Christian Campbell

Data Mining Final Project

Milestone 1

1.1. Importing libraries

In [1]:
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
```

1.2. Importing the dataset

In [2]:
```
database = r"C:\Users\chris\Documents\Bellevue University\Data Mining\Final Project\BankChurners.csv\Bank
bank_df = pd.read_csv(database)
bank_df.head(10)
```

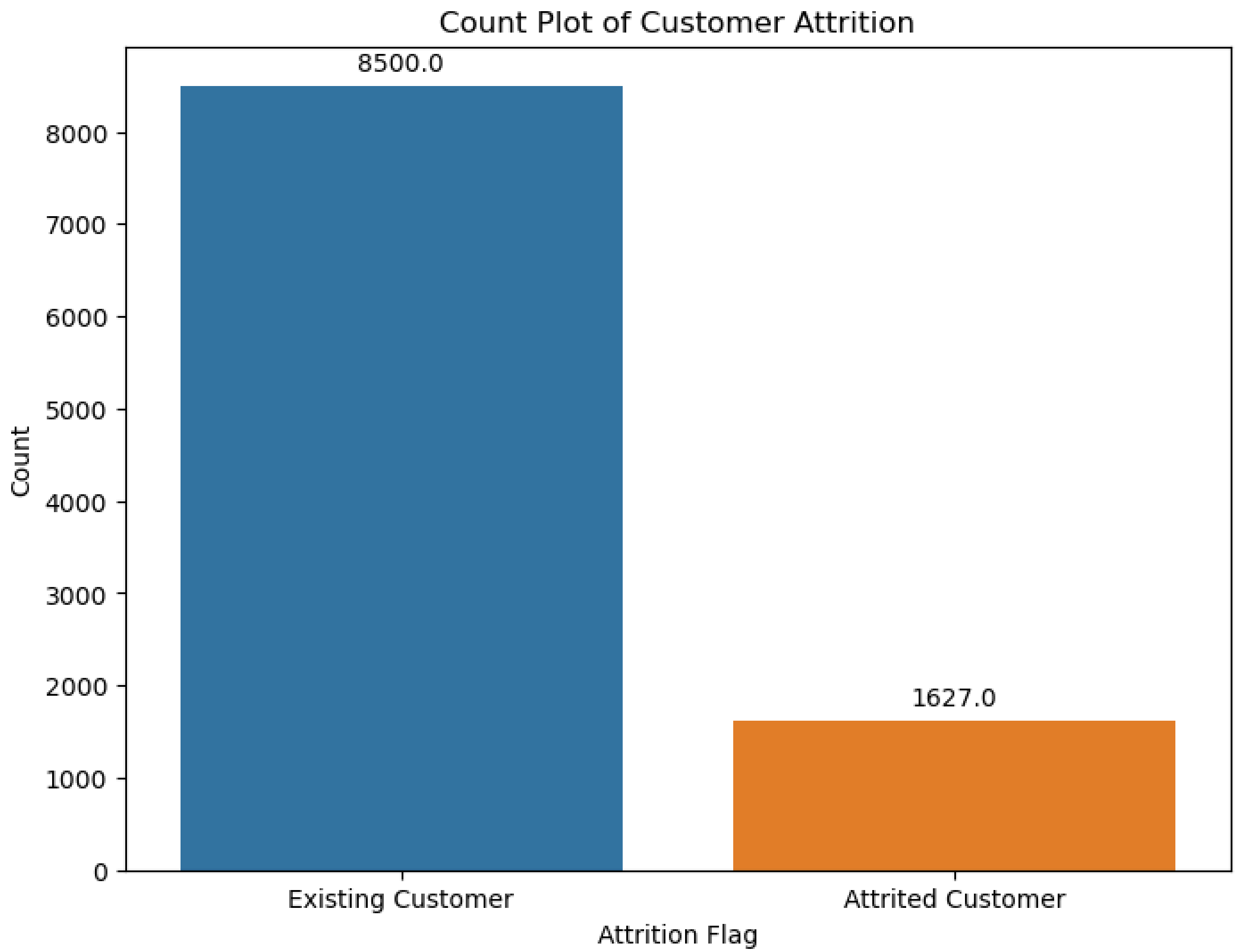
Out[2]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	Card_
0	768805383	Existing Customer	45	M	3	High School	Married	$60K-$80K	
1	818770008	Existing Customer	49	F	5	Graduate	Single	Less than $40K	
2	713982108	Existing Customer	51	M	3	Graduate	Married	$80K-$120K	
3	769911858	Existing Customer	40	F	4	High School	Unknown	Less than $40K	
4	709106358	Existing Customer	40	M	3	Uneducated	Married	$60K-$80K	
5	713061558	Existing Customer	44	M	2	Graduate	Married	$40K-$60K	
6	810347208	Existing Customer	51	M	4	Unknown	Married	$120K +	
7	818906208	Existing Customer	32	M	0	High School	Unknown	$60K-$80K	
8	710930508	Existing Customer	37	M	3	Uneducated	Single	$60K-$80K	
9	719661558	Existing Customer	48	M	2	Graduate	Single	$80K-$120K	

10 rows × 23 columns

1.3. Creating a "Customer Attrition" bar chart

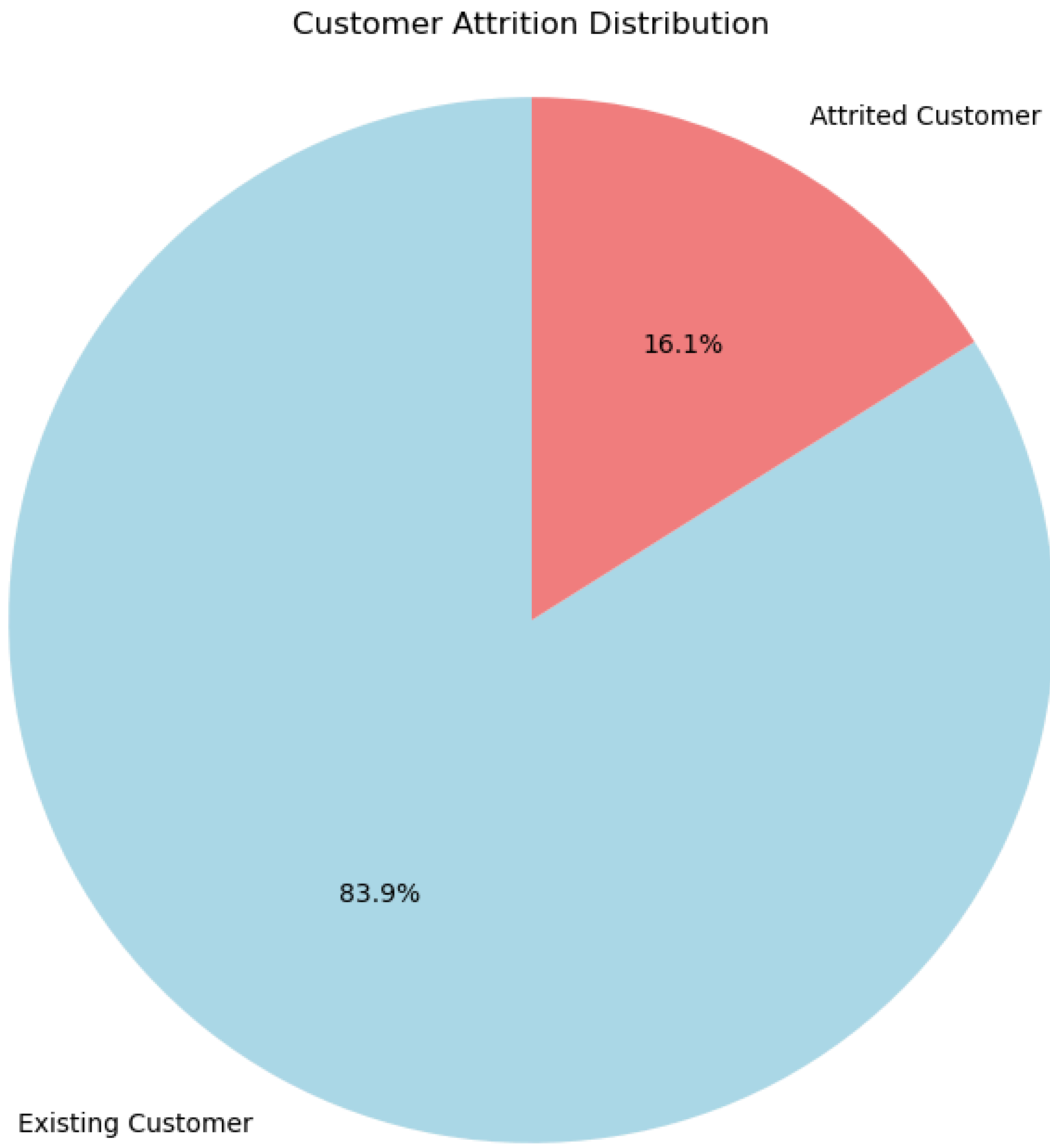
In [3]:
```
plt.figure(figsize=(8, 6))
ax = sns.countplot(data=bank_df, x='Attrition_Flag')

