دانشگاه اصفهان – دانشکده مهندسی کامپیوتر

مبانی یادگیری ماشین

استاد : دکتر کیانی

پیش‌بینی محدودیت اعتبار با استفاده از مدل رگرسیون

شیدا عابدپور

۴۰۰۳۶۲۳۰۲۵

بهار ۱۴۰۳

پیش‌پردازش:

در ابتدا پیش از پردازش داده‌ها توسط مدل لازم است پیش‌پردازش‌هایی صورت گیرد.

۱. در اولین مرحله ستون‌های نامرتبط و غیرموثر در پیش‌بینی درست لازم است حذف شوند. در دیتاست مربوطه، ستون CLIENTNUM برای هر مشتری خاص و غیرتکراری است و نمیتواند به عنوان ویژگی در نظر گرفته شود.

۲. در گام دوم، داده‌های تکراری از دیتاست پاک می‌شوند، زیرا ممکن است باعث انحراف در تحلیل‌های آماری شوند.

۳. همچنین جهت اطمینان لازم است مقادیر موجود در ویژگی‌های دسته‌ای چک شوند تا کارکتر نامربوطی به اشتباه وارد نشده باشد.

۴. گام بعدی در پیش‌پردازش، تبدیل ویژگی‌های دسته‌ای به عددی است، زیرا مدل‌های رگرسیون قادر به پردازش داده‌های غیرعددی نیستند. این مرحله لازم است پیش از جداسازی داده‌های تست صورت گیرد، زیرا ممکن است در جداسازی داده‌ها ویژگی‌های دسته‌ای با تکرار پایین در تست قرار نگرفته باشند و باعث مشکل در encoding شوند.

۵. پس از encoding ، باید داده‌های ترین و تست از یکدیگر جدا شوند تا پیش‌پردازش‌های آتی روی هرکدام جداگانه صورت گیرد. در جداسازی ترین و تست باید به این نکته توجه کرد که جداسازی باید به گونه‌ای باشد که احتمال ایجاد بایاس در داده‌ها کمتر شود و توزیع آن‌ها به یکدیگر نزدیک باشد(نباید انتظار داشت مدل قادر به پیش‌بینی چیزی باشد که آموزش ندیده است).

۶. پس از جداسازی ترین و تست، در اولین گام باید ویژگی‌های مفقود در داده‌های ترین و تست را پر کرد، زیرا مدل‌های رگرسیون نمی‌توانند داده‌های مفقود را مدیریت کنند.

۷. پس از کامل شدن داده‌ها، لازم است تا داده‌های پرت حذف شوند. وجود داده‌های پرت باعث ایجاد مشکل در پیش‌بینی مدل شده و احتمال overfit را بیشتر می‌کنند. داده‌های پرت می‌توانند به این علت باشند که کاربر اطلاعات را سهواً اشتباه وارد کرده است و یا ممکن است با توجه به زمینه یا موقعیتی خاص غیرعادی باشند.

۸. پس از تشخیص و مدیریت داده‌های پرت در داده‌های ترین، با توجه به میزان اهمیت و وابستگی یک ویژگی به ویژگی‌های دیگر، می‌توان یکسری از ویژگی‌های را حذف کرد و یا با ترکیب آن‌ها ویژگی‌های جدیدی ساخت.

۹. پس از تکمیل مراحل لازم، پیش از پردازش، لازم است داده‌های x_train و x_test نرمالایز شوند، زیرا ویژگی‌های مختلف در بازه‌های مختلفی هستند. نیازی به نرمالایز کردن برچسب نیست، یک متغیر نامحدود و پیوسته است که رنج آن در تست نامشخص است و اسکیل کردن آن بی‌معناست.

نکات تکمیلی در پیش‌پردازش داده‌ها:

- جهت encoing داده‌های دسته‌ای، می‌توان به ویژگی‌های ترتیبی با توجه به مفهوم ترتیب موجود در آن‌ها عدد تخصیص داد و در ویژگی‌های غیرترتیبی از روش‌هایی مانند one-hot-encoding استفاده کرد (در این پروژه تمام ویژگی‌های دسته‌ای به صورت mapping تبدیل شده‌اند تا در پر کردن داده‌های مفقود برای این دسته از داده‌ها مشکلی بوجود نیاید).

Encoding

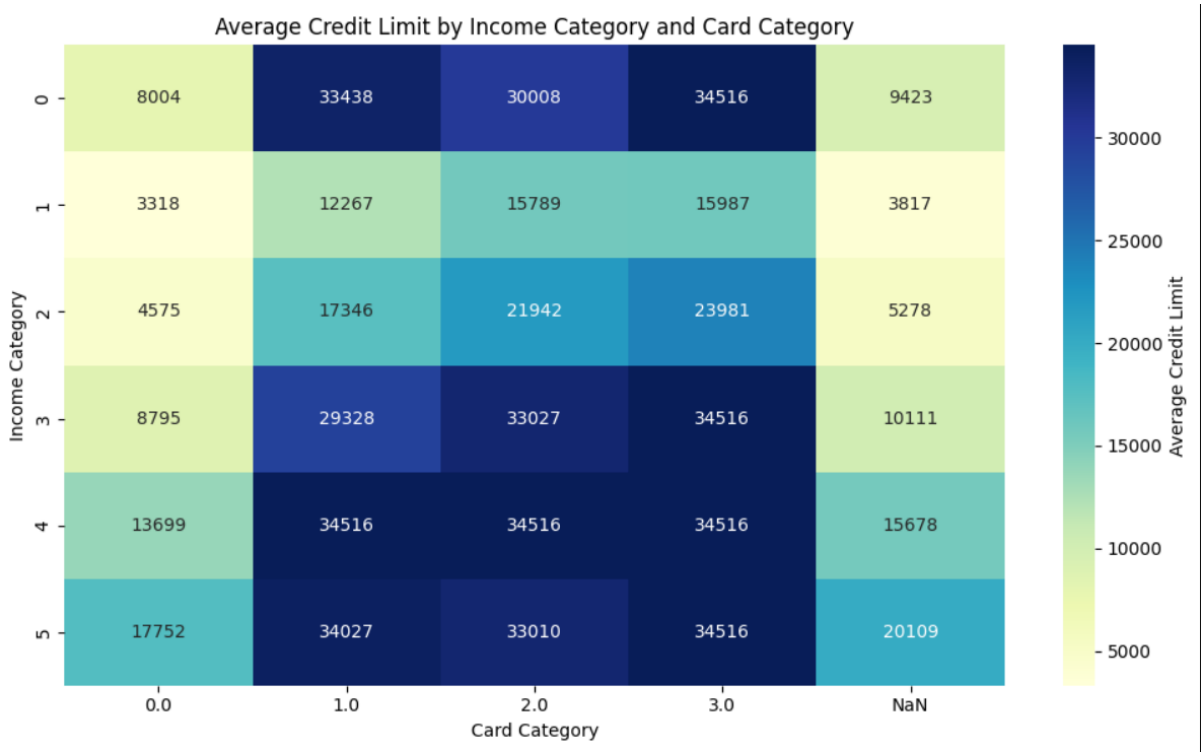
```python
def encoding(df):
    education_mapping = {'Unknown': 0, 'Uneducated': 1, 'High School': 2, 'College': 3,
                         'Graduate': 4, 'Post-Graduate': 5, 'Doctorate': 6}
    income_mapping = {'Unknown': 0, 'Less than $40K': 1, '$40K - $60K': 2, '$60K - $80K': 3,
                      '$80K - $120K': 4, '$120K +': 5}
    card_mapping = {'Blue': 0, 'Silver': 1, 'Gold': 2, 'Platinum': 3}
    gender_mapping = {'F': 0, 'M': 1}
    marital_mapping = {'Divorced': 0, 'Unknown': 1, 'Single': 2, 'Married': 3}


    # Apply mapping to each categorical feature
    df['Education_Level'] = df['Education_Level'].map(education_mapping)
    df['Income_Category'] = df['Income_Category'].map(income_mapping)
    df['Card_Category'] = df['Card_Category'].map(card_mapping)
    df['Gender'] = df['Gender'].map(gender_mapping)
    df['Marital_Status'] = df['Marital_Status'].map(marital_mapping)
```

- مدیریت داده‌های مفقود

	Total	%
Marital_Status	1939	23.9
Card_Category	1915	23.6
Months_on_book	221	2.7
Gender	199	2.5
Total_Relationship_Count	20	0.2

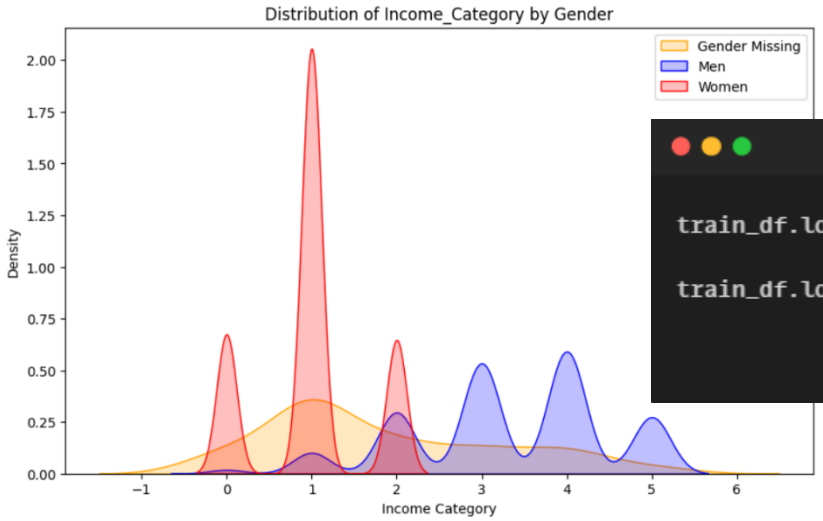
برای پرکردن داده‌های مفقود و روش مدیریت آن‌ها باید به درصد مفقودی داده، بررسی زمینه‌های احتمالی در خالی بودن آن‌ها و نیز نوع داده‌ها توجه داشت.

ویژگی card_category با توجه به ماتریس کواریانس ارتباط نزدیکی با Income_category دارد و می‌توان از ارتباط آن‌ها جهت تحلیل بهتر استفاده کرد.



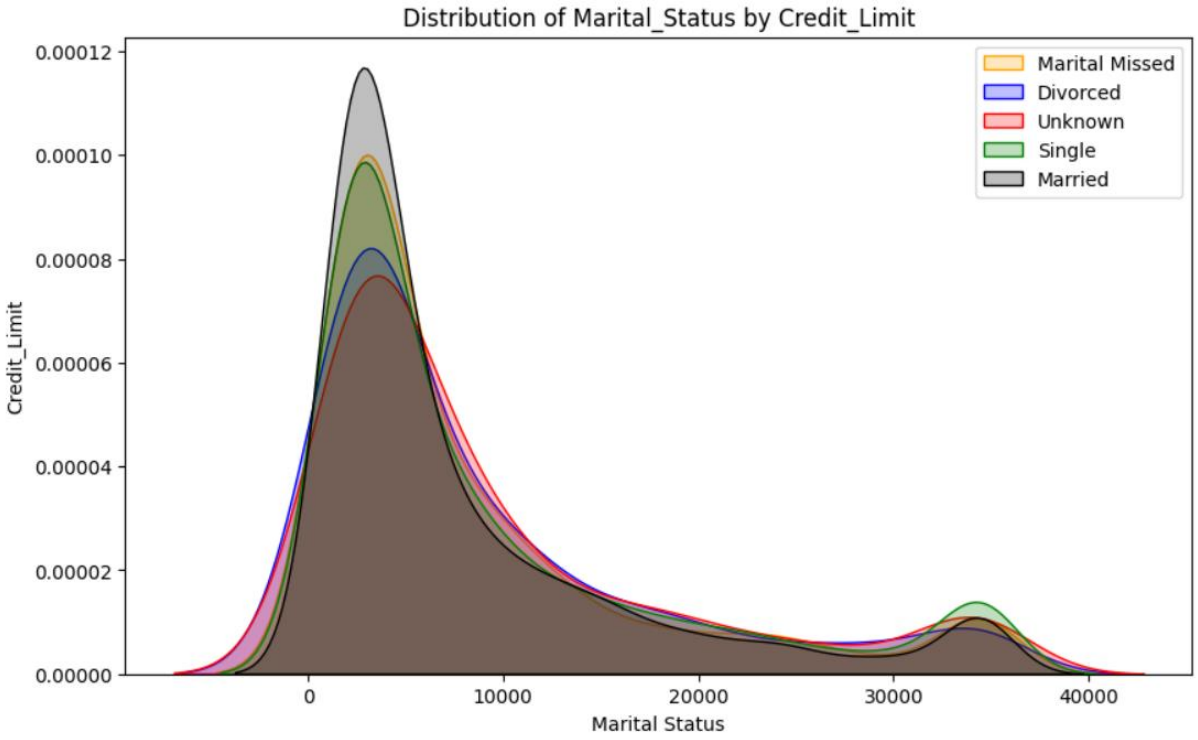
با توجه به اینکه میانگین credit_Limit در داده‌های مفقود شباهت زیادی به card_category_blue دارد، می‌توان داده‌های مفقود در این ویژگی را با مقدار صفر که بیانگر BLUE است، پر کرد.

جهت مدیریت کردن داده‌های مفقود در Gender می‌توان از income_category استفاده کرد.




```
train_df.loc[train_df['Card_Category'].isin([3, 4, 5]) &
             train_df['Gender'].isnull(), 'Gender'] = 1
train_df.loc[train_df['Card_Category'].isin([0, 1, 2]) &
             train_df['Gender'].isnull(), 'Gender'] = 0
```

همچنین جهت تکمیل داده‌های مفقودی وضعیت تاهل، از ارتباط آن با محدودیت اعتبار استفاده شده است:



(به نظر می‌رسد ممکن است افراد مجرد تمایل کمتری به پر کردن وضعیت تاهل داشته باشند، بنابراین داده‌های مفقود با single پر شدند.)

سایر ویژگی‌های(عددی) مفقود با استفاده از median تکمیل شدند.

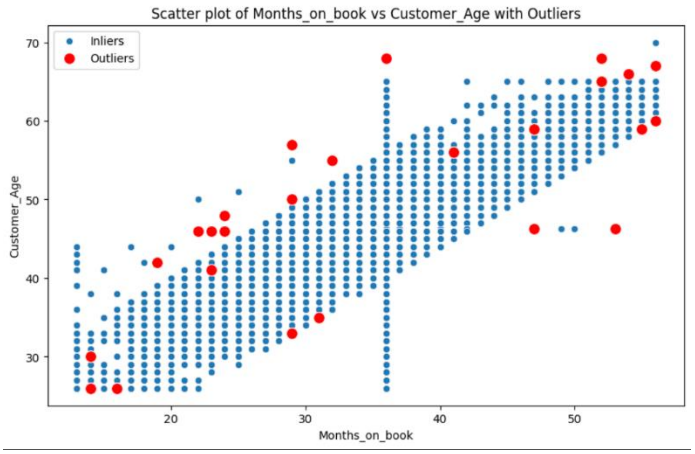
```
imputer = SimpleImputer(missing_values=np.nan, strategy='median')

imputer.fit(train_df[['Months_on_book']])
train_df['Months_on_book']= imputer.fit_transform(train_df[['Months_on_book']])
test_df['Months_on_book']= imputer.transform(test_df[['Months_on_book']])

imputer.fit(train_df[['Total_Relationship_Count']])
train_df['Total_Relationship_Count']= imputer.fit_transform(train_df[['Total_Relationship_Count']])
test_df['Total_Relationship_Count']= imputer.transform(test_df[['Total_Relationship_Count']])
```

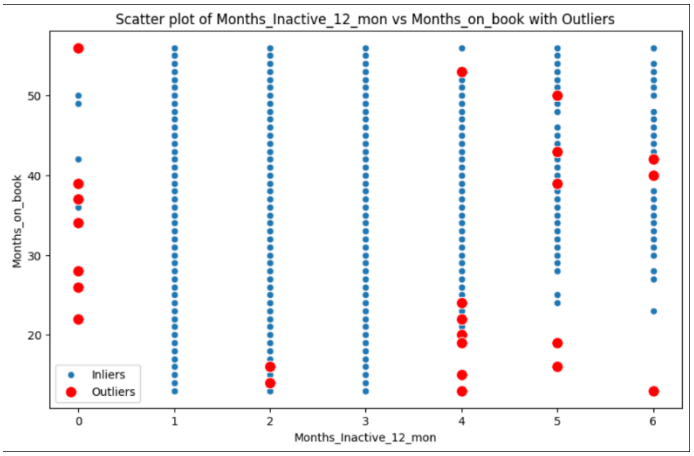
- مدیریت داده‌های پرت

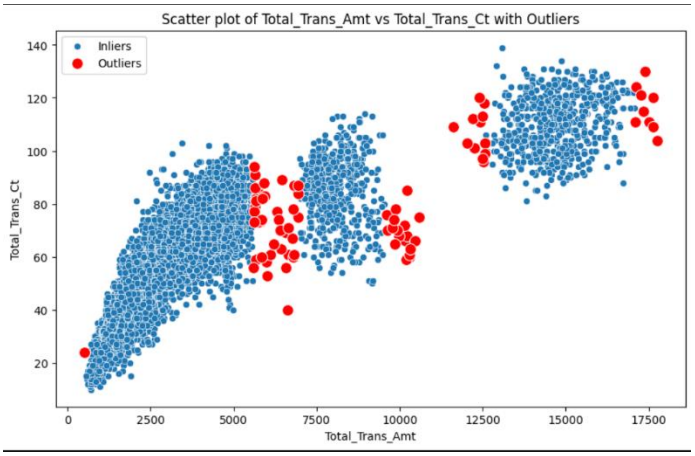
دسته‌ای از داده‌های پرت به این علت ایجاد شده‌اند که کاربر مقدار نادرستی وارد کرده باشد، همچنین رنج برخی ویژگی‌ها مشخص است. در این دیتاست، ویژگی سن نمی‌تواند بیشتر از ۱۲۵ باشد و در غیراین‌صورت توسط میانگین جایگزین می‌شود.

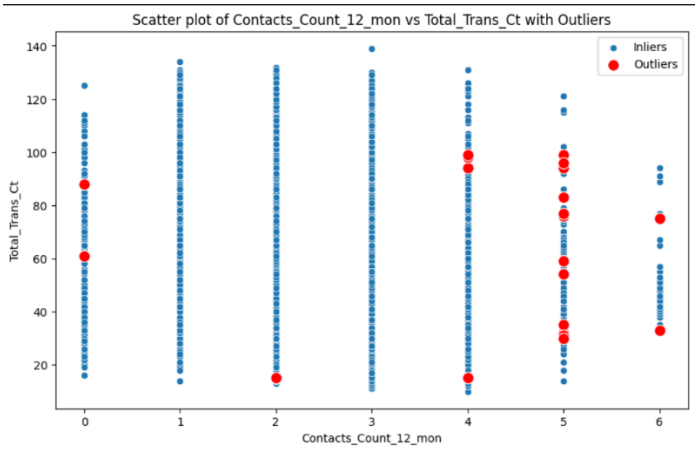
```
# Detect ages greater than 125 and set them to NaN
train_df['Customer_Age'] = train_df['Customer_Age'].apply(lambda x: np.nan if x > 125 else x)

# Impute the missing values
imputer = SimpleImputer(strategy='mean')
train_df['Customer_Age'] = imputer.fit_transform(train_df[['Customer_Age']])
```

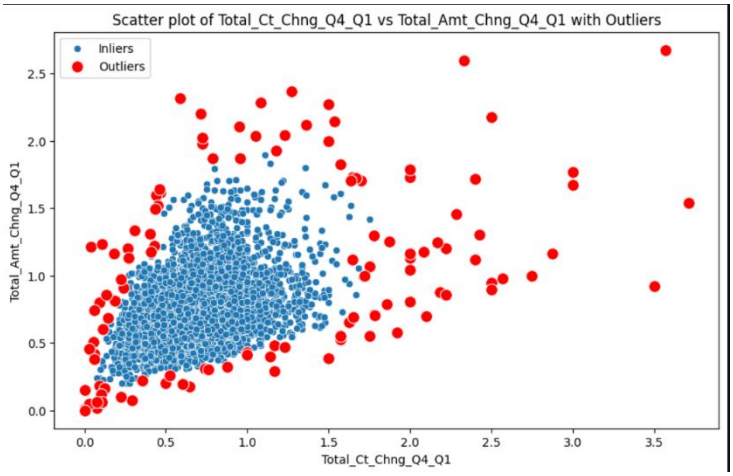
در ادامه جهت تشخیص داده‌های پرت در ویژگی‌های عددی از روش LOF بهره گرفته شده است و نیز جهت تشخیص بهتر، از ویژگی مرتبطتر با آن کمک گرفته شده است.

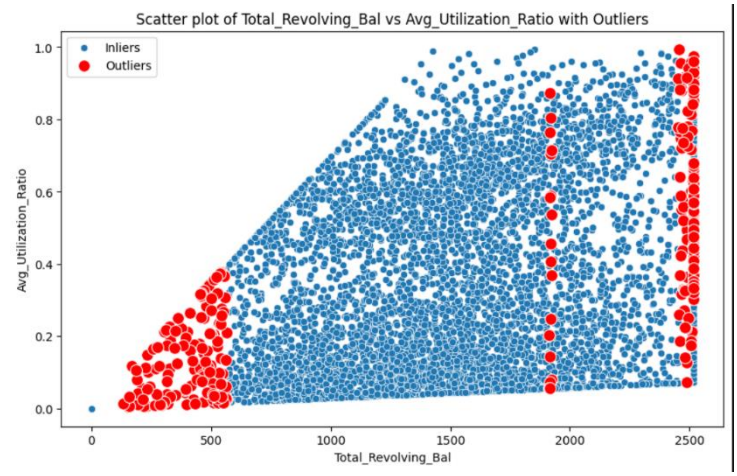












نتایج پس از حذف داده‌های پرت از ویژگی‌های غیر دسته‌ای:

	count	mean	std	min	25%	50%	75%	max
Customer_Age	7051.0	46.221101	7.919221	26.000	41.0000	46.000	52.000	70.000
Gender	7051.0	0.468586	0.499048	0.000	0.0000	0.000	1.000	1.000
Dependent_count	7051.0	2.352149	1.291687	0.000	1.0000	2.000	3.000	5.000
Education_Level	7051.0	2.591547	1.687487	0.000	1.0000	3.000	4.000	6.000
Marital_Status	7051.0	2.205503	0.788501	0.000	2.0000	2.000	3.000	3.000
Income_Category	7051.0	2.101120	1.475036	0.000	1.0000	2.000	3.000	5.000
Card_Category	7051.0	0.057864	0.273768	0.000	0.0000	0.000	0.000	3.000
Months_on_book	7051.0	35.858176	7.801911	13.000	32.0000	36.000	40.000	56.000
Total_Relationship_Count	7051.0	3.831513	1.549918	1.000	3.0000	4.000	5.000	6.000
Months_Inactive_12_mon	7051.0	2.336123	1.001799	0.000	2.0000	2.000	3.000	6.000
Contacts_Count_12_mon	7051.0	2.437668	1.090992	0.000	2.0000	2.000	3.000	6.000
Credit_Limit	7051.0	8387.680882	8546.904071	1438.300	2588.5000	4637.000	10947.500	34516.000
Total_Revolving_Bal	7051.0	1161.738193	788.124240	0.000	610.0000	1293.000	1753.000	2517.000
Total_Amt_Chng_Q4_Q1	7051.0	0.756057	0.197790	0.198	0.6315	0.735	0.856	1.893
Total_Trans_Amt	7051.0	4442.990923	3388.815260	563.000	2212.5000	3944.000	4740.000	17064.000
Total_Trans_Ct	7051.0	65.752517	23.211880	10.000	46.0000	68.000	81.000	139.000
Total_Ct_Chng_Q4_Q1	7051.0	0.706820	0.202007	0.074	0.5870	0.702	0.818	1.750
Avg_Utilization_Ratio	7051.0	0.276259	0.272817	0.000	0.0290	0.181	0.504	0.995

داده پرت برای ویژگی‌های دسته‌ای معنا ندارد، اما بهرحال ممکن است داده‌های غیرعادی مشاهده شوند که توسط الگوریتم svm شناسایی و حذف شده‌اند.(لازم به ذکر است در این روش غیرعادی بودن تمام داده‌ها بررسی می‌شود)

```python
# Train the OC-SVM model on the training data
clf = OneClassSVM(nu=0.8, kernel='rbf', gamma='auto')
clf.fit(train_df)

# Detect anomalies in the training data
train_anomaly_scores = clf.decision_function(train_df)
train_anomalies_idx = np.where(train_anomaly_scores < 0)[0]

# Remove anomalies from the training data
train_df = train_df.drop(train_df.index[train_anomalies_idx])
```

- انتخاب ویژگی

در حذف ویژگی‌ها، اهمیت و نیز ارتباط آن‌ها با سایر ویژگی‌ها ملاک قرار داده شده است.
ویژگی‌های با اهمیت پایین و وابستگی بالا بهتر است حذف شوند.

	Feature	Importance	VIF
0	Avg_Utilization_Ratio	0.420781	6.062172
1	Income_Category	0.239581	8.630447
2	Total_Revolving_Bal	0.163871	7.174995
3	Card_Category	0.061275	1.296993
4	Total_Trans_Amt	0.016861	8.821232
5	Total_Amt_Chng_Q4_Q1	0.015222	16.236073
6	Total_Ct_Chng_Q4_Q1	0.014307	15.424124
7	Total_Trans_Ct	0.012174	25.031854
8	Customer_Age	0.010757	76.824692
9	Months_on_book	0.010366	57.213571
10	Total_Relationship_Count	0.007021	7.831607
11	Education_Level	0.006195	3.251686
12	Contacts_Count_12_mon	0.005856	5.587047
13	Dependent_count	0.004754	4.209314
14	Months_Inactive_12_mon	0.004551	6.314962
15	Marital_Status	0.004440	8.258021
16	Gender	0.001987	4.977622

با توجه به نتایج، ویژگی‌های زیر حذف شدند:

- Gender به دلیل کمترین اهمیت
- Marital_Status به علت اهمت کم و نیز وابستگی غیرخطی بیشتر از ۵
- Months_Inactive_12_mon به علت اهمت کم و ارتباط آن با Contacts_Cont_12_mon
- Customer_Age به علت وجود ارتباط غیرخطی بسیار بالا با سایرین

برای کاهش بعد و ایجاد ویژگی‌های جدید در رگرسیون می‌توان از pca کمک گرفت، اما به علت افزایش خطا از آن صرف نظر شد.

- برای اسکیل کردن داده‌های X از روش MinMaxScaler استفاده شد.(زیرا فرض ما این است داده‌ها دارای ماهیت خطی هستند.)

آموزش مدل:

ابتدا مدل RandomForestRegressor را با پارامترهای زیر تعریف می‌کنیم:

- n_estimators=100: تعداد درختان تصمیم‌گیری در جنگل.
- random_state=42: تنظیم یک بذر برای تکرارپذیری نتایج.
- oob_score=True: فعال‌سازی امتیاز خارج از بسته‌بندی (Out-of-Bag) برای ارزیابی مدل

Out-of-Bag Score: معیاری از عملکرد مدل است که با استفاده از داده‌های خارج از بسته‌بندی محاسبه می‌شود. این امتیاز می‌تواند به عنوان تخمینی از دقت مدل روی داده‌های جدید باشد.

Mean Squared Error (MSE): میانگین مربعات خطا، معیاری از دقت مدل است که میزان خطای پیش‌بینی را نشان می‌دهد. مقدار کمتر بهتر است.

R-squared (R2): ضریب تعیین، معیاری از تطابق مدل با داده‌های واقعی است. مقدار آن بین ۰ و ۱ است که مقدار نزدیک به ۱ نشان‌دهنده مدل بهتر است.

جهت اطمینان از پایدرای مدل، لازم است چندین بار با رندم استیت های مختلف و جداسازی متفاوت داده‌های ترین و تست، آموزش انجام شده و با توجه به باکس پلات نتایج، از پایدار بودن مدل اطمینان حاصل شود.

Out-of-Bag Score: 0.8871661646762087

Mean Squared Error (MSE): 9697483.072144547

R-squared (R2): 0.8860854003358002

این مدل از رگرسیون خطی و چندمتغیره بهتر عمل کرد.

همجینین در روشی دیگر، ابتدا بر روی داده‌ها، خوشه بندی صورت گرفت و سپس برای هر خوشه جداگانه رگسیون اعمال شد و نتایج میانگین خطا خوشه‌ها به شرح زیر است:

Total R^2 score: 0.9519513089566658

Total MSE: 3006279.951800513

Cluster 0: R^2 Score = 0.9796674313993743

Cluster 1: R^2 Score = 0.9765218824605615

Cluster 2: R^2 Score = 0.9831157726179997

Cluster 3: R^2 Score = 0.45874270363522673

Cluster 4: R^2 Score = 0.880808310110989