Mini Project 1

RAMIL KHOSRAVI – 40004883

Link to Google Colab

Link to GitHub

(1.1

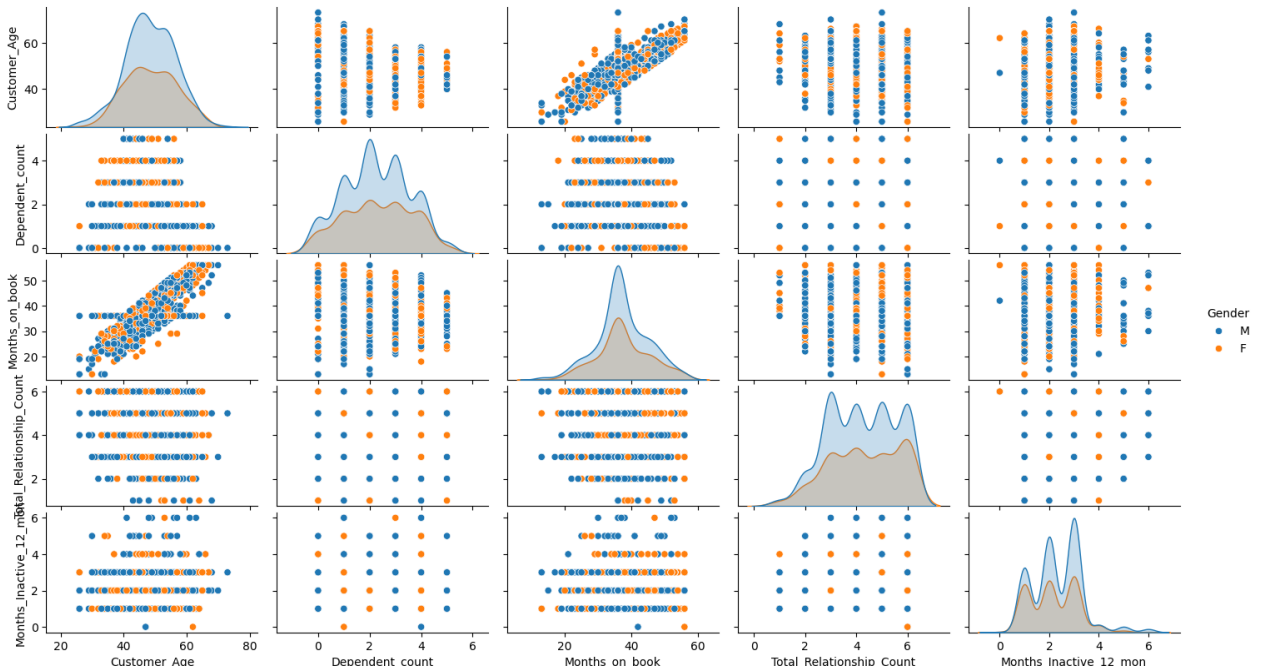
این داده‌ها مربوط به مشتریان یک بانک می‌باشند که به ازای هر شماره مشتری (CLIENTNUM) مشخصاتی از قبیل اینکه در حال حاضر مشتری هستند یا خیر، جنسیت، سن، درآمد و ... در آن آمده است.

ویژگی‌ها:

سن، جنسیت، سطح تحصیل، مجرد یا متاهل بودن، حدود درآمد، گردش مالی و ... .

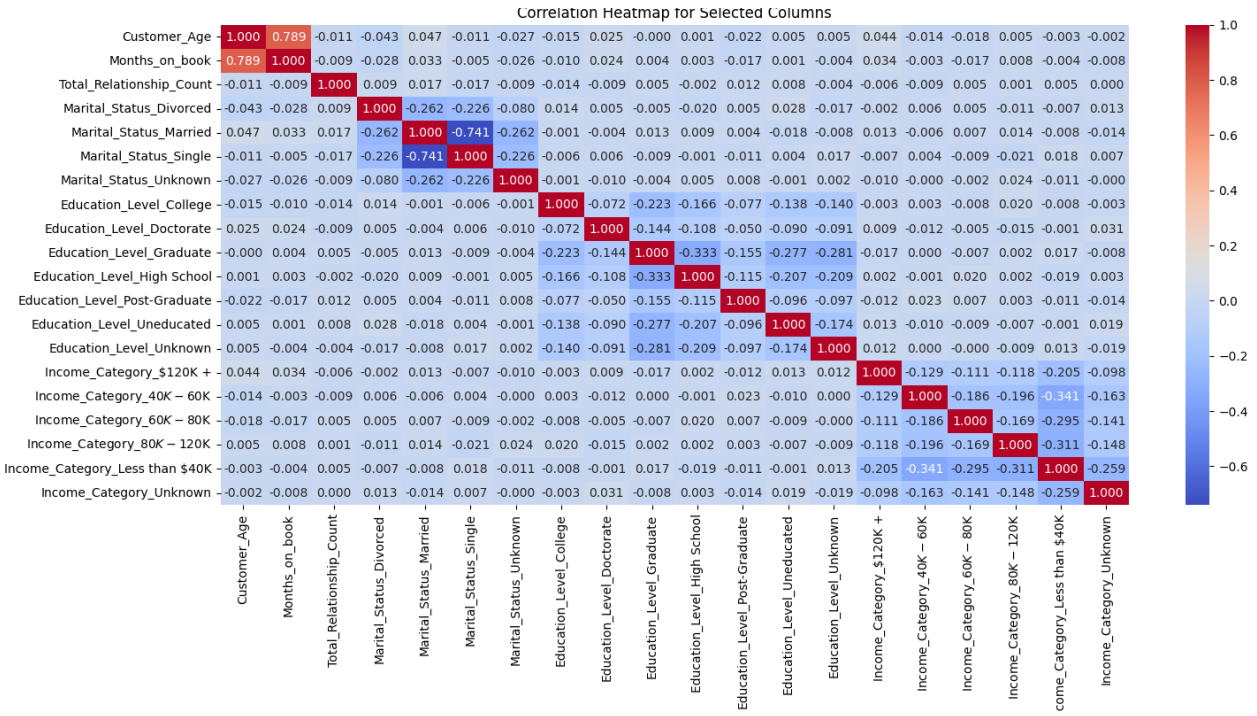
تعداد نمونه: ۱۰۱۲۸.

(1.2




```python
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

#1.2---------------------------------------------------------------------------------
dataSet = pd.read_csv("BankChurners.csv")

sns.pairplot(dataSet.iloc[0:1000:, 1:12], hue = "Gender")
plt.show()

#-----------------------------------------------------------------------------------
```

(1.3




```python
#1.3-------------------------------------------------------------------------------
HMdataSet = dataSet[['Marital_Status', 'Education_Level', 'Income_Category',
                     'Customer_Age', 'Months_on_book', 'Total_Relationship_Count']]

# Assigning a value to the categorial data
HMdataSet = pd.get_dummies(HMdataSet)

corrMat = HMdataSet.corr()

sns.heatmap(corrMat, annot=True, cmap="coolwarm", fmt=".3f")
plt.title("Correlation Heatmap for Selected Columns")
plt.show()
```

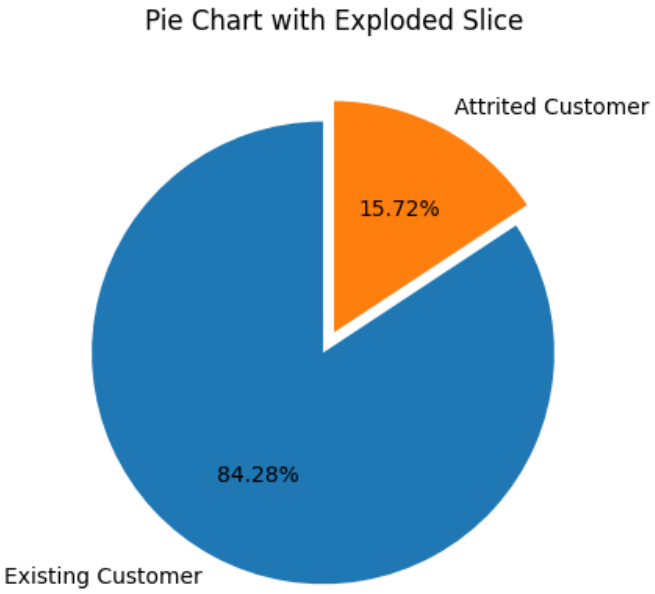
(1.4

برای ۳ کلاس همانطور که در شکل زیر مشاهده می‌شود، داده‌های Unknown داریم که آن‌ها را حذف می‌کنیم.

```
Education_Level
                  1519
Marital_Status
                   749
Income_Category
                  1112
```

(1.5

داده‌های موجود در ویژگی Attrition Flag دارای ۲ کلاس Attrited Customer و Existing Customer است.



بله، عدم تعادل در داده‌ها (Imbalanced Data) می‌تواند تأثیر قابل‌توجهی بر عملکرد و یادگیری مدل‌های یادگیری ماشین داشته باشد، به ویژه در مسائل دسته‌بندی (Classification). این وضعیت زمانی رخ می‌دهد که توزیع داده‌ها بین دسته‌ها به شدت نابرابر باشد، مثلاً یکی از دسته‌ها تعداد نمونه‌های بسیار بیشتری نسبت به دسته‌های دیگر داشته باشد.

تأثیر عدم تعادل در داده‌ها

گرایش مدل به اکثریت: در صورت عدم تعادل، مدل تمایل پیدا می‌کند که پیش‌بینی‌های خود را به سمت کلاس اکثریت سوق دهد؛ زیرا با پیش‌بینی اکثریت، به‌طور نسبی به دقت بالایی دست می‌یابد ولی دسته اقلیت نادیده گرفته می‌شود. در نتیجه آنها را به درستی یاد نمی‌گیرد؛ زیرا سهم آنها در تابع هزینه کمتر است.

معیارهای گمراه‌کننده: معیارهایی مانند دقت (Accuracy) در این شرایط گمراه‌کننده می‌شوند. مثلاً اگر ۹۵٪ داده‌ها مربوط به کلاس اکثریت باشند، مدل می‌تواند با پیش‌بینی تمام نمونه‌ها به کلاس اکثریت، دقت ۹۵٪ کسب کند، اما عملکرد واقعی برای دسته اقلیت ضعیف است.

تأثیر بر تعمیم‌دهی مدل: مدل ممکن است برای دسته‌های اقلیت تعمیم‌پذیری خوبی نداشته باشد و در دنیای واقعی (مثلاً تشخیص تقلب یا پیش‌بینی بیماری‌های نادر) عملکرد ضعیفی داشته باشد.

راهکارها برای مقابله با عدم تعادل داده‌ها

1 . روش‌های پیش‌پردازش داده

- **Oversampling (افزایش نمونه‌های کلاس اقلیت):** تعداد نمونه‌های کلاس اقلیت را از طریق تکثیر داده‌ها یا استفاده از روش‌های مصنوعی مانند SMOTE (Synthetic Minority Oversampling Technique) افزایش می‌دهیم.
- **Undersampling (کاهش نمونه‌های کلاس اکثریت):** تعداد نمونه‌های کلاس اکثریت را با حذف بخشی از آنها کاهش می‌دهیم. این روش در صورتی مفید است که داده‌های کافی برای مدل باقی بماند.
- **ایجاد داده‌های مصنوعی (Synthetic Data Generation):** تولید داده‌های جدید برای کلاس اقلیت از طریق تکنیک‌های پیشرفته.

2 . استفاده از مدل‌ها و الگوریتم‌های مقاوم در برابر عدم تعادل

- **وزن‌دهی به کلاس‌ها**: الگوریتم‌های یادگیری ماشین مانند Logistic Regression، SVM و Tree-Based Models از گزینه‌های وزن‌دهی برای کلاس‌های نامتعادل پشتیبانی می‌کنند. این کار باعث می‌شود که مدل به نمونه‌های کلاس اقلیت وزن بیشتری بدهد.
- **استفاده از الگوریتم‌های خاص**: الگوریتم‌هایی مانند BalancedBaggingClassifier یا BalancedRandomForestClassifier که به‌طور خاص برای داده‌های نامتعادل طراحی شده‌اند.

برای متعادل سازی داده هم می‌توان آن را قبل از تقسیم بندی داده به بخش‌های آموزش و آزمون انجام داد و هم بعد از تقسیم بندی.

1 . متعادل‌سازی قبل از تقسیم‌بندی (Pre-Split Balancing):

- **مزایا:** اگر کل مجموعه داده را پیش از تقسیم متعادل کنید، به طور مساوی نمونه‌های اقلیت و اکثریت را در داده‌های آموزش و آزمون توزیع می‌کنید. این روش برای تحلیل داده‌ها یا بررسی آمارهای کلی روی یک مجموعه متعادل‌شده مفید است.
- **معایب:** نشت اطلاعات (Data Leakage):

هنگام انجام oversampling (مانند SMOTE یا کپی نمونه‌های اقلیت) قبل از تقسیم داده‌ها، ممکن است نمونه‌های تولیدشده به هر دو مجموعه آموزش و آزمون راه پیدا کنند. این وضعیت باعث می‌شود مدل در آزمون داده‌هایی را ببیند که قبلاً در آموزش یاد گرفته است، و این باعث می‌شود که نتایج آزمون به طور مصنوعی خوب به نظر برسد.

2 .متعادل‌سازی بعد از تقسیم‌بندی (Post-Split Balancing):

- از نشت اطلاعات جلوگیری می‌کند؛ داده‌های تولیدشده برای متعادل‌سازی فقط روی مجموعه آموزشی (Train) اعمال می‌شوند و مجموعه آزمون (Test) دست‌نخورده باقی می‌ماند. این امر باعث می‌شود عملکرد مدل روی داده‌های ناآشنا ارزیابی شود. مجموعه آزمون همیشه نمایشی دقیق از داده‌های واقعی باقی می‌ماند.

(1.6

در این قسمت با استفاده از روش oversampling که در بخش قبل توضیح داده شد، داده‌ها را متعادل می‌کنیم.

بدون متعادل کردن داده‌ها و با استفاده از مدل Gradient Boosting نتایج زیر حاصل می‌شوند:

```
Unbalanced Confusion Matrix:

 [[ 170   42]
 [  20 1185]]

Performance Report:

                   precision    recall  f1-score   support

Attrited Customer       0.89      0.80      0.85       212
Existing Customer       0.97      0.98      0.97      1205

         accuracy                           0.96      1417
        macro avg       0.93      0.89      0.91      1417
     weighted avg       0.96      0.96      0.96      1417
```

با توجه به Confusion matrix مشاهده می‌شود که ۲۰ مورد از کلاس Existing customer به اشتباه Attrited customer تشخیص داده شده و ۴۲ مورد هم از کلاس Attrited customer به اشتباه Existing customer تشخیص داده شده‌اند. این مورد به دلیل اینکه داده‌های کلاس Existing customer نسبت به کلاس دیگر بیشتر هستند، رخ داده است و مدل یادگیری خود را بیشتر به سمت کلاس با داده‌های بیشتر سوق داده است. دقت در این حالت ۰.۹۶ می‌باشد که دقت خوبی به نظر می‌رسد.

```
Balanced Confusion Matrix:

 [[ 191   21]
 [  62 1143]]

Performance Report:

                   precision    recall  f1-score   support

Attrited Customer       0.75      0.90      0.82       212
Existing Customer       0.98      0.95      0.96      1205

         accuracy                           0.94      1417
        macro avg       0.87      0.92      0.89      1417
     weighted avg       0.95      0.94      0.94      1417
```

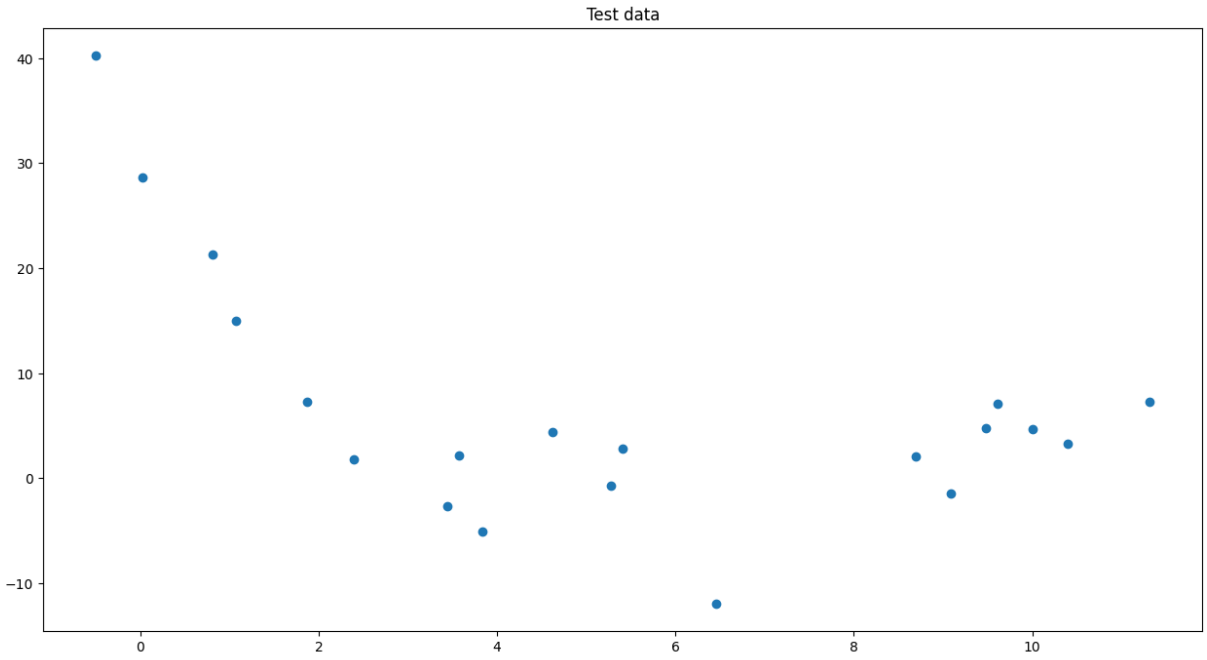
بعد از متعادل کردن داده‌ها مشاهده می‌شود که ۲۰ مورد اشتباه قبل به ۶۲ افزایش یافته است ولی ۴۲ مورد اشتباه دیگر به ۲۱ کاهش یافته است. دقت هم ۰.۹۴ است که ۲ درصد کاهش داشته است. نکته مهم در اینجا این است که مدل، کلاس Attrited customer را با داده‌های متعادل شده بهتر تشخیص می‌دهد. در حالت قبلی ۴۲ مورد را از این کلاس اشتباه تشخیص می‌دهد که تقریبا ۲۰ درصد کل موارد است که مقدار چشم گیری است. برای کلاس دیگر هم تقریبا ۱.۶ درصد موارد را اشتباه تشخیص داده است.

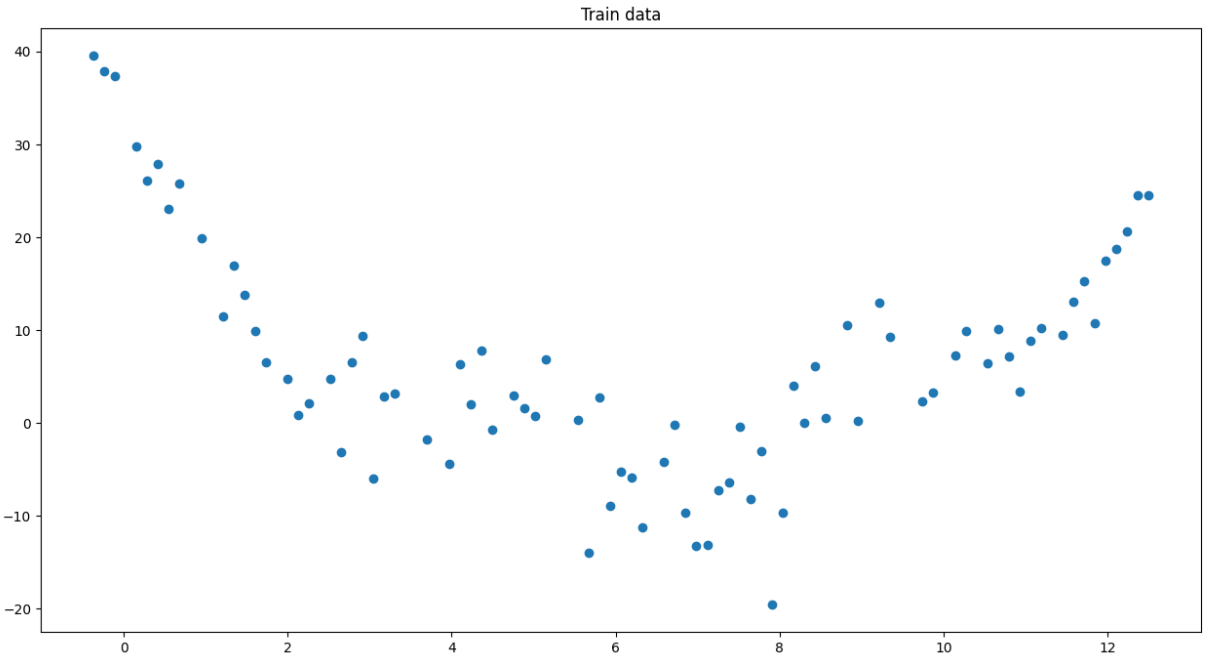
با داده‌های متعادل شده، موارد اشتباه برای کلاس Attrited customer، به ۱۰ درصد کاهش یافته است که یعنی نصف مقدار حالت قبل. برای کلاس دیگر هم تقریبا ۵ درصد موارد اشتباه تشخیص داده شده‌اند. اگر نسبت‌های موارد اشتباه را در نظر بگیریم به نظر می‌آید که مدل به تعادل بهتری بعد از متعادل کردن داده‌ها رسیده است. همانطور که در گزارش دیده می‌شود ۱۰ درصد بهبود در recall (تشخیص true positive) برای

کلاس Attrited customer داریم. در حالی که تنها ۳ درصد کاهش در recall برای کلاس Existing customer داریم.

(2.1

نمودار داده‌های Test و Train به صورت زیر می‌باشد:





(2.2

۱. میانگین مربعات خطا (Mean Squared Error - MSE):

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

این معیار میانگین مجذور تفاوت بین مقادیر واقعی و پیش بینی شده را اندازه گیری می‌کند. چون اختلاف‌ها به توان ۲ می‌رسند، خطاهای بزرگ تأثیر بیشتری بر نتیجه دارند. کاهش مقدار MSE به معنای بهبود پیش‌بینی مدل است.

۲. ضریب تعیین (R^2) یا R^2-Score:

$$R^2 = 1 - \frac{\sum_{1=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{1=1}^{N}(y_i - \bar{y})^2} \qquad R^2 = 1 - \frac{SS_{Regression}}{SS_{Total}}$$

مقدار این معیار هموار مقداری بین ۰ و ۱ است. اگر نتیجه آن برابر با ۱ باشد یعنی مدل تمام واریانس داده را توضیح داده است (عملکرد عالی) و اگر نتیجه آن ۰ باشد یعنی مدل هیچ توضیحی از واریانس داده‌ها ارائه نکرده است. همچنین مقادیر منفی ممکن است در صورت عملکرد بسیار ضعیف مدل رخ دهد.

۳. میانگین قدر مطلق خطا (Mean Absolute Error - MAE):

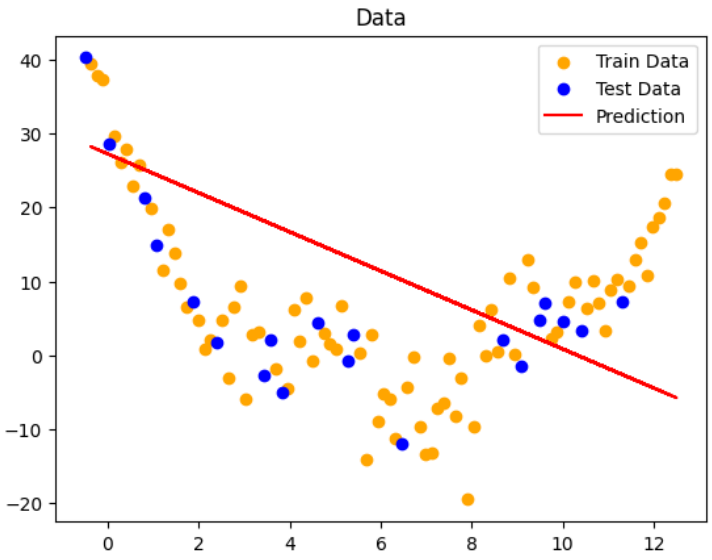
$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

این معیار میانگین مقدار مطلق خطاها را اندازه‌گیری می‌کند، بدون آنکه خطاها را به توان ۲ برساند. این معیار نسبت به مقادیر پرت (outliers) حساسیت کمتری دارد. کاهش MAE به معنای کاهش میانگین اختلاف بین مقادیر واقعی و پیش‌بینی شده است.

اگر MAE را نرمالیزه کنیم و به صورت درصد بنویسیم، به MAPE (Mean Absolute Percentage Error) تبدیل می‌شود که به دلیل اینکه یک عدد نسبی است، می‌توان گفت اگر زیر ۱۰ درصد باشد، مدل خطای قابل قبولی دارد.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \times 100$$

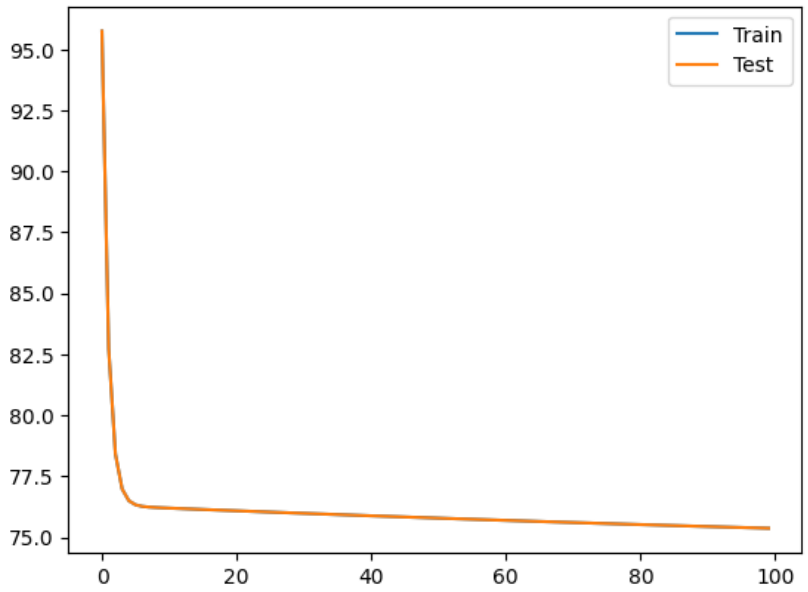
(2.3




```
MSE = 262.46412412520067
MAPE = 204.63138383231404
R2 = -0.7815365346108316
MAE = 13.340774851790457
```

همانطور که مشاهده می‌شود و با توجه به مقادیر خطا، رگرسور خطی برای این داده‌ها مناسب نیست.

(2.4



همانطور که از نمودار تابع هزینه معلوم است، با افزایش تعداد داده‌های تمرین خطا کاهش می‌یابد ولی بعد از تعداد خاصی از داده‌ی تمرین، شیب نمودار بسیار کم می‌شود و تقریبا به مقدار مشخصی همگرا می‌شود.

(2.5

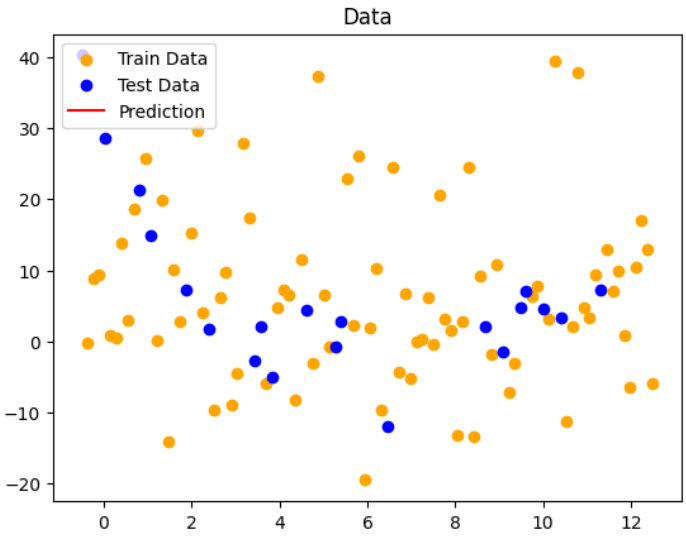
هر مسئله‌ای یک حداقل خطا دارد که به ذات داده و نویز موجود در آن بستگی دارد. اگر خطای انسانی ۱ باشد، احتمالاً این خطا نشان‌دهنده حداقل خطای ممکن **(Bayes Error Rate)** است. اگر خطای ذاتی داده بیشتر از ۱ باشد، حتی با داده‌های بیشتر یا مدل قوی‌تر هم نمی‌توان از این حد پایین‌تر رفت.

اگر مدل فعلی یادگیری ماشین خطای آموزش ۱۰ دارد، احتمالاً مدل به اندازه کافی پیچیده نیست یا به درستی طراحی نشده است؛ ممکن است مدل ظرفیت کافی برای یادگیری الگوهای موجود در داده را نداشته باشد **(Underfitting)**. با استفاده از مدل‌های پیچیده‌تر، احتمال کاهش خطا وجود دارد، اما این کار به داده‌های بیشتر و تنظیم دقیق مدل نیاز دارد.

خطای بالای آموزش ممکن است ناشی از تنظیمات نامناسب مدل باشد. تنظیم دقیق فوق پارامترها **(Hyperparameters)** مثل نرخ یادگیری، تعداد لایه‌ها یا نرون‌ها می‌تواند به بهبود مدل کمک کند. اما حتی با داده‌های بیشتر، مدل همچنان به تنظیمات دقیق نیاز دارد.

(2.6

برای این قسمت بدون normalize کردن داده‌ها، المان‌های لیست y_pred، مقادیر خیلی بزرگی می‌گیرند و Overflow می‌کنند. در نتیجه هیچ نمودار رگرسوری در خروجی نمی‌بینیم. برای حل این مشکل یا باید داده‌ها را normalize کنیم یا اینکه باید تعداد iteration را کم کنیم که همین باعث می‌شود رگرسور ما واگرا شود.



(2.7

رگرسیون خطی (Linear Regression):

رگرسیون خطی ارتباط بین ویژگی‌ها و متغیر هدف را با قرار دادن یک خط مستقیم (یا یک صفحه در حالت چندبعدی) مدل می‌کند. این مدل مقدار هدف را بر اساس معادله زیر پیش‌بینی می‌کند:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + x_n \beta_n$$

مزیت اصلی رگرسیون خطی سادگی آن است و به صورت محاسباتی بسیار سریع است. با این حال، اگر رابطه بین متغیرها غیرخطی باشد یا ویژگی‌ها همبستگی زیادی با یکدیگر داشته باشند، عملکرد آن ضعیف خواهد بود.

:Decision Tree Regressor

درخت تصمیم داده‌ها را بر اساس مقادیر ویژگی‌ها به بخش‌های مختلف تقسیم می‌کند و قوانین تصمیم‌گیری ایجاد می‌کند که ساختاری به شکل درخت تولید می‌شود. هر برگ در این درخت نشان‌دهنده یک ناحیه است که مقدار پیش‌بینی‌شده آن برابر میانگین مقادیر هدف در آن ناحیه است.

این مدل برای داده‌هایی که روابط غیرخطی بین ویژگی‌ها و هدف دارند بسیار مفید است و می‌تواند تعاملات پیچیده بین ویژگی‌ها را شناسایی کند. از مزایای آن می‌توان به قابلیت تفسیر آسان (به‌ویژه به‌صورت بصری) اشاره کرد. اما یکی از معایب آن این است که اگر درخت خیلی عمیق شود یا داده‌ها محدود باشند، مدل Overfitting خواهد کرد.

:Random Forest Regressor

Random Forest یک روش تجمعی (Ensemble) است که از ترکیب چندین درخت تصمیم ساخته می‌شود. هر درخت بر روی یک زیرمجموعه تصادفی از داده‌ها و ویژگی‌ها آموزش می‌بیند. پیش‌بینی نهایی از ترکیب پیش‌بینی‌های تمام درخت‌ها (مثلاً به‌صورت میانگین) به دست می‌آید که این کار باعث کاهش Overfitting می‌شود.

این مدل برای داده‌هایی که دارای روابط غیرخطی، نویز زیاد یا تعداد زیادی ویژگی هستند، بسیار مؤثر است. جنگل تصادفی نسبت به درخت تصمیم، عملکرد پایدارتری دارد و به خوبی از Overfitting جلوگیری می‌کند. با این حال، این مدل از لحاظ محاسباتی سنگین‌تر است و نسبت به درخت تصمیم، کمتر قابل تفسیر است.

نتایج مدل‌ها

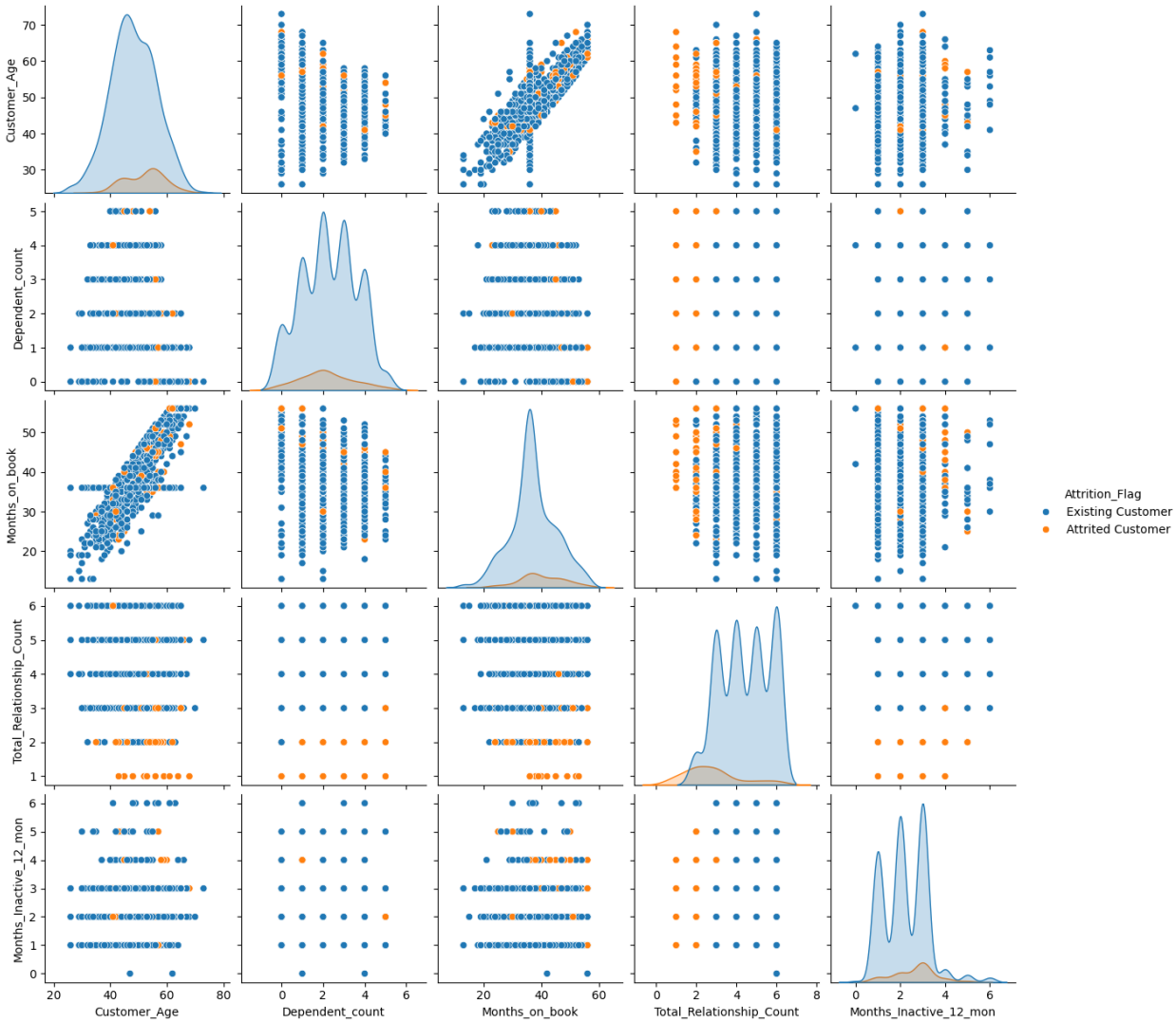
```
Errors for Linear Rergressor:
MSE = 110.59587101210086
R2 = 0.17829444391532445
MAE = 7.653499250826206

Errors for Decision Tree Regressor:
MSE = 16.378318346900286
R2 = 0.8783123179752432
MAE = 3.3553734487304894

Errors for Random Forest Regressor:
MSE = 11.409116190593874
R2 = 0.9152325120455821
MAE = 2.7610890230586866
```

با توجه به توضیحات مدل‌ها همانطور که انتظار داشتیم Linear Regressor بیش ترین خطا و Random Forest Regressor کم ترین خطا را در بین ۳ مدل داشتند.

خطای زیاد Linear Regressor به دلیل خطی نبودن داده‌ها است و به همین دلیل است که با یک معادله درجه ۱ نمی‌توان داده‌ها را به خوبی پیش بینی کرد.

تفاوت اندک بین خطاهای Decision Tree و Random Forest هم با توجه به توضیحات مدل‌ها، به دلیل پایداری و پیچیدگی بالای Random Forest نسبت به Decision Tree است.

امتیازی۱)



مشاهده می‌شود که پخش داده برای کلاس Attrited Customer بسیار کم تر از کلاس Existing Customer است.

امتیازی۲)

بعد از normalize کردن داده‌ها نمودار به شکل زیر خواهد بود: