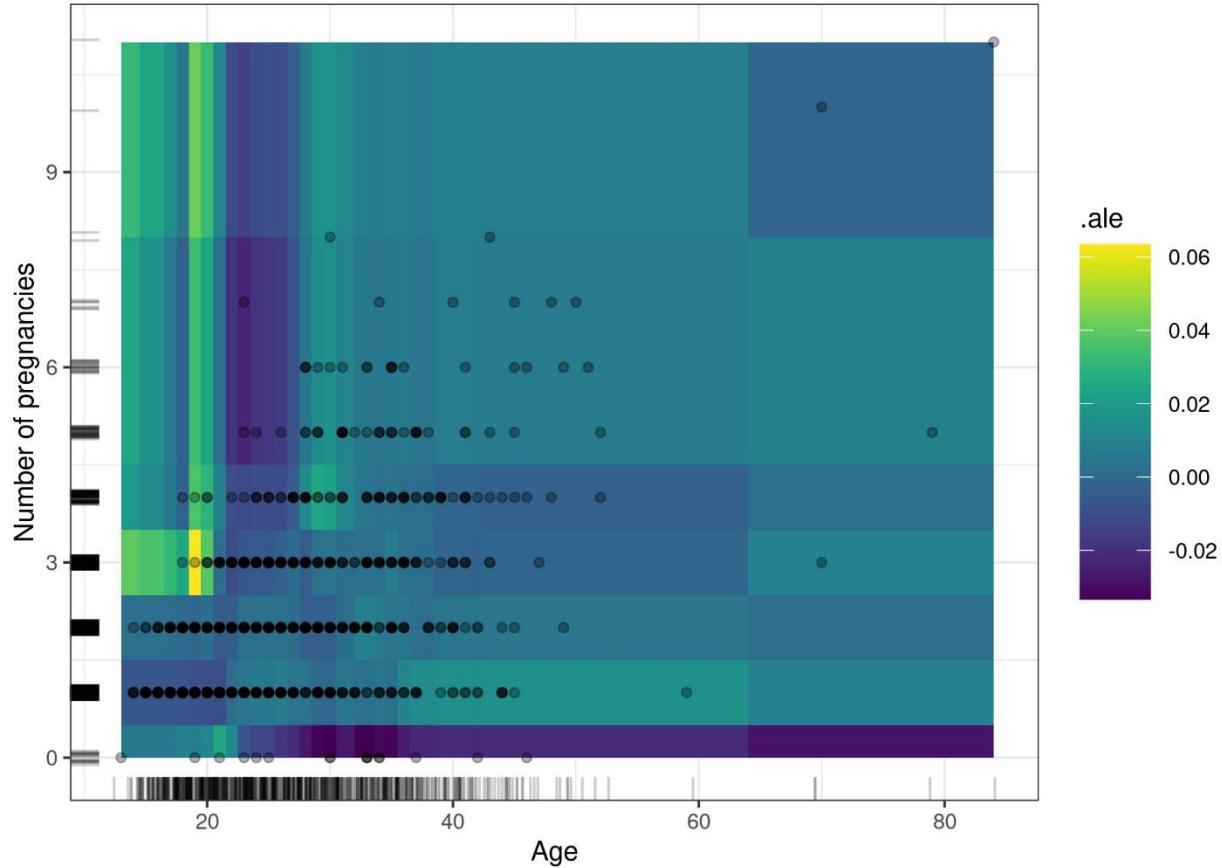


شکل ۸.۱۶: PDP اثر کل دما و رطوبت بر تعداد پیش‌بینی شده دوچرخه‌ها. نمودار اثر اصلی هر یک از ویژگی‌ها و اثر تعاملات آنها را ترکیب می‌کند، برخلاف طرح دو بعدی ALE که فقط تعامل را نشان می‌دهد. اگر فقط به تعامل علاقه دارید، باید به اثرات درجه دوم نگاه کنید، زیرا اثر کل، اثرات اصلی را در نمودار ترکیب می‌کند. اما اگر می‌خواهید اثر ترکیبی ویژگی‌ها را بدانید، باید به اثر کل (که PDP نشان می‌دهد) نگاه کنید. برای مثال، اگر می‌خواهید تعداد دوچرخه‌های مورد انتظار در دمای ۳۰ درجه سانتی‌گراد و رطوبت ۸۰ درصد را بدانید، می‌توانید آن را مستقیماً از PDP دو بعدی بخوانید. اگر می‌خواهید همان را از نمودارهای ALE بخوانید، باید به سه نمودار نگاه کنید: نمودار ALE برای دما، نمودار ALE برای رطوبت و نمودار ALE برای دما + رطوبت و همچنین باید پیش‌بینی میانگین کلی را بدانید. در سناریویی که دو ویژگی هیچ تعاملی با هم ندارند، نمودار اثر کل دو ویژگی می‌تواند گمراه‌کننده باشد، زیرا احتمالاً یک چشم‌انداز پیچیده را نشان می‌دهد، که نشان‌دهنده مقداری تعامل است. اما صرفاً محصول دو اثر اصلی است. اثر مرتبه دوم بلافاصله نشان می‌دهد که هیچ تعاملی وجود ندارد.

در حال حاضر دوچرخه کافی است، اجازه دهد به یک کار طبقه‌بندی بپردازیم. ما یک جنگل تصادفی برای پیش‌بینی احتمال سرطان دهانه رحم بر اساس عوامل خطر آموزش می‌دهیم. ما اثرات محلی انباسته را برای دو ویژگی تجسم می‌کنیم:



شکل ۸.۱۷: نمودارهای ALE اثر سن و سالهای استفاده از داروهای ضد بارداری هورمونی بر احتمال پیش‌بینی شده سرطان دهانه رحم. برای ویژگی سن، نمودار ALE نشان می‌دهد که احتمال سرطان پیش‌بینی شده به طور متوسط تا سن ۴۰ سالگی کم است و پس از آن افزایش می‌یابد. تعداد سالهای استفاده از داروهای ضد بارداری هورمونی با افزایش خطر سرطان پیش‌بینی شده بعد از ۸ سال مرتبط است. در ادامه، به تعامل بین تعداد بارداری و سن نگاه می‌کنیم.



شکل ۸.۱۸: نمودار ALE اثر مرتبه دوم تعداد بارداری و سن. تفسیر طرح کمی غیرقطعی است و به نظر می‌رسد بیش برآش رخ داده است. به عنوان مثال، نمودار مدل یک رفتار عجیب را در سنین ۱۸ تا ۲۰ سالگی و بیش از ۳ بارداری نشان می‌دهد (تا ۵ درصد افزایش در احتمال سلطان). تعداد زیادی زن در داده‌ها با این سن و این تعداد حاملگی وجود ندارد (داده‌های واقعی به صورت نقاط نمایش داده شده‌اند)، بنابراین مدل در طول آموزش به دلیل اشتباہش برای آن زنان جریمه جدی نمی‌شود.

۸.۲.۵ مزايا

نمودارهای ALE بی‌طرفانه هستند، به این معنی که وقتی ویژگی‌ها همبستگی دارند، همچنان کار می‌کنند. نمودارهای وابستگی جزئی در این سناریو شکست می‌خورند، زیرا ترکیب‌های بعيد یا حتی از لحاظ فیزیکی غیرممکن از مقادیر ویژگی را تجمعیع می‌کنند.

نمودارهای ALE سریع‌تر از PDP‌ها محاسبه می‌شوند و با $O(n)$ مقیاس می‌شوند، زیرا بیشترین تعداد بازه‌های ممکن برابر با تعداد نمونه‌هاست یعنی یک نمونه در هر بازه PDP به تخمین n برابر تعداد نقاط شبکه نیاز دارد. برای ۲۰ نقطه شبکه، PDP‌ها ۲۰ برابر بیشتر از بدینانه‌ترین نمودار ALE (در حالتی که در هر بازه فقط یک نمونه قرار دارد) به پیش‌بینی، نیاز دارند.

تفسیر نمودارهای ALE واضح است: مشروط بر یک مقدار معین، اثر نسبی تغییر ویژگی در پیش‌بینی را می‌توان از نمودار ALE خواند. نمودارهای ALE در مرکز متتمرکز شده اند. این باعث می‌شود تفسیر آنها خوب باشد، زیرا مقدار در هر نقطه از منحنی ALE تفاوت از پیش‌بینی میانگین است. نمودار دو بعدی ALE فقط تعامل را نشان می‌دهد: اگر دو ویژگی با هم تعامل نداشته باشند، نمودار هیچ چیزی را نشان نمی‌دهد. کل تابع پیش‌بینی را می‌توان به مجموع توابع ALE با ابعاد پایین‌تر تجزیه کرد، همان‌طور که در فصل تجزیه تابعی توضیح داده می‌شود.

در مجموع، در بیشتر موقعیت‌ها، نمودارهای ALE را به PDP ترجیح می‌دهم، زیرا ویژگی‌ها معمولاً تا حدی با هم همبستگی دارند.

۸.۲.۶ معایب

اگر ویژگی‌ها به شدت همبستگی داشته باشند، تفسیر اثر در فواصل مجاز نیست. موردی را در نظر بگیرید که در آن ویژگی‌های شما بسیار همبسته هستند، و شما به انتهای سمت چپ یک نمودار یک بعدی ALE نگاه می‌کنید. منحنی ALE ممکن است باعث تعبیر نادرست زیر شود: «منحنی ALE نشان می‌دهد که چگونه پیش‌بینی به‌طور متوسط، زمانی که به تدریج مقدار ویژگی مربوطه را برای یک نمونه داده تغییر می‌دهیم، و مقادیر دیگر ویژگی‌ها را ثابت نگه می‌داریم، تغییر می‌کند». اثرات در هر بازه (محلی) محاسبه می‌شوند و بنابراین تفسیر اثر فقط می‌تواند محلی باشد. برای راحتی، اثرات با مبنای بازه‌ای برای نشان دادن یک منحنی نرم انباسته می‌شوند، اما به خاطر داشته باشید که هر بازه با نمونه‌های داده متفاوتی ایجاد می‌شود.

اثرات ALE ممکن است با ضرایب مشخص شده در یک مدل رگرسیون خطی متفاوت باشد، زمانی که ویژگی‌ها با هم تعامل و همبستگی دارند. Grömping (2020) نشان داد که در یک مدل خطی با دو ویژگی همبسته و یک عبارت تعامل اضافی $\hat{f}(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$ نمودارهای ALE مرتبه اول یک خط مستقیم را نشان نمی‌دهند. در عوض، آنها کمی خمیده هستند زیرا بخش‌هایی از تعامل ضربی ویژگی‌ها را در خود جای می‌دهند. برای درک آنچه در اینجا اتفاق می‌افتد، توصیه می‌کنم فصل تجزیه تابع را بخوانید. به طور خلاصه، ALE اثرات مرتبه اول (یک بعدی) را متفاوت با فرمول خطی تعریف می‌کند. این لزوماً اشتباه نیست، زیرا وقتی ویژگی‌ها همبستگی دارند، نسبت دادن تعاملات به آن روشن نیست. اما مطمئناً غیر قابل درک است که چرا ALE و ضریب خطی مطابقت ندارند.

نمودارهای ALE می‌توانند کمی متزلزل¹ شوند (تعداد زیادی فراز و نشیب‌های کوچک) در صورت استفاده از تعداد بازه‌های زیاد. در این مورد، کاهش تعداد بازه‌ها، تخمین‌ها را پایدارتر می‌کند، اما برخی از پیچیدگی

¹ shaky

واقعی مدل پیش‌بینی را نرم و پنهان می‌کند. هیچ راه حل قطعی برای تنظیم تعداد فواصل وجود ندارد. اگر تعداد خیلی کم باشد، نمودارهای ALE ممکن است خیلی دقیق نباشند. اگر عدد خیلی زیاد باشد، منحنی می‌تواند متزلزل شود.

برخلاف PDP‌ها، نمودارهای ALE با منحنی‌های ICE همراه نیستند. برای PDP‌ها، منحنی‌های ICE عالی هستند زیرا می‌توانند ناهمگنی^۱ را در اثر ویژگی نشان دهند. ناهمگنی به این معنی که اثر یک ویژگی برای زیر مجموعه‌های داده، متفاوت می‌باشد. برای نمودارهای ALE فقط می‌توانید در هر بازه بررسی کنید که آیا اثر بین نمونه‌ها متفاوت است یا خیر، اما هر بازه دارای نمونه‌های متفاوتی است و با منحنی‌های ICE یکسان نیست. برآوردهای مرتبه دوم ALE دارای ثبات متفاوتی در فضای ویژگی هستند که به هیچ روشی تجسم نمی‌شود. دلیل این است که برآورد از یک اثر محلی در یک سلول از تعداد متفاوتی از نمونه‌ها استفاده می‌کند. در نتیجه، تخمین‌ها، دقت‌های متفاوتی دارند (اما هنوز بهترین تخمین‌های ممکن هستند). مشکل در یک نسخه از نمودارهای اثر اصلی ALE وجود دارد. به لطف استفاده از چندک‌ها برای شبکه بندی، تعداد نمونه‌ها در همه فواصل یکسان است، اما در برخی مناطق فواصل کوتاه زیادی پدید می‌آید و منحنی ALE از تخمین‌های بسیار بیشتری تشکیل می‌شود. اما برای بازه‌های طولانی، که می‌تواند بخش بزرگی از کل منحنی را تشکیل دهد، نمونه‌های نسبتاً کمتری وجود دارد. این اتفاق در نمودار ALE پیش‌بینی سلطان دهانه رحم برای سن بالا رخ داد.

تفسیر نمودارهای اثر درجه دوم می‌تواند کمی آزاردهنده باشد، زیرا همیشه باید اثرات اصلی را در ذهن داشته باشید. خواندن نقشه‌های حرارتی به عنوان اثر کلی دو ویژگی گمراх است، زیرا این نموار فقط شامل اثر اضافی تعامل است. اثر مرتبه دوم خالص برای کشف و کاوش تعاملات، مفید است، اما برای تفسیر این که یک اثر چگونه است، فکر می‌کنم ادغام اثرات اصلی در نمودار منطقی تر است.

پیاده‌سازی نمودارهای ALE در مقایسه با نمودارهای وابستگی جزئی بسیار پیچیده‌تر و کمتر شهودی است.

اگرچه نمودارهای ALE در مورد ویژگی‌های همبسته جانبدارانه نیستند، وقتی ویژگی‌ها به شدت همبستگی داشته باشند، تفسیر باز هم مشکل باقی می‌ماند. زیرا اگر همبستگی بسیار قوی باشد، تنها تحلیل اثر تغییر هر دو ویژگی با هم و نه به صورت مجزا، منطقی است. این نقطه ضعف مختص نمودارهای ALE نیست، بلکه یک مشکل کلی در ویژگی‌های با همبستگی قوی می‌باشد.

¹ heterogeneity

اگر ویژگی‌ها همبستگی ندارند و مشکل با زمان محاسبه وجود ندارد، PDP‌ها کمی ترجیح داده می‌شوند زیرا درک آنها آسان‌تر است و می‌توان آن‌ها را همراه با منحنی‌های ICE رسم کرد. لیست معایب بسیار طولانی شد، اما فریب تعداد کلماتی را که من به کار می‌برم نخورید: به عنوان یک قانون سرراست: به جای PDP از ALE استفاده کنید.

۸.۲.۷ پیاده‌سازی و جایگزین‌ها

قبل اشاره کردم که نمودارهای وابستگی جزئی و منحنی‌های انتظار شرطی فردی جایگزین یکدیگر هستند. نمودارهای ALE در R در پکیج ALEPlot¹ توسط خود پدیدآورنده و در پکیج iml² پیاده‌سازی شده است. ALE همچنین دارای حداقل دو پیاده‌سازی Python با پکیج ALEPython³ و Alibi⁴ دارد.

¹ <https://cran.r-project.org/web/packages/ALEPlot/index.html>

² <https://cran.r-project.org/web/packages/iml/index.html>

³ <https://github.com/blent-ai/ALEPython>

⁴ <https://docs.seldon.io/projects/alibi/en/stable/index.html>

۸.۳ تعامل ویژگی‌ها

هنگامی که ویژگی‌ها در یک مدل پیش‌بینی با یکدیگر تعامل دارند، پیش‌بینی را نمی‌توان به عنوان مجموع اثرات ویژگی‌ها بیان کرد، زیرا تأثیر یک ویژگی به مقدار ویژگی دیگر بستگی دارد. گفته ارسسطو «کل بزرگ‌تر از مجموع اجزای آن است» در حضور فعل و افعالات برقرار است.

۸.۳.۱ تعامل ویژگی؟

اگر یک مدل یادگیری ماشین بر اساس دو ویژگی پیش‌بینی کند، می‌توانیم پیش‌بینی را به چهار عبارت تجزیه کنیم: یک عبارت ثابت، یک عبارت برای ویژگی اول، یک عبارت برای ویژگی دوم و یک عبارت برای تعامل بین دو ویژگی.

تعامل بین دو ویژگی، تغییر در پیش‌بینی است که با تغییر ویژگی‌ها پس از در نظر گرفتن اثرات ویژگی‌های فردی رخ می‌دهد.

به عنوان مثال، یک مدل ارزش یک خانه را با استفاده از اندازه خانه (بزرگ یا کوچک) و مکان (خوب یا بد) به عنوان ویژگی‌ها پیش‌بینی می‌کند که چهار پیش‌بینی ممکن را ایجاد می‌کند:

Location	Size	Prediction
good	big	300,000
good	small	200,000
bad	big	250,000
bad	small	150,000

ما پیش‌بینی مدل را به بخش‌های زیر تجزیه می‌کنیم: یک جمله ثابت (۱۵۰۰۰۰)، یک اثر برای ویژگی اندازه (۱۰۰۰۰۰ + اگر بزرگ، +۰ اگر کوچک) و یک اثر برای مکان (۵۰۰۰۰ + اگر خوب، +۰ اگر بد است). این تجزیه به طور کامل پیش‌بینی‌های مدل را توضیح می‌دهد. هیچ اثر متقابلي وجود ندارد، زیرا پیش‌بینی مدل مجموع اثرات تک ویژگی برای اندازه و مکان است. وقتی یک خانه کوچک را بزرگ می‌کنید، بدون توجه به موقعیت مکانی، پیش‌بینی همیشه ۱۰۰۰۰۰ افزایش می‌یابد. همچنین تفاوت پیش‌بینی موقعیت مکانی خوب و بد بدون توجه به اندازه ۵۰۰۰۰ است.

بیایید اکنون به یک مثال با تعامل نگاه کنیم:

Location	Size	Prediction
good	big	400,000
good	small	200,000
bad	big	250,000
bad	small	150,000

جدول پیش‌بینی را به بخش‌های زیر تجزیه می‌کنیم: یک جمله ثابت (۱۵۰۰۰۰)، یک اثر برای ویژگی اندازه (۱۰۰۰۰۰ + اگر بزرگ، +۰ اگر کوچک) و یک اثر برای مکان (۵۰۰۰۰ + اگر خوب، +۰ اگر بد است). برای این

جدول ما به یک عبارت اضافی برای تعامل نیاز داریم: $+100000$ اگر خانه بزرگ و در موقعیت خوبی باشد. این یک تعامل بین اندازه و مکان است، زیرا در این مورد تفاوت در پیش‌بینی بین یک خانه بزرگ و یک خانه کوچک به مکان بستگی دارد.

یکی از راههای تخمین قدرت تعامل این است که اندازه‌گیری شود که چقدر از تغییرات پیش‌بینی به تعامل ویژگی‌ها بستگی دارد. این کار، اندازه گیری آماره H نامیده می‌شود که توسط Friedman and Popescu (2008) معرفی شده است.

۸.۳.۲ تئوری: آماره H فربدمان^۱

قصد داریم به دو مورد بپردازیم: اول، یک معیار تعامل دو طرفه که به ما می‌گوید آیا و تا چه حد دو ویژگی در مدل با یکدیگر تعامل دارند یا خیر. دوم، یک معیار تعامل کلی که به ما می‌گوید آیا و تا چه حد یک ویژگی در مدل با همه ویژگی‌های دیگر تعامل دارد یا خیر. در تئوری، تعاملات دلخواه بین هر تعداد ویژگی قابل اندازه گیری است، اما این دو مورد، جالب‌ترین موارد هستند.

اگر دو ویژگی با هم تعامل نداشته باشند، می‌توانیم تابع وابستگی جزئی را به صورت زیر تجزیه کنیم (با فرض اینکه توابع وابستگی جزئی در صفر متمرکز هستند):

$$PD_{jk}(x_j \cdot x_k) = PD_j(x_j) + PD_k(x_k)$$

در این رابطه $PD_{jk}(x_j \cdot x_k)$ تابع وابستگی جزئی دو طرفه هر دو ویژگی و $(x_j)PD_j(x_k)$ و $(x_k)PD_k(x_k)$ توابع وابستگی جزئی ویژگی‌های منفرد.

به همین ترتیب، اگر یک ویژگی با هیچ یک از ویژگی‌های دیگر تعامل نداشته باشد، می‌توانیم تابع پیش‌بینی $\hat{f}(x)$ را به عنوان مجموع تابع وابستگی جزئی، که در آن جمع اول فقط به j و دومی به تمام ویژگی‌های دیگر به جز زبستگی دارد، بیان کنیم:

$$\hat{f}(x) = PD_j(x_j) + PD_{-j}(x_{-j})$$

در اینجا $PD_{-j}(x_{-j})$ تابع وابستگی جزئی است که به همه ویژگی‌ها به جز ویژگی- j از بستگی دارد. این تجزیه، تابع وابستگی جزئی (یا پیش‌بینی کامل) را بدون تعامل (به ترتیب بین ویژگی‌های j و k یا j و همه ویژگی‌های دیگر) بیان می‌کند. در مرحله بعد، تفاوت بین تابع وابستگی جزئی مشاهده شده و تابع تجزیه شده را بدون تعامل اندازه گیری می‌کنیم. ما واریانس خروجی وابستگی جزئی (برای اندازه گیری تعامل بین دو ویژگی) یا کل تابع (برای اندازه گیری تعامل بین یک ویژگی و همه ویژگی‌های دیگر) را محاسبه می‌کنیم. مقدار واریانس توضیح داده شده توسط تعامل (تفاوت بین PD مشاهده شده و بدون تعامل) به عنوان آماره قدرت تعامل استفاده می‌شود. در صورتی که هیچ تعاملی وجود نداشته باشد، آماره $\hat{f}(x)$ با مجموع PD_{jk} یا $PD_j(x_j) + PD_k(x_k)$ برابر خواهد بود.

¹ Friedman's H-statistic

توابع وابستگی جزئی توضیح داده شود، آماره ۱ بین دو ویژگی به این معنی است که هر دوتابع PD ثابت است و تأثیر بر پیش‌بینی فقط از طریق تعامل حاصل می‌شود. آماره H همچنین می‌تواند بزرگتر از ۱ باشد که تفسیر آن دشوارتر است. این حالت می‌تواند زمانی اتفاق بیفتد که واریانس تعامل دو طرفه بزرگتر از واریانس نمودار وابستگی جزئی دو بعدی باشد.

از نظر ریاضی، آماره H ارائه شده توسط فریدمن و پوپسکو برای تعامل بین ویژگی زو k به صورت زیر است:

$$H_{jk}^2 = \frac{\sum_{i=1}^n \left[PD_{jk}(x_j^{(i)} \cdot x_k^{(i)}) - PD_j(x_j^{(i)}) - PD_k(x_k^{(i)}) \right]^2}{\sum_{i=1}^n PD_{jk}^2(x_j^{(i)} \cdot x_k^{(i)})}$$

همین امر در مورد اندازه گیری اینکه آیا یک ویژگی زبا هر ویژگی دیگری تعامل دارد یا خیر، صدق می‌کند:

$$H_j^2 = \frac{\sum_{i=1}^n \left[\hat{f}(x_j^{(i)}) - PD_j(x_j^{(i)}) - PD_{-j}(x_{-j}^{(i)}) \right]^2}{\sum_{i=1}^n \hat{f}^2(x_j^{(i)})}$$

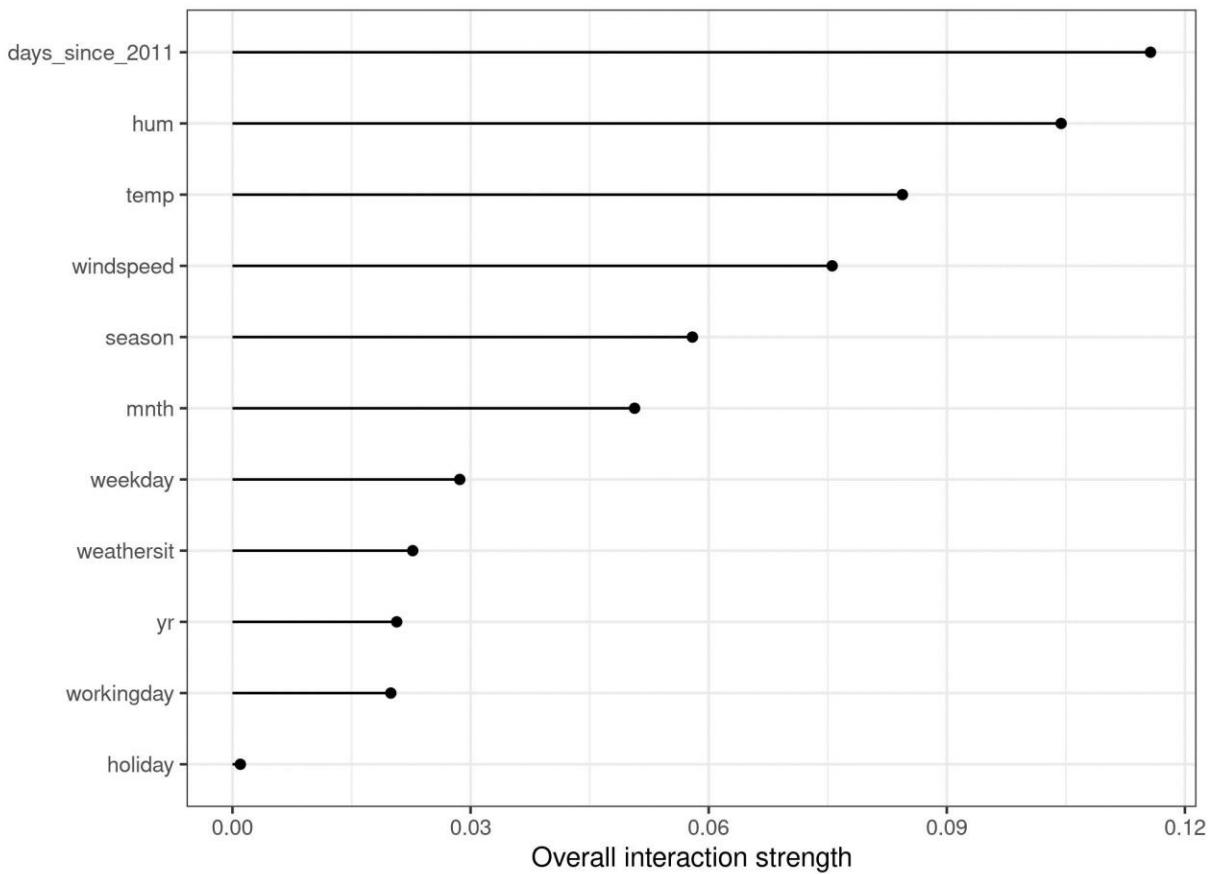
محاسبه آماره H سنگین است، زیرا در تمام نقاط داده تکرار می‌شود و در هر نقطه باید وابستگی جزئی ارزیابی شود که به نوبه خود با تمام n نقطه داده انجام می‌شود. در بدترین حالت، برای محاسبه آماره H دو طرفه (j دربرابر k) به $2n^2$ فراخوانی مدل‌های یادگیری ماشین پیش‌بینی تابع و $3n^2$ برای آماره H کل (j دربرابر همه نیاز داریم. برای سرعت بخشیدن به محاسبات، می‌توانیم از n نقطه داده نمونه برداری کنیم. این کار، نقطه ضعف افزایش واریانس تخمین‌های وابستگی جزئی را دارد که باعث می‌شود آماره H ناپایدار باشد. بنابراین اگر از نمونه برداری برای کاهش بار محاسباتی استفاده می‌کنید، مطمئن شوید که از نقاط داده به اندازه کافی نمونه برداری کرده‌اید.

فریدمن و پوپسکو همچنین یک آمار آزمایشی برای ارزیابی اینکه آیا آماره H به طور قابل توجهی با صفر متفاوت است پیشنهاد می‌کنند. فرضیه صفر عدم وجود تعامل است. برای ایجاد آمار تعامل تحت فرضیه صفر، باید بتوانید مدل را طوری تنظیم کنید که هیچ تعاملی بین ویژگی زو k یا همه موارد دیگر نداشته باشد. این امکان برای همه مدل‌ها وجود ندارد. بنابراین این آزمون خاص مدل است، نه آگنوسنیک مدل، و به این ترتیب در اینجا پوشش داده نشده است.

اگر پیش‌بینی یک احتمال باشد، آماره قدرت تعامل می‌تواند در یک مساله طبقه‌بندی نیز اعمال شود.

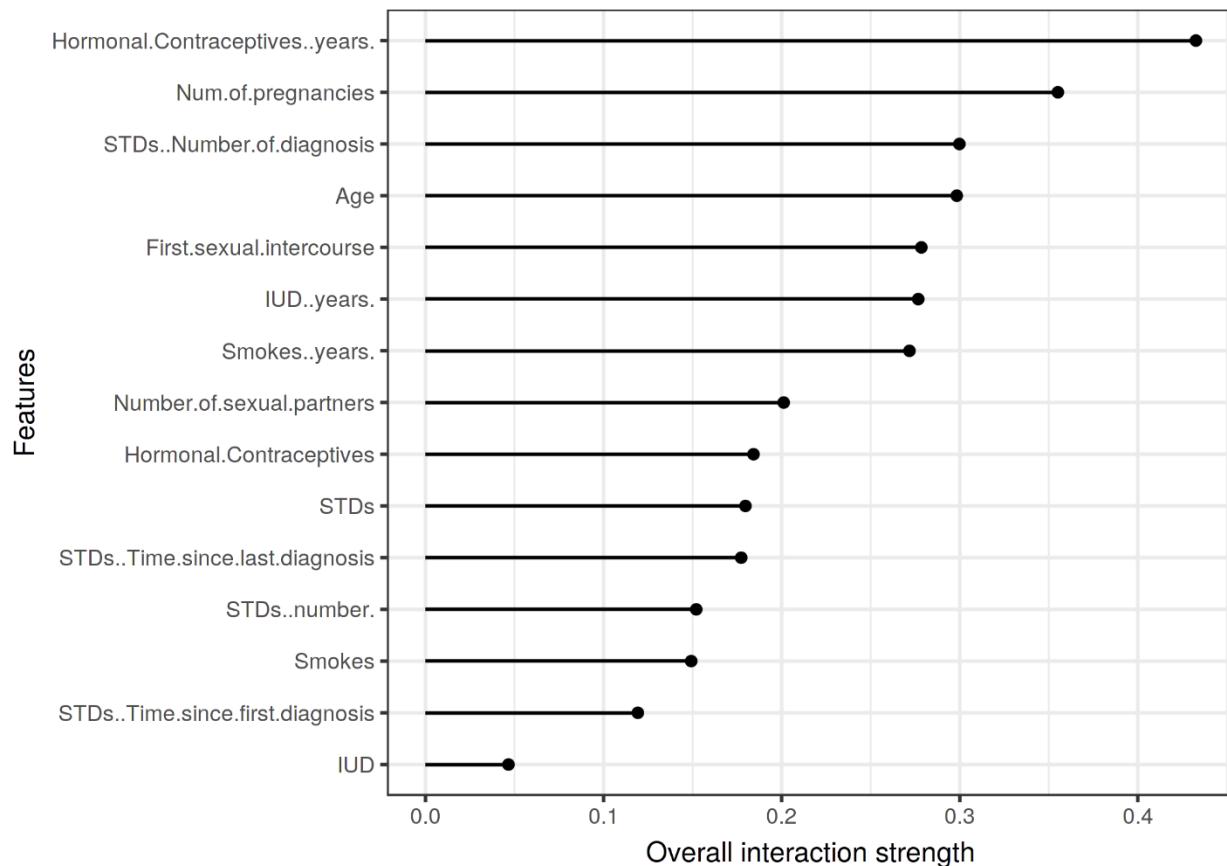
۸.۳.۳ مثال‌ها

بیایید بینیم تعاملات ویژگی در عمل چگونه است! ما قدرت تعامل ویژگی‌ها را در یک ماشین بردار پشتیبان اندازه‌گیری می‌کنیم که تعداد دوچرخه‌های اجاره‌ای را بر اساس ویژگی‌های آب و هوا و تقویم پیش‌بینی می‌کند. نمودار زیر آمار تعامل ویژگی H را نشان می‌دهد:

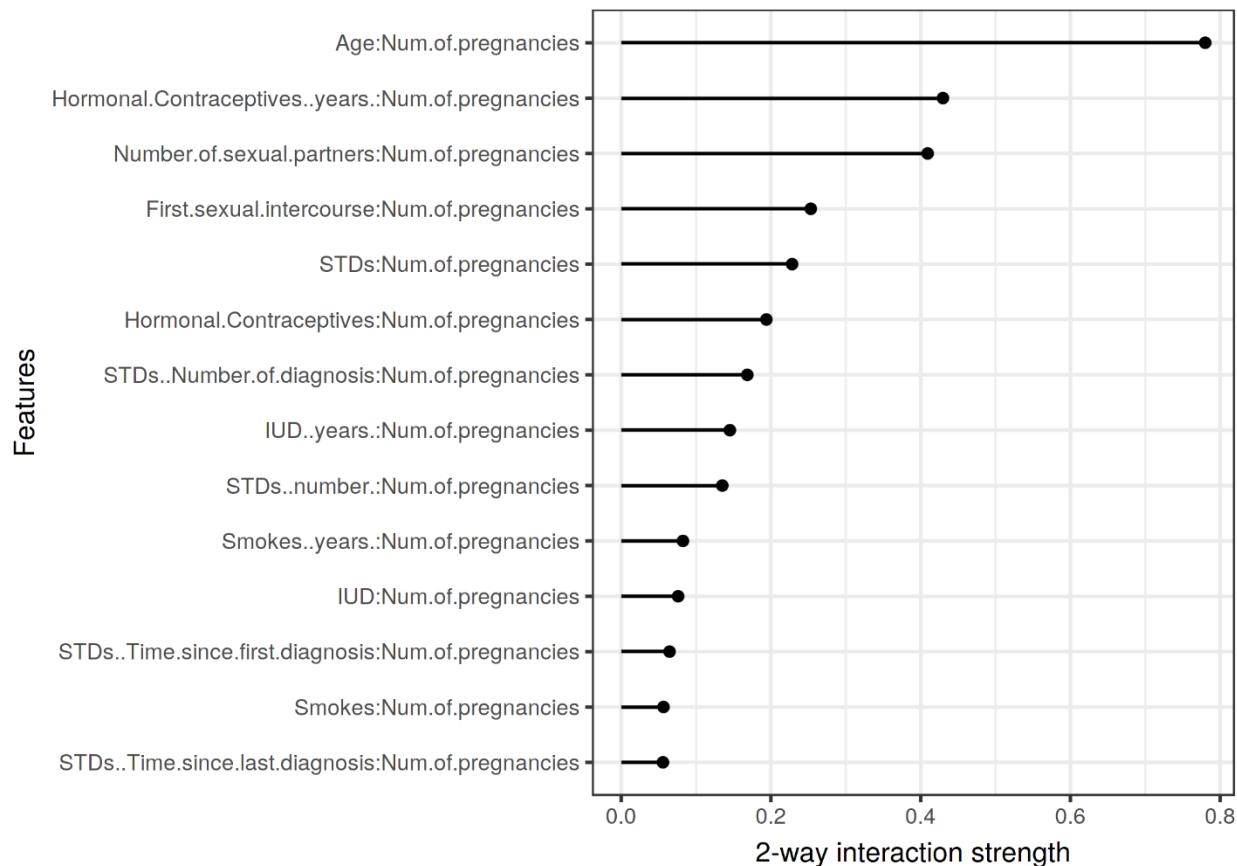


شکل ۸.۱۹: قدرت تعامل (آماره H) برای هر ویژگی با تمام ویژگی‌های دیگر برای ماشین بردار پشتیبان که اجاره دورچرخه را پیش‌بینی می‌کند. به طور کلی، اثرات متقابل بین ویژگی‌ها بسیار ضعیف است (زیر ۱۰ درصد واریانس توضیح داده شده در هر ویژگی).

در مثال بعدی، ما آماره تعامل را برای یک مسئله طبقه‌بندی محاسبه می‌کنیم. ما تعاملات بین ویژگی‌ها را در یک جنگل تصادفی که برای پیش‌بینی سلطان دهانه رحم آموخته دیده است، با توجه به برخی عوامل خطر تجزیه و تحلیل می‌کنیم.



شکل ۸.۲۰: قدرت تعامل (آماره H) برای هر ویژگی با تمام ویژگی‌های دیگر برای یک جنگل تصادفی که احتمال سرطان دهانه رحم را پیش‌بینی می‌کند. سالهای استفاده از داروهای ضدبارداری هورمونی بالاترین اثر مقابل نسبی را با سایر ویژگی‌ها دارد و به دنبال آن تعداد حاملگی‌ها قرار دارد. پس از بررسی تعاملات هر ویژگی با سایر ویژگی‌ها، می‌توانیم یکی از ویژگی‌ها را انتخاب کنیم و عمیق‌تر در تمام تعاملات دو طرفه بین ویژگی انتخاب شده و سایر ویژگی‌ها غوطه ور شویم.



شکل ۸.۲۱: قدرت تعامل دو طرفه (آماره H) بین تعداد حاملگی‌ها و ویژگی‌های دیگر. یک تعامل قوی بین تعداد بارداری و سن وجود دارد.

۸.۳.۴ مزایا

آماره تعامل H دارای یک تئوری اساسی از طریق تجزیه وابستگی جزئی است.

آماره H تعبیر معناداری دارد: تعامل به عنوان سهم واریانسی که توسط تعامل توضیح داده می‌شود، تعریف می‌شود.

از آنجایی که آماره بدون بعد است، در بین ویژگی‌ها و حتی در بین مدل‌ها قابل مقایسه است.

این آماره انواع تعاملات را بدون توجه به فرم خاص آنها تشخیص می‌دهد.

با آماره H همچنین امکان تحلیل تعاملات دلخواه بالاتر مانند قدرت برهمکنش بین ۳ یا چند ویژگی وجود دارد.

۸.۳.۵ معایب

اولین چیزی که متوجه خواهد شد: محاسبه آماره H تعامل زمان زیادی می‌برد، زیرا از نظر محاسباتی گران است.

محاسبات شامل تخمین توزیع‌های حاشیه ای است. اگر از تمام نقاط داده استفاده نکنیم، این تخمین‌ها دارای واریانس خاصی هستند. این بدان معناست که با نمونه‌برداری از نقاط، تخمین‌ها نیز از اجرا به اجرا متفاوت است و نتایج می‌توانند ناپایدار باشند. من توصیه می‌کنم محاسبه آماره H را چند بار تکرار کنید تا ببینید آیا داده‌های کافی برای به دست آوردن یک نتیجه پایدار دارید یا خیر.

مشخص نیست که آیا یک تعامل به طور قابل توجهی بیشتر از α است یا خیر. ما باید یک آزمایش آماری انجام دهیم، اما این تست (هنوز) در یک نسخه آگنوستیک مدل موجود نیست.

در مورد مسئله تست، دشوار است که بگوییم چه زمانی آماره H به اندازه کافی بزرگ است که بتوانیم یک تعامل را «قوی» در نظر بگیریم.

همچنین، آماره H می‌تواند بزرگ‌تر از 1 باشد که تفسیر را دشوار می‌کند.

زمانی که اثر کلی دو ویژگی ضعیف باشد، اما بیشتر از تعاملات تشکیل شده باشد، آماره H بسیار بزرگ خواهد بود. این تعاملات کاذب به مخرج کوچکی از آماره H نیاز دارند و زمانی که ویژگی‌ها همبستگی دارند بدتر می‌شوند. یک تعامل جعلی را به اشتباه می‌توان یک اثر تعامل قوی، تفسیر کرد، در حالی که در واقعیت هر دو ویژگی نقش کوچکی در مدل دارند. یک راه حل ممکن این است که نسخه نرمال نشده آماره H را تجسم کنید، که جذر صورت کسر آماره H است (Inglis et al., 2022). این کار آماره H را به همان سطح پاسخ، حداقل برای رگرسیون، مقیاس می‌کند و تأکید کمتری بر تعاملات جعلی دارد.

$$H_{jk}^* = \sqrt{\sum_{i=1}^n \left[PD_{jk} \left(x_j^{(i)} \cdot x_k^{(i)} \right) - PD_j \left(x_j^{(i)} \right) - PD_k \left(x_k^{(i)} \right) \right]^2}$$

آماره H قدرت تعاملات را به ما می‌گوید، اما به ما نمی‌گوید که تعاملات چگونه هستند. این همان چیزی است که نمودارهای وابستگی جزئی برای آن هستند. یک گردش کار مناسب این است که قدرت تعامل را اندازه گیری کنید و سپس برای تعاملاتی که به آنها علاقه دارید، نمودارهای وابستگی جزئی دو بعدی ایجاد کنید.

اگر ورودی‌ها پیکسل باشند، آماره H نمی‌تواند به طور معناداری استفاده شود. بنابراین این تکنیک برای طبقه‌بندی کننده تصویر مفید نیست.

آمار تعامل با این فرض کار می‌کند که ما می‌توانیم ویژگی‌ها را به طور مستقل به هم بزنیم. اگر ویژگی‌ها به شدت همبستگی داشته باشند، این فرض نقض می‌شود و ترکیب‌های ویژگی‌هایی را که در واقعیت بسیار

بعید هستند، ادغام می‌کنیم. این همان مشکلی است که نمودارهای وابستگی جزئی دارند. ویژگی‌های همبسته می‌توانند به مقادیر زیادی از آماره H منجر شوند.

گاهی اوقات نتایج عجیب هستند و برای شبیه سازی‌های کوچک نتایج مورد انتظار را به همراه نمی‌آورند. اما این بیشتر یک تجربه شخصی است.

۸.۳.۶ پیاده سازی‌ها

برای مثال‌های این کتاب، از پکیج iml برای نرم افزار R استفاده کردم که در CRAN¹ و نسخه توسعه یافته تر در GitHub² موجود است. پیاده سازی‌های دیگری نیز وجود دارد که بر روی مدل‌های خاص تمرکز دارند: پکیج pre³ نرم افزار R آماره H را برای RuleFit پیاده‌سازی می‌کند. پکیج gbm⁴ نرم افزار R مدل‌های تقویت‌شده گرادیان و آماره H را پیاده‌سازی می‌کند.

۸.۳.۷ گزینه‌های جایگزین

آماره H تنها راه برای اندازه گیری تعاملات نیست:

شبکه‌های تعامل متغیر (VIN⁵) ارائه شده توسط Hooker (2004) رویکردی است که تابع پیش‌بینی را به اثرات اصلی و تعاملات ویژگی تجزیه می‌کند. سپس تعاملات بین ویژگی‌ها به عنوان یک شبکه تجسم می‌شود. متاسفانه هنوز نرم افزاری برای این روش در دسترس نیست.

تعامل ویژگی مبتنی بر وابستگی جزئی توسط Greenwell et al. (۲۰۱۸) تعامل بین دو ویژگی را اندازه گیری می‌کند. این رویکرد اهمیت ویژگی (تعریف شده به عنوان واریانس تابع وابستگی جزئی) یک ویژگی را مشروط به نقاط مختلف و ثابت ویژگی دیگر می‌سنجد. اگر واریانس بالا باشد، ویژگی‌ها با یکدیگر تعامل دارند، اگر صفر باشد، تعامل ندارند. پکیج مربوطه vip نرم افزار R در GitHub⁶ در دسترس است. این پکیج همچنین نمودارهای وابستگی جزئی و اهمیت ویژگی را دارا می‌باشد.

¹ <https://cran.r-project.org/web/packages/iml>

² <https://github.com/christophM/iml>

³ <https://cran.r-project.org/web/packages/pre/index.html>

⁴ <https://github.com/gbm-developers/gbm3>

⁵ Variable Interaction Networks

⁶ <https://github.com/koalaverse/vip>

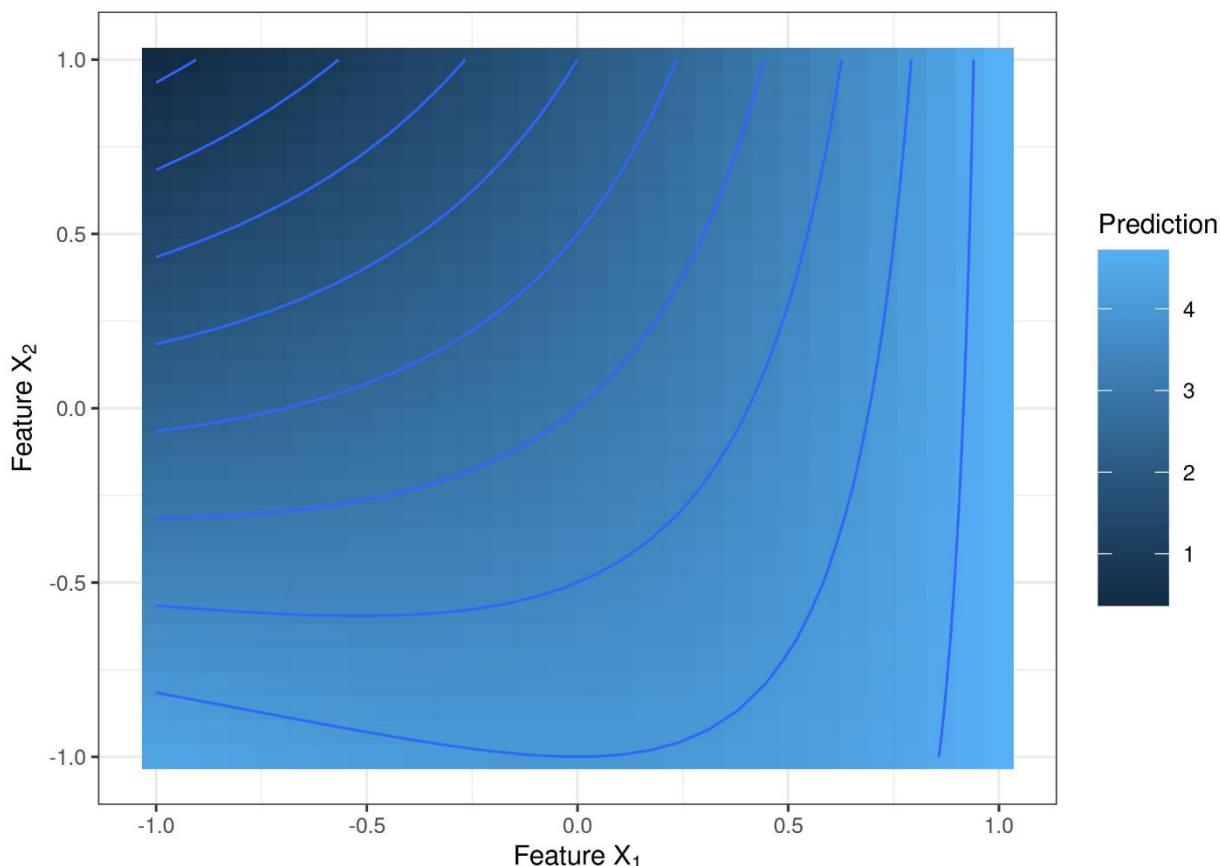
۸.۴ تجزیه تابعی^۱

یک مدل یادگیری ماشین نظارت شده را می‌توان به عنوان تابعی مشاهده کرد که یک بردار ویژگی با ابعاد بالا را به عنوان ورودی می‌گیرد و یک امتیاز پیش‌بینی یا طبقه‌بندی را به عنوان خروجی ایجاد می‌کند. تجزیه تابعی یک تکنیک تفسیری است که عملکرد با ابعاد بالا را تجزیه می‌کند و آن را به صورت مجموع اثرات ویژگی‌های فردی و اثرات تعاملی قابل تجسم بیان می‌کند. علاوه بر این، تجزیه تابعی یک اصل اساسی است که زیربنای بسیاری از تکنیک‌های تفسیری است – به شما کمک می‌کند روش‌های تفسیری دیگر را بهتر درک کنید.

اجازه دهید مستقیماً وارد شویم و یک تابع خاص را بررسی کنیم. این تابع دو ویژگی را به عنوان ورودی می‌گیرد و یک خروجی یک بعدی تولید می‌کند:

$$y = \hat{f}(x_1, x_2) = 2 + e^{x_1} - x_2 + x_1 \cdot x_2$$

تابع فوق را به عنوان یک مدل یادگیری ماشین در نظر بگیرید. ما می‌توانیم تابع را با یک نمودار سه بعدی یا یک نقشه حرارتی با خطوط کانتور تجسم کنیم:



شکل ۸.۲۲: سطح پیش‌بینی یک تابع با دو ویژگی X_1 و X_2

^۱ Functional Decomposition

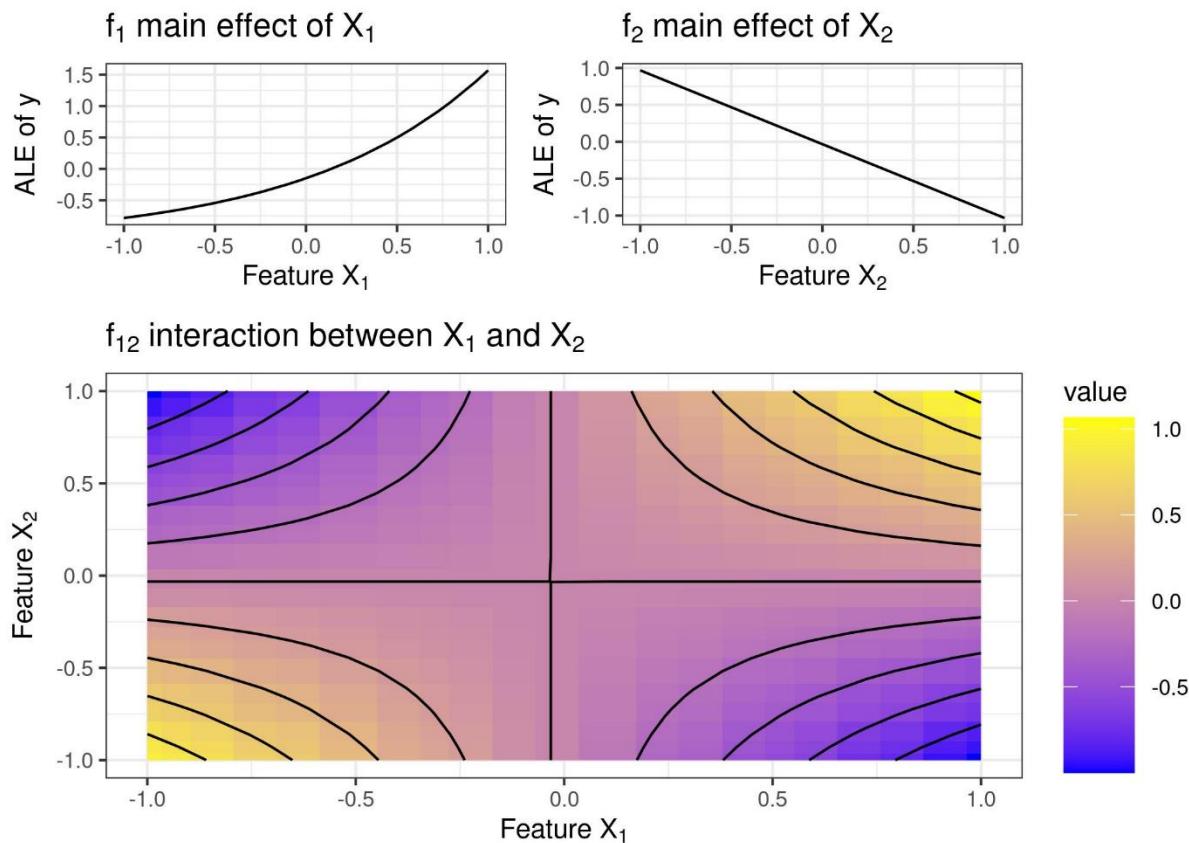
وقتی X_1 بزرگ و X_2 کوچک است، تابع، مقادیر بالا و هنگامی که X_2 بزرگ و X_1 کوچک است مقادیر پایین دارد. تابع پیش‌بینی صرفاً یک اثر افزایشی بین دو ویژگی نیست، بلکه یک تعامل بین این دو وجود دارد. وجود یک تعامل را می‌توان در شکل مشاهده کرد - اثر تغییر مقادیر برای ویژگی X_1 بستگی به مقدار ویژگی X_2 دارد.

کار ما اکنون این است که این تابع را به اثرات اصلی ویژگی‌های X_1 و X_2 و یک عبارت تعامل تجزیه کنیم. برای تابع دو بعدی f که فقط به دو ویژگی ورودی بستگی دارد: $\hat{f}(x_1 \cdot x_2)$ ، ما می‌خواهیم هر جزء یک اثر اصلی (\hat{f}_1) یا (\hat{f}_2)، اثر تعامل ($\hat{f}_{1.2}$) یا عرض از مبدا (\hat{f}_0) را نشان دهد.

$$\hat{f}(x_1 \cdot x_2) = \hat{f}_0 + \hat{f}_1(x_1) + \hat{f}_2(x_2) + \hat{f}_{1.2}(x_1 \cdot x_2)$$

اثرات اصلی نشان می‌دهد که چگونه هر ویژگی بر پیش‌بینی تأثیر می‌گذارد، مستقل از مقادیر ویژگی دیگر. اثر متقابل اثر مشترک ویژگی‌ها را نشان می‌دهد. عرض از مبدا به سادگی به ما می‌گوید که وقتی همه اثرات ویژگی روی صفر تنظیم می‌شوند، پیش‌بینی چیست. توجه داشته باشید که مولفه‌ها خود توابعی هستند (به جز عرض از مبدا) با ابعاد ورودی متفاوت.

من فقط مولفه‌ها را اکنون ارائه می‌دهم و بعداً توضیح می‌دهم که از کجا آمده‌اند. عرض از مبدا $\hat{f}_0 \sim 3 \cdot 18$ است. از آنجایی که سایر مؤلفه‌ها توابع هستند، می‌توانیم آنها را تجسم کنیم:



شکل ۸.۲۳: تجزیه یک تابع.

آیا با توجه به فرمول واقعی بالا، با نادیده گرفتن این که مقدار عرض از مبدا کمی تصادفی به نظر می‌رسد، آیا فکر می‌کنید مولفه‌ها منطقی هستند؟ ویژگی x_1 یک اثر اصلی نمایی را و x_2 اثر خطی منفی را نشان می‌دهد. عبارت تعامل کمی شبیه ورقه چیپس است. عبارت ریاضی آن، یک سهمی هذلولی است، همان‌طور که ما از x_1 و x_2 انتظار داریم. هشدار لو رفتن: تجزیه بر اساس نمودارهای اثر محلی انباشته است که در ادامه به آن خواهیم پرداخت.

۸.۴.۱ چگونه مولفه‌ها را محاسبه نکنیم I

اما چرا این همه هیجان؟ یک نگاه به فرمول قبل‌پاسخ تجزیه را به ما می‌دهد، بنابراین نیازی به روش‌های فانتزی نیست، درست است؟ برای ویژگی x_1 ، می‌توانیم تمام عباراتی را که فقط شامل x_1 می‌شوند، به عنوان مولفه آن ویژگی در نظر بگیریم. که خواهد بود $\hat{f}_1(x_1) = e^{x_1}$ و $\hat{f}_2(x_2) = -x_2$ برای ویژگی x_2 پس از آن تعامل $\hat{f}_{12}(x_1 \cdot x_2) = x_1 \cdot x_2$ است. در حالی که این پاسخ صحیح برای این مثال است (تا ثابت‌ها)، دو مشکل در این رویکرد وجود دارد: مشکل ۱): در حالی که مثال با فرمول شروع شد، واقعیت این است که تقریباً هیچ مدل یادگیری ماشین نمی‌تواند به اندازه آن فرمول واضح کار کند. ۲) در مورد تعامل قضیه بسیار پیچیده‌تر است و

به چیستی آن تعامل مربوط می‌شود. یک تابع ساده $x_1 \cdot x_2 = f(x_1 \cdot x_2)$ را تصور کنید، که در آن هر دو ویژگی مقادیر بزرگتر از صفر می‌گیرند و مستقل از یکدیگر هستند. با استفاده از تاکتیک نگاه‌کردن به فرمول، به این نتیجه می‌رسیم که یک تعامل بین ویژگی‌های x_1 و x_2 وجود دارد، اما نه اثرات ویژگی‌های فردی. اما آیا واقعاً می‌توانیم این ویژگی x_1 را بگوییم هیچ تأثیر فردی بر عملکرد پیش‌بینی ندارد؟ صرف‌نظر از اینکه ویژگی دیگر x_2 چه مقداری دارد، پیش‌بینی ما با افزایش x_1 افزایش می‌یابد. به عنوان مثال، برای $x_2 = 1$ اثر x_1 از رابطه $x_1 = f(x_1 \cdot 1)$ به دست می‌آید و وقتی $10 = x_2 = x_1 \cdot 10$ برای $f(x_1 \cdot 1)$. بنابراین، مشخص است که ویژگی x_1 تأثیر مثبتی بر پیش‌بینی دارد، مستقل از x_2 و صفر نیست.

برای حل مشکل ۱) عدم دسترسی به یک فرمول صریح، به روشنی نیاز داریم که فقط از امتیاز تابع پیش‌بینی یا طبقه‌بندی استفاده کند. برای حل مشکل ۲) عدم تعریف، به برخی بدهیات نیاز داریم که به ما بگوید مولفه‌ها چگونه باید باشند و چگونه با یکدیگر ارتباط دارند. اما ابتدا باید به طور دقیق‌تر تعریف کنیم که تجزیه تابعی چیست.

۸.۴.۲ تجزیه تابعی

یک تابع پیش‌بینی p ویژگی به عنوان ورودی می‌گیرد، $\mathbb{R} \rightarrow \mathbb{P}^p$ و یک خروجی تولید می‌کند. این تابع، می‌تواند یک تابع رگرسیون باشد، احتمال طبقه‌بندی برای یک کلاس معین یا امتیاز برای یک خوش^۱ معین (یادگیری ماشین بدون نظارت) باشد. می‌توانیم تابع پیش‌بینی را به صورت مجموع مؤلفه‌های تابعی نشان دهیم:

$$f(x) = f_0 + f_1(x_1) + \dots + f_p(x_p) + f_{1,2}(x_1 \cdot x_2) + \dots + f_{1,p}(x_1 \cdot x_p) + \dots + f_{p-1,p}(x_{p-1} \cdot x_p) + \dots + f_{1,\dots,p}(x_1 \cdot \dots \cdot x_p)$$

می‌توانیم فرمول تجزیه را با فهرست کردن همه زیرمجموعه‌های ممکن از ترکیبات ویژگی‌ها کمی‌زیباتر کنیم: $\{1, \dots, p\} \subseteq S$. این مجموعه شامل عرض از مبدا ($S = \emptyset$)، اثرات اصلی ($1 = |S|$) و تمام تعاملات ($|S| \geq 2$) است. با تعریف این زیرمجموعه، می‌توانیم تجزیه را به صورت زیر بنویسیم:

$$\hat{f}(x) = \sum_{S \subseteq \{1, \dots, p\}} \hat{f}_S(x_S)$$

در فرمول، x_S بردار ویژگی‌های مجموعه شاخص S است. و هر زیرمجموعه S یک مؤلفه تابعی را نشان می‌دهد، به عنوان مثال یک اثر اصلی اگر S فقط یک ویژگی داشته باشد یا یک تعاملی اگر $|S| > 1$.

