به نام خدا

مینی پروژه اول درس مبانی سیستم‌های هوشمند

محمدمهدی کرمی - 40008373

لینک گیتهاب مبانی سیستم‌های هوشمند

پرسش اول

لینک دفترچه کولب پرسش اول

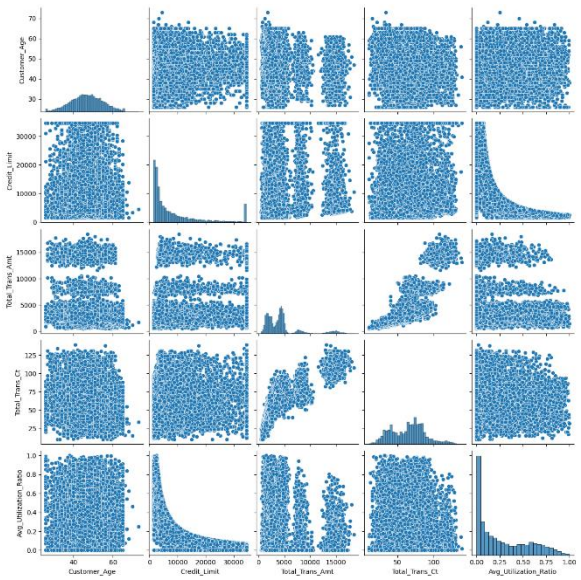
مجموعه داده را در کولب از ککل دانلود کرده و آن را از فایل فشرده خارج می‌کنیم.

پرسش اول - بخش اول

- این مجموعه داده مربوط به مشتریان یک بانک است که هدف آن پیش‌بینی ترک مشتریان از خدمات کارت اعتباری است. در این مجموعه، داده‌هایی نظیر سن، وضعیت تأهل، محدودیت اعتبار کارت، و دسته‌بندی کارت اعتباری جمع‌آوری شده‌اند. هدف اصلی از استفاده از این مجموعه داده، پیش‌بینی احتمال ترک مشتریان (churn) است تا بانک بتواند به صورت پیشگیرانه اقدامات لازم را انجام دهد و مشتریان خود را حفظ کند.
- در ابتدا، طبق توضیحات داده‌شده، باید دو ستون آخر که مربوط به پیش‌بینی مدل بیز ساده هستند را نادیده بگیریم، زیرا این ستون‌ها ویژگی نیستند و برای تجزیه و تحلیل نیاز به حذف دارند. در این مجموعه داده، ۲۱ ویژگی وجود دارند که عبارتند از: شناسه مشتری، پرچم ترک مشتری، سن مشتری، جنسیت، تعداد افراد تحت تکفل، سطح تحصیلات، وضعیت تأهل، دسته‌بندی درآمد، دسته‌بندی کارت اعتباری، تعداد ماه‌ها با بانک، تعداد ارتباطات با بانک، تعداد ماه‌های غیر فعال در ۱۲ ماه گذشته، تعداد تماس‌ها با بانک در ۱۲ ماه گذشته، محدودیت اعتبار کارت اعتباری، مجموع موجودی‌های چرخشی، میانگین اعتبار آزاد، تغییر در مجموع مبلغ تراکنش‌ها بین فصل‌های چهارم و اول، مجموع مبلغ تراکنش‌های انجام‌شده، تعداد تراکنش‌ها، تغییر در تعداد تراکنش‌ها بین فصل‌های چهارم و اول، میانگین نسبت استفاده از اعتبار.
- تعداد نمونه‌ها در این مجموعه داده ۱۰۱۲۷ است.

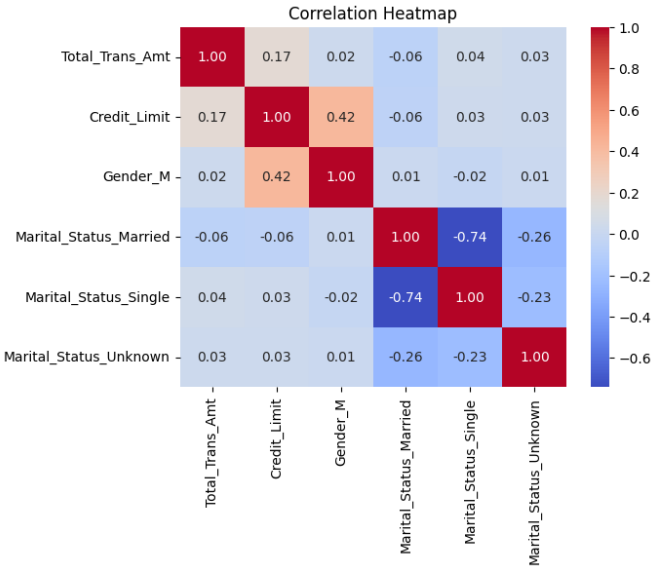
پرسش اول - بخش دوم

پنج ویژگی سن مشتری، محدودیت اعتبار کارت اعتباری، مجموع موجودی‌های چرخشی، مجموع مبلغ تراکنش‌های انجام شده و میانگین نسبت استفاده از اعتبار را انتخاب کرده و پخش داده را نمایش می‌دهیم.



پرسش اول - بخش سوم

چهار ویژگی جنسیت، وضعیت تاهل، مجموع موجودی‌های چرخشی و محدودیت اعتبار کارت اعتباری را انتخاب کرده و همبستگی موجود میان آن‌ها را به صورت نقشه حرارتی نمایش می‌دهیم.

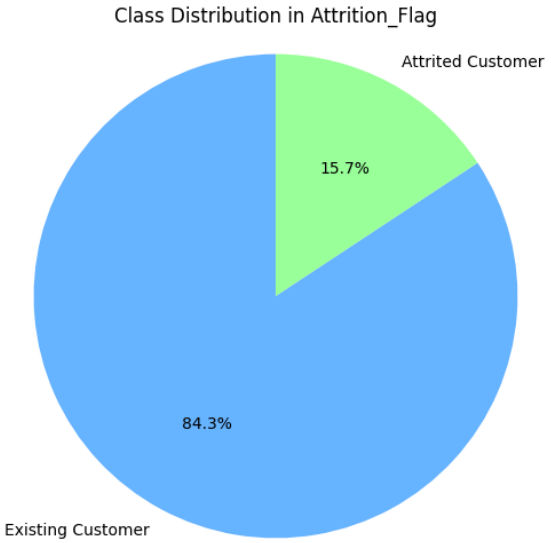


پرسش اول - بخش چهارم

قبل از حذف مقادیر مفقود (NaN)، مجموعه داده شامل ۱۰,۱۲۷ ردیف و ۲۱ ستون بود. پس از حذف ردیف‌هایی که دارای مقادیر NaN بودند، تعداد ردیف‌ها به ۷,۰۸۱ کاهش یافت و تعداد ستون‌ها همچنان ثابت ماند و ۲۱ باقی ماند. این تغییر نشان‌دهنده حذف ۳,۰۴۶ ردیف از داده‌ها به دلیل وجود مقادیر مفقود است. مقادیر مفقود (NaN) در سه ستون education_level (۱۵۱۹ مقدار مفقود)، marital_status (۷۴۹ مقدار مفقود) و income_category (۱۱۱۲ مقدار مفقود) وجود داشتند.

پرسش اول - بخش پنجم

- ویژگی Attrition_Flag دو کلاس دارد: مشتری موجود با ۸,۵۰۰ نمونه و مشتری ترک کرده با ۱,۶۲۷ نمونه.
- پخش داده موجود در این ویژگی را به صورت یک نمودار دایره‌ای نمایش می‌دهیم.



