

نام و نام خانوادگی: حسنا اویارحسینی	تمرین دوم داده کاوی
شماره دانشجویی: ۹۸۲۳۰۱۰	تاریخ: اردیبهشت - ۱۴۰۲

سوال (۱)

چون ممکن است توزیع داده های دیتاست جدید که میخواهیم برای هرس کردن درخت استفاده کنیم متفاوت از توزیع داده های آموزشی باشد و باعث شود تا هرس به خوبی و بر اساس داده هایی که درخت را ساخته اند صورت نگیرند. در واقع ممکن است درخت تولید شده برای داده های جدید مناسب نباشد و در نتیجه هرس کردن بی فایده خواهد بود چون میخواهد در واقع درخت را برای داده ها و توزیع جدید تغییر دهد نه اینکه بیش برزش مدل را بر روی داده های اولیه کم کند.

اگر از یک مجموعه داده جداگانه برای هرس درخت استفاده شود، فرآیند هرس ممکن است به سمت ویژگی های آن مجموعه داده خاص سوگیری داشته باشد، که ممکن است نماینده جمعیت واقعی یا داده هایی نباشد که مدل روی آن اعمال می شود. این می تواند منجر به یک درخت تصمیم شود که برای ویژگی های خاص مجموعه داده های هرس بهینه شده است اما ممکن است به خوبی به داده های جدید و دیده نشده تعمیم پیدا نکند. در نتیجه از یک مجموعه داده اعتبارسنجی گرفته شده از دادگان اولیه برای هرس استفاده میکنیم.

سوال (۲)

سوال (۲)

استاد و دیگران قدری لیست می‌کنیم

قد	160	162	165	170	172
تعداد	+	+	-	-	+
		163.5		171	

میزان بالقوه:

$$H(D)_{h=163.5} = \frac{2}{5} H(2,0) + \frac{3}{5} H(2,1) = 0.551$$

$$= -\frac{2}{5} \log \frac{2}{5} - \frac{3}{5} \log \frac{3}{5}$$

$$H(D)_{h=171} = \frac{4}{5} H(2,2) + \frac{1}{5} H(1,0) = 0.8$$

چون $h=163.5$ آنتروپی کمتر می‌دهد و آن را انتخاب می‌کنیم

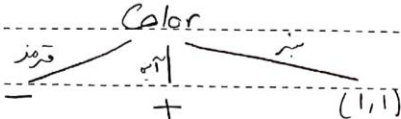
$$IG(\text{height}) = H(D)_{h=163.5} - H(D) = \left(-\frac{2}{5} \log \frac{2}{5} - \frac{3}{5} \log \frac{3}{5} \right) - 0.551 = 0.12$$

$$IG(\text{color}) = H(D)_{\text{color}} - H(D) = 0.971 - \left(\frac{1}{5} H(1,0) + \frac{2}{5} H(2,0) + \frac{2}{5} H(1,1) \right) = 0.571$$

$$IG(\text{weight}) = H(D)_{\text{weight}} - H(D) = 0.971 - \left(\frac{3}{5} H(1,2) + \frac{2}{5} H(1,1) \right) = 0.02$$

و دیگران هم می‌توانند IG را به عنوان اولین ویژگی در رستیم در جهت انتخاب

می‌کنیم و ضوابط داریم



در ساختن درخت طبقه‌بندی، از آنجا که داده‌ها دارای ویژگی‌های کیفی و کمی هستند، باید از روش‌های مختلف برای تقسیم‌بندی استفاده کرد. در اینجا، از روش IG برای انتخاب ویژگی‌ها استفاده می‌کنیم.

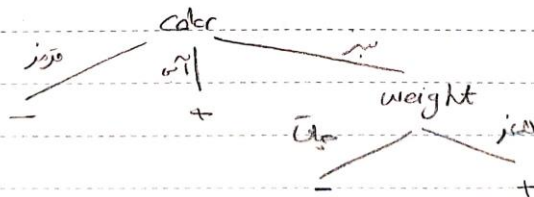
$$H(D) = H(1,1) = 1$$

color = G

$$H_{weight}(D) = \frac{1}{2} H(1,1) + \frac{1}{2} H(1,0) = 0 \rightarrow IG(weight) = 1 - 0 = 1$$

$$H_{height}(D) = H(1,1) = 1 \rightarrow IG(height) = 1 - 1 = 0$$

h = 163.5



و چون درخت طبقه‌بندی، از آنجا که داده‌ها دارای ویژگی‌های کیفی و کمی هستند، باید از روش‌های مختلف برای تقسیم‌بندی استفاده کرد.

سوال (۳)

$$P(\text{pass} | \text{GPA} = A, \text{study}, \text{presence}) \quad (\text{سوال ۳})$$

$$= \frac{P(X | \text{pass}) P(\text{pass})}{P(X)} = \frac{P(\text{GPA} = A | \text{pass}) P(\text{study} | \text{pass}) P(\text{presence} | \text{pass})}{P(X)} \times P(\text{pass})$$

$$= \left(\frac{2}{4} \times \frac{3}{4} \times \frac{1}{4} \right) \times \frac{4}{6} = \frac{1}{6}$$

$$P(\overline{\text{pass}} | \text{GPA} = A, \text{study}, \text{presence})$$

$$= \left[P(\text{GPA} = A | \overline{\text{pass}}) P(\text{study} | \overline{\text{pass}}) P(\text{presence} | \overline{\text{pass}}) \right] P(\overline{\text{pass}})$$

$$= \left[\frac{0}{2} + \frac{1}{1 \times 3} \times \frac{0}{2} + \frac{1}{2} \times \frac{2}{2} \right] \times \frac{2}{6} = \frac{1}{5} \times \frac{1}{4} \times \frac{1}{3} = \frac{1}{60}$$

$$P(\overline{\text{pass}} | X) < P(\text{pass} | X) \Rightarrow \text{label} = \text{pass}$$

سوال (۴)

۱. عدم تعادل کلاس: در مجموعه داده هایی که کلاس ها نامتعادل هستند، به این معنی که یک کلاس نمونه های بیشتری نسبت به سایرین دارد، دقت می تواند گمراه کننده باشد. مدلی که به سادگی کلاس اکثریت را برای همه موارد پیش بینی می کند، می تواند به دلیل توزیع نامتعادل به دقت بالایی دست یابد، حتی اگر نتواند کلاس اقلیت را به خوبی تشخیص دهد. در چنین مواردی، معیارهایی مانند دقت، یادآوری، امتیاز F_1 یا ناحیه زیر منحنی مشخصه عملکرد گیرنده (AUC-ROC) می توانند بینش بهتری ارائه دهند.

۲. کاربردهای حساس به هزینه: در برخی برنامه ها، خطاهای تشخیص اشتباه ممکن است هزینه ها یا پیامدهای مختلفی داشته باشند. به عنوان مثال، در تشخیص پزشکی، خطای منفی غلط (تشخیص بیمار به عنوان سالم) ممکن است از خطای مثبت غلط (تشخیص سالم به عنوان بیمار) شدیدتر باشد. در چنین مواردی، دقت به تنهایی هزینه های مختلف انواع خطاها را در نظر نمی گیرد. استفاده از معیارهایی مانند دقت (precision)، بازخوانی (recall) یا معیارهای ارزیابی مبتنی بر هزینه می تواند مناسب تر باشد.

۳. مسائل ترتیبی یا چند طبقه ای: دقت فرض می کند که همه طبقه بندی های اشتباه به یک اندازه نادرست هستند. با این حال، در مسائل طبقه بندی ترتیبی یا چند طبقه ای، طبقه بندی اشتباه یک نمونه به عنوان یک کلاس شبیه ممکن است کمتر از طبقه بندی اشتباه آن به عنوان یک کلاس کمتر شبیه، اشتباه باشد. معیارهای ارزیابی مانند میانگین مربعات خطا، دقت وزنی، یا معیارهای خاص ترتیبی مانند آماره کاپا می توانند ارزیابی های دقیق تری ارائه دهند.

۴. اطمینان پیش بینی: دقت، اطمینان یا سطح اطمینان پیش بینی ها را در نظر نمی گیرد. مدلی که برای پیش بینی های مطمئن بسیار دقیق است، اما برای موارد مرزی نامشخص است، ممکن است در کاربردهای خاص مطلوب نباشد. معیارهایی مانند کالیبراسیون، یا امتیاز بریر، تخمین های اطمینان یا احتمال پیش بینی ها را در نظر می گیرند.

سوال (۵)

هنگام استفاده از روش ۱۰-fold cross-validation برای انتخاب پارامتر α در مدل، می توان این مراحل را برای انتخاب مدل نهایی و برآورد خطا دنبال کرد:

انتخاب پارامتر: روش ۱۰ fold cross-validation را با تقسیم داده‌هایتان به ۱۰ زیرمجموعه یا fold انجام دهید. مدل خود را در ۹ fold آموزش داده و در هر مرحله مقدار پارامتر α را متغیر کنید. عملکرد مدل را با استفاده از معیار ارزیابی مناسب (مثل دقت، دقت، بازخوانی) روی fold یا مجموعه اعتبارسنجی باقیمانده ارزیابی کنید.

ارزیابی پارامتر: میانگین مقدار معیار عملکرد را در تمامی ۱۰ fold برای هر مقدار α محاسبه کنید. این کار به شما تخمینی از عملکرد مدل برای مقادیر مختلف α می‌دهد. مقدار α را انتخاب کنید که بهترین عملکرد میانگین را در بین fold ها داشته باشد.

برآورد خطا: برای برآورد خطای مدل نهایی، می‌توانید از روش‌هایی مانند holdout validation یا nested cross-validation, boosting, Estimating Confidence Intervals استفاده کنید.

- **Holdout Validation:** مجموعه داده‌های خود را به سه بخش تقسیم کنید: مجموعه آموزشی، مجموعه اعتبارسنجی و مجموعه تست. مدل نهایی خود را در مجموعه آموزشی آموزش دهید، با استفاده از مجموعه اعتبارسنجی، هایپرپارامترها (در صورت وجود) را تنظیم کنید و عملکرد مدل را در مجموعه آزمایشی ارزیابی کنید. مجموعه تست تخمین بی طرفانه‌ای از عملکرد مدل شما ارائه می‌دهد.
- **Nested Cross-Validation:** برای به دست آوردن یک تخمین بی طرفانه از عملکرد مدل خود، اعتبارسنجی متقابل تودرتو انجام دهید. حلقه بیرونی اعتبارسنجی متقاطع داده‌ها را به چند برابر تقسیم می‌کند، در حالی که حلقه داخلی برای تنظیم هایپرپارامتر استفاده می‌شود. این رویکرد به شما امکان می‌دهد عملکرد مدل را روی داده‌های دیده نشده ارزیابی کنید و در عین حال ابرپارامترها را نیز بهینه کنید.
- همچنین می‌توان از تخمین فواصل اطمینان که از null hypothesis, t-test برای پیدا کردن اینکه آیا دو مدل آیا از نظر آککاری تفاوت معنا داری دارند و کدام بهتر است استفاده کرد.

سوال ۶)

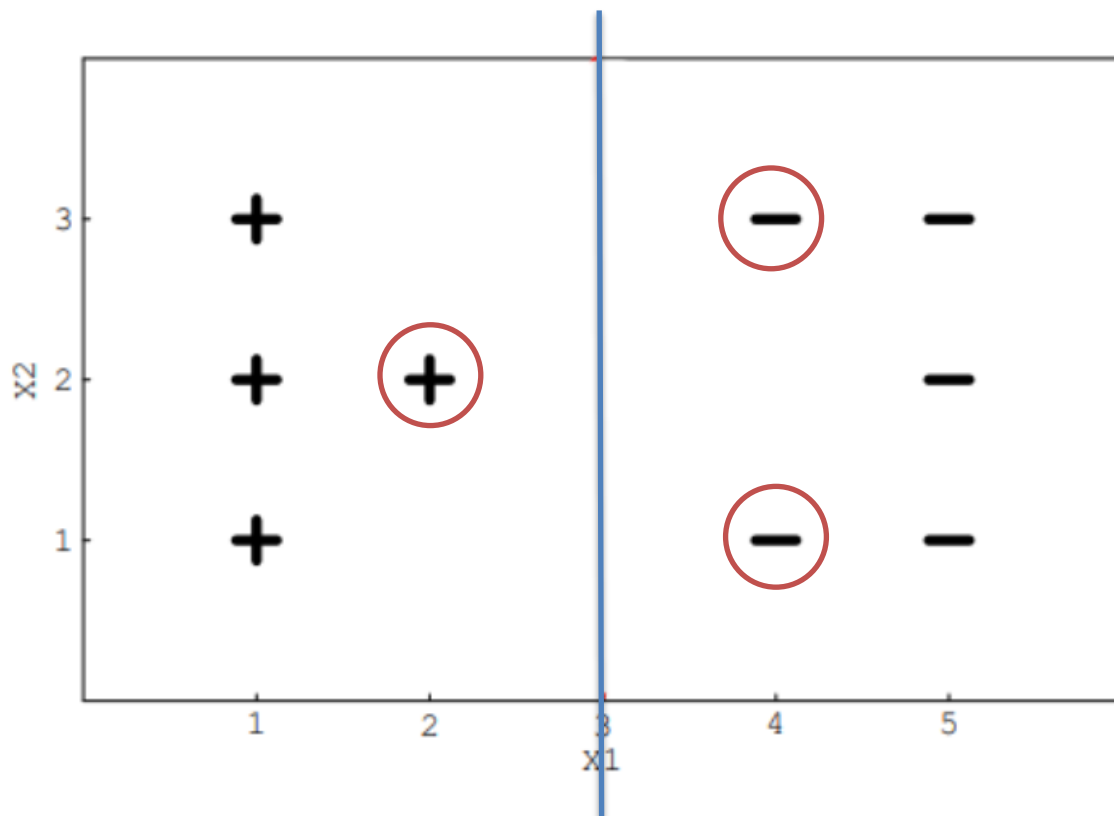
الف) خیر، boosting نسبت به بیش برآزش مقاوم است. خطای تست ممکن است حتی پس از صفر شدن خطای آموزش کاهش یابد. در این روش‌ها هدف تقویت، بهبود مکرر عملکرد گروه طبقه‌بندی‌کننده با تمرکز بر نمونه‌های طبقه‌بندی‌شده اشتباه است. الگوریتم‌های تقویت‌کننده، مانند AdaBoost، با اختصاص وزن‌های بالاتر به نمونه‌های طبقه‌بندی‌شده اشتباه در هر تکرار کار می‌کنند، در نتیجه به طبقه‌بندی‌کننده‌های ضعیف بعدی اجازه می‌دهند تا روی آن نمونه‌ها تمرکز بیشتری داشته باشند. تکرارها تا زمانی ادامه می‌یابند که تعداد از پیش تعریف‌شده‌ای از طبقه‌بندی‌کننده‌های ضعیف آموزش داده شوند یا به حداکثر تعداد تکرار برسد. هدف الگوریتم‌های تقویت معمولاً به حداقل رساندن میزان خطای کلی در داده‌های آموزشی یا بهینه‌سازی یک تابع ضرر خاص است.

در حالی که ممکن است طبقه‌بندی‌کننده ترکیبی به نرخ خطای \cdot در داده‌های آموزشی در طول فرآیند تقویت دست یابد، این لزوماً به این معنی نیست که مدل یک نمایش کامل از توزیع زیربنایی را یاد گرفته است یا اینکه به خوبی به داده‌های دیده نشده تعمیم می‌دهد. هدف اصلی تقویت، بهبود عملکرد تعمیم مدل، کاهش سوگیری و واریانس است. بنابراین، آموزش معمولاً تا زمانی ادامه می‌یابد که یک معیار توقف برآورده شود، مانند رسیدن به حداکثر تعداد تکرار، یک آستانه خطای از پیش تعریف‌شده، یا زمانی که عملکرد مدل در یک مجموعه اعتبارسنجی جداگانه شروع به بدتر شدن کند.

ب) بله، در این حالت، $\alpha t = +\infty$ ، و وزن همه مثال‌ها \cdot است. از طرف دیگر، طبقه‌بندی‌کننده ضعیف فعلی روی داده‌های تمرین وزن‌دار عالی است، بنابراین در مجموعه داده‌های اصلی نیز عالی است. نیازی به ترکیب این طبقه بندی با سایر طبقه بندی کننده ها نیست.

سوال (۷)

الف) این خط به نحوی می باشد که داده های دو کلاس دو طرف مرز قرار میگیرد و همچنین مجموع اندازه حاشیه مرز از نزدیکترین داده هر دسته بیشترین مقدار ممکن خواهد بود:



ب) سه داده قرمز به گونه ای قرار گرفته اند که حذف هر یک از آنها باعث ایجاد سستی در محدودیت ها می شود. بنابراین مرز تصمیم گیری کاملاً تغییر می کند. چون میتوانیم مرزی پیدا کنیم که حاشیه مرز از نزدیکترین داده ها بیشتر شود.

سوال ۸)

الف) غلط، در حالی که Naive Bayes استقلال ویژگی را فرض می کند، هنوز هم می تواند وابستگی بین متغیرها را به طور غیر مستقیم از طریق متغیر کلاس دریافت کند. Naive Bayes با تخمین احتمال شرطی کلاس با توجه به مقادیر ویژگی، روابط بین ویژگی ها و کلاس را در نظر می گیرد و می تواند بر اساس آن روابط پیش بینی کند.

ب) صحیح، هر چه درخت پر تر شود تمایل آن به بیش برآزش بیشتر میشود و می تواند نمونه های آموزشی، از جمله نویز را به خاطر بسپارد و در نتیجه مدلی به دست می آید که به خوبی به داده های دیده نشده تعمیم نمی یابد و توانایی بیشتری در گرفتن کوچکترین و خاص ترین الگوها، از جمله نویز، پیدا می کند.

ج) غلط، چون اگر $K=1$ باشد با پیدا کردن نزدیک ترین داده به هر داده جدید برچسب زنی را انجام میدهیم و اگر نویز در نزدیکی داده باشد برچسب زنی اشتباه میشود اما اگر K بیشتر باشد قدرت نویز در برچسب زنی کمتر می شود چون بقیه داده ها هم برچسب زنی را تحت تاثیر قرار می دهند.

سوال ۹)

Recall یا حساسیت معیاری است که نسبت نمونه های مثبت واقعی را که به درستی توسط یک مدل از بین تمام نمونه های مثبت موجود در داده ها شناسایی شده اند، اندازه گیری می کند. به عبارت دیگر، توانایی یک مدل را برای تشخیص صحیح تمام موارد مثبت اندازه گیری می کند.

استفاده از یادآوری به عنوان یک معیار در چنین سناریوهایی مناسب است زیرا بر شناسایی موارد صحیح مثبت واقعی تمرکز می کند، حتی اگر به قیمت افزایش مثبت کاذب باشد. به حداکثر رساندن یادآوری به حداقل رساندن تعداد منفی های کاذب کمک می کند و اطمینان حاصل می کند که همه موارد مثبت شناسایی می شوند.

سوال ۱۰)

شاخص جینی یا ناخالصی احتمال طبقه بندی اشتباه یک نمونه تصادفی را هنگام انتخاب تصادفی اندازه گیری می کند. هرچه شاخص جینی کمتر باشد، احتمال طبقه بندی اشتباه کمتر است. ناخالصی جینی را می توان با

جمع کردن احتمال‌های p_i برای یک عنصر که برچسب i برای آن انتخاب شده در احتمال دسته‌بندی اشتباه آن که برابر است با $\sum_{k \neq i} p_k = 1 - p_i$ بدست آورد.

کمترین مقدار ممکن برای این متریک برابر صفر است که در این حالت تمام المان‌های در یک مجموعه به یک کلاس تعلق دارند.

برای اینکه مقدار ناخالصی جینی را برای یک مجموعه از المان‌ها با J کلاس که p_i نسبتی از داده‌ها باشد که با برچسب i در این مجموعه برچسب زده شده‌اند محاسبه کنیم از رابطه زیر استفاده می‌کنیم:

$$I_G(p) = \sum_{i=1}^J \left(p_i \sum_{k \neq i} p_k \right) = \sum_{i=1}^J p_i (1 - p_i) = \sum_{i=1}^J (p_i - p_i^2) = \sum_{i=1}^J p_i - \sum_{i=1}^J p_i^2 = 1 - \sum_{i=1}^J p_i^2$$

درنهایت ویژگی با شاخص Gini پایین تر برای تقسیم در درخت انتخاب می شود.

کد محاسبه شاخص جینی:

```
unique, counts = np.unique(labels, return_counts=True) # unique classes & number of data in label
p = counts / labels.shape[0] # probability of each class
gini = 1 - np.sum(p**2) # calculate gini for each class using formul
```

سوال (۱۱)

(الف)

• تفاوت‌ها:

۱. تعداد متغیرهای مستقل: در رگرسیون خطی ساده، تنها یک متغیر مستقل وجود دارد، در حالی که در رگرسیون خطی چندگانه، دو یا چند متغیر مستقل وجود دارد. رگرسیون خطی ساده بر مدل‌سازی رابطه بین یک متغیر وابسته و یک متغیر پیش‌بینی‌کننده تک تمرکز دارد، در حالی که رگرسیون خطی چندگانه امکان تحلیل رابطه بین متغیر وابسته و پیش‌بینی‌کننده‌های متعدد را فراهم می‌کند.

۲. پیچیدگی: رگرسیون خطی ساده برای تفسیر و تجسم ساده‌تر است زیرا شامل یک پیش‌بین واحد است. از سوی دیگر، رگرسیون خطی چندگانه پیچیده‌تر است زیرا چندین پیش‌بینی‌کننده را به طور همزمان در نظر می‌گیرد. برای درک رابطه بین متغیر وابسته و هر متغیر مستقل، نیاز به تکنیک‌های تحلیل اضافی دارد.

۳. نمایش مدل: در رگرسیون خطی ساده، رابطه بین متغیر وابسته و متغیر مستقل با یک خط مستقیم نشان داده می‌شود. در رگرسیون خطی چندگانه، این رابطه توسط یک ابر صفحه در فضایی با ابعاد بالاتر نشان داده می‌شود که تجسم آن فراتر از سه بعد دشوار است.

۴. تفسیر: رگرسیون خطی ساده امکان تفسیر مستقیم رابطه بین متغیر مستقل و متغیر وابسته را فراهم می کند. شیب در رگرسیون خطی ساده نشان دهنده تغییر در متغیر وابسته برای تغییر واحد در متغیر مستقل است. در رگرسیون خطی چندگانه، تفسیر ضرایب پیچیده تر می شود زیرا آنها شیب در متغیر وابسته ارتباط با هر متغیر مستقل را نشان می دهند، در حالی که سایر متغیرها را ثابت نگه می دارند.

• شباهت ها:

۱. خطی بودن: هر دو رگرسیون خطی ساده و رگرسیون خطی چندگانه یک رابطه خطی بین متغیرهای مستقل و متغیر وابسته را فرض می کنند. هدف آنها مدل سازی این رابطه خطی با تخمین ضرایبی است که نشان دهنده شیب یا تأثیر متغیرهای مستقل بر متغیر وابسته است.

۲. مفروضات: هر دو تکنیک رگرسیون مفروضات مشابهی دارند، مانند خطی بودن، استقلال خطاها، واریانس ثابت خطاها (همسان سازی) و خطاهای معمولی توزیع شده. این مفروضات باید بررسی شوند تا از اعتبار مدل های رگرسیونی اطمینان حاصل شود.

۳. حداقل مربعات معمولی (OLS یا Ordinary Least Squares): هر دو رگرسیون خطی ساده و رگرسیون خطی چندگانه معمولاً از روش حداقل مربعات معمولی (OLS) برای تخمین ضرایب استفاده می کنند. OLS مجموع اختلاف مجذور بین مقادیر مشاهده شده و پیش بینی شده را به حداقل می رساند.

(ب)

رگرسیون ridge , Lasso برخی از تکنیک های ساده برای کاهش پیچیدگی مدل و جلوگیری از برازش بیش از حد است که ممکن است از رگرسیون خطی ساده ناشی شود.

• رگرسیون ridge: در رگرسیون پشته، تابع هزینه با افزودن جریمه ای معادل مجذور بزرگی ضرایب تغییر می کند.

$$\sum_{i=1}^M (y_i - \hat{y}_i)^2 = \sum_{i=1}^M \left(y_i - \sum_{j=0}^p w_j \times x_{ij} \right)^2 + \lambda \sum_{j=0}^p w_j^2$$

بنابراین رگرسیون ridge بر روی ضرایب (w) محدودیت ایجاد می کند. عبارت جریمه (لامبدا) ضرایب را منظم می کند به طوری که اگر ضرایب مقادیر زیادی بگیرند، تابع بهینه سازی جریمه می شود. بنابراین، رگرسیون ضرایب را کوچک می کند و به کاهش پیچیدگی و چند خطی بودن مدل کمک می کند. هر چه محدودیت (λ کم) روی ویژگی ها را کاهش دهیم، مدل بیشتر شبیه مدل رگرسیون خطی خواهد بود.

- رگرسیون Lasso: تابع هزینه برای رگرسیون Lasso را می توان به صورت زیر نوشت:

$$\sum_{i=1}^M (y_i - \hat{y}_i)^2 = \sum_{i=1}^M \left(y_i - \sum_{j=0}^p w_j \times x_{ij} \right)^2 + \lambda \sum_{j=0}^p |w_j|$$

تنها تفاوت با ridge این است که به جای گرفتن مجذور ضرایب، قدرمطلق آنها در نظر گرفته می شود. این نوع منظم سازی (L1) می تواند به ضرایب صفر منجر شود، یعنی برخی از ویژگی ها برای ارزیابی خروجی کاملاً نادیده گرفته می شوند. بنابراین رگرسیون Lasso نه تنها به کاهش تناسب بیش از حد کمک می کند، بلکه می تواند در انتخاب ویژگی به ما کمک کند.

- تفاوت ها:

۱. جمله پنالتی: تفاوت اصلی بین رگرسیون ridge و lasso در عبارت پنالتی مورد استفاده نهفته است. رگرسیون ridge، بزرگی مجذور ضرایب (هنجار 2L) را به عنوان جریمه اضافه می کند، در حالی که رگرسیون lasso از مقادیر مطلق ضرایب (هنجار 1L) به عنوان جریمه استفاده می کند.
۲. انقباض ضریب: رگرسیون ridge می تواند ضرایب را به سمت صفر کوچک کند اما نمی تواند آنها را به طور کامل حذف کند، در حالی که رگرسیون lasso می تواند برخی از ضرایب را دقیقاً صفر تنظیم کند و به طور موثر انتخاب ویژگی را انجام دهد.
۳. فضای راه حل: رگرسیون Ridge تمام ویژگی های مدل را حفظ می کند اما تأثیر آنها را کاهش می دهد، در حالی که رگرسیون lasso می تواند زیر مجموعه ای از ویژگی ها را انتخاب کند و بقیه را کنار بگذارد.
۴. پراکندگی: رگرسیون lasso با حذف خودکار ویژگی های نامربوط، مدل های پراکنده تولید می کند، در حالی که رگرسیون Ridge انتخاب خودکار ویژگی را انجام نمی دهد.

- شباهت ها:

۱. منظم سازی: هر دو رگرسیون ريج و رگرسیون کمند تکنیک های منظم سازی هستند که برای مدل های رگرسیون خطی اعمال می شوند. هدف آنها جلوگیری از تطبیق بیش از حد با اضافه کردن یک عبارت جریمه به تابع هدف است.
۲. پیچیدگی کنترل: هر دو روش پیچیدگی مدل را با اضافه کردن یک عبارت جریمه به تابع هدف کنترل می کنند که تأثیر برخی ویژگی ها را کاهش می دهد یا بزرگی ضرایب را کاهش می دهد.

۳. پارامتر λ : هر دو رگرسیون ریدج و رگرسیون کمند دارای پارامتر λ هستند که قدرت عبارت پناستی را کنترل می کند. یک مقدار λ بزرگتر منجر به منظم سازی قوی تر می شود که منجر به ضرایب کوچک تر و حذف ویژگی های بالقوه بیشتر می شود.

۴. جایگزینی منظم سازی: هر دو تکنیک شامل یک مبادله بین به حداقل رساندن مجموع باقیمانده مربع ها (برای تناسب خوب با داده ها) و حداقل کردن مدت مجازات (برای کنترل پیچیدگی و جلوگیری از برازش بیش از حد) است.

(ج)

$$X = \begin{bmatrix} 1 & 42 \\ 1 & 74 \\ 1 & 48 \\ 1 & 35 \\ 1 & 52 \\ 1 & 24 \\ 1 & 40 \end{bmatrix} \quad y = \begin{bmatrix} 98 \\ 130 \\ 140 \\ 111 \\ 182 \\ 100 \\ 135 \end{bmatrix}$$

$$\beta = \underbrace{(X^T X)^{-1}}_{\begin{bmatrix} 7 & 341 \\ 341 & 18181 \end{bmatrix}^{-1}} \underbrace{(X^T y)}_{\begin{bmatrix} 1133 \\ 42948 \end{bmatrix}} = \frac{1}{10989} \begin{bmatrix} 18181 & -341 \\ -341 & 7 \end{bmatrix} \begin{bmatrix} 1133 \\ 42948 \end{bmatrix} \approx \begin{bmatrix} 45.46 \\ 1.51 \end{bmatrix}$$

$$y = 1.51x + 45.46 \xrightarrow{x=40} y = 105.18$$