در فرمول بالا چند مؤلفه وجود دارد؟ پاسخ این است که چند زیرمجموعه ممکن S از ویژگی‌های p ...۱. می‌توانیم تشکیل دهیم. این مقدار برابر با $2^p = \sum_{i=0}^p \binom{p}{i}$ زیرمجموعه‌های ممکن است. به عنوان مثال، اگر یک تابع از ۱۰ ویژگی استفاده کند، می‌توانیم تابع را به ۱۰۴۲ مؤلفه تجزیه کنیم: ۱ عرض از مبدا، ۱۰ اثر اصلی، ۹۰ عبارت تعامل دو طرفه، ۷۲۰ عبارت تعامل سه طرفه،... و با هر ویژگی بیشتر، تعداد مؤلفه‌ها دو برابر می‌شود

^۱ Cluster

واضح است که برای اکثر توابع، محاسبه همه مؤلفه‌ها امکان‌پذیر نیست. دلیل دیگر برای محاسبه نکردن همه مؤلفه‌ها این است که مؤلفه‌ها با $2 > |S|$ قابل تجسم نیستند و تفسیر آن‌ها دشوار است.

۸.۴.۳ چگونه مؤلفه‌ها را محاسبه نکنیم II

تا کنون از صحبت در مورد چگونگی تعریف و محاسبه مؤلفه‌ها اجتناب کرده‌ام. تنها محدودیت‌هایی که به طور ضمنی در مورد آن صحبت کردیم، تعداد و ابعاد مؤلفه‌ها بود، و اینکه مجموع مؤلفه‌ها باید تابع اصلی را ایجاد کند. اما بدون محدودیت بیشتر در مورد اینکه چه مؤلفه‌ها یی باید باشند، مؤلفه‌ها منحصر به فرد نیستند. این بدان معناست که می‌توانیم جای اثرات اصلی و تعاملات را جایه‌جا کنیم، یا تعاملات مرتبه پایین (چند ویژگی) و تعاملات مرتبه بالاتر (ویژگی‌های بیشتر) را تغییر دهیم. در مثال ابتدای بخش می‌توانیم هر دو اثر اصلی را صفر کنیم و اثرات آن‌ها را به صورت اثر تعامل اضافه کنیم.

در اینجا یک مثال اغراق شده ذکر می‌کنم تا نیاز به محدودیت در مؤلفه‌ها را نشان می‌دهد. فرض کنید یک تابع سه بعدی داریم. واقعاً مهم نیست که این تابع چگونه به نظر می‌رسد، اما تجزیه زیر همیشه کار می‌کند: \hat{f}_0 برابر با $0/12$ می‌باشد. تعداد کفش‌هایی که دارید $f_1(x_1) = 2 \cdot x_1 + f_{2,3} = f_{1,2} \cdot f_3$ و $\hat{f}_{1,3}$ همه صفر هستند. و برای اینکه این ترفند کار کند، تعریف می‌کنم $\hat{f}(x) = \sum_{S \subset \{1, \dots, p\}} \hat{f}_S(x_S)$. بنابراین عبارت تعاملی که شامل همه ویژگی‌ها می‌شود، به سادگی تمام اثرات باقی‌مانده را دارا می‌باشد، که طبق تعریف همیشه کار می‌کند، به این معنا که مجموع همه مؤلفه‌ها تابع پیش‌بینی اصلی را به ما می‌دهد. اگر بخواهید این را به عنوان تفسیر مدل خود ارائه دهید، این تجزیه خیلی معنی دار نخواهد بود و کاملاً گمراه کننده می‌باشد.

ابهام را می‌توان با تعیین محدودیت‌های بیشتر یا روش‌های خاص برای محاسبه مؤلفه‌ها از بین برد. در این بخش، سه روش متفاوت تجزیه تابعی را مورد بحث قرار خواهیم داد:

- ANOVA تابعی (تعمیم یافته).
- اثرات محلی انباشته
- مدل‌های رگرسیون آماری

۸.۴.۴ ANOVA تابعی

ANOVA تابعی توسط Hooker (2004) پیشنهاد شد. لازمه این رویکرد این است که تابع پیش‌بینی مدل \hat{f} انتگرال پذیر مجذوری^۱ باشد. مانند هر تجزیه تابعی، ANOVA تابعی تابع را به مؤلفه‌ها ی زیر تجزیه می‌کند:

$$\hat{f}(x) = \sum_{S \subset \{1, \dots, p\}} \hat{f}_S(x_S)$$

¹ square integrable

Hooker (2004) هر مولفه را با فرمول زیر تعریف می‌کند:

$$\hat{f}_S(x) = \int_{X_{-S}} \left(\hat{f}(x) - \sum_{V \subset S} \hat{f}_V(x) \right) dX_{-S}$$

می‌توانیم مولفه را به صورت زیر بازنویسی کنیم:

$$\hat{f}_S(x) = \int_{X_{-S}} (\hat{f}(x)) dX_{-S} - \int_{X_{-S}} \left(\sum_{V \subset S} \hat{f}_V(x) \right) dX_{-S}$$

در سمت چپ انتگرال تابع پیش‌بینی روی ویژگی‌های حذف شده از مجموعه S است که با S نمایش داده می‌شود. به عنوان مثال، اگر مؤلفه تعامل دو طرفه را برای ویژگی‌های ۲ و ۳ محاسبه کنیم، روی ویژگی‌های ۱، ۴، ۵، و ... همچنین انتگرال را می‌توان به این شکل دید مقدار امید‌ریاضی تابع پیش‌بینی روی X_{-S} با فرض اینکه همه ویژگی‌ها از یک توزیع یکنواخت از حداقل تا حداکثر مقدارشان پیروی می‌کنند. از این بازه، همه مولفه‌های زیرمجموعه‌های S کم می‌کنیم. این کم کردن، اثر تمام اثرات مرتبه پایین را حذف می‌کند و اثر را در مرکز متمرکز می‌کند. برای $\{1,2\}$ اثرات اصلی هر دو ویژگی \hat{f}_1 و \hat{f}_2 و همچنین عرض از مبدا را کم می‌کنیم. وقوع این اثرات مرتبه پایین تر، فرمول را بازگشتی می‌کند: ما باید از سلسله مراتب زیر مجموعه‌ها تا عرض از مبدا برویم و همه این مولفه‌ها را محاسبه کنیم. برای مولفه عرض از مبدا \hat{f}_0 ، زیر مجموعه مجموعه تهی $\{\emptyset\} = S$ ، و بنابراین S -شامل تمام ویژگی‌هاست:

$$\hat{f}_0(x) = \int_X \hat{f}(x) dX$$

رابطه فوق همان انتگرال تابع پیش‌بینی است روی همه ویژگی‌ها می‌باشد. زمانی که فرض کنیم همه ویژگی‌ها به طور یکنواخت توزیع شده‌اند، عرض از مبدا را می‌توان به عنوان امید ریاضی تابع پیش‌بینی نیز تفسیر کرد. حالا که \hat{f}_0 را می‌دانیم، می‌توانیم \hat{f}_1 را محاسبه کنیم (و در ادامه \hat{f}_2).

$$\hat{f}_1(x) = \int_{X_{-1}} (\hat{f}(x) - \hat{f}_0) dX_{-S}$$

برای پایان دادن به محاسبه برای مولفه $\hat{f}_{1,2}$ ، می‌توانیم همه چیز را کنار هم بگذاریم:

$$\begin{aligned} \hat{f}_{1,2}(x) &= \int_{X_{3,4}} \left(\hat{f}(x) - (\hat{f}_0(x) + \hat{f}_1(x) - \hat{f}_0 + \hat{f}_2(x) - \hat{f}_0) \right) dX_3 \cdot X_4 \\ &= \int_{X_{3,4}} (\hat{f}(x) - \hat{f}_1(x) - \hat{f}_2(x) + \hat{f}_0) dX_3 \cdot X_4 \end{aligned}$$

این مثال نشان می‌دهد که هر اثر مرتبه بالاتر این طور تعریف می‌شود: انتگرال‌گیری روی سایر ویژگی‌ها، البته با حذف تمام تاثیرات مرتبه پایین تری که زیرمجموعه‌هایی از مجموعه مورد علاقه ما هستند.

Hooker (2004) نشان داده است که این تعریف از مولفه‌های تابعی این اصول مطلوب را برآورده می‌کند:

- میانگین های صفر^۱: $\int \bar{f}_S(x_s) dX_s = 0$ برای هر $S \neq 0$
- تعامد^۲: $\int \bar{f}_S(x_s) dX_s = 0$ برای $S \neq V$
- تجزیه واریانس^۳: اگر $\sigma^2(\hat{f}) = \sum_{S \subseteq \{1, \dots, p\}} \sigma_S^2$ آنگاه $\hat{f}(x)^2 dX = f(x)^2 dX$

اصل میانگین های صفر ایجاب می کند که همه اثرات یا تعاملات در اطراف صفر متتمرکز شوند. بدین ترتیب، تفسیر در موقعیت X نسبت به پیش‌بینی متتمرکز است و نه پیش‌بینی مطلق.

اصل تعامد ایجاب می کند مولفه ها اطلاعات را به اشتراک نگذارند. به عنوان مثال، اثر مرتبه اول ویژگی X_1 و جمله تعامل X_1 و X_2 همبستگی ندارند. به دلیل تعامد، همه مولفه ها "خالص" هستند به این معنا که اثرات را با هم مخلوط نمی‌کنند. بسیار منطقی است که مولفه برای مثلاً ویژگی X_4 باید مستقل از جمله تعامل بین ویژگی ها X_1 و X_2 باشد. پیامد جالب‌تر برآمده از تعامد مؤلفه‌های سلسله مراتبی برای مواردی است یک مؤلفه حاوی ویژگی‌های دیگری می‌باشد مثلاً تعامل بین X_1 و X_2 و اثر اصلی ویژگی X_1 بر عکس یک نمودار وابستگی جزئی دو بعدی که برای X_1 و X_2 شامل چهار اثر بود: عرض از مبدا، دو اثر اصلی X_1 و X_2 و تعامل بین آنها. مولفه تابعی ANOVA برای $(x_1 \cdot x_2)$ فقط شامل تعامل خالص است.

تجزیه واریانس به ما امکان می‌دهد واریانس تابع \hat{f} را در میان مولفه‌ها تقسیم کنیم و تضمین می‌کند که واریانس کل تابع را در پایان تجمعی می‌شود. خاصیت تجزیه واریانس همچنین می‌تواند دلیل نامگذاری ANOVA را برای ما توضیح دهد. در آمار، ANOVA مخفف functional ANOVA است. ANOVA به مجموعه‌ای از روش‌ها اطلاق می‌شود که تفاوت‌ها را در میانگین یک متغیر هدف تحلیل می‌کنند. ANOVA با تقسیم واریانس و نسبت دادن آن به متغیرها کار می‌کند. بدین ترتیب، ANOVA تابعی را می‌توان به عنوان بسط این مفهوم برای هر تابعی دید.

مشکلات با ANOVA تابعی زمانی است که ویژگی‌ها همبسته هستند. به عنوان راه حل، ANOVA تابعی تعمیم یافته پیشنهاد شده است.

۸.۴.۵ ANOVA تابعی تعمیم یافته برای ویژگی‌های وابسته

مشابه بیشتر تکنیک‌های تفسیری مبتنی بر داده‌های نمونه‌گیری (مانند PDP)، ANOVA عملکردی می‌تواند نتایج گمراه‌کننده‌ای را در صورت وجود همبستگی بین ویژگی‌ها ایجاد کند. اگر روی توزیع یکنواخت انتگرال گیری کنیم، زمانی که در واقعیت ویژگی‌ها وابسته هستند، یک مجموعه‌داده جدید ایجاد می‌کنیم که از توزیع توام منحرف شده است و ترکیبات غیرمحتمل مقادیر ویژگی را برونویابی می‌کند.

¹ Zero Means

² Orthogonality

³ Variance Decomposition

ANOVA تابعی تعمیم یافته را پیشنهاد کرد. این روش، تجزیه‌ای است که برای ویژگی‌های وابسته کار می‌کند. این یک تعمیم از ANOVA تابعی است که قبلًا با آن آشنا شدیم، به این معنی که ANOVA تابعی یک حالت خاص از ANOVA تابعی تعمیم یافته است. مولفه‌ها بعنوان تصویر f بر روی فضای توابع افزایشی تعریف می‌شوند:

$$\hat{f}_S(x_S) = \operatorname{argmin}_{g_S \in L^2(\mathbb{R}^S), S \subset P} \int \left(\hat{f}(x) - \sum_{S \subset P} g_S(x_S) \right)^2 w(x) dx$$

به جای تعامد، مولفه‌ها یک شرط تعامد سلسله مراتبی را برآورده می‌کنند:

$$\forall \hat{f}_S(x_S) | S \subset U: \int \hat{f}_S(x_S) \hat{f}_U(x_U) w(x) dx = 0$$

تعامد سلسله مراتبی با تعامد متفاوت است. برای دو مجموعه ویژگی S و U که هیچ کدام زیر مجموعه دیگری نیستند (به عنوان مثال $S = \{1, 2\}$ و $U = \{2, 3\}$) لازم نیست \hat{f}_S و \hat{f}_U متعامد باشند برای این که تجزیه، تعامد سلسله مراتبی باشد. اما همه مولفه‌ها برای همه زیر مجموعه‌های S باید به \hat{f}_S عمود باشند. در نتیجه، تفسیر متفاوت از روش‌های مرتبط متفاوت است: مشابه نمودار M در بخش ALE، مؤلفه‌های ANOVA تابعی تعمیم یافته می‌توانند در گیر اثرات (حاشیه‌ای) ویژگی‌های همبسته شوند. اینکه آیا مولفه‌ها اثرات حاشیه‌ای در گیر می‌شوند یا خیر، به انتخاب تابع وزن $w(x)$ نیز بستگی دارد. اگر w را به اندازه یکنواخت در مکعب واحد انتخاب کنیم، ANOVA تابعی بخش قبل بدست می‌آید. یک انتخاب طبیعی برای w تابع توزیع احتمال توام است. اما توزیع توام معمولاً ناشناخته و برآورد آن دشوار است. یک ترفند می‌تواند این باشد که با اندازه‌گیری یکنواخت روی مکعب واحد شروع کنید و مناطق بدون داده را درنظر نگیریم.

برآورد بر روی شبکه‌ای از نقاط در فضای ویژگی انجام می‌شود و بعنوان یک مسئله کمینه سازی بیان می‌شود که با استفاده از تکنیک‌های رگرسیون قابل حل است. با این حال، مولفه‌ها را نمی‌توان به صورت جداگانه یا به صورت سلسله مراتبی محاسبه کرد. باید یک سیستم پیچیده از معادلات که شامل مولفه‌ها ی دیگر نیز می‌باشد، حل شود. بنابراین سیستم بسیار پیچیده و محاسبات سنگین است.

۸.۴.۶ نمودارهای اثر محلی انباسته

Nمودارهای ALE، Apley and Zhu (2020) همچنین یک تجزیه تابعی ارائه می‌دهند، به این معنی که با افزودن تمام نمودارهای ALE از عرض از مبدا گرفته تا نمودارهای ALE یک بعدی، نمودارهای ALE دو بعدی و غیره، تابع پیش‌بینی به دست می‌آید. ALE با ANOVA تابعی عملکردی (تعمیم یافته) متفاوت است، زیرا مولفه‌ها تعامد ندارند، اما، همان‌طور که نویسنده‌گان آن نامگذاری کرده‌اند، شبیه تعامد^۱ دارند. برای درک شبیه تعامد، باید

^۱ pseudo-orthogonality

عملگر H_S را تعریف کنیم، که یکتابع \hat{f} می‌گیرد و آن را به نمودار ALE خود برای زیرمجموعه ویژگی S نگاشت می‌کند. به عنوان مثال، عملگر $H_{1,2}$ به عنوان ورودی یک مدل یادگیری ماشین را می‌گیرد و نمودار دو بعدی ALE را برای ویژگی های ۱ و ۲ تولید می‌کند: اگر یک عملگر را دو بار اعمال کنیم، همان نمودار ALE را به دست می‌آوریم. پس از اعمال اپراتور $H_{1,2}$ برای بار اول به f ما نمودار ALE دو بعدی \hat{f}_{ALE12} را بدست می‌آوریم. سپس دوباره عملگر را اعمال می‌کنیم البته نه به f بلکه به \hat{f}_{ALE12} . این امکان پذیر است زیرا مولفه ALE دو بعدی خود یکتابع است. نتیجه دوباره \hat{f}_{ALE12} است و بدین معنی است که می‌توانیم یک عملگر را چندین بار اعمال کنیم و همیشه همان نمودار ALE را دریافت کنیم. این بخش اول شبه متعامد است. اما اگر دو عملگر متفاوت را برای مجموعه ویژگی‌های مختلف اعمال کنیم، نتیجه چیست؟ مثلاً H_1 و $H_{1,2}$ یا $H_{3,4,5}$ ؟ جواب صفر است. اگر ابتدا عملگر H_S نمودار ALE را به یکتابع اعمال کنیم و سپس H_U به نتیجه اعمال شود (با فرض $U \neq S$)، آن گاه نتیجه صفر می‌شود. به عبارت دیگر، نمودار ALE برای نمودار ALE صفر است، مگر اینکه یک نمودار ALE را دو بار اعمال کنیم. یا به عبارت دیگر، نمودار ALE برای مجموعه ویژگی S شامل هیچ نمودار ALE دیگری نیست. با تعریف ریاضی، عملگر ALE توابع را به زیرفضاهای متعامد یک فضای ضرب داخلی، نگاشت می‌کند.

همان‌طور که Apley and Zhu (۲۰۲۰) اشاره می‌کنند، شبه تعامل ممکن است مطلوب‌تر از تعامل سلسله مراتبی باشد، زیرا اثرات حاشیه‌ای ویژگی‌ها را درگیر نمی‌کند. علاوه بر آن، ALE نیازی به تخمین توزیع توام ندارد. مولفه‌ها را می‌توان به صورت سلسله مراتبی تخمین زد، به این معنی که محاسبه ALE دو بعدی برای ویژگی‌های ۱ و ۲ فقط نیاز به محاسبات مولفه‌های ALE منفرد ۱ و ۲ و عرض از مبدأ دارد.

۸.۴.۷ مدل‌های رگرسیون آماری

این رویکرد با مدل‌های قابل تفسیر، به ویژه مدل‌های تجمعی تعمیم یافته (GAM) یکسانند. به جای تجزیه یکتابع پیچیده، می‌توانیم محدودیت‌هایی را در فرآیند مدل‌سازی ایجاد کنیم تا بتوانیم به راحتی مولفه‌ها را به صورت جداگانه بدست آوریم. در حالی که تجزیه را می‌توان به روشهای بالا به پایین انجام داد، جایی که ما با یکتابع با ابعاد بالا شروع می‌کنیم و آن را تجزیه می‌کنیم، مدل‌های افزایشی تعمیم یافته یک رویکرد از پایین به بالا ارائه می‌دهند و مدل را از مولفه‌های ساده می‌سازیم. هر دو رویکرد، هدفی مشترک دارند که ارائه مؤلفه‌های فردی و قابل تفسیر است. در مدل‌های آماری، تعداد مؤلفه‌ها را محدود می‌کنیم تا مجبور به برازش همه 2^p مولفه نباشیم. ساده ترین نسخه، رگرسیون خطی است:

$$\hat{f}(x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

فرمول بسیار شبیه به تجزیه تابعی است، اما با دو تغییر عمده. تغییر ۱: همه اثرات تعاملی حذف می‌شوند و ما فقط اثرات عرض از مبدأ و اصلی را نگه می‌داریم. تغییر ۲: اثرات اصلی ویژگی‌ها خطی هستند: $\hat{f}_j(x_j) = \beta_j x_j$

با مشاهده مدل رگرسیون خطی از طریق دیدگاه تجزیه عملکردی، می‌بینیم که خود مدل، یک تجزیه تابعی از تابع واقعی را نشان می‌دهد که ویژگی‌ها را به هدف نگاشت می‌کند، با این فرضیات قوی که اثرات، خطی هستند و هیچ تعاملی وجود ندارد.

مدل افزودنی تعمیم یافته با اجازه دادن به توابع انعطاف پذیرتر \hat{f}_j ، از طریق استفاده از اسپلاین‌ها از فرض دوم صرفنظر می‌کند. تعاملات نیز می‌توانند اضافه شوند، اما این فرآیند تقریباً دستی است. رویکردهایی مانند GA2M تلاش می‌کنند تا تعاملات دو طرفه را به طور خودکار به یک GAM اضافه کنند (Caruana et al., 2015).

فکر کردن به یک مدل رگرسیون خطی یا یک GAM به عنوان تجزیه تابعی نیز می‌تواند منجر به سردرگمی شود. رویکردهای تجزیه ای را که پیش‌تر ارائه شدند (ANOVA تابعی تعمیم یافته و اثرات محلی انباشته)، ممکن است مؤلفه‌هایی را محاسبه کنند که با مؤلفه‌هایی که مستقیماً از GAM بدست می‌آیند متفاوت باشند. این حالت می‌تواند زمانی اتفاق بیفتند که اثرات متقابل ویژگی‌های همبسته در GAM مدل شود. این اختلاف به این دلیل رخ می‌دهد که سایر رویکردهای تجزیه تابعی تأثیرات را به طور متفاوتی بین تعاملات و اثرات اصلی تقسیم می‌کنند.

پس چه زمانی باید از GAM به جای مدل پیچیده + تجزیه استفاده کرد؟ زمانی که بیشتر تعاملات صفر است، باید به GAM‌ها پایبند باشید، به خصوص زمانی که هیچ تعاملی بین سه یا تعداد بیشتری ویژگی وجود ندارد. اگر بدانیم که حداقل تعداد ویژگی‌های درگیر در تعاملات دو است ($2 \leq |S|$) می‌توانیم از رویکردهایی مانند MARS یا GA2M استفاده کنیم. در نهایت، عملکرد مدل در داده‌های آزمایشی ممکن است نشان دهد که آیا یک GAM کافی است یا یک مدل پیچیده‌تر بسیار بهتر عمل می‌کند.

۸.۴.۸ طرح وابستگی جزئی

آیا نمودار وابستگی جزئی نیز تجزیه تابعی را ارائه می‌دهد؟ پاسخ کوتاه: خیر. پاسخ طولانی‌تر: نمودار وابستگی جزئی برای مجموعه ویژگی S همیشه شامل تمام اثرات سلسله مراتب می‌باشد - PDP برای $\{1, 2\}$ نه تنها تعامل، بلکه اثرات ویژگی‌های فردی را نیز شامل می‌شود. در نتیجه، افزودن تمام PDP‌ها برای همه زیر مجموعه‌ها، تابع اصلی را به دست نمی‌آورد، و بنابراین تجزیه معتبری نیست. اما آیا می‌توانیم PDP را با حذف همه اثرات پایین‌تر تنظیم کنیم؟ بله، می‌توانیم، اما چیزی شبیه به ANOVA عملکردی دریافت می‌کنیم. با این تفاوت که، به جای انتگرالگیری بر روی یک توزیع یکنواخت، PDP روی توزیع حاشیه ای S -X انتگرالگیری می‌کند که با استفاده از نمونه گیری مونت کارلو برآورد شده است.

۴.۹. مزايا

من تجزیه تابعی را مفهوم اصلی تفسیرپذیری یادگیری ماشین می‌دانم.

تجزیه تابعی یک توجیه نظری برای تجزیه مدل‌های یادگیری ماشین با ابعاد بالا و پیچیده به اثرات و تعاملات فردی به ما می‌دهد - مرحله‌ای ضروری که به ما امکان می‌دهد تا اثرات فردی را تفسیر کنیم. تجزیه تابعی ایده اصلی تکنیک‌هایی مانند مدل‌های رگرسیون آماری، ALE، ANOVA تابعی (تعمیم‌یافته)، PDP، آماره H و منحنی‌های ICE است.

تجزیه تابعی همچنین درک بهتری از روش‌های دیگر فراهم می‌کند. به عنوان مثال، اهمیت ویژگی جایگشت ارتباط بین یک ویژگی و هدف را می‌شکند. اگر از طریق لنز تجزیه تابعی مشاهده کنیم، می‌توانیم ببینیم که جایگشت اثر تمام مولفه‌ها یی را که ویژگی در آن دخیل است، «از بین می‌برد». این کار بر اثر اصلی ویژگی، بلکه بر تمام تعاملات با سایر ویژگی‌ها نیز تأثیر می‌گذارد. به عنوان مثال دیگری، مقادیر Shapley یک پیش‌بینی را به اثرات افزایشی ویژگی فردی تجزیه می‌کند. اما تجزیه تابعی به ما می‌گوید که باید اثرات تعاملی نیز در تجزیه وجود داشته باشد، پس آنها کجا هستند؟ مقادیر Shapley یک نسبت منصفانه از اثرات را به ویژگی‌های فردی ایجاد می‌کند، به این معنی که همه تعاملات نیز به طور منصفانه به ویژگی‌ها نسبت داده می‌شوند و بنابراین بین مقادیر Shapley تقسیم می‌شوند.

هنگام در نظر گرفتن تجزیه تابعی به عنوان یک ابزار، استفاده از نمودارهای ALE مزایای بسیاری دارند. نمودارهای ALE تجزیه تابعی را ارائه می‌دهند که محاسبه آن سریع است، دارای پیاده‌سازی نرم افزاری می‌باشد (به فصل ALE مراجعه کنید)، و ویژگی‌های شبه متعامد مطلوب دارد.

۸.۴.۱۰ معايب

مفهوم تجزیه تابعی به سرعت مرز دو ویژگی را رد می‌کند و به مولفه‌های با ابعاد تر از تعامل بین دو ویژگی می‌رسد. این انفجار نمایی در تعداد ویژگی‌ها عملی بودن را محدود می‌کند، زیرا نمی‌توانیم به راحتی تعاملات مرتبه بالاتر را تجسم کنیم. علاوه بر این، اگر بخواهیم همه تعاملات را محاسبه کنیم، زمان محاسباتی دیوانه‌کننده می‌شود.

هر روشی از تجزیه تابعی دارای معايب مخصوص خود است. رویکرد پایین به بالا - ساخت مدل‌های رگرسیون - یک فرآيند کاملاً دستی است و محدودیت‌های زیادی را بر مدل تحمیل می‌کند که می‌تواند بر عملکرد پیش‌بینی تأثیر بگذارد. ANOVA تابعی به ویژگی‌های مستقل نیاز دارد. برآورد ANOVA تابعی تعمیم‌یافته بسیار دشوار است. نمودارهای اثر محلی انباسته تجزیه واریانس را ارائه نمی‌دهند. رویکرد تجزیه تابعی برای تجزیه و تحلیل داده‌های جدولی مناسب‌تر از متن یا تصاویر است.

۸.۵ اهمیت ویژگی جایگشتی

اهمیت ویژگی جایگشتی، افزایش خطای پیش‌بینی مدل را در حالتی اندازه گیری می‌کند، که ما مقادیر ویژگی را جابجا می‌کنیم و در حقیقت، رابطه بین ویژگی و خروجی واقعی را قطع کردیم.

۸.۵.۱ تئوری

مفهوم واقعاً ساده است: ما اهمیت یک ویژگی را با محاسبه افزایش خطای پیش‌بینی مدل پس از تغییر ویژگی اندازه می‌گیریم. یک ویژگی در صورتی "مهم" است که به هم زدن مقادیر آن، خطای مدل را افزایش دهد، زیرا در این صورت مدل برای پیش‌بینی به آن ویژگی متکی است. یک ویژگی "بی اهمیت" است اگر به هم زدن مقادیر آن خطای مدل را بدون تغییر باقی بگذارد، زیرا در این حالت، مدل ویژگی را برای پیش‌بینی نادیده می‌گیرد. اندازه گیری اهمیت ویژگی جایگشت توسط Breiman (2001) برای جنگل تصادفی معرفی شد. بر اساس این ایده، Fisher et al. (۲۰۱۹) یک نسخه آگنوستیک مدل از اهمیت ویژگی پیشنهاد کرد و آن را اடکای مدل^۱ نامیدند. آنها همچنین ایده‌های پیشرفت‌تری را در مورد اهمیت ویژگی معرفی کردند، به عنوان مثال یک نسخه (خاص مدل) که حاوی این نظر است که بسیاری از مدل‌های پیش‌بینی ممکن است داده‌ها را به خوبی پیش‌بینی کنند. مقاله آنها ارزش خواندن دارد.

الگوریتم اهمیت ویژگی جایگشتی براساس Fisher et al. (۲۰۱۹) :

وروودی: مدل آموزش‌دهنده \hat{f} ، ماتریس ویژگی X ، بردار هدف y ، اندازه خطای $L(y, \hat{f})$.

۱- خطای اصلی مدل $e_{orig} = L(y, \hat{f}(X))$ (به عنوان مثال میانگین مربعات خطای \hat{f}) را برآورد کنید.

۲- برای هر ویژگی $\{z\} \in \{1, \dots, p\}$ مراحل زیر را انجام دهید:

- ایجاد ماتریس ویژگی X_{perm} با جایگشتی کردن ویژگی z در داده X این کار ارتباط بین ویژگی z و خروجی واقعی y را از بین می‌برد.

- خطای $e_{perm} = L(Y, \hat{f}(X_{perm}))$ بر اساس پیش‌بینی داده‌های جایگشتی برآورد کنید.

- اهمیت ویژگی جایگشتی را با استفاده از نسبت $FI_j = e_{perm}/e_{orig}$ محاسبه کنید.

۳- ویژگی‌ها را بر اساس FI به صورت نزولی مرتب کنید.

Fisher et al. (۲۰۱۹) در مقاله خود پیشنهاد می‌کنند که مجموعه داده را به نصف تقسیم کرده و مقادیر ویژگی z دو نیمه را جابجا کنید به جای جایگشتی کردن ویژگی z . اگر در مورد آن فکر کنید، این دقیقاً مشابه جایگشتی کردن ویژگی z است. اگر برآورد دقیق‌تری می‌خواهید، می‌توانید با جفت کردن هر نمونه با مقدار ویژگی z نمونه‌های دیگر (به جز با خودش) خطای جایگشتی ویژگی z را تخمین بزنید. این کار یک مجموعه داده

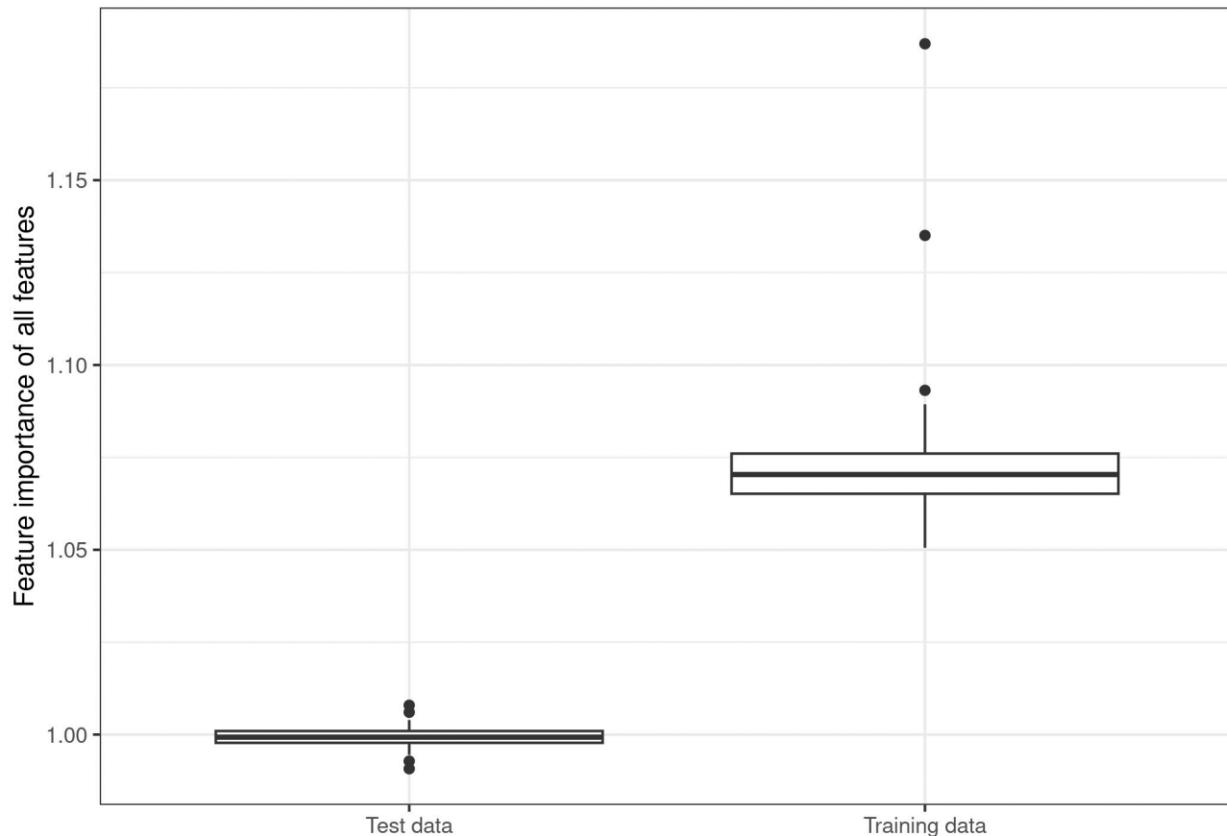
¹ model reliance

با اندازه $(n - n)$ برای تخمین خطای جایگشت به شما می‌دهد و زمان محاسباتی زیادی را می‌طلبد. من فقط در صورتی می‌توانم استفاده از روش $(n - n)$ را توصیه کنم که در مورد تخمین‌های بسیار دقیق جدی هستید.

۸.۵.۲ آیا باید اهمیت را روی داده‌های آموزش را محاسبه کنم یا نست؟

اگر حوصله خواندن این بخش را ندارید جواب این است: احتمالاً باید از داده‌های تست استفاده کنید. پاسخ به این سؤال درباره داده‌های آموزشی یا نست، به این سؤال اساسی که اهمیت ویژگی چیست، بستگی دارد. بهترین راه برای درک تفاوت بین اهمیت ویژگی بر اساس داده‌های آموزش در مقابل داده‌های تست، یک مثال «افراتی» است. من یک ماشین بردار پشتیبان را آموزش دادم تا یک مقدار خروجی پیوسته و تصادفی را با توجه به ۵۰ ویژگی تصادفی (۲۰۰ نمونه) پیش‌بینی کند. منظور من از "تصادفی" این است که خروجی هدف مستقل از ۵۰ ویژگی است. این مانند پیش‌بینی دمای فردا با توجه به آخرین اعداد قرعه کشی است. اگر مدل رابطه‌ای را "یاد بگیرد"، بیش رخ داده است و در واقع، SVM بر روی داده‌های آموزشی بیش از حد برازش داده شده است. میانگین خطای مطلق (به صورت خلاصه^۱ mae) برای داده‌های آموزشی $0/29$ و برای داده‌های تست $0/82$ است و همچنان خطای بهترین مدل ممکن است که همیشه میانگین نتیجه 0 را پیش‌بینی می‌کند ($mae = 0/78$). به عبارت دیگر مدل SVM به درد نخور است. چه مقادیری برای اهمیت ویژگی برای ۵۰ ویژگی این SVM بیش برازش شده انتظار دارید؟ آیا صفر انتظار دارید زیرا هیچ یک از ویژگی‌ها به بهبود عملکرد در داده‌های تست دیده نشده کمک نمی‌کند؟ یا اینکه آیا این اهمیت‌ها باید منعکس کنند که چقدر مدل به هر یک از ویژگی‌ها بستگی دارد، صرف نظر از اینکه آیا روابط آموخته شده به داده‌های دیده نشده تعمیم می‌یابد یا نه؟ اجازه دهید نگاهی بیندازیم که چگونه توزیع اهمیت ویژگی‌ها برای داده‌های آموزشی و تست متفاوت است.

^۱ mean absolute error



شکل ۸.۲۴: توزیع مقادیر اهمیت ویژگی بر اساس نوع داده. یک SVM بر روی یک مجموعه داده رگرسیون با ۵۰ ویژگی تصادفی و ۲۰۰ نمونه آموزش داده شد. SVM روی داده‌ها بیش برآذش دارند: اهمیت ویژگی بر اساس داده‌های آموزشی بسیاری از ویژگی‌های را مهم نشان می‌دهد. بر اساس داده‌های تست دیده نشده، اهمیت ویژگی‌ها نزدیک به نسبت یک ($=\text{بی اهمیت}$) است. برای من مشخص نیست که کدام یک از این دو نتیجه مطلوب‌تر است. بنابراین من سعی خواهم کرد برای هر دو نسخه موردنی ایجاد کنم.

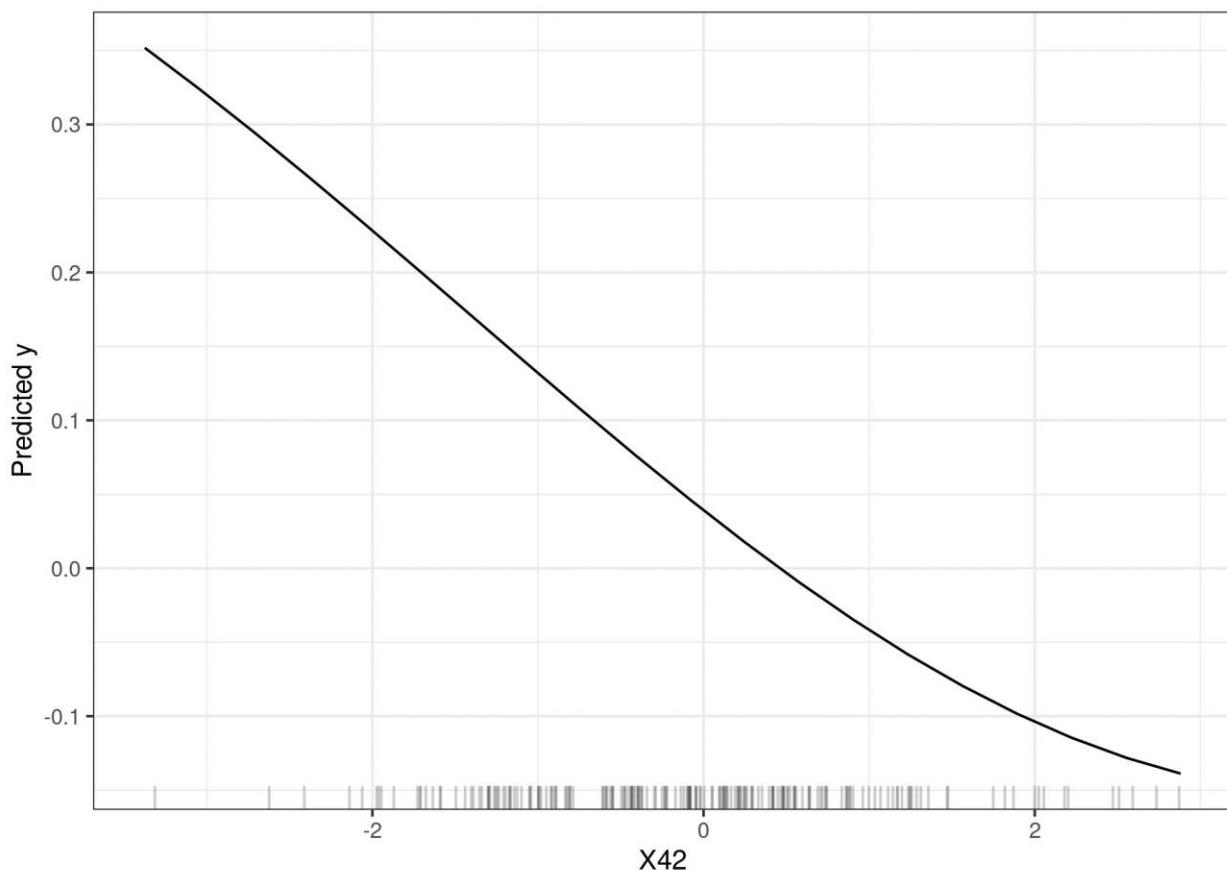
مورد برای داده‌های تست

این یک مورد ساده است: تخمین‌های خطای مدل بر اساس داده‌های آموزشی به درد نخور هستند و اهمیت ویژگی به تخمین خطای مدل متکی است در نتیجه اهمیت ویژگی براساس داده‌های آموزشی به درد نخور است. در واقع، این یکی از اولین چیزهایی است که در یادگیری ماشین یاد می‌گیرید: اگر خطای مدل (یا عملکرد) را بر روی همان داده‌هایی که مدل آموزش داده شده است اندازه گیری کنید، اندازه گیری معمولاً خیلی خوش‌بینانه است، به این معنی که به نظر می‌رسد مدل خیلی بهتر از واقعیت کار می‌کند و از آنجایی که اهمیت ویژگی جایگشتی به اندازه گیری خطای مدل بستگی دارد، باید از داده‌های تست دیده نشده استفاده کنیم.

اهمیت ویژگی بر اساس داده‌های آموزشی باعث می‌شود ما به اشتباه فکر کنیم که ویژگی‌ها برای پیش‌بینی‌ها مهم هستند، در حالی که در واقعیت مدل بیش برآذش داشت و ویژگی‌ها اصلاً مهم نبودند.

موردی برای داده‌های آموزشی

فرمول‌بندی استدلال‌های استفاده از داده‌های آموزشی تا حدودی دشوارتر است، اما به نظر من، به اندازه استدلال‌های استفاده از داده‌های تست قانع‌کننده است. نگاهی دیگر به SVM به درد نخور خود می‌اندازیم. بر اساس داده‌های آموزشی، مهم‌ترین ویژگی X42 بود. اجازه دهید به نمودار وابستگی جزئی ویژگی X42 نگاه کنیم. نمودار وابستگی جزئی نشان می‌دهد که چگونه خروجی مدل بر اساس تغییرات ویژگی تغییر می‌کند و بر خطای تعمیم تکیه نمی‌کند. فرقی نمی‌کند که PDP با داده‌های آموزشی یا تست محاسبه شود.



شکل ۸.۲۵ PDP: ویژگی X42، که با توجه‌به اهمیت ویژگی بر اساس داده‌های آموزشی، مهم‌ترین ویژگی است. نمودار نشان می‌دهد که چگونه SVM برای پیش‌بینی به این ویژگی وابسته است.

نمودار به وضوح نشان می‌دهد که SVM یاد گرفته است برای پیش‌بینی‌های خود به ویژگی X42 تکیه کند، اما با توجه‌به اهمیت ویژگی بر اساس داده‌های تست، این ویژگی مهم نیست. بر اساس داده‌های آموزشی، اهمیت

۱/۱۹ است که نشان می‌دهد مدل یاد گرفته است از این ویژگی استفاده کند. اهمیت ویژگی بر اساس داده‌های آموزشی به ما می‌گوید که کدام ویژگی برای مدل مهم است به این معنا که برای پیش‌بینی به آنها بستگی دارد. به عنوان بخشی از مورد استفاده از داده‌های آموزشی، می‌خواهیم استدلالی علیه داده‌های تست معرفی کنم. در عمل، شما می‌خواهید از تمام داده‌های خود برای آموزش مدل خود استفاده کنید تا در نهایت بهترین مدل ممکن را به دست آورید. این بدان معناست که هیچ داده تست استفاده نشده ای برای محاسبه اهمیت ویژگی باقی نمانده است. هنگامی که می‌خواهید خطای تعمیم مدل خود را تخمین بزنید، همین مشکل را دارید. اگر از اعتبارسنجی متقاطع (تودرتو^۱) برای تخمین اهمیت ویژگی استفاده کنید، با این مشکل مواجه خواهید شد که اهمیت ویژگی در مدل نهایی با همه داده‌ها محاسبه نمی‌شود، بلکه در مدل‌هایی با زیرمجموعه‌هایی از داده‌ها که ممکن است رفتار متفاوتی داشته باشند، محاسبه می‌شود.

با این حال، در پایان توصیه می‌کنم از داده‌های تست برای اهمیت ویژگی جایگشت استفاده کنید. زیرا اگر علاقه‌مند هستید که پیش‌بینی‌های مدل چقدر تحت تأثیر یک ویژگی است، باید از معیارهای اهمیت دیگری مانند اهمیت SHAP استفاده کنید.

در ادامه به چند مثال نگاه می‌کنیم. من محاسبه اهمیت را بر اساس داده‌های آموزشی انجام دادم، زیرا باید یکی را انتخاب می‌کردم و استفاده از داده‌های آموزشی به چند خط کد کمتر نیاز داشت.

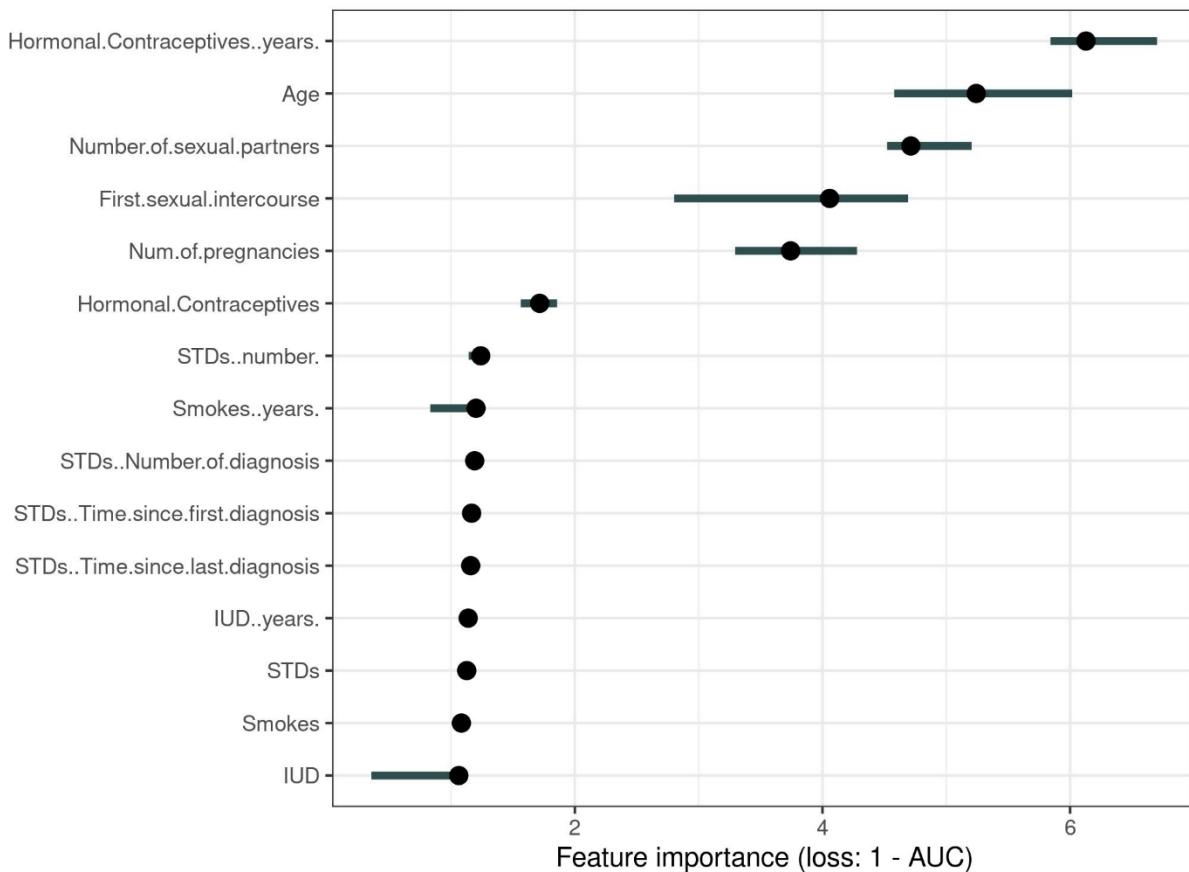
۸.۵.۳ مثال و تفسیر

من مثال‌هایی را برای طبقه‌بندی و رگرسیون نشان می‌دهم.

سرطان دهانه رحم (طبقه‌بندی)

ما یک مدل جنگل تصادفی را برای پیش‌بینی سرطان دهانه رحم برآذش می‌دهیم. افزایش خطای را با رابطه $1 - AUC$ (یک منهای سطح زیر منحنی ROC) اندازه گیری می‌کنیم. ویژگی‌های با ضریب افزایش خطای مدل ۱ (= بدون تغییر) برای پیش‌بینی سرطان دهانه رحم مهم نیستند.

^۱ nested

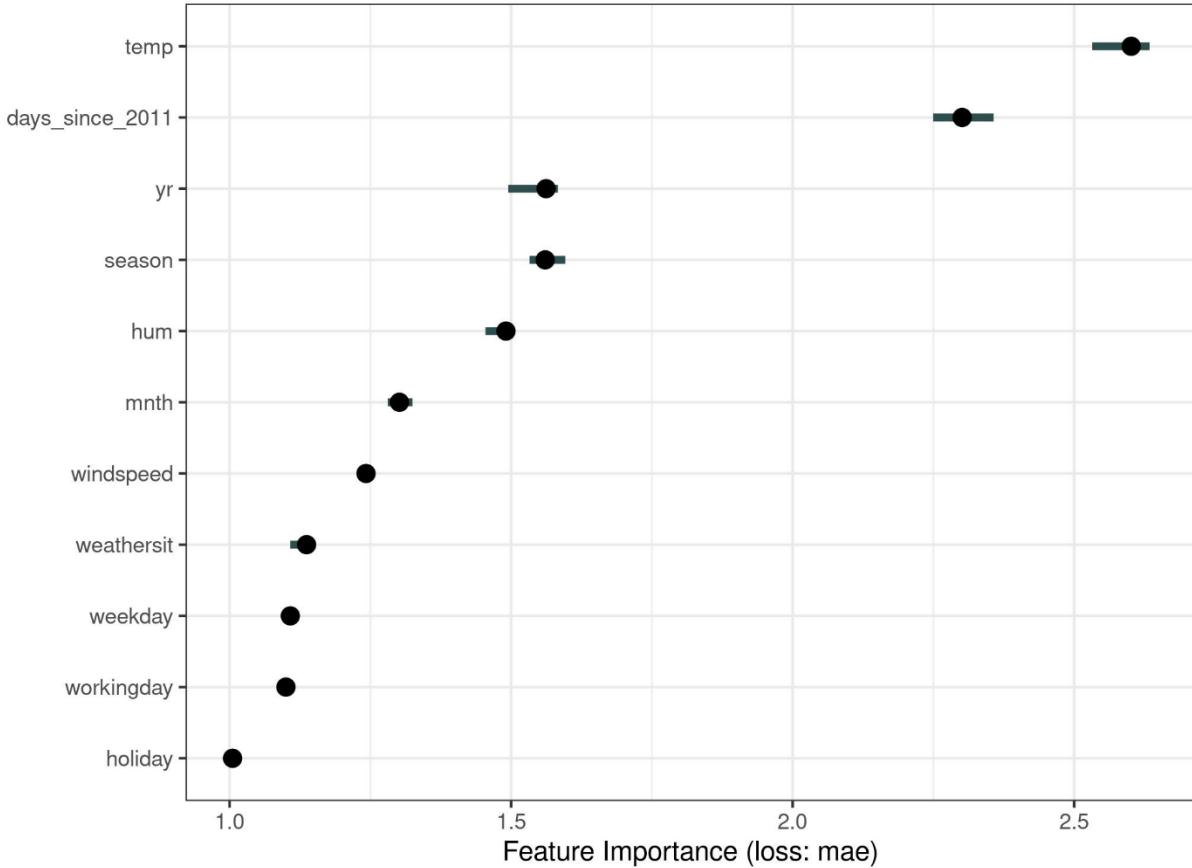


شکل ۸.۲۶: اهمیت هر یک از ویژگی‌ها برای پیش‌بینی سلطان دهانه رحم با یک جنگل تصادفی. مهم‌ترین ویژگی، سال‌های پیشگیری از بارداری هورمونی بود. جایگشت سال‌های پیشگیری از بارداری هورمونی، منجر به افزایش $AUC - 1$ با ضریب ۰.۱۳ شد.

مهم‌ترین ویژگی سال‌های پیشگیری هورمونی با افزایش خطای ۰.۱۳ پس از جایگشت بود.

اشتراک دوچرخه (رگرسیون)

ما یک مدل ماشین بردار پشتیبان برای پیش‌بینی تعداد دوچرخه‌های اجاره‌ای، با توجه به شرایط آب‌وهایی و اطلاعات تقویم، برآذش می‌دهیم. به عنوان اندازه گیری خطای میانگین خطای مطلق استفاده می‌کنیم.



شکل ۸.۲۷: اهمیت هر یک از ویژگی ها در پیش بینی شمارش دوچرخه با ماشین بردار پشتیبان. مهم ترین ویژگی دما بود، کمترین اهمیت مربوط تعطیلات بود.

۸.۵.۴ مزایا

تفسیر خوب: اهمیت ویژگی، افزایش خطای مدل است زمانی که اطلاعات ویژگی از بین می رود.
اهمیت ویژگی یک بینش بسیار فشرده و کلی در مورد رفتار مدل ارائه می کند.
یک جنبه مثبت استفاده از نسبت خطای تفاوت خطای تفاوت خطا این است که اندازه گیری اهمیت ویژگی در مسائل مختلف قابل مقایسه است.

اندازه گیری اهمیت به طور خودکار تمام تعاملات با سایر ویژگی ها را در نظر می گیرد. با جایگشتن کردن ویژگی، اثرات تعامل با سایر ویژگی ها را نیز از بین می برد. این بدان معنی است که اهمیت ویژگی جایگشتنی هم اثر ویژگی اصلی و هم اثرات تعاملی بر عملکرد مدل را در بر می گیرد. این موضوع همچنین یک نقطه ضعف نیز محسوب می شود زیرا اهمیت تعامل بین دو ویژگی در اندازه گیری اهمیت هر دو ویژگی گنجانده شده است. این بدان معناست که اهمیت ویژگی ها به کاهش کل عملکرد سرجمع نمی شود و مجموع بزرگ تر است. تنها در

صورتی که هیچ تعاملی بین ویژگی‌ها وجود نداشته باشد، مانند یک مدل خطی، اهمیت تقریباً سرجمع می‌شوند.

اهمیت ویژگی جایگشتی نیازی به آموزش مجدد مدل ندارد. برخی از روش‌های دیگر حذف یک ویژگی، آموزش مجدد مدل و سپس مقایسه خطای مدل را پیشنهاد می‌کنند. بازآموزی یک مدل یادگیری ماشین می‌تواند زمان زیادی طول بکشد. « فقط » جایگشت یک ویژگی می‌تواند در زمان صرفه‌جویی زیادی کند. روش‌های مهمی که مدل را با زیرمجموعه‌ای از ویژگی‌ها بازآموزی می‌کنند در نگاه اول شهودی به نظر می‌رسند، اما مدل با داده‌های کاهش‌یافته برای اهمیت ویژگی بی‌معنی است. ما به اهمیت ویژگی یک مدل ثابت علاقه مندیم. بازآموزی با مجموعه‌داده کاهش‌یافته، مدلی متفاوت از مدل مورد علاقه ما ایجاد می‌کند. فرض کنید یک مدل خطی محدود (با Lasso) با تعداد مشخصی از ویژگی‌ها با وزن غیر صفر را آموزش می‌دهید. مجموعه‌داده دارای ۱۰۰ ویژگی است، شما تعداد وزن‌های غیر صفر را ۵ می‌کنید. اهمیت یکی از ویژگی‌هایی که وزن غیر صفر دارند را تحلیل می‌کنید. شما ویژگی را حذف کرده و مدل را دوباره آموزش می‌دهید. عملکرد مدل ثابت می‌ماند زیرا یکی دیگر از ویژگی‌های به همان اندازه خوب وزن غیر صفر می‌گیرد و نتیجه شما این است که ویژگی مهم نبوده است. مثال دیگر: مدل یک درخت تصمیم است و ما اهمیت ویژگی را که به عنوان اولین تقسیم انتخاب شده است تجزیه و تحلیل می‌کنیم. شما ویژگی را حذف کرده و مدل را دوباره آموزش می‌دهید. از آنجایی که ویژگی دیگری به عنوان اولین تقسیم انتخاب شده است، کل درخت می‌تواند بسیار متفاوت باشد، به این معنی که ما نرخ‌های خطای درختان (به طور بالقوه) کاملاً متفاوت را مقایسه می‌کنیم تا تصمیم بگیریم که این ویژگی برای یکی از درختان چقدر مهم است.

۸.۵.۵ معایب

اهمیت ویژگی جایگشت به خطای مدل مرتبط است. این ذاتاً بد نیست، اما در برخی موارد آن چیزی نیست که شما نیاز دارید. در برخی موارد، ممکن است ترجیح دهید بدانید که خروجی مدل برای یک ویژگی چقدر متفاوت است بدون اینکه به معنای عملکرد آن توجه کنید. به عنوان مثال، شما می‌خواهید بفهمید که وقتی شخصی ویژگی‌ها را دستکاری می‌کند، خروجی مدل شما چقدر قوی است. در این مورد، شما علاقه‌ای به این خواهید داشت که با تغییر یک ویژگی، عملکرد مدل چقدر کاهش می‌یابد، بلکه چقدر از واریانس خروجی مدل توسط هر ویژگی توضیح داده می‌شود. واریانس مدل (توضیح داده شده توسط ویژگی‌ها) و اهمیت ویژگی زمانی که مدل به خوبی تعمیم می‌یابد (یعنی بیش برآش وجود نداشته باشد) به شدت با هم همبستگی دارند. شما نیاز به دسترسی به نتیجه واقعی دارید. اگر کسی فقط مدل و داده‌های بدون برچسب را در اختیار شما قرار دهد - نه نتیجه واقعی را - نمی‌توانید اهمیت ویژگی جایگشتی را محاسبه کنید.