- عدم تعادل در داده‌ها، مانند ویژگی Attrition_Flag، می‌تواند باعث بروز مشکلاتی در عملکرد مدل‌های یادگیری ماشین شود، زیرا مدل تمایل دارد پیش‌بینی‌های خود را به سمت کلاس اکثریت متمایل کند. این امر باعث می‌شود که مدل نتواند به خوبی کلاس اقلیت را شناسایی کند و دقت پیش‌بینی‌ها برای مشتریان ترک کرده پایین بیاید.
- برای اصلاح این مشکل، می‌توان از روش‌هایی مانند افزایش نمونه‌های کلاس اقلیت (مانند استفاده از SMOTE)، کاهش نمونه‌های کلاس اکثریت، تنظیم وزن‌های کلاس در الگوریتم‌های یادگیری ماشین، و یا استفاده از الگوریتم‌های متعادل‌سازی مانند BalancedRandomForest استفاده کرد. این روش‌ها به مدل کمک می‌کنند تا به طور متوازن‌تر بین دو کلاس عمل کند و پیش‌بینی‌های بهتری برای کلاس اقلیت داشته باشد.
- اگر بخواهیم از یک الگوریتم برای متعادل کردن داده‌ها استفاده کنیم، باید این کار قبل از تقسیم‌بندی داده‌ها به بخش‌های آموزش و آزمون انجام دهیم. زیرا اگر این کار پس از تقسیم داده‌ها صورت گیرد، داده‌های آموزش و آزمون به هم مخلوط می‌شوند و این ممکن است باعث نشت داده‌ها و ارزیابی نادرست عملکرد مدل شود. بنابراین، باید ابتدا داده‌ها متعادل شوند تا مدل به درستی آموزش ببیند و نتایج واقعی‌تر و دقیق‌تری به دست آید.

پرسش اول - بخش ششم

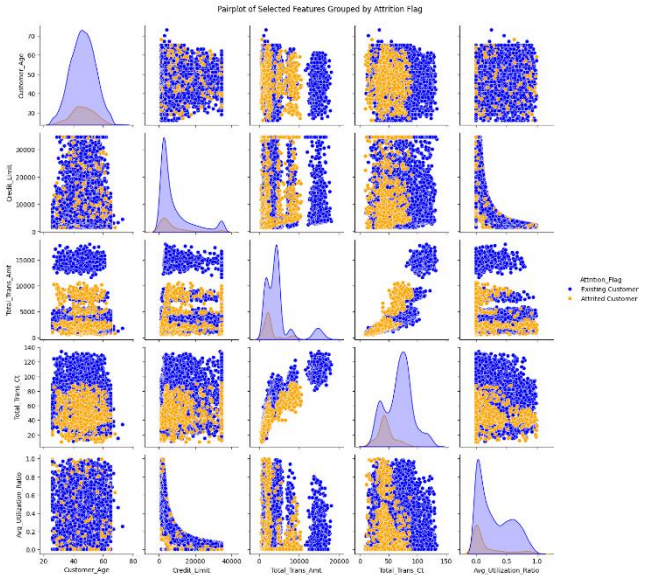
مدل بدون تعادل به خوبی کلاس غالب (کلاس صفر) را پیش‌بینی می‌کند و دقت بالایی در داده‌های آموزشی دارد، اما در شبیه‌سازی کلاس کمیاب (کلاس یک) ضعیف عمل می‌کند و حساسیت آن تنها ۷۷ درصد است. این مشکل در داده‌های نامتعادل شایع است، زیرا مدل تمایل دارد بیشتر روی پیش‌بینی کلاس پرتعداد تمرکز کند.

در مقابل، مدل با تعادل داده‌ها عملکرد بهتری در پیش‌بینی کلاس یک دارد. پس از متعادل کردن داده‌ها، حساسیت کلاس یک به ۸۶ درصد افزایش یافته و دقت کلی مدل حفظ شده است. این نشان می‌دهد که متعادل کردن داده‌ها به مدل کمک می‌کند تا بهتر بین کلاس‌ها تمایز قائل شود.

در نتیجه، مدل با تعادل داده‌ها کارایی بهتری در پیش‌بینی کلاس‌های کمیاب دارد و عملکرد کلی آن در مقایسه با مدل بدون تعادل بهبود یافته است.

پرسش اول - بخش امتیازی

پنج ویژگی سن مشتری، محدودیت اعتبار کارت اعتباری، مجموع موجودی‌های چرخشی، مجموع مبلغ تراکنش‌های انجام شده و میانگین نسبت استفاده از اعتبار را انتخاب کرده و پخش داده را با توجه به کلاس‌های مختلف موجود در ویژگی Attrition_Flag نمایش می‌دهیم.



پایان پرسش اول

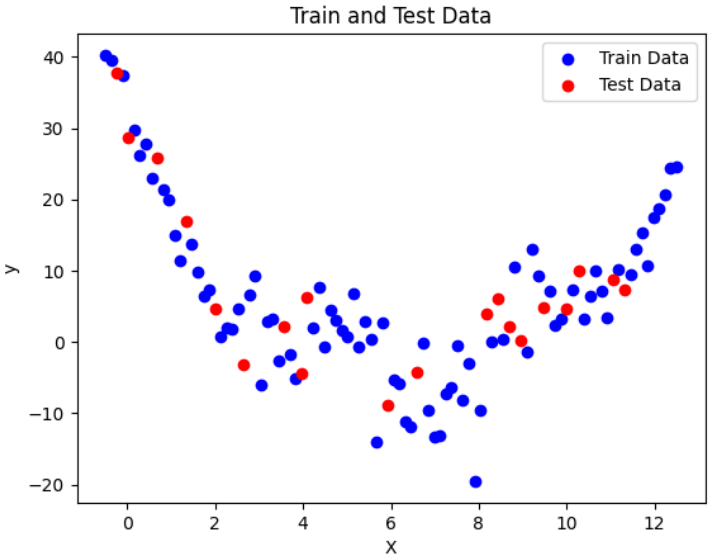
پرسش دوم

لینک دفترچه کولب پرسش دوم

مجموعه داده را از گوگل درایو دانلود کرده و آن را در کولب آپلود می‌کنیم.

پرسش دوم - بخش اول

داده‌ها را با به دو گروه آموزش و آزمون تقسیم می‌کنیم.



پرسش دوم - بخش دوم

سه معیار رایج برای سنجش عملکرد مدل‌های رگرسیون عبارتند از:

- **خطای میانگین مطلق** (MAE - Mean Absolute Error)**:** این معیار نشان می‌دهد که به طور متوسط، پیش‌بینی مدل چقدر از مقدار واقعی انحراف دارد.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- **خطای میانگین مربعات** (MSE - Mean Squared Error)**:** این معیار به میزان تفاوت مربعی بین مقادیر پیش‌بینی‌شده و واقعی می‌پردازد.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- **ضریب تعیین** (R^2)**:** این معیار نشان می‌دهد که مدل چه مقدار از واریانس داده‌ها را توضیح می‌دهد.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

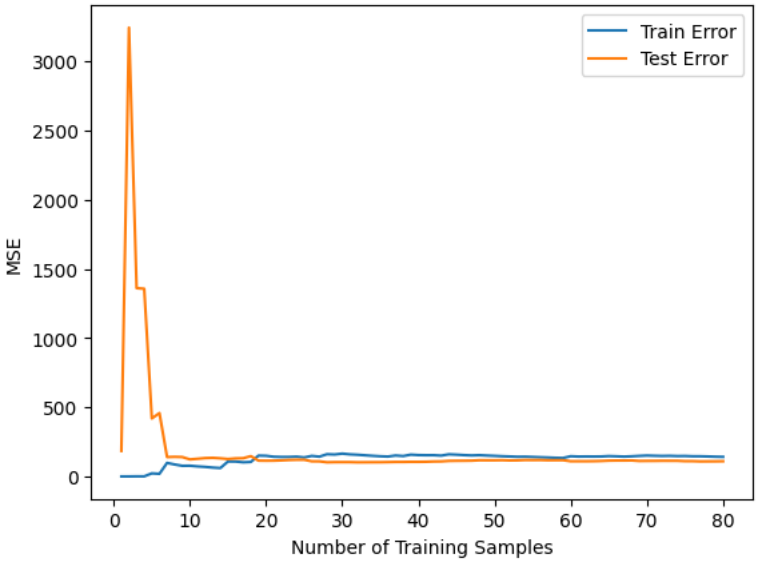
پرسش دوم - بخش سوم

مدل خطی درجه اول برای داده‌هایی که رابطه غیرخطی دارند، معمولاً قادر به ارائه تخمین‌های دقیقی نخواهد بود. در این حالت، چون مدل خطی تنها قادر به مدل‌سازی روابط خطی است، نمی‌تواند به خوبی ویژگی‌های داده‌هایی که به طور ذاتی دارای الگوی غیرخطی (مثل

رابطه درجه دو) هستند را بازسازی کند. بنابراین، نتیجه مدل خطی درجه اول ممکن است خطای زیادی داشته باشد و به طور کلی، این مدل برای این نوع داده‌ها مناسب نخواهد بود.

پرسش دوم - بخش چهارم

در حالت کلی، با افزایش تعداد داده‌های آموزشی، خطای آموزش معمولاً کاهش می‌یابد، زیرا مدل به تدریج بهتر می‌تواند داده‌های آموزش را پیش‌بینی کند و تطبیق بیشتری با آن‌ها پیدا می‌کند. از سوی دیگر، خطای آزمون در ابتدا با افزایش داده‌های آموزشی کاهش می‌یابد، زیرا مدل قادر به شبیه‌سازی بهتر ویژگی‌های داده‌های کلی می‌شود. اما پس از رسیدن به یک حد معین از داده‌های آموزشی، خطای آزمون ممکن است ثابت یا حتی افزایش یابد، چرا که مدل ممکن است به بیش‌برازش دچار شود و نتواند به درستی بر روی داده‌های جدید (آزمون) عمل کند.



پرسش دوم - بخش پنجم

در این شرایط، اگر خطای انسان برابر ۱ باشد و خطای مدل فعلی برابر ۱۰ باشد، افزایش داده‌های آموزشی ممکن است به کاهش خطای مدل کمک کند، اما به طور قطعی نمی‌توان گفت که خطای مدل دقیقاً به اندازه خطای انسان (۱) کاهش خواهد یافت.

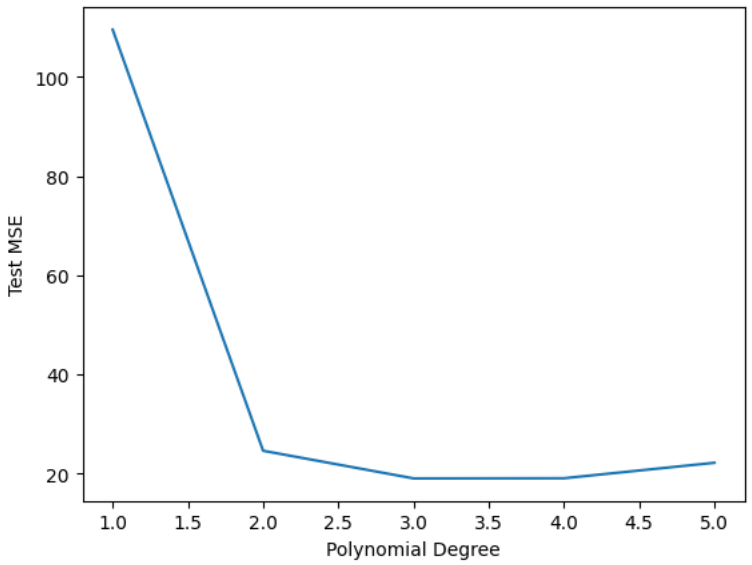
اگر مدل فعلی به دلیل **بیش‌برازش** (Overfitting)دارای خطای بالا است، استفاده از داده‌های بیشتر می‌تواند به کاهش خطای مدل کمک کند، زیرا مدل قادر به تعمیم بهتر به داده‌های جدید خواهد بود. اما اگر مدل هنوز نتواند به خوبی از داده‌ها الگو بگیرد یا ساختار پیچیده‌تری نیاز داشته باشد، ممکن است خطای آن حتی با داده‌های بیشتر کاهش پیدا نکند.

به طور کلی، هدف کاهش خطا تا حد ممکن است، اما نمی‌توان تضمین کرد که خطای مدل دقیقاً به اندازه خطای انسان کاهش یابد، زیرا مدل‌های یادگیری ماشین ممکن است محدودیت‌های خود را داشته باشند و همیشه قادر به رسیدن به عملکرد انسانی نباشند.

پرسش دوم - بخش ششم

در ابتدا، مدل رگرسیون خطی با درجه اول (خط ساده) دارای خطای زیادی بود، زیرا داده‌ها به طور واضح از یک رابطه غیرخطی (شبیه به تابع درجه دو) پیروی می‌کردند. با اضافه کردن درجه دوم (افزودن جمله x^2 به مدل)، خطا به طور چشمگیری کاهش یافت، چرا که مدل توانست رابطه غیرخطی میان ورودی و خروجی را بهتر شبیه‌سازی کند. پس از اضافه کردن جملات بیشتر (درجه‌های بالاتر)، تغییرات قابل توجهی در خطای مدل مشاهده نشد. این نشان می‌دهد که پس از درجه دوم، مدل به یک حد معین از تطابق با داده‌ها رسید. به طور کلی، با توجه به اینکه داده‌ها به صورت تابع درجه دو بودند، مدل با افزودن جمله x^2 پیش‌بینی دقیق‌تری انجام داد و بعد از آن، افزایش تعداد جملات مدل تاثیر زیادی بر کاهش خطا نداشت.

نمودار خطا در مقابل تعداد جملات، ابتدا کاهش زیادی را نشان می‌دهد و سپس به یک سطح ثابت می‌رسد که نشان‌دهنده اشباع مدل در تطابق با داده‌ها است.



پرسش دوم - بخش ششم

سه الگوریتم رگرسیون از کتابخانه scikit-learn که در اینجا مورد استفاده قرار گرفته‌اند، عبارتند از:

- **رگرسیون خطی (Linear Regression):** این الگوریتم یکی از ساده‌ترین مدل‌های رگرسیونی است که سعی می‌کند یک خط راست (یا چند خط در صورت وجود متغیرهای بیشتر) را پیدا کند که کمترین اختلاف را با نقاط داده‌ها داشته باشد. هدف آن کمینه کردن مجموع مربعات خطا (MSE) بین پیش‌بینی‌های مدل و مقادیر واقعی است. این مدل برای داده‌های خطی مناسب است و به سادگی قابل تفسیر است.
- **درخت تصمیم (Decision Tree):** درخت تصمیم یک الگوریتم رگرسیون غیرخطی است که داده‌ها را بر اساس ویژگی‌ها به دسته‌های مختلف تقسیم می‌کند و برای هر بخش یک مقدار پیش‌بینی می‌کند. درخت تصمیم مدل‌هایی است که به صورت سلسله‌مراتبی از سوالات بله/خیر به نتیجه می‌رسند. این مدل به خوبی می‌تواند روابط غیرخطی و پیچیده را مدل کند، ولی ممکن است به راحتی دچار بیش‌برازش شود.
- **درخت تصمیم (Decision Tree):** درخت تصمیم یک الگوریتم رگرسیون غیرخطی است که داده‌ها را بر اساس ویژگی‌ها به دسته‌های مختلف تقسیم می‌کند و برای هر بخش یک مقدار پیش‌بینی می‌کند. درخت تصمیم مدل‌هایی است که به صورت سلسله‌مراتبی از سوالات بله/خیر به نتیجه می‌رسند. این مدل به خوبی می‌تواند روابط غیرخطی و پیچیده را مدل کند، ولی ممکن است به راحتی دچار بیش‌برازش شود.

نتایج نشان می‌دهند که جنگل تصادفی بهترین عملکرد را از نظر دقت دارد و کمترین خطا را به خود اختصاص داده است. درخت تصمیم نیز نسبت به رگرسیون خطی عملکرد بهتری داشته، ولی هنوز نسبت به جنگل تصادفی دقت کمتری دارد. رگرسیون خطی به دلیل سادگی مدل و ناتوانی در مدل‌سازی روابط غیرخطی، بالاترین خطا را داشته است.

به طور کلی، مدل‌های پیچیده‌تر مانند درخت تصمیم و جنگل تصادفی عملکرد بهتری دارند، به ویژه زمانی که داده‌ها روابط غیرخطی یا پیچیده‌ای دارند.

پرسش دوم - بخش امتیازی

Regularization یکی از تکنیک‌های مهم در یادگیری ماشین است که برای مقابله با overfitting و بهبود توانایی تعمیم مدل استفاده می‌شود. در رگرسیون خطی، هدف این است که مقادیر پارامترهای مدل (ضریب‌ها) را طوری تنظیم کنیم که مدل هم به خوبی داده‌های

آموزشی را پیش‌بینی کند و هم قابلیت تعمیم به داده‌های جدید را داشته باشد. Regularization با افزودن یک اصطلاح جریمه به تابع هزینه (که معمولا میانگین مربعات خطا است)، پارامترها را محدود می‌کند تا از پیچیدگی مدل جلوگیری کند.

انواع Regularization:

- **Ridge Regression (L2 Regularization):** در این روش، یک جریمه به مربع مقادیر ضرایب مدل (پارامترها) اضافه می‌شود. هدف کاهش مقادیر بزرگ ضرایب است تا مدل از پیچیدگی زیاد و overfitting جلوگیری کند. فرمول تابع هزینه به صورت زیر است:

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(y_i - \hat{y_i})^2 + \lambda\sum_{j=1}^{n}\theta_j^2$$

که در آن، λ پارامتر تنظیم است که میزان جریمه را کنترل می‌کند.

- **Lasso Regression (L1 Regularization):** در این روش، جریمه به مقدار مطلق ضرایب اضافه می‌شود. این باعث می‌شود برخی ضرایب به صفر برسند و در نتیجه، مدل ساده‌تری به دست می‌آید که می‌تواند به انتخاب ویژگی‌های مهم‌تر کمک کند. فرمول تابع هزینه به شکل زیر است:

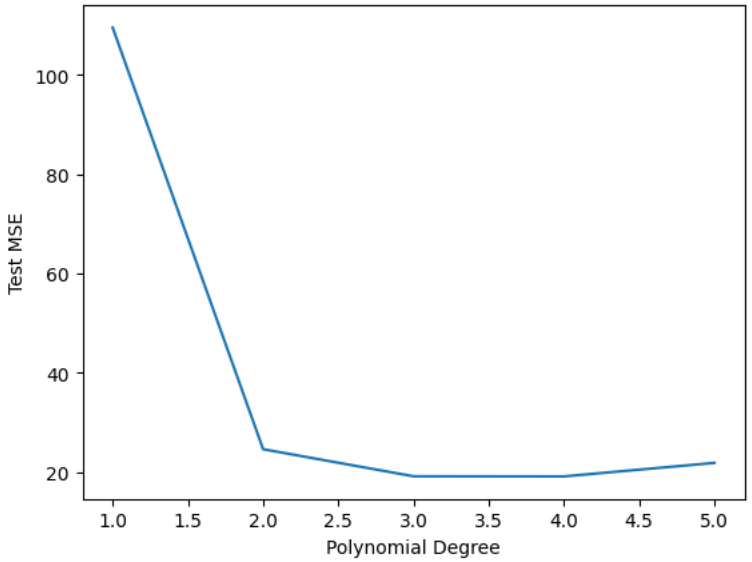
$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(y_i - \hat{y_i})^2 + \lambda\sum_{j=1}^{n}|\theta_j|$$

- **Elastic Net:** ترکیبی از L1 و L2 regularization است. این روش به شما این امکان را می‌دهد که از هر دو نوع جریمه (مطلق و مربعی) برای تعادل بین کاهش پیچیدگی مدل و انتخاب ویژگی‌ها استفاده کنید. فرمول آن به صورت زیر است:

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(y_i - \hat{y_i})^2 + \lambda_1\sum_{j=1}^{n}|\theta_j| + \lambda_2\sum_{j=1}^{n}\theta_j^2$$

که در آن λ_1 و λ_2 پارامترهای تنظیم هستند.

این روش‌ها می‌توانند در شرایط مختلف به بهبود عملکرد مدل و جلوگیری از بیش‌برازش کمک کنند.



پایان پرسش دوم