HW4_Part3_Henry_Romero

March 10, 2025

[2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import linear_model
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.preprocessing import MinMaxScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix,␣
  ↪classification_report, roc_curve, auc
# Load dataset
file_path = "~/Downloads/Bank Customer Churn Prediction.csv"  # Update to churn␣
  ↪dataset for logarithmic
df = pd.read_csv(file_path)

# Display dataset information
print("Dataset Info:\n", df.info())
print("\nDataset Description:\n", df.describe())
print("\nData Types:\n", df.dtypes)

# Preprocessing Section and handle missing values
print("\nChecking for missing values:\n", df.isnull().sum())

# Drop categorical columns
df.drop(columns=['customer_id', 'country', 'gender'], inplace=True,␣
  ↪errors='ignore')
df = df.apply(pd.to_numeric)
df.dropna(inplace=True)
continuous_columns = ['credit_score', 'age', 'tenure', 'balance',␣
  ↪'estimated_salary']
df[continuous_columns] = np.log1p(df[continuous_columns])


# Correlation Matrix
corr_matrix = df.corr()
print("\nCorrelation Matrix:\n", corr_matrix)
```

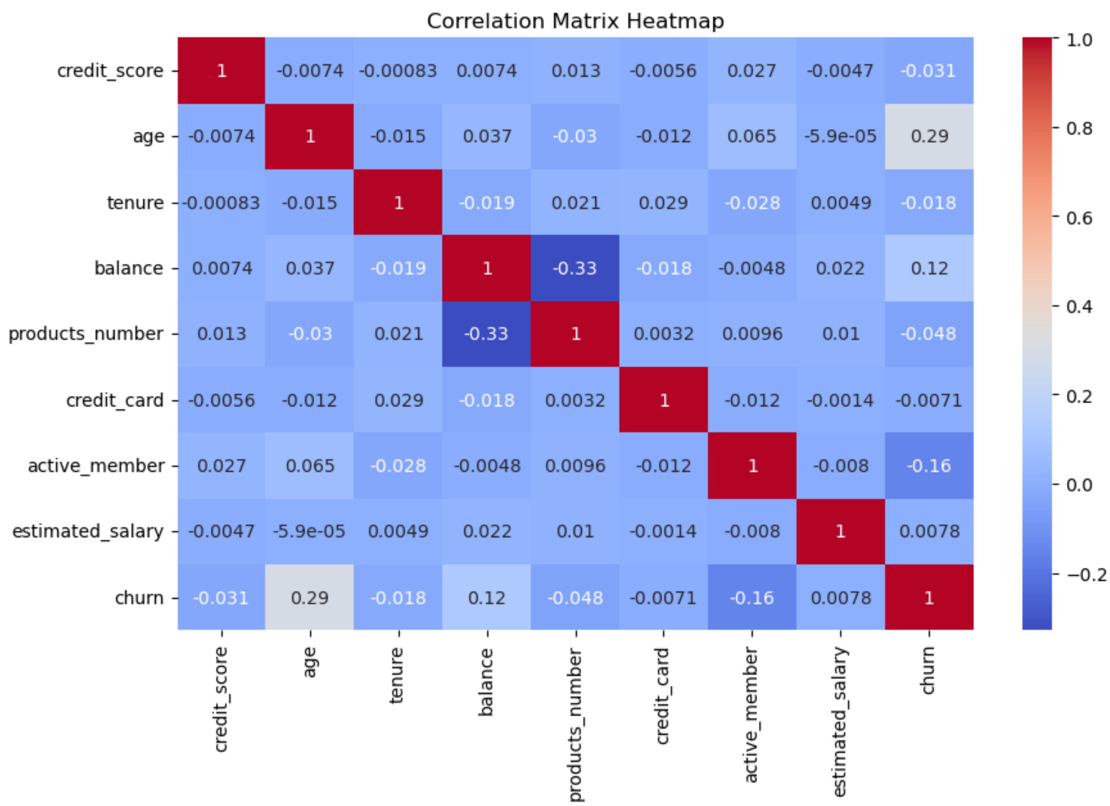
```python
# Generate heatmap
plt.figure(figsize=(10,6))
sns.heatmap(corr_matrix, annot=True, cmap="coolwarm")
plt.title("Correlation Matrix Heatmap")
plt.show()


# Normalize Data with the minmax scaler
scaler = MinMaxScaler()
df_scaled = pd.DataFrame(scaler.fit_transform(df), columns=df.columns)

# Split Data
X = df_scaled.iloc[:, :-1]  # Independent variables
y = df_scaled.iloc[:, -1]   # Dependent variable (Churn Prediction)

print("Mulivalue Variables:",X)
print("Dependent variable:",y.name)
print(y)


# Conduct and Train
model = linear_model.LinearRegression()
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state=42) # 20 percent in training here
model.fit(X_train, y_train)

# Make Predictions
y_pred = model.predict(X_test)
print("Prediction:", y_pred)


# Creating a confusion matrix
conf_matrix = confusion_matrix(y_test, np.round(y_pred))
print("\nConfusion Matrix:\n", conf_matrix)

# Evaluate the model
accuracy = accuracy_score(y_test, np.round(y_pred))
report = classification_report(y_test, np.round(y_pred))
cross_val = cross_val_score(model, X, y, cv=5).mean()

print("Accuracy Score:", accuracy)
print("Classification Report:\n", report)
print("Cross-Validation Score:", cross_val)


# Plotting graphs
plt.figure(figsize=(6,4))
```

```python
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap="Blues")
plt.title("Confusion Matrix Heatmap")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()


## Bar Chart to illuistrate the importance a variable is to sales by␣
↪coefficenient value
### Just like the heatmap, age is what impacts churn
plt.figure(figsize=(6,4))
coefficients = pd.DataFrame(model.coef_, X.columns, columns=['Coefficient'])
coefficients.plot(kind='bar', legend=False)
plt.title("Feature Importance")
plt.xlabel("Features")
plt.ylabel("Coefficient Value")
plt.grid(axis="y")
plt.show()

## ROC Curve
### Another graph i found interesting and relevant to the true positive rate.
fpr, tpr, _ = roc_curve(y_test, y_pred)
roc_auc = auc(fpr, tpr)
plt.figure(figsize=(6,4))
plt.plot(fpr, tpr, color="blue", label=f"ROC curve (area = {roc_auc:.2f})")
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("Receiver Operating Characteristic (ROC) Curve")
plt.legend(loc="lower right")
plt.show()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 12 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customer_id       10000 non-null  int64
 1   credit_score      10000 non-null  int64
 2   country           10000 non-null  object
 3   gender            10000 non-null  object
 4   age               10000 non-null  int64
 5   tenure            10000 non-null  int64
 6   balance           10000 non-null  float64
 7   products_number   10000 non-null  int64
 8   credit_card       10000 non-null  int64
 9   active_member     10000 non-null  int64
 10  estimated_salary  10000 non-null  float64
 11  churn             10000 non-null  int64
```

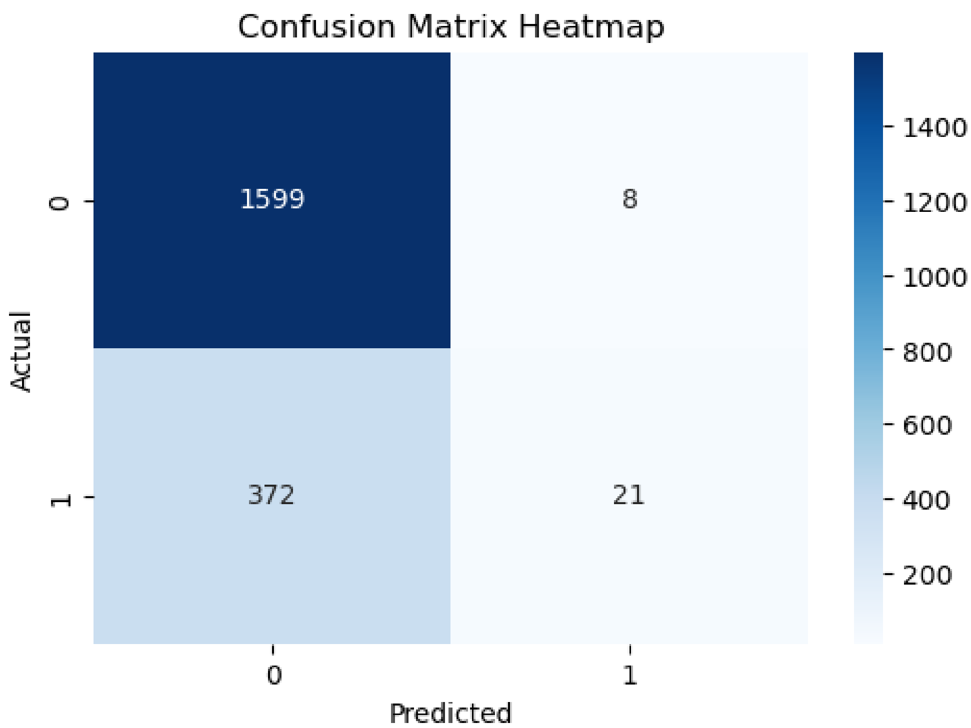
```
dtypes: float64(2), int64(8), object(2)
memory usage: 937.6+ KB
Dataset Info:
 None

Dataset Description:
        customer_id  credit_score           age        tenure        balance  \
count  1.000000e+04  10000.000000  10000.000000  10000.000000   10000.000000
mean   1.569094e+07    650.528800     38.921800      5.012800   76485.889288
std    7.193619e+04     96.653299     10.487806      2.892174   62397.405202
min    1.556570e+07    350.000000     18.000000      0.000000       0.000000
25%    1.562853e+07    584.000000     32.000000      3.000000       0.000000
50%    1.569074e+07    652.000000     37.000000      5.000000   97198.540000
75%    1.575323e+07    718.000000     44.000000      7.000000  127644.240000
max    1.581569e+07    850.000000     92.000000     10.000000  250898.090000

       products_number  credit_card  active_member  estimated_salary  \
count     10000.000000  10000.00000   10000.000000      10000.000000
mean          1.530200      0.70550       0.515100     100090.239881
std           0.581654      0.45584       0.499797      57510.492818
min           1.000000      0.00000       0.000000         11.580000
25%           1.000000      0.00000       0.000000      51002.110000
50%           1.000000      1.00000       1.000000     100193.915000
75%           2.000000      1.00000       1.000000     149388.247500
max           4.000000      1.00000       1.000000     199992.480000

              churn
count  10000.000000
mean       0.203700
std        0.402769
min        0.000000
25%        0.000000
50%        0.000000
75%        0.000000
max        1.000000

Data Types:
 customer_id           int64
credit_score          int64
country              object
gender               object
age                   int64
tenure                int64
balance             float64
products_number       int64
credit_card           int64
active_member         int64
estimated_salary    float64
```

```
churn                 int64
dtype: object

Checking for missing values:
 customer_id         0
credit_score        0
country             0
gender              0
age                 0
tenure              0
balance             0
products_number     0
credit_card         0
active_member       0
estimated_salary    0
churn               0
dtype: int64

Correlation Matrix:
                   credit_score       age    tenure   balance  products_number  \
credit_score           1.000000 -0.007398 -0.000829  0.007392         0.012914
age                   -0.007398  1.000000 -0.015047  0.037131        -0.030077
tenure                -0.000829 -0.015047  1.000000 -0.019075         0.021025
balance                0.007392  0.037131 -0.019075  1.000000        -0.329162
products_number        0.012914 -0.030077  0.021025 -0.329162         1.000000
credit_card           -0.005622 -0.011685  0.028520 -0.018177         0.003183
active_member          0.027196  0.065389 -0.028382 -0.004769         0.009612
estimated_salary      -0.004709 -0.000059  0.004877  0.022323         0.010037
churn                 -0.030519  0.294225 -0.018036  0.122630        -0.047820

                  credit_card  active_member  estimated_salary     churn
credit_score        -0.005622       0.027196         -0.004709 -0.030519
age                 -0.011685       0.065389         -0.000059  0.294225
tenure               0.028520      -0.028382          0.004877 -0.018036
balance             -0.018177      -0.004769          0.022323  0.122630
products_number      0.003183       0.009612          0.010037 -0.047820
credit_card          1.000000      -0.011866         -0.001373 -0.007138
active_member       -0.011866       1.000000         -0.007994 -0.156128
estimated_salary    -0.001373      -0.007994          1.000000  0.007791
churn               -0.007138      -0.156128          0.007791  1.000000
```

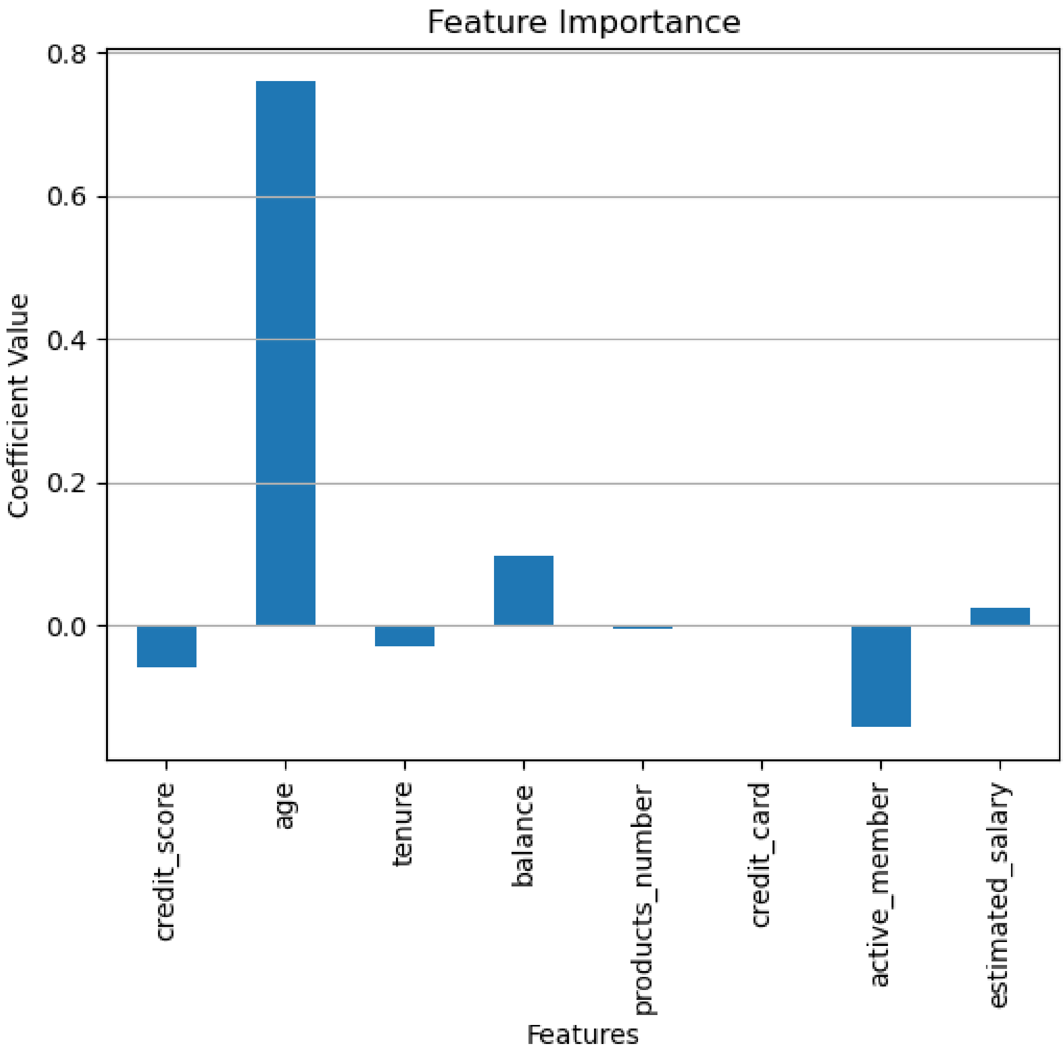


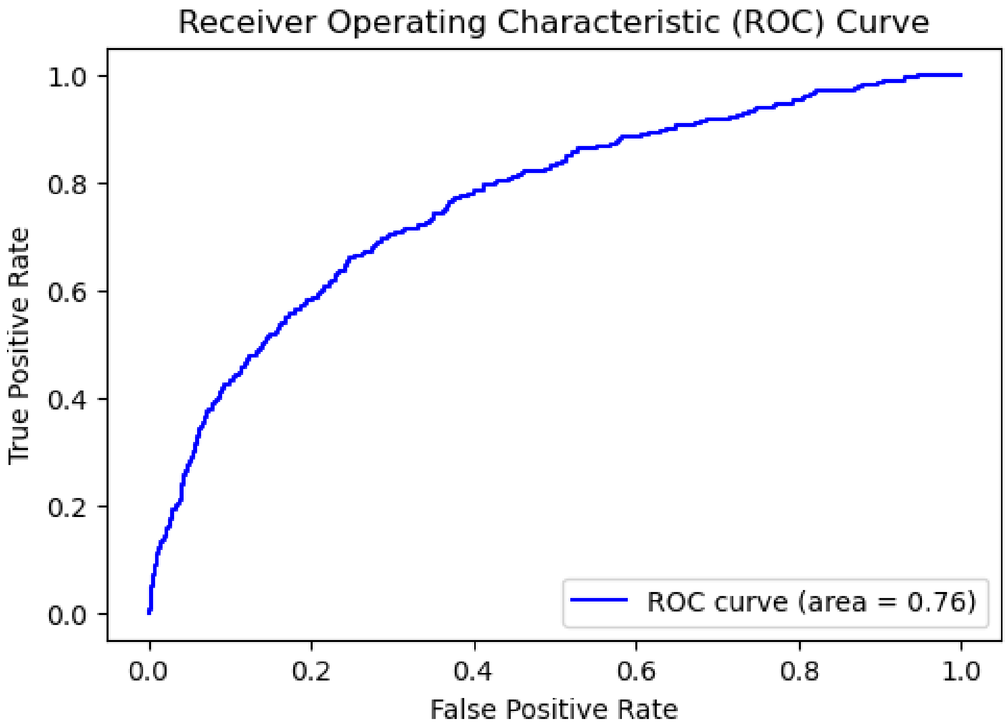

```
Mulivalue Variables:       credit_score        age    tenure    balance
products_number  \
0          0.642408  0.514281  0.458157  0.000000         0.000000
1          0.622195  0.499465  0.289065  0.911805         0.000000
2          0.406271  0.514281  0.916314  0.963645         0.666667
3          0.779442  0.468744  0.289065  0.000000         0.333333
4          1.000000  0.528757  0.458157  0.944288         0.000000
...             ...       ...       ...       ...              ...
9995       0.889990  0.468744  0.747222  0.000000         0.333333
9996       0.437269  0.402403  1.000000  0.881321         0.000000
9997       0.795459  0.419655  0.867194  0.000000         0.000000
9998       0.891452  0.514281  0.578130  0.902955         0.333333
9999       0.920295  0.266256  0.671188  0.947203         0.000000

      credit_card  active_member  estimated_salary
0             1.0            1.0          0.929738
1             0.0            1.0          0.940568
2             1.0            0.0          0.941836
3             0.0            0.0          0.921767
4             1.0            1.0          0.904097
...           ...            ...               ...
9995          1.0            0.0          0.924425
```

```
9996          1.0             1.0           0.930096
9997          0.0             1.0           0.838891
9998          1.0             0.0           0.920728
9999          1.0             0.0           0.828853

[10000 rows x 8 columns]
Dependent variable: churn
0       1.0
1       0.0
2       1.0
3       0.0
4       0.0
       …
9995    0.0
9996    0.0
9997    1.0
9998    1.0
9999    0.0
Name: churn, Length: 10000, dtype: float64
Prediction: [0.24104965 0.15601169 0.29663477 … 0.30858538 0.09519292
0.20152332]

Confusion Matrix:
 [[1599    8]
 [ 372   21]]
Accuracy Score: 0.81
Classification Report:
               precision    recall  f1-score   support

         0.0       0.81      1.00      0.89      1607
         1.0       0.72      0.05      0.10       393

    accuracy                           0.81      2000
   macro avg       0.77      0.52      0.50      2000
weighted avg       0.79      0.81      0.74      2000

Cross-Validation Score: 0.12841086279777084
```



<Figure size 600x400 with 0 Axes>


Feature Importance
Coefficient Value
0.8
0.6
0.4
0.2
0.0
credit_score
age
tenure
balance
products_number
credit_card
active_member
estimated_salary
Features


Receiver Operating Characteristic (ROC) Curve
1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
ROC curve (area = 0.76)