# This code annotates the bars with counts
for p in ax.patches:
    ax.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='center', xytext=(0, 9), textcoords='offset points')


plt.xlabel('Attrition Flag')
plt.ylabel('Count')
plt.title('Count Plot of Customer Attrition')
plt.show()
```



1.4. Creating a "Customer Attrition" Pie Chart

In [4]:

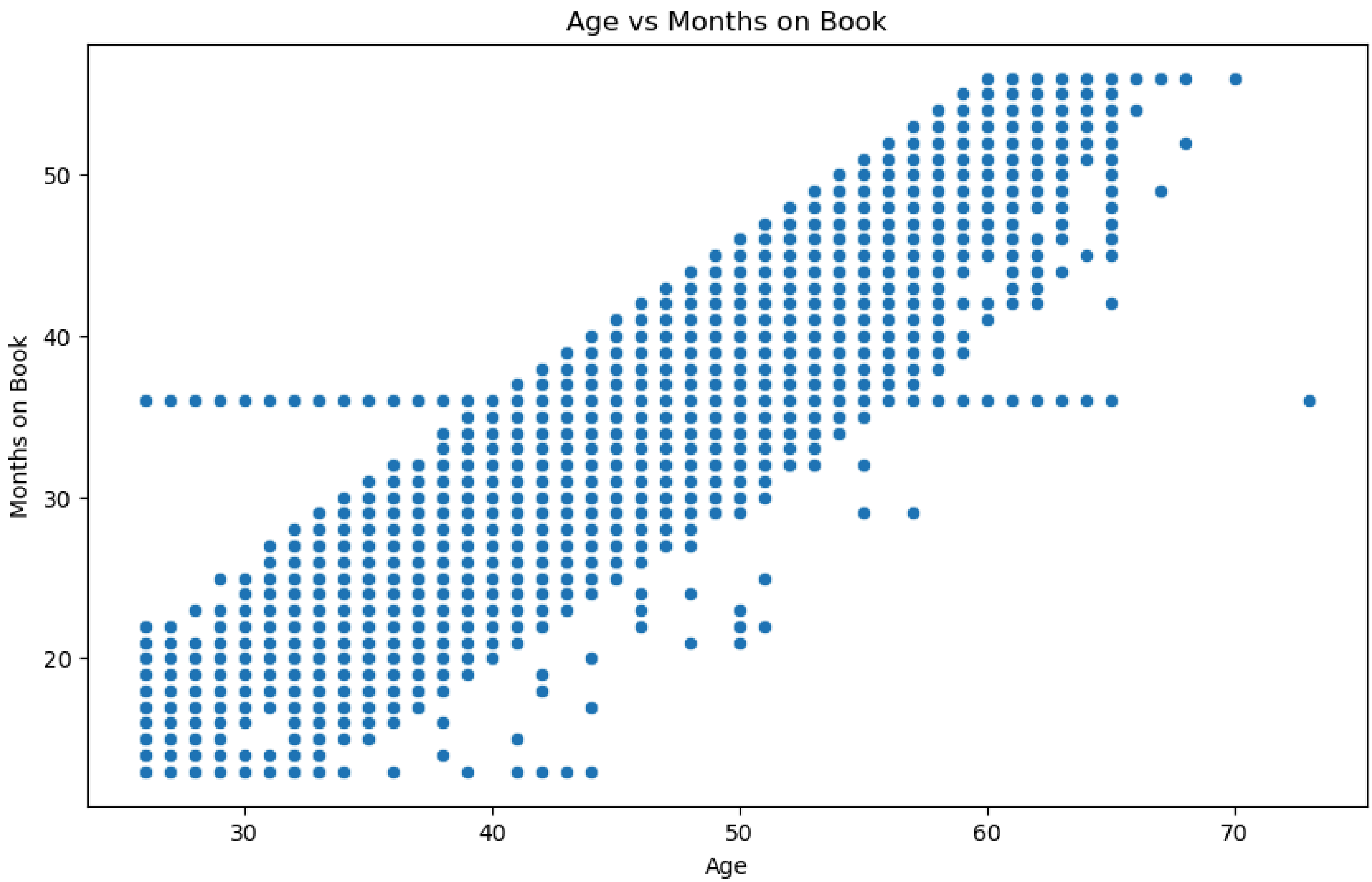
```
counts = bank_df['Attrition_Flag'].value_counts() # Calculates counts


plt.figure(figsize=(8, 8))
plt.pie(counts, labels=counts.index, autopct='%1.1f%%', startangle=90, colors=['lightblue', 'lightcoral']
plt.title('Customer Attrition Distribution')
plt.axis('equal')
plt.show()
```



1.5. Creating an "Age vs Months on Book" scatter plot

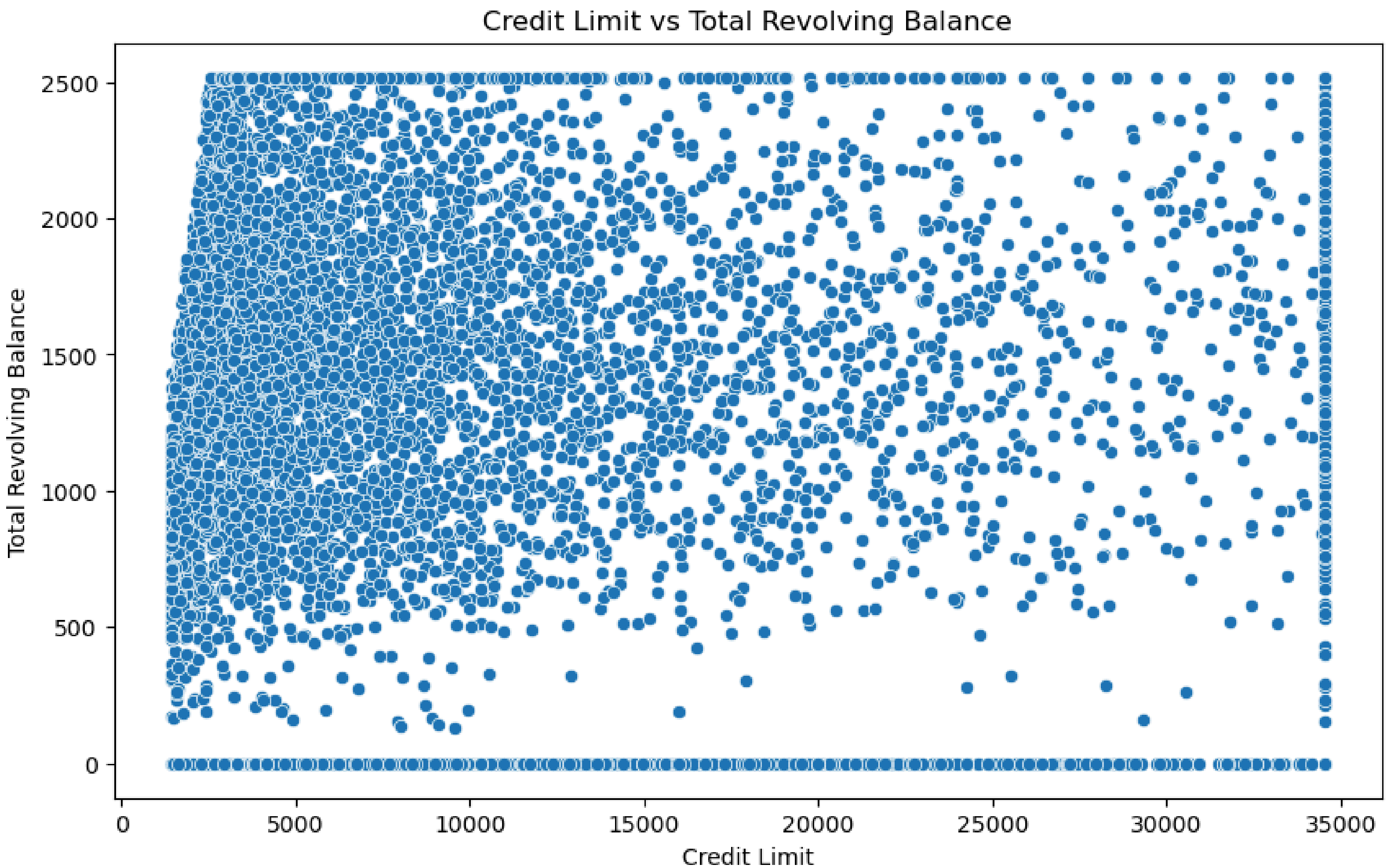
In [5]:

```
plt.figure(figsize=(10, 6))
sns.scatterplot(data=bank_df, x='Customer_Age', y='Months_on_book')
plt.xlabel('Age')
plt.ylabel('Months on Book')
plt.title('Age vs Months on Book')
plt.show()
```



1.6. Creating a "Credit Limit vs Total Revolving Balance" scatter plot

In [6]:
```
plt.figure(figsize=(10, 6))
sns.scatterplot(data=bank_df, x='Credit_Limit', y='Total_Revolving_Bal')
plt.xlabel('Credit Limit')
plt.ylabel('Total Revolving Balance')
plt.title('Credit Limit vs Total Revolving Balance')
plt.show()
```



Milestone 2

2.1. Deleting the last two columns that contain "naive baye classifier."

In [7]:
```
# Identifies columns that contain 'Naive_Bayes_Classifier' in their names
columns_to_drop = bank_df.filter(like='Naive_Bayes_Classifier').columns

# Drops these columns and creates a new DataFrame bank_df1
bank_df1 = bank_df.drop(columns=columns_to_drop)

bank_df1.head(10)
```

Out[7]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	Card_
0	768805383	Existing Customer	45	M	3	High School	Married	$60K-$80K	
1	818770008	Existing Customer	49	F	5	Graduate	Single	Less than $40K	
2	713982108	Existing Customer	51	M	3	Graduate	Married	$80K-$120K	
3	769911858	Existing Customer	40	F	4	High School	Unknown	Less than $40K	
4	709106358	Existing Customer	40	M	3	Uneducated	Married	$60K-$80K	
5	713061558	Existing Customer	44	M	2	Graduate	Married	$40K-$60K	
6	810347208	Existing Customer	51	M	4	Unknown	Married	$120K +	
7	818906208	Existing Customer	32	M	0	High School	Unknown	$60K-$80K	
8	710930508	Existing Customer	37	M	3	Uneducated	Single	$60K-$80K	
9	719661558	Existing Customer	48	M	2	Graduate	Single	$80K-$120K	

10 rows × 21 columns

2.2. Checking for null values.

In [8]:
```
# Checks for null values in each column
null_counts = bank_df1.isnull().sum()

# Displays the count of null values in each column
print(null_counts)
```

```
CLIENTNUM                   0
Attrition_Flag              0
Customer_Age                0
Gender                      0
Dependent_count             0
Education_Level             0
Marital_Status              0
Income_Category             0
Card_Category               0
Months_on_book              0
Total_Relationship_Count    0
Months_Inactive_12_mon      0
Contacts_Count_12_mon       0
Credit_Limit                0
Total_Revolving_Bal         0
Avg_Open_To_Buy             0
Total_Amt_Chng_Q4_Q1        0
Total_Trans_Amt             0
Total_Trans_Ct              0
Total_Ct_Chng_Q4_Q1         0
Avg_Utilization_Ratio       0
dtype: int64
```

2.3. Deleting CLIENTNUM column

In [9]:
```python
# Drop the column 'CLIENTNUM' and create a new DataFrame bank_df2
bank_df2 = bank_df1.drop(columns=['CLIENTNUM'])

# Display the updated DataFrame
bank_df2.head(10)
```

Out[9]:

	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	Card_Category	Mc
0	Existing Customer	45	M	3	High School	Married	$60K-$80K	Blue	
1	Existing Customer	49	F	5	Graduate	Single	Less than $40K	Blue	
2	Existing Customer	51	M	3	Graduate	Married	$80K-$120K	Blue	
3	Existing Customer	40	F	4	High School	Unknown	Less than $40K	Blue	
4	Existing Customer	40	M	3	Uneducated	Married	$60K-$80K	Blue	
5	Existing Customer	44	M	2	Graduate	Married	$40K-$60K	Blue	
6	Existing Customer	51	M	4	Unknown	Married	$120K +	Gold	
7	Existing Customer	32	M	0	High School	Unknown	$60K-$80K	Silver	
8	Existing Customer	37	M	3	Uneducated	Single	$60K-$80K	Blue	
9	Existing Customer	48	M	2	Graduate	Single	$80K-$120K	Blue	

2.4. Checking the data type of each column.

In [10]:
```
# Checks the data type of each column
column_data_types = bank_df2.dtypes

print(column_data_types)
```

```
Attrition_Flag               object
Customer_Age                  int64
Gender                       object
Dependent_count               int64
Education_Level              object
Marital_Status               object
Income_Category              object
Card_Category                object
Months_on_book                int64
Total_Relationship_Count      int64
Months_Inactive_12_mon        int64
Contacts_Count_12_mon         int64
Credit_Limit                float64
Total_Revolving_Bal           int64
Avg_Open_To_Buy             float64
Total_Amt_Chng_Q4_Q1        float64
Total_Trans_Amt               int64
Total_Trans_Ct                int64
Total_Ct_Chng_Q4_Q1         float64
Avg_Utilization_Ratio       float64
dtype: object
```

2.5. Checking the values in the following categorical columns

In [11]:
```
# List of known categorical columns
categorical_columns = ['Attrition_Flag', 'Gender', 'Education_Level', 'Marital_Status', 'Income_Category'

# Iterates over each categorical column and prints the unique values
for column in categorical_columns:
    unique_values = bank_df2[column].unique()
    print(f"Unique values in column '{column}':")
    print(unique_values)
    print()
```

```
Unique values in column 'Attrition_Flag':
['Existing Customer' 'Attrited Customer']

Unique values in column 'Gender':
['M' 'F']

Unique values in column 'Education_Level':
['High School' 'Graduate' 'Uneducated' 'Unknown' 'College' 'Post-Graduate'
 'Doctorate']

Unique values in column 'Marital_Status':
['Married' 'Single' 'Unknown' 'Divorced']

Unique values in column 'Income_Category':
['$60K - $80K' 'Less than $40K' '$80K - $120K' '$40K - $60K' '$120K +'
 'Unknown']

Unique values in column 'Card_Category':
['Blue' 'Gold' 'Silver' 'Platinum']
```

2.6. Checking to see how many "unknown" values are in each of the following columns.

In [12]:

```
# List of columns to check for "unknown" values
columns_to_check = ["Education_Level", "Marital_Status", "Income_Category"]

# Dictionary to store the count of "unknown" values for each column
unknown_counts = {}

# Iterates over each column and counts "unknown" values
for column in columns_to_check:
    unknown_count = (bank_df2[column] == "Unknown").sum()
    unknown_counts[column] = unknown_count

# Displays the count of "unknown" values in each specified column
for column, count in unknown_counts.items():
    print(f"Number of 'unknown' values in column '{column}': {count}")
```

```
Number of 'unknown' values in column 'Education_Level': 1519
Number of 'unknown' values in column 'Marital_Status': 749
Number of 'unknown' values in column 'Income_Category': 1112
```

2.7. Creating dummy Variables for the following columns.

In [13]:
```
# List of columns to convert into dummy variables
columns_to_dummy = ["Attrition_Flag", "Gender", "Education_Level", "Marital_Status", "Income_Category", '

# Converts specified columns to dummy variables
bank_df3 = pd.get_dummies(bank_df2, columns=columns_to_dummy, dtype=int)

# Set display option to show all columns
pd.set_option('display.max_columns', None)

print(bank_df3)
```

```
       Customer_Age  Dependent_count  Months_on_book  \
0                45                3              39   
1                49                5              44   
2                51                3              36   
3                40                4              34   
4                40                3              21   
...             ...              ...             ...   
10122            50                2              40   
10123            41                2              25   
10124            44                1              36   
10125            30                2              36   
10126            43                2              25   

       Total_Relationship_Count  Months_Inactive_12_mon  \
0                             5                       1   
1                             6                       1   
2                             4                       1   
3                             3                       4   
4                             5                       1   
...                         ...                     ...   
10122                         3                       2   
10123                         4                       2   
10124                         5                       3   
10125                         4                       3   
10126                         6                       2   

       Contacts_Count_12_mon  Credit_Limit  Total_Revolving_Bal  \
0                          3       12691.0                  777   
1                          2        8256.0                  864   
2                          0        3418.0                    0   
3                          1        3313.0                 2517   
4                          0        4716.0                    0   
...                      ...           ...                  ...   
10122                      3        4003.0                 1851   
10123                      3        4277.0                 2186   
10124                      4        5409.0                    0   
10125                      3        5281.0                    0   
10126                      4       10388.0                 1961   

       Avg_Open_To_Buy  Total_Amt_Chng_Q4_Q1  Total_Trans_Amt  Total_Trans_Ct  \
0              11914.0                 1.335             1144              42   
1               7392.0                 1.541             1291              33   
2               3418.0                 2.594             1887              20   
```

```
3               796.0                 1.405            1171              20
4              4716.0                 2.175             816              28
...               ...                   ...             ...             ...
10122          2152.0                 0.703           15476             117
10123          2091.0                 0.804            8764              69
10124          5409.0                 0.819           10291              60
10125          5281.0                 0.535            8395              62
10126          8427.0                 0.703           10294              61

       Total_Ct_Chng_Q4_Q1  Avg_Utilization_Ratio  \
0                    1.625                  0.061   
1                    3.714                  0.105   
2                    2.333                  0.000   
3                    2.333                  0.760   
4                    2.500                  0.000   
...                    ...                    ...   
10122                0.857                  0.462   
10123                0.683                  0.511   
10124                0.818                  0.000   
10125                0.722                  0.000   
10126                0.649                  0.189   

       Attrition_Flag_Attrited Customer  Attrition_Flag_Existing Customer  \
0                                     0                                 1   
1                                     0                                 1   
2                                     0                                 1   
3                                     0                                 1   
4                                     0                                 1   
...                                 ...                               ...   
10122                                 0                                 1   
10123                                 1                                 0   
10124                                 1                                 0   
10125                                 1                                 0   
10126                                 1                                 0   

       Gender_F  Gender_M  Education_Level_College  Education_Level_Doctorate  \
0             0         1                        0                          0   
1             1         0                        0                          0   
2             0         1                        0                          0   
3             1         0                        0                          0   
4             0         1                        0                          0   
...         ...       ...                      ...                        ...   
10122         0         1                        0                          0   
```

```
10123        0         1                        0                          0
10124        1         0                        0                          0
10125        0         1                        0                          0
10126        1         0                        0                          0

       Education_Level_Graduate  Education_Level_High School  \
0                             0                            1
1                             1                            0
2                             1                            0
3                             0                            1
4                             0                            0
...                         ...                          ...
10122                         1                            0
10123                         0                            0
10124                         0                            1
10125                         1                            0
10126                         1                            0

       Education_Level_Post-Graduate  Education_Level_Uneducated  \
0                                  0                           0
1                                  0                           0
2                                  0                           0
3                                  0                           0
4                                  0                           1
...                              ...                         ...
10122                              0                           0
10123                              0                           0
10124                              0                           0
10125                              0                           0
10126                              0                           0

       Education_Level_Unknown  Marital_Status_Divorced  \
0                            0                        0
1                            0                        0
2                            0                        0
3                            0                        0
4                            0                        0
...                        ...                      ...
10122                        0                        0
10123                        1                        1
10124                        0                        0
10125                        0                        0
10126                        0                        0
```

```
       Marital_Status_Married  Marital_Status_Single  Marital_Status_Unknown  \
0                           1                      0                       0
1                           0                      1                       0
2                           1                      0                       0
3                           0                      0                       1
4                           1                      0                       0
...                       ...                    ...                     ...
10122                       0                      1                       0
10123                       0                      0                       0
10124                       1                      0                       0
10125                       0                      0                       1
10126                       1                      0                       0

       Income_Category_$120K +  Income_Category_$40K - $60K  \
0                            0                            0
1                            0                            0
2                            0                            0
3                            0                            0
4                            0                            0
...                        ...                          ...
10122                        0                            1
10123                        0                            1
10124                        0                            0
10125                        0                            1
10126                        0                            0

       Income_Category_$60K - $80K  Income_Category_$80K - $120K  \
0                                1                             0
1                                0                             0
2                                0                             1
3                                0                             0
4                                1                             0
...                            ...                           ...
10122                            0                             0
10123                            0                             0
10124                            0                             0
10125                            0                             0
10126                            0                             0

       Income_Category_Less than $40K  Income_Category_Unknown  \
0                                   0                        0
1                                   1                        0
```

```
2                                   0                         0
3                                   1                         0
4                                   0                         0
...                               ...                       ...
10122                               0                         0
10123                               0                         0
10124                               1                         0
10125                               0                         0
10126                               1                         0

       Card_Category_Blue  Card_Category_Gold  Card_Category_Platinum  \
0                       1                   0                       0
1                       1                   0                       0
2                       1                   0                       0
3                       1                   0                       0
4                       1                   0                       0
...                   ...                 ...                     ...
10122                   1                   0                       0
10123                   1                   0                       0
10124                   1                   0                       0
10125                   1                   0                       0
10126                   0                   0                       0

       Card_Category_Silver
0                         0
1                         0
2                         0
3                         0
4                         0
...                     ...
10122                     0
10123                     0
10124                     0
10125                     0
10126                     1

[10127 rows x 39 columns]
```

2.8. Checking the data type of each column

In [14]:
```
# Checks the data type of each column
column_data_types = bank_df3.dtypes

print(column_data_types)
```

```
Customer_Age                          int64
Dependent_count                       int64
Months_on_book                        int64
Total_Relationship_Count              int64
Months_Inactive_12_mon                int64
Contacts_Count_12_mon                 int64
Credit_Limit                        float64
Total_Revolving_Bal                   int64
Avg_Open_To_Buy                     float64
Total_Amt_Chng_Q4_Q1                float64
Total_Trans_Amt                       int64
Total_Trans_Ct                        int64
Total_Ct_Chng_Q4_Q1                 float64
Avg_Utilization_Ratio               float64
Attrition_Flag_Attrited Customer      int32
Attrition_Flag_Existing Customer      int32
Gender_F                              int32
Gender_M                              int32
Education_Level_College               int32
Education_Level_Doctorate             int32
Education_Level_Graduate              int32
Education_Level_High School           int32
Education_Level_Post-Graduate         int32
Education_Level_Uneducated            int32
Education_Level_Unknown               int32
Marital_Status_Divorced               int32
Marital_Status_Married                int32
Marital_Status_Single                 int32
Marital_Status_Unknown                int32
Income_Category_$120K +               int32
Income_Category_$40K - $60K           int32
Income_Category_$60K - $80K           int32
Income_Category_$80K - $120K          int32
Income_Category_Less than $40K        int32
Income_Category_Unknown               int32
Card_Category_Blue                    int32
Card_Category_Gold                    int32
Card_Category_Platinum                int32
Card_Category_Silver                  int32
dtype: object
```

Milestone 3

3.1. Import necessary libraries

In [15]:
```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.linear_model import LogisticRegression
```

3.2. Splitting the data into training and testing data.

In [16]:
```
features = bank_df3.drop(columns=['Attrition_Flag_Attrited Customer']) # Features
target = bank_df3['Attrition_Flag_Attrited Customer'] # Target

# Splits the data into training and testing sets
features_train, features_test, target_train, target_test = train_test_split(features, target, test_size=
```

3.3. Training the Model on Gradient Boosting Classifier Model

In [17]:
```
# Initialize the Gradient Boosting Classifier
gb_classifier = GradientBoostingClassifier()

# Train the model
gb_classifier.fit(features_train, target_train)
```

Out[17]:
```
GradientBoostingClassifier()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

3.4. Testing the accuracy of the gb classifier model

In [18]:

```
# Makes predictions on the test set
target_pred = gb_classifier.predict(features_test)

# Calculates the accuracy
accuracy = accuracy_score(target_test, target_pred)

print(f'Accuracy of the Gradient Boosting model: {accuracy:.2f}')
```

```
Accuracy of the Gradient Boosting model: 1.00
```

3.5. Further evaluation of the gb classifier model

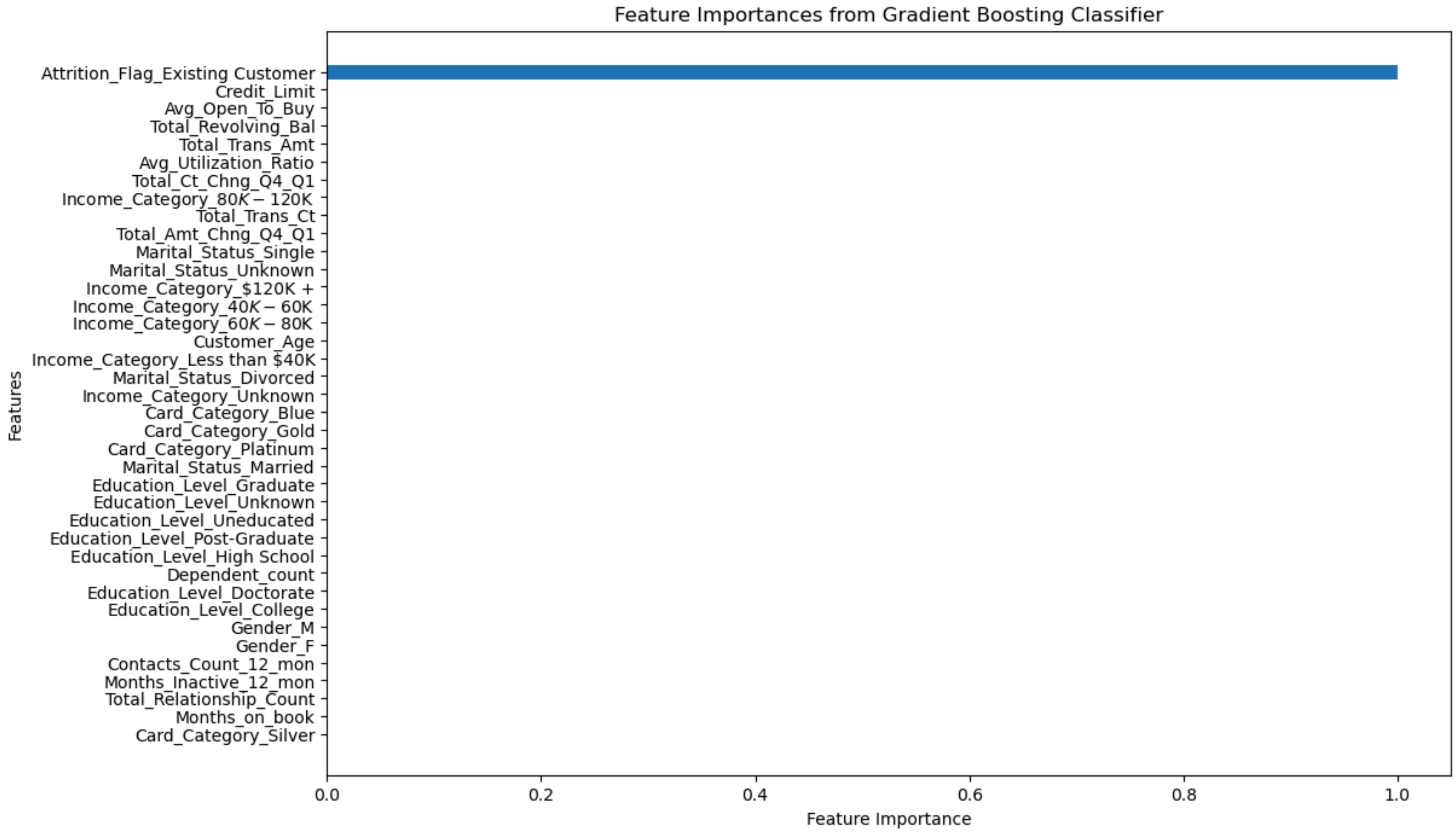
In [19]:

```
# Generate a classification report
print("Classification Report:")
print(classification_report(target_test, target_pred))

# Generate a confusion matrix
print("Confusion Matrix:")
print(confusion_matrix(target_test, target_pred))
```

```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      1699
           1       1.00      1.00      1.00       327

    accuracy                           1.00      2026
   macro avg       1.00      1.00      1.00      2026
weighted avg       1.00      1.00      1.00      2026

Confusion Matrix:
[[1699    0]
 [   0  327]]
```

3.6. Looking for feature importance

In [20]:
```
# Gets the feature importances
feature_importances = gb_classifier.feature_importances_

# Creates a DataFrame for better visualization
feature_importance_df = pd.DataFrame({
    'Feature': features.columns,
    'Importance': feature_importances
})

# Sorts the DataFrame by importance
feature_importance_df = feature_importance_df.sort_values(by='Importance', ascending=False)

plt.figure(figsize=(12, 8))
plt.barh(feature_importance_df['Feature'], feature_importance_df['Importance'])
plt.xlabel('Feature Importance')
plt.ylabel('Features')
plt.title('Feature Importances from Gradient Boosting Classifier')
plt.gca().invert_yaxis()
plt.show()

feature_importance_df
```


Feature Importances from Gradient Boosting Classifier
Attrition_Flag_Existing Customer
Credit_Limit
Avg_Open_To_Buy
Total_Revolving_Bal
Total_Trans_Amt
Avg_Utilization_Ratio
Total_Ct_Chng_Q4_Q1
Income_Category_80K − 120K
Total_Trans_Ct
Total_Amt_Chng_Q4_Q1
Marital_Status_Single
Marital_Status_Unknown
Income_Category_$120K +
Income_Category_40K − 60K
Income_Category_60K − 80K
Customer_Age
Income_Category_Less than $40K
Marital_Status_Divorced
Income_Category_Unknown
Card_Category_Blue
Card_Category_Gold
Card_Category_Platinum
Marital_Status_Married
Education_Level_Graduate
Education_Level_Unknown
Education_Level_Uneducated
Education_Level_Post-Graduate
Education_Level_High School
Dependent_count
Education_Level_Doctorate
Education_Level_College
Gender_M
Gender_F
Contacts_Count_12_mon
Months_Inactive_12_mon
Total_Relationship_Count
Months_on_book
Card_Category_Silver
Features
0.0
0.2
0.4
0.6
0.8
1.0
Feature Importance

Out[20]:

	Feature	Importance
14	Attrition_Flag_Existing Customer	1.000000e+00
6	Credit_Limit	1.952855e-14
8	Avg_Open_To_Buy	1.445049e-14
7	Total_Revolving_Bal	6.301624e-15
10	Total_Trans_Amt	4.704759e-15
13	Avg_Utilization_Ratio	1.550176e-15
12	Total_Ct_Chng_Q4_Q1	1.515230e-15
31	Income_Category_$80K-$120K	9.332270e-16
11	Total_Trans_Ct	7.069132e-16
9	Total_Amt_Chng_Q4_Q1	1.506989e-18
26	Marital_Status_Single	0.000000e+00
27	Marital_Status_Unknown	0.000000e+00
28	Income_Category_$120K +	0.000000e+00
29	Income_Category_$40K-$60K	0.000000e+00
30	Income_Category_$60K-$80K	0.000000e+00
0	Customer_Age	0.000000e+00
32	Income_Category_Less than $40K	0.000000e+00
24	Marital_Status_Divorced	0.000000e+00
33	Income_Category_Unknown	0.000000e+00
34	Card_Category_Blue	0.000000e+00
35	Card_Category_Gold	0.000000e+00
36	Card_Category_Platinum	0.000000e+00
25	Marital_Status_Married	0.000000e+00
19	Education_Level_Graduate	0.000000e+00
23	Education_Level_Unknown	0.000000e+00
22	Education_Level_Uneducated	0.000000e+00
21	Education_Level_Post-Graduate	0.000000e+00

	Feature	Importance
20	Education_Level_High School	0.000000e+00
1	Dependent_count	0.000000e+00
18	Education_Level_Doctorate	0.000000e+00
17	Education_Level_College	0.000000e+00
16	Gender_M	0.000000e+00
15	Gender_F	0.000000e+00
5	Contacts_Count_12_mon	0.000000e+00
4	Months_Inactive_12_mon	0.000000e+00
3	Total_Relationship_Count	0.000000e+00
2	Months_on_book	0.000000e+00
37	Card_Category_Silver	0.000000e+00

3.7. Training a Logistic Regression Model

In [21]:

```
# Initialize the Logistic Regression model
logistic_model = LogisticRegression()

# Train the model using the training data
logistic_model.fit(features_train, target_train)
```

```
C:\Users\chris\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:458: ConvergenceWarning: l
bfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://scikit-learn.org/stable/module
s/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (https://scikit-lear
n.org/stable/modules/linear_model.html#logistic-regression)
  n_iter_i = _check_optimize_result(
```

Out[21]:

```
LogisticRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

3.8. Evaluating the logistic regression model

In [22]:

```
# Make predictions on the testing data
predictions = logistic_model.predict(features_test)

# Evaluate the model
accuracy = accuracy_score(target_test, predictions)
conf_matrix = confusion_matrix(target_test, predictions)
class_report = classification_report(target_test, predictions)

# Print the evaluation metrics
print(f"Accuracy: {accuracy}")
print("Confusion Matrix:")
print(conf_matrix)
print("Classification Report:")
print(class_report)
```

```
Accuracy: 0.8716683119447186
Confusion Matrix:
[[1627   72]
 [ 188  139]]
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.96      0.93      1699
           1       0.66      0.43      0.52       327

    accuracy                           0.87      2026
   macro avg       0.78      0.69      0.72      2026
weighted avg       0.86      0.87      0.86      2026
```

3.9. Looking for feature importance

In [23]:
```
# Gets the coefficients of the features
coefficients = logistic_model.coef_[0]

# Creates a DataFrame to display feature importance
feature_importance = pd.DataFrame({
    'Feature': features.columns,
    'Coefficient': coefficients
})

# Sorts the DataFrame by the absolute value of the coefficient
feature_importance['Abs_Coefficient'] = np.abs(feature_importance['Coefficient'])
feature_importance = feature_importance.sort_values(by='Abs_Coefficient', ascending=False)

# Prints the feature importance
print("Feature Importance in Logistic Regression Model:")
print(feature_importance[['Feature', 'Coefficient']])
```

```
Feature Importance in Logistic Regression Model:
                              Feature  Coefficient
14  Attrition_Flag_Existing Customer    -0.388041
5              Contacts_Count_12_mon     0.321220
4             Months_Inactive_12_mon     0.260960
3           Total_Relationship_Count    -0.214982
1                    Dependent_count     0.190291
0                       Customer_Age     0.177105
2                     Months_on_book    -0.161208
11                    Total_Trans_Ct    -0.101106
15                          Gender_F     0.074435
12               Total_Ct_Chng_Q4_Q1    -0.057723
26             Marital_Status_Single     0.048810
32    Income_Category_Less than $40K     0.042639
16                          Gender_M    -0.041239
25            Marital_Status_Married    -0.032527
34                Card_Category_Blue     0.026500
33           Income_Category_Unknown     0.015733
30       Income_Category_$60K - $80K    -0.014250
9                Total_Amt_Chng_Q4_Q1    -0.012099
23           Education_Level_Unknown     0.011429
18         Education_Level_Doctorate     0.011191
24           Marital_Status_Divorced     0.011011
21     Education_Level_Post-Graduate     0.006752
19          Education_Level_Graduate     0.006225
27            Marital_Status_Unknown     0.005901
31      Income_Category_$80K - $120K    -0.005827
35                Card_Category_Gold     0.004058
17           Education_Level_College    -0.003474
29       Income_Category_$40K - $60K    -0.003427
22        Education_Level_Uneducated     0.002600
13             Avg_Utilization_Ratio     0.002496
28           Income_Category_$120K +    -0.001672
20       Education_Level_High School    -0.001527
36            Card_Category_Platinum     0.001468
37              Card_Category_Silver     0.001171
7                Total_Revolving_Bal    -0.000640
10                   Total_Trans_Amt     0.000394
6                       Credit_Limit    -0.000323
8                      Avg_Open_To_Buy     0.000317
```

In [ ]: