1. مقایسه رگرسیون لجستیک، LDA و ماشین‌های بردار پشتیبان:

رگرسیون لجستیک یک مدل discriminative است که احتمالات را بر اساس مرزهای تصمیم خطی تخمین می زند. هیچ توزیع خاصی برای ویژگی ها در نظر نمی گیرد اما از یک تابع لجستیک برای طبقه بندی باینری استفاده می کند.

LDA یک مدل generative است که فرض می‌کند ویژگی‌های هر کلاس از یک توزیع گاوسی با میانگین مخصوص کلاس اما یک ماتریس کوواریانس مشترک پیروی می‌کند LDA با یافتن ترکیب خطی از ویژگی هایی که به بهترین وجه دو یا چند کلاس را از هم جدا می کند، کار می کند.

SVM روی یافتن هایپرپلن بهینه تمرکز دارد که حاشیه بین کلاس‌ها را به حداکثر می‌رساند و زمانی که داده‌ها به‌طور خطی از طریق ترفندهای مربوط به کرنل‌ها قابل تفکیک نیستند، مفید است.

شرایطی که LR ممکن است بهتر از LDA عمل کند:

اگر مفروضات LDA (توزیع گاوسی، کوواریانس برابر) برقرار نباشد، رگرسیون لجستیک ممکن است بهتر عمل کند زیرا به آن مفروضات نیاز ندارد.

هنگامی که مجموعه داده دارای ویژگی های غیر گاوسی باشد، رگرسیون لجستیک نسبت به LDA انعطاف پذیرتر است.

شرایطی که LDA ممکن است بهتر از LR عمل کند:

LDA می تواند در مواردی که مفروضات آن درست است از رگرسیون لجستیک بهتر عمل کند، زیرا با فرض توزیع های گاوسی، به ویژه با حجم نمونه کوچک، ساختار بیشتری را در داده ها اعمال می کند.

داده های با ابعاد بالا: SVM اغلب بر LDA در فضاهای با ابعاد بالا ترجیح داده می شود زیرا از ترفندهای مربوط به کرنل برای کشیدن مرزهای تصمیم گیری پیچیده استفاده می کند و برای ابعاد بالا قوی تر است، جایی که LDA ممکن است به دلیل مسائل تخمین ماتریس کوواریانس با مشکل مواجه شود.

2. مقایسه SVM و رگرسیون بردار پشتیبان SVR

SVM برای مشکلات طبقه بندی استفاده می شود. هایپرپلینی را پیدا می کند که مرز بین دو کلاس را به حداکثر می رساند.

SVR توسعه‌یSVM برای مسائل رگرسیون است. به جای جدا کردن داده ها به کلاس ها، هدف SVR قرار دادن داده ها در میان دو خط با عرض ε است(فاصله‌ی بین دو خط در واقع 2ε است و فاصله‌ی بین خط رگرسیون و خطوط حاشیه‌ای ε است) و انحرافات کوچک را نادیده می گیرد و فقط انحرافات بزرگتر را جریمه می کند.

فرمول مربوط به SVR :





ϵ به این معنی است که ما چقدر خطا را می خواهیم داشته باشیم.

3 . تبدیل SVC به فرم ضرب خطی:

هدف SVC یافتن ابر صفحه ای است که حاشیه بین دو کلاس را به حداکثر برساند. مسئله بهینه سازی را می توان به صورت زیر بیان کرد:



همچنین چنین محدودیتی هم برای این مورد داریم:



اپسیلون متغیر slack یا سستی است که misclassified را اجازه می‌دهد و متغیر C هم پارامتر regularization است.

در قدم بعدی با ضرب لاگرانژ را انجام دهیم:



که آلفا ضریب لاگرانژ است و میو ضریب برای محدودیت‌های غیر منفی بر روی اپسیلون است.

برای یافتن مقادیر بهینه، مشتقات لاگرانژ را با توجه به β، β0، ξ به دست می‌اوریم و حاصل را برابر صفر قرار می‌دهیم.







با جایگذاری



در فرمول لاگرانژ، به فرمول زیر خواهیم رسید:



توجه داشته باشید که :



در نهایت با جایگذاری β در فرمول تابع تصمیم می‌توانیم به فرمول زیر که مقصود ماست برسیم:



4. مقایسه کرنل‌های SVM:

کرنل خطی: برای داده های قابل جداسازی خطی استفاده می شود. تابع تصمیم یک ضرب داخلی ساده است.

کرنل چند جمله ای: روابط چند جمله ای را شامل می‌شود. فرمول آن به صورت ⟨x,xi​⟩d +c است و ثابت c تضمین می کند که حتی داده های ناهمگن را می توان از هم جدا کرد. حذف یا تغییر این ثابت بر نحوه برخورد کرنل با مبدا بردارهای ویژگی تأثیر می گذارد.

RBF: الگوهای محلی را شامل می‌شود و به طور گسترده برای داده های غیر خطی استفاده می شود. کرنل RBF به دلیل توانایی آن در رسیدگی به مواردی که رابطه بین برچسب‌های کلاس و ویژگی‌ها غیرخطی است، محبوب است. این کرنل فضای ورودی را به یک فضای بینهایت بعدی نگاشت می کند.

کرنل سیگموئید: مشابه شبکه های عصبی، تابع تصمیم گیری بر اساس فعال سازی سیگموئید است. این کرنل می تواند روابط غیر خطی را مدل کند اما در مقایسه با هسته های RBF و چند جمله ای کمتر مورد استفاده قرار می گیرد.

5. مقایسه CatBoost، LightGBM و XGBoost:

CatBoost: به طور ویژه برای ویژگی های طبقه بندی شده بهینه شده است. این الگوریتم درخت‌های تصمیم‌گیری متقارن و boosting منظم را به کار می‌گیرد و بیش برازش را کاهش می‌دهد. همچنین نیازی به preprocessing ندارد. علاوه بر این موارد باعث بهینگی در مموری و زمان هم می‌شود.

LightGBM: از یادگیری مبتنی بر هیستوگرام و رشد مبتنی بر برگ استفاده می کند، که آن را برای مجموعه داده های بزرگ بسیار کارآمد می کند. One-hot encoding و ویژگی‌های طبقه‌بندی شده را نیز ساپوزت می‌کند.

XGBoost: بر اساس gradient boosting، با بهینه‌سازی‌هایی مانند regularization و sparsity awareness ، برای طیف وسیعی از مجموعه‌های داده قوی است. برعکس LightGBM، از رشد مبتنی بر عمق استفاده می‌کند که باعث افزایش زمان و حجم حافظه در قیاس با آن می‌شود ولی در حالت کلی سریع و مقیاس‌پذیر است.

هر سه مدل با استفاده از تقویت، منظم‌سازی و مدیریت پیشرفته متغیرهای طبقه‌بندی، درخت‌های تصمیم استاندارد را بهبود می‌بخشند. آنها بیش برازش را کاهش می دهند، دقت را افزایش می دهند و به طور موثر در مجموعه داده های بزرگ مقیاس‌پذیر می شوند.

6. راهکارهایی برای جلوگیری از برازش بیش از حد در درختان تصمیم:

هرس: با حذف شاخه هایی که اهمیت کمی دارند، اندازه درخت را کاهش می دهد.

حداکثر عمق: برای جلوگیری از یادگیری الگوهای خیلی خاص، عمق درخت را محدود می کند.

Min Samples Split/Leaf: تضمین می‌کند که تقسیم کردن فقط در صورت وجود حداقل تعداد نمونه اتفاق می‌افتد، و باعث کاهش بیش برازش و حساسیت به نویز می‌شود.

7. شاخص gini :

این شاخص ناخالصی یک مجموعه داده را اندازه گیری می کند. این احتمال را محاسبه می کند که یک عنصر به طور تصادفی انتخاب شده به اشتباه طبقه بندی شود اگر به طور تصادفی بر اساس توزیع برچسب ها در زیر مجموعه برچسب گذاری شده باشد.



که D دیتاست ماست، C تعداد کلاس‌های ماست و p احتمال یک کلاس در یک نود است.

شاخص جینی اغلب برای کارهای طبقه بندی که در آن سرعت یک موضوع مهم است ترجیح داده می شود، زیرا از نظر محاسباتی فشرده تر از آنتروپی است. همچنین تمایل به ایجاد درختان متعادل تری دارد.

آنتروپی:

آنتروپی میزان عدم قطعیت یا بی نظمی را در یک مجموعه داده اندازه گیری می کند. از تئوری اطلاعات مشتق شده است و غیرقابل پیش بینی بودن محتوای اطلاعات را کمی می کند. آنتروپی 0 به این معنی است که همه نمونه ها به یک کلاس تعلق دارند، در حالی که حداکثر مقدار زمانی رخ می دهد که کلاس ها به طور مساوی توزیع شوند.



در اینجا هم p مثل gini index معنای یکسانی دارد. نسبت به تغییرات در توزیع کلاس‌ها حساس‌تر است، و برای مجموعه‌های داده‌ای که درک دقیقی از توزیع کلاس ضروری است، مناسب است.

Information gain :



که A ویژگی است که مورد ارزیابی قرار گرفته است، Dv زیرمجموعه‌ای از D است که متغیر A مقدار v دارد.

تعریف: این مقیاس کاهش آنتروپی را پس از تقسیم یک مجموعه داده بر روی یک ویژگی اندازه گیری می کند. مقادیر بالاتر نشان دهنده ویژگی بهتر برای طبقه بندی است.

این مقیاس به ویژه در سناریوهایی که هدف به حداکثر رساندن اطلاعات به دست آمده از هر تقسیم است مفید است و آن را برای الگوریتم های درخت تصمیم مانند ID3 مناسب می کند.

Gain ratio :





Gain Ratio با در نظر گرفتن تعداد شاخه‌هایی که از تقسیم شدن بر روی یک مشخصه به وجود می‌آیند، information gain را تنظیم می‌کند. این مقیاس ویژگی هایی را که شاخه های کوچک زیادی تولید می کنند جریمه می کند.

8 .

Bias

Biasبه خطای معرفی شده توسط تقریب یک مسئله دنیای واقعی (که ممکن است پیچیده باشد) توسط یک مدل ساده اشاره دارد. بایاس زیاد می‌تواند باعث شود که الگوریتم روابط مربوطه بین ویژگی‌ها و خروجی‌های هدف را از دست بدهد و منجر به عدم تناسب شود.

یک مدل با bias بالا توجه کمی به داده های آموزشی می کند و مدل را بیش از حد ساده می کند و در نتیجه عملکرد ضعیفی در مجموعه داده های آموزشی و آزمایشی دارد.

واریانس:

واریانس به خطای ایجاد شده توسط حساسیت مدل به نوسانات کوچک در مجموعه آموزشی اشاره دارد. واریانس بالا می تواند باعث شود که یک الگوریتم به جای خروجی های مورد نظر، نویز تصادفی را در داده های آموزشی مدل کند و منجر به بیش برازش شود.

یک مدل با واریانس بالا توجه زیادی به داده‌های آموزشی، گرفتن نویز و نقاط پرت می‌کند که منجر به عملکرد خوب در مجموعه آموزشی اما عملکرد ضعیف در داده‌های دیده نشده می‌شود.

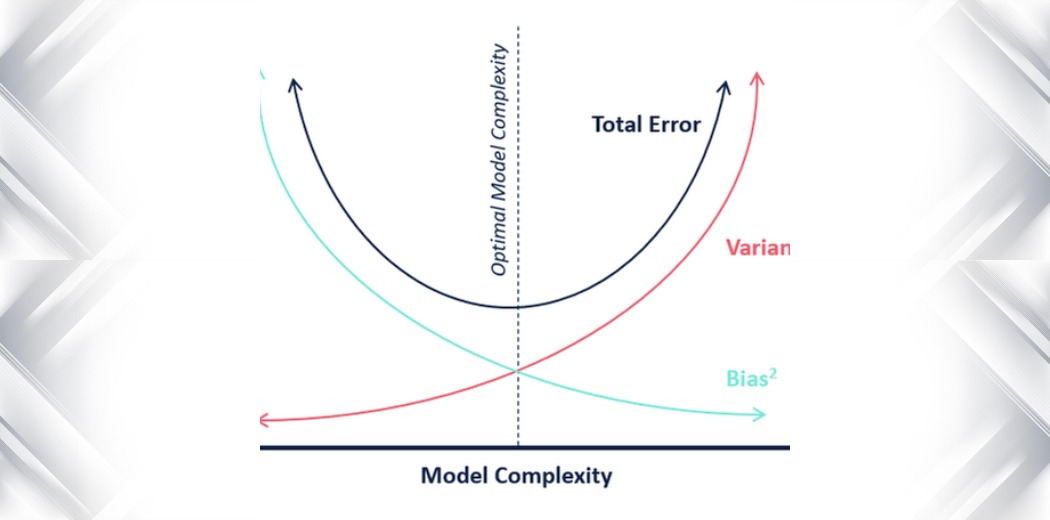
Bias-Variance tradeoff

مبادله را می توان به صورت زیر خلاصه کرد:

underfitting زمانی اتفاق می افتد که یک مدل خیلی ساده باشد (bias زیاد). نمی تواند الگوهای اساسی در داده ها را ضبط کند، که منجر به خطای آموزش زیاد و خطاهای تست می شود.

overfitting زمانی اتفاق می افتد که یک مدل بیش از حد پیچیده باشد (واریانس بالا). داده‌های آموزشی از جمله نویز را خیلی خوب یاد می‌گیرد که منجر به خطای آموزشی کم اما خطای تست بالا می‌شود.





9 .

cross validation K-fold یک تکنیک قوی برای ارزیابی مدل است که به ارزیابی میزان تعمیم یک مدل به یک مجموعه داده مستقل کمک می کند..

فرآیند cross validation K-Fold

تقسیم داده ها:

تقسیم مجموعه داده: مجموعه داده به طور تصادفی به زیرمجموعه‌ها با اندازه مساوی 'k' تقسیم می شود. به عنوان مثال، اگر شما 100 نمونه داشته باشید و k=5 را انتخاب کنید، هر فولد شامل 20 نمونه خواهد بود.

آموزش و اعتبارسنجی مدل:

Iterate Through Folds: برای هر یک از زیرمجموعه‌های 'k':

از یکی به عنوان مجموعه اعتبار سنجی و از k-1 تای باقی مانده به عنوان مجموعه آموزشی استفاده کنید.

مدل را در مجموعه آموزشی آموزش دهید و عملکرد آن را در مجموعه اعتبارسنجی ارزیابی کنید.

رکورد عملکرد: معیار عملکرد (به عنوان مثال، دقت، F1) را برای هر تکرار محاسبه و ثبت کنید.

جمع‌بندی نتایج:

پس از استفاده از تمام زیرمجموعه‌ها برای اعتبارسنجی، معیارهای عملکرد را در تمام تکرارهای 'k' میانگین کنید. این تخمین قابل اعتماد تری از عملکرد مدل ارائه می دهد.

ترتیب اجرا:

مقیاس بندی و تقسیم داده ها

ترتیب صحیح عملیات برای جلوگیری از data leakage و اطمینان از ارزیابی معتبر مدل بسیار مهم است:

مقیاس بندی داده ها:

قبل از تقسیم: کل مجموعه داده را قبل از تقسیم به fold‌ها مقیاس‌بندی کنید. این تضمین می کند که پارامترهای مقیاس بندی (میانگین، انحراف استاندارد، مقادیر حداقل حداکثر و غیره) از کل مجموعه داده مشتق شده اند.

در داخل هر فولد: با این حال، هنگام استفاده از تکنیک‌هایی مانند استانداردسازی (نرمال‌سازی z)، مهم است که مقیاس‌کننده را فقط روی داده‌های آموزشی در هر فولد قرار دهید و سپس تبدیل را در مجموعه‌های آموزشی و اعتبارسنجی اعمال کنید. این موضوع نشت اطلاعات مجموعه اعتبار سنجی به فرآیند آموزش جلوگیری می کند.

تقسیم K-Fold:

پس از مقیاس بندی مناسب داده ها، تقسیم k-fold را همانطور که در بالا توضیح داده شد انجام دهید.

تأثیر انتخاب مقادیر مختلف «k»

انتخاب "k" در k-fold cross validation بر bias-variance tradeoff در طول ارزیابی مدل تاثیر می گذارد:

مقادیر کوچک «k» (به عنوان مثال، k=2 یا k=3):

واریانس بالاتر: با fold‌های کمتر، هر مجموعه آموزشی بزرگ‌تر است و مدل ممکن است به نمونه‌های خاص در داده‌های آموزشی حساس‌تر باشد. این می تواند منجر به واریانس بالاتر در برآوردهای عملکرد شود.

bias پایین: مدل بر روی بخش بزرگتری از داده ها آموزش داده شده است، که ممکن است منجر به یادگیری بهتر الگوهای اساسی شود.

مقادیر بزرگ «k» (به عنوان مثال، k=10 یا k=20):

واریانس پایین: fold های بیشتر به این معنی است که هر مجموعه آموزشی کوچکتر است، که می تواند حساسیت مدل به نویز در داده ها را کاهش دهد. این به طور کلی منجر به برآوردهای عملکرد پایدارتر و قابل اعتمادتر می شود.

bias بالاتر: از آنجایی که هر مجموعه آموزشی کوچکتر است، مدل ممکن است به طور موثر از داده ها یاد نگیرد و به طور بالقوه منجر به سوگیری بالاتر شود.

انتخاب k:

یک روش معمول این است که k=5 یا k=10 را انتخاب کنید، که معمولاً تعادل خوبی بین bias و واریانس ایجاد می‌کند.

10 .

تعریف:

K-Fold Cross-Validation: این تکنیک شامل تقسیم مجموعه داده به k زیرمجموعه‌ی هم اندازه است. این مدل بر روی k-1 زیرمجموعه آموزش داده می شود و روی زیرمججموعه‌های باقی مانده آزمایش می شود. این فرآیند k بار تکرار می شود و هر fold یک بار به عنوان مجموعه تست عمل می کند.

K-Fold Cross-Validation Stratified :مشابه K-Fold است، اما تضمین می کند که هر فولد تقریباً همان توزیع کلاس های هدف را با کل مجموعه داده دارد. این موضوع به ویژه برای مجموعه داده های نامتعادل مهم است.

2. رسیدگی به عدم تعادل کلاس:

K-Fold: توزیع کلاس ها را در نظر نمی گیرد. اگر مجموعه داده نامتعادل باشد، ممکن است برخی از فولدها نمونه‌های بسیار کمی از یک کلاس اقلیت داشته باشند یا هیچ نمونه‌ای نداشته باشند، که منجر به تخمین‌های عملکرد مغرضانه می‌شود.

Stratified K-Fold: نسبت هر کلاس را در هر فولد حفظ می کند. این تضمین می‌کند که همه کلاس‌ها به اندازه کافی در مجموعه‌های آموزشی و آزمایشی نمایش داده می‌شوند و تخمین مطمئن‌تری از عملکرد مدل ارائه می‌دهند.

3. معیارهای عملکرد:

K-Fold: ممکن است معیارهای عملکرد گمراه‌کننده‌ای ایجاد کند اگر مدل بر روی فولدهایی ارزیابی شود که توزیع کلی کلاس را نشان نمی‌دهند.

K-Fold : Stratifiedمعیارهای عملکرد سازگارتر و قابل اعتمادتری را به خصوص برای مشکلات طبقه بندی با کلاس های نامتعادل ارائه می دهد.

چه زمانی باید از اعتبارسنجی متقاطع K-Fold طبقه بندی شده استفاده کرد؟

مجموعه داده های نامتعادل: زمانی که متغیر هدف دارای عدم تعادل طبقه قابل توجهی است (به عنوان مثال، تشخیص تقلب، تشخیص بیماری)، K-Fold طبقه بندی شده یاStratified K-Foldترجیح داده می شود تا اطمینان حاصل شود که هر دسته دارای توزیع نماینده ای از طبقات است.

مشکلات طبقه بندی: به ویژه برای کارهای طبقه بندی که حفظ توزیع کلاس برای ارزیابی عملکرد مدل بسیار مهم است مفید است.

مجموعه داده‌های کوچک: وقتی مجموعه داده کوچک است، K-Fold طبقه‌بندی شده به اطمینان از اینکه همه کلاس‌ها به اندازه کافی در مجموعه‌های آموزشی و اعتبارسنجی نمایش داده می‌شوند، کمک می‌کند.

11 .

1. دقت:

تعریف: دقت نسبت پیش‌بینی‌های مثبت واقعی به کل مثبت‌های پیش‌بینی‌شده (مثبت واقعی + مثبت کاذب) است.

فرمول:



تفسیر: دقت به این سوال پاسخ می دهد: "از همه نمونه های پیش بینی شده به عنوان مثبت، چند مورد واقعا مثبت بودند؟" دقت بالا نشان دهنده نرخ مثبت کاذب کم است.

2. recall (حساسیت یا نرخ مثبت واقعی):

تعریف: recall نسبت پیش بینی های مثبت واقعی به کل مثبت های واقعی (مثبت های واقعی + منفی های کاذب) است.

فرمول:



تفسیر: recall به این سوال پاسخ می دهد: "از بین تمام موارد مثبت واقعی، چند مورد به درستی مثبت پیش بینی شده است؟" recall بالا نشان دهنده نرخ منفی کاذب پایین است.

3. F1:

تعریف: F1 میانگین هارمونیک دقت و recall است. این یک معیار واحد فراهم می کند که دقت و یادآوری را متعادل می کند.

فرمول:



تفسیر: امتیاز F1 به ویژه زمانی مفید است که به تعادل بین دقت و یادآوری نیاز دارید و زمانی که توزیع کلاس نامتعادل است.

Recall and precision tradeoff

توضیح: دقت و recallاغلب رابطه معکوس دارند. افزایش دقت معمولاً یادآوری را کاهش می دهد و بالعکس.

به عنوان مثال:

اگر مدلی به گونه‌ای تنظیم شده باشد که بسیار دقیق باشد (یعنی فقط زمانی که خیلی مطمئن باشد موارد مثبت را پیش‌بینی می‌کند)، ممکن است بسیاری از موارد مثبت واقعی را از دست بدهد و در نتیجه recall کمتری داشته باشد.

برعکس، اگر مدلی به گونه‌ای تنظیم شود که تا حد ممکن موارد مثبت را ثبت کند (recallبالا)، ممکن است شامل بسیاری از موارد مثبت کاذب نیز باشد که منجر به دقت کمتری می‌شود.

انتخاب متریک مناسب: انتخاب بین دقت و recall به زمینه خاص مشکل بستگی دارد:

نیاز به دقت بالا: در سناریوهایی که مثبت کاذب ارزشمند هستند (به عنوان مثال، تشخیص هرزنامه)، دقت بالا در اولویت است.

نیاز به recall زیاد: در سناریوهایی که از دست دادن یک نمونه مثبت حیاتی است (به عنوان مثال غربالگری بیماری)، recall بالا در اولویت است.

F1 به عنوان یک مصالحه: F1 زمانی که می‌خواهید دقت و recall را متعادل کنید، به عنوان یک مصالحه مفید عمل می‌کند، به خصوص در شرایطی که یک معیار برای ارزیابی عملکرد مدل کافی نیست.

12 .

1. تعریف:

ROC یک نمایش گرافیکی است که توانایی تشخیصی یک طبقه‌بندی کننده باینری را به عنوان آستانه تمایز آن نشان می‌دهد. این منحنی نرخ مثبت واقعی (TPR) را در برابر نرخ مثبت کاذب (FPR) در تنظیمات آستانه مختلف ترسیم می کند.

2. اجزاء:

نرخ مثبت واقعی (TPR) (recall)

نرخ مثبت کاذب (FPR)

زمان استفاده از منحنی ROC

مشکلات طبقه‌بندی باینری: منحنی ROC معمولاً برای ارزیابی مدل‌هایی استفاده می‌شود که نتایج باینری را پیش‌بینی می‌کنند (به عنوان مثال، بله/خیر، مثبت/منفی).

مجموعه داده های نامتعادل: این منحنی به ویژه در هنگام برخورد با مجموعه داده های نامتعادل مفید است، زیرا به جای تمرکز صرف بر روی دقت، بینشی در مورد عملکرد مدل در آستانه های مختلف ارائه می دهد.

مقایسه مدل ها: منحنی ROC امکان مقایسه آسان مدل های طبقه بندی مختلف را بر اساس عملکرد آنها در آستانه های مختلف فراهم می کند.

تفسیر ناحیه زیر منحنی (AUC)

1. تعریف AUC:

ناحیه زیر منحنی ROC (AUC) عملکرد کلی طبقه بندی کننده را کمی می کند. این احتمال را نشان می دهد که یک نمونه مثبت به طور تصادفی انتخاب شده بالاتر از یک نمونه منفی تصادفی انتخاب شده است.

2. تفسیر امتیاز AUC:

AUC = 0.5: عملکرد مدل بهتر از حدس زدن تصادفی نیست. این نشان دهنده یک مدل ضعیف است.

AUC < 0.5: مدل بدتر از حدس زدن تصادفی است، که نشان می دهد طبقه بندی کننده نمونه ها را به اشتباه طبقه بندی می کند.

AUC = 1.0: مدل کاملاً تمام موارد مثبت و منفی را طبقه بندی می کند که نشان دهنده عملکرد عالی است.

AUC بین 0.5 و 1.0: مقادیر بالاتر عملکرد بهتر مدل را نشان می دهد. به طور کلی، AUC از:

0.6 - 0.7: عملکرد قابل قبول

0.7 - 0.8: عملکرد خوب

0.8 - 0.9: عملکرد بسیار خوب

0.9 - 1.0: عملکرد عالی

13 .

نورم L1 در Lasso regression :



نورم L1 باعث پراکندگی در ضرایب مدل می‌شود. این نورم می تواند برخی از ضرایب را دقیقاً به صفر برساند و به طور موثر انتخاب متغیر را انجام دهد. این زمانی مفید است که ما شک کنیم که بسیاری از ویژگی‌ها نامربوط هستند.

Lasso regression زمانی مفید است که ما یک مدل ساده‌تر با پیش‌بینی‌کننده‌های کمتر می‌خواهیم و آن را قابل تفسیر می‌کند.

نورم L2 در ridge regression :



نورم L2 ضرایب را به سمت صفر کوچک می کند اما آنها را صفر نمی کند. این بدان معنی است که همه پیش بینی کننده ها در مدل باقی می مانند، اما تاثیر آنها کاهش می یابد.

Ridge regression زمانی مفید است که پیش‌بینی‌کننده‌های زیادی داریم که همبسته هستند یا زمانی که می‌خواهیم با نگه داشتن همه پیش‌بینی‌کننده‌ها اما کنترل تاثیر آنها، از بیش‌برازش جلوگیری کنیم.

تاثیر بر ضرایب مدل

L1 Norm: برخی از ضرایب می توانند دقیقاً صفر شوند. این منجر به مدل‌های ساده‌تر می‌شود که تفسیر آنها آسان‌تر است و ممکن است در موقعیت‌هایی که انتخاب ویژگی بسیار مهم است کمک کند.

L2 Norm: به طور کلی ضرایب کوچکتری تولید می کند اما همه پیش بینی کننده ها را در مدل حفظ می کند. این می تواند به ویژه در موقعیت های چند خطی که پیش بینی کننده ها همبستگی بالایی دارند مفید باشد، زیرا به تثبیت تخمین ها کمک می کند.

Elastic net regularization



در واقع این نورم، ترکیب L1 و L2 است.

Elastic Net ویژگی انتخاب ویژگی Lasso را حفظ می کند و در عین حال ثبات رگرسیون Ridge را نیز در خود دارد. به ویژه هنگامی که چندین ویژگی همبسته وجود دارد، مؤثر است، زیرا می تواند گروه هایی از پیش بینی کننده های همبسته را انتخاب کند.

سناریوهایی که در آن elastic net مفید است

داده های با ابعاد بالا: زمانی که تعداد پیش بینی کننده ها از تعداد مشاهدات بیشتر شود، Elastic Net می تواند با انتخاب ویژگی های مربوطه و در عین حال پایداری مدل، عملکرد خوبی داشته باشد.

ویژگی‌های همبسته: در سناریوهایی که پیش‌بینی‌کننده‌ها همبستگی بالایی دارند، Elastic Net می‌تواند یک یا چند ویژگی را از گروهی از ویژگی‌های همبسته انتخاب کند، و از محدودیت‌های Lasso اجتناب کند، که ممکن است خودسرانه یکی را انتخاب کند و دیگران را نادیده بگیرد.

انتخاب ویژگی با پایداری: هنگامی که هم انتخاب متغیر و هم منظم سازی را می خواهید، Elastic Net یک رویکرد متعادل ارائه می دهد و آن را به یک انتخاب همه کاره در بسیاری از کاربردهای عملی تبدیل می کند.

***به دلیل سخت بودن تایپ فرمول‌ها در ورد، فرمول‌ها از داخل کتاب گرد‌اوری شده است.***