به نام خدا

1928
K. N. Toosi University of Technology

دانشگاه صنعتی خواجه نصیرالدین طوسی

دانشکده برق

مبانی سیستم های هوشمند

مینی پروژه ۱

[شهاب مقدادی نیشابوری]

[۴۰۰۰۹۴۴۳]

استاد : آقای دکتر مهدی علیاری

سوال ۱

سوال ۱.۱:

این مجموعه داده شامل اطلاعات ۱۰,۰۰۰ مشتری بانک است که جزئیاتی مانند سن، درآمد، وضعیت تأهل، حد اعتبار کارت و دسته‌بندی کارت اعتباری را در ۱۸ ویژگی ارائه می‌دهد. تنها ۱۶.۰۷٪ از مشتریان قطع همکاری کرده‌اند که پیش‌بینی دقیق آن‌ها را برای مدل‌های آموزشی چالش‌برانگیز می‌کند.

در این دیتاست به مجموعه ای از اطلاعات که در ۲۳ ستون پخش هستند، اشاره شده. در این بخش به این که هر ستون نشان دهنده چه پارامتری است میپردازیم:

۱. شماره مشتری(یکتا)
۲. وضعیت حساب(در صورت بسته بودن ۱ و در غیر این صورت ۰)
۳. سن مشتری
۴. جنسیت(m و f)
۵. تعداد افراد وابسته(۰ تا ۵)
۶. سطح تحصیلات
۷. وضعیت تاهل
۸. سطح درامد
۹. نوع کارت اعتباری
۱۰. تعداد ماه همکاری با بانک
۱۱. تعداد محصولاتی که مشتری استفاده کرده
۱۲. تعداد ماه هایی که مشتری در آن فعال نبوده در ۱۲ ماه بخیر
۱۳. تعداد تماس ها در ۱۲ ماه اخیر
۱۴. محدودیت هزینه روی کارت اعتباری
۱۵. کل موجودی گردان روی کارت اعتباری
۱۶. خط اعتبار قابل استفاده (میانگین ۱۲ ماه گذشته)
۱۷. تغییر در مبلغ تراکنش (سه‌ماهه چهارم نسبت به سه‌ماهه اول)
۱۸. مجموع مبلغ تراکنش‌ها (۱۲ ماه گذشته)
۱۹. مجموع تعداد تراکنش‌ها (۱۲ ماه گذشته)

۲۰. تغییر در تعداد تراکنش‌ها (سه‌ماهه چهارم نسبت به سه‌ماهه اول)

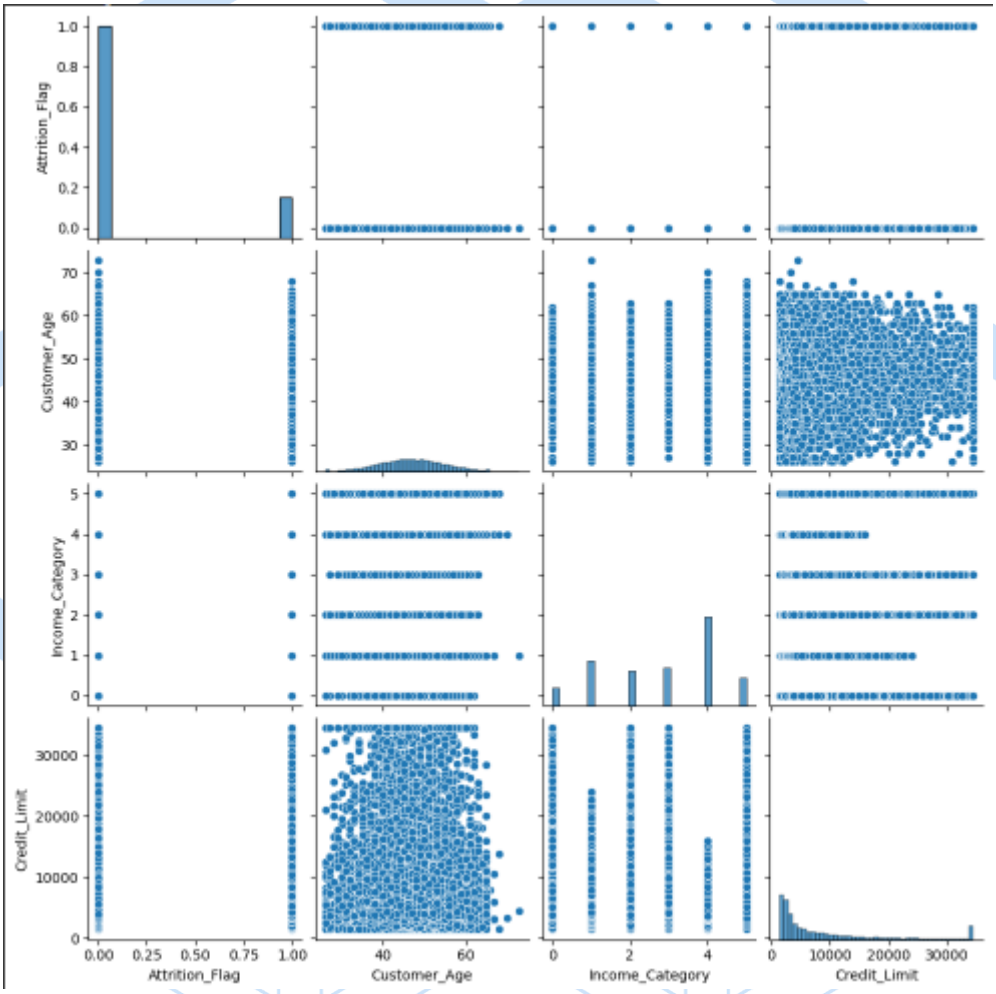
۲۱. نسبت میانگین استفاده از کارت

۲۲. (Naive Bayes)

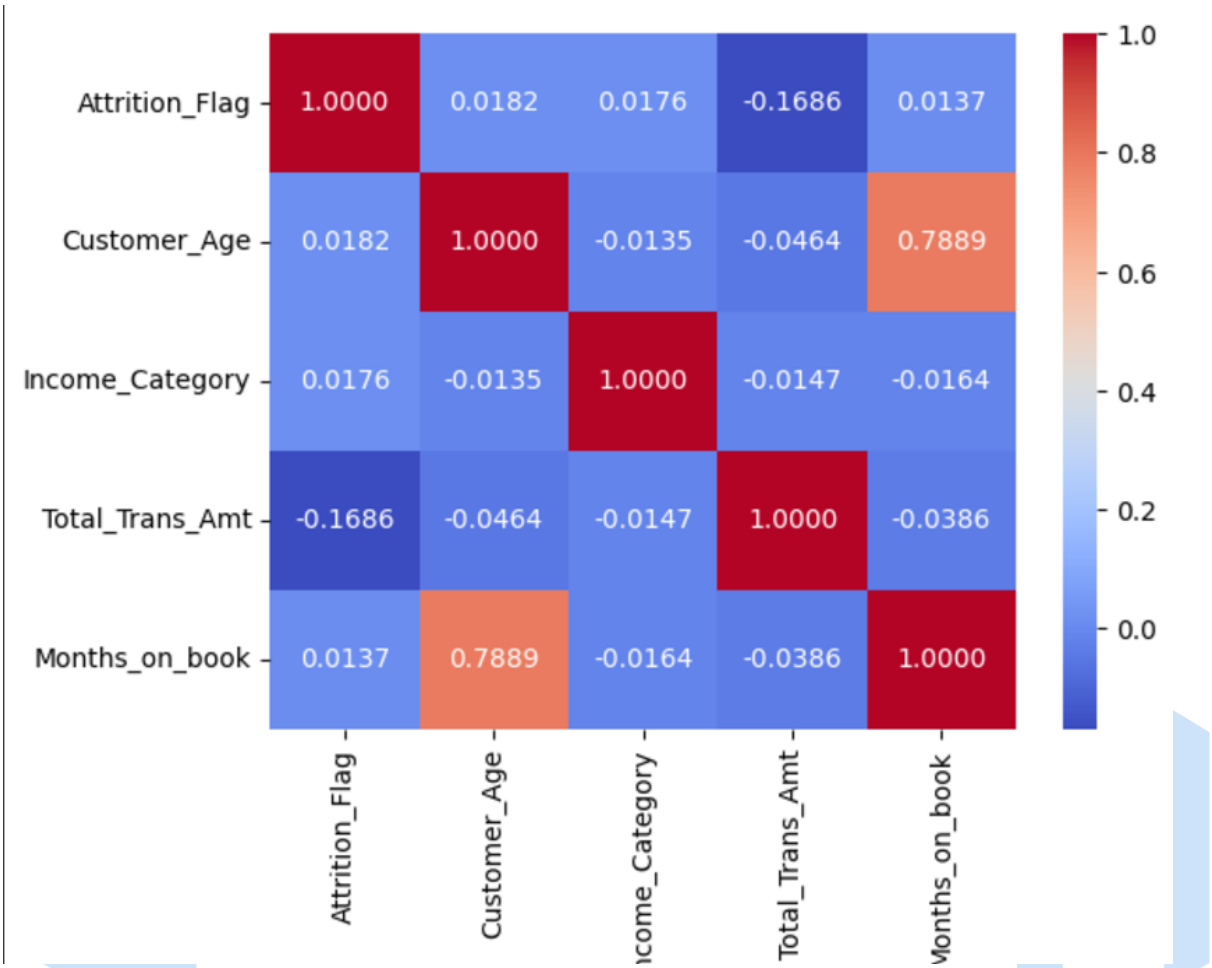
۲۳. (Naive Bayes)

لازم به ذکر است که پیشنهاد شده که به دو ستون اخر توجه نشود و قبل از پردازش حذف شوند

سوال ۱.۲:



سوال ۱.۳:

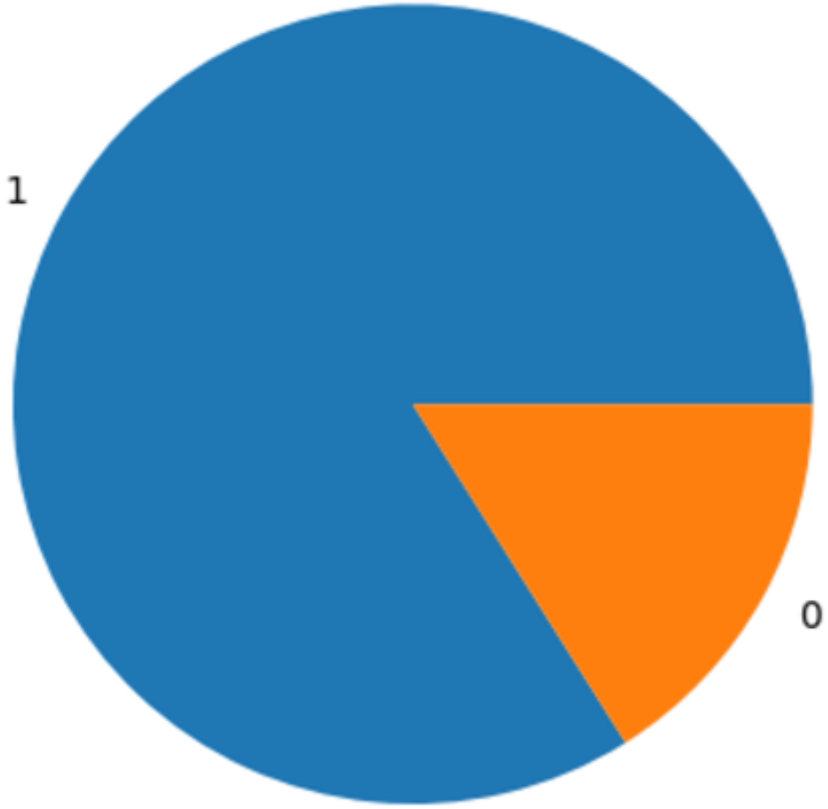

Attrition_Flag
Customer_Age
Income_Category
Total_Trans_Amt
Months_on_book
1.0000 0.0182 0.0176 -0.1686 0.0137
0.0182 1.0000 -0.0135 -0.0464 0.7889
0.0176 -0.0135 1.0000 -0.0147 -0.0164
-0.1686 -0.0464 -0.0147 1.0000 -0.0386
0.0137 0.7889 -0.0164 -0.0386 1.0000
1.0
0.8
0.6
0.4
0.2
0.0
Attrition_Flag
Customer_Age
Total_Trans_Amt

سوال ۱.۴:

```
[7]  data.isna().any().any()

     False
```

سوال ۱.۵:

دو کلاس در این ستون وجود دارد، در صورت بسته بوده حساب ۱، و در غیر این صورت ۰ میباشد.

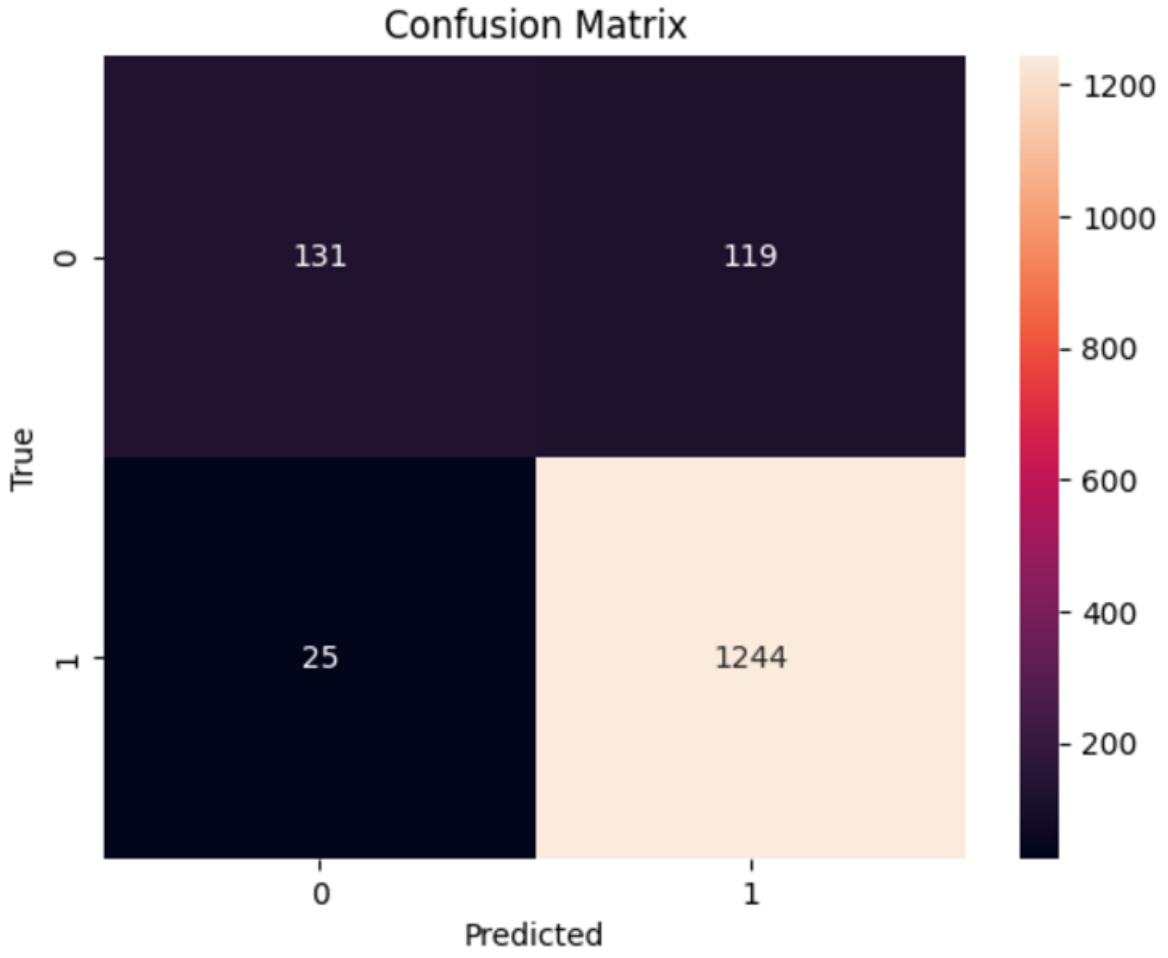


همانطور که مشاهده میشود، دیتای ارایه شده پخش مناسبی ندارد. این عامل باعث میشود که دقت مدل برای مشتریانی که قصد خروج دارند به صورت دقیق، و در عین حال برای کسانی که قصد خروج ندارند دقت خوبی نداشته باشند. برای حل این مشکل میتوان تعدادی از ستون های حاوی ۱ را حذف کرد یا میتوان با عوض کردن تابع هزینه، بار زیادی به اطلاعات ۰ تحمیل کند. برای نمونه متد SMOTE، دیتا

هایی را در میان دیتا های اقلیت موجود میکند که در نتیجه، به بالانس شدن دیتاست و در نتیجه منطقی تر شدن تابع هزینه کمک میکند. اگر قصد پیاده سازی این قبیل الگوریتم ها را داریم، باید بعد از split کردن این کار را انجام داد، چرا که نیازی به وجود این قبیل اطلاعات ساخته شده در دیتای train خود نداریم.

سوال ۱.۶:

نتایج پس از متعادل سازی نکردن اطلاعات:



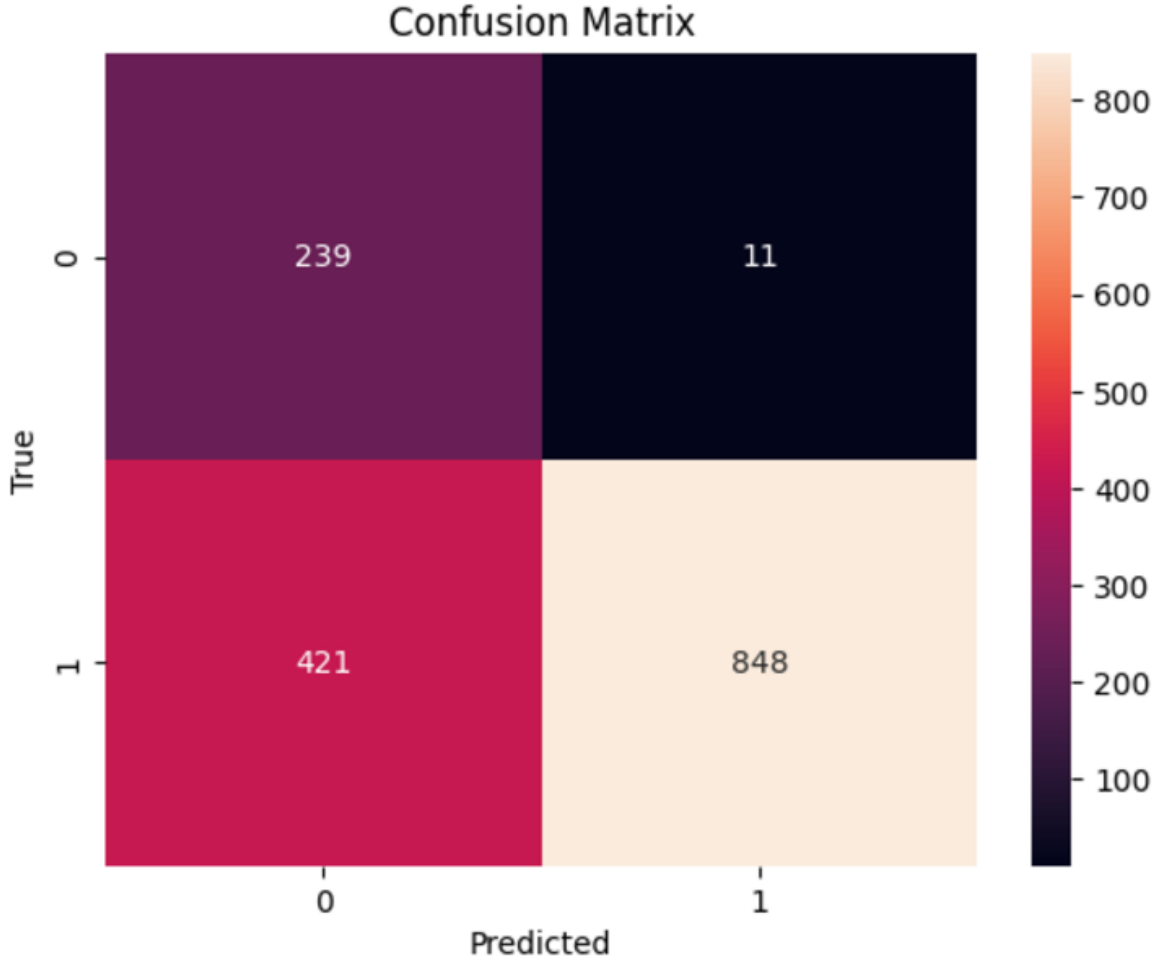
	precision	recall	f1-score	support
0.0	0.84	0.52	0.65	250
1.0	0.91	0.98	0.95	1269
accuracy			0.91	1519
macro avg	0.88	0.75	0.80	1519
weighted avg	0.90	0.91	0.90	1519

برای حل این سوال از الگوریتم Logistic Regression استفاده شده که در کلاس به صورت مفصل توضیح داده شده بود. برای penalty از مدل l1 استفاده شده(برای حذف فیچر های کم اهمیت تر). همینطور از تلرانس 0.001 استفاده شده و از solver liblinear استفاده شده که بهترین گزینه برای دیتاست های کوچک میباشد.

برای متعادل سازی اطلاعات، همانطور که اشاره شد میتوان از متد SMOTE استفاده کرد. پس از ساخت دیتای مصنوعی با استفاده از متد smote، مشاهده میشود که مدل توانایی پیش بینی کلاس minority را به صورت کامل از دست میدهد. لازم به ذکر است که برای مدل این بخش، از saga solver استفاده شده است که به صورت کلی، از liblinear سریع تر میباشد. برای آن که مدل همگرا شود نیز پارامتر max_iter به مقدار ۱۰۰۰، مقدار دهی شده است.

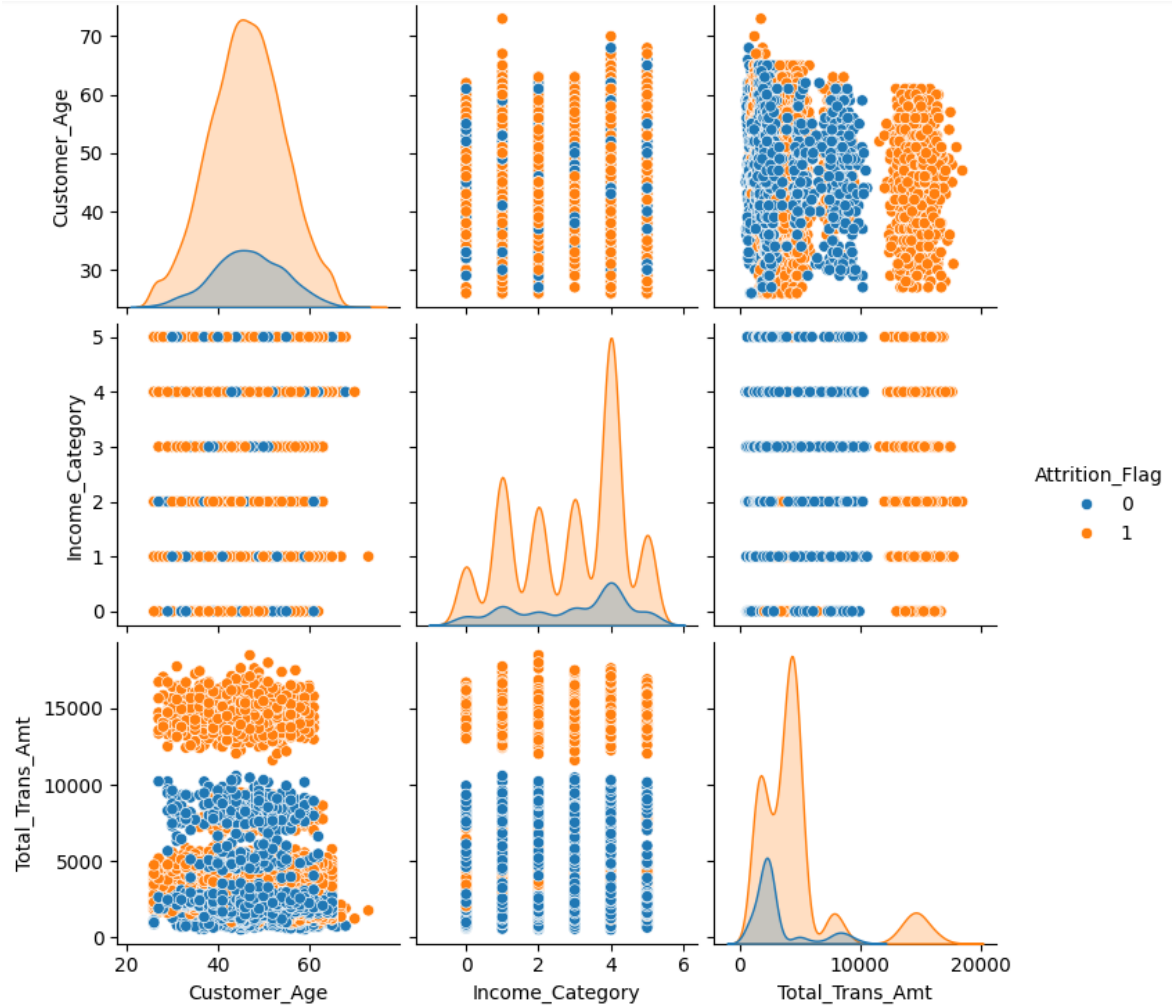
همچنین برای رسیدن به نتایج بهتر، فیچر ها را نرمالایز میکنیم:

```
0.7156023699802502
              precision    recall  f1-score   support

         0.0       0.36      0.96      0.53       250
         1.0       0.99      0.67      0.80      1269

    accuracy                           0.72      1519
   macro avg       0.67      0.81      0.66      1519
weighted avg       0.88      0.72      0.75      1519
```



امتیازی:

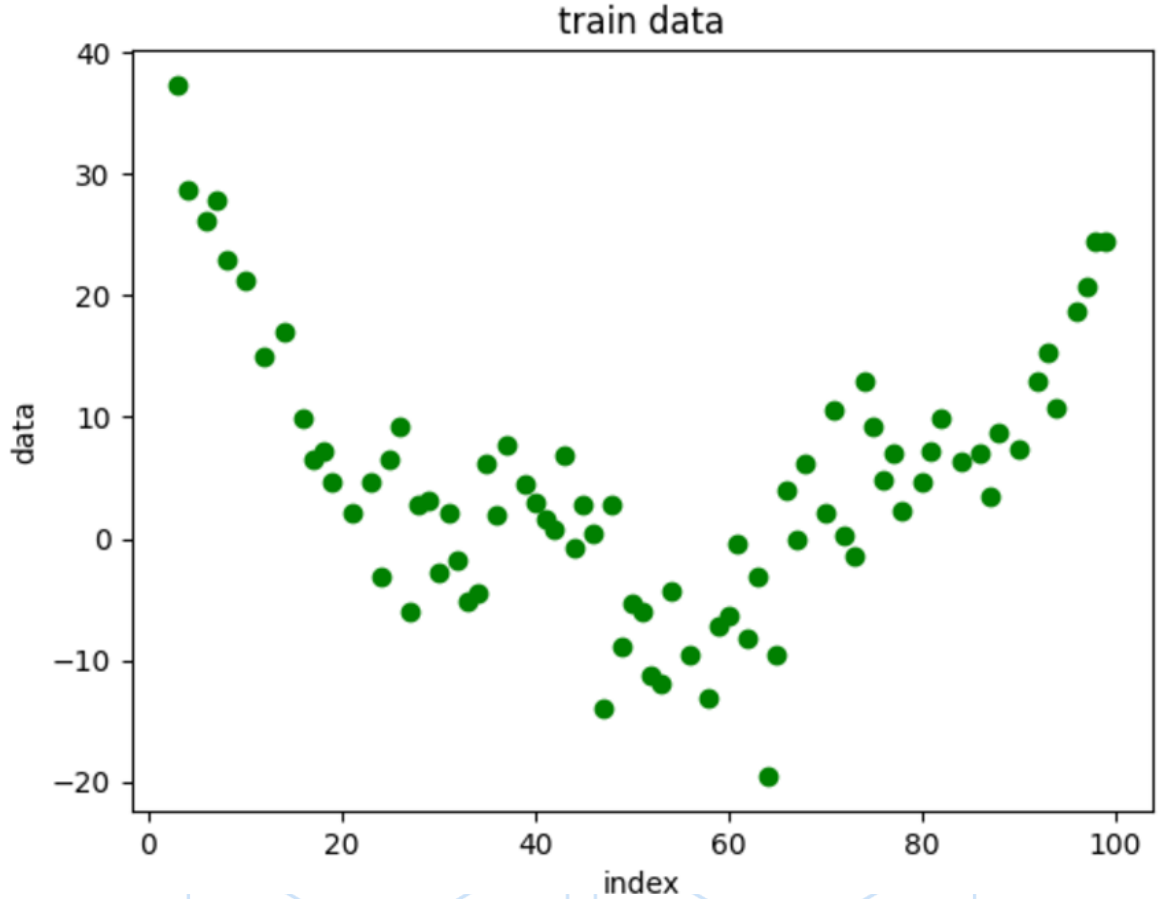
با استفاده از ست کردن hue به Attrition_Flag، میتوانیم پخش داده ها را نسبت به این پارامتر مشاهده کنیم:

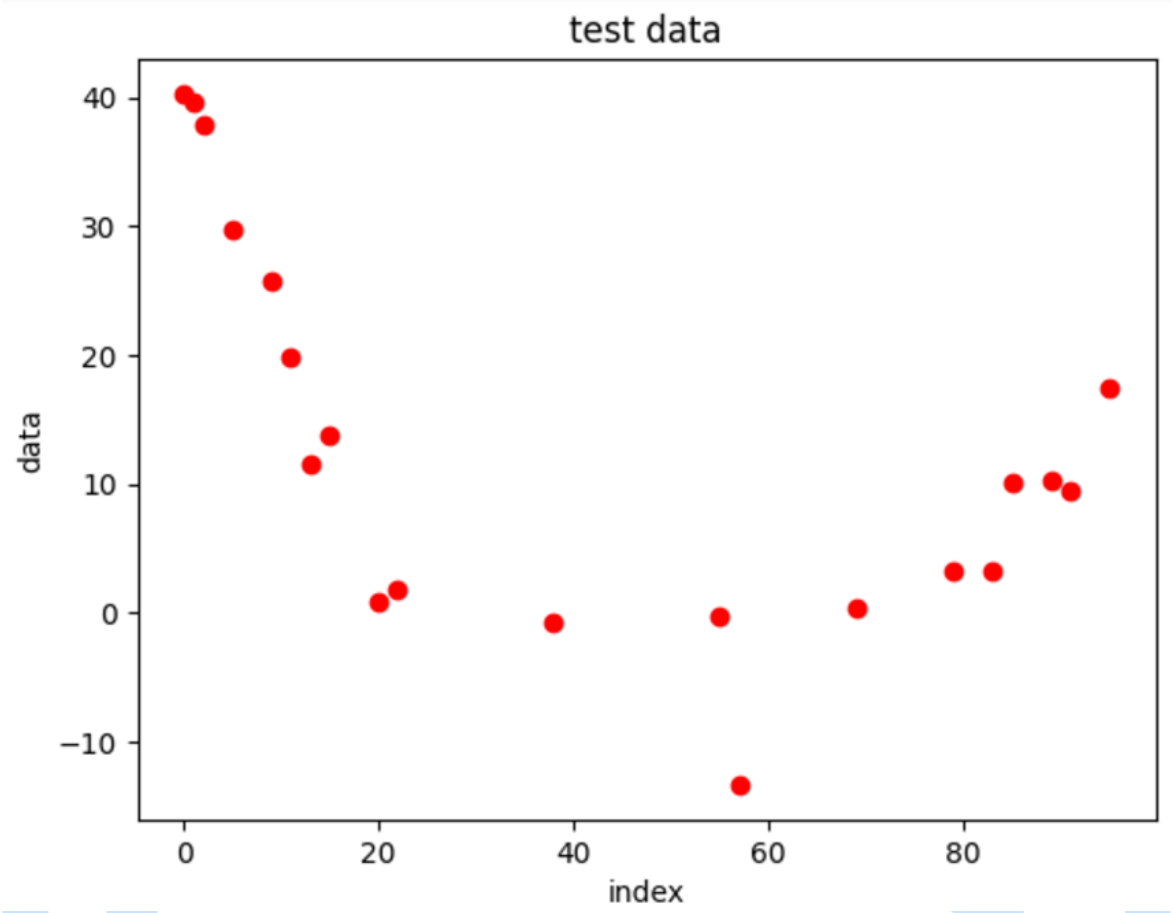


سوال ۲

سوال ۲.۱:

با استفاده از تابع train_test_split، با نسبت ۰.۲ به ۰.۸ اطلاعات test را از train جدا میکنیم، سپس با استفاده از تابع scatter، آنها را نمایش میدهیم:





سوال ۲.۲:

۱. میانگین مربع خطا :(MSE) میانگین مربع تفاوت بین مقادیر پیش‌بینی‌شده و واقعی را اندازه‌گیری می‌کند و به دلیل توان دوم، خطاهای بزرگ‌تر را برجسته می‌سازد.

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

۲. ضریب تعیین :(R-squared) نشان می‌دهد چه نسبتی از واریانس متغیر وابسته توسط مدل توضیح داده شده است، به طوری که مقادیر نزدیک به ۱ نشان‌دهنده تناسب بهتر مدل هستند.

$$R^2 = 1 - \frac{\mathrm{SS}_{\mathrm{res}}}{\mathrm{SS}_{\mathrm{tot}}}$$

$$\mathrm{SS}_{\mathrm{res}} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$\mathrm{SS}_{\mathrm{tot}} = \sum_{i=1}^{n}(y_i - \bar{y})^2$$

۳. میانگین قدر مطلق خطا :(MAE) میانگین تفاوت مطلق بین مقادیر پیش‌بینی‌شده و واقعی را نشان می‌دهد و معیاری قابل فهم‌تر از بزرگی خطا ارائه می‌دهد.

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

سوال ۲.۳:

در این بخش خواسته شده تا با یک مدل رگرسیون خطی، مدل داده شده تخمین زده شود. رگرسیون خطی یکی از ساده‌ترین و پرکاربردترین روش‌های تحلیل داده‌ها در آمار و یادگیری ماشین است که برای پیش‌بینی یک متغیر وابسته (هدف) بر اساس یک یا چند متغیر مستقل (ویژگی‌ها) استفاده می‌شود. ایده اصلی این روش بر مبنای پیدا کردن یک خط مستقیم (در حالت یک متغیر مستقل) یا یک صفحه/فراسطح (در حالت چند متغیر مستقل) است که بهترین برازش را برای داده‌ها داشته باشد. مدل رگرسیون خطی ساده با معادله $y=wx+b$ نمایش داده می‌شود که در آن y مقدار پیش‌بینی‌شده، x متغیر مستقل، wشیب خط (ضرایب) و b عرض از مبدأ است. در حالت رگرسیون خطی چندگانه، این معادله به صورت کلی‌تر به $y=w1x1+w2x2+\cdots+wnxn$ تبدیل می‌شود.

لازم به ذکر است که در این سوال، پیش از شروع یادگیری داده ها را نرمالایز کردیم تا به مشکل overflow برخورد نکنیم.

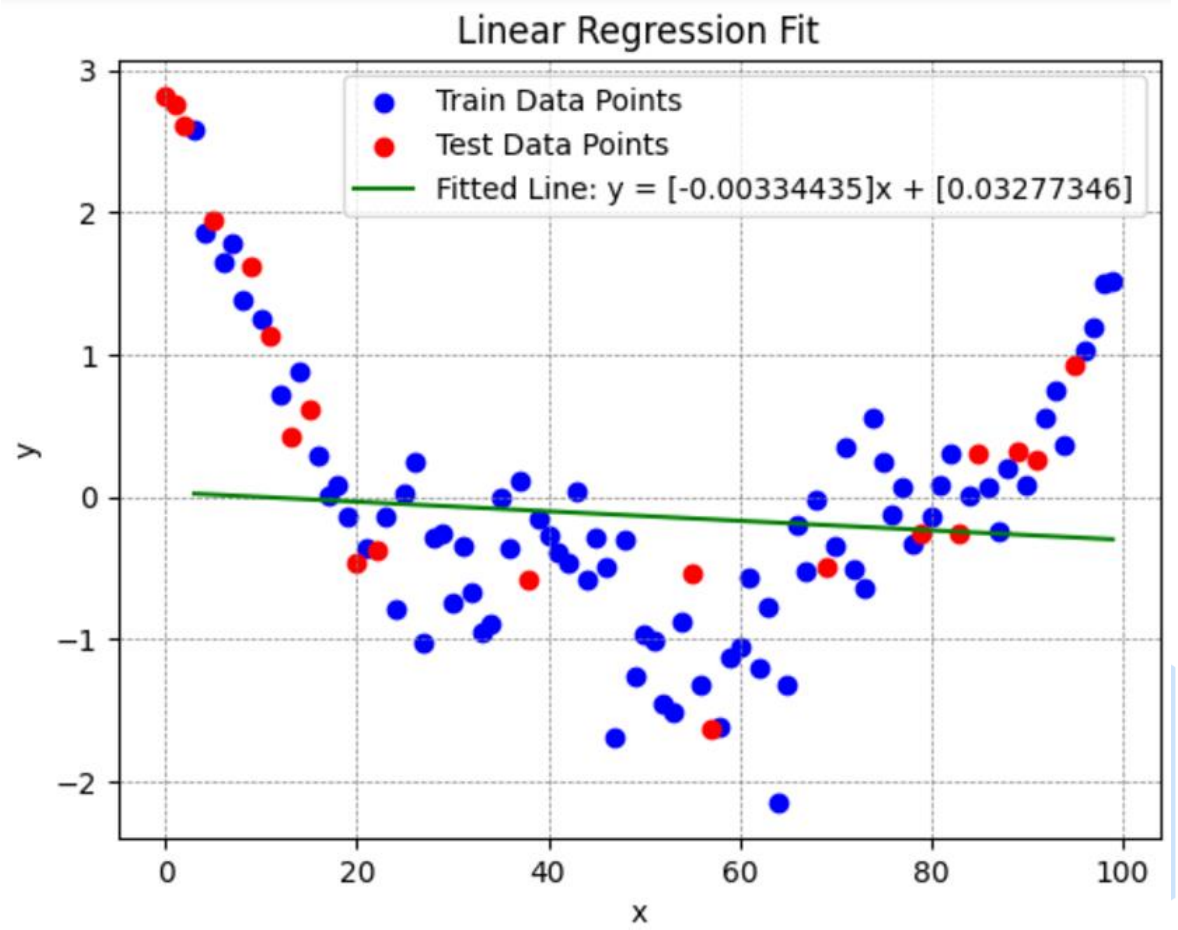
در این سوال، با استفاده از تابع گرادیان نزولی، در هر مرحله مشتق تابع هزینه نسبت به هر یک از پارامتر ها را حساب کرده، و سپس با نرخ یادگیری 0.0005، پارامتر های خود را به روز رسانی کردیم.

$$b := b - \alpha\frac{\partial J(w,b)}{\partial b}$$
$$w := w - \alpha\frac{\partial J(w,b)}{\partial w}$$

$$\frac{\partial J(w,b)}{\partial b} = \frac{1}{m}\sum_{i=0}^{m-1}(f_{w,b}(x^{(i)}) - y^{(i)})$$

$$\frac{\partial J(w,b)}{\partial w} = \frac{1}{m}\sum_{i=0}^{m-1}(f_{w,b}(x^{(i)}) - y^{(i)})x^{(i)}$$

و در نهایت شکل زیر، با ضرایب بیان شده در شکل حاصل میشود(شکل شامل نقاط train، و خط فیت شده روی این نقاط میباشد)



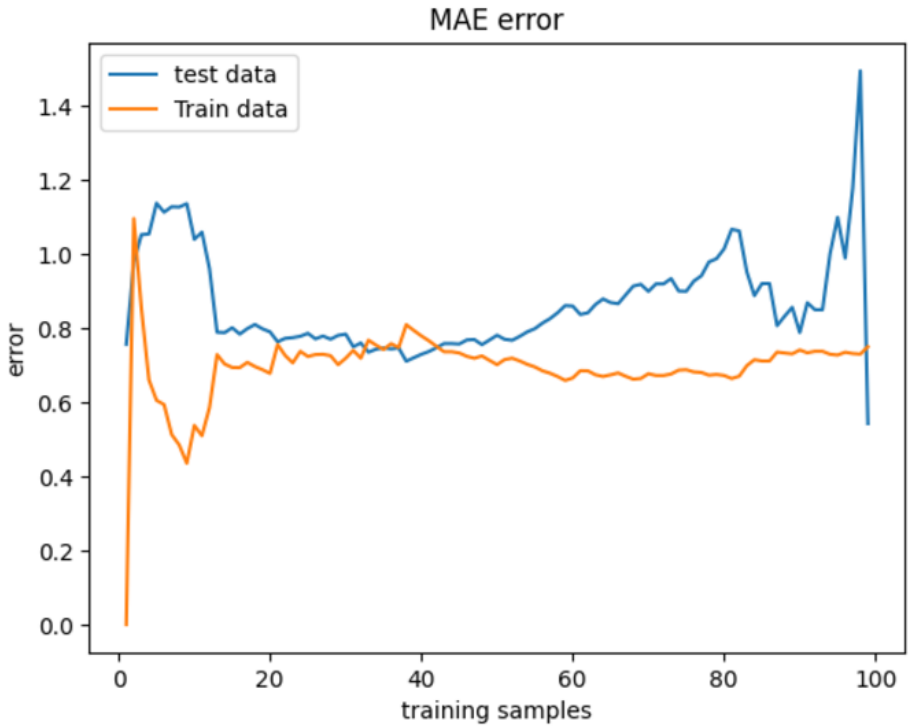
حال بر اساس سه معیار معرفی شده، عملکرد مدل را مورد بررسی قرار میدهیم:

```
MSE: 1.7744571991157332
MAE: 1.0095350590212147
R^2: -0.20065911626495003
```

مدل خطی بر پایه این فرض بنا شده است که رابطه بین متغیرهای مستقل (ویژگی‌ها) و متغیر وابسته (هدف) یک رابطه خطی است. این بدان معناست که مدل تنها قادر است خطوط مستقیم یا سطوح خطی را برای برازش داده‌ها استفاده کند. اگر داده‌ها ساختاری غیرخطی مانند یک تابع درجه دوم (مثلاً $y=ax^2+bx+cy$) داشته باشند، مدل خطی قادر نخواهد بود الگوی واقعی را به‌درستی شناسایی کند. در این حالت، پیش‌بینی‌های مدل دچار خطای زیادی خواهد شد، زیرا یک خط مستقیم نمی‌تواند انحنای ذاتی تابع درجه دو را دنبال کند.

سوال ۲.۴:

در این مرحله خواسته شده تا ۳ معیار گفته شده را، با نسبت های مختلف از تقسیم داده آموزش و تست به دست آوریم. برای این کار در یک حلقه مرحله قبل را انجام میدهیم و در هر گام، ۳ معیار مد نظر را برای داده های train و test حساب کرده و ذخیره میکنیم. خروجی به حالت زیر خواهد بود:

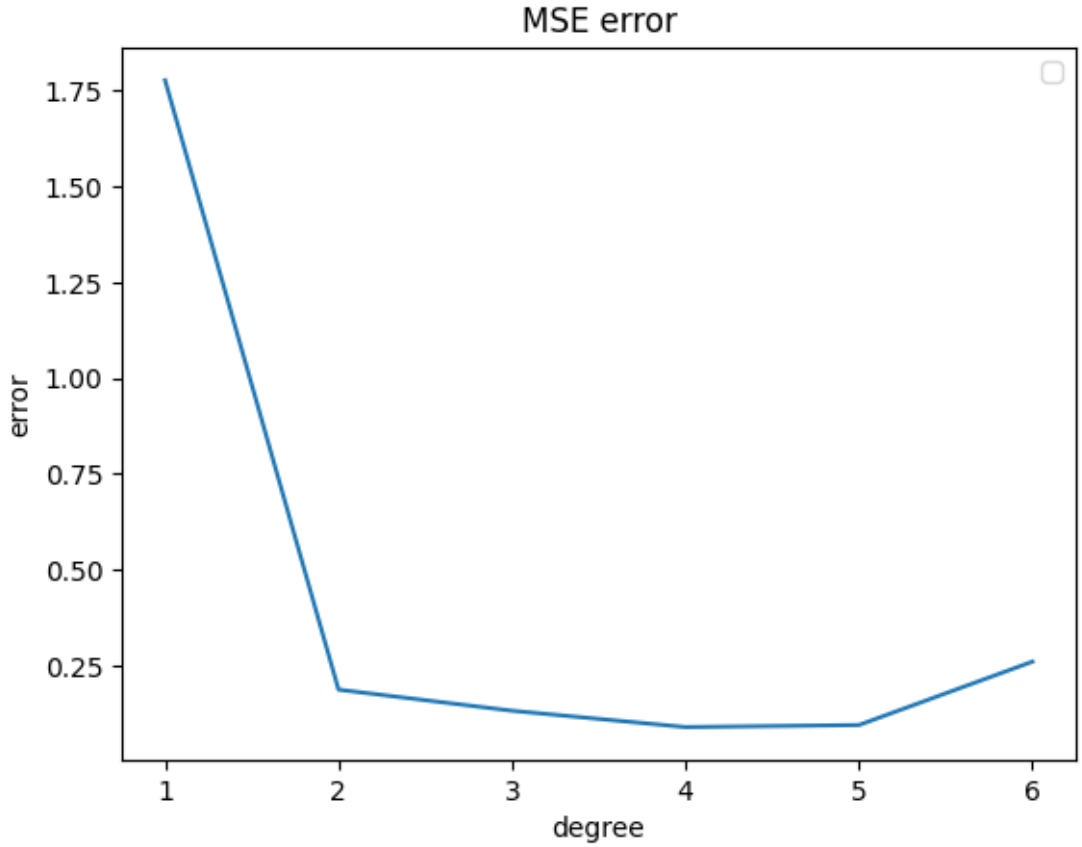


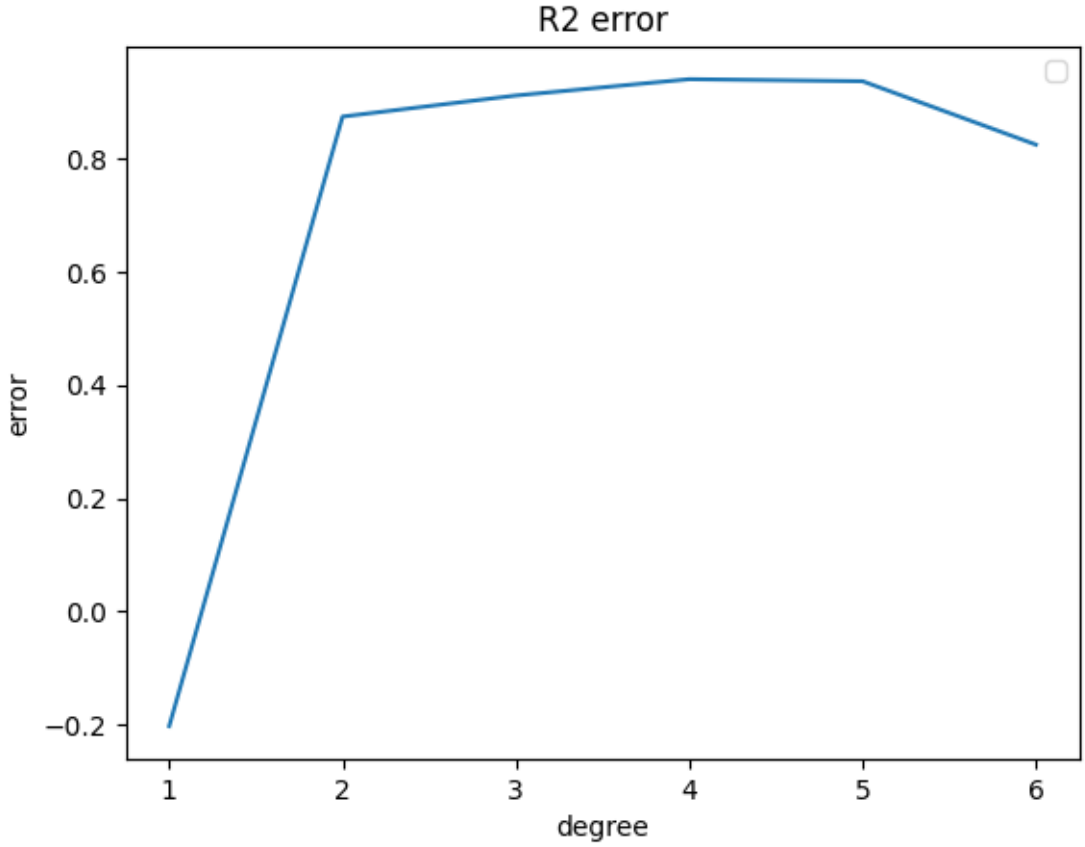
سوال ۲.۵:

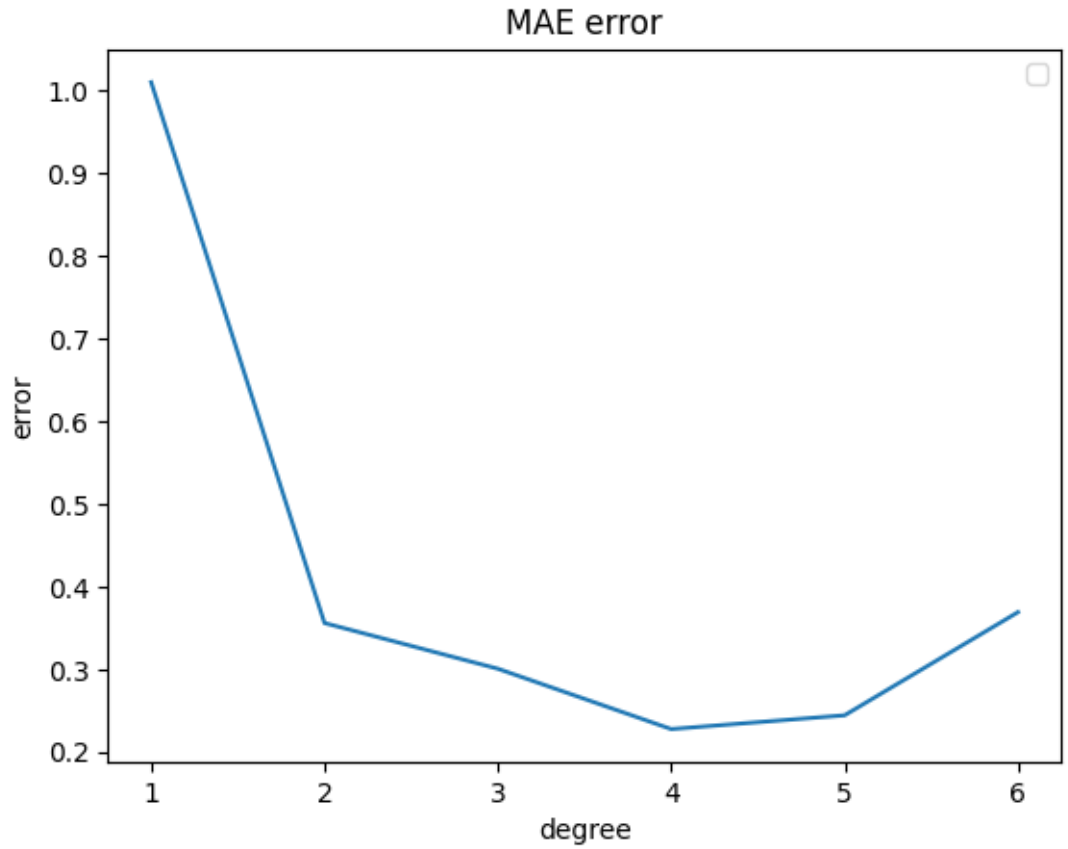
با توجه به مساله قبل، دیدیم که تا حدی افزایش داده های train به بهبود مدل شناسایی شده کمک میکند، اما این بهبود مدل نیز محدودیتی دارد و به عنوان نمونه در این مثال، یک مدل خطی هیچ گاه نمیتواند با دقت خوبی یک تابع شبیه به درجه ۲ را تخمین بزند. بنابراین برای انجام کار، انسان به علت یادگیری بهتر و عمیق تر، همیشه از مدل کامپیوتری خطای کمتری خواهد داشت اما با افزایش داده های train مدل کامپیوتری، میتوان فاصله انسان و مدل را کاهش داد. اما لازم به ذکر است که این تفاوت هیچ گاه به ۰ نخواهد رسید.

سوال ۲.۶:

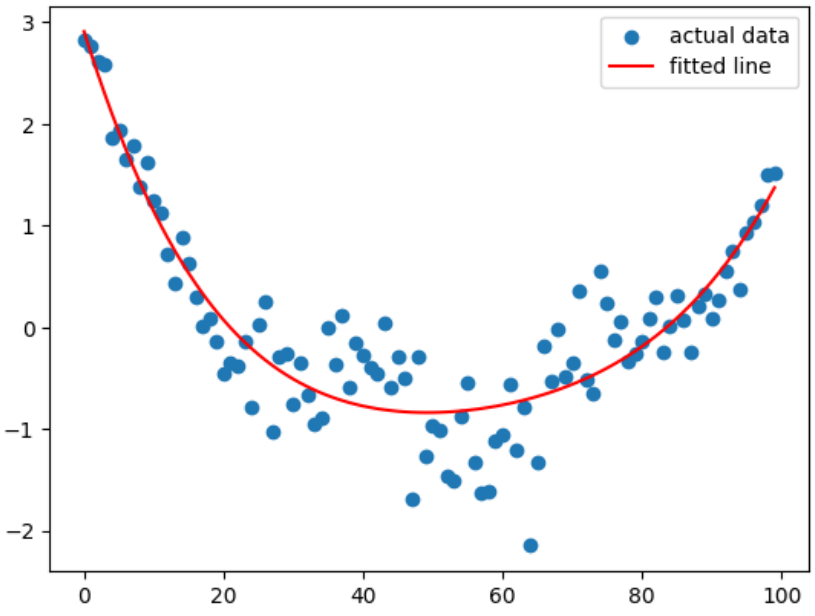
با کمک دستور PolynomialFeatures، ویژگی های تا مرتبه ۷ برای ورودی تابع رگرسیون خطی را تولید میکنیم. و نتایج خطا ها به صورت زیر خواهند شد:


MSE error
1.75
1.50
1.25
1.00
0.75
0.50
0.25
error
1
2
3
4
5
6
degree


R2 error
0.8
0.6
0.4
0.2
0.0
−0.2
error
1
2
3
4
5
6
degree


MAE error
error
degree
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
1
2
3
4
5
6

مشاهده میشود که تا حدود درجه ۴ام، مدل در تمام معیار ها به سوی بهبودی پیش میرود، اما از ۴ به بعد و درجه ۵ و ۶، با پیچیده تر شدن مدل، خطای آن نیز افزایش میابد، این امر به این دلیل است که در سیستم مبنا، مولفه درجه ۵ و ۶ وجود ندارد و مدل با تلاش خود برای فیت کردن چند جمله ای های درجه بالاتر، در واقع منجر به افزایش خطای خود میشود. در پایین، شکل چند جمله مرتبه ۴ را میبینیم:



سوال ۲.۷:

- **Linear Regression:**

رگرسیون خطی ساده‌ترین الگوریتم رگرسیون است که فرض می‌کند رابطه‌ای خطی بین متغیرهای مستقل (ویژگی‌ها) و متغیر وابسته (هدف) وجود دارد. این مدل تلاش می‌کند با حداقل کردن میانگین مربعات خطا (MSE) بین مقادیر پیش‌بینی‌شده و واقعی، یک خط مستقیم به داده‌ها تطبیق دهد. این الگوریتم برای پیش‌بینی مقادیر عددی زمانی که رابطه بین ورودی و خروجی خطی است، استفاده می‌شود.

```
MSE: 1.776908277806887
MAE: 1.0099982013452427
R^2: -0.20231760088587136
```

- **Ridge Regression**:

رگرسیون ریج یک نوع از رگرسیون خطی است که شامل یک عبارت منظم‌سازی برای کاهش پیچیدگی مدل و جلوگیری از overfitting می‌باشد. این منظم‌سازی توسط یک پارامتر α کنترل می‌شود. این روش در مواردی که مدل دچار اورفیتینگ روی داده‌های آموزشی می‌شود، کمک می‌کند و مدل را قادر می‌سازد که بهتر روی داده‌های جدید تعمیم یابد.

```
MSE: 1.7769112962634146
MAE: 1.0099988128155766
R^2: -0.20231964327796104
```

- **Random Forest Regression**:

رگرسیون جنگل تصادفی یک تکنیک یادگیری مجموعه‌ای است که مجموعه‌ای از درخت‌های تصمیم‌گیری (درخت‌ها) را ایجاد می‌کند. هر درخت بر روی زیرمجموعه‌ای از داده‌ها آموزش داده می‌شود و پیش‌بینی نهایی به‌صورت میانگین (یا رأی اکثریت) از پیش‌بینی‌های تمام درخت‌ها به دست می‌آید. جنگل تصادفی نسبت به یک درخت تصمیم‌گیری منفرد، واریانس را کاهش می‌دهد و این مدل را برای مسائل رگرسیون غیرخطی بسیار قدرتمند می‌سازد.

```
MSE: 0.10311313747096944
MAE: 0.2671814364825708
R^2: 0.9302300846856741
```

امتیازی:

رگولاریزیشن به تکنیک‌هایی اطلاق می‌شود که در یادگیری ماشین و مدل‌های آماری برای جلوگیری از بیش‌برازش (Overfitting) و بهبود تعمیم‌پذیری مدل به داده‌های جدید استفاده می‌شود. بیش‌برازش زمانی رخ می‌دهد که مدل نه تنها الگوهای زیرین داده‌ها را یاد می‌گیرد، بلکه نویز و داده‌های غیرطبیعی (outliers) را نیز یاد می‌گیرد، که منجر به عملکرد ضعیف مدل بر روی داده‌های جدید و دیده‌نشده می‌شود. رگولاریزیشن با افزودن یک عبارت جریمه به تابع زیان مدل، از پیچیده شدن بیش از حد مدل جلوگیری می‌کند و به مدل کمک می‌کند تا ساده‌تر و مقاوم‌تر شود.

رگولاریزیشن کمک می‌کند تا مدل ساده‌تر و مقاوم‌تر باشد، با کنترل اینکه مدل چقدر می‌تواند داده‌های آموزشی را فیت کند. دو نوع اصلی رگولاریزیشن در مدل‌های خطی، رگولاریزیشن L1 (Lasso)و رگولاریزیشن L2 (Ridge) وجود دارد.

۱- رگولاریزیشن L1

رگولاریزیشن L1 یک پنالتی برابر با مقدار مطلق ضرایب اضافه می‌کند. این باعث می‌شود که مدل به سمت اسپارسیتی (sparsity) میل کند، به این معنا که برخی از ضرایب به طور دقیق صفر می‌شوند. در نتیجه، Lassoمی‌تواند انتخاب ویژگی انجام دهد و ویژگی‌های غیرمهم را از مدل حذف کند.

$$\text{تابع هزینه} = \text{میانگین خطای مربعات} + \lambda_i \sum_{i=1}^{n} |\beta|$$

۲- رگولاریزیشن L2

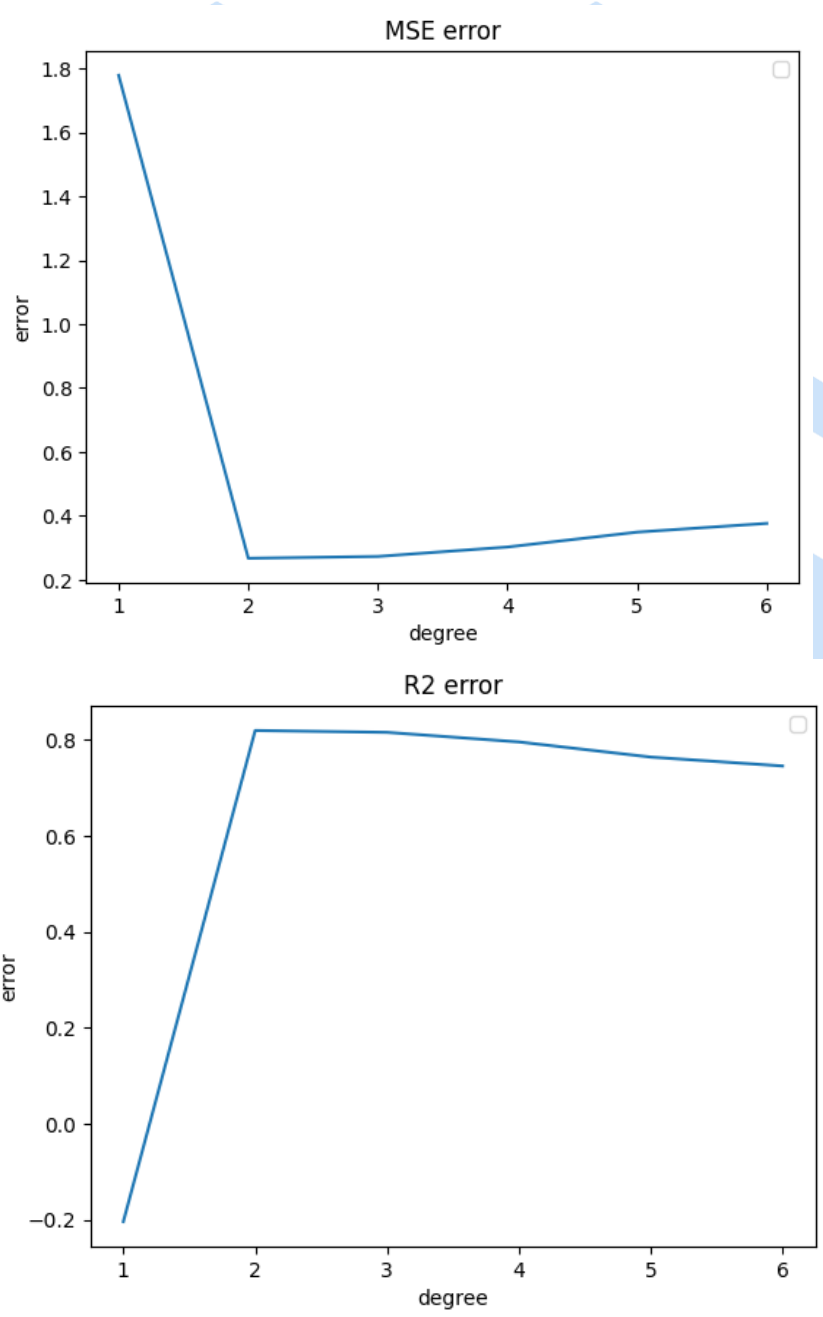
رگولاریزیشن L2 یک جریمه برابر با مربع اندازه ضرایب اضافه می‌کند. این باعث می‌شود که مدل از ضرایب بزرگ جلوگیری کند، اما لزوماً آنها را صفر نمی‌کند، برخلاف .Lasso رگرسیون Ridge تمایل دارد ضرایب را به طور یکنواخت کاهش دهد، که مدل را پایدارتر می‌کند، اما مدل را کاملاً ساده نمی‌کند.

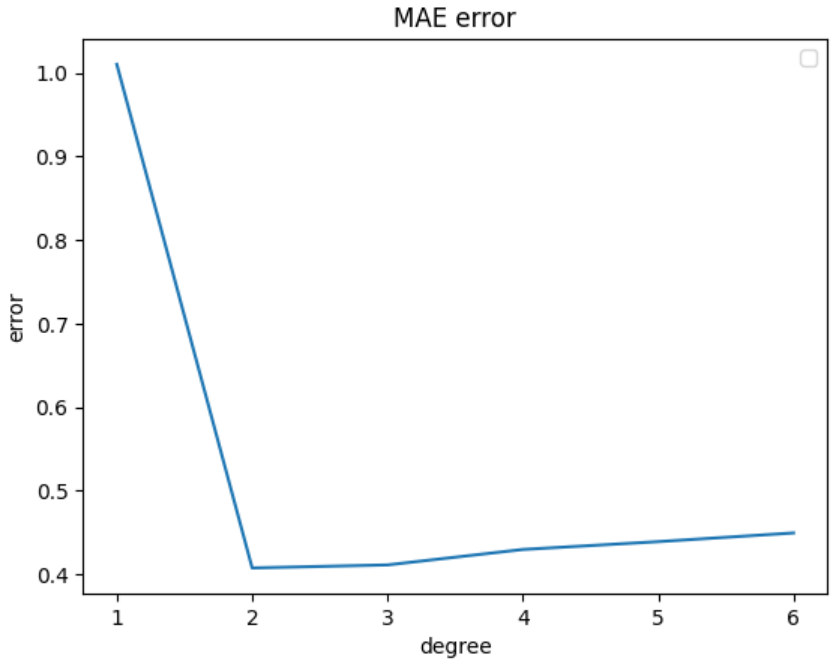
$$\text{تابع هزینه} = \text{میانگین خطای مربعات} + \lambda_i^2 \sum_{i=1}^{n} \beta$$

با استفاده از رگولاریزیشن L2 یا همان استفاده از الگوریتم Ridge، و آلفا برابر با ۱، حاصل به شکل زیر تبدیل خواهد شد(لازم به ذکر است که در کد، مقدار عرض از مبدا برابر با ۰ درنظر گرفته شده است):

```
MSE: 1.81048896018509
MAE: 1.0162652014670623
R^2: -0.22503945207950915
```

حاصل به دست آمده، بدون در نظر گرفتن intercept میباشد، در مرحله بعدی، عرض از مبدا را اضافه کرده و بعد از آن، در مراحل بعدی، به تدریج پیچیدگی مدل را بیشتر میکنیم:(حاصل روی نمودار در مرحله ۱، با در نظر گرفتن intercept میباشد)





این نمودار نشان می‌دهد که خطای مدل (MAE) در درجه یک بسیار بالا است، که نشان‌دهنده ناکافی بودن مدل خطی برای تطبیق داده‌ها است. با افزایش درجه به ۲، خطا به‌طور قابل‌توجهی کاهش می‌یابد، که نشان می‌دهد رابطه بین ویژگی‌ها و هدف به احتمال زیاد درجه دوم (Quadratic) است. برای درجات بالاتر (۳ به بالا)، تغییرات خطا ناچیز است و حتی کمی افزایش می‌یابد، که احتمالاً به دلیل بیش‌برازش (Overfitting) است. به‌طور کلی، درجه ۲ به‌عنوان بهترین درجه برای مدل در این داده‌ها به نظر می‌رسد، زیرا تعادل مناسبی بین خطا و پیچیدگی مدل ایجاد می‌کند.