اهمیت ویژگی جایگشت بستگی به نحوه به هم زدن ویژگی دارد که تصادفی بودن را به اندازه گیری اضافه می‌کند. هنگامی که جایگشت تکرار می‌شود، نتایج ممکن است بسیار متفاوت باشد. تکرار جایگشت و میانگین معیارهای اهمیت نسبت به تکارها، اندازه گیری را تثبیت می‌کند، اما زمان محاسبه را افزایش می‌دهد. اگر ویژگی‌ها با هم همبستگی داشته باشند، اهمیت ویژگی جایگشت می‌تواند توسط نمونه‌های داده غیرواقعی سوگیری داشته باشد. مشکل همانند نمودارهای وابستگی جزئی است: جایگشت ویژگی‌ها در هنگام همبستگی دو یا چند ویژگی، نمونه داده‌های بعيد ایجاد می‌کند. وقتی همبستگی مثبت دارند (مانند قد و وزن یک فرد) و من یکی از ویژگی‌ها را به هم می‌زنم، نمونه‌های جدیدی ایجاد می‌کنم که بعيد یا حتی از نظر فیزیکی غیرممکن است (مثالاً فرد ۲ متری با وزن ۳۰ کیلوگرم)، اما از این نمونه‌های جدید برای اندازه گیری اهمیت استفاده می‌کنم. به عبارت دیگر، برای اهمیت ویژگی جایگشتی یک ویژگی همبسته، ما در نظر می‌گیریم که وقتی ویژگی را با مقادیری که هرگز در واقعیت مشاهده نمی‌کنیم مبادله می‌کنیم، عملکرد مدل چقدر کاهش می‌یابد. بررسی کنید که آیا ویژگی‌ها به شدت همبستگی دارند و در صورت وجود، در مورد تفسیر اهمیت ویژگی دقت کنید. با این حال، همبستگی‌های زوجی^۱ ممکن است برای آشکار کردن مشکل کافی نباشد. نکته دشوار دیگر: افزودن یک ویژگی همبسته می‌تواند اهمیت ویژگی مرتبط را کاهش دهد با تقسیم اهمیت بین هر دو ویژگی. اجازه دهید مثالی از منظورم از «تقسیم کردن» اهمیت ویژگی به شما بگویم: ما می‌خواهیم احتمال باران را پیش‌بینی کنیم و از دمای ساعت ۸ صبح روز قبل به عنوان یک ویژگی همراه با سایر ویژگی‌های نامرتب استفاده کنیم. من یک جنگل تصادفی آموزش می‌دهم و معلوم می‌شود که دما مهم‌ترین ویژگی است و همه چیز خوب است و شب بعد خوب می‌خوابم. حال سناریوی دیگر را تصور کنید که در آن دمای ۹:۰۰ صبح را به عنوان یک ویژگی که به شدت با دمای ساعت ۸:۰۰ صبح مرتبط است، لحاظ کنم. دمای ساعت ۹:۰۰ صبح اگر از قبل دمای ساعت ۸:۰۰ صبح را بدانم، اطلاعات بیشتری به من نمی‌دهد. اما داشتن ویژگی‌های بیشتر همیشه خوب است، درست است؟ من یک جنگل تصادفی را با دو ویژگی دما و ویژگی‌های نامرتب آموزش می‌دهم. برخی از درختان در جنگل تصادفی دمای ۸ صبح را می‌گیرند، برخی دیگر دمای ۹:۰۰ صبح، برخی دیگر هر دو و برخی دیگر هیچ کدام. دو ویژگی دما در کنار هم کمی اهمیت بیشتری نسبت به ویژگی دمای واحد قبلی دارند، اما به جای قرار گرفتن در بالای لیست ویژگی‌های مهم، هر دما اکنون جایی در وسط است. با معرفی یک ویژگی همبسته، مهم‌ترین ویژگی را از بالای نرdban اهمیت به حد متوسط رساندم. از یک طرف این خوب است، زیرا به سادگی رفتار مدل یادگیری ماشین زیربنایی، در اینجا جنگل تصادفی را منعکس می‌کند. دمای ۸:۰۰ صبح به سادگی از اهمیت کمتری برخوردار شده است زیرا مدل اکنون می‌تواند به اندازه گیری ۹:۰۰ صبح نیز تکیه کند. از سوی دیگر، تفسیر اهمیت ویژگی را به میزان قابل توجهی دشوارتر

¹ pairwise correlations

می‌کند. تصور کنید می‌خواهید ویژگی‌ها را برای خطاهای اندازه گیری بررسی کنید. چک گران است و شما تصمیم می‌گیرید فقط ۳ مورد از مهم‌ترین ویژگی‌ها را بررسی کنید. در مورد اول دما را بررسی می‌کنید، در مورد دوم هیچ ویژگی دما را فقط به این دلیل که آنها اکنون اهمیت را به اشتراک می‌گذارند درج نمی‌کنند. حتی اگر مقادیر اهمیت ممکن است در سطح رفتار مدل معنا پیدا کند، اگر ویژگی‌های همبسته داشته باشد گیج کننده است.

۸.۵.۶ گزینه‌های جایگزین

الگوریتمی به نام PIMP (Altmann et al., 2010) الگوریتم اهمیت ویژگی جایگشت را برای محاسبه p-value برای اهمیت‌ها به روزرسانی می‌کند. یکی دیگر از جایگزین‌های مبتنی بر خطا، حذف ویژگی از داده‌های آموزشی، آموزش مجدد مدل و اندازه‌گیری افزایش خطاست. جایگشت یک ویژگی و اندازه گیری افزایش خطا تنها راه برای اندازه گیری اهمیت یک ویژگی نیست. معیارهای مختلف اهمیت را می‌توان به روش‌های خاص مدل و آگنوستیک مدل تقسیم کرد. اهمیت جینی^۱ برای جنگل‌های تصادفی یا ضرایب رگرسیون استاندارد برای مدل‌های رگرسیون نمونه‌هایی از معیارهای اهمیت ویژه مدل هستند.

یک جایگزین آگنوستیک مدل برای اهمیت ویژگی جایگشت، معیارهای مبتنی بر واریانس هستند. معیارهای اهمیت ویژگی مبتنی بر واریانس مانند شاخص‌های Sobol یا ANOVA تابعی به ویژگی‌هایی که باعث واریانس بالایی در تابع پیش‌بینی می‌شوند اهمیت بیشتری می‌دهند. همچنین اهمیت SHAP شباهت‌هایی به اندازه گیری اهمیت مبتنی بر واریانس دارد. اگر تغییر یک ویژگی، خروجی را تا حد زیادی تغییر دهد، مهم است. این تعریف اهمیت با تعریف مبتنی بر خطا در مورد اهمیت ویژگی جایگشت متفاوت است. این در مواردی مشهود است که یک مدل بیش برازش داده شده باشد. اگر مدلی بیش از حد برازش می‌کند و از ویژگی غیرمرتبه با خروجی استفاده می‌کند، اهمیت ویژگی جایگشتی اهمیتی برابر با صفر می‌دهد زیرا این ویژگی به تولید پیش‌بینی‌های صحیح کمک نمی‌کند. از سوی دیگر، اندازه‌گیری اهمیت مبتنی بر واریانس، ممکن است به ویژگی اهمیت بالایی بددهد زیرا زمانی که ویژگی تغییر می‌کند، پیش‌بینی می‌تواند تغییرات زیادی داشته باشد. نمای کلی خوبی از تکنیک‌های مختلف اهمیت در مقاله توسط Wei et al. (۲۰۱۵) ارائه شده است.

۸.۵.۷ نرم افزار

پکیج iml نرم افزار R برای مثال‌ها استفاده شد. پکیج‌های DALEX و vip نرم افزار R همچنین کتابخانه scikit-learn و rfpimp اهمیت ویژگی جایگشتی آگنوستیک مدل را پیاده‌سازی می‌کنند.

^۱ Gini importance

۸.۶ جایگزین کلی

یک مدل جایگزین کلی یک مدل قابل تفسیر است که برای تقریب پیش‌بینی‌های یک مدل جعبه سیاه آموزش داده شده است. ما می‌توانیم با تفسیر مدل جایگزین در مورد مدل جعبه سیاه نتیجه گیری کنیم. حل تفسیر پذیری یادگیری ماشین با استفاده از یادگیری ماشین بیشتر!

۸.۶.۱ تئوری

مدل‌های جایگزین در مهندسی نیز استفاده می‌شوند: اگر یک نتیجه مورد علاقه گران، زمان‌بر یا اندازه‌گیری آن دشوار باشد (مثلاً به دلیل اینکه از یک شبیه‌سازی رایانه‌ای پیچیده می‌آید)، می‌توان به جای آن از یک مدل جایگزین ارزان و سریع نتیجه استفاده کرد. تفاوت بین مدل‌های جایگزین مورد استفاده در مهندسی و یادگیری ماشین قابل تفسیر در این است که مدل زیربنایی یک مدل یادگیری ماشین است (نه شبیه‌سازی) و اینکه مدل جایگزین باید قابل تفسیر باشد. هدف از مدل‌های جایگزین (قابل تفسیر) تقریب پیش‌بینی‌های مدل زیربنایی تا حد امکان دقیق و در عین حال قابل تفسیر بودن است. ایده مدل‌های جایگزین را می‌توان با نام‌های مختلفی یافت: مدل تقریبی^۱، متمامدل^۲، مدل سطح پاسخ^۳، شبیه‌ساز^۴، ...

درباره تئوری: در واقع برای درک مدل‌های جایگزین به نظریه زیادی نیاز نیست. ما می‌خواهیم تابع پیش‌بینی جعبه سیاه f را تا حد امکان با تابع پیش‌بینی مدل جایگزین g ، تحت این محدودیت که g قابل تفسیر است، تقریب بزنیم. برای تابع g می‌توان از هر مدل قابل تفسیر - به عنوان مثال از فصل مدل‌های قابل تفسیر - استفاده کرد.

به عنوان مثال یک مدل خطی:

$$g(x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

یا درخت تصمیم:

$$g(x) = \sum_{m=1}^M c_m I\{x \in R_m\}$$

آموزش یک مدل جایگزین یک روش آگنوستیک مدل است، زیرا به هیچ اطلاعاتی در مورد عملکرد درونی مدل جعبه سیاه نیاز ندارد، فقط دسترسی به داده‌ها و عملکرد پیش‌بینی ضروری است. اگر مدل یادگیری ماشین زیربنایی با مدل دیگری جایگزین شد، همچنان می‌توانید از روش جایگزین استفاده کنید. انتخاب نوع مدل جعبه سیاه و نوع مدل جایگزین جدا شده است.

¹ Approximation model

² metamodel

³ response surface model

⁴ emulator

برای به دست آوردن مدل جایگزین مراحل زیر را انجام دهید:

- ۱- یک مجموعه‌داده X را انتخاب کنید. این مجموعه می‌تواند همان مجموعه‌داده ای باشد که برای آموزش مدل جعبه سیاه استفاده شده است یا یک مجموعه‌داده جدید از همان توزیع باشد. حتی می‌توانید بسته به برنامه خود زیر مجموعه‌ای از داده‌ها یا شبکه ای از نقاط را انتخاب کنید.
- ۲- برای مجموعه‌داده انتخابی X , پیش‌بینی‌های مدل جعبه سیاه را دریافت کنید.
- ۳- یک نوع مدل قابل تفسیر (مدل خطی، درخت تصمیم، و ...) را انتخاب کنید.
- ۴- مدل قابل تفسیر را بر روی مجموعه‌داده X و پیش‌بینی‌های آن آموزش دهید.
- ۵- تبریک می‌گوییم! شما اکنون یک مدل جایگزین دارید.
- ۶- اندازه گیری کنید که مدل جایگزین چقدر پیش‌بینی‌های مدل جعبه سیاه را تکرار می‌کند.
- ۷- مدل جایگزین را تفسیر کنید.

ممکن است روش‌هایی برای مدل‌های جایگزین پیدا کنید که مراحل اضافی دارند یا کمی متفاوت هستند، اما ایده کلی معمولاً همان‌طور که در اینجا توضیح داده شده است.

یکی از راه‌های اندازه‌گیری اینکه جایگزین چقدر مدل جعبه سیاه را تکرار می‌کند، اندازه‌گیری R-squared است:

$$R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{\sum_{i=1}^n (\hat{y}_*^{(i)} - \hat{y}^{(i)})^2}{\sum_{i=1}^n (\hat{y}^{(i)} - \bar{y})^2}$$

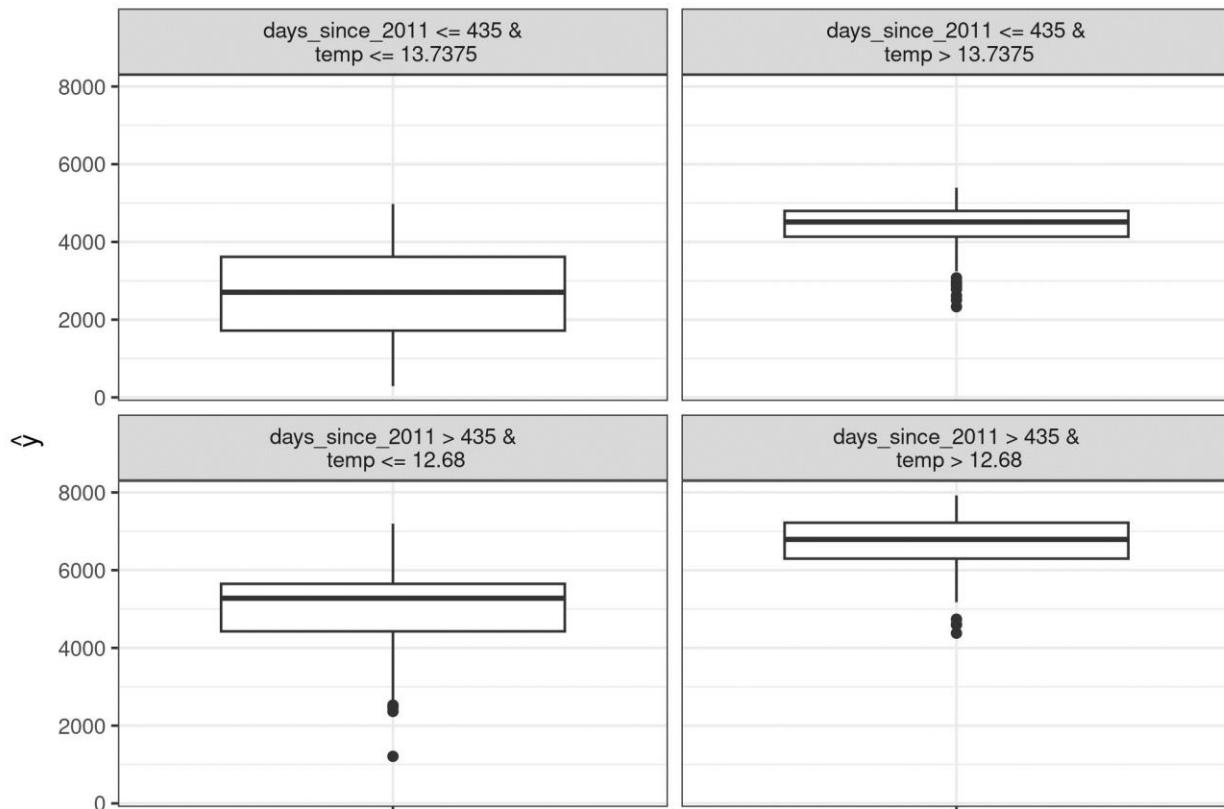
در این رابطه \hat{y}_* پیش‌بینی مدل جایگزین برای نمونه- i است، $\hat{y}^{(i)}$ پیش‌بینی مدل جعبه سیاه و \bar{y} میانگین پیش‌بینی‌های مدل جعبه سیاه است. SSE مخفف مجموع مربعات خطا و SST مخفف مجموع مربعات کل است. معیار R-squared را می‌توان به عنوان درصد واریانسی که توسط مدل جانشین توضیح داده شده است تفسیر کرد. اگر R-squared نزدیک به ۱ (۱ کم) باشد، مدل قابل تفسیر رفتار مدل جعبه سیاه را به خوبی تقریب می‌کند. اگر مدل قابل تفسیر بسیار نزدیک با مدل جعبه سیاه باشد، ممکن است بخواهید مدل پیچیده را با مدل قابل تفسیر جایگزین کنید. اگر R-squared نزدیک به ۰ (۰ زیاد) باشد، مدل قابل تفسیر نمی‌تواند مدل جعبه سیاه را توضیح دهد.

توجه داشته باشید که ما در مورد عملکرد مدل جعبه سیاه زیربنایی صحبت نکرده ایم، یعنی اینکه چقدر خوب یا بد در پیش‌بینی خروجی واقعی عمل می‌کند. عملکرد مدل جعبه سیاه نقشی در آموزش مدل جانشین ندارد. تفسیر مدل جانشین همچنان معتبر است زیرا در مورد مدل اظهاراتی می‌کند نه در مورد دنیای واقعی. اما مسلماً اگر مدل جعبه سیاه بد باشد، تفسیر مدل جایگزین بی‌ربط می‌شود، زیرا در این صورت مدل جعبه سیاه خود بی‌ربط است.

همچنین می‌توانیم یک مدل جایگزین براساس زیرمجموعه‌ای از داده‌های اصلی بسازیم یا نمونه‌ها را دوباره وزن دهی کنیم. به این ترتیب، توزیع ورودی مدل جایگزین را تغییر می‌دهیم، که تمرکز تفسیر را تغییر می‌دهد (پس دیگر واقعاً جهانی نیست). اگر داده‌ها را به صورت محلی با یک نمونه خاص از داده‌ها وزن دهیم (هر چه نمونه‌ها به نمونه انتخاب شده نزدیک‌تر باشند، وزن آنها بیشتر است)، یک مدل جایگزین محلی دریافت می‌کنیم که می‌تواند پیش‌بینی فردی نمونه را توضیح دهد. در فصل بعدی در مورد مدل‌های محلی بیشتر خواهید خواند.

۸.۶.۲ مثال

برای نشان دادن مدل‌های جایگزین، یک رگرسیون و یک مثال طبقه‌بندی را در نظر می‌گیریم. ابتدا، ما یک ماشین بردار پشتیبان را آموزش می‌دهیم تا تعداد دوچرخه‌های اجاره شده روزانه را با توجه به اطلاعات آب و هوا و تقویم پیش‌بینی کند. ماشین بردار پشتیبان خیلی قابل تفسیر نیست، بنابراین ما یک جایگزین را با یک درخت تصمیم CART به عنوان مدل قابل تفسیر برای تقریب رفتار ماشین بردار پشتیبان آموزش می‌دهیم.

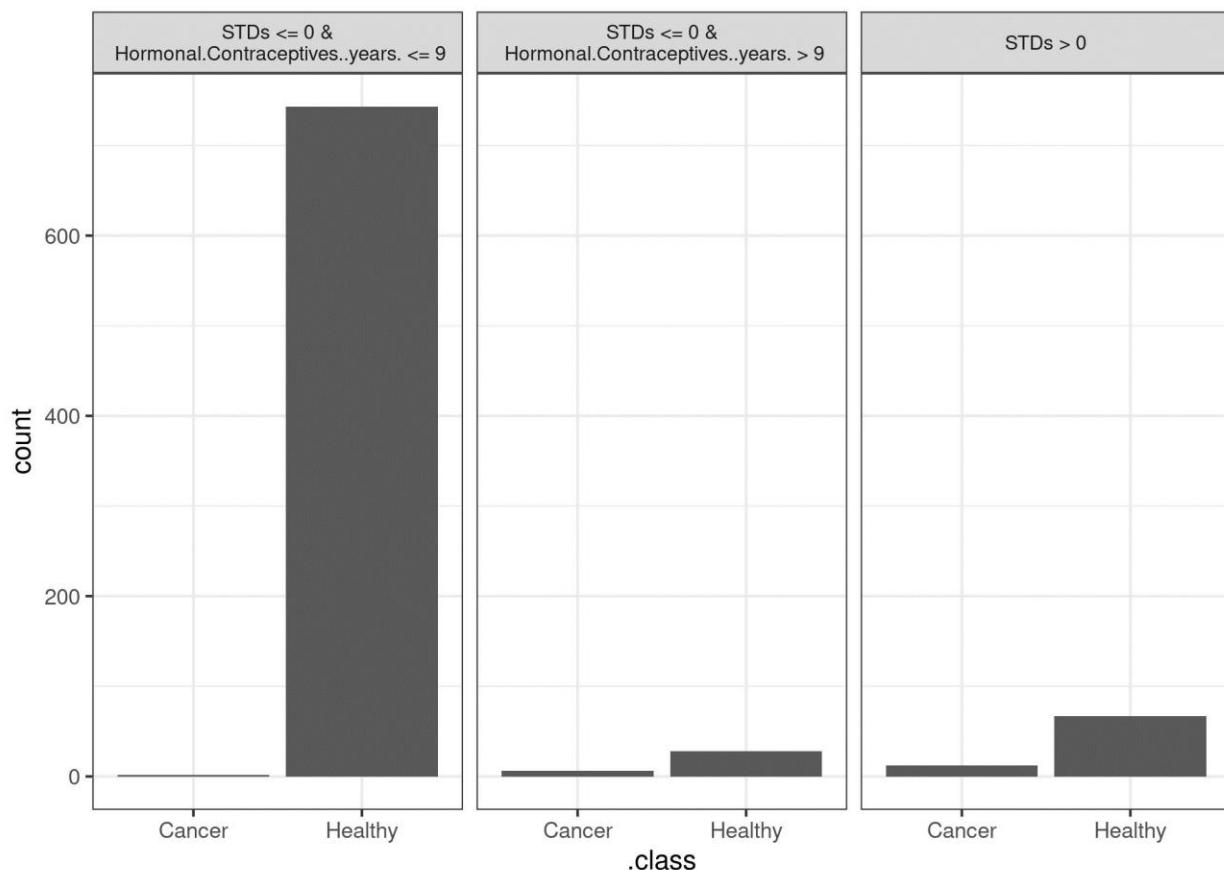


شکل ۸.۲۸: گره‌های پایانی یک درخت جایگزین که پیش‌بینی‌های یک ماشین بردار پشتیبان آموزش دیده بر روی مجموعه داده‌های اجاره دوچرخه را تقریب می‌زنند. توزیع‌ها در گره‌ها نشان می‌دهد که درخت جایگزین

تعداد بیشتری از دوچرخه‌های اجاره‌ای را زمانی که دما بالای ۱۳ درجه سانتی‌گراد است و زمانی که روز بعد در دوره ۲ ساله بود (نقطه برش در ۴۳۵ روز) پیش‌بینی می‌کند.

مدل جایگزین دارای یک R-squared (واریانس توضیح داده شده) ۰/۷۷ است که به این معنی است که رفتار جعبه سیاه زیرین را کاملاً خوب تقریب می‌کند، اما نه کاملاً. اگر برآش کامل بود، می‌توانیم دستگاه بردار پشتیبانی را دور بیندازیم و به جای آن از درخت استفاده کنیم.

در مثال دوم، احتمال سرطان دهانه رحم را با یک جنگل تصادفی پیش‌بینی می‌کنیم. دوباره یک درخت تصمیم را با مجموعه‌داده اصلی آموزش می‌دهیم، اما با پیش‌بینی جنگل تصادفی به عنوان نتیجه، به جای کلاس‌های واقعی (سالم در مقابل سرطان) از داده‌ها.



شکل ۸.۲۹: گره‌های انتهایی یک درخت جایگزین که پیش‌بینی‌های یک جنگل تصادفی آموزش‌دیده بر روی مجموعه‌داده سرطان دهانه رحم را تقریب می‌زنند. تعداد در گره‌ها، فراوانی طبقه‌بندی مدل‌های جعبه سیاه را در گره‌ها نشان می‌دهد.

مدل جایگزین دارای یک R-squared (واریانس توضیح داده شده) ۰/۱۹ است، به این معنی که به خوبی جنگل تصادفی را تقریب نمی‌کند و ما نباید درخت را در هنگام نتیجه‌گیری در مورد مدل پیچیده تفسیر کنیم.

۸.۶.۳ مزايا

روش مدل جايگزين انعطاف‌پذير است: هر مدلی از بخش مدل‌های قابل‌تفسیر قابل استفاده است. اين همچنین به اين معنى است که شما می‌توانيد نه تنها مدل قابل‌تفسیر، بلکه مدل جعبه سياه زيرين را نيز مبادله کنيد. فرض کنيد مدل پيچide اى ايجاد کرده ايد و آن را برای تيم‌های مختلف شركت خود توضيح می‌دهيد. يك تيم با مدل‌های خطی آشنا است، تيم دیگر می‌تواند درخت تصميم را درک کند. شما می‌توانيد دو مدل جايگزين (مدل خطی و درخت تصميم) را برای مدل جعبه سياه اصلی آموزش دهيد و دو نوع توضيح ارائه دهيد. اگر مدل جعبه سياه با عملکرد بهتری پيدا کرديد، لازم نیست روش تفسير خود را تغيير دهيد، زيرا می‌توانيد از همان کلاس مدل‌های جايگزين استفاده کنيد.

من استدلال می‌کنم که رویکرد بسيار شهودی و سرراست است. اين بدان معناست که پياده‌سازی آن آسان است، اما توضيح آن برای افرادي که با علم داده یا يادگيري ماشين آشناي ندارند نيز آسان است.

با اندازه گيري **R-squared**، می‌توانيم به راحتی اندازه گيري کنيم که مدل‌های جايگزين ما در تقریب پيش‌بینی‌های جعبه سياه چقدر خوب هستند.

۸.۶.۴ معایب

شما باید آگاه باشید که در مورد مدل نتيجه می‌گيريد نه در مورد داده‌ها، زира مدل جايگزين هرگز نتيجه واقعی را نمی‌بیند.

مشخص نیست بهترین نقطه برش برای **R-squared** چيست تا مطمئن شویم که مدل جايگzin به اندازه کافی به مدل جعبه سياه نزديک است. ۸۰ درصد واريанс توضيح داده شده است؟ ۵۰ درصد؟ ۹۹ درصد؟ ما می‌توانيم اندازه گيري کنيم که مدل جايگzin چقدر به مدل جعبه سياه نزديک است. بيايد فرض کنيم خيلي نزديک نیستيم، اما به اندازه کافی نزديک هستيم. ممکن است اين اتفاق بيفتد که مدل قابل‌تفسیر برای يك زير مجموعه از مجموعه‌داده بسيار نزديک باشد، اما برای زيرمجموعه دیگر کاملاً متفاوت باشد. در اين مورد، تفسير مدل ساده برای همه نقاط داده به يك اندازه خوب نخواهد بود.

مدل قابل تفسيری که شما به عنوان جايگzin انتخاب می‌کنيد با تمام مزايا و معایب خود همراه است. برخی افراد استدلال می‌کنند که به طور کلي، هیچ مدل ذاتی قابل‌تفسیر (از جمله مدل‌های خطی و درخت‌های تصميم گيري) وجود ندارد و حتی داشتن توهם تفسيرپذيری خطرناک است. اگر شما هم با اين نظر موافق هستيد، مطمئناً اين روش برای شما مناسب نیست.

۸.۶.۵ نرم افزار

من از پکیج `iml` نرم افزار R برای مثال‌ها استفاده کردم. اگر می‌توانید یک مدل یادگیری ماشین آموزش دهید، باید خودتان بتوانید مدل‌های جایگزین را پیاده‌سازی کنید. به سادگی یک مدل قابل تفسیر را برای پیش‌بینی پیش‌بینی‌های مدل جعبه سیاه آموزش دهید.

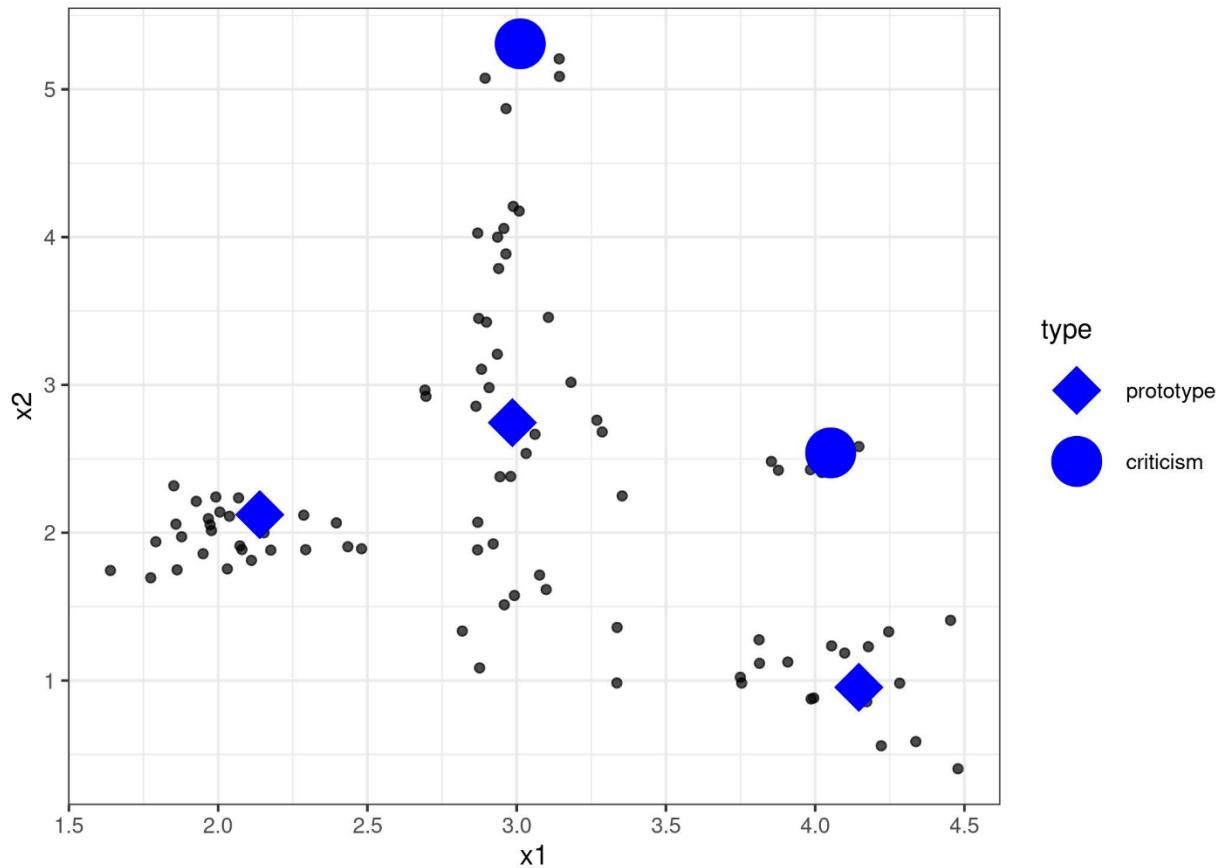
۸.۷ نمونه‌های اولیه^۱ و انتقادات^۲

نمونه اولیه یک نمونه داده است که نماینده همه داده‌ها است. انتقاد یک نمونه داده است که به خوبی توسط مجموعه نمونه‌های اولیه نمایندگی نمی‌شود. هدف از انتقاد، ارائه بینش همراه با نمونه‌های اولیه، به ویژه برای نقاط داده‌ای است که نمونه‌های اولیه به خوبی نمایندگی نمی‌کنند. نمونه‌های اولیه و انتقادات را می‌توان به طور مستقل از یک مدل یادگیری ماشین برای توصیف داده‌ها استفاده کرد، اما همچنین می‌توان از آنها برای ایجاد یک مدل قابل تفسیر یا برای تفسیرپذیر ساختن مدل جعبه سیاه استفاده کرد.

در این فصل از عبارت « نقطه داده » برای اشاره به یک نمونه استفاده می‌کنم، تا بر این تفسیر تأکید کنم که یک نمونه همچنین نقطه‌ای در یک سیستم مختصات است که در آن هر ویژگی یک بعد است. شکل زیر توزیع داده‌های شبیه سازی شده را نشان می‌دهد که برخی از نمونه‌ها به عنوان نمونه اولیه و برخی به عنوان انتقاد انتخاب شده‌اند. نقاط کوچک، داده‌ها، نقاط بزرگ، انتقادات و مربع‌های بزرگ نمونه‌های اولیه هستند. نمونه‌های اولیه (به صورت دستی) برای پوشش مراکز توزیع داده‌ها انتخاب می‌شوند و انتقادات، نقاطی در یک خوشه بدون نمونه اولیه هستند. نمونه‌های اولیه و انتقادات همیشه نمونه‌های واقعی از داده‌ها هستند.

¹ Prototypes

² Criticisms



شکل ۸.۳۰: نمونه‌های اولیه و انتقادات برای توزیع داده با دو ویژگی x_1 و x_2 من نمونه‌های اولیه را به صورت دستی انتخاب کردم که مقیاس خوبی ندارد و احتمالاً منجر به نتایج ضعیف می‌شود. روش‌های زیادی برای یافتن نمونه‌های اولیه در داده‌ها وجود دارد. یکی از آنها k-medoids یک الگوریتم خوشبندی مرتبط با الگوریتم k-means می‌باشد. هر الگوریتم خوشبندی که نقاط داده واقعی را به عنوان مرکز خوش برمی‌گرداند، واحد شرایط انتخاب نمونه‌های اولیه است. اما اکثر این روش‌ها فقط نمونه‌های اولیه را پیدا می‌کنند، اما هیچ انتقادی ندارند. این فصل MMD-critic ارائه شده توسط al. Kim et (۲۰۱۶) را ارائه می‌کند، رویکردنی که نمونه‌های اولیه و انتقادات را در یک چارچوب واحد ترکیب می‌کند.

MMD-critic توزیع داده‌ها و توزیع نمونه‌های اولیه انتخاب شده را مقایسه می‌کند. این مفهوم اصلی برای درک روش MMD-critic است. MMD-critic نمونه‌های اولیه‌ای را انتخاب می‌کند که اختلاف بین دو توزیع را به حداقل می‌رساند. نقاط داده در مناطق با تراکم بالا نمونه‌های اولیه خوبی هستند، به خصوص زمانی که نقاط از "خوش‌های داده" مختلف انتخاب می‌شوند. نقاط داده از مناطقی که به خوبی توسط نمونه‌های اولیه توضیح داده نشده اند به عنوان انتقاد انتخاب می‌شوند.

اجازه دهید عمیق‌تر به نظریه بپردازیم.

۸.۷.۱ تئوری

روش MMD-critic در سطح بالا را می‌توان به طور خلاصه اینگونه بیان کرد:

۱- تعداد نمونه‌های اولیه و انتقاداتی را که می‌خواهید پیدا کنید انتخاب کنید.

۲- نمونه‌های اولیه را با جستجوی حریصانه پیدا کنید. نمونه‌های اولیه به گونه‌ای انتخاب می‌شوند که توزیع نمونه‌های اولیه به توزیع داده‌ها نزدیک باشد.

۳- انتقادات را با جستجوی حریصانه پیدا کنید. نقاطی به عنوان انتقاد انتخاب می‌شوند که در آن توزیع نمونه‌های اولیه با توزیع داده‌ها متفاوت است.

برای یافتن نمونه‌های اولیه و انتقادات برای مجموعه داده با MMD-critic، به چند عنصر نیاز داریم. به عنوان اساسی‌ترین عنصر، ما به یک تابع هسته^۱ برای تخمین چگالی داده‌ها نیاز داریم. هسته تابعی است که دو نقطه داده را با توجه به مجاورت آنها وزن می‌دهد. بر اساس برآوردهای چگالی، ما به معیاری نیاز داریم که به ما بگوید دو توزیع چقدر متفاوت هستند تا بتوانیم تعیین کنیم که آیا توزیع نمونه‌های اولیه که انتخاب می‌کنیم به توزیع داده نزدیک است یا خیر. این کار با اندازه گیری **حداکثر مغایرت میانگین**^۲ (MMD) حل می‌شود. همچنین بر اساس تابع هسته، به تابع شاهد **witness function** نیاز داریم تا به ما بگوید که دو توزیع در یک نقطه داده خاص چقدر متفاوت هستند. با تابع شاهد، می‌توانیم انتقادات را انتخاب کنیم، یعنی نقاط داده‌ای که در آن توزیع نمونه‌های اولیه و داده‌ها از هم جدا می‌شود و تابع شاهد مقادیر مطلق زیادی به خود می‌گیرد. آخرین عنصر یک استراتژی جستجو برای نمونه‌های اولیه و انتقادات خوب این است که با یک **جستجوی حریصانه**^۳ ساده حل می‌شود.

اجازه دهید با **حداکثر مغایرت میانگین** (MMD) شروع کنیم، که اختلاف بین دو توزیع را اندازه گیری می‌کند. انتخاب نمونه‌های اولیه، توزیع چگالی نمونه‌های اولیه را ایجاد می‌کند. ما می‌خواهیم ارزیابی کنیم که آیا توزیع نمونه اولیه با توزیع داده متفاوت است یا خیر. ما هر دو را با تابع چگالی هسته تخمین می‌زنیم. **حداکثر مغایرت میانگین** تفاوت بین دو توزیع را اندازه گیری می‌کند، که کوچک‌ترین کران بالا^۴ در یک فضای تابعی از تفاوت بین انتظارات بر اساس دو توزیع است. همه چیز روشن است؟ من شخصاً وقتی می‌بینم که چگونه چیزی با داده‌ها محاسبه می‌شود، این مفاهیم را خیلی بهتر درک می‌کنم. فرمول زیر نحوه محاسبه محدود اندازه گیری MMD (MMD²) را نشان می‌دهد:

¹ kernel function

² maximum mean discrepancy

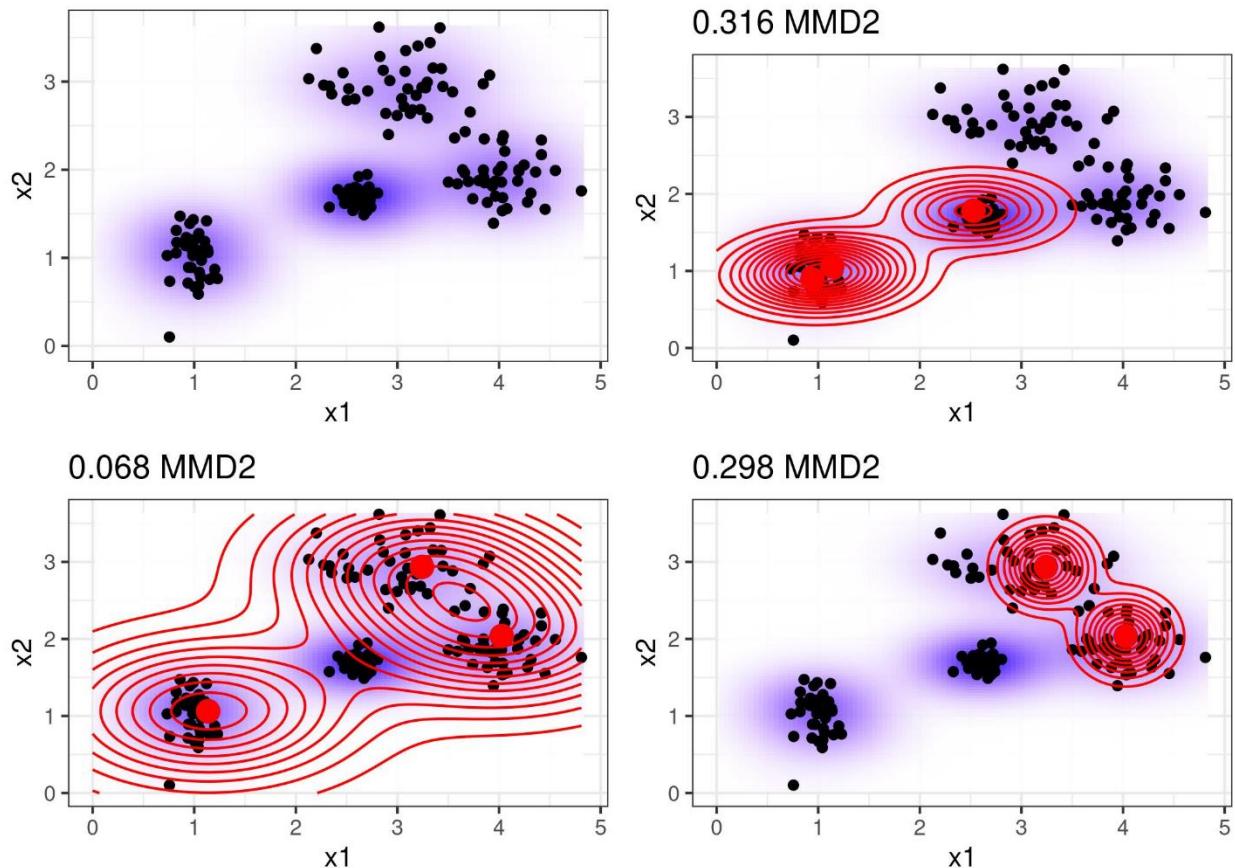
³ greedy search

⁴ supremum

$$MMD^2 = \frac{1}{m^2} \sum_{i,j=1}^m k(z_i \cdot z_j) - \frac{2}{mn} \sum_{i,j=1}^{m,n} k(z_i \cdot x_j) + \frac{1}{n^2} \sum_{i,j=1}^n k(x_i \cdot x_j)$$

K یک تابع هسته است که شباهت دو نقطه را اندازه گیری می کند، اما در ادامه در مورد آن بیشتر توضیح خواهیم داد. m تعداد نمونه های اولیه z و n تعداد نقاط داده x در مجموعه داده اصلی ما است. نمونه های اولیه z مجموعه ای از نقاط داده x هستند. هر نقطه چند بعدی است، یعنی می تواند چندین ویژگی داشته باشد. هدف MMD-critic به حداقل رساندن MMD^2 است. هرچه MMD^2 به صفر نزدیک تر باشد، توزیع نمونه های اولیه بهتر با داده ها مطابقت دارد. کلید به صفر رساندن MMD^2 عبارت وسطی است که میانگین نزدیکی بین نمونه های اولیه و سایر نقاط داده را محاسبه می کند (ضرب در ۲). اگر این عبارت با عبارت اول (میانگین نزدیکی نمونه های اولیه به یکدیگر) به اضافه جمله آخر (میانگین نزدیکی داده ها به یکدیگر اشاره می کند) جمع شود، نمونه های اولیه داده ها را کاملاً توضیح می دهند.

شکل زیر نحوه اندازه گیری MMD^2 را نشان می دهد. نمودار اول نقاط داده را با دو ویژگی نشان می دهد که به موجب آن تخمین چگالی داده ها با پس زمینه سایه دار نمایش داده می شود. هر یک از نمودارهای دیگر انتخاب های مختلفی از نمونه های اولیه را به همراه معیار MMD^2 در عناوین نمودار نشان می دهد. نمونه های اولیه نقاط بزرگ هستند و توزیع آنها به صورت خطوط کانتور نشان داده شده است. انتخاب نمونه های اولیه که بهترین شکل داده ها را در این سناریوها پوشش می دهند (پایین سمت چپ) کمترین مقدار مغایرت را دارد.



شکل ۸.۳۱: محدود اندازه گیری حداقل مغایرت میانگین (MMD^2) برای یک مجموعه داده با دو ویژگی و انتخاب‌های مختلف نمونه‌های اولیه.

یک انتخاب برای هسته، تابع پایه شعاعی است:

$$k(x, x') = \exp(\gamma \|x - x'\|^2)$$

در این رابطه $\|x - x'\|^2$ فاصله اقلیدسی بین دو نقطه و γ یک پارامتر مقیاس بندی است. مقدار هسته با فاصله بین دو نقطه کاهش می‌یابد و بین صفر و یک قرار می‌گیرد: زمانی که دو نقطه بینهایت از هم فاصله دارند، صفر می‌شود. یک زمانی است که دو نقطه یکی باشند.

ما اندازه گیری MMD^2 ، هسته و جستجوی حریصانه را در الگوریتمی برای یافتن نمونه‌های اولیه ترکیب می‌کنیم:

- با یک لیست خالی از نمونه‌های اولیه شروع کنید.
- در حالی که تعداد نمونه‌های اولیه کمتر از عدد انتخابی m است:

- برای هر نقطه از مجموعه داده، بررسی کنید که با افزودن نقطه به لیست نمونه های اولیه، چقدر کاهش می یابد. نقطه داده ای را که MMD^2 را به حداقل می رساند به لیست اضافه کنید.

- لیست نمونه های اولیه را برگردانید.

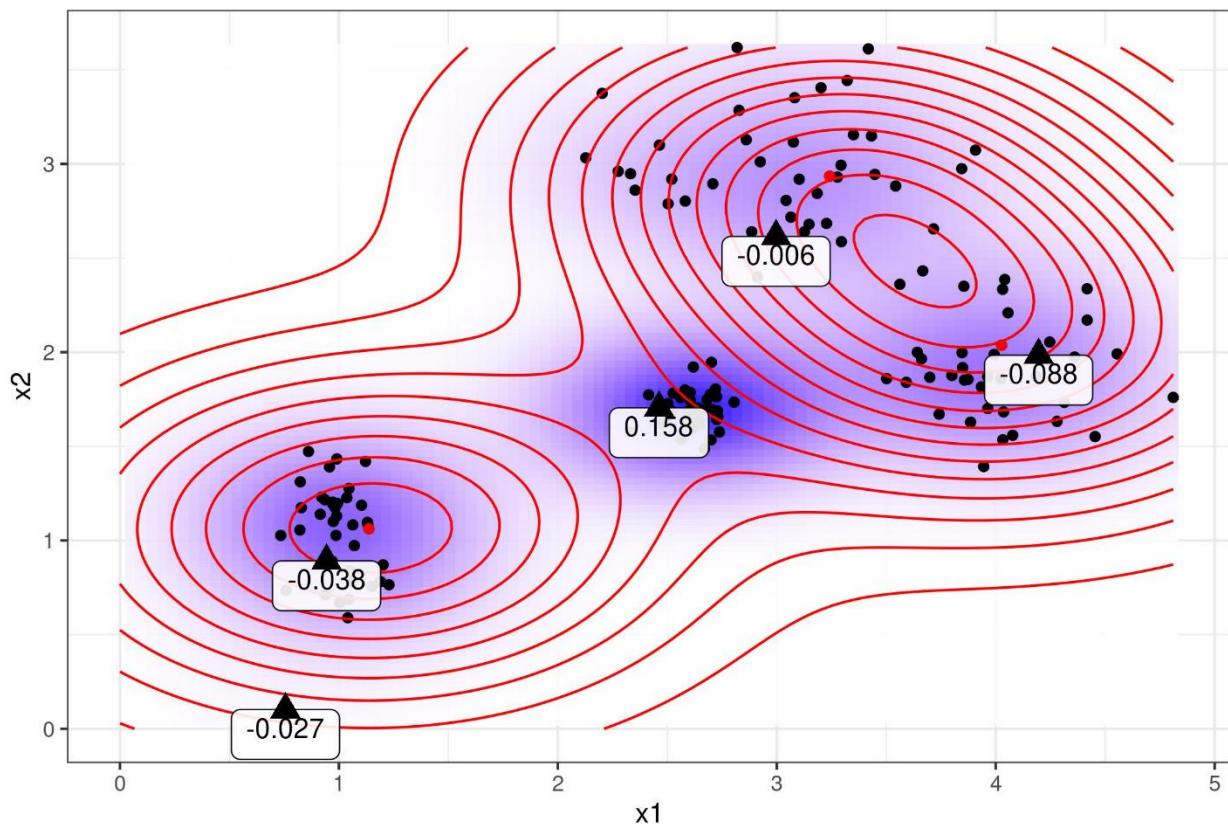
عنصر باقیمانده برای یافتن انتقادات تابع شاهد است که به ما می گوید چقدر دو تخمین چگالی در یک نقطه خاص تفاوت دارند. می توان با استفاده از:

$$witness(x) = \frac{1}{n} \sum_{i=1}^n k(x, x_i) - \frac{1}{m} \sum_{j=1}^m k(x, z_j)$$

برای دو مجموعه داده (با ویژگی های یکسان)، تابع شاهد به شما ابزاری برای ارزیابی اینکه در کدام توزیع تجربی نقطه x بهتر برازنده است، می دهد. برای یافتن انتقادات، ما به دنبال مقادیر افراطی عملکرد شاهد در دو جهت منفی و مثبت هستیم. عبارت اول در تابع شاهد میانگین نزدیکی بین نقطه x و داده ها و عبارت دوم به ترتیب میانگین نزدیکی بین نقطه x و نمونه های اولیه است. اگر تابع شاهد برای نقطه x نزدیک به صفر باشد، تابع چگالی داده ها و نمونه های اولیه به هم نزدیک هستند، به این معنی که توزیع نمونه های اولیه شبیه توزیع داده ها در نقطه x است. تابع شاهد منفی در نقطه x به این معنی است که توزیع نمونه اولیه توزیع داده ها را بیش از حد تخمین می زند (به عنوان مثال اگر نمونه اولیه را انتخاب کنیم اما تنها چند نقطه داده در این نزدیکی وجود دارد). تابع شاهد مثبت در نقطه x به این معنی است که توزیع نمونه اولیه توزیع داده را دست کم می گیرد (به عنوان مثال اگر نقاط داده زیادی در اطراف x وجود داشته باشد اما ما هیچ نمونه اولیه ای را در این نزدیکی انتخاب نکرده باشیم).

برای اینکه شهود بیشتری به شما بدهیم، اجازه دهید از نمونه های اولیه نمودار با کمترین MMD^2 مجدداً استفاده کنیم و عملکرد شاهد را برای چند نقطه انتخاب شده به صورت دستی نمایش دهیم. برچسب های نمودار زیر مقدار تابع شاهد را برای نقاط مختلفی که به صورت مثلث مشخص شده اند نشان می دهد. فقط نقطه وسط ارزش مطلق بالایی دارد و بنابراین کاندیدی خوبی برای انتقاد است.

0.068 MMD²



شکل ۸.۳۲: ارزیابی عملکرد شاهد در نقاط مختلف.

تابع شاهد به ما این امکان را می‌دهد که به طور صریح نمونه‌های داده‌ای را جستجو کیم که به خوبی توسط نمونه‌های اولیه نمایش داده نمی‌شوند. نقدمها نقاطی هستند که ارزش مطلق بالایی در تابع شاهد دارند. مانند نمونه‌های اولیه، انتقادات نیز از طریق جستجوی حریصانه یافت می‌شود. اما به جای کاهش کلی MMD^2 ، ما به دنبال نقاطی هستیم که یک تابع هزینه را که شامل تابع شاهد و یک عبارت تنظیم کننده است، به حداقل می‌رساند. عبارت اضافی در تابع بهینه‌سازی، تنوع در نقاط را اعمال می‌کند، که لازم است تا نقاط از خوش‌های مختلف آمده باشند.

این مرحله دوم مستقل از نحوه یافتن نمونه‌های اولیه است. همچنین می‌توانستم چند نمونه اولیه را انتخاب کنم و از روشی که در اینجا توضیح داده شد برای یادگیری انتقادات استفاده کنم. یا نمونه‌های اولیه می‌توانند از هر روش خوشبندی مانند k-medoids حاصل شوند.

این در مورد بخش‌های مهم نظریه MMD-critic است. یک سوال باقی می‌ماند: چگونه می‌توان از critic برای یادگیری ماشین قابل تفسیر استفاده کرد؟

MMD-critic می‌تواند به سه طریق قابلیت تفسیر را اضافه کند: با کمک به درک بهتر توزیع داده‌ها. با ساخت یک مدل قابل تفسیر؛ با ساختن یک مدل جعبه سیاه قابل تفسیر.

اگر MMD-critic را روی داده‌های خود اعمال کنید تا نمونه‌های اولیه و انتقادات را بیابید، درک شما از داده‌ها را بهبود می‌بخشد، به خصوص اگر توزیع داده‌ای پیچیده با موارد لبه (داده پرت) داشته باشید. اما با critic می‌توانید به چیزهای بیشتری دست پیدا کنید!

برای مثال، می‌توانید یک مدل پیش‌بینی قابل تفسیر ایجاد کنید: یک مدل به اصطلاح «نزدیک‌ترین نمونه اولیه^۱».تابع پیش‌بینی به صورت زیر تعریف می‌شود:

$$\hat{f}(x) = \operatorname{argmax}_{i \in S} k(x, x_i)$$

به این معنی که نمونه اولیه x را از مجموعه نمونه‌های اولیه S انتخاب می‌کنیم که به نقطه داده جدید نزدیک است، به این معنا که بالاترین مقدار تابع هسته را به دست می‌دهد. خود نمونه اولیه به عنوان توضیحی برای پیش‌بینی بازگردانده می‌شود. این روش دارای سه پارامتر تنظیم است: نوع هسته، پارامتر مقیاس بندی هسته و تعداد نمونه‌های اولیه. تمام پارامترها را می‌توان در یک حلقه اعتبار سنجی متقطع بهینه کرد. در این رویکرد از انتقادها استفاده نمی‌شود.

به عنوان گرینه سوم، می‌توانیم از MMD-critic استفاده کنیم تا با بررسی نمونه‌های اولیه و انتقادات همراه با پیش‌بینی‌های مدل، هر مدل یادگیری ماشین را در سطح کلی قابل توضیح کنیم. روند کار به صورت زیر است:

۱- نمونه‌های اولیه و انتقادات را با MMD-critic بیابید.

۲- یک مدل یادگیری ماشین را طبق معمول آموزش دهید.

۳- پیش‌بینی نتایج برای نمونه‌های اولیه و انتقادات با مدل یادگیری ماشین.

۴- تجزیه و تحلیل پیش‌بینی‌ها: در چه مواردی الگوریتم اشتباه بود؟ اکنون تعدادی مثال دارید که داده‌ها را به خوبی نشان می‌دهد و به شما کمک می‌کند تا نقاط ضعف مدل یادگیری ماشین را پیدا کنید.

چگونه کمک می‌کند؟ زمانی را به خاطر دارید که طبقه‌بندی کننده تصویر گوگل، سیاه‌پوستان را به عنوان گوریل شناسایی کرد؟ شاید آنها باید قبل از استقرار مدل تشخیص تصویر خود از روشی که در اینجا توضیح داده شده استفاده می‌کردند. فقط بررسی عملکرد مدل کافی نیست، زیرا اگر ۹۹٪ درست بود، این موضوع همچنان می‌تواند در ۱٪ باشد. و برچسب‌ها نیز ممکن است اشتباه باشند! بررسی همه داده‌های آموزشی و انجام یک بررسی سلامت عقل در صورت مشکل‌ساز بودن پیش‌بینی ممکن است مشکل را آشکار کند، اما غیرممکن است. اما انتخاب - مثلاً چند هزار - نمونه اولیه و انتقاد امکان‌پذیر است و می‌تواند مشکلی را در داده‌ها نشان دهد: ممکن است نشان داده باشد که کمبود تصاویری از افراد با پوست تیره وجود دارد که نشان دهنده مشکل با تنوع

¹ nearest prototype model

در مجموعه‌داده یا می‌توانست یک یا چند تصویر از یک فرد با پوست تیره را به عنوان نمونه اولیه یا (احتمالاً) به عنوان انتقاد با طبقه‌بندی بدنام "گوریل" نشان دهد. من قول نمی‌دهم که MMD-critic مطمئناً این نوع اشتباها را رهگیری کند، اما این یک بررسی عقلانی خوبی است.

۸.۷.۲ مثال‌ها

مثال زیر از MMD-critic از یک مجموعه‌داده ارقام دست‌نویس استفاده می‌کند. با نگاهی به نمونه‌های اولیه واقعی، ممکن است متوجه شوید که تعداد تصاویر در هر رقم متفاوت است. این به این دلیل است که تعداد ثابتی از نمونه‌های اولیه در کل مجموعه‌داده جستجو شد و نه با تعداد ثابت در هر کلاس. همان‌طور که انتظار می‌رفت، نمونه‌های اولیه روش‌های مختلفی را برای نوشتن ارقام نشان می‌دهند.



شکل ۸.۳۳: نمونه‌های اولیه برای مجموعه‌داده ارقام دست‌نویس.

۸.۷.۳ مزایا

در یک مطالعه کاربری، نویسنده‌گان MMD-critic تصاویری را به شرکت‌کنندگان دادند، که آنها باید به صورت بصری با یکی از دو مجموعه تصاویر مطابقت می‌دادند که هر کدام یکی از دو کلاس را نشان می‌داد (مثلاً دو نژاد سگ). شرکت کنندگان زمانی بهترین عملکرد را داشتند که مجموعه‌ها به جای تصاویر تصادفی یک کلاس، نمونه‌های اولیه و انتقادات را نشان دادند.

شما در انتخاب تعداد نمونه اولیه و انتقاد آزاد هستید.

MMD با تخمین چگالی داده‌ها کار می‌کند. این با هر نوع داده و هر نوع مدل یادگیری ماشین کار می‌کند.

پیاده‌سازی الگوریتم آسان است.

MMD-critic در روشی که برای افزایش تفسیرپذیری استفاده می‌شود بسیار انعطاف‌پذیر است. می‌توان از آن برای درک توزیع داده‌های پیچیده استفاده کرد. می‌توان از آن برای ساخت یک مدل یادگیری ماشین قابل تفسیر استفاده کرد. یا می‌تواند تصمیم گیری در مورد مدل یادگیری ماشین جعبه سیاه را روشن کند.

یافتن انتقادات، مستقل از فرآیند انتخاب نمونه‌های اولیه است. اما انتخاب نمونه‌های اولیه بر اساس MMD منطقی است، زیرا در این صورت هم نمونه‌های اولیه و هم انتقادات با استفاده از روش مشابه مقایسه نمونه‌های اولیه و تراکم داده‌ها ایجاد می‌شوند.

۸.۷.۴ معایب

در حالی که، از نظر ریاضی، نمونه‌های اولیه و انتقادات به طور متفاوتی تعریف می‌شوند، تمایز آنها بر اساس یک مقدار برش (تعداد نمونه‌های اولیه) است. فرض کنید تعداد بسیار کمی از نمونه‌های اولیه را برای پوشش توزیع داده انتخاب کرده اید. انتقادات به حوزه‌هایی ختم می‌شود که به خوبی توضیح داده نشده اند. اما اگر بخواهید نمونه‌های اولیه بیشتری را اضافه کنید، آنها نیز در همان مناطق قرار می‌گیرند. هر تفسیری باید در نظر بگیرد که انتقادها به شدت به نمونه‌های اولیه موجود و مقدار قطعی (اختیاری) تعداد نمونه‌های اولیه بستگی دارد.

شما باید تعداد نمونه اولیه و انتقادات را انتخاب کنید. به همان اندازه که این می‌تواند خوب باشد، یک نقطه ضعف نیز محسوب می‌شود. واقعاً به چند نمونه اولیه و انتقاد نیاز داریم؟ هرچی بیشتر بهتر؟ هر چه کمتر بهتر؟ یک راه حل این است که تعداد نمونه‌های اولیه و انتقادات را با اندازه گیری زمان برای کار نگاه‌کردن به تصاویر، که بستگی به کاربرد خاص دارد، انتخاب کنید. تنها زمانی که از MMD-critic برای ساختن یک طبقه‌بندی کننده استفاده می‌کنیم، راهی برای بهینه سازی مستقیم آن داریم. یکی از راه حل‌ها می‌تواند نقشه‌ای باشد که تعداد نمونه‌های اولیه را در محور x و اندازه گیری² MMD در محور y نشان می‌دهد. ما تعداد نمونه‌های اولیه را انتخاب می‌کنیم که منحنی² MMD صاف می‌شود.

پارامترهای دیگر انتخاب کرنل و پارامتر مقیاس بندی هستند. ما مشکل مشابهی با تعداد نمونه‌های اولیه و انتقادات داریم: چگونه یک هسته و پارامتر مقیاس پذیری آن را انتخاب کنیم؟ دوباره، وقتی از MMD-critic به عنوان نزدیک‌ترین طبقه‌بندی کننده نمونه اولیه استفاده می‌کنیم، می‌توانیم پارامترهای هسته را تنظیم

کنیم. با این حال، برای موارد استفاده بدون نظارت از MMD-critic، مشخص نیست. (شاید من در اینجا کمی خشن باشم، زیرا همه روش‌های بدون نظارت این مشکل را دارند).

همه ویژگی‌ها را به عنوان ورودی می‌گیرد، بدون توجه به این واقعیت که برخی از ویژگی‌ها ممکن است برای پیش‌بینی نتیجه مورد علاقه مرتبط نباشند. یک راه حل این است که فقط از ویژگی‌های مرتبط استفاده کنید، برای مثال جاسازی تصویر به جای پیکسل‌های خام. این تا زمانی کار می‌کند که ما راهی برای نمایش نمونه اصلی بر روی نمایشی داشته باشیم که فقط حاوی اطلاعات مرتبط باشد.

تعدادی کد موجود است، اما هنوز به عنوان نرم افزار پکیج شده و مستند به خوبی پیاده‌سازی نشده است.

۸.۷.۵ کد و جایگزین

پیاده‌سازی MMD-critic را می‌توان در مخزن GitHub¹ نویسنده‌گان (Protodash) می‌توان در GitHub¹ نویسنده‌گان (critic) یافت.

اخيراً یک تعمیم از MMD-critic به نام Protodash توسعه داده شده است. نویسنده‌گان در انتشارات خود ادعای مزايايی نسبت به MMD-critic دارند (Gurumoorthy et al., 2019). یک پیاده‌سازی Protodash در ابزار IBM AIX360² موجود است.

ساده‌ترین جایگزین برای یافتن نمونه‌های اولیه، k-medoids ارائه شده توسط Kaufman and Rousseeuw (1987) است.

¹ <https://github.com/BeenKim/MMD-critic>

² <https://github.com/Trusted-AI/AIX360>

فصل ۹ مدل محلی-روش‌های آگنوستیک

روش‌های تفسیر محلی پیش‌بینی‌های فردی را توضیح می‌دهند. در این فصل، با روشهای توضیح محلی زیر آشنا خواهید شد:

- منحنی‌های انتظار شرطی فردی، بلوک‌های سازنده نمودارهای وابستگی جزئی هستند و توضیح می‌دهند که چگونه تغییر یک ویژگی، پیش‌بینی را تغییر می‌دهد.
 - مدل‌های جایگزین محلی (LIME) یک پیش‌بینی را با جایگزینی مدل پیچیده با یک مدل جایگزین قابل تفسیر محلی توضیح می‌دهند.
 - قوانین محدوده (لنگ‌ها) قوانینی هستند که توصیف می‌کنند که کدام مقادیر ویژگی یک پیش‌بینی را ثابت می‌کنند، به این معنا که پیش‌بینی را در جای خود قفل می‌کنند.
 - توضیحات خلاف واقع یک پیش‌بینی را با بررسی اینکه کدام ویژگی برای دستیابی به یک پیش‌بینی مطلوب نیاز به تغییر دارد، توضیح می‌دهد.
 - مقادیر Shapley یک روش انتساب است که پیش‌بینی را به صورت منصفانه به ویژگی‌های فردی اختصاص می‌دهد.
 - SHAP یکی دیگر از روشهای محاسباتی برای مقادیر Shapley است، اما همچنین روشهای تفسیر کلی را بر اساس ترکیبی از مقادیر Shapley در میان داده‌ها پیشنهاد می‌کند.
- مقادیر LIME و Shapley روشهای انتساب هستند، به طوری که پیش‌بینی یک نمونه واحد به عنوان مجموع اثرات ویژگی توصیف می‌شود. روشهای دیگر، مانند توضیحات خلاف واقع، مبتنی بر مثال هستند

۱-۹- انتظار شرطی فردی (ICE)

نمودارهای انتظار شرطی فردی (ICE^۱) یک خط را برای هر نمونه رسم می‌کند و نشان می‌دهد چگونه پیش‌بینی نمونه با تغییر یک ویژگی تغییر می‌کند.

نمودار وابستگی جزئی برای اثر متوسط یک ویژگی یک روش کلی است زیرا بر روی نمونه‌های خاص تمرکز نمی‌کند، بلکه بر میانگین کلی تمرکز می‌کند. معادل یک PDP برای نمونه‌های داده فردی، نمودار انتظار شرطی فردی (ICE) نامیده می‌شود (Goldstein et al., 2015). نمودار ICE وابستگی پیش‌بینی به یک ویژگی را برای هر کدام به تصویر می‌کشد نمونه به طور جداگانه، منجر به یک خط در هر نمونه، در مقایسه با یک خط کلی در نمودارهای وابستگی جزئی است. PDP میانگین خطوط یک نمودار ICE است. مقادیر یک خط (و یک نمونه) را می‌توان با ثابت نگه داشتن سایر ویژگی‌ها، ایجاد انواعی از این نمونه با جایگزینی مقدار ویژگی با مقادیر یک شبکه و پیش‌بینی با مدل جعبه سیاه برای این نمونه‌های جدید محاسبه کرد. نتیجه مجموعه‌ای از نقاط برای نمونه با مقدار ویژگی از شبکه و پیش‌بینی‌های مربوطه است.

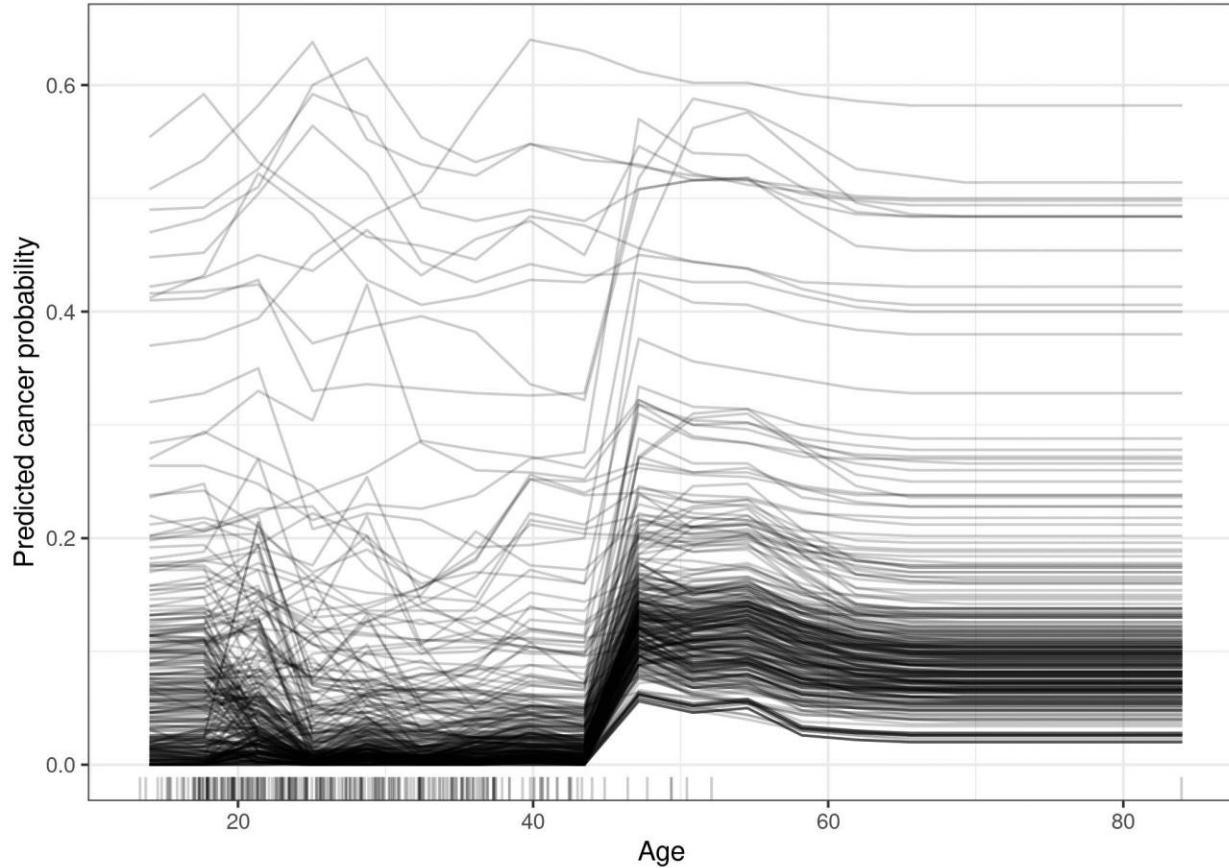
مزیت انتظارات فردی نسبت به وابستگی‌های جزئی چیست؟ نمودارهای وابستگی جزئی می‌توانند یک رابطه ناهمگن ایجاد شده توسط تعاملات را پنهان کنند. PDP‌ها می‌توانند به شما نشان دهند که میانگین رابطه بین یک ویژگی و پیش‌بینی چگونه است. این تنها زمانی به خوبی کار می‌کند که تعامل بین ویژگی‌هایی که PDP برای آنها محاسبه می‌شود و سایر ویژگی‌ها ضعیف باشد. در صورت تعامل، نمودار ICE بینش بسیار بیشتری را ارائه می‌دهد.

یک تعریف رسمی‌تر: در نمودارهای ICE، برای هر نمونه در $\hat{f}_S^{(i)}$ منحنی $\left\{ \left(x_S^{(i)}, x_C^{(i)} \right) \right\}$ در حالیه $x_C^{(i)}$ ثابت باقی می‌ماند.

۱-۹-۱- مثال‌ها

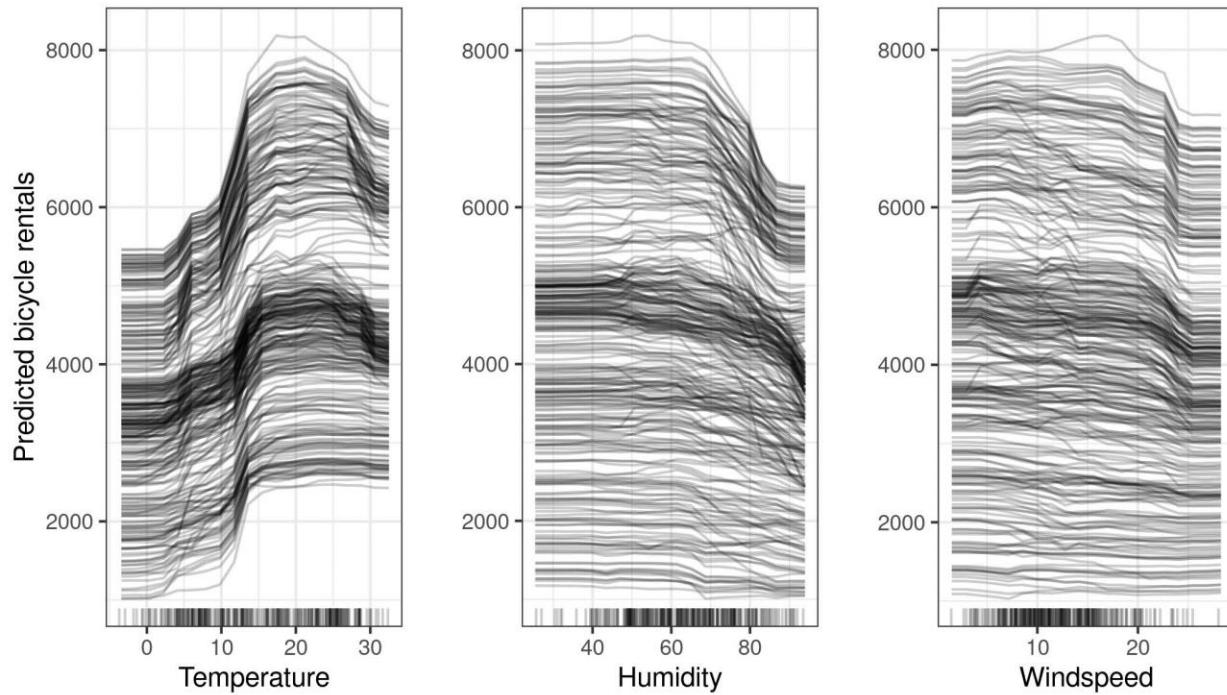
بیایید به مجموعه داده‌های سرطان دهانه رحم بازگردیم و ببینیم که چگونه پیش‌بینی هر نمونه با ویژگی "سن" مرتبط است. ما یک جنگل تصادفی را تجزیه و تحلیل خواهیم کرد که احتمال سرطان را برای یک زن با توجه به عوامل خطر پیش‌بینی می‌کند. در طرح وابستگی جزئی مشاهده کردہ‌ایم که احتمال ابتلا به سرطان در حدود سن ۵۰ سالگی افزایش می‌یابد، اما آیا این برای هر زن در مجموعه داده صادق است؟ نمودار ICE نشان می‌دهد که برای اکثر زنان اثر سن از الگوی متوسط افزایش در سن ۵۰ سالگی پیروی می‌کند، اما برخی استثنایا وجود دارد: برای معدود زنانی که احتمال پیش‌بینی شده بالایی در سنین جوانی دارند، احتمال سرطان پیش‌بینی شده، خیلی با افزایش سن تغییر نمی‌کند.

^۱ Individual Conditional Expectation



شکل ۹.۱: نمودار ICE احتمال سرطان دهانه رحم بر اساس سن. هر خط نشان دهنده یک زن است. برای اکثر زنان با افزایش سن احتمال سرطان پیش‌بینی شده افزایش می‌یابد. برای برخی از زنان با احتمال سرطان پیش‌بینی شده بالای ۴۰٪، پیش‌بینی در سن بالاتر تغییر چندانی نمی‌کند.

شکل بعدی نمودارهای ICE را برای پیش‌بینی اجاره دوچرخه نشان می‌دهد. مدل پیش‌بینی زیربنایی یک جنگل تصادفی است.



شکل ۹.۲: نمودارهای ICE از اجاره دوچرخه پیش‌بینی شده بر اساس شرایط آب و هوایی. همان اثرات را می‌توان در نمودارهای وابستگی جزئی مشاهده کرد.

به نظر می‌رسد همه منحنی‌ها مسیر یکسانی را دنبال می‌کنند، بنابراین هیچ تعامل آشکاری وجود ندارد. این بدان معنی است که PDP در حال حاضر خلاصه خوبی از روابط بین ویژگی‌های نمایش داده شده و تعداد پیش‌بینی شده دوچرخه است.

۱-۱-۹- طرح ICE مرکزی

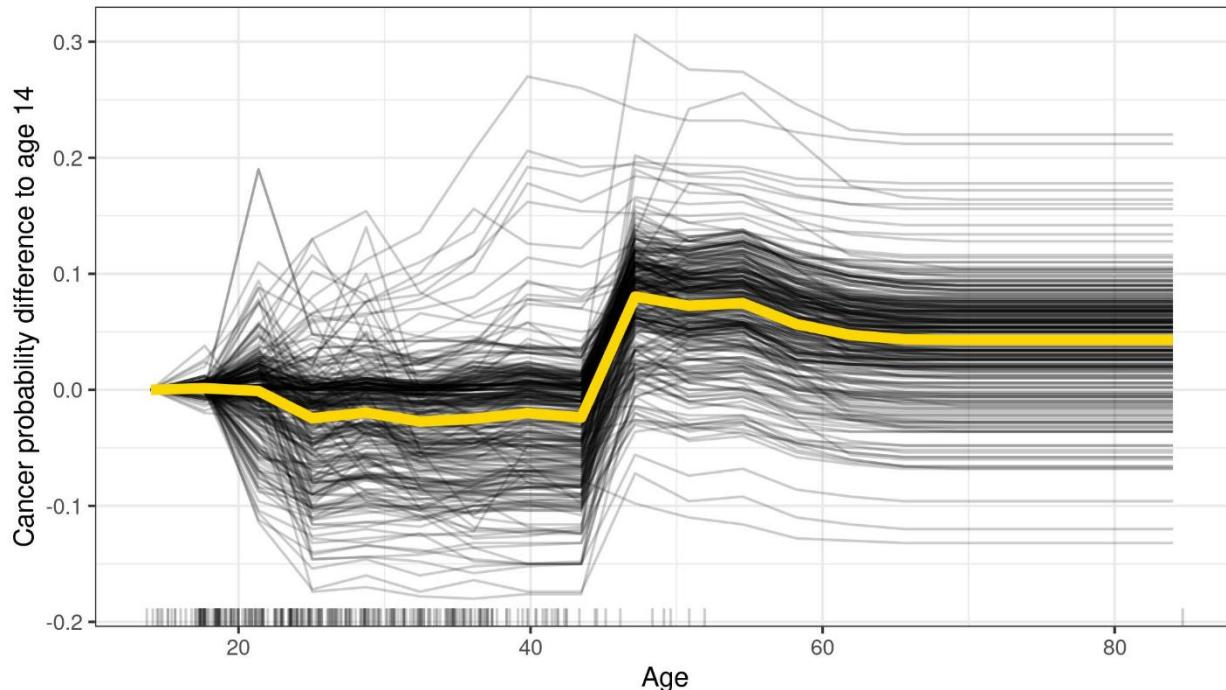
در نمودارهای ICE مشکلی وجود دارد: گاهی اوقات تشخیص اینکه آیا منحنی‌های ICE بین افراد متفاوت است یا خیر، دشوار است، زیرا آنها با پیش‌بینی‌های متفاوت شروع می‌شوند. یک راه حل ساده این است که منحنی‌ها را در نقطه خاصی از ویژگی متمرکز کنید و فقط تفاوت پیش‌بینی را تا این نقطه نشان دهید. نمودار حاصل را نمودار ICE مرکزی (c-ICE) می‌نامند. لنگر انداختن منحنی‌ها در انتهای پایین ویژگی انتخاب خوبی است. منحنی‌های جدید به صورت زیر تعریف می‌شوند:

$$\hat{f}_{cent}^{(i)} = \hat{f}^{(i)} - 1\hat{f}\left(x^a \cdot x_c^{(i)}\right)$$

در رابطه فوق، 1 ، یک بردار از a ‌ها با تعداد بعد مناسب (معمولًاً یک یا دو) است، \hat{f} مدل برآشش شده و x^a نقطه لنگر است.

۲-۱-۱-۹- مثال

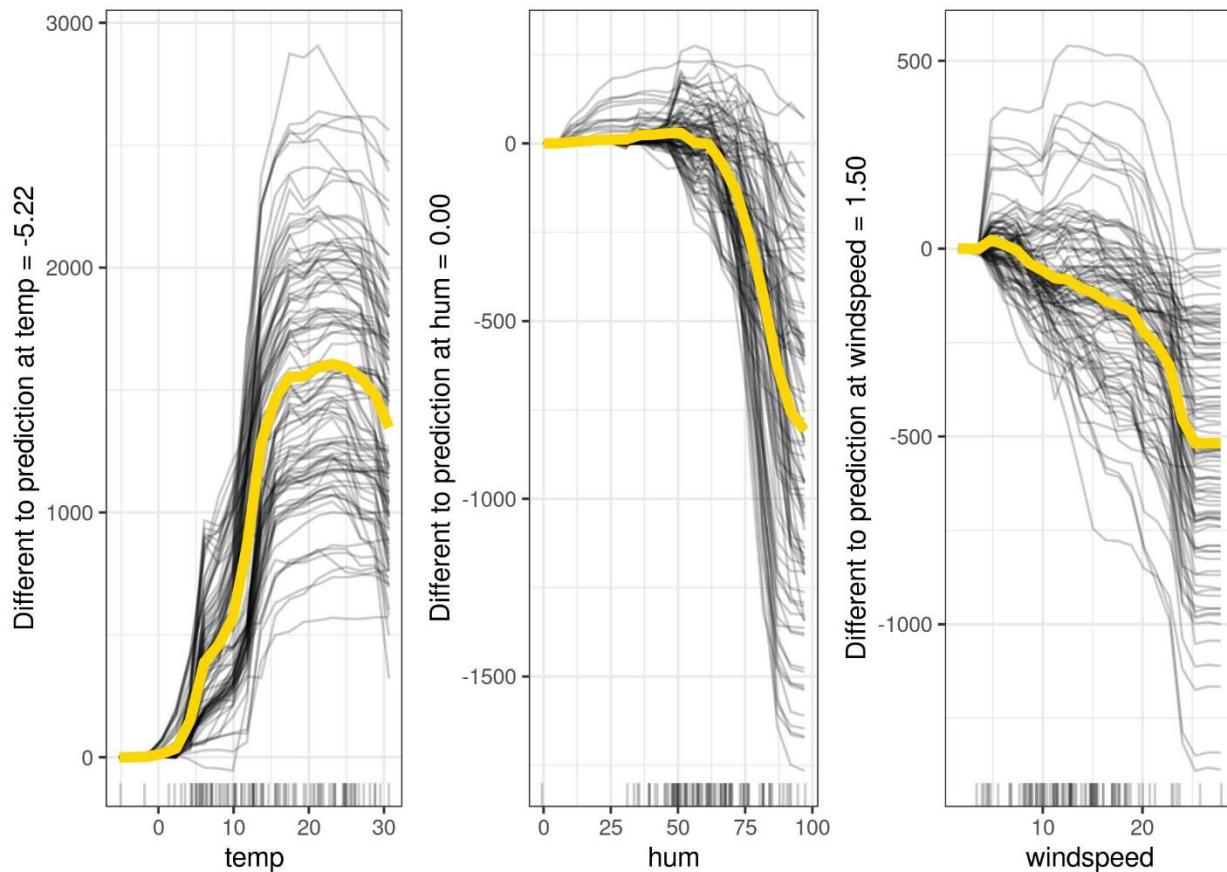
برای مثال، نمودار ICE سرطان دهانه رحم را برای سن در نظر بگیرید و خطوط را روی جوانترین سن مشاهده شده متumerکز کنید:



شکل ۹.۳: نمودار ICE مرکزی برای احتمال سرطان پیش‌بینی شده بر اساس سن. خطوط روی ۰ در سن ۱۴ سالگی ثابت می‌شوند. در مقایسه با سن ۱۴ سالگی، پیش‌بینی‌ها برای اکثر زنان تا سن ۴۵ سالگی بدون تغییر باقی می‌ماند، جایی که احتمال پیش‌بینی شده افزایش می‌یابد.

نمودارهای ICE در مرکز، مقایسه منحنی‌های نمونه‌های جداگانه را آسان‌تر می‌کند. این می‌تواند مفید باشد اگر بخواهیم تغییر مطلق یک مقدار پیش‌بینی شده را مشاهده نکنیم، اما تفاوت در پیش‌بینی را در مقایسه با یک نقطه ثابت از محدوده ویژگی مشاهده کنیم.

بیایید نگاهی به نمودارهای ICE متumerکز برای پیش‌بینی اجاره دوچرخه بیندازیم:



شکل ۹.۴: نمودارهای مرکز ICE تعداد پیش‌بینی شده دوچرخه‌ها بر اساس شرایط آب و هوایی. خطوط تفاوت در پیش‌بینی را در مقایسه با پیش‌بینی با مقدار ویژگی مربوطه در حداقل مشاهده شده نشان می‌دهد.

۹-۱-۳- نمودار ICE مشتق

راه دیگر برای آسان‌تر کردن تشخیص ناهمگونی از نظر بصری، نگاه‌کردن به مشتقات منفرد تابع پیش‌بینی با توجه‌به یک ویژگی است. نمودار حاصل، نمودار ICE مشتق (d-ICE) نامیده می‌شود. مشتقات یک تابع (یا منحنی) به شما می‌گوید که آیا تغییرات رخ می‌دهند و در کدام جهت رخ می‌دهند. با نمودار ICE مشتق، به راحتی می‌توان محدوده‌هایی از مقادیر ویژگی را که در آن پیش‌بینی‌های جعبه سیاه برای (حداقل برخی) موارد تغییر می‌کند، تشخیص داد. اگر هیچ تعاملی بین ویژگی تحلیل شده x_S وجود نداشته باشد و سایر ویژگی‌ها x_C سپس تابع پیش‌بینی را می‌توان به صورت زیر بیان کرد:

$$\hat{f}(x) = \hat{f}(x_S, x_C) = g(x_S) + h(x_C). \quad \text{with} \quad \frac{\delta \hat{f}(x)}{\delta x_S} = g'(x_S)$$

بدون تعاملات، مشتقات جزئی منفرد باید برای همه نمونه‌ها یکسان باشند. اگر تفاوت دارند، به دلیل تعامل است و در نمودار d-ICE قابل مشاهده است. علاوه بر نمایش منحنی‌های منفرد برای مشتق تابع پیش‌بینی با توجه‌به ویژگی در S ، نشان دادن انحراف استاندارد مشتق به برجسته کردن مناطق در ویژگی در S با ناهمگنی در

مشتقات برآورده شده کمک می‌کند. نمودار ICE مشتق برای محاسبه زمان زیادی طول می‌کشد و نسبتاً غیر عملی است.

۹-۱-۲- مزایا

منحنی‌های انتظار شرطی فردی حتی از نمودارهای وابستگی جزئی قابل درک تر هستند. اگر ویژگی مورد نظر را تغییر دهیم، یک خط پیش‌بینی‌ها را برای یک نمونه نشان می‌دهد. برخلاف نمودارهای وابستگی جزئی، منحنی‌های ICE می‌توانند روابط ناهمگن را آشکار کنند.

۹-۱-۳- معایب

منحنی‌های ICE فقط می‌توانند یک ویژگی را به‌طور معنی‌دار نشان دهند، زیرا دو ویژگی به ترسیم چندین سطح همپوشانی نیاز دارند و شما چیزی در نمودار نخواهید دید.

منحنی‌های ICE از همان مشکل PDP رنج می‌برند: اگر ویژگی مورد نظر با ویژگی‌های دیگر همبستگی داشته باشد، ممکن است برخی از نقاط در خطوط براساس توزیع ویژگی مشترک، نقاط داده نامعتبر باشند.

اگر منحنی‌های ICE زیادی رسم شود، نمودار می‌تواند بیش از حد شلوغ شود و شما چیزی نخواهید دید. راه حل: یا مقداری شفافیت به خطوط اضافه کنید یا فقط تعدادی از خطوط را بکشید.

در نمودارهای ICE ممکن است مشاهده میانگین آسان نباشد. این یک راه حل ساده دارد: منحنی‌های انتظار شرطی فردی را با نمودار وابستگی جزئی ترکیب کنید.

۹-۱-۴- نرم افزار و جایگزین

نمودارهای ICE در پکیج‌های iml (استفاده شده برای مثال‌های این بخش)، ICEbox (Goldstein et al., 2017) و pdp نرم افزار R پیاده‌سازی شده‌اند. یکی دیگر از پکیج‌های R که کاری بسیار شبیه به ICE انجام می‌دهد condvis است. در Python، طرح‌های وابستگی جزئی در scikit-learn از نسخه 0.24.0 پیاده‌سازی شدند.

۹.۲ جایگزین محلی (LIME)

مدل‌های جایگزین محلی مدل‌های قابل تفسیری هستند که برای توضیح پیش‌بینی‌های فردی مدل‌های یادگیری ماشین جعبه سیاه استفاده می‌شوند. توضیحات مدل قابل تفسیر محلی (LIME¹) (Ribeiro et al., 2016b) مقاله‌ای است که در آن نویسندها اجرای ملموسی از مدل‌های جایگزین محلی را پیشنهاد می‌کنند. مدل‌های جایگزین برای تقریب پیش‌بینی‌های مدل جعبه سیاه زیربنایی آموزش دیده‌اند. به جای آموزش یک مدل جایگزین کلی، LIME بر آموزش مدل‌های جایگزین محلی برای توضیح پیش‌بینی‌های فردی تمرکز می‌کند.

ایده کاملاً شهودی است. ابتدا داده‌های آموزشی را فراموش کنید و تصور کنید که فقط مدل جعبه سیاه را دارید که می‌توانید نقاط داده را وارد کنید و پیش‌بینی‌های مدل را به دست آورید. می‌توانید هر چندبار که بخواهید جعبه را بررسی کنید. هدف شما درک این موضوع است که چرا مدل یادگیری ماشین پیش‌بینی خاصی انجام داده است. LIME آزمایش می‌کند که وقتی تغییراتی از داده‌های خود را در مدل یادگیری ماشین می‌دهید، چه اتفاقی برای پیش‌بینی‌ها می‌افتد. LIME یک مجموعه‌داده جدید متشکل از نمونه‌های به هم ریخته و پیش‌بینی‌های مربوط به مدل جعبه سیاه تولید می‌کند. در این مجموعه‌داده جدید، LIME سپس یک مدل قابل تفسیر را آموزش می‌دهد که با نزدیکی نمونه‌های تولید شده به نمونه مورد نظر وزن دهی می‌شود. مدل قابل تفسیر می‌تواند هر مدلی از فصل مدل‌های قابل تفسیر باشد، برای مثال Lasso یا درخت تصمیم. مدل آموخته شده باید تقریب خوبی از پیش‌بینی‌های مدل یادگیری ماشین به صورت محلی باشد، اما لزومی ندارد که یک تقریب کلی خوب باشد. به این نوع دقت، وفاداری محلی² نیز می‌گویند.

از نظر ریاضی، مدل‌های جایگزین محلی با محدودیت تفسیرپذیری را می‌توان به صورت زیر بیان کرد:

$$\text{explanation}(x) = \underset{g \in G}{\operatorname{argmin}} L(f \cdot g \cdot \pi_x) + \Omega(g)$$

مدل توضیحی برای نمونه x ، مدل g (مثلاً مدل رگرسیون خطی) است که خطای L (مثلاً میانگین مربعات خطای x به حداقل می‌رساند، که میزان نزدیک بودن توضیح را به پیش‌بینی مدل اصلی f (مثلاً یک مدل xgboost) اندازه‌گیری می‌کند. پیچیدگی مدل (g) Ω پایین نگه داشته می‌شود (مثلاً ترجیح داده می‌شود که ویژگی‌های کمتری استفاده شود). G از خانواده توضیحات ممکن است، برای مثال تمام مدل‌های رگرسیون خطی ممکن. اندازه گیری مجاورت³ π_x تعیین می‌کند که همسایگی اطراف نمونه x چقدر است که برای توضیح در نظر می‌گیریم. در عمل، LIME فقط بخش خطای را بهینه می‌کند. کاربر باید پیچیدگی را تعیین کند، به عنوان مثال با انتخاب حداکثر تعداد ویژگی‌هایی که مدل رگرسیون خطی ممکن است استفاده کند.

¹ Local interpretable model-agnostic explanations

² local fidelity

³ proximity

دستور العمل برای آموزش مدل‌های جایگزین محلی:

- نمونه مورد علاقه خود را که می‌خواهید توضیحی درباره پیش‌بینی جعبه سیاه آن داشته باشد را انتخاب کنید.

- داده‌های خود را آشفته کنید و پیش‌بینی‌های جعبه سیاه را برای این نقاط جدید دریافت کنید.
- نمونه‌های جدید را با توجه به نزدیکی آنها به نمونه مورد نظر وزن کنید.
- یک مدل وزن دار و قابل تفسیر روی مجموعه داده با تغییرات آموزش دهید.
- پیش‌بینی را با تفسیر مدل محلی توضیح دهید.

به عنوان مثال، در پیاده سازی‌های فعلی در ¹R² و Python، رگرسیون خطی را می‌توان به عنوان مدل جایگزین قابل تفسیر انتخاب کرد. از قبل، باید K (تعداد ویژگی‌هایی که می‌خواهید در مدل قابل تفسیر خود داشته باشد) را انتخاب کنید. هرچه K کمتر باشد، تفسیر مدل آسان‌تر است. K بالاتر به طور بالقوه مدل‌هایی با وفاداری بالاتر تولید می‌کند. روش‌های مختلفی برای آموزش مدل‌هایی با ویژگی‌های دقیقاً K وجود دارد. یک انتخاب خوب Lasso است. مدل Lasso با پارامتر تنظیم λ بالا مدلی بدون هیچ ویژگی به ایجاد می‌کند. با بازآموزی مدل‌های Lasso با کاهش تدریجی λ ، یکی پس از دیگری، ویژگی‌ها تخمین وزنی را دریافت می‌کنند که با صفر متفاوت است. اگر K ویژگی در مدل وجود داشته باشد، به تعداد ویژگی‌های مورد نظر رسیده اید. راهبردهای دیگر انتخاب ویژگی‌ها رو به جلو یا عقب است. این بدان معناست که شما یا با مدل کامل (= شامل همه ویژگی‌ها) یا با مدلی که فقط عرض از مبدا دارد شروع کنید و سپس تست کنید که کدام ویژگی با اضافه یا حذف بیشترین پیشرفت را به همراه خواهد داشت تا زمانی که به مدلی با ویژگی‌های K برسید.

چگونه تغییرات داده‌ها را به دست می‌آورید؟ این بستگی به نوع داده دارد که می‌تواند متن، تصویر یا داده‌های جدولی باشد. برای متن و تصاویر، راه حل این است که تک کلمات یا سوپرپیکسل‌ها را روشن یا خاموش کنید. در مورد داده‌های جدولی، LIME با ایجاد اختلال در هر ویژگی به صورت جداگانه، از یک توزیع نرمال با میانگین و انحراف استاندارد گرفته شده از ویژگی، نمونه‌های جدیدی را ایجاد می‌کند.

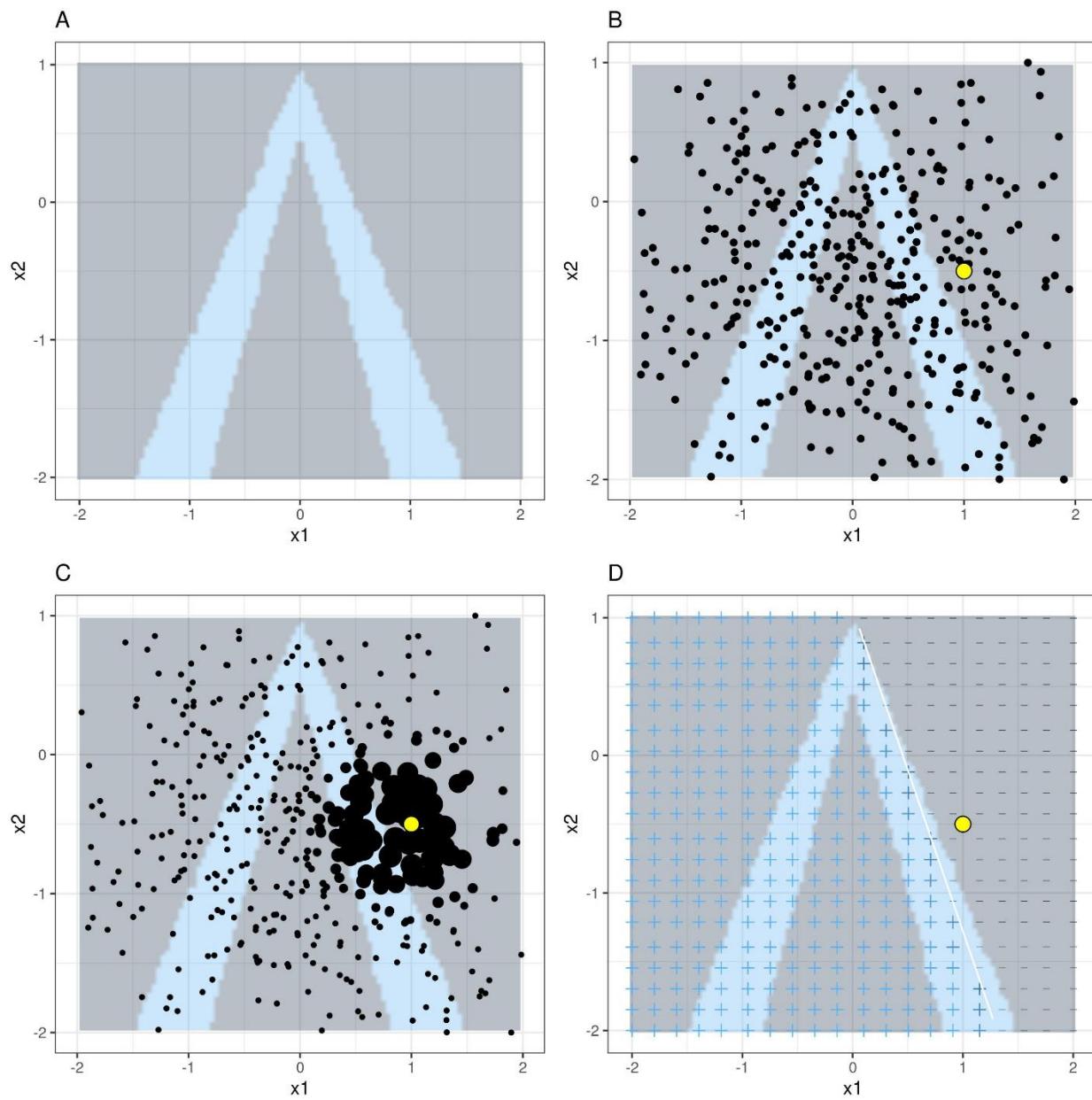
۹.۲.۱ LIME برای داده‌های جدولی

داده‌های جدولی داده‌هایی هستند که در جداول قرار می‌گیرند و هر ردیف نشان دهنده یک نمونه و هر ستون یک ویژگی است. نمونه‌های LIME در اطراف نمونه مورد نظر گرفته نمی‌شوند، بلکه از مرکز جرم داده‌های آموزشی گرفته می‌شوند، که مشکل‌ساز است. اما این احتمال را افزایش می‌دهد که نتیجه برخی از پیش‌بینی‌های نقاط نمونه با نقطه داده مورد علاقه متفاوت است و LIME می‌تواند حداقل توضیحی را بیاموزد.

¹ <https://github.com/thomasasp85/lime>

² <https://github.com/marcotcr/lime>

بهتر است به صورت تصویری توضیح دهیم که چگونه نمونه گیری و آموزش مدل محلی کار می‌کند:

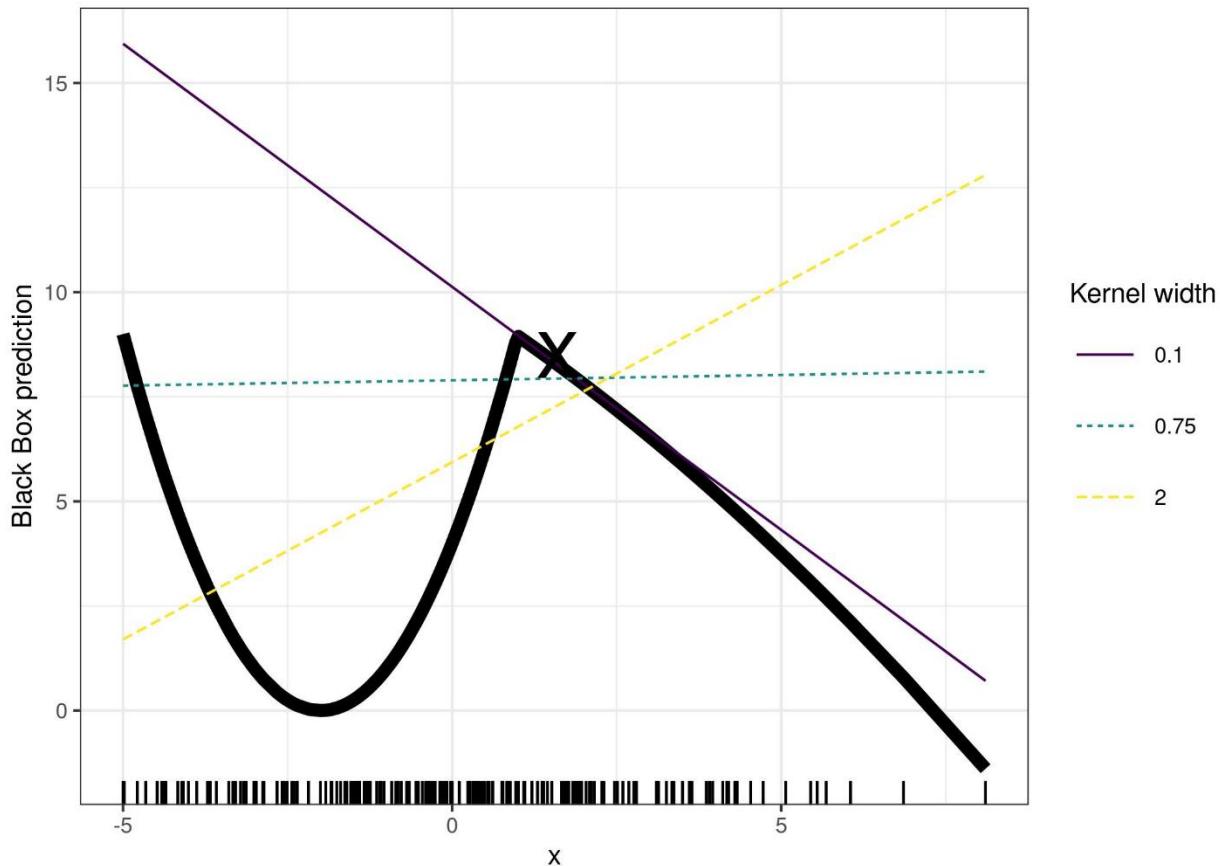


شکل ۹.۵: الگوریتم LIME برای داده‌های جدولی. الف) پیش‌بینی‌های جنگل تصادفی با ویژگی‌های x_1 و x_2 کلاس‌های پیش‌بینی شده: ۱ (تاریک) یا ۰ (روشن). ب) نمونه مورد علاقه (نقطه بزرگ) و داده‌های نمونه برداری شده از یک توزیع نرمال (نقاط کوچک). ج) وزن بیشتری را به نقاط نزدیک به نمونه مورد نظر اختصاص دهدید. د) علائم شبکه، طبقه‌بندی مدل محلی آموخته شده از نمونه‌های وزنی را نشان می‌دهد. خط سفید مرز تصمیم را مشخص می‌کند. $(P(\text{class}=1) = 0.5)$

مثل همیشه، مشکلات در جزئیات است. تعریف یک همسایگی معنادار در اطراف یک نقطه دشوار است. LIME در حال حاضر از یک هسته هموارسازی نمایی^۱ برای تعریف همسایگی استفاده می‌کند. یک هسته هموارسازی تابعی است که دو نمونه داده را می‌گیرد و یک اندازه گیری مجاورت را برمی‌گرداند. عرض هسته تعیین می‌کند که همسایگی چقدر بزرگ است: عرض هسته کوچک به این معنی است که یک نمونه باید بسیار نزدیک باشد تا بر مدل محلی تأثیر بگذارد، عرض هسته بزرگ‌تر به این معنی است که نمونه‌هایی که دورتر هستند نیز روی مدل تأثیر می‌گذارند. اگر به پیاده‌سازی LIME در Python^۲ (file lime/lime_tabular.py) نگاه کنید (فایل خواهید دید که از یک هسته هموارسازی نمایی (بر روی داده‌های نرم‌ال شده) استفاده می‌کند و عرض هسته ۰/۷۵ برابر ریشه دوم تعداد ستون‌های داده‌های آموزشی است. به نظر می‌رسد یک خط کد بی خطر است، اما مانند یک فیل است که در اتاق نشیمن شما در کنار ظروف چینی خوبی که از پدربزرگ و مادربزرگ‌تان گرفته اید، نشسته است. مشکل بزرگ این است که ما راه خوبی برای یافتن بهترین هسته یا عرض نداریم. ۰/۷۵ از کجا می‌آید؟ در سناریوهای خاصی، همان‌طور که در شکل زیر نشان داده شده است، می‌توانید به راحتی با تغییر عرض هسته، توضیح خود را تغییر دهید:

^۱ exponential smoothing kernel

^۲ <https://github.com/marcotcr/lime/tree/ce2db6f20f47c3330beb107bb17fd25840ca4606>



شکل ۹.۶: توضیح پیش‌بینی نمونه $x = 1.6$. مدل جعبه سیاه بسته به یک ویژگی منفرد به صورت یک خط ضخیم و توزیع داده‌ها با علاوه روی محور افقی نشان داده می‌شود. سه مدل جایگزین محلی با عرض هسته‌های مختلف محاسبه شده است. مدل رگرسیون خطی حاصل به عرض هسته بستگی دارد: آیا این ویژگی برای $x = 1.6$ تأثیر منفی، مثبت یا بدون تأثیر دارد؟

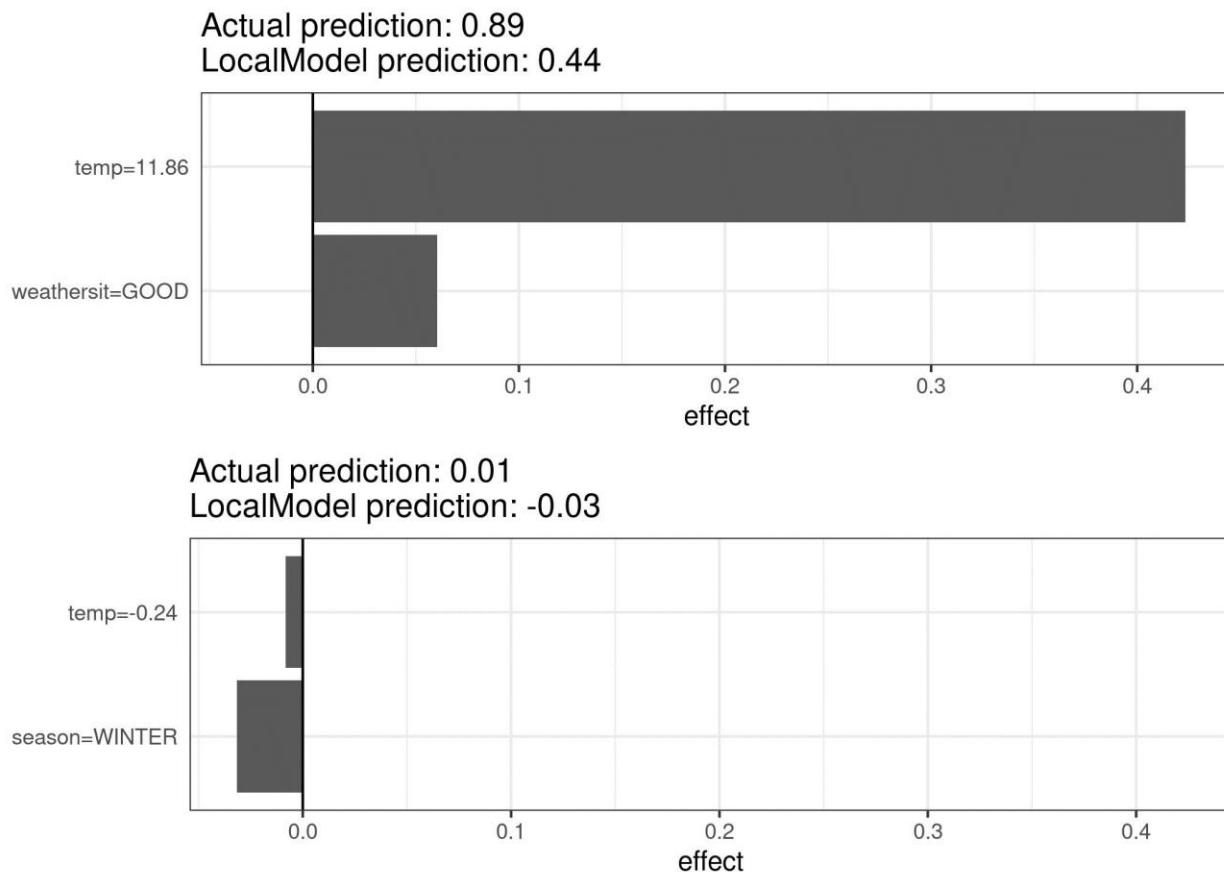
مثال فقط یک ویژگی را نشان می‌دهد. در فضاهای با ویژگی با ابعاد بالاتر اوضاع بدتر می‌شود. همچنین بسیار نامشخص است که آیا اندازه‌گیری فاصله باید همه ویژگی‌ها را یکسان در نظر بگیرد. آیا یک واحد فاصله برای x^1 با یک واحد برای x^2 یکسان است؟ اندازه‌گیری‌های فاصله کاملاً دلخواه هستند و فاصله‌ها در ابعاد (ویژگی‌های) مختلف ممکن است اصلاً قابل مقایسه نباشند.

۹.۲.۱.۱ مثال

اجازه دهید به یک مثال عینی نگاه کنیم. به داده‌های اجاره دوچرخه برمی‌گردیم و مسئله پیش‌بینی را به یک طبقه‌بندی تبدیل می‌کنیم: پس از در نظر گرفتن روندی که کرایه دوچرخه در طول زمان محبوب‌تر شده است، می‌خواهیم در یک روز مشخص بدانیم که آیا تعداد دوچرخه‌های اجاره‌شده، بالا یا زیر خط روند خواهد بود یا

خیر. همچنین می‌توانید «بالا» را به عنوان بالاتر از میانگین تعداد دوچرخه‌ها تعبیر کنید، اما برای روند تنظیم شده است.

ابتدا یک جنگل تصادفی با ۱۰۰ درخت را در کار طبقه‌بندی آموزش می‌دهیم. بر اساس اطلاعات آب و هوا و تقویم، در چه روزی تعداد دوچرخه‌های اجاره‌ای بالاتر از میانگین بدون روند خواهد بود؟ توضیحات با ۲ ویژگی ایجاد شده است. نتایج مدل‌های خطی محلی محدود برای دو نمونه با کلاس‌های پیش‌بینی شده متفاوت آموزش داده شد:



شکل ۹.۷: توضیحات LIME برای دو نمونه از مجموعه داده‌های اجاره دوچرخه. دمای گرمتر و وضعیت آب و هوای خوب تأثیر مثبتی بر پیش‌بینی دارد. محور x اثر ویژگی را نشان می‌دهد: وزن ضربدر مقدار واقعی ویژگی. از شکل مشخص می‌شود که تفسیر ویژگی‌های طبقه‌بندی آسان‌تر از ویژگی‌های عددی است. یک راه حل این است که ویژگی‌های عددی را به bin‌ها دسته‌بندی کنیم.

۹.۲.۲ LIME برای متن

برای متن با LIME برای داده‌های جدولی متفاوت است. تغییرات داده‌ها به طور متفاوتی تولید می‌شوند: با شروع از متن اصلی، متون جدید با حذف تصادفی کلمات از متن اصلی ایجاد می‌شوند. مجموعه داده با

ویژگی‌های باینری برای هر کلمه نشان داده می‌شود. یک ویژگی اگر کلمه مربوطه گنجانده شود ۱ و اگر حذف شده باشد ۰ است.

۹.۲.۱ مثال

در این مثال ما نظرات YouTube را به عنوان هرزنامه یا عادی طبقه‌بندی می‌کنیم. مدل جعبه سیاه یک درخت تصمیم عمیق است که بر روی ماتریس کلمه سند آموزش داده شده است. هر نظر یک سند (= یک ردیف) و هر ستون تعداد تکرار یک کلمه داده شده است. درخت‌های تصمیم گیری کوتاه به راحتی قابل درک هستند، اما در این مورد درخت بسیار عمیق است. همچنین به جای این درخت می‌توانست یک شبکه عصبی مکرر یا یک ماشین بردار پشتیبان آموزش داده شده بر روی جاسازی کلمات (بردارهای انتزاعی) وجود داشته باشد. اجازه دهید به دو نظر این مجموعه‌داده و کلاس‌های مربوطه نگاه کنیم (۱ برای هرزنامه، ۰ برای نظر عادی):

	CONTENT	CLASS
267	PSY is a good guy	0
173	For Christmas Song visit my channel! ;)	1

گام بعدی ایجاد برخی تغییرات از مجموعه داده‌های مورد استفاده در یک مدل محلی است. به عنوان مثال، برخی از تغییرات یکی از نظرات:

For	Christmas	Song	visit	my	channel!	;)	prob	weight
1	0	1	1	0	0	1	0.17	0.57
0	1	1	1	1	0	1	0.17	0.71
1	0	0	1	1	1	1	0.99	0.71
1	0	1	1	1	1	1	0.99	0.86
0	1	1	1	0	0	1	0.17	0.57

هر ستون مربوط به یک کلمه در جمله است. هر ردیف یک تغییر است، ۱ به این معنی است که کلمه بخشی از این تغییر است و ۰ به معنای حذف کلمه است. جمله مربوط به یکی از تغییرات "Christmas Song visit my channel!" (؛) است. ستون "prob" احتمال پیش‌بینی شده هرزنامه را برای هر یک از تغییرات جمله نشان می‌دهد. ستون "وزن" نزدیکی تغییر به جمله اصلی را نشان می‌دهد که به صورت ۱ منهای نسبت کلمات حذف شده محاسبه می‌شود، برای مثال اگر ۱ کلمه از ۷ کلمه حذف شود، نزدیکی $1 - \frac{7}{1} = 0.86$ است.

در اینجا دو جمله (یک هرزنامه، یکی بدون هرزنامه) با وزن محلی تخمین زده شده توسط الگوریتم LIME آمده است:

case	label_prob	feature	feature_weight
1	0.1701170	PSY	0.000000
1	0.1701170	guy	0.000000
1	0.1701170	good	0.000000
2	0.9939024	channel!	6.180747
2	0.9939024	;)	0.000000
2	0.9939024	visit	0.000000

کلمه "کانال" نشان دهنده احتمال بالای اسپم است. برای نظر غیر هرزنامه، وزن غیر صفر تخمین زده نشد، زیرا مهم نیست کدام کلمه حذف شود، کلاس پیش‌بینی شده ثابت می‌ماند.

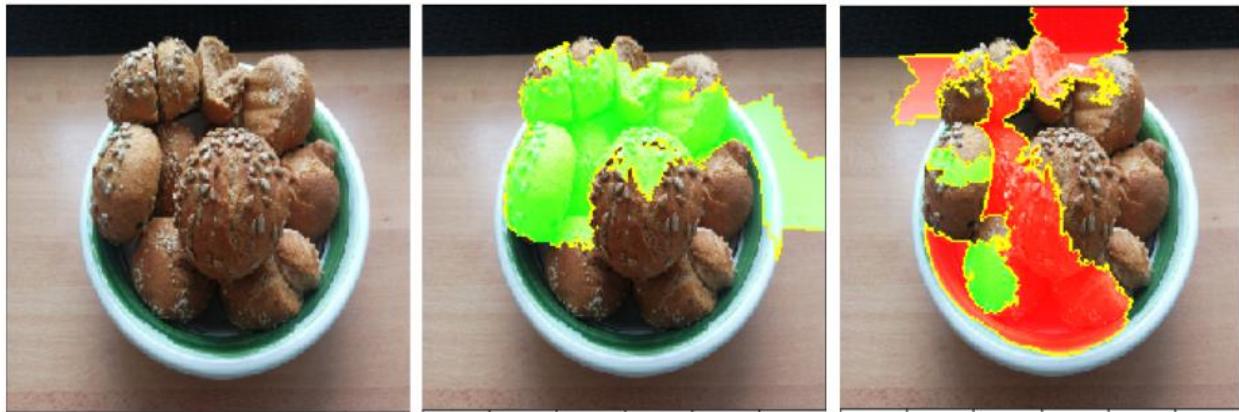
۹.۲.۲ LIME برای تصاویر

این بخش توسط Verena Haunschmid نوشته شده است.

LIME برای تصاویر متفاوت از LIME برای داده‌ها و متن جدولی عمل می‌کند. به طور شهودی، ایجاد مزاحمت برای پیکسل‌های مجزا چنان منطقی نخواهد بود، زیرا بیش از یک پیکسل در یک کلاس مشارکت دارند. تغییر تصادفی پیکسل‌های فردی احتمالاً پیش‌بینی‌ها را تغییر زیادی نمی‌دهد. بنابراین، تغییراتی در تصاویر با تقسیم بندی تصویر به "سوپرپیکسل" و خاموش یا روشن کردن سوپرپیکسل‌ها ایجاد می‌شود. سوپرپیکسل‌ها پیکسل‌های به هم پیوسته با رنگ‌های مشابه هستند و می‌توان با جایگزین کردن هر پیکسل با یک رنگ تعریف شده توسط کاربر مانند خاکستری، آن را خاموش کرد. کاربر همچنین می‌تواند یک احتمال برای خاموش کردن یک سوپرپیکسل در هر جایگشت مشخص کند.

۹.۲.۳.۱ مثال

در این مثال ما به طبقه‌بندی ساخته شده توسط شبکه عصبی Inception V3 نگاه می‌کنیم. تصویر استفاده شده مقداری نان را نشان می‌دهد که من پختم که در یک کاسه است. از آنجایی که می‌توانیم در هر تصویر چندین برچسب پیش‌بینی شده داشته باشیم (مرتب‌سازی شده بر اساس احتمال)، می‌توانیم برچسب‌های بالایی را توضیح دهیم. بالاترین پیش‌بینی "Bagel" با احتمال ۷۷٪ و پس از آن "توت فرنگی" با احتمال ۴٪ است. تصاویر زیر برای "Bagel" و "Strawberry" LIME توضیحات را نشان می‌دهد. توضیحات را می‌توان مستقیماً روی نمونه‌های تصویر نمایش داد. سبز به این معنی است که این قسمت از تصویر احتمال برچسب را افزایش می‌دهد و قرمز به معنای کاهش است.



شکل ۹.۸: سمت چپ: تصویر یک کاسه نان. وسط و راست: توضیحات LIME برای ۲ کلاس برتر (نان شیرینی، توت فرنگی) برای طبقه‌بندی تصاویر ساخته شده توسط شبکه عصبی Inception V3 Google پیش‌بینی و توضیح برای "Bagel" بسیار معقول است، حتی اگر پیش‌بینی اشتباه باشد - اینها به وضوح هیچ چیز شیرینی نیستند زیرا سوراخ در وسط آن وجود ندارد.

۹.۲.۴ مزایا

حتی اگر مدل اصلی یادگیری ماشین را جایگزین کنید، همچنان می‌توانید از همان مدل محلی و قابل تفسیر برای توضیح استفاده کنید. فرض کنید افرادی که به توضیحات نگاه می‌کنند درخت تصمیم را بهتر درک می‌کنند. از آنجایی که شما از مدل‌های جایگزین محلی استفاده می‌کنید، از درخت‌های تصمیم به عنوان توضیح استفاده می‌کنید بدون اینکه واقعاً مجبور باشید از درخت تصمیم برای پیش‌بینی‌ها استفاده کنید. برای مثال می‌توانید از SVM استفاده کنید. و اگر معلوم شد که یک مدل xgboost بهتر کار می‌کند، می‌توانید SVM را جایگزین کنید و همچنان از درخت تصمیم برای توضیح پیش‌بینی‌ها استفاده کنید.

مدل‌های جایگزین محلی از ادبیات و تجربه آموزش و تفسیر مدل‌های قابل تفسیر بهره می‌برند.

هنگام استفاده از Lasso یا درختان کوتاه، توضیحات حاصل کوتاه (= انتخابی) و احتمالاً مقابله‌ای هستند. بنابراین، آنها توضیحات انسان پسند ایجاد می‌کنند. به همین دلیل است که من LIME را بیشتر در برنامه‌هایی می‌بینم که دریافت‌کننده توضیح یک فرد عادی یا فردی با زمان بسیار کم است. این روش برای انتسابات کامل کافی نیست، بنابراین من LIME را در سناریوهای پذیرش که ممکن است از نظر قانونی ملزم به توضیح کامل یک پیش‌بینی باشد، نمی‌بینم. همچنین برای اشکال زدایی مدل‌های یادگیری ماشین، داشتن همه دلایل به جای چند دلیل مفید است.

LIME یکی از محدود روش‌هایی است که برای داده‌های جدولی، متن و تصاویر کار می‌کند.

معیار وفاداری (مدل قابل تفسیر تا چه حد به پیش‌بینی‌های جعبه سیاه تقریب می‌کند) به ما ایده خوبی درباره اینکه مدل قابل تفسیر در توضیح پیش‌بینی‌های جعبه سیاه در همسایگی نمونه داده مورد نظر چقدر قابل اعتماد است، ارائه می‌دهد.

LIME در پایتون (کتابخانه¹ lime²) و در R (پکیج‌های lime³ و iml⁴) پیاده‌سازی شده است و استفاده از آن بسیار آسان است.

توضیحات ایجاد شده با مدل‌های جایگزین محلی می‌تواند از ویژگی‌های دیگری (قابل تفسیر) نسبت به مدل اصلی استفاده کند. البته، این ویژگی‌های قابل تفسیر باید از نمونه‌های داده مشتق شوند. یک طبقه‌بندی‌کننده متن می‌تواند بر تعابیه‌های کلمات انتزاعی به عنوان ویژگی تکیه کند، اما توضیح می‌تواند بر اساس وجود یا عدم وجود کلمات در یک جمله باشد. یک مدل رگرسیون می‌تواند بر تبدیل غیر قابل تفسیر برخی از ویژگی‌ها تکیه کند، اما توضیحات را می‌توان با ویژگی‌های اصلی ایجاد کرد. به عنوان مثال، مدل رگرسیون را می‌توان بر روی مولفه‌های یک تجزیه و تحلیل مؤلفه اصلی (PCA⁴) پاسخ‌های یک نظرسنجی آموزش داد، اما LIME ممکن است در مورد سؤالات نظرسنجی اصلی آموزش داده شود. استفاده از ویژگی‌های قابل تفسیر برای LIME می‌تواند یک مزیت بزرگ نسبت به روش‌های دیگر باشد، به خصوص زمانی که مدل با ویژگی‌های غیر قابل تفسیر آموزش داده شده است.

۹.۲.۵ معایب

هنگام استفاده از LIME با داده‌های جدولی، تعریف صحیح همسایگی یک مشکل بسیار بزرگ و حل نشده است. به نظر من این بزرگ‌ترین مشکل LIME است و دلیل اینکه توصیه می‌کنم از LIME فقط با دقت زیاد استفاده کنید. برای هر برنامه باید تنظیمات هسته مختلف را امتحان کنید و خودتان ببینید که آیا توضیحات منطقی هستند یا خیر. متأسفانه، این بهترین توصیه ای است که می‌توانم برای یافتن پنهانی هسته خوب داشته باشم. نمونه برداری را می‌توان در پیاده‌سازی فعلی LIME بهبود بخشید. نقاط داده از یک توزیع گاووسی نمونه برداری می‌شوند و همبستگی بین ویژگی‌ها را نادیده می‌گیرند. این می‌تواند به نقاط داده غیرمحتمل منجر شود که سپس می‌توان از آنها برای یادگیری مدل‌های توضیح محلی استفاده کرد.

پیچیدگی مدل توضیح باید از قبل تعریف شود. این فقط یک شکایت کوچک است، زیرا در نهایت کاربر همیشه باید مصالحه بین وفاداری و پراکندگی را تعریف کند.

¹ <https://github.com/marcotcr/lime>

² <https://cran.r-project.org/web/packages/lime/index.html>

³ <https://cran.r-project.org/web/packages/iml/index.html>

⁴ principal component analysis

مشکل واقعاً بزرگ دیگر بی ثباتی توضیحات است. در مقاله (Alvarez-Melis & Jaakkola, 2018) نویسنده‌گان نشان دادند که توضیحات دو نقطه بسیار نزدیک در یک محیط شبیه سازی شده بسیار متفاوت است. همچنین، طبق تجربه من، اگر فرآیند نمونه برداری را تکرار کنید، توضیحاتی که می‌آید می‌تواند متفاوت باشد. بی ثباتی به این معناست که اعتماد به توضیحات دشوار است و باید بسیار انتقادپذیر باشد.

توضیحات LIME می‌توانند توسط دانشمند داده دستکاری^۱ شود تا سوگیری‌ها را پنهان کند (Slack et al., 2020). امکان دستکاری، اعتماد به توضیحات تولید شده با LIME را دشوارتر می‌کند.

نتیجه‌گیری: مدل‌های جایگزین محلی، با LIME به عنوان یک پیاده‌سازی عینی، بسیار امیدوارکننده هستند. اما این روش هنوز در مرحله توسعه است و بسیاری از مشکلات باید حل شود تا بتوان به طور ایمن از آن استفاده کرد.

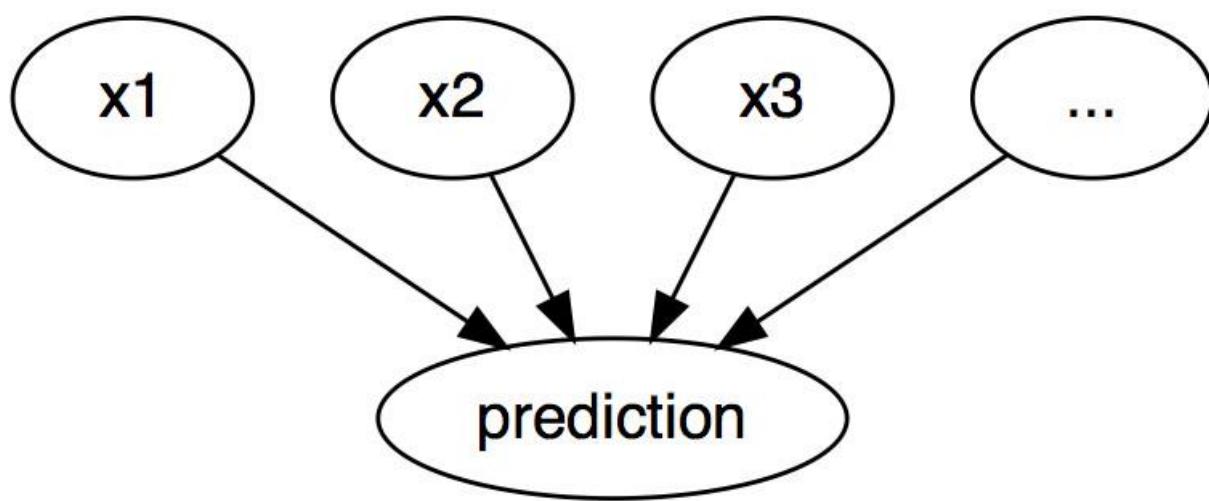
¹ manipulation

۹.۳ توضیحات خلاف واقع^۱

نویسندها: سوزان دنل و کریستوف مولنار

یک توضیح خلاف واقع یک وضعیت علی را به این شکل توصیف می‌کند: "اگر X رخ نمی‌داد، Y رخ نمی‌داد." به عنوان مثال: "اگر جرعه‌ای از این قهقهه داغ را ننوشیده بودم، زبانم نمی‌سوخت." رویداد Y این است که زبانم را سوزاندم. دلیل X این است که من یک قهقهه داغ خوردم. تفکر در خلاف واقع مستلزم تصور واقعیتی فرضی است که با واقعیت‌های مشاهده شده در تضاد است (مثلًاً دنیایی که در آن قهقهه داغ را ننوشیده ام)، از این رو نام آن را «خلاف واقع» گذاشته اند. توانایی تفکر خلاف واقع، ما انسان‌ها را در مقایسه با سایر حیوانات بسیار باهوش می‌کند.

در یادگیری ماشین قابل تفسیر، می‌توان از توضیحات خلاف واقع برای توضیح پیش‌بینی‌های نمونه‌های فردی استفاده کرد. «رویداد» نتیجه پیش‌بینی شده یک نمونه است، «علت‌ها» مقادیر ویژگی‌های خاص این نمونه هستند که به مدل وارد شده و یک پیش‌بینی مشخص را «سبب» می‌کنند. که به صورت نمودار نمایش داده می‌شود، رابطه بین ورودی‌ها و پیش‌بینی بسیار ساده است: مقادیر ویژگی باعث پیش‌بینی می‌شود.



شکل ۹.۹: روابط علی بین ورودی‌های یک مدل یادگیری ماشین و پیش‌بینی‌ها، زمانی که مدل صرفاً به عنوان یک جعبه سیاه دیده می‌شود. ورودی‌ها علت پیش‌بینی هستند (الزاماً منعکس کننده رابطه علی واقعی داده‌ها نیستند).

حتی اگر در واقعیت رابطه بین ورودی‌ها و نتیجه‌های که باید پیش‌بینی شود ممکن است علی نباشد، می‌توانیم ورودی‌های یک مدل را به عنوان علت پیش‌بینی بینیم.

^۱ Counterfactual Explanations

باتوجه به این نمودار ساده، به راحتی می‌توان فهمید که چگونه می‌توانیم خلاف واقع‌ها را برای پیش‌بینی‌های مدل‌های یادگیری ماشین شبیه‌سازی کنیم: ما به سادگی مقادیر ویژگی‌های یک نمونه را قبل از انجام پیش‌بینی‌ها تغییر می‌دهیم و چگونگی تغییر پیش‌بینی را تحلیل می‌کنیم. ما به سناریوهایی علاقه‌مند هستیم که در آن پیش‌بینی به شیوه‌ای مرتبط تغییر می‌کند، مانند جابجایی در کلاس پیش‌بینی شده (به عنوان مثال، درخواست اعتبار پذیرفته یا رد می‌شود)، یا در آن پیش‌بینی به آستانه خاصی می‌رسد (برای مثال، احتمال سرطان به ۱۰ درصد می‌رسد). توضیح خلاف واقع یک پیش‌بینی، کوچک‌ترین تغییر در مقادیر ویژگی را توصیف می‌کند که پیش‌بینی را به یک خروجی از پیش‌تعریف‌شده، تغییر می‌دهد.

هر دو روش توضیح خلاف واقع برای آگنوستیک مدل و خاص مدل وجود دارد، اما در این فصل ما بر روی روش‌های آگنوستیک مدل تمرکز می‌کنیم که فقط با ورودی‌ها و خروجی‌های مدل (و نه ساختار داخلی مدل‌های خاص) کار می‌کنند. این روش‌ها همچنین در فصل آگنوستیک مدل نیز قرار گیرند، زیرا تفسیر را می‌توان به صورت خلاصه‌ای از تفاوت‌ها در مقادیر ویژگی بیان کرد («تغییر ویژگی‌های A و B برای تغییر پیش‌بینی»). اما توضیح خلاف واقع خود یک نمونه جدید است، بنابراین در این فصل قرار گرفت («با شروع از مثال X، A و B را تغییر دهید تا یک نمونه خلاف واقع به دست آورید»). برخلاف نمونه‌های اولیه، خلاف واقع‌ها نباید نمونه‌های واقعی از داده‌های آموزشی باشند، بلکه می‌توانند ترکیب جدیدی از مقادیر ویژگی باشند.

قبل از بحث در مورد چگونگی ایجاد خلاف واقع، می‌خواهم در مورد موارد استفاده از خلاف واقع و اینکه چگونه یک توضیح خلاف واقع خوب به نظر می‌رسد، صحبت کنم.

در این مثال اول، پیتر برای وام درخواست می‌کند و توسط نرم‌افزار بانکی (مجهز به یادگیری ماشین) رد می‌شود. او تعجب می‌کند که چرا درخواستش رد شد و چگونه ممکن است شانس خود را برای دریافت وام افزایش دهد. سؤال «چرا» را می‌توان به عنوان خلاف واقع فرمول‌بندی کرد: کوچک‌ترین تغییر در ویژگی‌ها (درآمد، تعداد کارت‌های اعتباری، سن، و ...) که پیش‌بینی را از رد به تأیید تغییر می‌دهد چیست؟ یک پاسخ ممکن می‌تواند این باشد: اگر پیتر ۱۰۰۰۰ بیشتر در سال درآمد داشت، وام را دریافت می‌کرد. یا اگر پیتر کارت‌های اعتباری کمتری داشت و پنج سال پیش وام را نکول نکرده بود، وام را دریافت می‌کرد. پیتر هرگز دلایل رد را نمی‌داند، زیرا بانک علاقه‌ای به شفافیت ندارد، اما این داستان دیگری است.

در مثال دوم خود، می‌خواهیم مدلی را توضیح دهیم که یک نتیجه پیوسته را با توضیحات خلاف واقع پیش‌بینی می‌کند. آنا می‌خواهد آپارتمانش را اجاره کند، اما مطمئن نیست که چقدر برای آن هزینه بگیرد، بنابراین تصمیم می‌گیرد یک مدل یادگیری ماشین برای پیش‌بینی اجاره‌ها آموزش دهد. البته از آنجایی که آنا یک دانشمند داده است، مشکلات خود را از این طریق حل می‌کند. پس از وارد کردن تمام جزئیات در مورد اندازه، مکان، مجاز بودن حیوانات خانگی و غیره، مدل به او می‌گوید که می‌تواند ۹۰۰ یورو اجاره بگیرد. او انتظار ۱۰۰۰ یورو

یا بیشتر را داشت، اما به مدل خود اعتماد می‌کند و تصمیم می‌گیرد با ارزش‌های ویژگی‌های آپارتمان بازی کند تا ببیند چگونه می‌تواند ارزش آپارتمان را بهبود بخشد. او متوجه می‌شود که آپارتمان را می‌توان بیش از ۱۰۰۰ یورو اجاره کرد، اگر ۱۵ متر مربع بزرگ‌تر بود. دانش جالبی است، اما غیر قابل عمل، زیرا او نمی‌تواند آپارتمان خود را بزرگ کند. در نهایت، با تغییر دادن فقط مقادیر ویژگی‌های تحت کنترل خود (آشپزخانه داخلی بله/خیر، حیوانات خانگی مجاز هستند بله/خیر، نوع کف و غیره)، متوجه می‌شود که اگر اجازه حیوانات خانگی را بدهد و پنجره‌هایی با عایق بهتر نصب کند، او می‌تواند ۱۰۰۰ یورو شارژ کند. آنا به طور شهودی با عوامل خلاف واقع بازی کرده است تا نتیجه را تغییر دهد.

خلاف واقع‌ها توضیحاتی انسان پسند هستند، زیرا آنها با نمونه فعلی متضاد هستند و به دلیل مقابله‌ای بودن آنها، به این معنی که آنها معمولاً روی تعداد کمی از تغییرات ویژگی تمرکز می‌کنند. اما خلاف واقع‌ها از «اثر راشومون» رنج می‌برند. راشومون یک فیلم ژاپنی است که در آن قتل یک سامورایی توسط افراد مختلف روایت می‌شود. هر یک از داستان‌ها نتیجه را به یک اندازه به خوبی توضیح می‌دهند، اما داستان‌ها با یکدیگر تناقض دارند. همین امر می‌تواند در مورد خلاف واقع نیز اتفاق بیفتد، زیرا معمولاً چندین توضیح خلاف واقع مختلف وجود دارد. هر خلاف واقع «داستان» متفاوتی از چگونگی دستیابی به یک نتیجه معین می‌گوید. یک خلاف واقع ممکن است بگوید ویژگی A را تغییر دهید، خلاف واقع دیگر ممکن است بگوید A را همان طور باقی بگذارید اما ویژگی B را تغییر دهید که این یک تناقض است.

در مورد معیارها، چگونه توضیح خلاف واقع خوب را تعريف کنیم؟ ابتدا، کاربر یک توضیح خلاف واقع، یک تغییر مرتبط در پیش‌بینی یک نمونه (= واقعیت جایگزین) را تعريف می‌کند. اولین نیاز آشکار این است که یک نمونه خلاف واقع، پیش‌بینی از پیش تعريف شده را تا حد امکان نزدیک تولید کند. همیشه نمی‌توان با پیش‌بینی از پیش تعريف شده، یک خلاف واقع پیدا کرد. برای مثال، در یک تنظیم طبقه‌بندی با دو کلاس، یک کلاس نادر و یک کلاس مکرر، مدل ممکن است همیشه یک نمونه را به عنوان کلاس مکرر طبقه‌بندی کند. تغییر مقادیر ویژگی به طوری که برچسب پیش‌بینی شده از کلاس مکرر به کلاس نادر تغییر کند ممکن است غیرممکن باشد. بنابراین ما می‌خواهیم این شرط را که پیش‌بینی خلاف واقع باید دقیقاً با نتیجه از پیش تعريف شده مطابقت داشته باشد، کاهش دهیم. در مثال طبقه‌بندی، می‌توانیم به دنبال خلاف واقع باشیم که در آن احتمال پیش‌بینی شده کلاس نادر بهجای ۲ درصد فعلی به ۱۰ درصد افزایش می‌یابد. پس سؤال این است که حداقل تغییرات در ویژگی‌ها چیست به طوری که احتمال پیش‌بینی شده از ۲٪ به ۱۰٪ (یا نزدیک به ۱۰٪) تغییر می‌کند؟

یکی دیگر از معیارهای کیفی این است که یک خلاف واقع باید تا حد امکان مشابه نمونه مربوط به مقادیر ویژگی باشد. فاصله بین دو نمونه را می‌توان به عنوان مثال با فاصله منهتن^۱ یا فاصله Gower اندازه گیری کرد اگر هم ویژگی‌های گسسته و هم پیوسته داشته باشیم. خلاف واقع نه تنها باید به نمونه اصلی نزدیک باشد، بلکه باید تا حد امکان ویژگی‌های کمتری را تغییر دهد. برای سنجش میزان خوب بودن توضیح خلاف واقع در این متريک، می‌توانيم به سادگي تعداد ویژگي‌های تغيير يافته را بشماريم يا به تعبيير رياضي فانتزى، نرم L_0 آن را بين مثال خلاف واقع و واقعی اندازه گيري کنيم.

ثالثاً، اغلب مطلوب است که چندين توپیخ خلاف واقع متنوع ایجاد شود، به طوری که تصمیم گیرنده به چندین روش ممکن برای ایجاد یک نتیجه متفاوت دسترسی پیدا کند. به عنوان مثال، در ادامه مثال وام ما، یک توضیح خلاف واقع ممکن است فقط دوبرابر کردن درآمد برای دریافت وام را پیشنهاد کند، در حالی که خلاف واقع ممکن است انتقال به یک شهر مجاور را پیشنهاد کند و درآمد را با مقدار کمی افزایش دهد تا وام دریافت کند. می‌توان اشاره کرد که در حالی که اولین خلاف واقع ممکن است برای برخی امکان‌پذیر باشد، دومی ممکن است برای دیگران قابل اجراء باشد. بنابراین، علاوه بر ارائه یک موضوع تصمیم با روش‌های مختلف برای رسیدن به نتیجه مطلوب، تنوع همچنین افراد "متنوع" را قادر می‌سازد تا ویژگی‌هایی را که برای آنها مناسب است تغیير دهند.

آخرین شرط این است که یک نمونه خلاف واقع باید حاوی مقادیر ویژگی که محتمل است. ایجاد توضیح خلاف واقع برای مثال اجاره که در آن اندازه یک آپارتمان منفی است یا تعداد اتاق‌ها روی ۲۰۰ اتاق تنظیم شده است، منطقی نیست. حتی بهتر است که خلاف واقع بر اساس توزیع توان داده‌ها باشد. به عنوان مثال، یک آپارتمان با ۱۰ اتاق و ۲۰ متر مربع نباید به عنوان توضیح خلاف واقع در نظر گرفته شود. در حالت ایده‌آل، اگر تعداد متر مربع افزایش یابد، افزایش تعداد اتاق‌ها نیز باید پیشنهاد شود.

۹.۳.۱ ایجاد توضیحات خلاف واقع

یک رویکرد ساده و ساده لوحانه برای ایجاد توضیحات خلاف واقع، جستجو با آزمون و خطای است. این رویکرد شامل تغییر تصادفی مقادیر ویژگی نمونه مورد علاقه و توقف زمانی است که خروجی مورد نظر پیش‌بینی می‌شود. مانند مثالی که آنا سعی کرد نسخه‌ای از آپارتمان خود را پیدا کند که بتواند برای آن اجاره بیشتری بگیرد. اما رویکردهای بهتری نسبت به آزمون و خطای وجود دارد. ابتدا یک تابع ضرر را بر اساس معیارهای ذکر شده در بالا تعریف می‌کنیم. این ضرر به عنوان ورودی نمونه مورد علاقه، خلاف واقع و نتیجه مطلوب (خلاف واقع) را می‌گیرد. سپس، می‌توانیم توضیح خلاف واقع را پیدا کنیم که این تلفات را با استفاده از یک الگوریتم

^۱ Manhattan distance

بهینه‌سازی به حداقل می‌رساند. بسیاری از روش‌ها به این شکل پیش می‌روند، اما در تعریف تابع ضرر و روش بهینه سازی متفاوت هستند.

در ادامه، به دو مورد از آنها می‌پردازیم: اول، یکی از (۲۰۱۷) Wachter et al که تبیین خلاف واقع را به عنوان یک روش تفسیری معرفی کردند و دوم، توضیح (۲۰۲۰) Dandl et al که هر چهار معیار ذکر شده در بالا را در نظر می‌گیرد.

۹.۳.۱.۱ روش توسط و اچتر و همکاران .

(۲۰۱۷) پیشنهاد به حداقل رساندن خطای زیر را دادند: Wachter et al

$$L(x, x', y', \lambda) = \lambda \cdot (\hat{f}(x') - y')^2 + d(x, x')$$

عبارت اول فاصله درجه دوم بین پیش‌بینی مدل برای x' خلاف واقع و نتیجه مطلوب y' است که کاربر باید از قبل آن را تعریف کند. جمله دوم فاصله d بین مثال x که باید توضیح داده شود و x' خلاف واقع است. زیان اندازه‌گیری می‌کند که نتیجه پیش‌بینی شده خلاف واقع تا چه اندازه با نتیجه از پیش تعریف شده فاصله دارد و خلاف واقع چقدر با نمونه مورد علاقه فاصله دارد. تابع فاصله d به عنوان فاصله منهتن وزن شده با میانگین معکوس انحراف مطلق (MAD) هر ویژگی تعریف می‌شود.

$$d(x, x') = \sum_{j=1}^p \frac{|x_j - x'_j|}{MAD_j}$$

فاصله کل مجموع تمام فواصل p از نظر ویژگی است، یعنی تفاوت مطلق مقادیر ویژگی بین مثال x و خلاف واقع x' . فواصل از نظر ویژگی با معکوس انحراف مطلق میانه ویژگی زبر روی مجموعه داده تعریف شده به صورت زیر مقیاس می‌شوند:

$$MAD_j = median_{i \in \{1, \dots, n\}}(|x_{i,j} - median_{l \in \{1, \dots, n\}}(x_{l,j})|)$$

میانه یک بردار مقداری است که در آن نیمی از مقادیر بردار بزرگ‌تر و نیمی دیگر کوچک‌تر باشد. MAD معادل واریانس یک ویژگی است، اما به جای استفاده از میانگین به عنوان مرکز و جمع بر روی فواصل مربع، از میانه به عنوان مرکز و جمع در فواصل مطلق استفاده می‌کنیم. تابع فاصله پیشنهادی این مزیت را نسبت به فاصله اقلیدسی دارد که نسبت به نقاط پرت قوی‌تر است. مقیاس بندی با MAD برای رساندن همه ویژگی‌ها به یک مقیاس ضروری است - مهم نیست که اندازه یک آپارتمان را در متر مربع یا فوت مربع اندازه گیری کنید.

پارامتر λ فاصله را در پیش‌بینی (عبارت اول) با فاصله در مقادیر ویژگی (عبارت دوم) متعادل می‌کند. خطای برای یک λ معین حل می‌شود و یک x خلاف واقع را برمی‌گرداند. ارزش بالاتر از λ به این معنی است که ما خلاف واقع‌ها را با پیش‌بینی‌های نزدیک به نتیجه مطلوب y ترجیح می‌دهیم، یک مقدار کمتر به این معنی است که ما خلاف واقع‌های x را ترجیح می‌دهیم که در مقادیر ویژگی بسیار شبیه به x هستند. اگر λ بسیار بزرگ است،

نمونه ای با پیش‌بینی نزدیک به y انتخاب خواهد شد، صرف‌نظر از اینکه چقدر از x فاصله دارد. در نهایت، کاربر باید تصمیم بگیرد که چگونه بین شرطی که پیش‌بینی خلاف واقع با نتیجه دلخواه مطابقت دارد، تعادل برقرار کند. نویسنده‌گان روش، به جای انتخاب مقداری برای λ برای انتخاب یک تلوانس ϵ برای چقدر دوری از y' پیش‌بینی نمونه خلاف واقع مجاز است. این محدودیت را می‌توان به صورت زیر نوشت:

$$|\hat{f}(x') - y'| \leq \epsilon$$

برای به حداقل رساندن اینتابع خطأ، می‌توان از هر الگوریتم بهینه سازی مناسبی مانند Nelder-Mead استفاده کرد. اگر به گرادیان‌های مدل یادگیری ماشین دسترسی دارید، می‌توانید از روش‌های مبتنی بر گرادیان مانند ADAM استفاده کنید. نمونه x که باید توضیح داده شود، خروجی مورد نظر y و پارامتر تلوانس ϵ باید از قبل تنظیم شود. تابع خطأ برای x' و خلاف واقع بهینه (محلی) x' به حداقل می‌رسد در حالی با افزایش λ تا یک راه حل به حدکافی نزدیک یافت شود (= درون پارامتر تلوانس):

$$\arg \min_{x'} \max_{\lambda} L(x, x', y', \lambda)$$

به طور کلی، دستورالعمل تهیه خلاف واقع، ساده است:

۱- یک نمونه x را برای توضیح انتخاب کنید، نتیجه مورد نظر y ، یک تلوانس ϵ و یک مقدار اولیه (km) برای λ

۲- یک نمونه تصادفی را به عنوان خلاف واقع اولیه نمونه بگیرید.

۳- خطأ را با نمونه اولیه خلاف واقع به عنوان نقطه شروع بهینه کنید.

۴- تا وقتی که $|\hat{f}(x') - y'| < \epsilon$:

○ λ را افزایش دهید.

○ ضرر را با خلاف واقع فعلی به عنوان نقطه شروع بهینه کنید.

○ خلاف واقع را که ضرر را به حداقل می‌رساند، برگردانید.

۵- مراحل ۴-۲ را تکرار کنید و لیستی از موارد خلاف واقع یا موردی که ضرر را به حداقل می‌رساند را برگردانید.

روش پیشنهادی دارای معایبی است. این روش فقط معیارهای اول و دوم را در نظر می‌گیرد و دو معیار آخر ("تولید موارد خلاف واقع تنها با چند تغییر ویژگی و مقادیر ویژگی موردنظر) را در نظر نمی‌گیرد. d راه حل‌های محدود را ترجیح نمی‌دهد زیرا افزایش ۱۰ ویژگی در ۱ همان فاصله را به x می‌دهد که با افزایش یک ویژگی به ۱۰ رخ می‌دهد. به x می‌دهد. ترکیب‌های غیرواقعی ویژگی جریمه نمی‌شوند.

این روش ویژگی‌های طبقه‌بندی شده با سطوح مختلف را به خوبی مدیریت نمی‌کند. نویسنده‌گان روش پیشنهاد کردند که روش را به طور جداگانه برای هر ترکیبی از مقادیر ویژگی ویژگی‌های طبقه‌بندی اجرا کنید،

اما اگر چندین ویژگی طبقه‌بندی با مقادیر زیاد داشته باشید، این امر منجر به انفجار ترکیبی می‌شود. به عنوان مثال، شش ویژگی طبقه‌بندی شده با ده سطح منحصر به فرد به معنای یک میلیون اجرا است. اجازه دهید اکنون نگاهی به رویکرد دیگری برای غلبه بر این مشکلات بیندازیم.

۹.۳.۱.۲ روش دندل و همکاران

Dandl et al (۲۰۲۰) پیشنهاد کردند به طور همزمان یک خطا چهار هدفه به حداقل برسد:

$$L(x, x', y', X^{obs}) = \left(o_1(\hat{f}(x), y') \cdot o_2(x, x') \cdot o_3(x, x') \cdot o_4(x', X^{obs}) \right)$$

هر یک از چهار هدف o_1 تا o_4 با یکی از چهار معیار ذکر شده در بالا مطابقت دارد. هدف اول o_1 نشان می‌دهد که پیش‌بینی ' x' خلاف واقع ما باید تا حد امکان به پیش‌بینی ' y' مورد نظر ما نزدیک باشد. بنابراین ما می‌خواهیم فاصله بین (x, \hat{f}) و (y', X) را به حداقل برسانیم. در اینجا فاصله با متريک منهتن (نرم L_1) محاسبه می‌شود:

$$o_1(\hat{f}(x), y') = \begin{cases} 0 & \text{if } \hat{f}(x') \in y' \\ \inf_{y' \in y'} & \text{else} \end{cases}$$

هدف دوم o_2 نشان می‌دهد که خلاف واقع ما باید تا حد امکان مشابه نمونه x ما باشد. فاصله بین ' x' و x را با استفاده از روش Gower محاسبه می‌شود:

$$o_2(x, x') = \frac{1}{p} \sum_{j=1}^p \delta_G(x_j, x'_j)$$

در این رابطه p تعداد ویژگی‌ها است. مقدار δ_G بستگی به نوع ویژگی x_j دارد:

$$\delta_G(x_j, x'_j) = \begin{cases} \frac{1}{\hat{R}_j} |x_j - x'_j| & \text{if } x_j \text{ numerical} \\ \mathbb{I}_{x_j \neq x'_j} & \text{if } x_j \text{ categorical} \end{cases}$$

تقسیم فاصله یک ویژگی عددی \hat{R}_j (محدوده مقدار مشاهده شده)، مقیاس‌های δ_G برای همه ویژگی‌های را بین ۰ و ۱ قرار می‌دهد.

فاصله Gower می‌تواند هم ویژگی‌های عددی و هم ویژگی‌های طبقه‌ای را کنترل کند، اما تعداد ویژگی‌های تغییر یافته را محاسبه نمی‌کند. بنابراین، تعداد ویژگی‌ها را در هدف سوم o_3 با استفاده از نرم L_0 می‌شماریم:

$$o_3(x, x') = \|x - x'\|_0 = \sum_{j=1}^p \mathbb{I}_{x_j \neq x'_j}$$

با به حداقل رساندن هدف o_3 برای سومین معیار (تغییرات محدود ویژگی‌ها) است.

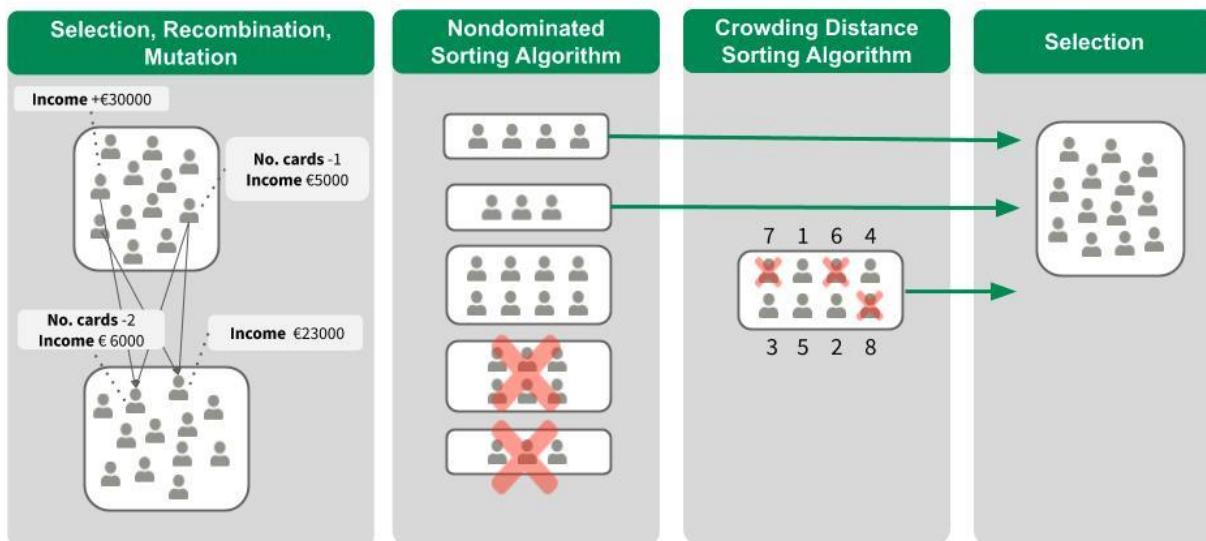
هدف چهارم o_4 نشان می‌دهد که خلاف واقع‌های ما باید مقادیر/ترکیب‌های ویژگی کمی داشته باشند. ما می‌توانیم حدس بزنیم که یک نقطه داده چقدر "شابت" از داده‌های آموزشی یا مجموعه‌داده دیگری استفاده

می‌کند. ما این مجموعه‌داده را با X^{obs} نشان می‌دهیم. به عنوان تقریبی برای احتمال، o_4 میانگین فاصله Gower بین ' x ' و نزدیکترین نقطه داده مشاهده شده $x^{[1]} \in X^{obs}$ را اندازه می‌گیرد:

$$o_4(x'.X^{obs}) = \frac{1}{p} \sum_{j=1}^p \delta_G(x'_j \cdot x_j^{[1]})$$

در مقایسه با $L(x \cdot x' \cdot y' \cdot \lambda)$, Wachter et al. (۲۰۱۷) هیچ پارامتر متعادل کردن یا وزن دهی شبیه λ ندارد. ما نمی‌خواهیم چهار هدف (o_1, o_2, o_3, o_4) را با وزن دهی و تجمعی در یک هدف واحد از بین ببریم بلکه می‌خواهیم هر چهار عبارت را به طور همزمان بهینه کنیم.

چطور می‌توانیم انجامش دهیم؟ ما از الگوریتم ژنتیک با مرتب‌سازی نامغلوب^۱ (Deb et al., 2002) یا به اختصار NSGA-II استفاده می‌کنیم. NSGA-II یک الگوریتم الهام گرفته از طبیعت است که قانون داروین را در مورد "بقاء قویترین"^۲ اعمال می‌کند. ما هزینه یک خلاف واقع را با بردار مقادیر اهداف آن (o_1, o_2, o_3, o_4) نشان می‌دهیم. هرچه مقادیر اهداف برای یک خلاف واقع کمتر باشد، "مناسب تر" است. این الگوریتم از چهار مرحله تشکیل شده است که تا زمانی که یک معیار توقف (به عنوان مثال، حداقل تعداد تکرار / نسل) برآورده شود، تکرار می‌شود. شکل زیر چهار مرحله یک نسل را به تصویر می‌کشد.



شکل ۹.۱۰: تجسم یک نسل از الگوریتم NSGA-II.

در نسل اول، گروهی از کاندیدهای خلاف واقع با تغییر تصادفی برخی از ویژگی‌ها در مقایسه با مثال x مقداردهی اولیه می‌شوند. با رعایت مثال اعتباری بالا، یک مخالف می‌تواند افزایش درآمد را ۳۰۰۰۰ یورو

^۱ Nondominated Sorting Genetic Algorithm

^۲ survival of the fittest

پیشنهاد کند، در حالی که یکی دیگر پیشنهاد می‌کند که در پنج سال گذشته نکول نداشته باشد و سن را تا ۱۰ کاهش دهید. همه مقادیر دیگر ویژگی برابر با مقادیر x هستند. سپس هر کاندید با استفاده از چهار تابع هدف فوق ارزیابی می‌شود. از بین آنها، ما به صورت تصادفی چند کاندید را انتخاب می‌کنیم، که در آن کاندیدهای متناسب با احتمال بیشتری انتخاب می‌شوند. کاندیدها به صورت جفتی دوباره ترکیب می‌شوند تا با میانگین‌گیری مقادیر مشخصه عددی یا با عبور از ویژگی‌های طبقه‌بندی، فرزندانی شبیه به آنها تولید کنند.

علاوه بر این، مقادیر ویژگی‌های فرزندان را کمی جهش می‌دهیم تا کل فضای ویژگی را کشف کنیم. از دو گروه به دست آمده، یکی با والدین و دیگری با فرزندان، ما فقط بهترین نیمه را با استفاده از دو الگوریتم مرتب سازی می‌خواهیم. الگوریتم مرتب سازی نامغلوب، کاندیدها را بر اساس مقادیر هدف آنها مرتب می‌کند. اگر کاندیدها به همان اندازه خوب باشند، الگوریتم مرتب‌سازی فاصله ازدحام^۱ کاندیدها را بر اساس تنوع آنها مرتب می‌کند.

باتوجه به رتبه‌بندی دو الگوریتم مرتب‌سازی، امیدوارکننده‌ترین و/یا متنوع‌ترین نیمی از کاندیدها را انتخاب می‌کنیم. ما از این مجموعه برای نسل بعدی استفاده می‌کنیم و دوباره با فرآیند انتخاب، نوترکیب^۲ و جهش^۳ شروع می‌کنیم. با تکرار مکرر مراحل، امیدواریم به مجموعه متنوعی از کاندیدهای امیدوارکننده با مقادیر هدف پایین نزدیک شویم. از این مجموعه می‌توانیم مواردی را انتخاب کنیم که از آنها رضایت بیشتری داریم، یا می‌توانیم خلاصه‌ای از همه موارد خلاف واقع را با برجسته کردن این که کدام و چند بار ویژگی‌ها تغییر کرده‌اند ارائه دهیم.

۹.۳.۲ مثال

مثال زیر بر اساس نمونه داده اعتباری در (Dandl et al., 2020) است. مجموعه‌داده ریسک اعتباری آلمان را می‌توان در پلتفرم چالش‌های یادگیری ماشین kaggle.com پیدا کرد.

نویسنده‌گان یک ماشین بردار پشتیبان (با هسته پایه شعاعی) را آموزش دادند تا احتمال اینکه یک مشتری ریسک اعتباری خوبی دارد را پیش‌بینی کند. مجموعه‌داده مربوطه دارای ۵۲۲ مشاهدات کامل و ۹ ویژگی حاوی اطلاعات اعتباری و مشتری است.

هدف یافتن توضیحات خلاف واقع برای مشتری با مقادیر ویژگی زیر است:

age	sex	job	housing	savings	amount	duration	purpose
58	f	unskilled	free	little	6143	48	car

پیش‌بینی می‌کند که زن دارای ریسک اعتباری خوبی با احتمال ۲۴/۲ درصد است. خلاف واقع‌ها باید پاسخ دهند که چگونه ویژگی‌های ورودی باید تغییر کنند تا احتمال پیش‌بینی شده بزرگ‌تر از ۵۰٪ به دست آید؟

¹ crowding distance

² recombination

³ mutation

جدول زیر بهترین ده خلاف واقع را نشان می‌دهد:

age	sex	job	amount	duration	o2	o3	o4	$\hat{f}(x')$
		skilled		-20	0.108	2	0.036	0.501
		skilled		-24	0.114	2	0.029	0.525
		skilled		-22	0.111	2	0.033	0.513
-6		skilled		-24	0.126	3	0.018	0.505
-3		skilled		-24	0.120	3	0.024	0.515
-1		skilled		-24	0.116	3	0.027	0.522
-3	m			-24	0.195	3	0.012	0.501
-6	m			-25	0.202	3	0.011	0.501
-30	m	skilled		-24	0.285	4	0.005	0.590
-4	m		-1254	-24	0.204	4	0.002	0.506

پنج ستون اول شامل تغییرات ویژگی پیشنهادی است (فقط ویژگی‌های تغییر یافته نمایش داده می‌شود)، سه ستون بعدی مقادیر هدف را نشان می‌دهد (01 در تمام موارد برابر با ۰ است) و آخرین ستون احتمال پیش‌بینی شده را نشان می‌دهد.

همه خلاف واقع‌ها احتمالات بیشتر از ۵۰٪ را پیش‌بینی کرده اند و همدیگر را مغلوب ^۱ نمی‌کنند. همدیگر را مغلوب نمی‌کنند به این معنی است که هیچ یک از خلاف واقع‌ها در همه اهداف دارای مقادیر کوچک‌تری نسبت به سایر موارد خلاف واقع نیستند. ما می‌توانیم خلاف واقع‌های خود را به عنوان مجموعه‌ای از راه حل‌های مبادله ای در نظر بگیریم.

همه آنها کاهش مدت زمان را از ۴۸ ماه به حداقل ۲۳ ماه پیشنهاد می‌کنند، برخی از آنها پیشنهاد می‌کنند که زن باید به جای غیر ماهر، ماهر شود. برخی خلاف واقع‌ها حتی پیشنهاد می‌کنند که جنسیت را از زن به مرد تغییر دهید که نشان دهنده تعصب جنسیتی مدل است. این تغییر همیشه با کاهش سن بین یک تا ۳۰ سال همراه است. همچنین می‌توانیم بینیم که، اگرچه برخی خلاف واقع‌ها تغییراتی را در چهار ویژگی پیشنهاد می‌کنند، اما این خلاف واقع‌ها آنهایی هستند که به داده‌های آموزشی نزدیک‌تر هستند.

۹.۳.۳ مزایا

تفسیر توضیحات خلاف واقع بسیار روشن است. اگر مقادیر ویژگی یک نمونه بر اساس خلاف واقع تغییر کند، پیش‌بینی به پیش‌بینی از پیش تعریف شده تغییر می‌کند. هیچ فرض اضافی و هیچ جادوی در پس زمینه وجود ندارد. این همچنین به این معنی است که به اندازه روش‌هایی مانند LIME خطرناک نیست، جایی که مشخص نیست تا چه حد می‌توانیم مدل محلی را برای تفسیر تعبیر کنیم.

¹ dominate

روش خلاف واقع یک نمونه جدید ایجاد می‌کند، اما می‌توانیم یک خلاف واقع را با گزارش دادن اینکه کدام مقادیر ویژگی تغییر کرده است، خلاصه کنیم. این کار دو گزینه برای گزارش نتایج به ما می‌دهد. می‌توانید نمونه خلاف واقع را گزارش دهید یا مشخص کنید که کدام ویژگی بین نمونه مورد علاقه و نمونه خلاف واقع تغییر کرده است.

روش خلاف واقع نیازی به دسترسی به داده یا مدل ندارد. این فقط نیاز به دسترسی به تابع پیش‌بینی مدل دارد، که برای مثال از طریق یک وب API نیز کار می‌کند. این برای شرکت‌هایی که توسط اشخاص ثالث حسابرسی می‌شوند یا بدون افسای مدل یا داده‌ها توضیحاتی را برای کاربران ارائه می‌دهند جذاب است. یک شرکت به دلیل اسرار تجاری یا دلایل حفاظت از داده‌ها، علاقه‌مند به محافظت از مدل و داده است. توضیحات خلاف واقع تعادلی بین توضیح پیش‌بینی‌های مدل و حفاظت از منافع مالک مدل ارائه می‌دهد.

این روش همچنین با سیستم‌هایی کار می‌کند که از یادگیری ماشین استفاده نمی‌کنند. ما می‌توانیم برای هر سیستمی که ورودی‌ها را دریافت می‌کند و خروجی‌ها را برمی‌گرداند، خلاف واقع ایجاد کنیم. سیستمی که احراه آپارتمان را پیش‌بینی می‌کند همچنین می‌تواند شامل قوانین دستنویس باشد و توضیحات خلاف واقع همچنان کارساز است.

پیاده‌سازی روش توضیح خلاف واقع نسبتاً آسان است، زیرا اساساً یک تابع ضرر (با یک یا چند هدف) است که می‌تواند با کتابخانه‌های بهینه‌ساز استاندارد بهینه شود. برخی جزئیات اضافی باید در نظر گرفته شود، مانند محدود کردن مقادیر ویژگی به محدوده‌های معنی دار (مثلاً فقط اندازه‌های آپارتمان مثبت).

۹.۳.۴ معايب

برای هر نمونه معمولاً چندین توضیح خلاف واقع (اثر راشومون) پیدا خواهد کرد. این عیب است چون بیشتر مردم توضیحات ساده را به پیچیدگی دنیای واقعی ترجیح می‌دهند. همچنین یک چالش عملی است. فرض کنید برای یک نمونه ۲۳ توضیح خلاف واقع ایجاد کردیم. آیا ما همه آنها را گزارش می‌کنیم؟ تنها بهترین؟ اگر همه آنها نسبتاً "خوب" اما بسیار متفاوت باشند چه؟ برای هر پروژه باید دوباره به این سوالات پاسخ داد. همچنین داشتن چندین توضیح خلاف واقع می‌تواند سودمند باشد، زیرا انسان‌ها می‌توانند آن‌ها را انتخاب کنند که با دانش قبلی شان مطابقت دارند.

۹.۳.۵ نرم افزار و جایگزین

روش توضیح خلاف واقع چندهدفه توسط Dandl et al. (۲۰۲۰) در یک مخزن GitHub^۱ پیاده‌سازی شده است.

^۱ <https://github.com/susanne-207/moc/tree/master/counterfactuals>

در پکیج^۱ Alibi، نرم افزار Python نویسنده‌گان یک روش خلاف واقع ساده و همچنین یک روش توسعه یافته را پیاده‌سازی کردند که از نمونه‌های اولیه کلاس برای بهبود تفسیرپذیری و همگرایی خروجی‌های الگوریتم استفاده می‌کند.(Van Looveren & Klaise, 2021)

Karimi et al (۲۰۲۰) همچنین پیاده‌سازی پایتون از الگوریتم MACE خود را در یک مخزن GitHub^۲ ارائه کرد. آنها معیارهای لازم برای خلاف واقع‌های مناسب را به فرمول‌های منطقی ترجمه کردند و از حل‌کننده‌های رضایت‌پذیری برای یافتن خلاف واقع‌هایی که آن‌ها را برآورده می‌کنند، استفاده کردند.

Mothilal et al (۲۰۲۰)، DiCE^۳ (تبیین متضاد متنوع)^۴ را برای تولید مجموعه متنوعی از توضیحات خلاف واقع بر اساس فرآیندهای نقطه تعیین کننده توسعه داد DiCE . هم یک روش آگنوستیک مدل و هم یک روش مبتنی بر گرادیان را پیاده‌سازی می‌کند.

روش دیگر برای جستجوی خلاف واقع، الگوریتم Growing Spheres توسط Laugel et al (۲۰۱۷) می‌باشد. آنها از کلمه خلاف واقع در مقاله خود استفاده نمی‌کنند، اما روش کاملا مشابه است. آنها همچنین یکتابع ضرر را تعریف می‌کنند که به نفع خلاف واقع‌ها با کمترین تغییرات ممکن در مقادیر ویژگی است. به جای بهینه سازی مستقیم تابع، آنها پیشنهاد می‌کنند ابتدا یک کره در اطراف نقطه مورد نظر ترسیم کنند، از نقاط آن کره نمونه برداری کنید و بررسی کنید که آیا یکی از نقاط نمونه برداری شده پیش‌بینی مورد نظر را به دست می‌دهد یا خیر. سپس کره را بر این اساس منقبض یا منبسط می‌کنند تا زمانی که یک خلاف واقع (محدود) پیدا شود و در نهایت برگردانده شود.

Ribeiro et al (۲۰۱۸) بر عکس خلاف واقع هستند، به فصل مربوط به قوانین محدوده (لنگرها) مراجعه کنید.

¹ <https://github.com/SeldonIO/alibi>

² <https://github.com/amirhk/mace>

³ <https://github.com/interpretml/DiCE>

⁴ Diverse Counterfactual Explanation

۹.۴ قوانین محدوده^۱ (لنگرها^۲)

نویسندها: Tobias Goerke & Magdalena Lang

روش لنگرها، پیش‌بینی‌های فردی هر مدل طبقه‌بندی جعبه سیاه را با یافتن یک قانون تصمیم‌گیری که پیش‌بینی را به اندازه کافی «لنگر» می‌کند، توضیح می‌دهد. در صورتی که تغییرات در سایر مقادیر ویژگی بر پیش‌بینی تأثیری نداشته باشد، یک قانون یک پیش‌بینی را لنگر می‌کند. لنگرها از تکنیک‌های یادگیری تقویتی^۳ در ترکیب با الگوریتم جستجوی گراف^۴ استفاده می‌کند تا تعداد تماس‌های مدل (و در نتیجه زمان اجرا موردنیاز) را به حداقل کاهش دهد و در عین حال قادر به عبور از بهینه محلی باشد. Robnik-Šikonja and Bohanec (2018) (همان محققانی که الگوریتم LIME را معرفی کردند) الگوریتم را پیشنهاد کردند.

مانند سلف خود، رویکرد لنگرها یک استراتژی مبتنی بر اختلال^۵ را برای ایجاد توضیحات محلی برای پیش‌بینی مدل‌های یادگیری ماشین جعبه سیاه به کار می‌گیرد. با این حال، به جای مدل‌های جایگزینی که توسط LIME استفاده می‌شود، توضیحات به دست آمده با استفاده از قوانین قابل درک IF-THEN بیان می‌شوند که لنگر نامیده می‌شوند. این قوانین قابل استفاده مجدد هستند، زیرا دارای محدوده هستند: لنگرها شامل مفهوم پوشش^۶ است که نشان می‌دهد که دقیقاً در مورد کدام نمونه‌های دیگر، احتمالاً دیده نشده، بکار برده می‌شود. یافتن لنگرها شامل یک مساله اکتشافی یا راهزن چند دست^۷ است که منشا آن در رشته یادگیری تقویتی است. برای این منظور، همسایگان، یا اختلالات، برای هر نمونه ای که توضیح داده می‌شود، ایجاد و ارزیابی می‌شود. انجام این کار به رویکرد اجازه می‌دهد تا ساختار جعبه سیاه و پارامترهای داخلی آن را نادیده بگیرد تا این پارامترها هم مشاهده نشده و هم بدون تغییر باقی بمانند. بنابراین، الگوریتم آگنوستیک مدل است، به این معنی که می‌توان آن را برای هر کلاس از مدل اعمال کرد.

نویسندها در مقاله خود، هر دو الگوریتم خود را مقایسه می‌کنند و تجسم می‌کنند که چگونه این الگوریتم‌ها با همسایگی یک نمونه متفاوت مشورت می‌کنند تا نتایج را به دست آورند. برای این کار، شکل زیر هم LIME و هم لنگرها را نشان می‌دهد که به صورت محلی یک طبقه‌بندی‌کننده باینری پیچیده (که - یا + پیش‌بینی می‌کند) را با استفاده از دو نمونه الگو، توضیح می‌دهد. نتایج LIME بیان نمی‌کنند که چقدر وفادار هستند، زیرا LIME تنها یک مرز تصمیم‌گیری خطی را می‌آموزد که بهترین مدل تقریب در فضای اختلال D است. با توجه به فضای اختلال یکسان، رویکرد لنگرها توضیحاتی را می‌سازد که پوشش آن با رفتار مدل تطبیق داده شده است و

¹ Scoped Rules

² Anchors

³ reinforcement learning

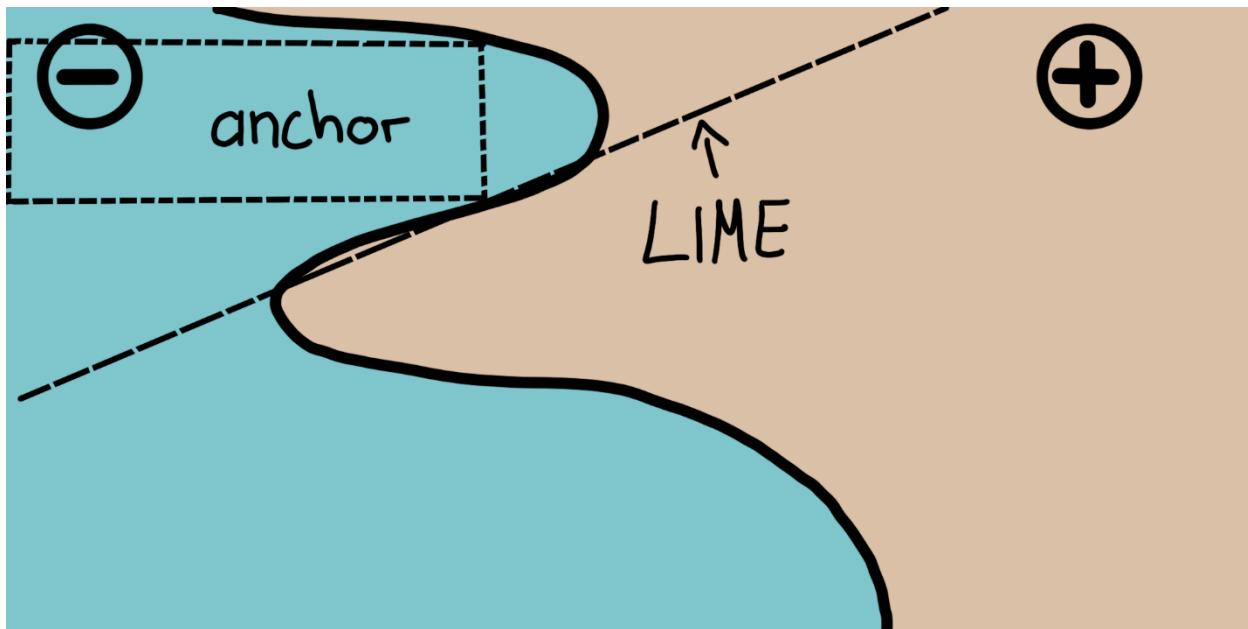
⁴ graph search algorithm

⁵ perturbation-based

⁶ coverage

⁷ multi-armed bandit problem

رویکرد به وضوح مرزهای آنها را بیان می‌کند. بنابراین، آنها از نظر طراحی وفادار هستند و نشان می‌دهند که دقیقاً برای کدام نمونه‌ها معتبر هستند. این ویژگی لنگرها را شهودی و درک آن را آسان می‌کند.



شکل ۹.۱۱: LIME در مقابل لنگرها – یک تجسم ساده. شکلی از (Ribeiro et al., 2018) همان‌طور که قبلاً ذکر شد، نتایج یا توضیحات الگوریتم در قالب قوانینی به نام لنگر آمده است. مثال ساده زیر چنین لنگری را نشان می‌دهد. به عنوان مثال، فرض کنید یک مدل جعبه سیاه به ما داده می‌شود که پیش‌بینی می‌کند مسافری از فاجعه تایتانیک جان سالم به در برده است یا خیر. اکنون می‌خواهیم بدانیم چرا مدل برای یک فرد خاص پیش‌بینی زنده ماندن می‌کند. الگوریتم لنگرها یک توضیح را مانند آنچه در زیر آورده شده است ارائه می‌دهد.

Feature	Value
Age	20
Sex	female
Class	First
Ticket price	300\$
More attributes	...
Survived	True

و توضیح لنگرهای مربوطه این گونه است:

IF SEX = female AND Class = first THEN PREDICT Survived = true WITH PRECISION 97% AND COVERAGE 15%

این مثال نشان می‌دهد که چگونه لنگرها می‌توانند بینش‌های اساسی را در مورد پیش‌بینی یک مدل و استدلال زیربنایی آن ارائه دهند. نتیجه نشان می‌دهد که کدام ویژگی‌ها توسط مدل در نظر گرفته شده است که در این مورد، جنسیت زن و کلاس درجه یک است. انسان‌ها که مشتاق صحت هستند، می‌توانند از این قانون برای

اعتبارسنجی رفتار مدل استفاده کنند. لنگر علاوه بر این به ما می‌گوید این قانون برای ۱۵ درصد نمونه های فضای اختلال برقرار است. در این موارد، توضیح ۹۷٪ دقیق است، به این معنی که گزاره ها^۱ نمایش داده شده تقریباً به طور انحصاری مسئول خروجی پیش‌بینی شده هستند.

یک لنگر A به طور رسمی به شرح زیر تعریف می‌شود:

$$\mathbb{E}_{D_x(z|A)} \left[1_{\hat{f}_{(x)} = \hat{f}(z)} \right] \geq \tau \cdot A(x) = 1$$

که در آن:

- x نمونه ای را نشان می‌دهد که توضیح داده می‌شود (به عنوان مثال یک ردیف در مجموعه داده‌های جدولی).
- A مجموعه‌ای از گزاره هاست، یعنی قانون یا لنگر حاصل، به گونه ای که $A(x) = 1$ هنگامی که تمام گزاره های ویژگی تعریف شده توسط A مرتبط با مقادیر ویژگی x.
- f نشان دهنده مدل طبقه‌بندی است که باید توضیح داده شود (به عنوان مثال یک مدل شبکه عصبی مصنوعی). می‌توان پیش‌بینی یک برچسب x و اختلالات آن را بر اساس آن به دست آورد.
- $D_x(\cdot | A)$ نشان دهنده توزیع همسایگان x مطابق با A.
- $1 \leq \tau \leq 0$ یک آستانه دقت را مشخص می‌کند. فقط قوانینی که حداقل وفاداری محلی τ دست می‌یابند، نتیجه معتبر محسوب می‌شوند.

توصیف رسمی ممکن است ترسناک باشد و می‌تواند در قالب کلمات بیان شود:

برای یک نمونه معین x که می‌خواهد توضیح داده شود، یک قانون یا یک لنگر A یافت می‌شود، به گونه ای به x اعمال می‌شود تا مادامیکه همان کلاس پیش‌بینی شده برای x برای یک کسر حداقلی τ از همسایگان x برای همان A قابل اعمال شدن است. دقت یک قانون از ارزیابی همسایگان یا انحرافات به شرح $D_x(z|A)$ با استفاده از مدل یادگیری ماشین ارائه شده (که با تابع نشانگر $1_{\hat{f}_{(x)} = \hat{f}(z)}$ مشخص می‌شود).

۹.۴.۱ یافتن لنگرها

اگرچه ممکن است توصیف ریاضی لنگرها واضح و ساده به نظر برسد، ساختن قوانین خاص غیرممکن است. نیاز به ارزیابی $1_{\hat{f}_{(x)} = \hat{f}(z)}$ برای تمام $z \in D_x(\cdot | A)$ می‌باشد که این کار در فضاهای ورودی پیوسته یا بزرگ امکان‌پذیر نیست. بنابراین، نویسنده‌گان پیشنهاد می‌کنند که پارامتر $1 \leq \tau \leq 0$ برای ایجاد یک تعریف احتمالی تعریف شود. به این ترتیب، نمونه‌ها تا زمانی که اطمینان آماری در مورد دقت آنها وجود داشته باشد، ترسیم می‌شوند. تعریف احتمالی به شرح زیر است:

¹ predicates

$$P(prec(A) \geq \tau) \geq 1 - \delta \quad \text{with} \quad prec(A) = \mathbb{E}_{D_x(z|A)} \left[1_{\hat{f}(x)=\hat{f}(z)} \right]$$

دو تعریف قبلی با مفهوم پوشش ترکیب شده و گسترش یافته است. منطق آن شامل یافتن قوانینی است که ترجیحاً برای بخش بزرگی از فضای ورودی مدل اعمال می‌شود. پوشش به طور رسمی به عنوان احتمال یک لنگر برای اعمال به همسایگانش، یعنی فضای انحراف آن تعریف می‌شود:

$$cov(A) = \mathbb{E}_{D_z} [A(z)]$$

گنجاندن این عنصر به تعریف نهایی لنگر با در نظر گرفتن حداکثر کردن پوشش منجر می‌شود:

$$\max_{A \text{ s.t. } P(prec(A) \geq \tau) \geq 1 - \delta} cov(A)$$

بنابراین، روند رسیدگی برای قانونی تلاش می‌کند که بالاترین پوشش را در بین همه قوانین واجد شرایط داشته باشد (همه قوانینی که آستانه دقت را باتوجه به تعریف احتمالی برآورده می‌کنند). تصور می‌شود که این قوانین مهم‌تر هستند، زیرا بخش بزرگ‌تری از مدل را توصیف می‌کنند. توجه داشته باشید که قوانین با پیش‌بینی‌های بیشتر نسبت به قوانین با پیش‌بینی‌های کمتر دقت بیشتری دارند. به طور خاص، قانونی که هر ویژگی x را درگیر می‌کند، همسایگی ارزیابی شده را به نمونه‌های مشخص کاهاش می‌دهد. بنابراین، مدل همه همسایگان را به طور مساوی طبقه‌بندی می‌کند و دقت قانون آن ۱ است. در عین حال، قانونی که بسیاری از ویژگی‌ها را درگیر می‌کند، بیش از حد خاص است و فقط برای چند نمونه قابل اجرا است. از این رو، بین دقت و پوشش تعادل وجود دارد.

رویکرد لنگرها از چهار جزء اصلی برای یافتن توضیحات استفاده می‌کند.

تولید کاندیدها: این جزء کاندیدهای توضیح جدیدی را ایجاد می‌کند. در دور اول، یک کاندید در هر ویژگی از x ایجاد می‌شود و مقدار مربوطه اختلالات احتمالی را ثابت می‌کند. در هر دور دیگر، بهترین کاندیدهای دور قبلی با یک پیش‌بینی ویژگی که هنوز در آن وجود ندارد، گسترش می‌یابد.

شناسایی بهترین کاندید: قوانین کاندید باید باتوجه به اینکه کدام قانون بهترین توضیح x را می‌دهد با هم مقایسه شوند. برای این منظور، اختلال‌هایی که با قانون مشاهده شده در حال حاضر مطابقت دارند ایجاد و با فراخوانی مدل، ارزیابی می‌شوند. با این حال، این فراخوانی‌ها باید به حداقل برسد تا سربار محاسباتی^۱ محدود شود. به همین دلیل است که در هسته این مؤلفه، یک راهزن چند دست با اکتشاف خالص (به طور دقیق MAB; KL-LUCB Kaufmann and Kalyanakrishnan (2013) وجود دارد. MAB‌ها برای کاوش و بهره‌برداری مؤثر از استراتژی‌های مختلف (که در قیاس با ماشین‌های اسلات، بازو نامیده می‌شوند) با استفاده از انتخاب متوالی استفاده می‌شوند. در تنظیمات داده شده، هر قانون کاندید باید به عنوان بازویی دیده شود که می‌توان آن را کشید. هر بار که کشیده می‌شود، همسایگان مربوطه مورد ارزیابی قرار می‌گیرند، و از این طریق

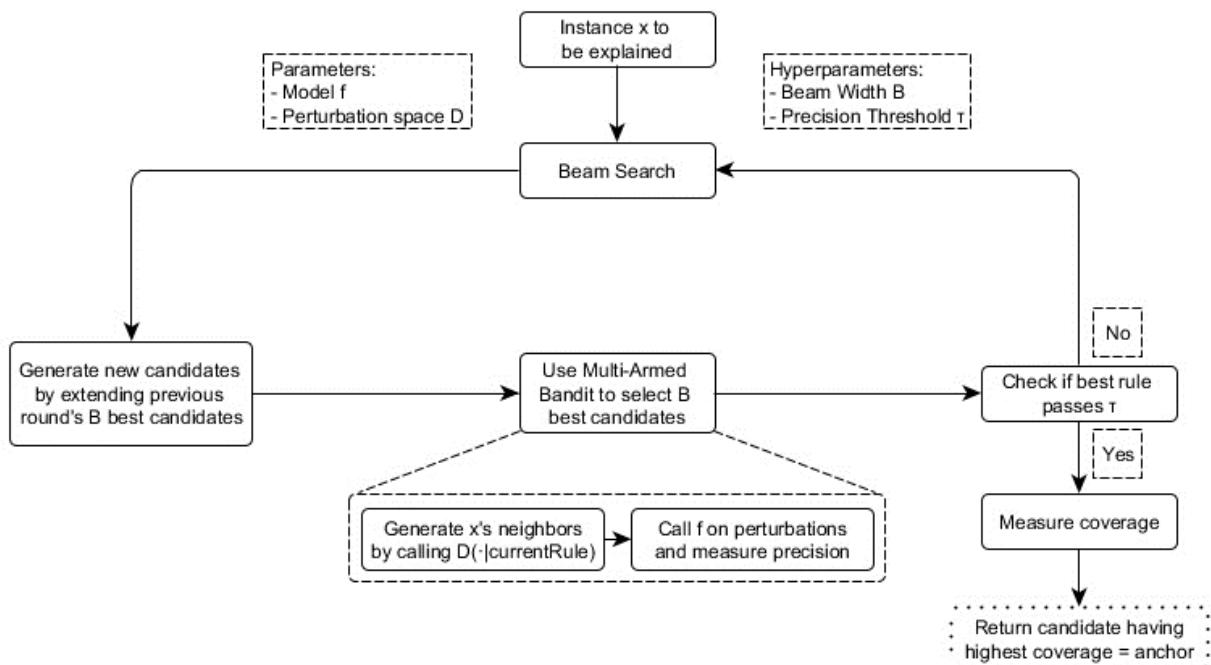
^۱ computational overhead

اطلاعات بیشتری در مورد بازده قانون کاندید به دست می‌آوریم (دقت در مورد لنگر). بنابراین دقต بیان می‌کند که قانون تا چه حد نمونه‌ای را که باید توضیح داده شود، توصیف می‌کند.

اعتبار سنجی دقت کاندید: در صورتی که هنوز اطمینان آماری وجود نداشته باشد که کاندید از آستانه τ عبور کرده است، نمونه‌های بیشتری گرفته می‌شود.

جستجوی ستونی اصلاح شده: همه اجزای فوق در یک جستجوی ستونی جمع‌آوری می‌شوند، که یک الگوریتم جستجوی گراف و گونه‌ای از الگوریتم جستجوی اول سطح^۱ است. این الگوریتم، B تا از بهترین کاندیدهای هر دور را به دور بعد، حمل می‌کند (در اینجا B عرض ستون^۲ نامیده می‌شود). این B بهترین قوانین در ادامه برای ایجاد قوانین جدید استفاده می‌شوند. جستجوی ستونی در بیشتر دورهای این قوانین در ایجاد قوانین جدید استفاده می‌شوند. جستجوی ستونی در $featureCount(x)$ هدایت می‌کند تا هر ویژگی فقط در یک قانون با بیشترین آن‌ها، موجود باشد (هر ویژگی حداقل یک بار می‌تواند در یک قانون گنجانده شود. بنابراین، در هر دور n ، کاندیدهایی با دقیقاً n پیش بینی ایجاد می‌شود و B تا از بهترین‌ها انتخاب می‌شوند. بنابراین با تنظیم B بالا، الگوریتم به احتمال زیاد از گیر افتادن در بهینه محلی اجتناب می‌کند. اما در عوض، به تعداد بالایی از فراخوانی‌های مدل نیاز دارد و در نتیجه بار محاسباتی را افزایش می‌دهد.

این چهار جزء در شکل زیر نشان داده شده است.



¹ breadth-first algorithm

² Beam Width

شکل ۹.۱۲: اجزای الگوریتم لنگرها و روابط درونی آنها (ساده شده)

این رویکرد دستور العمل ظاهراً کاملی برای استخراج کارآمد اطلاعات آماری صحیح در مورد اینکه چرا هر سیستمی یک نمونه را به روشی که انجام می‌داد طبقه‌بندی می‌کند است. به طور سیستماتیک با ورودی مدل آزمایش می‌کند و با مشاهده خروجی‌های مربوطه نتیجه می‌گیرد. برای کاهش تعداد تماس‌های انجام‌شده با مدل، بر روش‌های یادگیری ماشین (MABs) به خوبی ثبت شده و تحقیق شده متکی است. این به نوبه خود، زمان اجرای الگوریتم را به شدت کاهش می‌دهد.

۹.۴.۲ پیچیدگی و زمان اجرا

دانستن رفتار زمان اجرا مجانبی^۱ رویکرد لنگرها به ارزیابی اینکه چقدر انتظار می‌رود در مسائل خاص عملکرد خوبی داشته باشد، کمک می‌کند. اگر B عرض ستون و p تعداد تمام ویژگی‌ها باشد، آن گاه الگوریتم لنگرها، تابع رابطه زیر است:

$$\mathcal{O}(B \cdot p^2 + p^2 \cdot \mathcal{O}_{MAB[B,p,B]})$$

این مرز از هایپرپارامترهای مستقل از مسئله^۲، مانند اطمینان آماری δ انتزاعی می‌شود. نادیده گرفتن هایپرپارامترها به کاهش پیچیدگی مرز کمک می‌کند (برای اطلاعات بیشتر به مقاله اصلی مراجعه کنید). از آنجایی که MAB بهترین B را از کاندیدهای $B.p$ در هر دور استخراج می‌کند، اکثر MAB‌ها و زمان اجرا آنها را ضربدر فاکتور p^2 بیشتر از هر پارامتر دیگری.

بنابراین آشکار می‌شود: کارایی الگوریتم زمانی که ویژگی‌های زیادی وجود دارد کاهش می‌یابد.

۹.۴.۳ مثال داده‌های جدولی

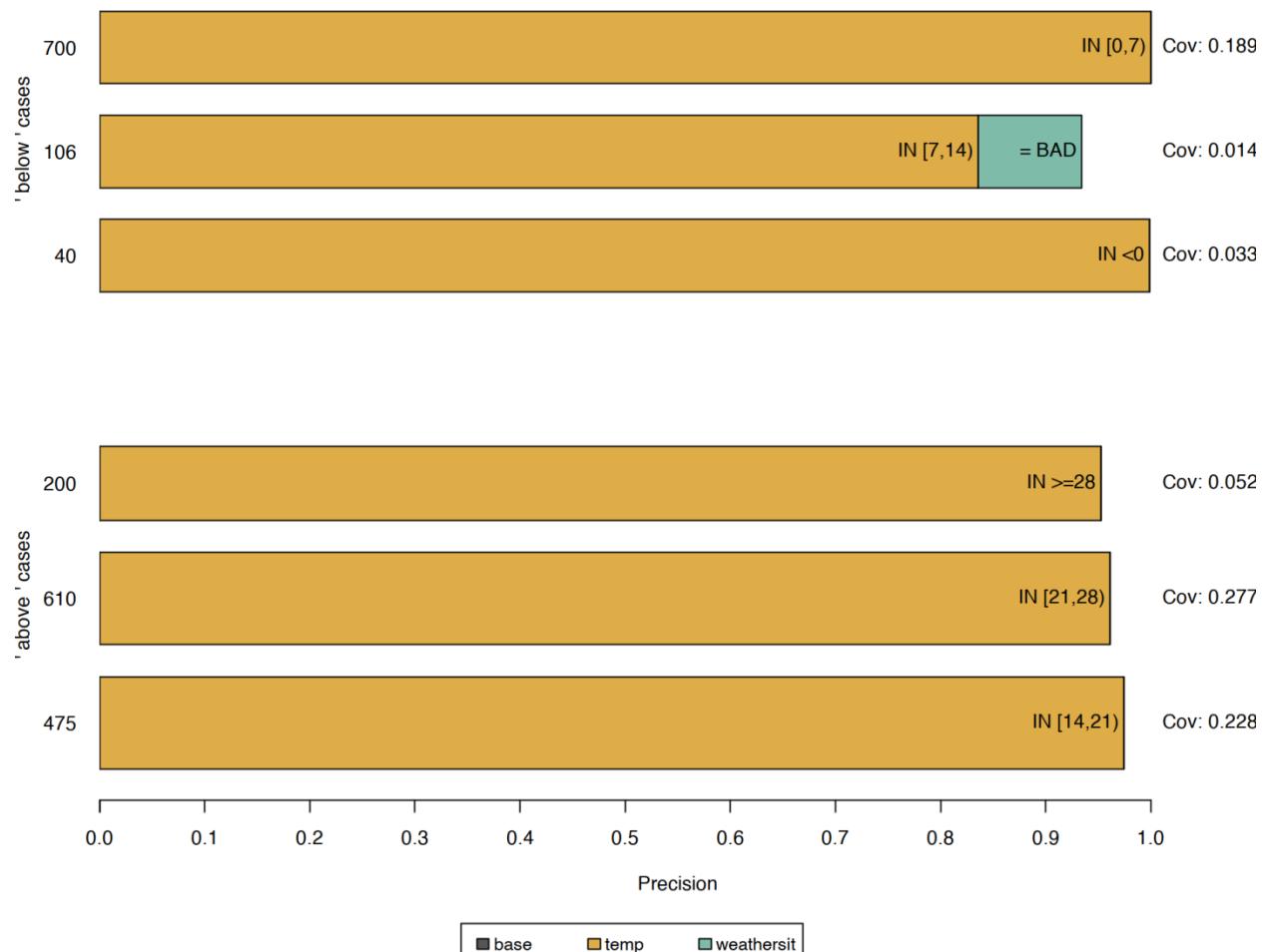
داده‌های جدولی، داده‌های ساختاری هستند که با جداول نشان داده می‌شوند، که در آن ستون‌ها ویژگی‌ها و نمونه‌های ردیف‌ها را نشان می‌دهند. برای مثال، ما از داده‌های اجاره دوچرخه برای نشان دادن پتانسیل رویکرد لنگرها برای توضیح پیش‌بینی‌های ML برای نمونه‌های انتخاب شده استفاده می‌کنیم. برای این، ما رگرسیون را به یک مسئله طبقه‌بندی تبدیل می‌کنیم و یک جنگل تصادفی را به عنوان مدل جعبه سیاه خود آموزش می‌دهیم. این برای طبقه‌بندی است که آیا تعداد دوچرخه‌های کرایه شده بالاتر یا پایین‌تر از خط روند است.

قبل از ایجاد توضیحات لنگر، باید یک تابع اختلال تعریف شود. یک راه آسان برای انجام این کار استفاده از یک فضای اختلال پیش‌فرض شهودی برای موارد توضیح جدولی است که می‌تواند با نمونه‌برداری از داده‌های آموزشی ساخته شود. هنگام ایجاد اختلال در یک نمونه، این رویکرد پیش‌فرض مقادیر ویژگی را که تحت تاثیر

¹ asymptotic runtime

² problem-independent hyperparameters

پیش بینی لنگر هستند حفظ می کند، در حالی که ویژگی های غیر ثابت را با مقادیری که از نمونه های نمونه گیری تصادفی دیگری با احتمال مشخص گرفته شده جایگزین می کند. این فرآیند نمونه های جدیدی را به دست می دهد که مشابه موارد توضیح داده شده هستند، اما مقادیری را از نمونه های تصادفی دیگر اتخاذ کرده اند. بنابراین، آنها شبیه همسایگان نمونه توضیح داده شده هستند.

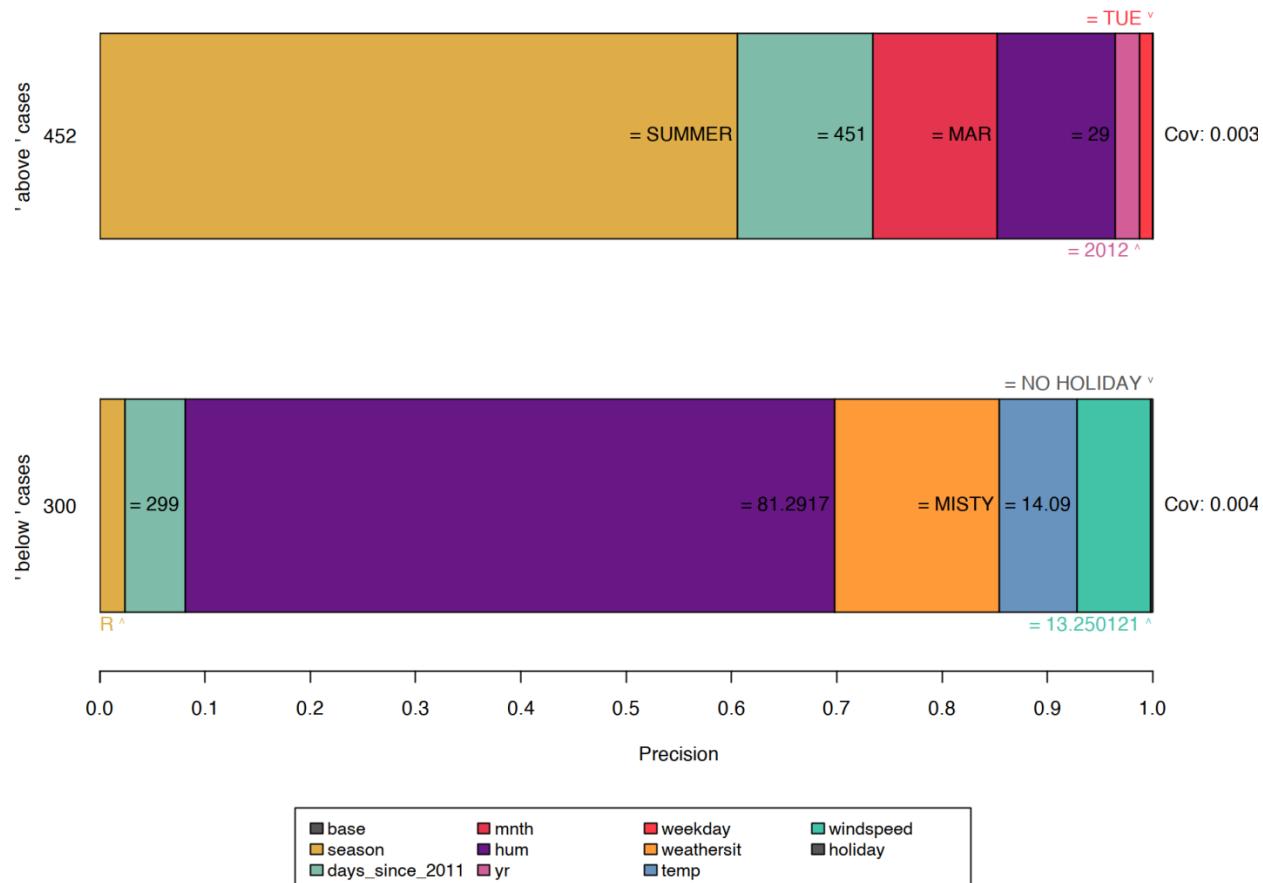


شکل ۹.۱۳: لنگرهایی که شش نمونه از مجموعه داده های اجاره دوچرخه را توضیح می دهند. هر ردیف یک توضیح یا لنگر را نشان می دهد و هر نوار پیش بینی های ویژگی موجود در آن را نشان می دهد. محور x دقت یک قانون را نشان می دهد و ضخامت یک میله با پوشش آن مطابقت دارد. قانون «پایه» هیچ پیش بینی ندارد. این لنگرها نشان می دهند که مدل عمدتاً دما را برای پیش بینی در نظر می گیرد.

نتایج به طور خودبخودی قابل تفسیر هستند و برای هر نمونه توضیح داده شده نشان می دهند که کدام ویژگی برای پیش بینی مدل مهمتر است. از آنجایی که لنگرها فقط دارای چند محمول هستند، علاوه بر این، پوشش بالایی دارند و از این رو در موارد دیگر کاربرد دارند. قوانین نشان داده شده در بالا با $\tau = 0.9$ ایجاد شده است.

بنابراین، ما از لنگرهایی درخواست می‌کنیم که اختلالات ارزیابی شده آنها به طور صادقانه برچسب را با دقت حداقل ۹۰ درصد پشتیبانی کند. همچنین از گسسته سازی برای افزایش بیان و کاربرد ویژگی‌های عددی استفاده شد.

همه قوانین قبلی برای نمونه‌هایی ایجاد شده‌اند که مدل با اطمینان بر اساس چند ویژگی تصمیم می‌گیرد. با این حال، نمونه‌های دیگر به طور مشخص توسط مدل طبقه‌بندی نمی‌شوند زیرا ویژگی‌های بیشتری اهمیت دارند. در چنین مواردی، لنگرها خاص‌تر می‌شوند، ویژگی‌های بیشتری را شامل می‌شوند و برای نمونه‌های کمتری اعمال می‌شوند.



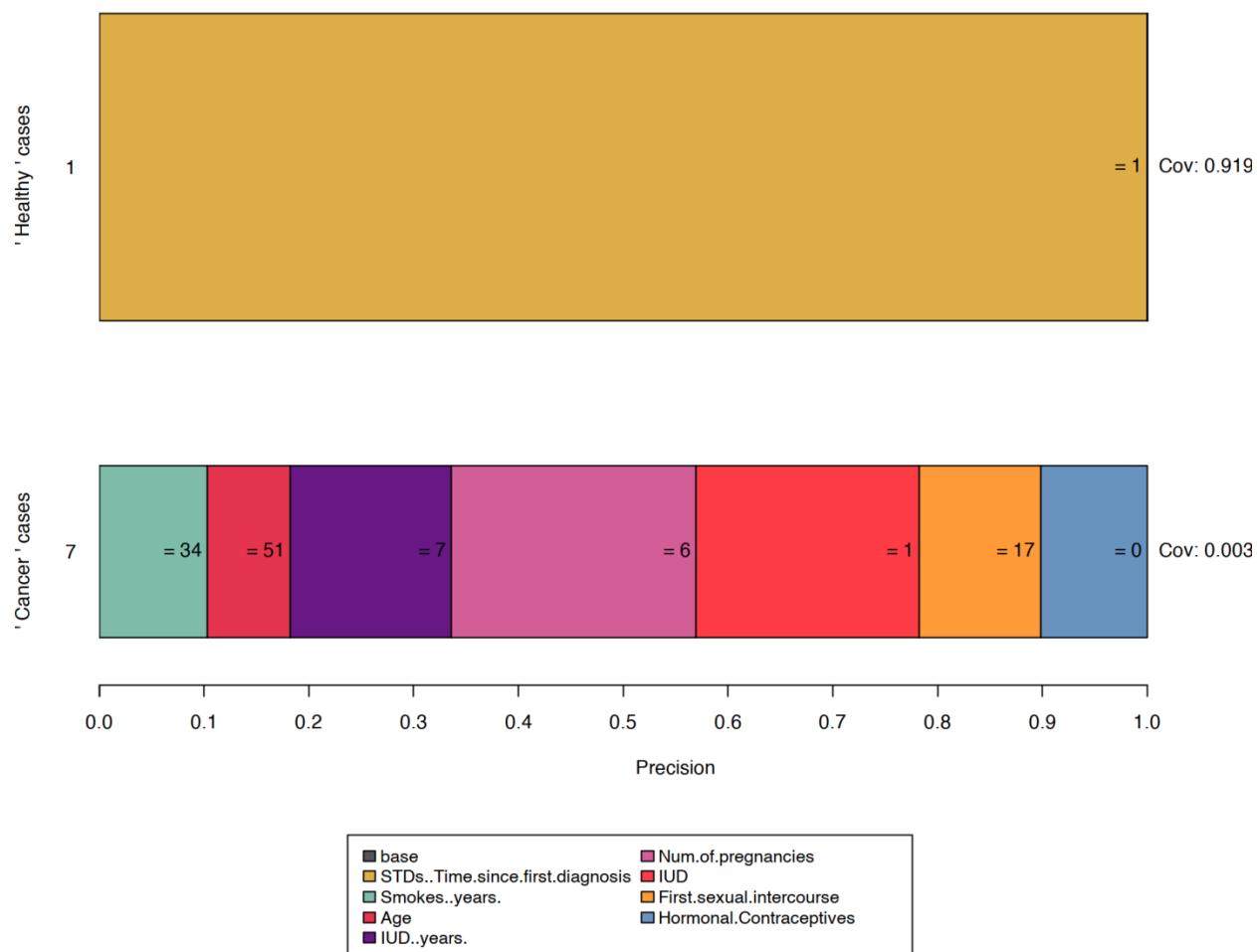
شکل ۹.۱۴: توضیح موارد نزدیک به مرزهای تصمیم منجر به قوانین خاصی می‌شود که شامل تعداد بیشتری از محمولات ویژگی و پوشش کمتر است. همچنین، قانون خالی، یعنی ویژگی پایه، اهمیت کمتری پیدا می‌کند. این می‌تواند به عنوان یک سیگنال برای یک مرز تصمیم تفسیر شود، زیرا نمونه در یک محله فرار قرار دارد. در حالی که انتخاب فضای اختلال پیش‌فرض یک انتخاب راحت است، ممکن است تأثیر زیادی بر الگوریتم داشته باشد و در نتیجه منجر به نتایج مغرضانه شود. به عنوان مثال، اگر مجموعه قطار نامتعادل باشد (تعداد نمونه‌های

نامساوی از هر کلاس وجود دارد، فضای اختلال نیز وجود دارد. این شرایط بیشتر بر قوانین یابی و دقت نتیجه تأثیر می‌گذارد.

مجموعه داده‌های سلطان دهانه رحم یک مثال عالی از این وضعیت است. اعمال الگوریتم لنگرها منجر به یکی از شرایط زیر می‌شود:

-توضیح نمونه‌هایی که برچسب سالم دارند، قوانین خالی را به دست می‌دهد زیرا همه همسایگان تولید شده به سالم ارزیابی می‌کنند.

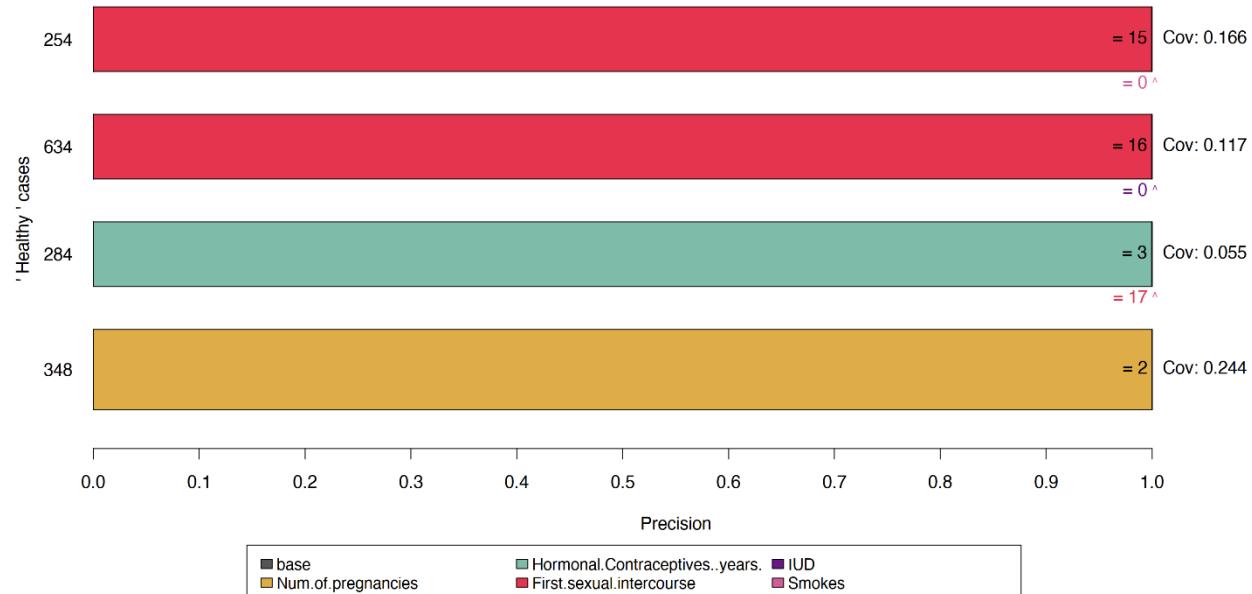
-توضیحات برای نمونه‌هایی که سلطان برچسب‌گذاری شده‌اند، بیش از حد خاص هستند، به عنوان مثال، محموله‌های ویژگی بسیاری را شامل می‌شوند، زیرا فضای اختلال عمدتاً مقادیر نمونه‌های سالم را پوشش می‌دهد.



شکل ۹.۱۵: ساخت لنگرها در فضاهای اختلال نامتعادل منجر به نتایج غیر قابل بیان می‌شود.

این نتیجه ممکن است ناخواسته باشد و می‌توان به روش‌های مختلفی به آن نزدیک شد. به عنوان مثال، یک فضای اختلال سفارشی را می‌توان تعریف کرد. این اختلال سفارشی می‌تواند نمونه‌های متفاوتی داشته باشد، به عنوان مثال از یک مجموعه‌داده نامتعادل یا یک توزیع نرمال. با این حال، این یک عارضه جانبی دارد: همسایگان نمونه‌گیری شده نماینده نیستند و دامنه پوشش را تغییر می‌دهند. از طرف دیگر، می‌توانیم اطمینان MAB را تغییر دهیم δ و مقادیر پارامتر خطای ϵ را باعث می‌شود MAB نمونه‌های بیشتری بکشد و در نهایت منجر به نمونه برداری بیشتر از اقلیت به صورت مطلق می‌شود.

برای این مثال، ما از زیرمجموعه‌ای از مجموعه سرطان دهانه رحم استفاده می‌کنیم که در آن اکثر موارد سرطان برچسب‌گذاری شده‌اند. سپس چارچوبی برای ایجاد یک فضای اختلال مربوطه از آن داریم. اختلالات در حال حاضر بیشتر به پیش‌بینی‌های مختلف منجر می‌شوند و الگوریتم لنگرها می‌تواند ویژگی‌های مهم را شناسایی کند. با این حال، باید تعریف پوشش را در نظر گرفت: این پوشش فقط در فضای اختلال تعریف می‌شود. در مثال‌های قبلی از مجموعه قطار به عنوان پایه فضای اختلال استفاده کردیم. از آنجایی که ما در اینجا فقط از یک زیرمجموعه استفاده می‌کنیم، پوشش بالا لزوماً نشان دهنده اهمیت بالای قانون در سطح جهانی نیست.



شکل ۹.۱۶: متعادل کردن مجموعه داده‌ها قبل از ساختن لنگرها، استدلال مدل را برای تصمیم گیری در موارد اقلیت نشان می‌دهد.

۹.۴.۴ مزايا

رویکرد لنگرها مزاياي متعددی را نسبت به LIME ارائه می‌دهد. اولاً، درک خروجی الگوریتم آسان‌تر است، زیرا قوانین به راحتی قابل تفسیر هستند (حتی برای افراد عادی).

علاوه بر این، لنگرها قابل تنظیم هستند و حتی با گنجاندن مفهوم پوشش، اندازه ای از اهمیت را بیان می کنند. دوم، رویکرد لنگرها زمانی کار می کند که پیش‌بینی‌های مدل غیرخطی یا پیچیده در همسایگی یک نمونه باشند. از آنجایی که این رویکرد به جای برآش مدل‌های جایگزین، تکنیک‌های یادگیری تقویتی را به کار می‌گیرد، احتمال کمتری دارد که مدل را نادیده بگیرد.

جدای از آن، الگوریتم آگنوستیک مدل است و بنابراین برای هر مدلی قابل استفاده است.

علاوه بر این، بسیار کارآمد است زیرا می‌توان با استفاده از MAB هایی که از نمونه برداری دسته ای پشتیبانی می‌کنند (مثلًا BatchSAR) موازی سازی کرد.

۹.۴.۵ معایب

این الگوریتم از یک تنظیم بسیار قابل تنظیم و تأثیرگذار رنج می‌برد، درست مانند اکثر توضیح دهنده‌گان مبتنی بر اختلال. نه تنها فرآپارامترهایی مانند عرض پرتو یا آستانه دقیق باید تنظیم شوند تا نتایج معنی داری به دست آورند، بلکه تابع اختلال نیز باید به صراحت برای یک دامنه/مورد استفاده طراحی شود. به این فکر کنید که چگونه داده‌های جدولی آشفته می‌شوند و به این فکر کنید که چگونه مفاهیم مشابه را در داده‌های تصویری اعمال کنید (نکته: اینها قابل اعمال نیستند). خوشبختانه، رویکردهای پیش‌فرض ممکن است در برخی از حوزه‌ها (مثلًا جدولی) مورد استفاده قرار گیرند، که تنظیم توضیحات اولیه را تسهیل می‌کند.

همچنین، بسیاری از سناریوها نیاز به گسسته سازی دارند، زیرا در غیر این صورت نتایج بسیار خاص هستند، پوشش کمی دارند و به درک مدل کمک نمی‌کنند. در حالی که گسسته سازی می‌تواند کمک کند، اما اگر بی‌دقیق استفاده شود، ممکن است مزه‌های تصمیم را محو کند و در نتیجه دقیقاً اثر معکوس داشته باشد. از آنجایی که بهترین تکنیک گسسته سازی وجود ندارد، کاربران باید قبل از تصمیم گیری در مورد نحوه گسسته سازی داده‌ها برای به دست آوردن نتایج ضعیف از داده‌ها آگاه باشند.

ساختن لنگرها به فراخوانی‌های زیادی به مدل ML نیاز دارد، درست مانند همه توضیح دهنده‌گان مبتنی بر اختلال. در حالی که الگوریتم MAB ها را برای به حداقل رساندن تعداد تماس‌ها مستقر می‌کند، زمان اجرای آن هنوز هم بسیار به عملکرد مدل بستگی دارد و بنابراین بسیار متغیر است.

در نهایت، مفهوم پوشش در برخی حوزه‌ها تعریف نشده است. به عنوان مثال، هیچ تعریف واضح یا جهانی در مورد اینکه چگونه سوپرپیکسل در یک تصویر با آن در سایر تصاویر مقایسه می‌شود، وجود ندارد.

۹.۴.۶ نرم افزار و جایگزین

در حال حاضر، دو پیاده‌سازی در دسترس است anchor ::، یک بسته پایتون (همچنین توسط Alibi یکپارچه شده است) و یک پیاده‌سازی جاوا. اولی مرجع نویسنده‌گان الگوریتم لنگرها و دومی یک پیاده‌سازی با کارایی بالا است که با یک رابط R به نام anchors ارائه می‌شود که برای مثال‌های این فصل استفاده شد. در حال حاضر،

پیاده‌سازی anchors فقط از داده‌های جدولی پشتیبانی می‌کند. با این حال، لنگرها ممکن است از نظر تئوری برای هر دامنه یا نوع داده‌ای ساخته شوند.

۹.۵ مقادیر Shapley

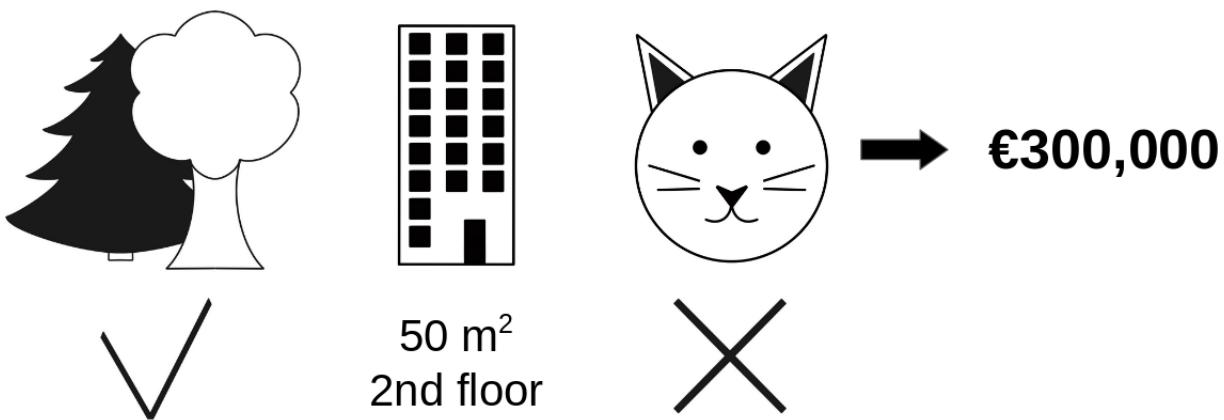
یک پیش‌بینی را می‌توان با این فرض توضیح داد که هر مقدار ویژگی نمونه، یک «بازیکن^۱» در بازی است که در این بازی، پیش‌بینی، پرداخت^۲ است. ارزش Shapley -روشی از تئوری بازی‌های انتلافی^۳ - به ما می‌گوید که چگونه "پرداخت" را به طور عادلانه بین ویژگی‌ها توزیع کنیم.

آیا به دنبال یک کتاب عمیق و کاربردی در مورد ارزش‌های SHAP و Shapley هستید؟ به این کتاب من مراجعه کنید.

۹.۵.۱ ایده کلی

سناریوی زیر را فرض کنید:

شما یک مدل یادگیری ماشین برای پیش‌بینی قیمت آپارتمان آموزش داده اید. برای یک آپارتمان خاص ۳۰۰۰۰۰ یورو پیش‌بینی می‌شود و شما باید این پیش‌بینی را توضیح دهید. آپارتمان دارای مساحت ۵۰ متر مربع است، در طبقه ۲ واقع شده است، دارای پارک در نزدیکی است و گربه ممنوع است:



شکل ۹.۱۷: قیمت پیش‌بینی شده برای ۵۰ مترمربع آپارتمان طبقه دوم با پارک نزدیک و ممنوعیت گربه ۳۰۰۰۰۰ یورو است. هدف ما توضیح این است که چگونه هر یک از این مقادیر ویژگی به پیش‌بینی کمک کرده است.

میانگین پیش‌بینی برای همه آپارتمان‌ها ۳۱۰۰۰۰ یورو است. مقدار هر ویژگی در مقایسه با میانگین پیش‌بینی چقدر در پیش‌بینی نقش داشته است؟

¹ player

² payout

³ coalitional game theory

پاسخ برای مدل‌های رگرسیون خطی ساده است. تأثیر هر ویژگی وزن ویژگی ضربدر مقدار ویژگی است. این فقط به دلیل خطی بودن مدل کار می‌کند. برای مدل‌های پیچیده‌تر، ما به راه حل متفاوتی نیاز داریم. به عنوان مثال، LIME مدل‌های محلی را برای تخمین اثرات پیشنهاد می‌کند. راه حل دیگر از نظریه بازی‌های توافقی^۱ ناشی می‌شود: مقادیر Shapley، که توسط (1953) Shapley ابداع شد، روشی برای تخصیص پرداخت‌ها به بازیکنان بسته به سهم آنها در کل پرداخت است. بازیکنان به صورت توافقی همکاری می‌کنند و از این همکاری سود مشخصی دریافت می‌کنند.

بازیکنان؟ بازی؟ پرداخت؟ ارتباط با پیش‌بینی‌های یادگیری ماشین و قابلیت تفسیر چیست؟ "بازی" وظیفه پیش‌بینی برای یک نمونه واحد از مجموعه‌داده است. "سود"^۲ پیش‌بینی واقعی برای این نمونه منهای میانگین پیش‌بینی میانگین برای همه نمونه‌ها می‌باشد. «بازیکن‌ها» مقادیر ویژگی نمونه‌ای هستند که برای دریافت سود (= مقدار معینی را پیش‌بینی می‌کنند) همکاری می‌کنند. در مثال آپارتمان‌ما، مقادیر ویژگی-
park, nearby, cat-banned, area-50 و floor-2nd برای دستیابی به پیش‌بینی ۳۰۰۰۰۰ یورو با هم کار کردند. هدف ما توضیح تفاوت بین پیش‌بینی واقعی (۳۰۰۰۰۰ یورو) و میانگین پیش‌بینی (۳۱۰۰۰۰ یورو) است: اختلاف ۱۰۰۰۰ یورو.

پاسخ می‌تواند این باشد: floor-2nd park-nearby ۳۰۰۰۰ یورو کمک کرد، area-50 ۱۰۰۰۰ یورو کمک کرد، ۵۰۰۰۰ cat-banned یورو کمک کرد. مجموع کمک‌ها به ۱۰۰۰۰ یورو می‌رسد، که پیش‌بینی نهایی منهای میانگین قیمت آپارتمان پیش‌بینی شده است.

چگونه مقدار Shapley را برای یک ویژگی محاسبه کنیم؟

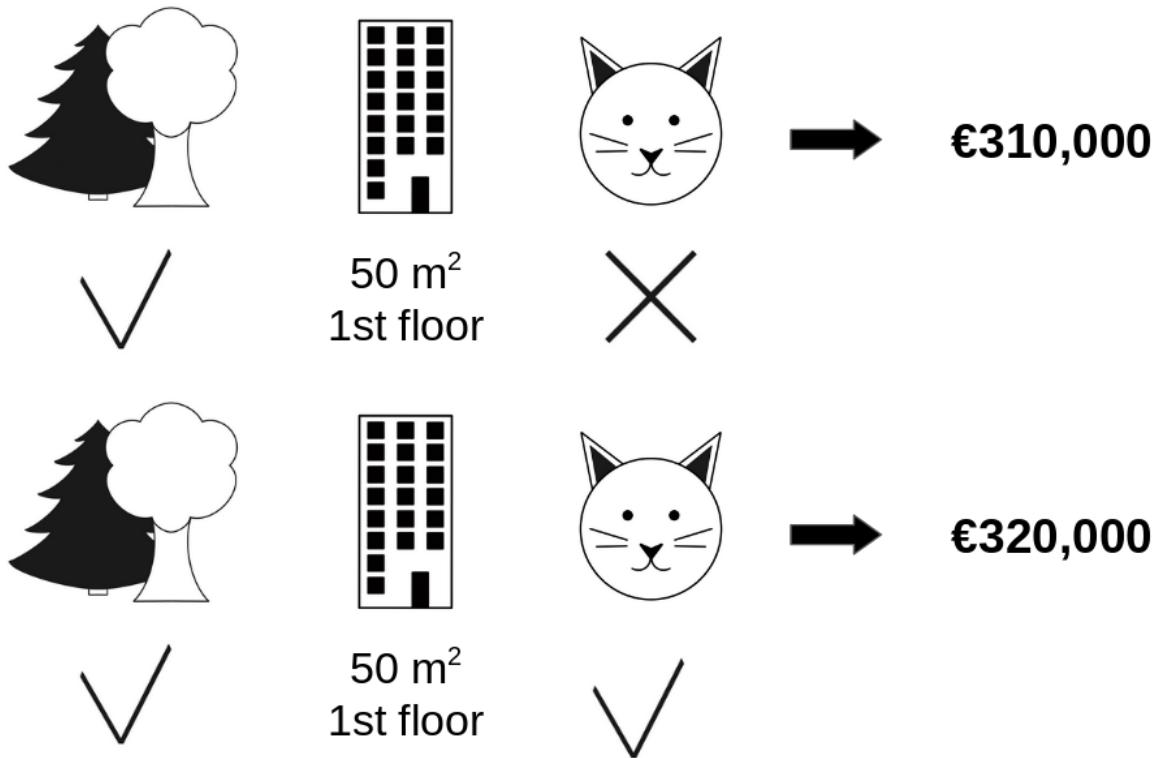
مقدار Shapley میانگین سهم حاشیه‌ای یک مقدار ویژگی در تمام ائتلاف‌های ممکن است. الان همه چی واضحه؟

در شکل زیر سهم مقدار ویژگی cat-banned را هنگامی که به ائتلافی از area-50 و park-nearby اضافه می‌شود، ارزیابی می‌کنیم. ما فقط حالتی را شبیه‌سازی می‌کنیم که park-nearby, area-50 و cat-banned در یک ائتلاف هستند در یک آپارتمان دیگر که به صورت تصادفی از داده‌ها بیرون کشیده شده است و از مقدار ویژگی طبقه آن آپارتمان استفاده گردیده است. مقدار floor-2nd با قرعه تصادفی floor-1st جایگزین شد. سپس قیمت آپارتمان را با این ترکیب پیش‌بینی می‌کنیم (۳۱۰۰۰۰ یورو). در مرحله دوم، ائتلاف حذف می‌شود و با یک مقدار تصادفی از cat allowed/banned در آپارتمان تصادفی بیرون کشیده جایگزین می‌شود. در مثال cat-allowed بود، اما می‌توانست دوباره cat-banned باشد. قیمت آپارتمان را برای

¹ cooperative game theory

² gain

ائتلاف park-nearby و area-50، پیش‌بینی می‌کنیم (۳۲۰۰۰ یورو). سهم cat-banned بود. این تخمین به مقادیر آپارتمانی که به طور تصادفی بیرون کشیده شده است، بستگی دارد که به عنوان «اهداکننده^۱» برای مقادیر ویژگی‌های cat و floor عمل می‌کرد. اگر این مرحله نمونه‌گیری را تکرار کنیم و مشارکت‌ها را میانگین‌گیری کنیم، تخمین‌های بهتری به دست خواهیم آورد.



شکل ۹.۱۸: یک تکرار نمونه برای تخمین سهم cat-banned در پیش‌بینی هنگامی که به ائتلاف park-nearby و area-50 اضافه می‌شود.

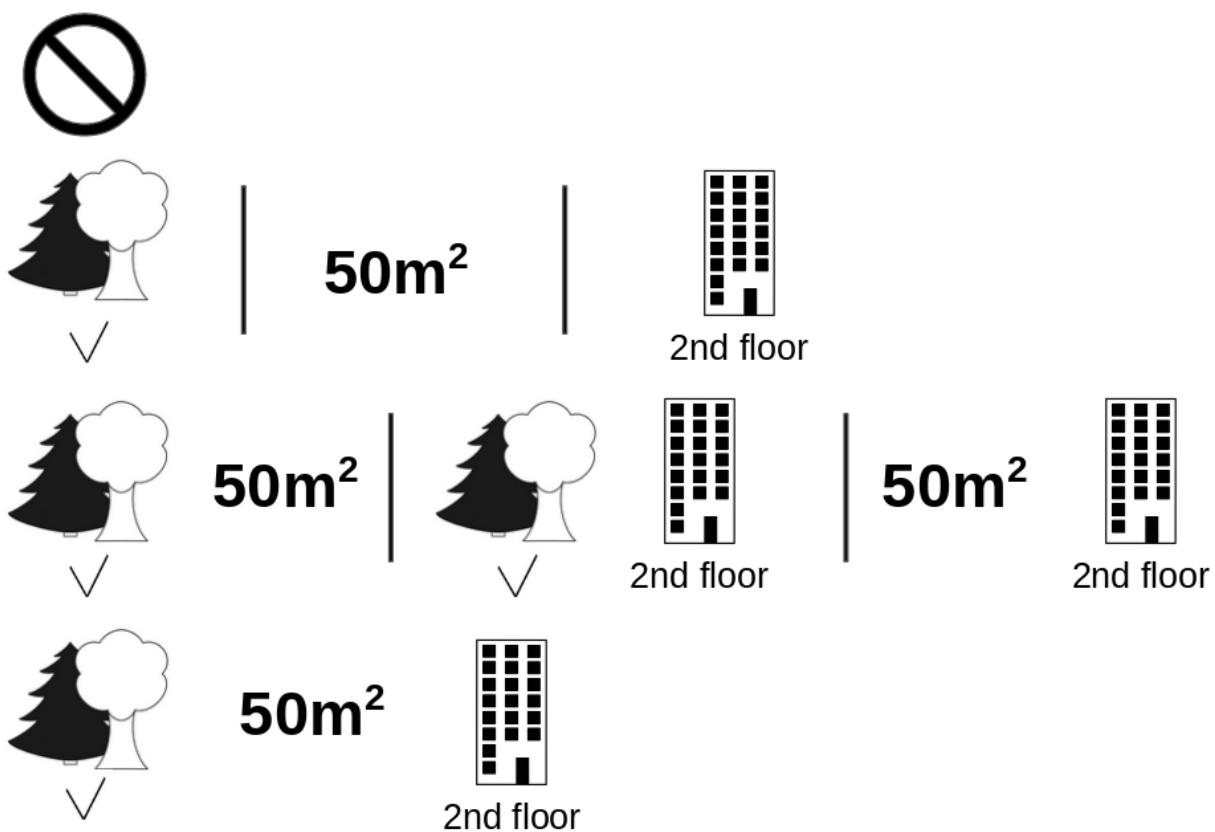
ما این محاسبه را برای همه ائتلاف‌های ممکن تکرار می‌کنیم. مقدار Shapley، میانگین تمام مشارکت‌های حاشیه‌ای برای همه ائتلاف‌های ممکن است. زمان محاسبه با تعداد ویژگی‌ها به طور تصاعدی افزایش می‌یابد. یک راه برای مدیریت زمان محاسبات، محاسبه مشارکت تنها برای چند نمونه از ائتلاف‌های ممکن است.

شکل زیر تمام ائتلاف‌های مقادیر ویژگی که برای تعیین مقدار Shapley برای cat-banned مورد نیاز است را نشان می‌دهد. ردیف اول ائتلاف را بدون هیچ مقدار ویژگی نشان می‌دهد. ردیف‌های دوم، سوم و چهارم ائتلاف‌های متفاوتی را با افزایش اندازه ائتلاف نشان می‌دهند که با «|» از هم جدا شده‌اند. در مجموع، ائتلاف‌های زیر ممکن است:

¹ donor

- No feature values
- park-nearby
- area-50
- floor-2nd
- park-nearby+area-50
- park-nearby+floor-2nd
- area-50+floor-2nd
- park-nearby+ area-50+ floor-2nd.

برای هر یک از این ائتلاف‌ها، قیمت آپارتمان پیش‌بینی شده را با و بدون ارزش ویژگی `cat-banned` محاسبه می‌کنیم و مابه‌التفاوت را می‌گیریم تا سهم حاشیه‌ای را به دست آوریم. مقدار Shapley میانگین (وزنی) مشارکت‌های حاشیه‌ای است. ما مقادیر ویژگی‌هایی را که در ائتلاف نیستند با مقادیر ویژگی تصادفی از مجموعه‌داده آپارتمان جایگزین می‌کنیم تا از مدل یادگیری ماشین پیش‌بینی بگیریم.

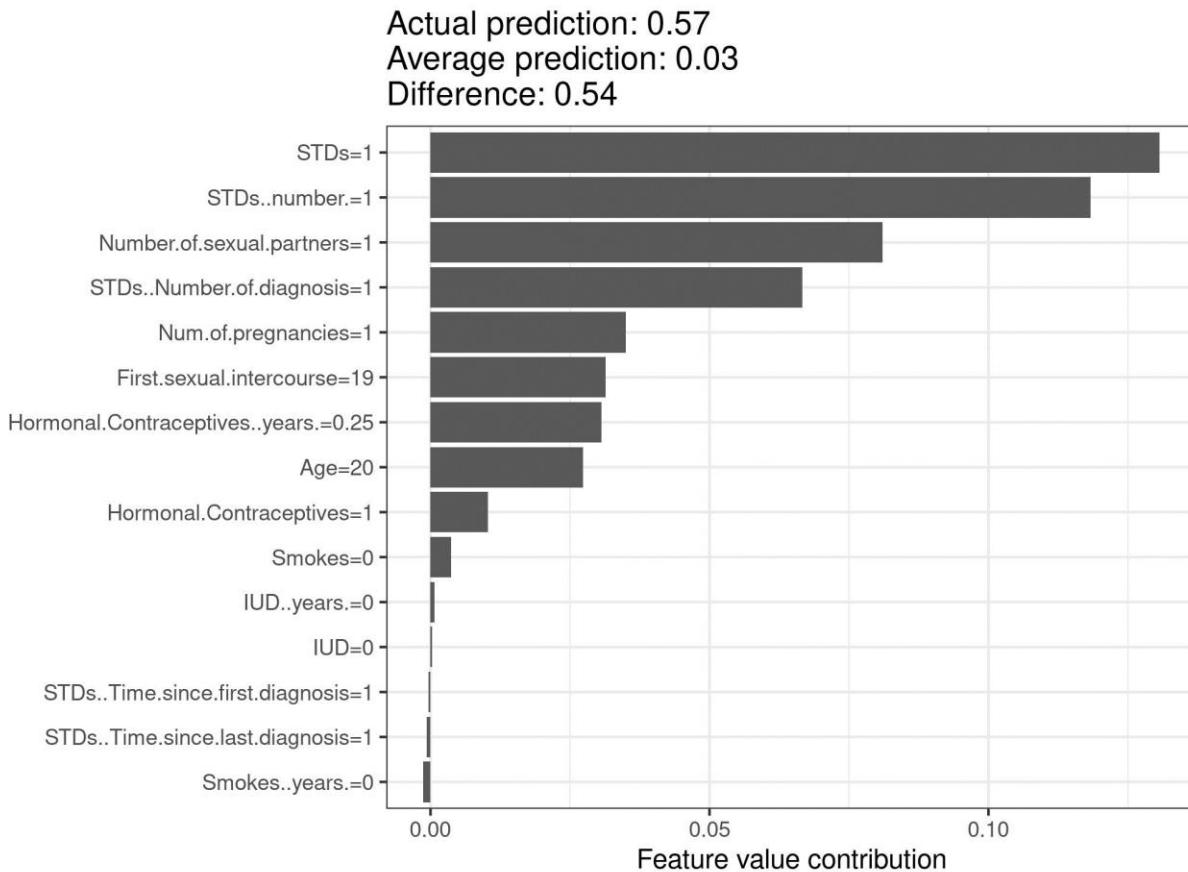


شکل ۹.۱۹: همه ۸ ائتلاف موردنیاز برای محاسبه مقدار دقیق Shapley برای مقدار ویژگی `cat-banned`. اگر مقادیر Shapley را برای همه مقادیر ویژگی تخمین بزنیم، توزیع کامل پیش‌بینی (منهای میانگین) را در بین مقادیر ویژگی به دست می‌آوریم.

۹.۵.۲ مثال‌ها و تفسیر

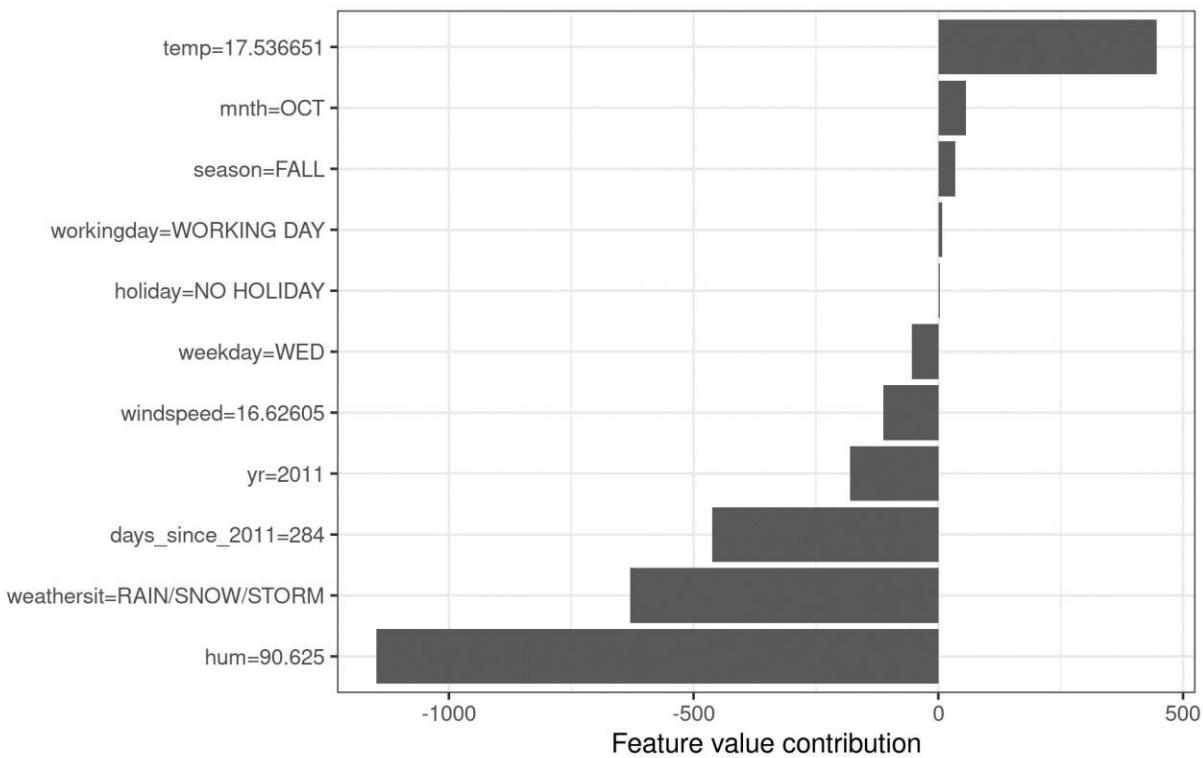
تفسیر مقدار Shapley برای مقدار ویژگی زاین است: مقدار \hat{z} -امین ویژگی به اندازه ϕ به پیش‌بینی این نمونه خاص در مقایسه با میانگین پیش‌بینی مجموعه‌داده مشارکت داشته است.

مقدار Shapley هم برای طبقه‌بندی (اگر با احتمالات سر و کار داشته باشیم) و هم برای رگرسیون کار می‌کند. ما از مقدار Shapley برای تجزیه و تحلیل پیش‌بینی‌های یک مدل جنگل تصادفی که سرطان دهانه رحم را پیش‌بینی می‌کند، استفاده می‌کنیم:



شکل ۹.۲۰: مقادیر Shapley برای یک زن در مجموعه‌داده سرطان دهانه رحم. با پیش‌بینی $0.57/0.03 = 19$ احتمال سرطان این زن $54/54 = 1$ بالاتر از میانگین پیش‌بینی ($0/3$) است. تعداد STD‌های تشخیص داده شده، بیشترین افزایش احتمال را دارد. مجموع مشارکت‌ها تفاوت بین پیش‌بینی واقعی و متوسط ($0.54/0.03 = 18$) را نشان می‌دهد. برای مجموعه داده‌های اجاره دوچرخه، ما همچنین یک جنگل تصادفی را آموزش می‌دهیم تا با توجه به اطلاعات آب و هوای تقویم، تعداد دوچرخه‌های اجاره‌ای را برای یک روز پیش‌بینی کند. توضیحات ایجاد شده برای پیش‌بینی تصادفی جنگل یک روز خاص:

Actual prediction: 2409
 Average prediction: 4518
 Difference: -2108



شکل ۹.۲۱: مقادیر Shapley برای روز ۲۸۵ با پیش‌بینی ۲۴۰۹ دوچرخه اجاره‌ای، این روز ۲۱۰۸ کمتر از میانگین پیش‌بینی (۴۵۱۸) است. وضعیت آب و هوا و رطوبت بیشترین سهم منفی را داشتند. دمای هوا در این روز سهم مثبتی داشت. مجموع مقادیر Shapley تفاوت پیش‌بینی واقعی و میانگین (-۲۱۰۸) را نشان می‌دهد. مراقب باشید که مقدار Shapley را به درستی تفسیر کنید: مقدار Shapley مشارکت متوسط یک مقدار ویژگی در پیش‌بینی در ائتلاف‌های مختلف است. مقدار Shapley تفاوت در پیش‌بینی زمانی که ویژگی را از مدل حذف کنیم، نیست.

۹.۵.۳ جزئیات مقدار Shapley

این بخش بیشتر به تعریف و محاسبه مقدار Shapley برای خواننده کنجدکاو می‌پردازد. اگر به جزئیات فنی علاقه ندارید، از این بخش رد شوید و مستقیماً به "مزایا و معایب" بروید.
 ما علاقه مندیم که ببینیم هر ویژگی چگونه بر پیش‌بینی یک نقطه داده تأثیر می‌گذارد. در یک مدل خطی محاسبه اثرات فردی آسان است. پیش‌بینی مدل خطی برای یک نمونه داده این گونه است:

$$\hat{f}(x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

که در آن x نمونه‌ای است که می‌خواهیم مشارکت‌ها را برای آن محاسبه کنیم، هر j یک مقدار ویژگی است، با $p = 1, \dots, j$. این j وزن مربوط به ویژگی j است.

سهم ϕ از ویژگی j ام در پیش‌بینی $(\hat{f}(x))$ است:

$$\phi_j(\hat{f}) = \beta_j x_j - E(\beta_j X_j) = \beta_j x_j - \beta_j E(X_j)$$

در این رابطه $E(\beta_j X_j)$ برآورد میانگین اثر برای ویژگی j است. مشارکت، تفاوت بین اثر ویژگی منهای اثر متوسط است. خوب! اکنون می‌دانیم که هر ویژگی چقدر در پیش‌بینی مشارکت داشته است. اگر تمام مشارکت‌های ویژگی را برای یک نمونه جمع کنیم، نتیجه به شرح زیر است:

$$\begin{aligned} \sum_{j=1}^p \phi_j(\hat{f}) &= \sum_{j=1}^p (\beta_j x_j - E(\beta_j X_j)) \\ &= \left(\beta_0 + \sum_{j=1}^p \beta_j x_j \right) - \left(\beta_0 + \sum_{j=1}^p E(\beta_j X_j) \right) \\ &= \hat{f}(x) - E(\hat{f}(x)) \end{aligned}$$

این مقدار پیش‌بینی شده برای نقطه داده x منهای میانگین مقدار پیش‌بینی شده است. مشارکت ویژگی می‌تواند منفی باشد.

آیا می‌توانیم برای هر مدلی همین کار را انجام دهیم؟ بسیار عالی خواهد بود که این ابزار، آگنوستیک مدل باشد. از آنجایی که معمولاً در مدل‌های دیگر چیزی مشابه وزن نداریم، به راه حل متفاوتی نیاز داریم. کمک از مکان‌های غیرمنتظره می‌آید: نظریه بازی‌های توافقی. مقدار Shapley راه حلی برای محاسبه مشارکت ویژگی‌ها برای پیش‌بینی‌های منفرد برای هر مدل یادگیری ماشین است.

۹.۵.۳.۱ Shapley مقدار

مقدار Shapley از طریق یکتابع مقدار val از بازیکنان در S تعریف می‌شود. مقدار Shapley یک مقدار ویژگی، مشارکت آن در پرداخت است، وزن‌دهی شده و جمع‌بندی شده بر روی تمام ترکیب‌های مقدار ویژگی ممکن:

$$\phi_{(val)} = \sum_{S \subseteq \{1, \dots, p\} \setminus \{j\}} \frac{|S|! (p - |S| - 1)!}{p!} (val(S \cup \{j\}) - val(S))$$

که در آن S زیرمجموعه‌ای از ویژگی‌های استفاده شده در مدل است، x بردار مقادیر ویژگی نمونه‌ای که توضیح داده می‌شود و p تعداد ویژگی‌ها است. $val_x(S)$ پیش‌بینی مقادیر ویژگی در مجموعه S است که نسبت به ویژگی‌هایی که در مجموعه S گنجانده نشده‌اند به صورا حاشیه‌ای تجمعی شده‌اند:

$$val_x(S) = \int \hat{f}(x_1, \dots, x_p) d\mathbb{P}_{x \notin S} - E_X(\hat{f}(X))$$

شما در واقع چندین انتگرال را برای هر ویژگی که در S نیست انجام می‌دهید. یک مثال عینی: مدل یادگیری ماشین با 4 ویژگی x_1, x_2, x_3 و x_4 کار می‌کند و ما پیش‌بینی ائتلاف S متشکل از مقادیر ویژگی x_1 و x_3 را ارزیابی می‌کنیم:

$$val_x(S) = val_x(\{1,3\}) = \int_{\mathbb{R}} \int_{\mathbb{R}} \hat{f}(x_1, X_2, x_3, X_4) d\mathbb{P}_{X_2 X_4} - E_X(\hat{f}(X))$$

این رابطه شبیه به مشارکت‌های ویژگی در مدل خطی به نظر می‌رسد!

با کاربردهای زیاد کلمه "مقدار" گیج نشوید: مقدار ویژگی، مقداری عدد یا مقداری طبقه‌ای از یک نمونه است. مقدار Shapley مشارکت ویژگی در پیش‌بینی است.تابع مقدار، تابع پرداخت برای ائتلاف بازیکنان (مقادیر ویژگی) است.

مقدار Shapley تنها روش انتسابی است که ویژگی‌های بازده^۱، تقارن^۲، مجازی^۳ و افزودنی^۴ را دارا می‌شد که وجود این صفات در کنار یکدیگر را می‌توان پرداخت منصفانه در نظر گرفت.
بازده: مشارکت ویژگی‌ها باید با اختلاف پیش‌بینی برای x و میانگین جمع شود.

$$\sum_{j=1}^p \phi_j = \hat{f}(x) - E_X(\hat{f}(X))$$

تقارن: مشارکت دو مقدار ویژگی j و k باید یکسان باشد اگر مشارکت مساوی در همه ائتلاف‌های ممکن داشته باشند. اگر

$$val(S \cup \{j\}) = val(S)$$

برای همه

$$S \subseteq \{1, \dots, p\} \setminus \{j\}$$

سپس

$$\phi_j = \phi_k$$

مجازی: ویژگی j که مقدار پیش‌بینی شده را تغییر نمی‌دهد - صرفنظر از اینکه به کدام ائتلاف مقادیر ویژگی اضافه می‌شود - باید مقدار Shapley صفر داشته باشد.

$$val(S \cup \{j\}) = val(S)$$

برای همه

$$S \subseteq \{1, \dots, p\}$$

¹ Efficiency

² Symmetry

³ Dummy

⁴ Additivity

$$\phi_j = 0$$

افزودنی: برای یک بازی با پرداختهای ترکیبی $val + val^+$ مقادیر Shapley مربوطه به شرح زیر است:

$$\phi_j + \phi_j^+$$

فرض کنید شما یک جنگل تصادفی را آموزش داده اید، به این معنی که پیش‌بینی، میانگین تعداد زیادی درخت تصمیم است. ویژگی افزودنی تضمین می‌کند که برای یک مقدار ویژگی، می‌توانید مقدار Shapley را برای هر درخت به صورت جداگانه محاسبه کنید، آنها را میانگین بگیرید و مقدار Shapley را برای مقدار ویژگی برای جنگل تصادفی بدست آورید.

۹.۵.۳.۲ شهود

یک روش شهودی برای درک مقدار Shapley، این تصور است: مقادیر ویژگی به ترتیب تصادفی وارد اتاق می‌شوند. همه مقادیر ویژگی در اتاق در بازی شرکت می‌کنند (= به پیش‌بینی کمک می‌کنند). مقدار Shapley یک مقدار ویژگی، میانگین تغییر در پیش‌بینی است که ائتلاف موجود در اتاق زمانی که مقدار ویژگی به آنها می‌پیوندد، دریافت می‌کند.

۹.۵.۳.۳ Shapley برآورد ارزش

همه ائتلاف‌ها (مجموعه‌های) ممکن از مقادیر ویژگی باید با و بدون ویژگی z -ام ارزیابی شوند تا مقدار دقیق محاسبه شود. برای مسائل با بیش از چند ویژگی، محاسبه راه حل دقیق مساله، مشکل ساز می‌شود زیرا تعداد ائتلاف‌های احتمالی به طور تصاعدی افزایش می‌یابد و ویژگی‌های بیشتری اضافه می‌شود. Štrumbelj (2014) and Kononenko روشهای تقریبی را با نمونه گیری مونت کارلو^۱ پیشنهاد می‌کند:

$$\phi = \frac{1}{M} \sum_{m=1}^M \left(\hat{f}(x_{+j}^m) - \hat{f}(x_{-j}^m) \right)$$

در این رابطه $\hat{f}(x_{+j}^m)$ پیش‌بینی x است، اما با تعدادی تصادفی از مقادیر ویژگی که با مقادیر ویژگی از نقطه داده تصادفی z جایگزین شده‌اند، به جز مقدار مربوط به ویژگی z . بردار x_{-j}^m تقریباً مشابه x_{+j}^m است، اما مقدار x_j^m همچنین از نمونه z گرفته شده است. هر یک از این M نمونه جدید نوعی «هیولای فرانکشتاین»^۲ است که از دو نمونه مونتاژ شده است. توجه داشته باشید که در الگوریتم زیر، ترتیب ویژگی‌ها عملأً تغییر نمی‌کند - هر ویژگی وقتی به تابع پیش‌بینی منتقل می‌شود در همان موقعیت برداری باقی می‌ماند. در اینجا از ترتیب فقط به عنوان یک «ترفند» استفاده می‌شود: با دادن یک ترتیب جدید به ویژگی‌ها، یک مکانیسم تصادفی بدست می‌آوریم که به ما کمک می‌کند «هیولای فرانکشتاین» را کنار هم قرار دهیم. برای ویژگی‌هایی که در سمت

¹ Monte-Carlo sampling

² Frankenstein's Monster

چپ ویژگی x_j ظاهر می‌شوند، مقادیر را از مشاهدات اصلی می‌گیریم و برای ویژگی‌های سمت راست، مقادیر را از یک نمونه تصادفی می‌گیریم.

تخمین تقریبی Shapley برای مقدار ویژگی تک:

- خروجی: مقدار Shapley برای مقدار ویژگی j -ام

موردنیاز: تعداد تکرار M ، نمونه موردعالقه x ، شاخص ویژگی z ، ماتریس داده X و مدل یادگیری ماشین f

$$\circ \quad \text{برای همه } m = 1, \dots, M$$

▪ نمونه تصادفی z را از ماتریس داده X بیرون بکشید

▪ یک جایگشت تصادفی z از مقادیر ویژگی را انتخاب کنید

▪ نمونه x را مرتب کنید: $(x_{(1)}, \dots, x_{(j)}, \dots, x_{(p)})$

▪ نمونه z را مرتب کنید: $(z_{(1)}, \dots, z_{(j)}, \dots, z_{(p)})$

▪ دو نمونه جدید بسازید

$$\bullet \quad x_{+j} = (x_{(1)}, \dots, x_{(j-1)}, x_{(j)}, z_{(j+1)}, \dots, z_{(p)})$$

$$\bullet \quad x_{-j} = (x_{(1)}, \dots, x_{(j-1)}, z_{(j)}, z_{(j+1)}, \dots, z_{(p)})$$

▪ مشارکت حاشیه ای را محاسبه کنید: $\phi_j^m = \hat{f}(x_{+j}) - \hat{f}(x_{-j})$

$$\bullet \quad \text{مقدار Shapley با میانگین گیری محاسبه کنید: } \phi_j(x) = \frac{1}{M} \sum_{m=1}^M \phi_j^m$$

ابتدا یک نمونه موردعالقه x ، یک ویژگی z و تعداد تکرار M را انتخاب کنید. برای هر تکرار، یک نمونه تصادفی z از داده‌ها انتخاب شده و ترتیب تصادفی ویژگی‌ها تولید می‌شود. دو نمونه جدید با ترکیب مقادیر از نمونه موردعالقه x و نمونه z ایجاد می‌شود. نمونه x_{+j} نمونه موردعالقه است، اما تمام مقادیر به ترتیب بعد از ویژگی z با مقادیر ویژگی از نمونه z جایگزین می‌شوند. نمونه x_{-j} مشابه x_{+j} می‌باشد، اما ویژگی z با مقدار ویژگی z از نمونه z جایگزین شده است. تفاوت پیش‌بینی با استفاده از جعبه سیاه محاسبه می‌شود:

$$\phi_j^m = \hat{f}(x_{+j}^m) - \hat{f}(x_{-j}^m)$$

همه این تفاوت‌ها میانگین گیری می‌شوند و به این نتیجه می‌رسند:

$$\phi_j(x) = \frac{1}{M} \sum_{m=1}^M \phi_j^m$$

میانگین گیری به طور ضمنی نمونه‌ها را با توزیع احتمال X وزن دهی می‌کند.

این روش باید برای هر یک از ویژگی‌ها تکرار شود تا تمام مقادیر Shapley به دست آید.

۹.۵.۴ مزایا

تفاوت بین پیش‌بینی و پیش‌بینی میانگین به طور منصفانه بین مقادیر ویژگی نمونه توزیع می‌شود – (خاصیت بازده مقادیر Shapley). این ویژگی مقدار Shapley را از سایر روش‌ها مانند LIME متمایز می‌کند. LIME تضمین نمی‌کند که پیش‌بینی به طور منصفانه بین ویژگی‌ها توزیع شده است. مقدار Shapley ممکن است تنها روش برای ارائه توضیح کامل باشد. در شرایطی که قانون مستلزم تبیین پذیری است - مانند "حق توضیح" اتحادیه اروپا^۱ – ارزش Shapley ممکن است تنها روش سازگار قانونی باشد، زیرا مبتنی بر یک نظریه محکم است و تأثیرات را به طور منصفانه توزیع می‌کند. من وکیل نیستم، بنابراین این فقط شهود من را در مورد الزامات معنکس می‌کند.

مقدار Shapley می‌تواند منجر به توضیح مقابله‌ای شود. به جای مقایسه یک پیش‌بینی با میانگین پیش‌بینی کل مجموعه‌داده، می‌توانید آن را با یک زیر مجموعه یا حتی با یک نقطه داده مقایسه کنید. این تقابلی است که مدل‌های محلی مانند LIME ندارند.

مقدار Shapley تنها روش توضیحی با یک تئوری استوار است. اصول - بازده، تقارن، مجازی، افزودمی - به توضیح یک پایه معقول می‌دهد. روش‌هایی مانند LIME رفتار مدل یادگیری ماشین را به صورت محلی، خطی فرض می‌کنند، اما هیچ نظریه‌ای مبنی بر اینکه چرا این موضوع باید برقرار باشد، وجود ندارد. توضیح دادن یک پیش‌بینی به عنوان یک بازی که توسط مقادیر ویژگی انجام می‌شود، شگفت‌آور است.

۹.۵.۵ معایب

مقدار Shapley به زمان محاسباتی زیادی نیاز دارد. در ۹۹.۹ درصد از مسائل دنیای واقعی، تنها راه حل تقریبی امکان‌پذیر است. محاسبه دقیق مقدار Shapley از نظر محاسباتی گران است زیرا ^{۲k} ائتلاف‌های احتمالی از مقادیر ویژگی وجود دارد و «عدم وجود» یک ویژگی باید با بیرون کشیدن نمونه‌های تصادفی شبیه‌سازی شود، که واریانس تخمین مقادیر Shapley را افزایش می‌دهد. تعداد تصاعدی ائتلافها با نمونه برداری از ائتلافها و محدود کردن تعداد تکرارها انجام می‌شود. کاهش M زمان محاسبه را کاهش می‌دهد، اما واریانس مقدار Shapley را افزایش می‌دهد. هیچ قانون ساده خوبی برای تعداد تکرارها M وجود ندارد. M باید به اندازه کافی بزرگ باشد تا مقادیر Shapley را دقیقاً تخمین بزند، اما به اندازه کافی کوچک باشد تا محاسبات را در یک زمان معقول کامل کند. انتخاب M می‌تواند بر اساس مرزهای Chernoff باشد، اما من هیچ مقاله‌ای در مورد انجام این کار برای مقادیر Shapley برای پیش‌بینی‌های یادگیری ماشین ندیده‌ام.

¹ EU's "right to explanations"

مقدار Shapley را می‌توان به اشتباه تفسیر کرد. مقدار Shapley یک مقدار ویژگی، تفاوت مقدار پیش‌بینی شده پس از حذف ویژگی از آموزش مدل نیست. تفسیر مقدار Shapley این است: با توجه به مجموعه فعلی مقادیر ویژگی، مشارکت یک مقدار ویژگی در تفاوت بین پیش‌بینی واقعی و میانگین پیش‌بینی، مقدار تخمینی Shapley است.

اگر به دنبال توضیحات محدود هستید (توضیحاتی که ویژگی‌های کمی‌دارند)، مقدار Shapley روش توضیح اشتباهی است. توضیحات ایجاد شده با روش مقدار Shapley همیشه از تمام ویژگی‌ها استفاده می‌کنند. انسان‌ها توضیحات انتخابی را ترجیح می‌دهند، مانند آنچه توسط LIME ارائه شده است. LIME ممکن است انتخاب بهتری برای توضیحاتی باشد که افراد غیرمتخصص باید با آن سروکار دارند. راه حل دیگر SHAP است که توسط Lundberg and Lee (2017) معرفی شده است که بر اساس مقدار Shapley است، اما می‌تواند توضیحاتی را با ویژگی‌های کمی ارائه دهد.

مقدار Shapley یک مقدار ساده برای هر ویژگی برمی‌گردد، اما هیچ مدل پیش‌بینی مانند LIME ندارد. این بدان معناست که نمی‌توان از آن برای بیان تغییرات در پیش‌بینی در ازای تغییرات در ورودی استفاده کرد، مانند: "اگر سالانه ۳۰۰ یورو بیشتر درآمد داشته باشم، امتیاز اعتباری من ۵ امتیاز افزایش می‌یابد".

یکی دیگر از معایب این است که اگر می‌خواهید مقدار Shapley را برای یک نمونه داده جدید محاسبه کنید، نیاز به دسترسی به داده‌ها دارید. دسترسی به تابع پیش‌بینی کافی نیست، زیرا به داده‌ها نیاز دارید تا بخش‌هایی از نمونه موردعلاجه را با مقادیر نمونه‌های تصادفی داده‌ها جایگزین کنید. تنها در صورتی می‌توان از این امر جلوگیری کرد که بتوانید نمونه‌های داده ای ایجاد کنید که شبیه نمونه‌های داده واقعی هستند اما نمونه‌های واقعی از داده‌های آموزشی نیستند.

مانند بسیاری از روش‌های دیگر تفسیر مبتنی بر جایگشت، روش مقدار Shapley وقتی ویژگی‌ها با هم همبستگی دارند دارای مشکل نمونه‌های غیرواقعی می‌باشد. برای شبیه سازی وقتی که یک مقدار ویژگی در یک ائتلاف وجود ندارد، ویژگی را به صورت حاشیه ای تجمعی می‌کنیم. این کار با نمونه برداری از مقادیر توزیع حاشیه ای ویژگی به دست می‌آید. این روش تا زمانی که ویژگی‌ها مستقل باشند خوب است. وقتی ویژگی‌ها وابسته هستند، ممکن است مقادیر ویژگی را که برای این نمونه معنی ندارند نمونه برداری کنیم. اما ما از آنها برای محاسبه مقدار Shapley ویژگی استفاده می‌کنیم. یک راه حل ممکن است این باشد که ویژگی‌های همبسته را با هم جابجا کنید و یک مقدار Shapley دوسره^۱ برای آنها به دست آورید. راه حل دیگر، نمونه گیری مشروط است: ویژگی‌ها مشروط به ویژگی‌هایی که قبلاً در تیم هستند، نمونه برداری می‌شوند. در حالی که نمونه گیری مشروط مشکل نقاط داده غیرواقعی را برطرف می‌کند، یک مشکل جدید بوجود می‌آید: مقادیر

¹ mutual

حاصل دیگر مقادیر Shapley برای بازی ما نیستند، زیرا آنها اصل تقارن را نقض می‌کنند. همان‌طور که Sundararajan and Najmi (2020) بیان نمود و بیشتر توسط Janzing et al (۲۰۲۰) مورد بحث قرار گرفت.

۹.۵.۶ نرم افزار و جایگزین

مقادیر Shapley در هر دو پکیج iml^۱ و fastshap^۲ نرم افزار R پیاده‌سازی شده است. در Julia، می‌توانید از Shapley.jl² استفاده کنید.

SHAP، یک روش تخمین جایگزین برای مقادیر Shapley می‌باشد و در بخش بعدی ارائه شده است. breakdown رویکرد دیگر breakdown نام دارد که در پکیج R نرم افزار Shapley پیاده‌سازی شده است (Staniak & Biecek, 2018). این روش نیز مشارکت هر ویژگی را در پیش‌بینی نشان می‌دهد، اما آنها را گام به گام محاسبه می‌کند. اجازه دهید از قیاس بازی دوباره استفاده کنیم: ما با یک تیم خالی شروع می‌کنیم، مقدار ویژگی را که بیشترین مشارکت را در پیش‌بینی دارد اضافه می‌کنیم و تا زمانی که همه مقادیر ویژگی اضافه شوند، این کار تکرار می‌کنیم. این که هر مقدار ویژگی چقدر کمک می‌کند به مقادیر ویژگی مربوطه بستگی دارد که قبلًا در "تیم" وجود دارد، که اشکال بزرگ روش breakdown است. این روش سریعتر از روش Shapley است و برای مدل‌های بدون تعامل، نتایج یکسانی ارائه می‌دهد.

¹ <https://github.com/bgreenwell/fastshap>

² <https://gitlab.com/ExpandingMan/Shapley.jl>

SHAP^۱ ۹.۶

SHAP توسط Lundberg and Lee (2017) روشی برای توضیح پیش‌بینی‌های فردی است. SHAP بر اساس بازی از لحاظ تئوریک بهینه مقادیر Shapley است.

به دنبال یک کتاب عمیق و کاربردی در مورد ارزش‌های SHAP و Shapley هستید؟ من شما را تحت پوشش قرار دادم.

دو دلیل وجود دارد که SHAP بخش خودش را دارد و زیربخش مقادیر Shapley نیست. دلیل اول، نویسنده‌گان KernelSHAP را برای جایگزینی مقادیر Shapley پیشنهاد کردند که رویکردی تخمینی مبتنی بر هسته می‌باشد و از مدل‌های جایگزین محلی الهام گرفته است. نویسنده‌گان TreeSHAP را پیشنهاد کردند، که یک رویکرد تخمین موثر برای مدل‌های مبتنی بر درخت می‌باشد. دلیل دوم، اینکه SHAP با بسیاری از روش‌های تفسیر کلی مبتنی بر تجمعیت مقادیر Shapley ارائه می‌شود. این فصل هم رویکردهای برآورده جدید و هم روش‌های تفسیر کلی را توضیح می‌دهد.

توصیه می‌کنم ابتدا بخش‌های مربوط به مقادیر Shapley و مدل‌های محلی (LIME) را بخوانید.

۹.۶.۱ تعریف

هدف SHAP توضیح پیش‌بینی یک نمونه x با محاسبه مشارکت هر ویژگی در پیش‌بینی است. روش توضیح SHAP مقادیر Shapley را از تئوری بازی‌های ائتلافی محاسبه می‌کند. مقادیر ویژگی یک نمونه داده به عنوان بازیکن در یک ائتلاف عمل می‌کند. مقادیر Shapley به ما می‌گویند که چگونه «پرداخت» (=پیش‌بینی) را بین ویژگی‌ها به طور منصفانع توزیع کنیم. یک بازیکن می‌تواند یک مقدار ویژگی فردی در داده‌های جدولی باشد. یک بازیکن همچنین می‌تواند گروهی از مقادیر ویژگی باشد. به عنوان مثال برای توضیح یک تصویر، پیکسل‌ها را می‌توان به سوپرپیکسل‌ها گروه بندی کرد و پیش‌بینی را بین آنها توزیع کرد. یکی از نوآوری‌هایی که SHAP در جدول آورده است این است که توضیح مقدار Shapley به عنوان یک روش انتساب ویژگی افزودنی، یک مدل خطی نشان داده می‌شود. از این نما، SHAP مقادیر Shapley و LIME را به هم متصل می‌کند. SHAP، توضیح را به صورت زیر مشخص می‌کند:

$$g(z') = \phi_0 + \sum_{j=1}^M \phi_j z'_j$$

در اینجا g مدل توضیحی است، $z' \in \{0,1\}^M$ بردار ائتلاف^۲، M حداکثر اندازه ائتلاف و ϕ_j انتساب ویژگی برای یک ویژگی j ، مقادیر Shapley است. آنچه من "بردار ائتلاف" می‌نامم در مقاله SHAP "ویژگی‌های ساده

¹ SHapley Additive exPlanations

² coalition vector

شده^۱ نامیده می‌شود. فکر می‌کنم این نام انتخاب شده است، زیرا برای داده‌های تصویری، تصاویر در سطح پیکسل نمایش داده نمی‌شوند، بلکه در سوپرپیکسل‌ها جمع می‌شوند. فکر می‌کنم مفید است در مورد \mathbb{Z} به عنوان توصیف کننده ائتلاف‌ها در نظر گرفته شود: در بردار ائتلاف، ورودی ۱ به این معنی است که مقدار ویژگی مربوطه "حاضر" و ۰ به معنی "غایب" است. اگر با مقادیر Shapley آشنا باشید، این موضوع باید برای شما آشنا به نظر برسد. برای محاسبه مقادیر Shapley، فقط برخی از مقادیر ویژگی در حال بازی ("حاضر") هستند، شبیه سازی مس کنیم و برخی دیگر نیستند ("غایب"). ارائه به عنوان یک مدل خطی از ائتلاف ترفندی برای محاسبه ϕ ‌ها می‌باشد. برای x ، نمونه مورد علاقه، بردار ائتلاف x' بردار همه ۱‌ها است، یعنی همه مقادیر ویژگی "حاضر" هستند. فرمول به این صورت ساده می‌شود:

$$g(x') = \phi_0 + \sum_{j=1}^M \phi_j$$

شما می‌توانید این فرمول را با نماد مشابه در بخش مقدار Shapley پیدا کنید. اطلاعات بیشتر در مورد برآورده واقعی بعداً ارائه می‌شود. اجزه دهید ابتدا در مورد خواص ϕ ‌ها صحبت کنیم قبل از اینکه به جزئیات تخمین آنها بپردازیم.

مقادیر Shapley تنها راه حلی است که خاصیت‌های بازده، تقارن، مجازی و افزودنی را دارا می‌باشد. SHAP همچنین این خاصیت‌ها را دارد، زیرا مقادیر Shapley را محاسبه می‌کند. در مقاله SHAP، تفاوت‌هایی بین خاصیت‌های SHAP و خاصیت‌های Shapley خواهد دید. SHAP سه خاصیت مطلوب زیر را شرح می‌دهد:

(۱) دقت محلی

$$\hat{f}(x) = g(x') = \phi_0 + \sum_{j=1}^M \phi_j x'_j$$

اگر $\phi_0 = E_X(\hat{f}(x))$ تعریف کنید و همه x'_j را تنظیم کنید، این خاصیت بازده Shapley می‌شود، فقط با نامی دیگر و با استفاده از بردار ائتلاف.

$$\hat{f}(x) = \phi_0 + \sum_{j=1}^M \phi_j x'_j = E_X(\hat{f}(x)) + \sum_{j=1}^M \phi_j$$

(۲) غیبت^۲

$$x'_j = 0 \Rightarrow \phi_j = 0$$

غیبت می‌گوید که یک ویژگی از دست رفته نسبتی صفر دریافت می‌کند. توجه داشته باشید که x'_j به ائتلاف‌هایی اشاره دارد که در آن مقدار ۰ نشان دهنده عدم وجود یک مقدار ویژگی است. در نماد ائتلاف، همه مقادیر

¹ simplified features

² Missingness

ویژگی‌ها x'_j نمونه‌ای که باید توضیح داده شود باید 1^0 باشد. وجود 0 به این معنی است که مقدار ویژگی برای نمونه مورد علاقه وجود ندارد. این ویژگی در میان ویژگی‌های مقادیر Shapley عادی نیست. پس چرا برای SHAP به آن نیاز داریم؟ Lundberg آن را "minor book-keeping property"¹ می‌نامد. یک ویژگی از دست‌رفته می‌تواند - در تئوری - دارای یک مقدار Shapley دلخواه بدون آسیب رساندن به خاصیت دقت محلی باشد، زیرا در $0 = j'$ ضرب می‌شود. خاصیت غیبت باعث می‌شود که ویژگی‌های از دست رفته یک مقدار Shapley برابر با 0 دریافت کنند. در عمل، این فقط برای ویژگی‌هایی که ثابت هستند مرتبط است.

(۳) سازگاری^۲

اجازه دهید $((\hat{f}_x(z') - \hat{f}_x(z'_j)) \geq (\hat{f}_x(z') - \hat{f}_x(z'_j))$ نشان می‌دهد که $z' = 0$ برای هر دو مدل f و \hat{f} که در رابطه زیر برقرار باشند:

$$\hat{f}'_x(z') - \hat{f}'_x(z'_j) \geq \hat{f}_x(z') - \hat{f}_x(z'_j)$$

برای همه ورودی‌ها $z' \in \{0,1\}^M$, سپس:

$$\phi_j(\hat{f}', x) \geq \phi_j(\hat{f}, x)$$

خاصیت سازگاری می‌گوید که اگر یک مدل تغییر کند به طوری که مشارکت حاشیه ای یک مقدار ویژگی افزایش یابد یا بماند (صرف‌نظر از سایر ویژگی‌ها)، مقدار Shapley نیز افزایش می‌یابد یا بماند. از سازگاری، خاصیت‌های Shapley، خطی، مجازی و تقارن دنبال می‌شوند، همان‌طور که در ضمیمه Lundberg و Lee توضیح داده شده است.

KernelSHAP ۹.۶.۲

KernelSHAP برای یک نمونه x مشارکت هر یک از مقدار ویژگی‌ها در پیش‌بینی را تخمین می‌زند. KernelSHAP شامل پنج مرحله است:

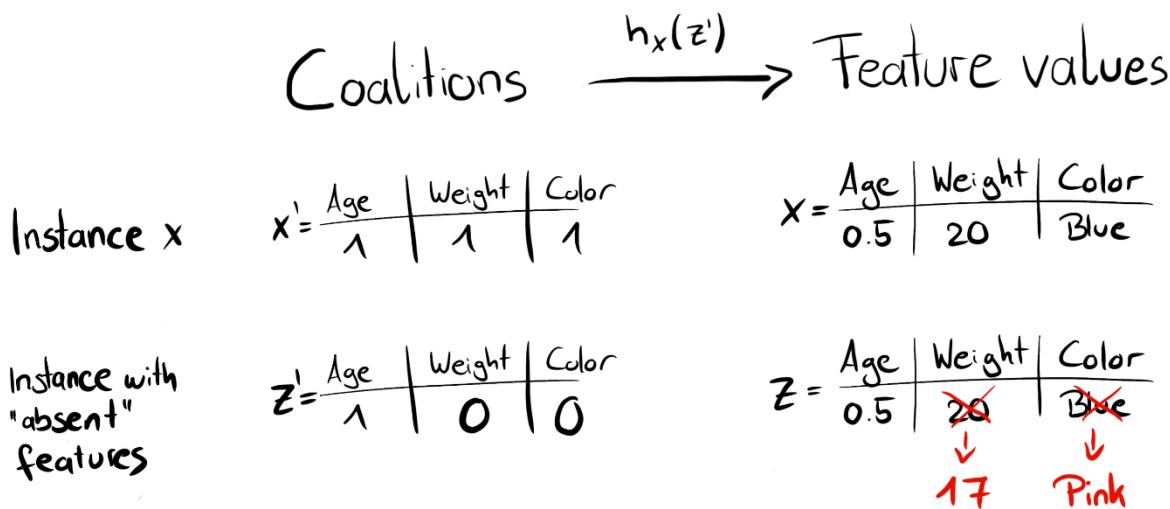
- ائتلاف‌های نمونه $\{k = 1, \dots, M\}$ ، $z'_k \in \{0,1\}^M$ ، $z_k \in \{0,1\}^M$ ، $=$ ویژگی حاضر در ائتلاف، $=$ ویژگی غائب
- برای هر z'_k پیش‌بینی بگیرید، در ابتدا با تبدیل z'_k به فضای ویژگی اصلی و سپس اعمال مدل $\hat{f}: \hat{f}(h_x(z'_k))$
- وزن را برابر هر z'_k با هسته SHAP محاسبه کنید.
- مدل خطی وزنی را برازش دهید.
- مقادیر Shapley ϕ_k ضرایب مدل خطی را برگردانید.

می‌توانیم با پرتاب‌های مکرر سکه یک ائتلاف تصادفی ایجاد کنیم تا زنجیره‌ای از 0 و 1 داشته باشیم. به عنوان مثال، بردار $(1, 0, 1, 0, 1, 0)$ به این معنی است که ما یک ائتلاف از ویژگی‌های اول و سوم داریم. K ائتلاف

¹ <https://github.com/slundberg/shap/issues/175#issuecomment-407134438>

² Consistency

نمونه برداری شده، به مجموعه داده‌ای برای مدل رگرسیون تبدیل می‌شوند. هدف مدل رگرسیون، پیش‌بینی یک ائتلاف است. (شما می‌گویید «صبر کنید!» «مدل روی این داده‌های ائتلاف بازیگری آموزش ندیده است و نمی‌تواند برای آنها پیش‌بینی کند.») برای رسیدن از ائتلاف مقادیر ویژگی به نمونه‌های داده معتبر، به یک تابع $h_x(z') = z$ نیاز داریم به طوری که: $\{0,1\}^M \rightarrow \mathbb{R}^p$: تابع h_x اها را به مقدار متناظر از نمونه x که از می‌خواهیم توضیح دهیم، نگاشت می‌کند. برای داده‌های جدولی، این تابع h_x را به مقادیر نمونه دیگری که از داده‌ها نمونه برداری می‌کنیم، نگاشت می‌کند. این بدان معنی است که ما "مقدار ویژگی غایب است" را با "مقدار ویژگی با مقدار ویژگی تصادفی از داده ها جایگزین شده است" برابر می‌کنیم. برای داده‌های جدولی، شکل زیر نگاشت از ائتلافها به مقادیر ویژگی را به تصویر می‌کشد:



شکل ۹.۲۲: تابع h_x یک ائتلاف را به یک نمونه معتبر نگاشت می‌کند. برای ویژگی‌های حاضر (۱)، h_x به مقادیر ویژگی x نگاشت می‌کند. برای ویژگی‌های غایب (۰)، h_x ، به مقادیر به یک نمونه داده نمونه‌گیری تصادفی نگاشت می‌کند.

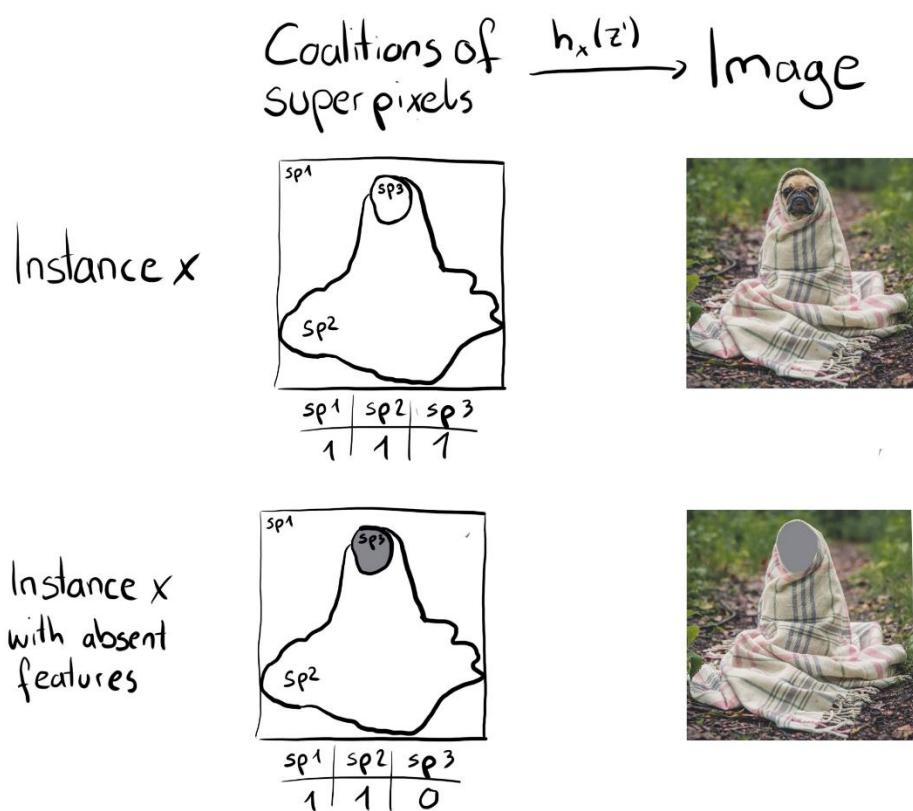
برای داده‌های جدولی، با X_C و X_S به عنوان مستقل، رفتار می‌کند و روی توزیع حاشیه‌ای، تجمعی می‌کند:

$$\hat{f}(h_x(z')) = E_{X_C}[\hat{f}(x)]$$

نمونه برداری از توزیع حاشیه‌ای به معنای نادیده گرفتن ساختار وابستگی بین ویژگی‌های موجود و غایب است. بنابراین KernelSHAP از مشکلی مشابه با همه روش‌های تفسیر مبتنی بر جایگشت رنج می‌برد. این تخمین به موارد غیر محتمل اهمیت زیادی می‌دهد. در نتیجه نتایج می‌توانند غیر قابل اعتماد شوند. اما نمونه برداری از توزیع حاشیه‌ای ضروری است. راه حل نمونه برداری از توزیع شرطی است که تابع مقدار و در نتیجه بازی را که مقادیر Shapley راه حل آن است تغییر می‌دهد. در نتیجه، مقادیر Shapley تفسیر متفاوتی دارند: برای مثال، یک ویژگی که ممکن است اصلاً توسط مدل استفاده نشده باشد، می‌تواند یک مقدار Shapley غیر صفر داشته

باشد وقتی که از نمونه‌گیری شرطی استفاده می‌شود. برای بازی حاشیه‌ای، این مقدار ویژگی همیشه مقدار Shapley برابر با 0 می‌گیرد.

برای تصاویر، شکل زیر یک تابع نگاشت ممکن را توضیح می‌دهد:



شکل ۹.۲۳: تابع h_x ائتلاف‌های سوپرپیکسل‌ها (sp) را به تصاویر نگاشت می‌کند. سوپرپیکسل‌ها گروهی از پیکسل‌ها هستند. برای ویژگی‌های حاضر (۱)، h_x قسمت مربوطه از تصویر اصلی را برمی‌گرداند. برای ویژگی‌های غایب (۰)، h_x ناحیه مربوطه را خاکستری می‌کند. تعیین میانگین رنگ پیکسل‌های اطراف یا موارد مشابه نیز یک گزینه خواهد بود.

تفاوت بزرگ با LIME وزن نمونه‌ها در مدل رگرسیون است. LIME نمونه‌ها را با توجه به نزدیک بودن آنها به نمونه اصلی وزن دهی می‌کند. هر چه 0 در بردار ائتلاف بیشتر باشد، وزن در LIME کوچک‌تر است. نمونه‌های نمونه برداری شده را با توجه به وزنی که ائتلاف در تخمین مقدار Shapley به دست می‌آورد وزن می‌دهد. ائتلاف‌های کوچک (یعنی تعداد کمی ۱) و ائتلاف‌های بزرگ (یعنی بسیاری از ۱) بیشترین وزن را دارند. شهود پشت آن این است: ما بیشتر در مورد ویژگی‌های فردی می‌آموزیم اگر بتوانیم اثرات آنها را به صورت

جداگانه مطالعه کنیم. اگر یک اختلاف از یک ویژگی واحد تشکیل شده باشد، می‌توانیم در مورد تأثیر اصلی جدا شده این ویژگی بر پیش‌بینی بیاموزیم. اگر یک اختلاف از همه ویژگی‌ها به جز یک ویژگی تشکیل شده باشد، می‌توانیم در مورد تأثیر کلی این ویژگی (اثر اصلی به اضافه تعاملات ویژگی) اطلاعات کسب کنیم. اگر یک اختلاف از نیمی از ویژگی‌ها تشکیل شده باشد، ما اطلاعات کمی در مورد مشارکت یک ویژگی فردی می‌دانیم، زیرا اختلاف‌های احتمالی زیادی با نیمی از ویژگی‌ها وجود دارد. برای دستیابی به وزن‌بندی مطابق با Shapley، Lundberg and Lee (۲۰۱۷) پیشنهاد هسته SHAP را دادند:

$$\pi_x(z') = \frac{(M-1)}{\binom{M}{|z'|} |z'| (M-|z'|)}$$

در اینجا، M حداکثر اندازه اختلاف و $|z'|$ تعداد ویژگی‌های حاضر در نمونه z' است. Lundberg and Lee (۲۰۱۷) نشان می‌دهند که رگرسیون خطی با این وزن هسته، منجر به مقادیر Shapley می‌شود. اگر از هسته SHAP با LIME در داده‌های اختلاف استفاده کنید، LIME نیز مقادیر Shapley را تخمین می‌زند! ما می‌توانیم در مورد نمونه‌گیری از اختلاف‌ها کمی‌هوشمندتر باشیم: کوچک‌ترین و بزرگ‌ترین اختلاف‌ها بیشترین وزن را به خود اختصاص می‌دهند. ما با استفاده از برخی از بودجه نمونه برداری K برای گنجاندن این اختلاف‌ها پر وزن به جای نمونه برداری کورکورانه، تخمین‌های مقدار Shapley بهتری دریافت می‌کنیم. ما با همه اختلاف‌های ممکن با تعداد ویژگی‌های 1 و $M-1$ شروع می‌کنیم که در مجموع 2 برابر M اختلاف می‌شود. وقتی بودجه باقی مانده کافی باشد (بودجه فعلی $2M - K$ است)، می‌توانیم اختلاف‌هایی با 2 ویژگی و با $M-2$ ویژگی و غیره را شامل کنیم. از اندازه‌های اختلاف باقی‌مانده، با وزن‌های مجدد تنظیم‌شده نمونه‌برداری می‌کنیم. ما داده‌ها، هدف و وزن‌ها را داریم. همه چیزهایی که برای ساختن مدل رگرسیون خطی وزنی خود نیاز داریم:

$$g(z') = \phi_0 + \sum_{j=1}^M \phi_j z'_j$$

ما مدل خطی g را با بهینه سازی تابع ضرر L زیر آموخته می‌دهیم:

$$L(\hat{f}, g, \pi_x) = \sum_{z' \in Z} [\hat{f}(h_x(z')) - g(z')]^2 \pi_x(z')$$

که در آن Z داده‌های آموزشی است. این رابطه متداول مجموع خطاهای مربعی است که ما عموماً برای مدل‌های خطی بهینه می‌کنیم. ضرایب تخمینی مدل رفتهای، مقادیر Shapley هستند.

از آنجایی که ما در یک تنظیم رگرسیون خطی هستیم، می‌توانیم از ابزارهای استاندارد برای رگرسیون نیز استفاده کنیم. برای مثال، می‌توانیم عبارت منظم‌سازی را اضافه کنیم تا مدل محدود شود. اگر یک پنالتی $L1$ به ضرر L اضافه کنیم، می‌توانیم توضیحات محدود ایجاد کنیم. (من خیلی مطمئن نیستم که آیا ضرایب حاصل هنوز هم به عنوان مقادیر Shapley معتبر هستند یا خیر).

TreeSHAP ۹.۶.۳

Lundberg et al (۲۰۱۸)، TreeSHAP را پیشنهاد دادند، گونه‌ای از SHAP برای مدل‌های یادگیری ماشین مبتنی بر درخت مانند درخت‌های تصمیم‌گیری، جنگل‌های تصادفی و درخت‌های تقویت‌شده گرادیان. TreeSHAP به عنوان یک جایگزین سریع و مخصوص مدل برای KernelSHAP معرفی شد، اما مشخص شد که می‌تواند انتسابات غیر شهودی را تولید کند.

تابع مقدار را با استفاده از انتظار شرطی $E_{X_S|X_C}(\hat{f}(x)|x_S)$ تعریف می‌کند به جای انتظار حاشیه ای. مشکل انتظار شرطی این است که ویژگی‌هایی که هیچ تأثیری بر تابع پیش‌بینی f ندارند، می‌توانند تخمین TreeSHAP متفاوت از صفر دریافت کنند که توسط Janzing et al (2020) و Sundararajan and Najmi (2020) نشان داده شده است. تخمین غیر صفر زمانی می‌تواند اتفاق بیفتد که ویژگی با ویژگی دیگری که در واقع بر پیش‌بینی تأثیر دارد، همبستگی داشته باشد.

TreeSHAP چقدر سریع‌تر است؟ در مقایسه با KernelSHAP دقیق، پیچیدگی محاسباتی از $O(TL2^M)$ به $O(TLD^2)$ را کاهش می‌دهد. در اینجا T تعداد درختان، L حداکثر تعداد برگ در هر درخت و D حداکثر عمق هر درخت است.

TreeSHAP از انتظار شرطی $E_{X_S|X_C}(\hat{f}(x)|x_S)$ استفاده می‌کند. برای تخمین اثرات من به شما شهودی در مورد اینکه چگونه می‌توانیم پیش‌بینی مورد انتظار را برای یک درخت واحد، یک نمونه x و زیرمجموعه ویژگی S محاسبه کنیم. اگر همه ویژگی‌ها را مشروط کنیم - اگر S مجموعه همه ویژگی‌ها باشد - پیش‌بینی گره ای که نمونه x در آن قرار می‌گیرد، پیش‌بینی مورد انتظار خواهد بود. اگر پیش‌بینی را با هیچ ویژگی شرطی نکنیم - اگر S خالی بود - از میانگین وزنی پیش‌بینی‌های تمام گره‌های پایانی استفاده می‌کنیم. اگر S شامل برخی ویژگی‌ها، اما نه همه، باشد، پیش‌بینی گره‌های غیرقابل دسترس را نادیده می‌گیریم. دست نیافتنی به این معنی است که مسیر تصمیم گیری که به این گره منتهی می‌شود با مقادیر موجود در تقابل با x است. از گره‌های پایانی باقی‌مانده، پیش‌بینی‌های وزن شده را بر اساس اندازه گره (یعنی تعداد نمونه‌های آموزشی در آن گره) با میانگین گیری بدست می‌آوریم. میانگین گره‌های پایانی باقی‌مانده، وزن‌دهی شده با تعداد نمونه‌های هر گره، پیش‌بینی مورد انتظار برای x به شرط S است. مشکل این است که ما باید این رویه را برای هر زیر مجموعه S ممکن از مقادیر ویژگی اعمال کنیم. TreeSHAP در زمان چند جمله‌ای به جای نمایی محاسبه را انجام می‌دهد. ایده اصلی این است که همه زیرمجموعه‌های ممکن S را به طور همزمان به پایین درخت فشار دهید. برای هر گره تصمیم باید تعداد زیر مجموعه‌ها را پیگیری کنیم. این بستگی به زیر مجموعه‌های گره والد و ویژگی تقسیم دارد. به عنوان مثال، هنگامی که اولین تقسیم در یک درخت روی ویژگی x_3 باشد، آنگاه تمام زیر مجموعه‌هایی که دارای ویژگی x_3 هستند به یک گره (گرهی که x_3 می‌رود) می‌روند. زیرمجموعه‌هایی که دارای

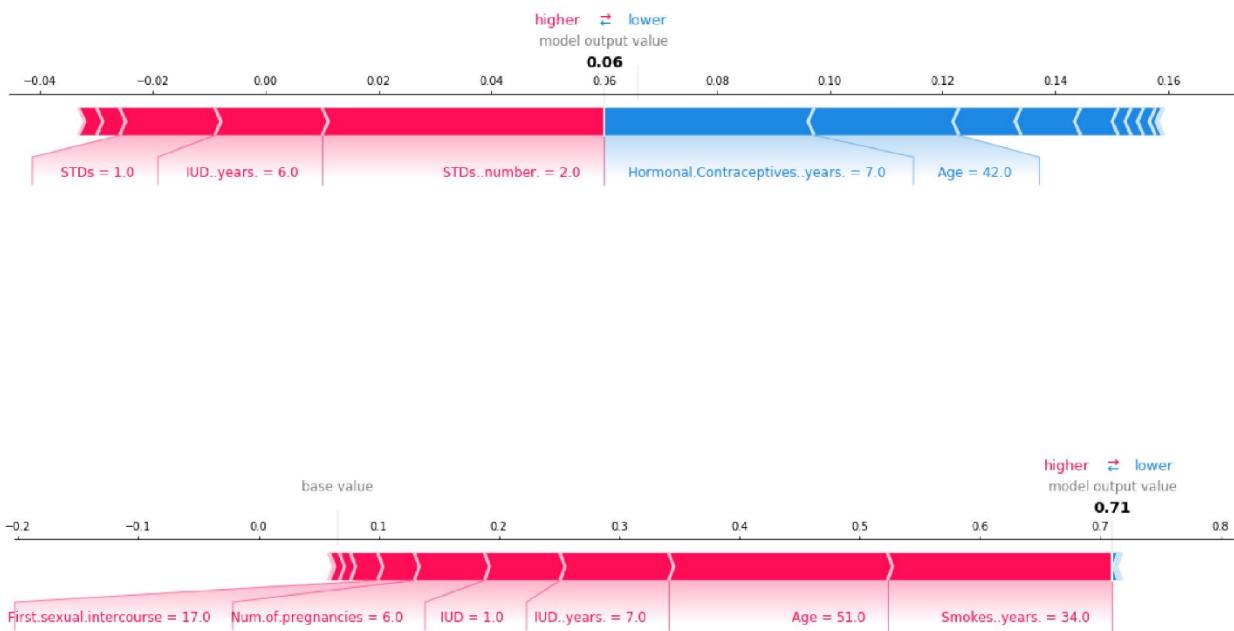
ویژگی x^3 نیستند با کاهش وزن به هر دو گره می‌روند. متأسفانه زیر مجموعه‌هایی با اندازه‌های مختلف وزن‌های متفاوتی دارند. الگوریتم باید وزن کلی زیر مجموعه‌ها را در هر گره پیگیری کند. این کار الگوریتم را پیچیده می‌کند. برای جزئیات TreeSHAP به مقاله اصلی مراجعه کنید. محاسبات را می‌توان به درختان بیشتری گسترش داد: به لطف ویژگی افزودنی مقادیر Shapley. مقادیر Shapley یک مجموعه درختی، میانگین (وزنی) مقادیر Shapley هر درخت است.

در ادامه، به توضیحات SHAP در عمل خواهیم پرداخت.

۹.۶.۴ مثال‌ها

من یک طبقه‌بندی کننده جنگل تصادفی را با 100×100 درخت آموزش دادم تا خطر ابتلا به سلطان دهانه رحم را پیش‌بینی کند. از SHAP برای توضیح پیش‌بینی‌های فردی استفاده خواهیم کرد. می‌توانیم از روش تخمین سریع TreeSHAP به جای روش کندر KernelSHAP استفاده کنیم، زیرا یک جنگل تصادفی مجموعه‌ای از درختان است. اما این مثال به جای تکیه بر توزیع شرطی، از توزیع حاشیه‌ای استفاده می‌کند. این روش در پکیج توضیح داده شده است، اما در مقاله اصلی نیست.تابع Python TreeSHAP با توزیع حاشیه‌ای کندر است، اما همچنان سریع‌تر از KernelSHAP است، زیرا به صورت خطی با ردیف‌های داده مقیاس می‌شود. از آنجا که ما در اینجا از توزیع حاشیه‌ای استفاده می‌کنیم، تفسیر همان است که در فصل مقدار Shapley گفته شد. اما با پکیج Python SHAP ترسیم متفاوتی ارائه می‌شود: می‌توانید ویژگی‌های ویژگی‌هایی مانند مقادیر Shapley را به عنوان «نیروها» تجسم کنید. هر مقدار ویژگی، نیرویی است که پیش‌بینی را افزایش یا کاهش می‌دهد. پیش‌بینی از پایه شروع می‌شود. خط پایه برای مقادیر Shapley میانگین همه پیش‌بینی‌ها است. در نمودار، هر مقدار Shapley یک فلش است که برای افزایش (مقدار مثبت) یا کاهش (مقدار منفی) پیش‌بینی فشار می‌آورد. این نیروها در پیش‌بینی واقعی نمونه داده، یکدیگر را متعادل می‌کنند.

شكل زیر نمودار نیروی توضیحی SHAP را برای دو زن از مجموعه‌داده سلطان دهانه رحم نشان می‌دهد:



شکل ۹.۲۴: مقادیر SHAP برای توضیح احتمالات پیش‌بینی شده سرطان دو فرد. خط پایه - میانگین احتمال پیش‌بینی شده - $0.66/60$ است. اولین زن دارای پیش‌بینی خطر کم $0.60/60$ است. اثرات افزایش خطر مانند بیماری‌های مقاربته با کاهش اثراتی مانند سن جبران می‌شود. زن دوم دارای خطر بالای $0.71/71$ پیش‌بینی شده است. سن ۵۱ سالگی و سیگار کشیدن ۳۴ سال خطر ابتلا به سرطان را افزایش می‌دهد. اینها توضیحاتی برای پیش‌بینی‌های فردی بود.

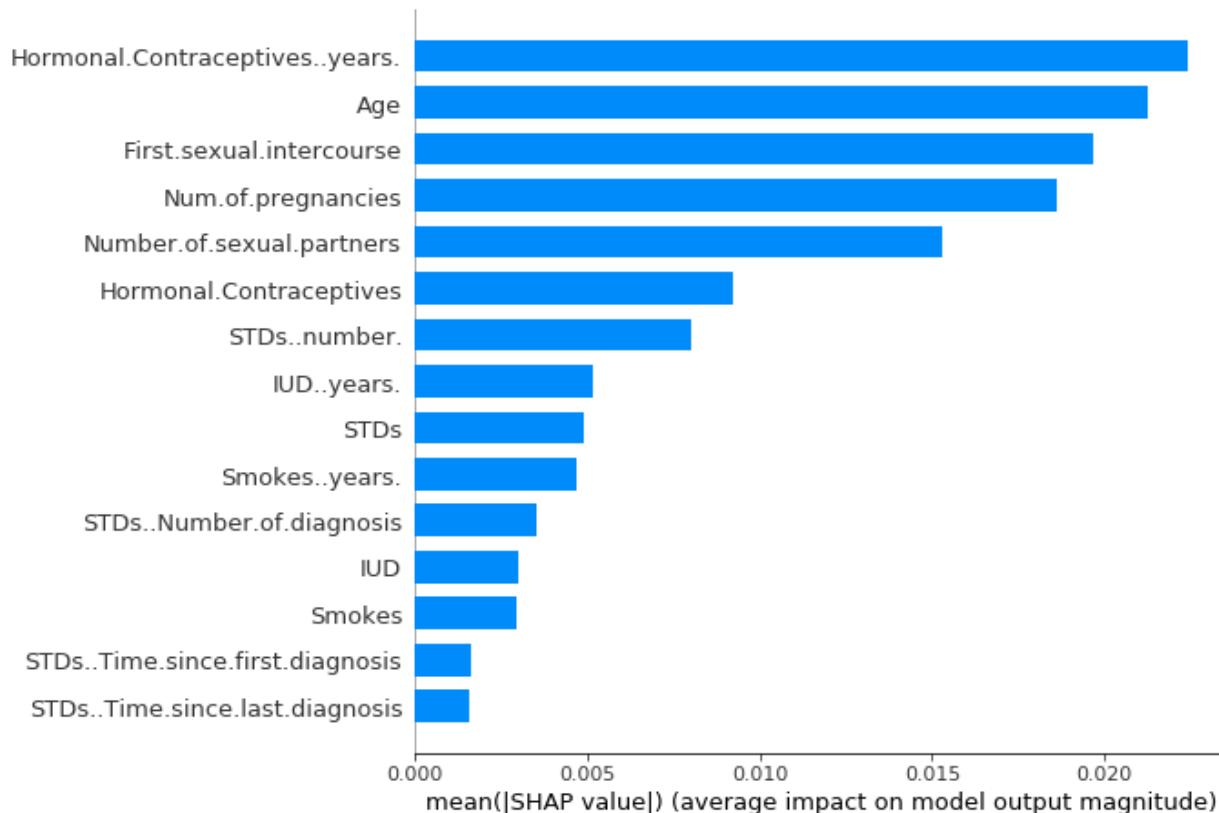
مقادیر Shapley را می‌توان در توضیحات کلی ترکیب کرد. اگر SHAP را برای هر نمونه اجرا کنیم، ماتریسی از مقادیر Shapley را دریافت می‌کنیم. این ماتریس دارای یک ردیف برای هر نمونه داده و یک ستون در هر ویژگی است. ما می‌توانیم کل مدل را با تجزیه و تحلیل مقادیر Shapley در این ماتریس تفسیر کنیم. ما با اهمیت ویژگی SHAP شروع می‌کنیم.

۹.۶.۵ اهمیت ویژگی SHAP

ایده اهمیت ویژگی SHAP ساده است: ویژگی‌هایی با مقادیر مطلق بزرگ Shapley مهم هستند. از آنجایی که ما اهمیت کلی را می‌خواهیم، میانگین مقادیر مطلق Shapley را برای هر ویژگی در میان داده‌ها محاسبه می‌کنیم:

$$I_j = \frac{1}{n} \sum_{i=1}^n |\phi_j^{(i)}|$$

در مرحله بعد، ویژگی‌ها را به صورت نزولی بر اساس اهمیت مرتب می‌کنیم و آنها را رسم می‌کنیم. شکل زیر اهمیت ویژگی SHAP را برای جنگل تصادفی که قبلاً برای پیش‌بینی سرطان دهانه رحم آموزش داده شده را نشان می‌دهد.



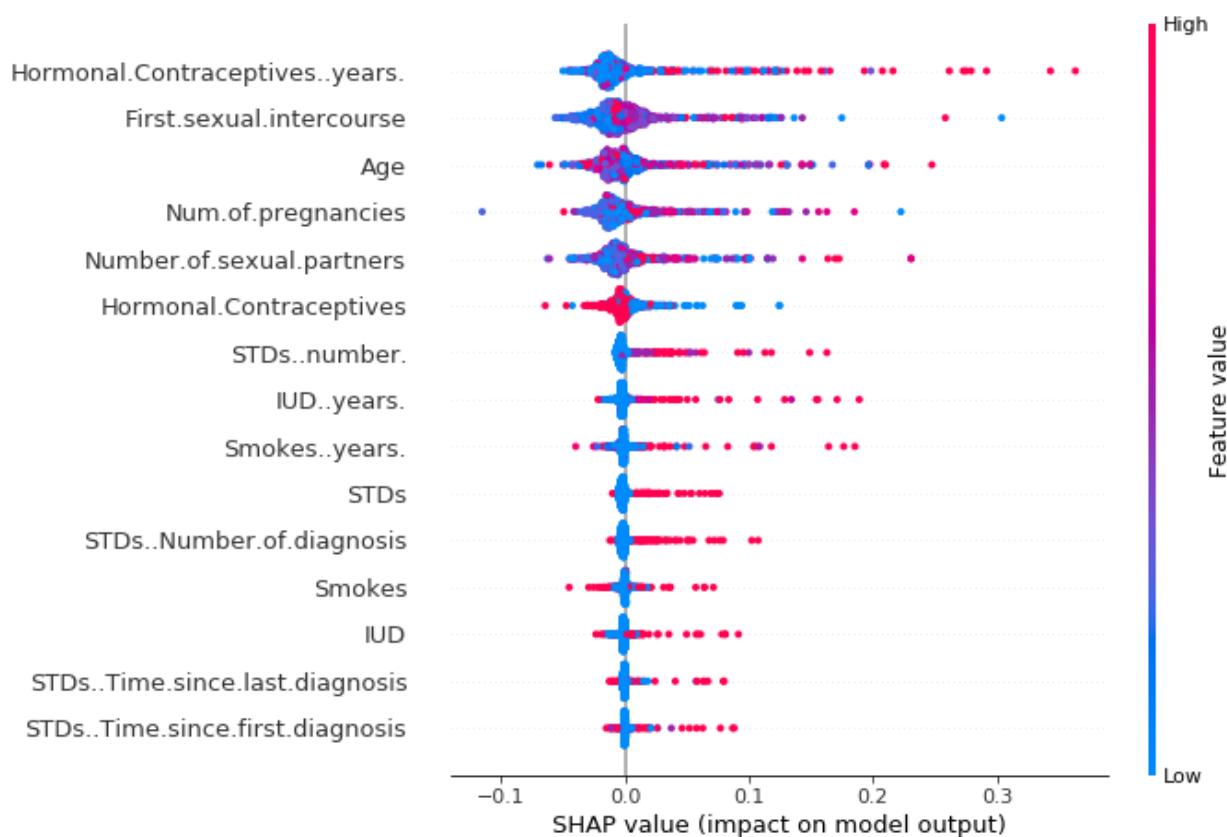
شکل ۹.۲۵: اهمیت ویژگی SHAP به صورت میانگین مقادیر مطلق Shapley اندازه گیری می‌شود. تعداد سالهای استفاده از داروهای ضد بارداری هورمونی مهم‌ترین ویژگی بود که احتمال سرطان مطلق پیش‌بینی شده را به طور متوسط ۲/۴ درصد (۰/۰۲۶) در محور x تغییر داد.

اهمیت ویژگی SHAP جایگزینی برای اهمیت ویژگی جایگشتی است. تفاوت زیادی بین هر دو معیار اهمیت وجود دارد: اهمیت ویژگی جایگشت بر اساس کاهش عملکرد مدل است. SHAP بر اساس مقدار انتساب ویژگی است.

نمودار اهمیت ویژگی مفید است، اما حاوی اطلاعاتی فراتر از اهمیت نیست. نمودار خلاصه که در ادامه آورده شده است دارای اطلاعات مفیدتری می‌باشد.

۹.۶.۶ نمودار خلاصه SHAP

طرح خلاصه اهمیت ویژگی را با اثرات ویژگی ترکیب می‌کند. هر نقطه در نمودار خلاصه یک مقدار Shapley برای یک ویژگی و یک نمونه است. موقعیت روی محور y توسط ویژگی و روی محور x با مقدار Shapley تعیین می‌شود. رنگ نشان دهنده ارزش ویژگی از کم به زیاد است. نقاط همپوشانی در جهت محور y جایه جا شده اند تا همه نقاط دیده شوند، بنابراین ما یک حس از توزیع مقادیر Shapley در هر ویژگی دریافت می‌کنیم. ویژگی‌ها با توجه به اهمیت آنها مرتب شده‌اند.



شکل ۹.۲۶: نمودار خلاصه SHAP. سال‌های کم استفاده از داروهای ضد بارداری هورمونی خطر سرطان پیش‌بینی شده را کاهش می‌دهد، تعداد زیاد سال‌ها این خطر را افزایش می‌دهد. یادآوری همیشگ: همه اثرات رفتار مدل را توصیف می‌کنند و لزوماً در دنیای واقعی علت نیستند.

در نمودار خلاصه، اولین نشانه‌های رابطه بین ارزش یک ویژگی و تأثیر آن بر پیش‌بینی را می‌بینیم. اما برای دیدن شکل دقیق رابطه، باید به نمودارهای وابستگی SHAP مراجعه کنیم.

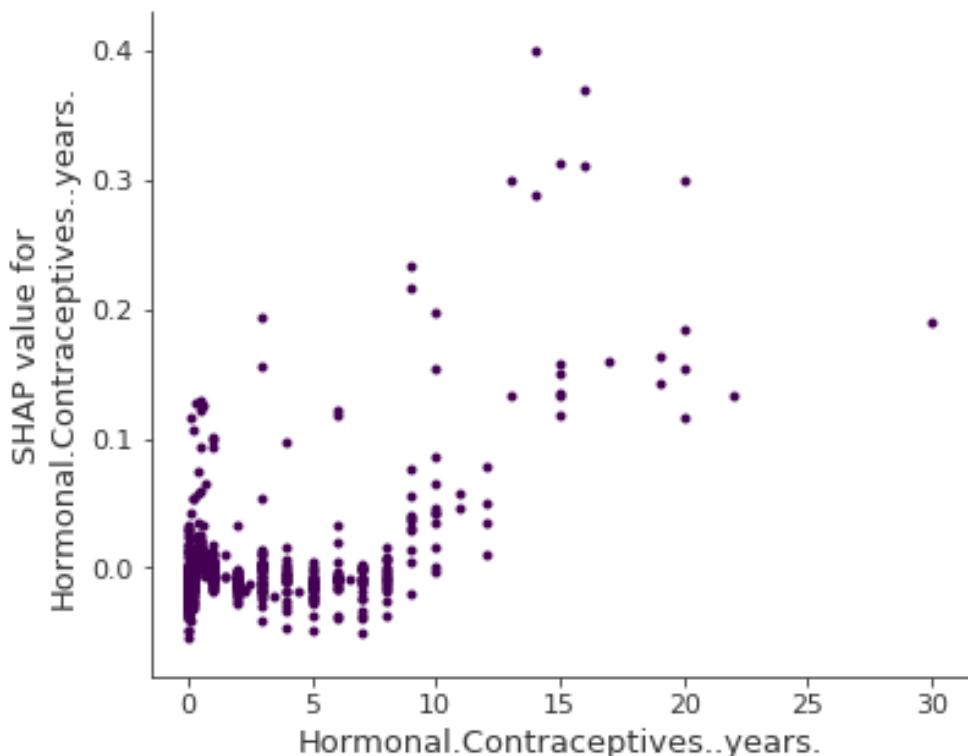
۹.۶.۷ نمودار وابستگی SHAP

وابستگی ویژگی SHAP ممکن است ساده‌ترین طرح تفسیر کلی باشد: ۱) یک ویژگی را انتخاب کنید. ۲) برای هر نمونه داده، یک نقطه با مقدار ویژگی در محور x و مقدار Shapley مربوطه در محور y رسم کنید. ۳) تمام شد.

از نظر ریاضی، نمودار حاوی این نکات است:

$$\left\{ \left(x_j^{(i)}, \phi_j^{(i)} \right) \right\}_{i=1}^n$$

شکل زیر وابستگی ویژگی SHAP را برای سال‌ها به داروهای ضد بارداری هورمونی نشان می‌دهد:



شکل ۹.۲۷: نمودار وابستگی SHAP برای سال‌ها به داروهای ضد بارداری هورمونی. در مقایسه با ۰ سال، در تعداد سال کم، احتمال پیش‌بینی شده کاهش و در تعداد سال‌های زیاد احتمال سرطان پیش‌بینی شده افزایش می‌یابد.

نمودارهای وابستگی SHAP جایگزینی برای نمودارهای وابستگی جزئی و اثرات محلی انباشته هستند. در حالی که نمودار PDP و ALE اثرات متوسط را نشان می‌دهند، وابستگی SHAP نیز واریانس را در محور y نشان می‌دهد. به خصوص در صورت تعاملات، نمودار وابستگی SHAP در محور y پراکنده‌تر خواهد بود. نمودار وابستگی را می‌توان با برجسته کردن این تعاملات ویژگی بهبود بخشید.

۹.۶.۸ مقادیر تعامل SHAP

اثر متقابل، اثر ویژگی ترکیبی اضافی پس از محاسبه اثرات ویژگی‌های فردی است. شاخص تعامل Shapley از تئوری بازی به صورت زیر تعریف می‌شود:

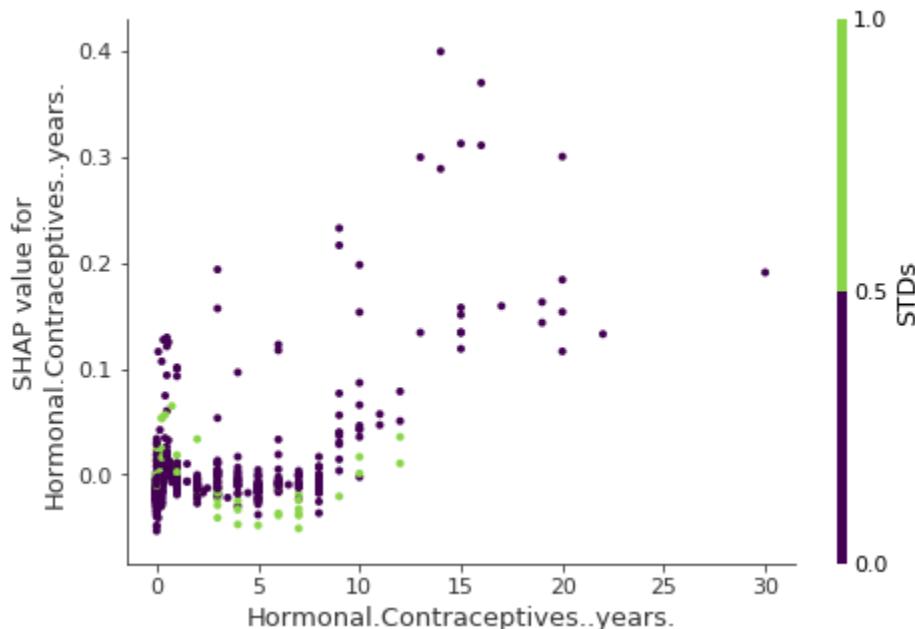
$$\phi_{i,j} = \sum_{S \subseteq \{i,j\}} \frac{|S|!(M - |S| - 2)!}{2(M - 1)!} \delta_{ij}(S)$$

هنگامی که $i \neq j$

$$\delta_{ij}(S) = \hat{f}_x(S \cup \{i,j\}) - \hat{f}_x(S \cup \{i\}) - \hat{f}_x(S \cup \{j\}) + \hat{f}_x(S)$$

این فرمول اثر اصلی ویژگی‌ها را از اثرات فردی کم می‌کند، در نتیجه اثر تعامل خالص را دریافت کنیم. مقادیر SHAP را با میان‌گیری از همه ائتلاف‌های ممکن ویژگی S محاسبه می‌کنیم، مانند محاسبه مقدار M^*M به دست مقادیر تعامل SHAP را برای همه ویژگی‌ها محاسبه کردیم، برای هر نمونه یک ماتریس با ابعاد M^*M می‌آوریم که M تعداد ویژگی‌ها است.

چگونه می‌توانیم از شاخص تعامل استفاده کنیم؟ به عنوان مثال، برای رنگ آمیزی خودکار نمودار وابستگی ویژگی SHAP با قوی‌ترین تعامل:



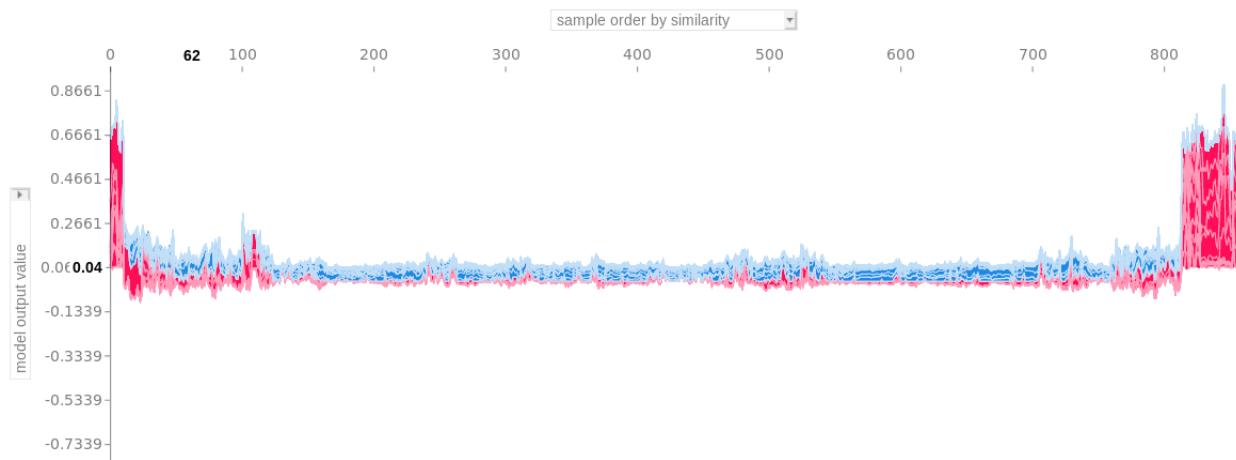
شکل ۹.۲۸: نمودار وابستگی ویژگی SHAP با تصویر تعامل. سال‌ها استفاده از داروهای ضد بارداری هورمونی با بیماری‌های مقاربی تداخل دارد. در موارد نزدیک به صفر سال، وقوع STD خطر سرطان پیش‌بینی شده را افزایش می‌دهد. برای سال‌های بیشتر در مورد داروهای ضد بارداری، وقوع STD خطر پیش‌بینی شده را کاهش می‌دهد. باز هم تاکید می‌شود، این یک مدل علی نیست. اثرات ممکن است به دلیل گیج کننده‌ای باشد (مثلًاً بیماری‌های مقاربی و خطر کمتر سرطان می‌تواند با مراجعه بیشتر به پزشک همبستگی داشته باشد).

۹.۶.۹ خوشبندی مقادیر Shapley

شما می‌توانید داده‌های خود را با کمک مقادیر Shapley خوشبندی کنید. هدف از خوشبندی یافتن گروههایی از نمونه‌های مشابه است. به طور معمول، خوشبندی بر اساس ویژگی‌ها صورت می‌گیرد. ویژگی‌ها اغلب در مقیاس‌های مختلف هستند. برای مثال، ارتفاع ممکن است بر حسب متر، شدت رنگ از ۰ تا ۱۰۰ و مقداری خروجی سنسور بین ۱ و ۱۰۰ اندازه گیری شود. مشکل در محاسبه فاصله بین نمونه‌هایی با چنین ویژگی‌هایی متفاوت و غیر قابل مقایسه است.

خوشبندی SHAP با خوشبندی مقادیر Shapley هر نمونه کار می‌کند. این بدان معنی است که شما نمونه‌ها را با تشابه توضیح خوشبندی می‌کنید. همه مقادیر SHAP واحد یکسانی دارند - واحد فضای پیش‌بینی. شما می‌توانید از هر روش خوشبندی استفاده کنید. مثال زیر از خوشبندی تجمعی سلسله مراتبی^۱ برای مرتب سازی نمونه‌ها استفاده می‌کند.

نمودار شامل نمودارهای نیرو زیادی است که هر کدام پیش‌بینی یک نمونه را توضیح می‌دهد. نمودارهای نیرو را به صورت عمومی می‌چرخانیم و با توجه به شباهت خوشه‌ای آنها در کنار هم قرار می‌دهیم.



شکل ۹.۲۹: توضیحات SHAP پشتۀ ای که بر اساس تشابه توضیح خوشبندی شده‌اند. هر موقعیت در محور x نمونه ای از داده‌ها است. مقادیر قرمز SHAP پیش‌بینی را افزایش می‌دهد، مقادیر آبی آن را کاهش می‌دهد. یک خوشه بیرون زده است: در سمت راست گروهی با خطر سلطان پیش‌بینی شده بالا قرار دارد.

۹.۶.۱۰ مزایا

از آنجایی که SHAP مقادیر Shapley را محاسبه می‌کند، تمام مزایای مقادیر Shapley برقرار است: یک پایه تئوری محکم در تئوری بازی‌ها دارد. پیش‌بینی به طور عادلانه بین مقادیر ویژگی توزیع شده است. ما توضیحات مقابله ای را دریافت می‌کنیم که پیش‌بینی را با پیش‌بینی میانگین مقایسه می‌کند.

^۱ hierarchical agglomerative clustering

است. همچنین به متحدد کردن زمینه یادگیری ماشین قابل تفسیر کمک می‌کند.

SHAP یک پیاده‌سازی سریع برای مدل‌های مبتنی بر درخت دارد. من معتقدم که این کلید محبوبیت SHAP بود، زیرا بزرگ‌ترین مانع برای پذیرش مقادیر Shapley محاسبه کند است.

محاسبات سریع امکان محاسبه بسیاری از مقادیر Shapley موردنیاز برای تفاسیر مدل کلی را فراهم می‌کند. روش‌های تفسیر کلی شامل اهمیت ویژگی، وابستگی ویژگی، تعاملات، خوشه‌بندی و نمودارهای خلاصه است. با SHAP، تفاسیر کلی با توضیحات محلی سازگار است، زیرا مقادیر Shapley "واحد اتمی" تفاسیر جهانی هستند. اگر از LIME برای توضیحات محلی و نمودارهای وابستگی جزئی به اضافه اهمیت ویژگی جایگشت برای توضیحات کلی استفاده می‌کنید، شما قادر به یک پایه مشترک هستید.

۹.۶.۱۱ معایب

KernelSHAP کند است. این موضوع باعث می‌شود وقتی می‌خواهید مقادیر Shapley را برای تعداد زیادی از نمونه‌ها محاسبه کنید، استفاده از KernelSHAP غیرعملی است. همچنین تمام روش‌های کلی SHAP، مانند اهمیت ویژگی ShAP نیاز به محاسبه مقادیر Shapley برای بسیاری از نمونه‌ها دارند.

KernelSHAP وابستگی ویژگی را نادیده می‌گیرد. اکثر روش‌های دیگر تفسیر مبتنی بر جایگشت این مشکل را دارند. با جایگزینی مقادیر ویژگی با مقادیر نمونه‌های تصادفی، معمولاً نمونه برداری تصادفی از توزیع حاشیه‌ای آسان‌تر است. با این حال، اگر ویژگی‌ها وابسته باشند، به عنوان مثال همبستگی داشته باشند، این کار منجر به اعمال وزن بیش از حد بر روی نقاط داده غیرمحتمل می‌شود. TreeSHAP این مشکل را با مدل سازی صریح پیش‌بینی مورد انتظار شرطی حل می‌کند.

TreeSHAP می‌تواند انتسابات ویژگی‌های غیرشهودی را تولید کند. در حالی که TreeSHAP مشکل برون یابی به نقاط داده غیرمحتمل را حل می‌کند، این کار را با تغییر تابع مقدار انجام می‌دهد و بنابراین کمی بازی را تغییر می‌دهد. TreeSHAP تابع مقدار را با تکیه بر پیش‌بینی مورد انتظار مشروط تغییر می‌دهد. با تغییر در تابع مقدار، ویژگی‌هایی که هیچ تاثیری بر پیش‌بینی ندارند می‌توانند یک مقدار TreeSHAP متفاوت از صفر دریافت کنند.

معایب مقادیر Shapley در مورد SHAP نیز صدق می‌کند: مقادیر Shapley می‌توانند اشتباه تفسیر شوند و برای محاسبه داده‌های جدید به داده‌ها نیاز است (به جز TreeSHAP).

امکان ایجاد تفاسیر عمدی گمراه کننده با SHAP وجود دارد که می‌تواند سوگیری‌ها را پنهان کند (Slack et al., 2020). اگر شما دانشمند داده‌ای هستید که توضیحات را ایجاد می‌کند، این یک مشکل واقعی

نیست (اگر شما دانشمند داده شیطان صفتی باشید که می‌خواهد توضیحات گمراه کننده ایجاد کند، یک مزبت نیز خواهد بود). برای گیرندگان توضیح SHAP، این موضوع یک نقطه ضعف است: آنها نمی‌توانند در مورد صحت توضیح مطمئن باشند.

۹.۶.۱۲ نرم افزار

نویسندهای SHAP را در پکیج^۱ shap نرم افزار Python پیاده‌سازی کردند. کتاب تفسیر مدل‌های یادگیری ماشین با SHAP کاربرد SHAP با shap بسته را به طور عمیق پوشش می‌دهد. این پیاده‌سازی برای مدل‌های مبتنی بر درخت در کتابخانه یادگیری ماشین^۲ برای Python Scikit-Learn^۳ کار می‌کند. برای مثال‌های این فصل از پکیج shap استفاده شده است. SHAP در چارچوب‌های تقویت درخت fastshap^۴ و xgboost^۵ ادغام شده است. در نرم افزار R، پکیج‌های shapper^۶ و xgbplot^۷ وجود دارد. SHAP در پکیج xgboost^۷ نرم افزار R نیز گنجانده شده است.

^۱ <https://github.com/slundberg/shap>

^۲ <https://scikit-learn.org/stable/>

^۳ <https://github.com/dmlc/xgboost/tree/master/python-package>

^۴ <https://github.com/microsoft/LightGBM>

^۵ <https://modeloriented.github.io/shapper/>

^۶ <https://github.com/bgreenwell/fastshap>

^۷ <https://rdrr.io/cran/xgboost/man/xgb.plot.shap.html>

فصل ۱۱ نگاهی به آینده یادگیری ماشین قابل تفسیر

آینده یادگیری ماشین قابل تفسیر چیست؟ این بخش یک تمرین ذهنی نظری و یک حدس شخصی است که یادگیری ماشین قابل تفسیر چگونه توسعه خواهد یافت. من کتاب را با داستان‌های کوتاه نسبتاً بدینانه شروع کردم و می‌خواهم با نگاهی خوشبینانه‌تر به پایان برسم.
من «پیش‌بینی‌های» خود را بر سه فرض استوار کرده‌ام:

۱- **دیجیتال سازی:** هر گونه اطلاعات (جالب) دیجیتالی می‌شود. به پول نقد الکترونیکی و معاملات آنلاین فکر کنید. به کتاب‌های الکترونیکی، موسیقی و ویدئو فکر کنید. به تمام داده‌های حسی در مورد محیط، رفتار انسانی، فرآیندهای تولید صنعتی و غیره فکر کنید. محرك‌های دیجیتالی کردن همه چیز عبارت‌اند از: رایانه‌ها/حسگرها/ذخیره‌سازی‌های ارزان، تاثیرات مقیاس (برنده همه چیز را می‌گیرد)، مدل‌های تجاری جدید، زنجیره‌های ارزش مدولار، فشار هزینه و بسیاری موارد دیگر.

۲- **اتوماسیون:** هنگامی که یک وظیفه می‌تواند خودکار باشد و هزینه اتوماسیون کمتر از هزینه انجام وظیفه در طول زمان باشد، وظیفه خودکار می‌شود. حتی قبل از معرفی کامپیوتر ما درجه خاصی از اتوماسیون را داشتیم. به عنوان مثال، ماشین بافندگی، بافندگی خودکار یا ماشین بخار با توان اسپ خودکار. اما کامپیوترها و دیجیتالی شدن، اتوماسیون را به سطحی بالاتر می‌برد. صرفاً این واقعیت که می‌توانید برای حلقه‌ها برنامه‌نویسی کنید، ماکروهای اکسل بنویسید، پاسخ‌های ایمیل را خودکار کنید، و غیره، نشان می‌دهد که یک فرد چقدر می‌تواند کارهای خود را خودکار کند. ماشین‌های بليط، خريد بليط قطار را خودکار می‌کنند (ديگر نيازي به صندوق‌دار نیست)، ماشین‌های لباسشوبي، شستشوی لباس‌ها را خودکار می‌کنند، سفارش‌های دائمي، معاملات پرداخت را خودکار می‌کنند و غیره. خودکار کردن وظایيف زمان و پول را آزاد می‌کند، بنابراین انگيزه اقتصادي و شخصي زيادي برای خودکار کردن کارها وجود دارد. ما در حال حاضر در حال مشاهده اتوماسیون ترجمه زبان، رانندگی و تا حدی حتی کشف علمی هستیم.

۳- **تشخيص اشتباه:** ما نمی‌توانیم یک هدف را با تمام محدودیت‌هاییش کاملاً تشخیص دهیم. به غول در بطری فکر کنید که همیشه خواسته‌های شما را به معنای واقعی کلمه می‌پذیرد: "من می‌خواهم ثروتمندترین فرد جهان باشم!" -> شما تبدیل به ثروتمندترین فرد می‌شوید، اما به عنوان یک عارضه جانبی، ارزی که در اختیار دارید به دلیل تورم سقوط می‌کند.

"من می‌خواهم تا آخر عمرم خوشحال باشم!" -> ۵ دقیقه بعد از این که خیلی احساس خوشبختی می‌کنی، غول تو را می‌کشد.

¹ Misspecification

"آرزوی صلح جهانی دارم! " -> غول همه انسان‌ها را می‌کشد.

ما اهداف را به اشتباه تشخیص می‌کنیم، یا به این دلیل که همه محدودیت‌ها را نمی‌دانیم یا به این دلیل که نمی‌توانیم آنها را اندازه گیری کنیم. بیایید به شرکت‌ها به عنوان نمونه ای از تشخیصات هدف ناقص نگاه کنیم. یک شرکت هدف ساده کسب درآمد برای سهامداران خود دارد. اما این تشخیصات هدف واقعی را با تمام محدودیت‌هایش که ما واقعاً برای آن تلاش می‌کنیم، نشان نمی‌دهد: به عنوان مثال، ما از شرکتی که مردم را برای کسب درآمد می‌کشد، رودخانه‌ها را مسموم می‌کند یا صرفاً پول خود را چاپ می‌کند قدردانی نمی‌کنیم. ما قوانین، مقررات، تحریم‌ها، رویه‌های انطباق، اتحادیه‌های کارگری و موارد دیگر را برای اصلاح تشخیصات هدف ناقص ابداع کردہ‌ایم. نمونه دیگری که می‌توانید خودتان تجربه کنید، گیره کاغذ است، بازی ای که در آن با یک ماشین با هدف تولید هرچه بیشتر گیره کاغذ بازی می‌کنید. هشدار: اعتیاد آور است. من نمی‌خواهم آن را خیلی خراب کنم، اما بیایید بگوییم که همه چیز خیلی سریع از کنترل خارج می‌شود. در یادگیری ماشین، نقص در تشخیصات هدف ناشی از انتزاع داده‌های ناقص (جمعیت‌های مغرضانه، خطاهای اندازه گیری، و ...)، توابع از دست دادن نامحدود، عدم آگاهی از محدودیت‌ها، تغییر توزیع بین داده‌های آموزشی و برنامه کاربردی و بسیاری موارد دیگر است.

دیجیتالی شدن اتوماسیون رانندگی است. تشخیصات هدف ناقص با اتوماسیون در تعارض است. من ادعا می‌کنم که این تعارض تا حدی با روش‌های تفسیری جبران می‌شود. صحنه برای پیش‌بینی‌های ما آماده شده است، جام جم آماده است، اکنون ما نگاه می‌کنیم که زمین به کجا می‌تواند برسد

11. آینده یادگیری ماشینی

بدون یادگیری ماشینی، هیچ یادگیری ماشینی قابل تفسیری وجود ندارد. بنابراین، قبل از اینکه بتوانیم در مورد تفسیرپذیری صحبت کیم، باید حدس بزنیم که یادگیری ماشین به کجا می‌رود.

یادگیری ماشینی (یا "AI") با وعده‌ها و انتظارات زیادی همراه است. اما اجازه دهید با یک مشاهده کمتر خوش بینانه شروع کنیم: در حالی که علم ابزارهای یادگیری ماشینی فانتزی زیادی را توسعه می‌دهد، در تجربه من، ادغام آنها در فرآیندها و محصولات موجود بسیار دشوار است. نه به این دلیل که امکان پذیر نیست، بلکه صرفاً به این دلیل که شرکت‌ها و مؤسسات به زمان نیاز دارند تا به آن برسند. در هجوم و هیاهوی فعلی هوش مصنوعی، شرکت‌ها «آزمایشگاه‌های هوش مصنوعی»، «واحدهای یادگیری ماشین» را باز می‌کنند و «دانشمندان داده»، «متخصصان یادگیری ماشین»، «مهندسين هوش مصنوعی» و غیره را استخدام می‌کنند، اما واقعیت این است که در تجربه من، نسبتاً بی فایده است. اغلب شرکت‌ها حتی داده‌هایی به شکل مورد نیاز ندارند و دانشمندان داده ماهها بیکار می‌مانند. گاهی اوقات شرکت‌ها به دلیل رسانه‌ها، انتظارات زیادی از هوش مصنوعی و علم داده دارند که دانشمندان داده هرگز نمی‌توانند آنها را برآورده کنند. و اغلب هیچ کس نمی‌داند چگونه دانشمندان داده را در ساختارهای موجود و بسیاری از مسائل دیگر ادغام کند. این منجر به اولین پیش‌بینی من می‌شود: یادگیری ماشینی به آرامی اما پیوسته رشد خواهد کرد.

دیجیتالی شدن در حال پیشرفت است و وسوسه خودکارسازی دائمًا در حال افزایش است. حتی اگر مسیر پذیرش یادگیری ماشین کند و سنگلاخ باشد، یادگیری ماشین به طور مداوم از علم به فرآیندهای تجاری، محصولات و کاربردهای دنیای واقعی در حال حرکت است.

من معتقدم که باید به افراد غیر متخصص بهتر توضیح دهیم که چه نوع مسائلی را می‌توان به عنوان مسائل یادگیری ماشین فرموله کرد. من بسیاری از دانشمندان داده پردرآمد را می‌شناسم که به جای یادگیری ماشینی از محاسبات اکسل یا هوش تجاری کلاسیک را با گزارش و پرس و جوهای SQL استفاده می‌کنند. اما چند شرکت در حال حاضر با موفقیت از یادگیری ماشینی استفاده می‌کنند و شرکت‌های بزرگ اینترنتی در خط مقدم هستند. ما باید راههای بهتری برای ادغام یادگیری ماشینی در فرآیندها و محصولات، آموزش افراد و توسعه ابزارهای یادگیری ماشینی که استفاده از آنها آسان باشد، پیدا کنیم. من معتقدم که استفاده از یادگیری ماشین بسیار آسان‌تر خواهد شد: ما می‌توانیم در حال حاضر ببینیم که یادگیری ماشینی در دسترس‌تر می‌شود، به عنوان مثال از طریق سرویس‌های ابری ("یادگیری ماشین به عنوان یک سرویس" - فقط برای استفاده از چند واژه باب روز^۱). هنگامی که یادگیری ماشین به بلوغ رسید - و این کودک نوپا اولین قدم‌های خود را برداشته است - پیش‌بینی بعدی من این است:

¹ buzzwords

یادگیری ماشینی به چیزهای زیادی دامن می‌زند.

بر اساس اصل «هر آنچه که می‌تواند خودکار شود، خودکار خواهد شد»، نتیجه می‌گیرم که در صورت امکان، وظایف به عنوان مسائل پیش‌بینی فرموله می‌شوند و با یادگیری ماشین حل می‌شوند. یادگیری ماشین نوعی اتوماسیون است یا حداقل می‌تواند بخشی از آن باشد. بسیاری از وظایفی که در حال حاضر توسط انسان انجام می‌شود با یادگیری ماشینی جایگزین شده است. در اینجا چند نمونه از وظایفی وجود دارد که در آنها از یادگیری ماشین برای خودکارسازی بخش‌هایی از آن استفاده می‌شود:

- مرتب سازی / تصمیم‌گیری / تکمیل اسناد (به عنوان مثال در شرکت‌های بیمه، بخش حقوقی یا شرکت‌های مشاوره)
- تصمیمات مبتنی بر داده مانند برنامه‌های اعتباری
- کشف مواد مخدر
- کنترل کیفیت در خطوط مونتاژ
- ماشین‌های خودران
- تشخیص بیماری‌ها
- ترجمه. برای این کتاب، من از یک سرویس ترجمه به نام (DeepL) استفاده کردم که توسط شبکه‌های عصبی عمیق پشتیبانی می‌شود تا جملات خود را با ترجمه آنها از انگلیسی به آلمانی و دوباره به انگلیسی بهبود بخشم.
- ...

پیشرفت برای یادگیری ماشین نه تنها از طریق رایانه‌های بهتر / داده‌های بیشتر / نرم افزار بهتر به دست می‌آید، بلکه:

ابزارهای تفسیرپذیری مقبولیت یادگیری ماشین را تسريع می‌کنند.

بر اساس این فرض که هدف یک مدل یادگیری ماشینی هرگز نمی‌تواند به طور کامل تشخیص داده شود، نتیجه می‌شود که یادگیری ماشین قابل تفسیر برای بستن شکاف بین هدف نادرست و هدف واقعی ضروری است. در بسیاری از حوزه‌ها و بخش‌ها، تفسیرپذیری کاتالیزوری برای مقبولیت یادگیری ماشین خواهد بود. برخی تجربه‌های شخصی: بسیاری از افرادی که با آنها صحبت کرده ام از یادگیری ماشینی استفاده نمی‌کنند زیرا نمی‌توانند مدل‌ها را برای دیگران توضیح دهند. من معتقدم که تفسیرپذیری به این موضوع می‌پردازد و یادگیری ماشینی را برای سازمان‌ها و افرادی که خواهان شفافیت هستند جذاب می‌کند. علاوه بر تسریع نادرست مشکل، بسیاری از صنایع به دلایل قانونی، به دلیل ریسک گریزی یا به دست آوردن بینش در مورد وظیفه اساسی، نیاز به تفسیر دارند. یادگیری ماشینی فرآیند مدل‌سازی را خودکار می‌کند و انسان را کمی دورتر

از داده‌ها و کار اصلی می‌برد: این امر خطر بروز مشکلات در طراحی آزمایشی، انتخاب توزیع آموزش، نمونه‌برداری، رمزگذاری داده‌ها، مهندسی ویژگی و غیره را افزایش می‌دهد. ابزارهای تفسیری شناسایی این مشکلات را آسان تر می‌کند.

۱۱.۲ آینده تفسیرپذیری

اجازه دهد نگاهی به آینده احتمالی تفسیرپذیری یادگیری ماشین بیندازیم.

تمرکز بر ابزارهای تفسیرپذیری مدل-آگنوستیک خواهد بود.

تفسیرپذیری وقتی از مدل یادگیری ماشینی جدا شده باشد، خودکارسازی بسیار آسان‌تر است. مزیت تفسیرپذیری مدل-آگنوستیک در مدولار بودن آن نهفته است. ما به راحتی می‌توانیم مدل یادگیری ماشینی اساسی را جایگزین کنیم. به همین راحتی می‌توانیم روش تفسیر را جایگزین کنیم. به این دلایل، روش‌های مدل-آگنوستیک بسیار بهتر مقیاس خواهند شد. به همین دلیل است که من معتقد‌ام که روش‌های مدل-آگنوستیک در دراز مدت غالب‌تر خواهند شد. اما روش‌های قابل تفسیر ذاتی نیز جایگاهی خواهند داشت.

یادگیری ماشینی خودکار خواهد شد و به همراه آن قابلیت تفسیر نیز خودکار خواهد شد.

یک روند قابل مشاهده، اتوماسیون آموزش مدل است. این شامل مهندسی خودکار و انتخاب ویژگی‌ها، بهینه‌سازی خودکار هایپرپارامتر، مقایسه مدل‌های مختلف، و مجموعه‌بندی یا انباسته کردن مدل‌ها می‌شود. نتیجه بهترین مدل پیش‌بینی ممکن است. وقتی از روش‌های تفسیر مدل-آگنوستیک استفاده می‌کنیم، می‌توانیم به طور خودکار آنها را برای هر مدلی که از فرآیند یادگیری ماشین خودکار پدید می‌آید اعمال کنیم. به نوعی، می‌توانیم این مرحله دوم را نیز خودکار کنیم: به طور خودکار اهمیت ویژگی را محاسبه کنید، و استگی جزئی را ترسیم کنید، یک مدل جایگزین را آموزش دهید، و غیره. هیچ کس شما را از محاسبه خودکار تمام این تفاسیر مدل باز نمی‌دارد. تفسیر واقعی هنوز به مردم نیاز دارد. تصور کنید: شما یک مجموعه داده را آپلود می‌کنید، هدف پیش‌بینی را مشخص می‌کنید و با فشار دادن یک دکمه بهترین مدل پیش‌بینی آموزش داده می‌شود و برنامه تمام تفاسیر مدل را بیرون می‌دهد. در حال حاضر اولین محصولات وجود دارد و من استدلال می‌کنم که برای بسیاری از برنامه‌ها استفاده از این خدمات یادگیری ماشین خودکار کافی است. امروزه هر کسی می‌تواند بدون دانستن HTML، CSS و Javascript وب سایت بسازد، اما هنوز هم توسعه دهنده‌گان وب زیادی در اطراف وجود دارند. به همین ترتیب، من معتقد‌ام که همه می‌توانند مدل‌های یادگیری ماشینی را بدون اطلاع از نحوه برنامه‌نویسی آموزش دهند و همچنان به متخصصان یادگیری ماشین نیاز است.

ما داده‌ها را تجزیه و تحلیل نمی‌کنیم، ما مدل‌ها را تجزیه و تحلیل می‌کنیم.

خود داده‌های خام همیشه بی‌فایده هستند. (من عمداً اغراق می‌کنم. واقعیت این است که برای انجام یک تحلیل معنی‌دار به درک عمیق داده‌ها نیاز دارید). من به داده‌ها اهمیتی نمی‌دهم. من به دانش موجود در داده‌ها اهمیت می‌دهم. یادگیری ماشینی قابل تفسیر راهی عالی برای استخراج دانش از داده‌ها است. شما می‌توانید مدل را به طور گسترده بررسی کنید، مدل به طور خودکار تشخیص می‌دهد که آیا و چگونه ویژگی‌ها برای پیش‌بینی مرتبط هستند (بسیاری از مدل‌ها دارای انتخاب ویژگی داخلی هستند)، مدل می‌تواند به طور

خودکار نحوه نمایش روابط را تشخیص دهد، و - اگر به درستی آموزش داده شود - مدل نهایی تقریب بسیار خوبی از واقعیت است.

بسیاری از ابزارهای تحلیلی در حال حاضر مبتنی بر مدل‌های داده هستند (زیرا آنها بر اساس مفروضات توزیع هستند):

- آزمون‌های فرضیه ساده مانند آزمون تی استیوونت.
- آزمون‌های فرضیه با تنظیمات برای مخدوش کننده‌ها (GLM) معمولاً
- تجزیه و تحلیل واریانس (ANOVA)
- ضریب همبستگی (ضریب رگرسیون خطی استاندارد شده مربوط به ضریب همبستگی پیرسون است)
- ...

آنچه در اینجا به شما می‌گوییم در واقع چیز جدیدی نیست. پس چرا از تحلیل مدل‌های شفاف و مبتنی بر فرض به تحلیل مدل‌های جعبه سیاه بدون فرض تغییر می‌کنیم؟ زیرا ایجاد همه این مفروضات مشکل ساز است: آنها معمولاً اشتباه هستند (مگر اینکه معتقد باشید که اکثر جهان از توزیع گاوی پیروی می‌کنند)، بررسی آنها دشوار است، بسیار انعطاف ناپذیر و خودکار کردن آنها دشوار است. در بسیاری از حوزه‌ها، مدل‌های مبتنی بر فرض معمولاً نسبت به مدل‌های یادگیری ماشین جعبه سیاه، عملکرد پیش‌بینی بدتری روی داده‌های تست دست‌نخورده دارند. این فقط برای مجموعه داده‌های بزرگ صادق است، زیرا مدل‌های قابل تفسیر با مفروضات خوب اغلب با مجموعه داده‌های کوچک بهتر از مدل‌های جعبه سیاه عمل می‌کنند. روش یادگیری ماشین جعبه سیاه برای اینکه خوب کار کند به داده‌های زیادی نیاز دارد. با دیجیتالی شدن همه چیز، مجموعه داده‌های بزرگتری خواهیم داشت و بنابراین رویکرد یادگیری ماشین جذاب‌تر می‌شود. ما فرضیات نمی‌کنیم، ما واقعیت را تا حد ممکن نزدیک می‌کنیم (در حالی که از تطبیق بیش از حد داده‌های آموزشی اجتناب می‌کنیم). من استدلال می‌کنم که ما باید تمام ابزارهایی را که در آمار داریم برای پاسخ به سؤالات (آزمون‌های فرضیه، معیارهای همبستگی، معیارهای تعامل، ابزارهای تجسم، فواصل اطمینان، مقادیر p، فواصل پیش‌بینی، توزیع‌های احتمال) توسعه دهیم و آنها را برای مدل‌های جعبه سیاه بازنویسی کنیم. به نوعی، این در حال حاضر اتفاق می‌افتد:

- اجازه دهید یک مدل خطی کلاسیک را در نظر بگیریم: ضریب رگرسیون استاندارد شده در حال حاضر یک معیار اهمیت ویژگی است. با اندازه‌گیری اهمیت ویژگی جایگشت، ابزاری داریم که با هر مدلی کار می‌کند.
- در یک مدل خطی، ضرایب اثر یک ویژگی واحد را بر نتیجه پیش‌بینی شده اندازه‌گیری می‌کنند. نسخه تعمیم یافته این طرح وابستگی جزئی است.

- تست بهتر بودن A یا B برای این کار می توانیم از توابع وابستگی جزئی نیز استفاده کنیم. چیزی که ما هنوز نداریم (تا جایی که من می دانم) تست های آماری برای مدل های جعبه سیاه دلخواه است. دانشمندان داده، خود را خودکار می کنند.

من معتقدم که دانشمندان داده در نهایت برای بسیاری از وظایف تحلیل و پیش‌بینی خود را از کار خارج خواهند کرد. برای اینکه این اتفاق بیفتد، وظایف باید به خوبی تعریف شده باشند و فرآیندها و روال‌هایی در اطراف آنها وجود داشته باشد. امروزه، این روال‌ها و فرآیندها گم شده‌اند، اما دانشمندان داده و همکاران روی آنها کار می کنند. همانطور که یادگیری ماشین به بخشی جدایی ناپذیر از بسیاری از صنایع و موسسات تبدیل می‌شود، بسیاری از وظایف به صورت خودکار انجام می‌شود.

ربات‌ها و برنامه‌ها خودشان توضیح خواهند داد.

ما به رابطه‌های بصری بیشتری برای ماشین‌ها و برنامه‌هایی نیاز داریم که از یادگیری ماشینی استفاده زیادی می‌کنند. چند مثال: یک خودروی خودران که دلیل توقف ناگهانی آن را گزارش می‌دهد ("احتمال ۷۰٪ عبور یک کودک از جاده"). یک برنامه پیش‌فرض اعتبار که به کارمند بانک توضیح می‌دهد که چرا درخواست اعتبار رد شده است ("متقاضی کارت‌های اعتباری زیادی دارد و در شغلی ناپایدار مشغول به کار است."). یک بازوی رباتی که توضیح می‌دهد که چرا کالا را از تسمه نقاله به سطل زباله منتقل کرده است ("محصول در پایین آن یک جنون دارد.").

تفسیرپذیری می تواند تحقیقات هوش ماشینی را تقویت کند.

می‌توانم تصور کنم که با انجام تحقیقات بیشتر در مورد اینکه چگونه برنامه‌ها و ماشین‌ها می‌توانند خودشان را توضیح دهند، می‌توانیم در ک خود را از هوش بهبود بخشیم و در ایجاد ماشین‌های هوشمند بهتر شویم. در نهایت، همه این پیش‌بینی‌ها حدس و گمان هستند و باید دید آینده واقعاً چه چیزی را به همراه خواهد داشت. نظر خود را شکل دهید و به یادگیری ادامه دهید!

فصل ۱۳ با استناد به این کتاب

اگر این کتاب را برای پست و بلاگ، مقاله تحقیقاتی یا محصول خود مفید دیدید، ممنون می‌شوم اگر به این کتاب استناد کنید. شما می‌توانید کتاب را به این صورت استناد کنید:

Molnar, C. (2022). Interpretable Machine Learning:
A Guide for Making Black Box Models Explainable (2nd ed.).
christophm.github.io/interpretable-ml-book/

یا از ورودی bibtex زیر استفاده کنید:

```
@book{molnar2022,  
  title = {Interpretable Machine Learning(,  
  author = {Christoph Molnar(,  
  year = {2022(,  
  subtitle = {A Guide for Making Black Box Models Explainable(,  
  edition = {2(,  
  url = {https://christophm.github.io/interpretable-ml-book/(
```

من همیشه کنجکاو هستم که کجا و چگونه از روش‌های تفسیر در صنعت و تحقیق استفاده می‌شود. اگر از کتاب به عنوان مرجع استفاده می‌کنید، خیلی خوب می‌شود اگر یک خط برای من بنویسید و بگویید برای چه این البته اختیاری است و فقط برای ارضای کنجکاوی خودم و تحریک مبادلات جالب است. ایمیل من christoph.molnar.ai@gmail.com است.

فصل ۱۴ ترجمه‌ها

به ترجمه کتاب علاقه دارید؟

این کتاب تحت مجوز Creative Commons Attribution-NonCommercial-ShareAlike 4.0 بین المللی مجوز دارد. یعنی شما مجاز به ترجمه و قرار دادن آن در اینترنت هستید. شما باید من را به عنوان نویسنده اصلی ذکر کنید و اجازه فروش کتاب را ندارید.

اگر علاقه‌مند به ترجمه کتاب هستید می‌توانید پیام بنویسید و ترجمه شما را اینجا لینک کنم. آدرس من christoph.molnar.ai@gmail.com

فهرست ترجمه‌ها

باهاسا اندونزی

ترجمه کامل توسط Smart City & Cybersecurity Laboratory, Information و Hatma Suryotrisongko Technology, ITS.

چینی:

ترجمه کامل نسخه دوم توسط Jiazen CSDN از . یک جامعه آنلاین برنامه نویسان.

ترجمه‌های کامل توسط Mingchao Zhu. نسخه الکترونیکی و چاپی این ترجمه موجود است.

ترجمه اکثر فصول توسط CSDN.

ترجمه چند فصل. این وب سایت همچنین شامل سوالات و پاسخ‌های کاربران مختلف است.

ژاپنی

ترجمه کامل توسط Ryoji Masoe و تیم HACARUS.

کره ای:

ترجمه کامل کره ای توسط TooTouch

ترجمه جزئی کره ای توسط An Subin

اسپانیایی

ترجمه کامل اسپانیایی توسط Fderiko فلیگر

ویتنامی

ترجمه کامل Hoang Nguyen، Tri Le، Hung-Quang Nguyen، Duy-Tung Nguyen، Giang Nguyen و.

اگر ترجمه دیگری از کتاب یا هر فصل دیگری می‌شناسید، ممنون می‌شوم که درباره آن بشنوم و آن را در اینجا فهرست کنید. می‌توانید از طریق ایمیل با من تماس بگیرید: christoph.molnar.ai@gmail.com.

فصل ۱۵ سپاسگزاری‌ها

نوشتن این کتاب بسیار سرگرم کننده بود (و هنوز هم هست). اما کار باقیمانده هم زیاد است و از حمایتی که دریافت کردم بسیار خوشحالم.

بزرگترین تشکر من از کاترین است که سخت‌ترین کار را از نظر ساعت و تلاش داشت: او کتاب را از ابتدای آنها تصحیح کرد و بسیاری از اشتباهات املایی و تناقضات را کشف کرد که من هرگز آنها را پیدا نمی‌کردم. من از حمایت او بسیار سپاسگزارم.

از همه نویسنندگان مهمان تشکر می‌کنم. من واقعاً متعجب شدم وقتی فهمیدم مردم علاقه‌مند به مشارکت در این کتاب هستند. و به لطف تلاش آنها، می‌توان مطالب کتاب را بهبود بخشید! Tobias Goerke و Magdalena Lang فصلی را در مورد قوانین محدوده (لنگرهای) نوشتند. Fangzhou Li در بخش تشخیص مفاهیم مشارکت داشت. و Susanne Dandl فصل مربوط به مثال‌های خلاف واقع را بسیار بهبود بخشید. آخرین اما نه کم اهمیت، Verena Haunschmid بخش توضیحات LIME برای تصاویر را نوشت. همچنین می‌خواهم از همه خوانندگانی که بازخورد و مشارکتشان را مستقیماً در GitHub ارائه کردند تشکر کنم!

علاوه بر این، من می‌خواهم از همه کسانی که تصاویر را تولید کردند تشکر کنم: جلد توسط دوست من @YvonneDoinel طراحی شده است. گرافیک‌های فصل مقدار Shapley Value (Shapley Value) و همچنین نمونه‌لایک پشت در فصل نمونه‌های متخاصل توسط Heidi Seibold ایجاد شده است. Verena Haunschmid این گرافیک‌های RuleFit را ایجاد کرد.

همچنین از همسر و خانواده ام که همیشه از من حمایت کردند تشکر می‌کنم. به خصوص همسرم مجبور بود به صحبت‌های من درباره کتاب گوش دهد. او به من کمک کرد تا تصمیمات زیادی در مورد نوشتن کتاب بگیرم. نحوه انتشار این کتاب کمی غیر متعارف است. اولاً، این نه تنها به عنوان جلد شومیز و کتاب الکترونیکی، بلکه به عنوان یک وب سایت، کاملاً رایگان در دسترس است. نرم افزاری که من برای ایجاد این کتاب استفاده کردم، bookdown نام دارد، که توسط Yihui Xie نوشته شده است. ایشان بسیاری از بسته‌های R را ایجاد کرد که ترکیب کد R و متن را آسان می‌کند. خیلی ممنون! من کتاب را به عنوان کتاب در دست انتشار منتشر کردم که به من کمک زیادی کرد تا بازخورد دریافت کنم و در طول مسیر از آن درآمدزایی کنم. همچنین می‌خواهم از شما خواننده عزیز تشکر کنم که این کتاب را بدون این که ناشر بزرگی داشته باشد، خواندید.

من از بودجه تحقیقاتم در مورد یادگیری ماشین قابل تفسیر توسط وزارت علوم و هنرهای ایالت باواریا در چارچوب مرکز دیجیتال سازی باواریا (Z.D.B) و موسسه تحقیقاتی باواریا برای تحول دیجیتال (bidt) سپاسگزارم.

منابع

- Aamodt, A., & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 7(1), 39-59 .
- Alberto, T. C., Lochter, J. V., & Almeida, T. A. (2015). Tubespam: Comment spam filtering on youtube. 2015 IEEE 14th international conference on machine learning and applications (ICMLA) ,
- Altmann, A., Tološi, L., Sander, O., & Lengauer, T. (2010). Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10), 1340-1347 .
- Alvarez-Melis, D., & Jaakkola, T. S. (2018). On the robustness of interpretability methods. *arXiv preprint arXiv:1806.08049* .
- Apley, D. W., & Zhu, J. (2020). Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 82(4), 1059-1086 .
- Borgelt, C. (2005). An Implementation of the FP-growth Algorithm. Proceedings of the 1st international workshop on open source data mining: frequent pattern mining implementations ,
- Breiman, L. (2001). Random forests. *Machine learning*, 45, 5-32 .
- Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., & Elhadad, N. (2015). Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining ,
- Cohen, W. W. (1995). Fast effective rule induction. In *Machine learning proceedings 1995* (pp. 115-123). Elsevier .
- Dandl, S., Molnar, C., Binder, M., & Bischl, B .(۲۰۱۷) .Multi-objective counterfactual explanations. International Conference on Parallel Problem Solving from Nature ,
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2), 182-197 .
- Definition of Algorithm.* (2017). <https://www.merriam-webster.com/dictionary/algorithm>
- Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1705.07115* .
- Fanaee-T, H., & Gama, J. (2014). Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, 2, 113-127 .
- Fernandes, K., Cardoso, J. S., & Fernandes, J. (2017). Transfer learning with partial observability applied to cervical cancer screening. Pattern Recognition and Image Analysis: 8th Iberian Conference, IbPRIA 2017, Faro, Portugal, June 20-23, 2017, Proceedings 8 ,
- Fisher, A., Rudin, C., & Dominici, F. (2019). All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously. *J. Mach. Learn. Res.*, 20(177), 1-81 .
- Fokkema, M. (2017). Fitting prediction rule ensembles with R package pre. *arXiv preprint arXiv:1707.07149* .
- Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189-1232 .
- Friedman, J. H., & Popescu, B. E. (2008). Predictive learning via rule ensembles .
- Fürnkranz, J., Gamberger, D., & Lavrač, N. (2012). *Foundations of rule learning*. Springer Science & Business Media .
- Goldstein, A., Kapelner, A., Bleich, J., & Kapelner, M. A. (2017). Package 'ICEbox .' .
- Goldstein, A., Kapelner, A., Bleich, J., & Pitkin, E. (2015). Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics*, 24(1), 44-65 .

- Greenwell, B. M., Boehmke, B. C., & McCarthy, A. J. (2018). A simple and effective model-based variable importance measure. *arXiv preprint arXiv:1805.04755*.
- Grömping, U. (2020). Model-agnostic effects plots for interpreting machine learning models. *Reports in Mathematics, Physics and Chemistry, Department II, Beuth University of Applied Sciences Berlin Report, 1*, 2020.
- Gurumoorthy, K. S., Dhurandhar, A., Cecchi, G., & Aggarwal, C. (2019). Efficient data representation by selecting prototypes with importance weights. 2019 IEEE International Conference on Data Mining (ICDM),
- Hastie, T. (2009). The Elements of Statistical Learning Edited by Hastie T, Tibshirani R, Friedman J. In: Springer New York.:
- Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction* (Vol. 2). Springer .
- Heider, F & ,Simmel, M. (1944). An experimental study of apparent behavior. *The American journal of psychology*, 57(2), 243-259 .
- Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine learning*, 11, 63-90 .
- Hooker, G. (2004). Discovering additive structure in black box functions. Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining ,
- Hooker, G. (2007). Generalized functional anova diagnostics for high-dimensional functions of dependent variables. *Journal of Computational and Graphical Statistics*, 16(3), 709-732 .
- Inglis, A., Parnell, A., & Hurley, C. B. (2022). Visualizing variable importance and variable interaction effects in machine learning models. *Journal of Computational and Graphical Statistics*, 31(3), 766-778 .
- Janzing, D., Minorics, L., & Blöbaum, P. (2020). Feature relevance quantification in explainable AI: A causal problem. International Conference on artificial intelligence and statistics ,
- Kahneman, D & ,Tversky, A. (1981). *The simulation heuristic*. National Technical Information Service .
- Karimi, A.-H., Barthe, G., Balle, B., & Valera, I. (2020). Model-agnostic counterfactual explanations for consequential decisions. International Conference on Artificial Intelligence and Statistics ,
- Kaufman, L., & Rousseeuw, P. J. (1987). Clustering by means of Medoids. Statistical data analysis based on the L1-norm and related methods, edited by Y. Dodge. In: North-Holland.
- Kaufmann, E., & Kalyanakrishnan, S. (2013).Information complexity in bandit subset selection. Conference on Learning Theory ,
- Kim, B., Khanna, R., & Koyejo, O. O. (2016). Examples are not enough, learn to criticize! criticism for interpretability. *Advances in neural information processing systems*, 29 .
- Laugel, T., Lesot, M.-J., Marsala, C., Renard, X., & Detyniecki, M. (2017). Inverse classification for comparison-based interpretability in machine learning. *arXiv preprint arXiv:1712.08443* .
- Letham, B., Rudin, C., McCormick, T. H., & Madigan, D .(Y .10) .Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model .
- Lipton, P. (1990). Contrastive explanation. *Royal Institute of Philosophy Supplements*, 27, 247-266 .
- Lipton, Z. C. (2018). The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3), 31-57 .
- Lundberg, S. M., Erion, G. G., & Lee, S.-I. (2018). Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888* .
- Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30 .
- Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267, 1-38 .

- Mothilal, R. K., Sharma, A., & Tan, C. (2020). Explaining machine learning classifiers through diverse counterfactual explanations. Proceedings of the 2020 conference on fairness, accountability, and transparency ,
- Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., & Yu, B. (2019). Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44), 22071-22080 .
- Nickerson, R. S. (1998). Confirmation bias: A ubiquitous phenomenon in many guises. *Review of general psychology*, 2(2), 175-220 .
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016a). Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386* .
- Ribeiro, M. T ,Singh, S., & Guestrin, C. (2016b). " Why should i trust you?" Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining ,
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Anchors: High-precision model-agnostic explanations. Proceedings of the AAAI conference on artificial intelligence ,
- Robnik-Šikonja, M., & Bohanec, M. (2018). Perturbation-based explanations of prediction models. *Human and Machine Learning: Visible ,Explainable, Trustworthy and Transparent*, 159-175 .
- Shapley, L. S. (1953). A value for n-person games. Contributions to the Theory of Games 2, 28 (1953), 307–317. In.
- Slack, D., Hilgard, S., Jia, E., Singh, S., & Lakkaraju, H. (2020). Fooling lime and shap: Adversarial attacks on post hoc explanation methods. Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society ,
- Staniak, M., & Biecek, P. (2018). Explanations of model predictions with live and breakDown packages. *arXiv preprint arXiv:1804.019 .00*
- Štrumbelj, E., & Kononenko, I. (2011). A general method for visualizing and explaining black-box regression models. Adaptive and Natural Computing Algorithms: 10th International Conference, ICANNGA 2011, Ljubljana, Slovenia, April 14-16, 2011, Proceedings, Part II 10 ,
- Štrumbelj, E., & Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41, 647-665 .
- Sundararajan, M., & Najmi, A. (2020). The many Shapley values for model explanation. International conference on machine learning ,
- Van Looveren, A., & Klaise, J. (2021). Interpretable counterfactual explanations guided by prototypes. Joint European Conference on Machine Learning and Knowledge Discovery in Databases ,
- Wachter, S., Mittelstadt, B., & Russell, C. (2017). Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harv. JL & Tech.*, 31, 841 .
- Wei, P., Lu, Z., & Song, J. (2015). Variable importance analysis: A comprehensive review. *Reliability Engineering & System Safety*, 142, 399-432 .
- Yang, H., Rudin, C., & Seltzer, M. (2017). Scalable Bayesian rule lists. International conference on machine learning ,
- Zhao, Q., & Hastie, T. (2021). Causal interpretations of black-box models. *Journal of Business & Economic Statistics*, 39(1), 272-281 .