Bank Customer Churn Analysis

In the rapidly evolving banking sector, customer retention has become a critical concern. Banks are increasingly seeking to understand the factors that influence customer decisions to stay with or leave their banking service provider. This project focuses on analyzing a dataset containing various attributes of bank customers to identify key predictors of customer churn. By leveraging data analytics, we aim to uncover patterns and insights that could help devise strategies to enhance customer retention and reduce churn rates.

In [301…

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
```

In [302…

```
df = pd.read_csv('Bank-Records.csv')
```

In [303…

```
df
```

Out[303]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	Esti
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	
9995	9996	15606229	Obijiaku	771	France	Male	39	5	0.00	2	1	0	
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	2	1	0	
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1	0	

10000 rows × 18 columns

In [304…

```
df.shape
```

Out[304]:

```
(10000, 18)
```

Dataset contains 10000 rows and 18 columns

In [305…

```
df.dtypes
```

Out[305]:

```
RowNumber              int64
CustomerId             int64
Surname               object
CreditScore            int64
Geography             object
Gender                object
Age                    int64
Tenure                 int64
Balance              float64
NumOfProducts          int64
HasCrCard              int64
IsActiveMember         int64
EstimatedSalary      float64
Exited                 int64
Complain               int64
Satisfaction Score     int64
Card Type             object
Point Earned           int64
dtype: object
```

Dataset moslty have categorical variables and 5 continuous variables

In [306…

```
df.isnull().any()
```

Out[306]:

```
RowNumber             False
CustomerId            False
Surname               False
CreditScore           False
Geography             False
Gender                False
Age                   False
Tenure                False
Balance               False
NumOfProducts         False
HasCrCard             False
IsActiveMember        False
EstimatedSalary       False
Exited                False
Complain              False
Satisfaction Score    False
Card Type             False
Point Earned          False
dtype: bool
```

There is no null values.

In [307…

```
df.nunique()
```

Out[307]:

```
RowNumber             10000
CustomerId            10000
Surname                2932
CreditScore             460
Geography                 3
Gender                    2
Age                      70
Tenure                   11
Balance                6382
NumOfProducts             4
HasCrCard                 2
IsActiveMember            2
EstimatedSalary        9999
Exited                    2
Complain                  2
Satisfaction Score        5
Card Type                 4
Point Earned            785
dtype: int64
```

Droping Data

In [308…

```
df = df.drop(['RowNumber', 'CustomerId', 'Surname'], axis = 1)
df.head()
```

Out[308]:

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Complain	Satisfacti Scc
0	619	France	Female	42	2	0.00	1	1	1	101348.88	1	1	
1	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0	1	
2	502	France	Female	42	8	159660.80	3	1	0	113931.57	1	1	
3	699	France	Female	39	1	0.00	2	0	0	93826.63	0	0	
4	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0	0	

We won't need the first two attributes because they're specific to each customer. As for the surname, we'll leave it out too, just to keep things more general.

In [309…
```
df.describe()
```

Out[309]:

	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Comp
count	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.00000	10000.000000	10000.000000	10000.000000	10000.00(
mean	650.528800	38.921800	5.012800	76485.889288	1.530200	0.70550	0.515100	100090.239881	0.203800	0.204
std	96.653299	10.487806	2.892174	62397.405202	0.581654	0.45584	0.499797	57510.492818	0.402842	0.40:
min	350.000000	18.000000	0.000000	0.000000	1.000000	0.00000	0.000000	11.580000	0.000000	0.00(
25%	584.000000	32.000000	3.000000	0.000000	1.000000	0.00000	0.000000	51002.110000	0.000000	0.00(
50%	652.000000	37.000000	5.000000	97198.540000	1.000000	1.00000	1.000000	100193.915000	0.000000	0.00(
75%	718.000000	44.000000	7.000000	127644.240000	2.000000	1.00000	1.000000	149388.247500	0.000000	0.00(
max	850.000000	92.000000	10.000000	250898.090000	4.000000	1.00000	1.000000	199992.480000	1.000000	1.00(

- Average age of customers is approximately 39 years, with most customers aged between 32 and 44 years.
- Average satisfaction score is 3.01 on a scale of 1 to 5.

In [310…
```
df.describe(include=['object'])
```

Out[310]:

	Geography	Gender	Card Type
count	10000	10000	10000
unique	3	2	4
top	France	Male	DIAMOND
freq	5014	5457	2507

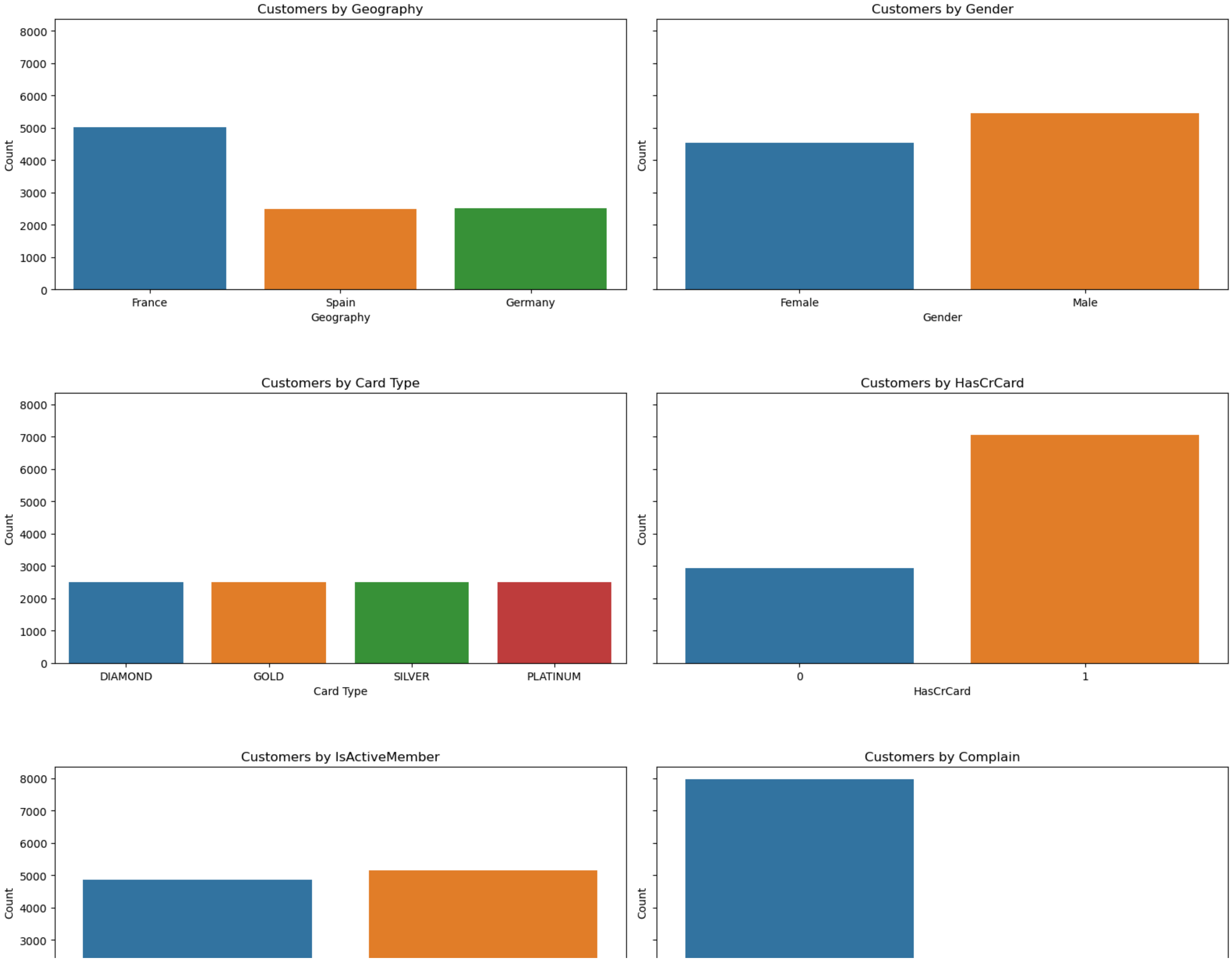
From the categorical data, we can observe that the majority of the customers are from France, most users are male, and a significant number of them hold diamond cards.

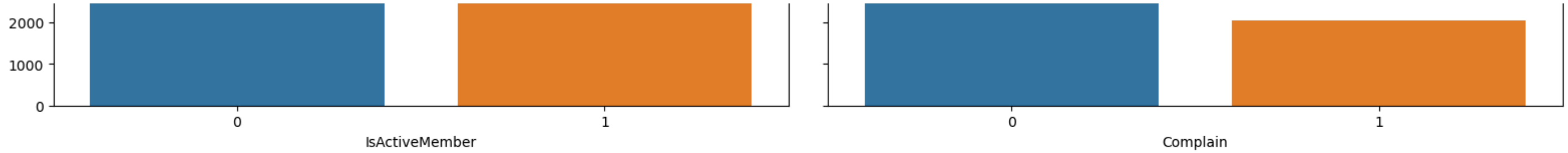
Exploratory Data Analysis (EDA)

In [311…

```
categorical_vars = ['Geography', 'Gender', 'Card Type', 'HasCrCard', 'IsActiveMember', 'Complain']
fig, axes = plt.subplots(3, 2, figsize=(16, 14), sharey=True)
axes = axes.flatten()

for ax, var in zip(axes, categorical_vars):
    sns.countplot(x=var, data=df, ax=ax)
    ax.set_title(f'Customers by {var}')
    ax.set_xlabel(var)
    ax.set_ylabel('Count')
plt.tight_layout(h_pad=5)
plt.show()
```


Customers by Geography
Count
8000
7000
6000
5000
4000
3000
2000
1000
0
France
Spain
Germany
Geography
Customers by Gender
Count
Female
Male
Gender
Customers by Card Type
Count
8000
7000
6000
5000
4000
3000
2000
1000
0
DIAMOND
GOLD
SILVER
PLATINUM
Card Type
Customers by HasCrCard
Count
0
1
HasCrCard
Customers by IsActiveMember
Count
8000
7000
6000
5000
4000
3000
Customers by Complain
Count

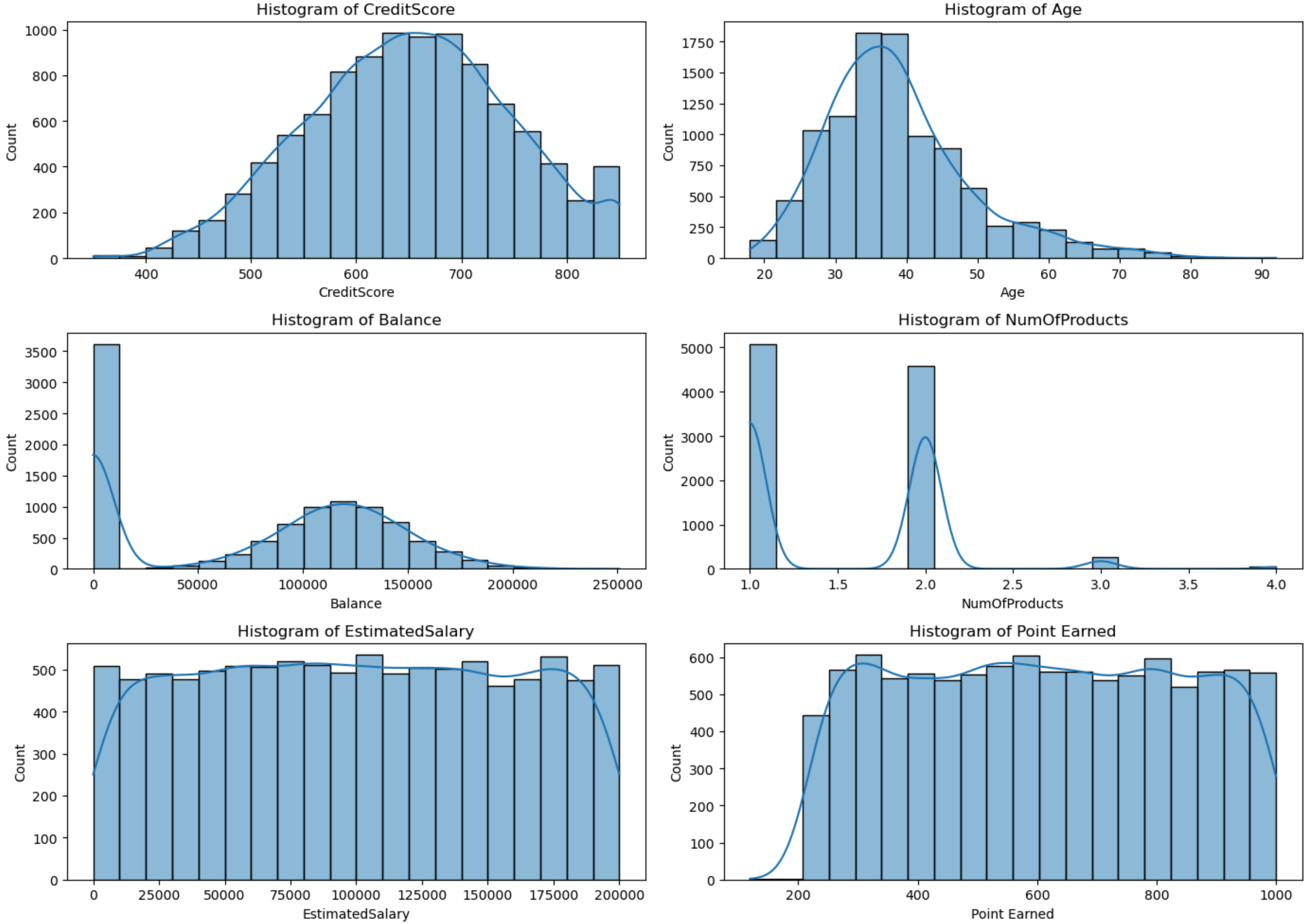


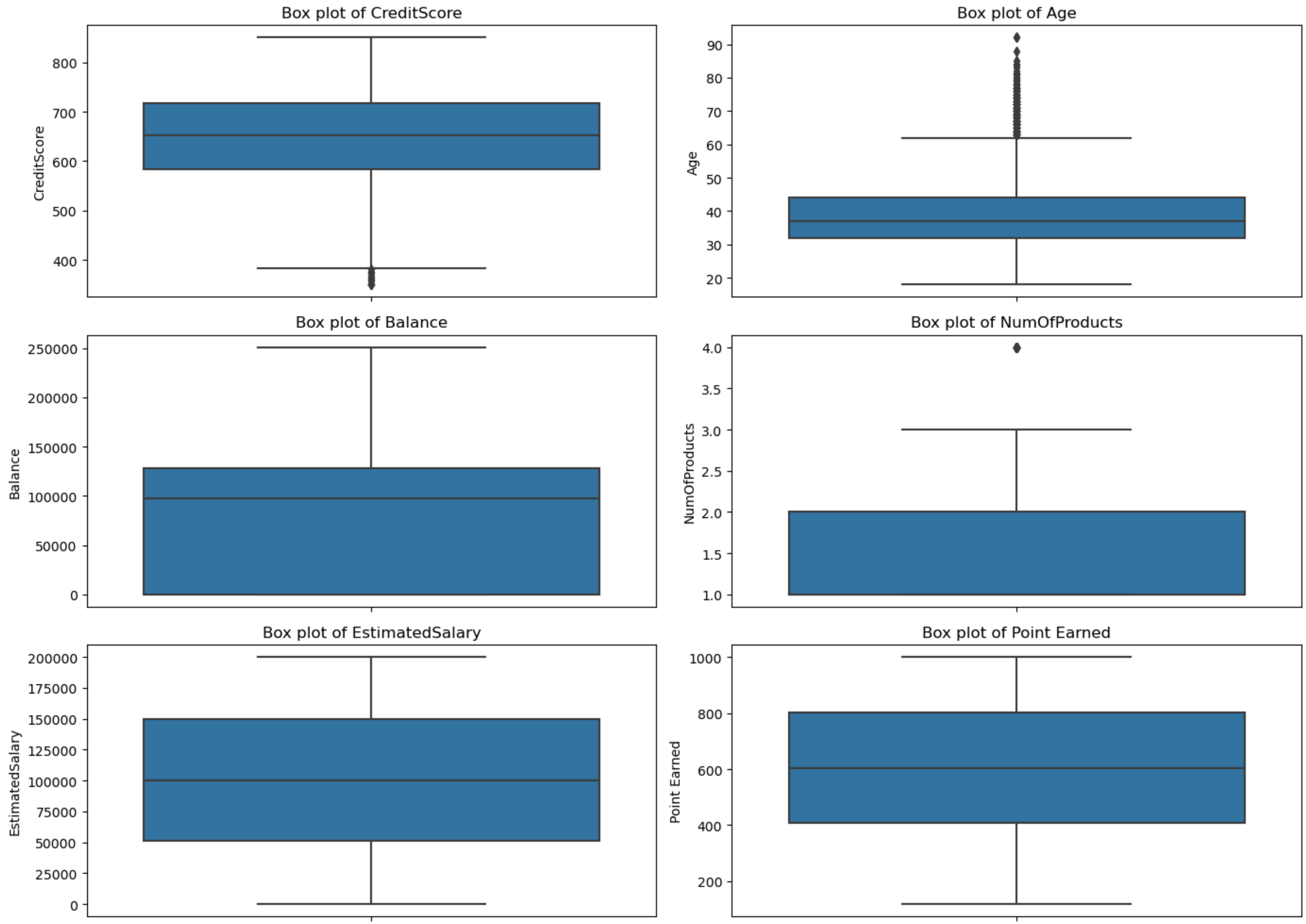
- Most customers are from France.
- More male customers compared to female customers.
- Many customers have Credit cards.
- There is only slight difference in Active and Inactive members.
- Most of the Customer have no Complain.

In [312…

```
numerical_columns = ['CreditScore', 'Age', 'Balance', 'NumOfProducts', 'EstimatedSalary', 'Point Earned']
plt.figure(figsize=(14, 10))
for i, column in enumerate(numerical_columns, 1):
    plt.subplot(3, 2, i)
    sns.histplot(df[column], bins=20, kde=True)
    plt.title(f'Histogram of {column}')
plt.tight_layout()
plt.show()


plt.figure(figsize=(14, 10))
for i, column in enumerate(numerical_columns, 1):
    plt.subplot(3, 2, i)
    sns.boxplot(y=df[column])
    plt.title(f'Box plot of {column}')
plt.tight_layout()
plt.show()
```


Histogram of CreditScore
Count
CreditScore
Histogram of Age
Count
Age
Histogram of Balance
Count
Balance
Histogram of NumOfProducts
Count
NumOfProducts
Histogram of EstimatedSalary
Count
EstimatedSalary
Histogram of Point Earned
Count
Point Earned

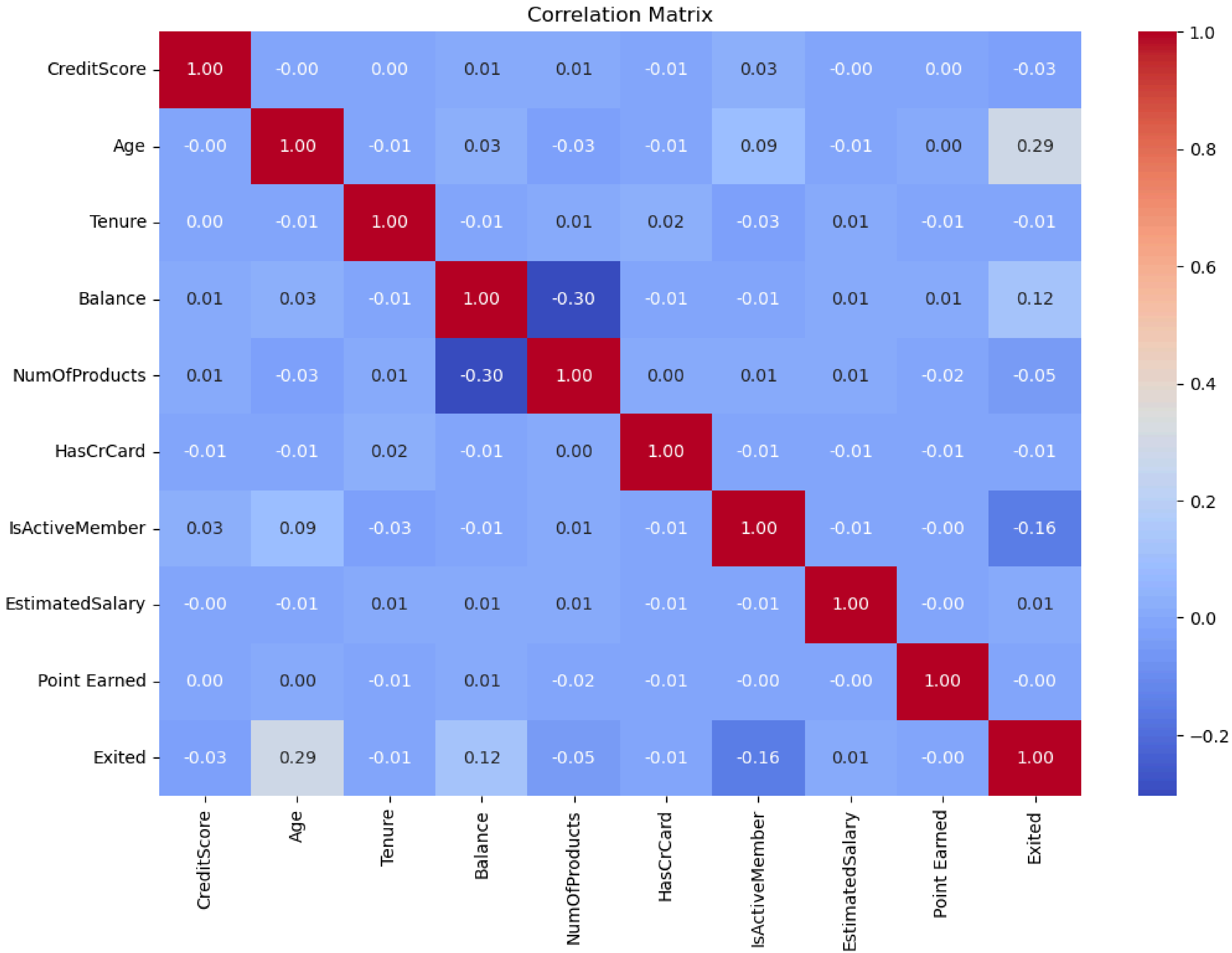


- Most customers fall within the credit score range of 650-700.

- Majority of customers are in the age group of 30-40.
- Most customers fall within the balance range of 100,000 and some of the customers have 0 balance.
- Most customers hold product 1 and 2.
- Most customers estimated salary is between 50000 to 150000.
- Majority of customers earn points in the range of 400 to 800.

In [313…

```
numerical_df = df[['CreditScore', 'Age', 'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard', 'IsActiveMember', 'EstimatedSalary',
correlation_matrix = numerical_df.corr()
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Matrix')
plt.show()
```

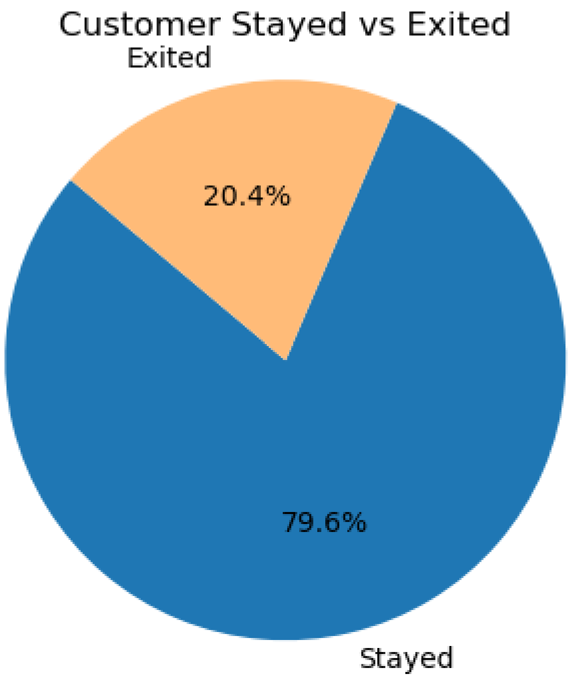

Correlation Matrix
CreditScore
Age
Tenure
Balance
NumOfProducts
HasCrCard
IsActiveMember
EstimatedSalary
Point Earned
Exited
1.00 -0.00 0.00 0.01 0.01 -0.01 0.03 -0.00 0.00 -0.03
-0.00 1.00 -0.01 0.03 -0.03 -0.01 0.09 -0.01 0.00 0.29
0.00 -0.01 1.00 -0.01 0.01 0.02 -0.03 0.01 -0.01 -0.01
0.01 0.03 -0.01 1.00 -0.30 -0.01 -0.01 0.01 0.01 0.12
0.01 -0.03 0.01 -0.30 1.00 0.00 0.01 0.01 -0.02 -0.05
-0.01 -0.01 0.02 -0.01 0.00 1.00 -0.01 -0.01 -0.01 -0.01
0.03 0.09 -0.03 -0.01 0.01 -0.01 1.00 -0.01 -0.00 -0.16
-0.00 -0.01 0.01 0.01 0.01 -0.01 -0.01 1.00 -0.00 0.01
0.00 0.00 -0.01 0.01 -0.02 -0.01 -0.00 -0.00 1.00 -0.00
-0.03 0.29 -0.01 0.12 -0.05 -0.01 -0.16 0.01 -0.00 1.00
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

- Overall, the correlations are very weak. There's a weak positive correlation with age, a very weak positive correlation with balance, and very weak negative correlations with the number of products and membership.

In [314…

```
exited_counts = df['Exited'].value_counts()
labels = 'Stayed', 'Exited'
sizes = exited_counts[0], exited_counts[1]
colors = ['#1f77b4', '#ffbb78']

plt.figure(figsize=(7, 4))
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=140)
plt.title('Customer Stayed vs Exited')
plt.axis('equal')
plt.show()
```



- 20% of the customers have churned and 80% of the customers retained.

Customer Churn Analysis by Gender

In [315…

```
gender_exit_counts = pd.crosstab(df['Gender'], df['Exited'])
gender_exit_counts
```

Out[315]:

Exited	0	1
Gender		
Female	3404	1139
Male	4558	899

In [316…

```
gender_exit_percentages = pd.crosstab(df['Gender'], df['Exited'], normalize='index').round(2) * 100
gender_exit_percentages
```

Out[316]:

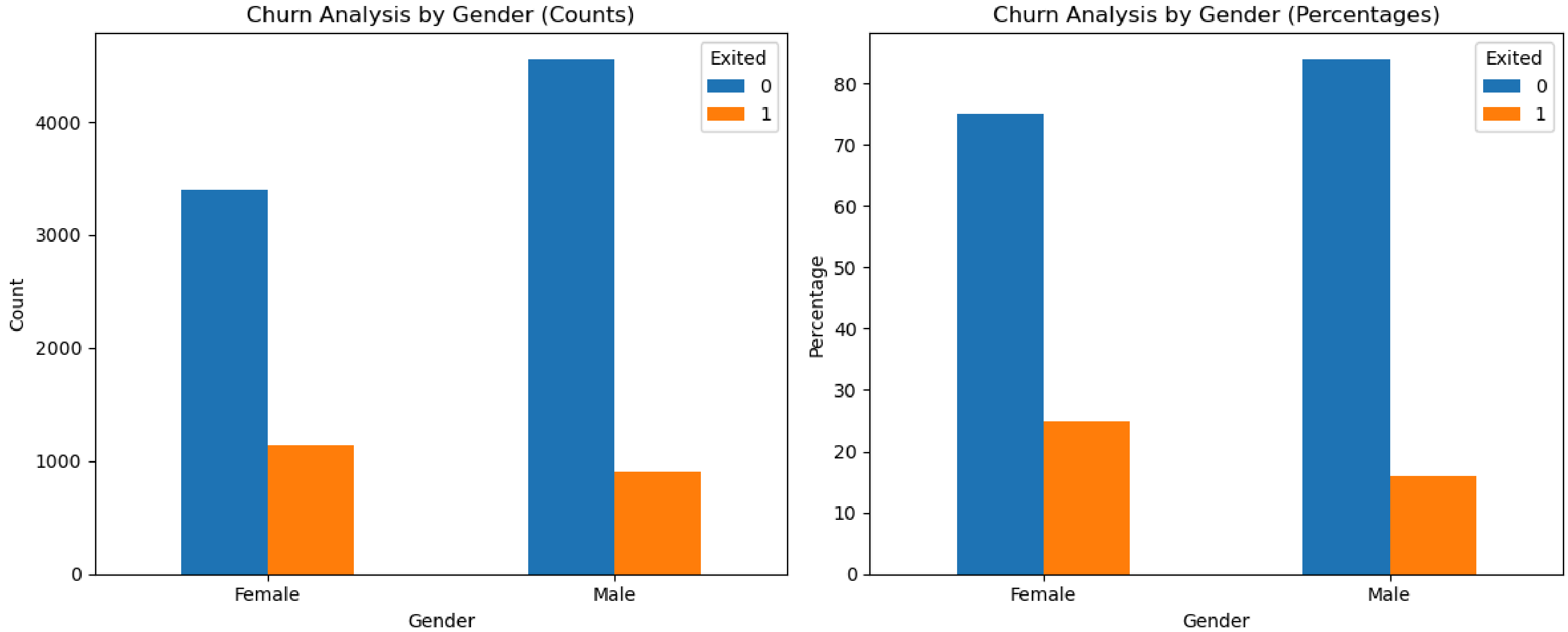
Exited	0	1
Gender		
Female	75.0	25.0
Male	84.0	16.0

In [317…

```
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(12, 5))
gender_exit_counts.plot(kind='bar', stacked=False, ax=axes[0])
axes[0].set_title('Churn Analysis by Gender (Counts)')
axes[0].set_xlabel('Gender')
axes[0].set_ylabel('Count')
axes[0].legend(title='Exited')
axes[0].grid(False)

gender_exit_percentages.plot(kind='bar', stacked=False, ax=axes[1])
axes[1].set_title('Churn Analysis by Gender (Percentages)')
axes[1].set_xlabel('Gender')
axes[1].set_ylabel('Percentage')
axes[1].legend(title='Exited')
axes[1].grid(False)

plt.setp(axes[0].xaxis.get_majorticklabels(), rotation=0)
plt.setp(axes[1].xaxis.get_majorticklabels(), rotation=0)
plt.tight_layout()
plt.show()
```



- It appears that 25% of female customers churn, compared to 16% of male customers.

Customer Churn Analysis by Geography

In [318…

```
geography_exit_counts = pd.crosstab(df['Geography'], df['Exited'])
geography_exit_counts
```

Out[318]:

Exited	0	1
Geography		
France	4203	811
Germany	1695	814
Spain	2064	413

In [319…

```
geography_exit_percentages = pd.crosstab(df['Geography'], df['Exited'], normalize='index').round(2) * 100
geography_exit_percentages
```

Out[319]:

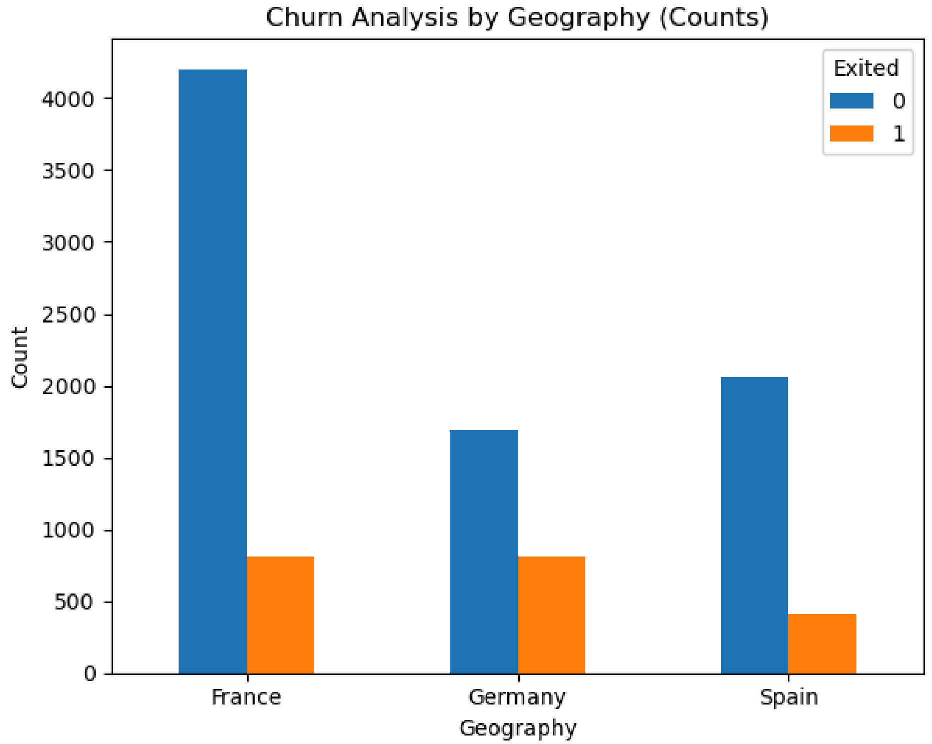
Exited	0	1
Geography		
France	84.0	16.0
Germany	68.0	32.0
Spain	83.0	17.0

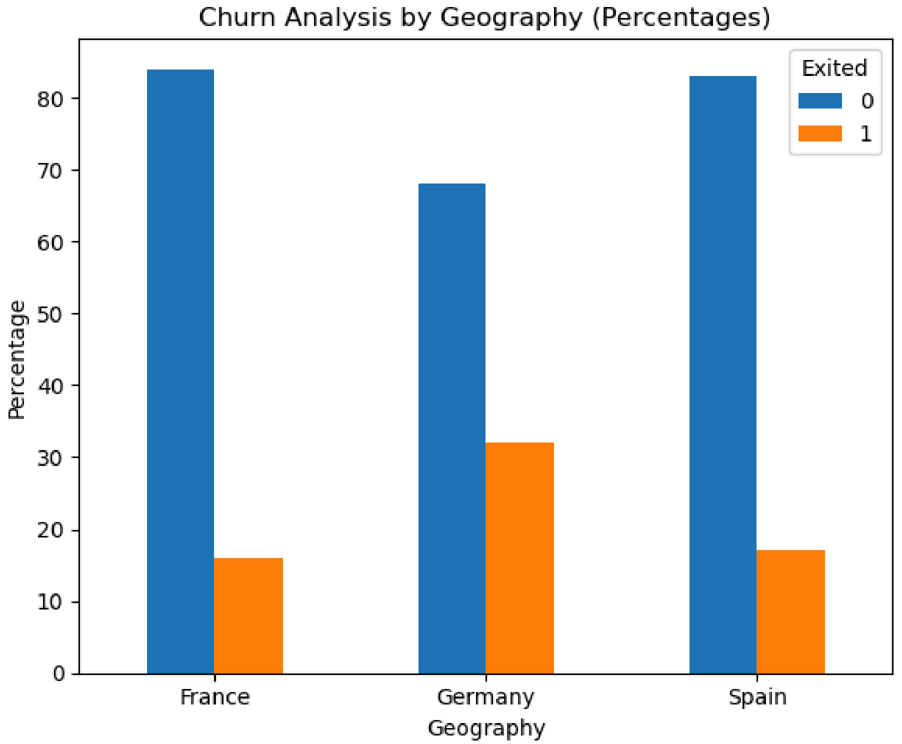
In [320…

```
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(12, 5))
geography_exit_counts.plot(kind='bar', stacked=False, ax=axes[0])
axes[0].set_title('Churn Analysis by Geography (Counts)')
axes[0].set_xlabel('Geography')
axes[0].set_ylabel('Count')
axes[0].legend(title='Exited')
axes[0].grid(False)

geography_exit_percentages.plot(kind='bar', stacked=False, ax=axes[1])
axes[1].set_title('Churn Analysis by Geography (Percentages)')
axes[1].set_xlabel('Geography')
axes[1].set_ylabel('Percentage')
axes[1].legend(title='Exited')
axes[1].grid(False)

plt.setp(axes[0].xaxis.get_majorticklabels(), rotation=0)
plt.setp(axes[1].xaxis.get_majorticklabels(), rotation=0)
plt.tight_layout()
plt.show()
```





- It seems that Germany has the smallest number of customers, but they are the ones who left the most among all three geographical regions.

Customer Churn Analysis by Credit Card

In [321…

```
hasCrCard_exit_counts = pd.crosstab(df['HasCrCard'], df['Exited'])
hasCrCard_exit_counts
```

Out[321]:

Exited	0	1
HasCrCard		
0	2332	613
1	5630	1425

In [322…

```
hasCrCard_exit_percentages = pd.crosstab(df['HasCrCard'], df['Exited'], normalize='index').round(2) * 100
hasCrCard_exit_percentages
```

Out[322]:

Exited	0	1
HasCrCard		
0	79.0	21.0
1	80.0	20.0

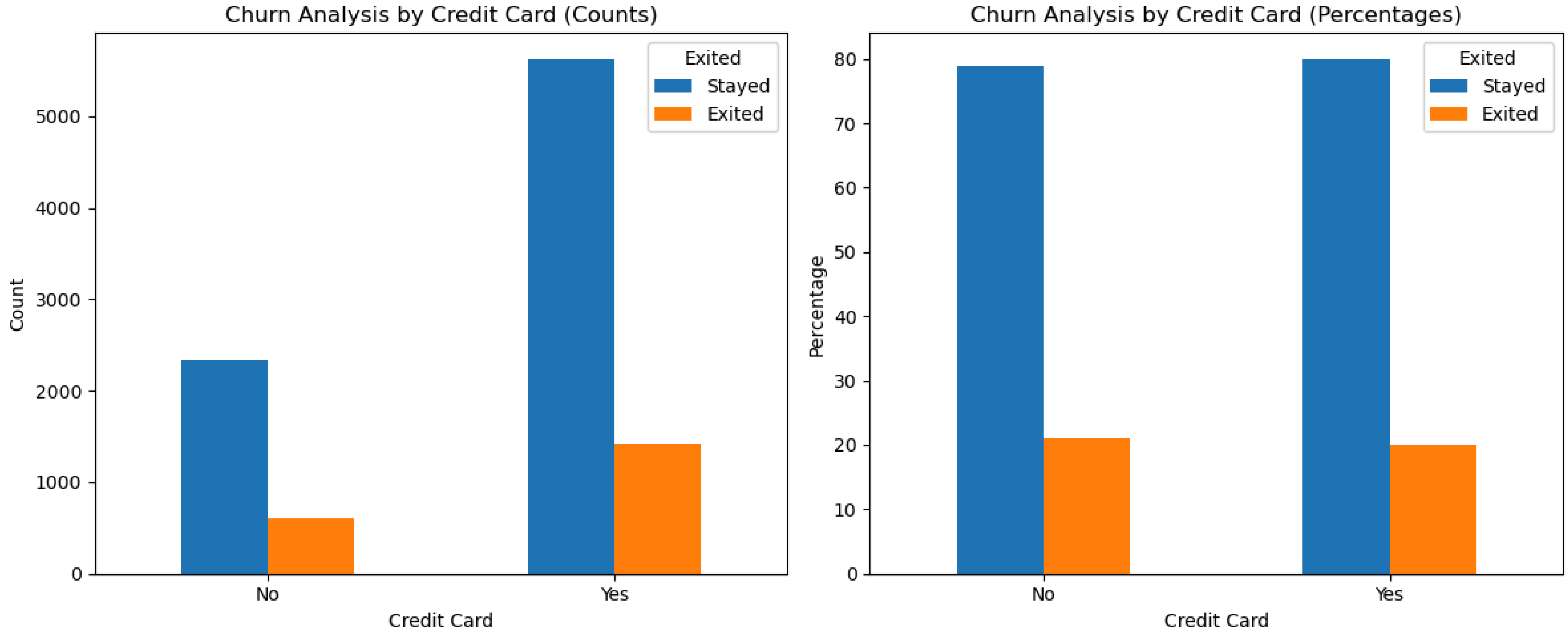
In [323…

```
hasCrCard_exit_counts = pd.crosstab(df['HasCrCard'].map({1: 'Yes', 0: 'No'}), df['Exited'])
hasCrCard_exit_percentages = pd.crosstab(df['HasCrCard'].map({1: 'Yes', 0: 'No'}), df['Exited'], normalize='index').round(2) * 10

fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(12, 5))
hasCrCard_exit_counts.plot(kind='bar', stacked=False, ax=axes[0])
axes[0].set_title('Churn Analysis by Credit Card (Counts)')
axes[0].set_xlabel('Credit Card')
axes[0].set_ylabel('Count')
axes[0].legend(['Stayed', 'Exited'], title='Exited')
axes[0].grid(False)

hasCrCard_exit_percentages.plot(kind='bar', stacked=False, ax=axes[1])
axes[1].set_title('Churn Analysis by Credit Card (Percentages)')
axes[1].set_xlabel('Credit Card')
axes[1].set_ylabel('Percentage')
axes[1].legend(['Stayed', 'Exited'], title='Exited')
axes[1].grid(False)

plt.setp(axes[0].xaxis.get_majorticklabels(), rotation=0)
plt.setp(axes[1].xaxis.get_majorticklabels(), rotation=0)
plt.tight_layout()
plt.show()
```



- Looks like majority of customers with credit card have churned.

Customer Churn Analysis by Products

In [324…

```
numOfProducts_exit_counts = pd.crosstab(df['NumOfProducts'].astype(str), df['Exited'])
numOfProducts_exit_counts
```

Out[324]:

Exited	0	1
NumOfProducts		
1	3675	1409
2	4241	349
3	46	220
4	0	60

In [325…

```
numOfProducts_exit_percentages = pd.crosstab(df['NumOfProducts'].astype(str), df['Exited'], normalize='index').round(2) * 100
numOfProducts_exit_percentages
```

Out[325]:

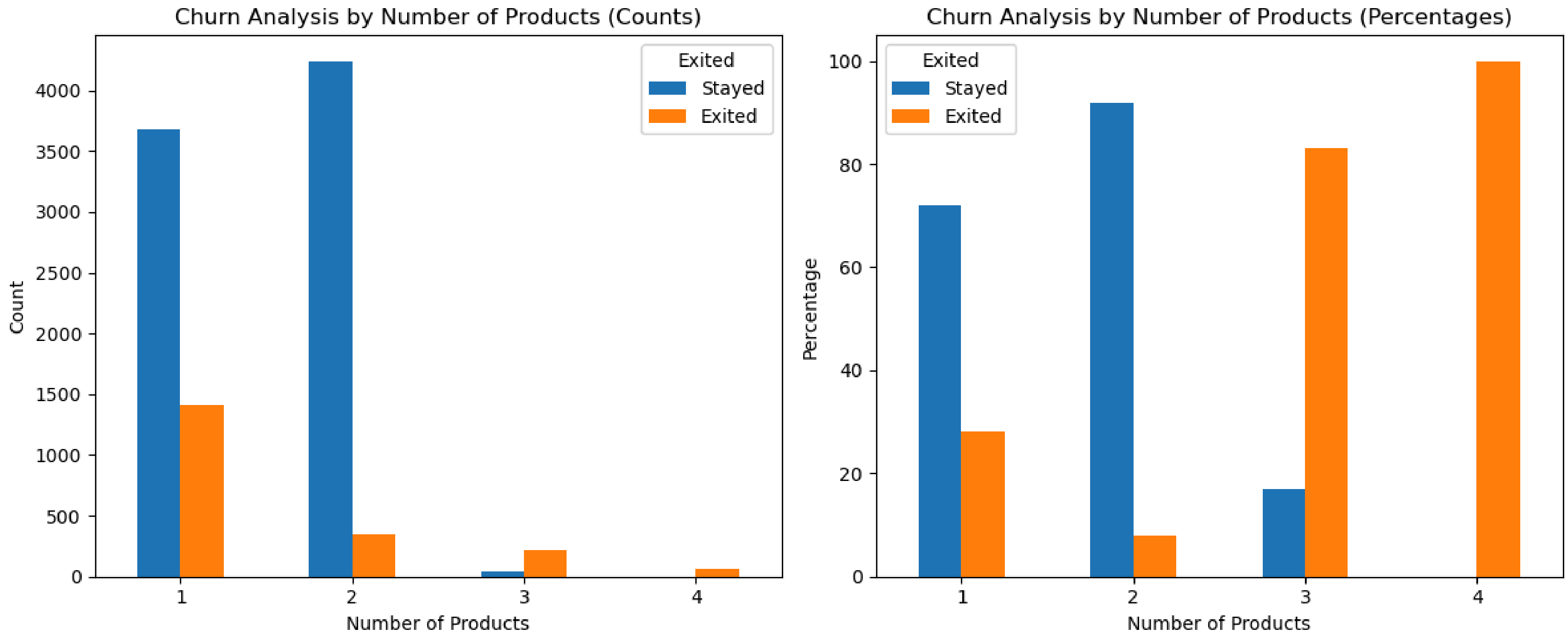
Exited	0	1
NumOfProducts		
1	72.0	28.0
2	92.0	8.0
3	17.0	83.0
4	0.0	100.0

In [326…

```
numOfProducts_exit_counts = pd.crosstab(df['NumOfProducts'].astype(str), df['Exited'])
numOfProducts_exit_percentages = pd.crosstab(df['NumOfProducts'].astype(str), df['Exited'], normalize='index').round(2) * 100
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(12, 5))
numOfProducts_exit_counts.plot(kind='bar', stacked=False, ax=axes[0])
axes[0].set_title('Churn Analysis by Number of Products (Counts)')
axes[0].set_xlabel('Number of Products')
axes[0].set_ylabel('Count')
axes[0].legend(['Stayed', 'Exited'], title='Exited')
axes[0].grid(False)

numOfProducts_exit_percentages.plot(kind='bar', stacked=False, ax=axes[1])
axes[1].set_title('Churn Analysis by Number of Products (Percentages)')
axes[1].set_xlabel('Number of Products')
axes[1].set_ylabel('Percentage')
axes[1].legend(['Stayed', 'Exited'], title='Exited')
axes[1].grid(False)
```

```
plt.setp(axes[0].xaxis.get_majorticklabels(), rotation=0)
plt.setp(axes[1].xaxis.get_majorticklabels(), rotation=0)
plt.tight_layout()
plt.show()
```



- There's a significant drop in the count of Stayed customers when moving from 1 product to 4 products.
- Customers with product 4 have a much higher churn rate or likelihood to exit.

Customer Churn Analysis by Active Members

In [327…

```
isActiveMember_exit_counts = pd.crosstab(df['IsActiveMember'].map({1: 'Yes', 0: 'No'}), df['Exited'])
isActiveMember_exit_counts
```

Out[327]:

Exited	0	1
IsActiveMember		
No	3546	1303
Yes	4416	735

In [328...

```
isActiveMember_exit_percentages = pd.crosstab(df['IsActiveMember'].map({1: 'Yes', 0: 'No'}), df['Exited'], normalize='index').rou
isActiveMember_exit_percentages
```

Out[328]:

Exited	0	1
IsActiveMember		
No	73.0	27.0
Yes	86.0	14.0

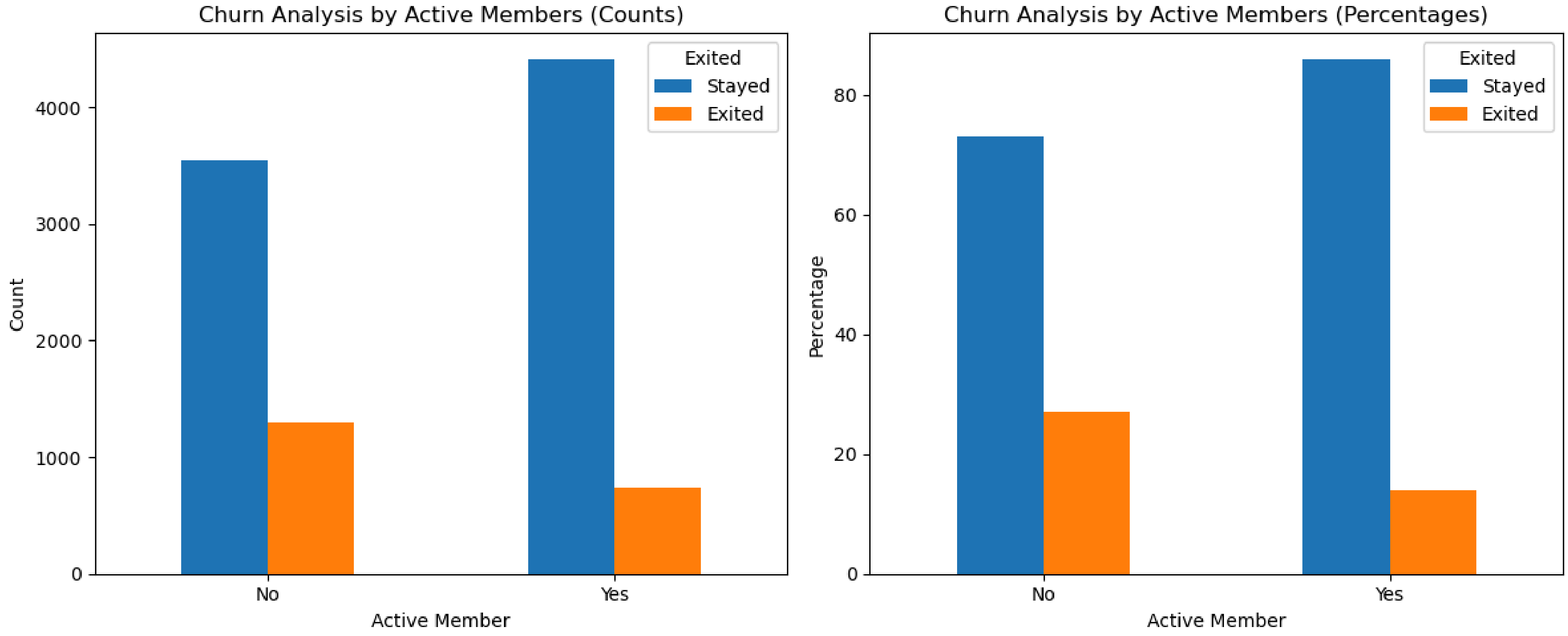
In [329...

```
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(12, 5))

isActiveMember_exit_counts.plot(kind='bar', stacked=False, ax=axes[0])
axes[0].set_title('Churn Analysis by Active Members (Counts)')
axes[0].set_xlabel('Active Member')
axes[0].set_ylabel('Count')
axes[0].legend(['Stayed', 'Exited'], title='Exited')
axes[0].grid(False)

isActiveMember_exit_percentages.plot(kind='bar', stacked=False, ax=axes[1])
axes[1].set_title('Churn Analysis by Active Members (Percentages)')
axes[1].set_xlabel('Active Member')
axes[1].set_ylabel('Percentage')
axes[1].legend(['Stayed', 'Exited'], title='Exited')
axes[1].grid(False)

plt.setp(axes[0].xaxis.get_majorticklabels(), rotation=0)
plt.setp(axes[1].xaxis.get_majorticklabels(), rotation=0)
plt.tight_layout()
plt.show()
```

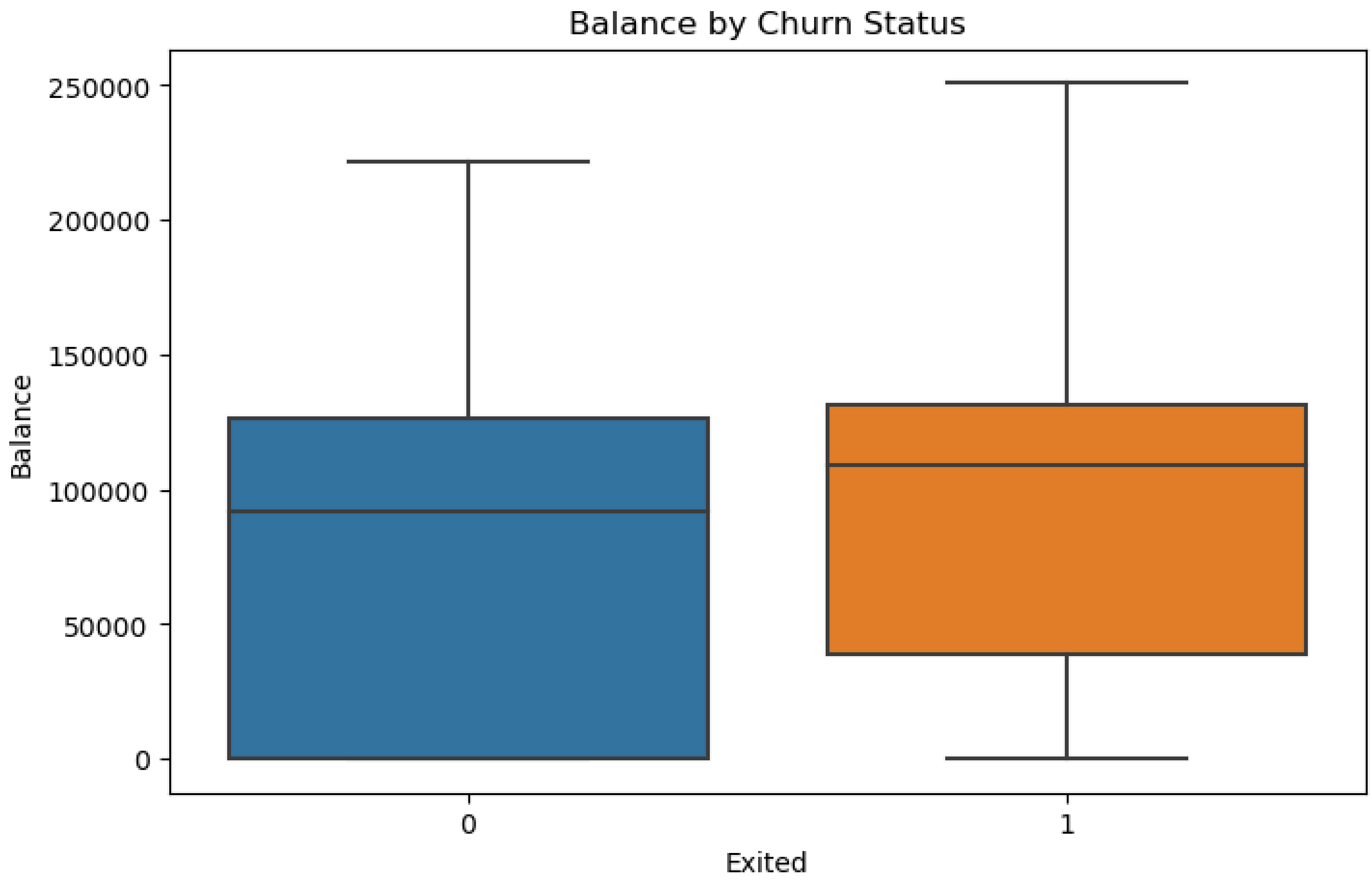


- Inactive members have a greater churn and active member is associated with lower churn.

Customer Churn Analysis by Balance

In [330…

```
plt.figure(figsize=(8, 5))
sns.boxplot(x='Exited', y='Balance', data=df)
plt.title('Balance by Churn Status')
plt.xlabel('Exited')
plt.ylabel('Balance')
plt.show()
```



- Both churned and stayed customers have a similar median balance, around 100,000. This suggests that the central tendency of account balances is similar for both groups.

Behavioural Analysis

Customer Churn Analysis by Complain

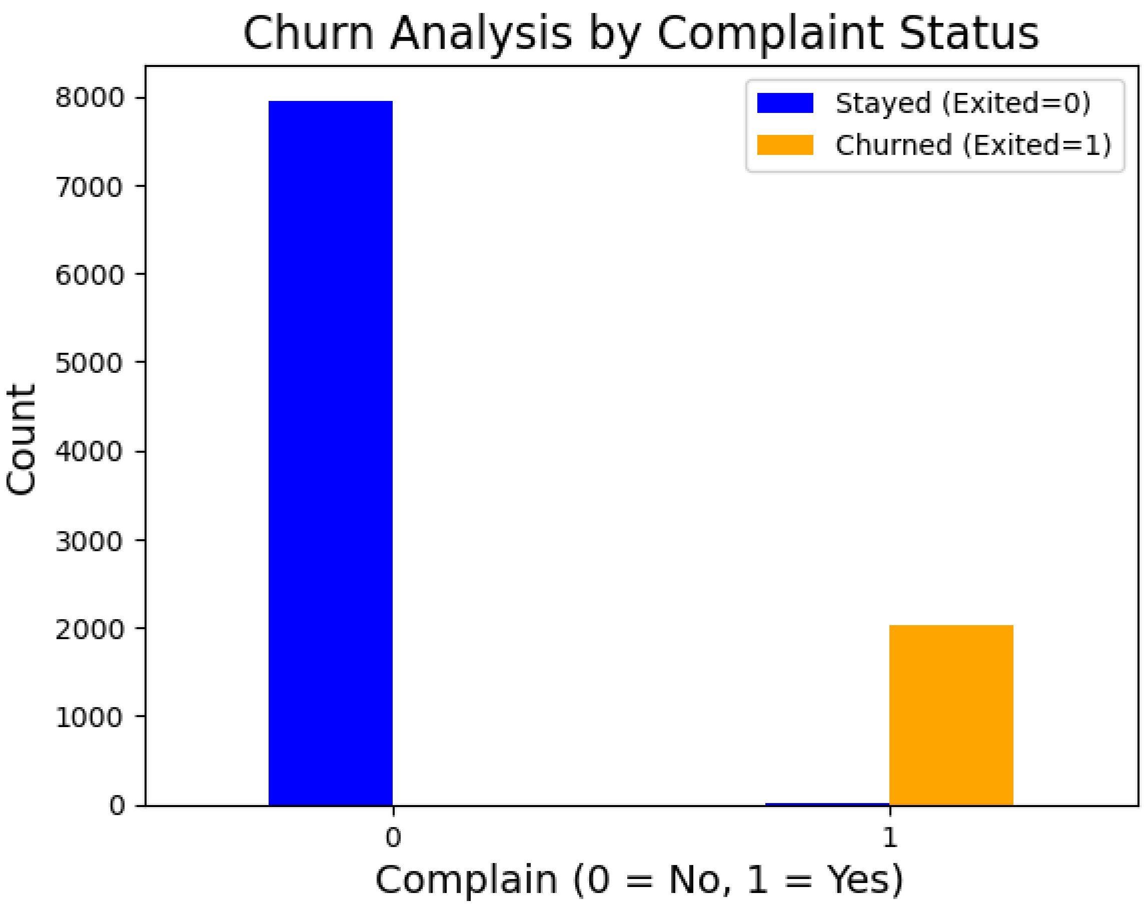
In [331…

```
complaint_churn = df.groupby(['Complain', 'Exited']).size().unstack(fill_value=0).reset_index()

plt.figure(figsize=(10, 6))
complaint_churn.plot(kind='bar', x='Complain', stacked=False, color=['blue', 'orange'])
```

```
plt.title('Churn Analysis by Complaint Status', fontsize=16)
plt.xlabel('Complain (0 = No, 1 = Yes)', fontsize=14)
plt.ylabel('Count', fontsize=14)
plt.legend(['Stayed (Exited=0)', 'Churned (Exited=1)'])
plt.xticks(rotation=0)
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```



- Most bank customers that complained were churned.

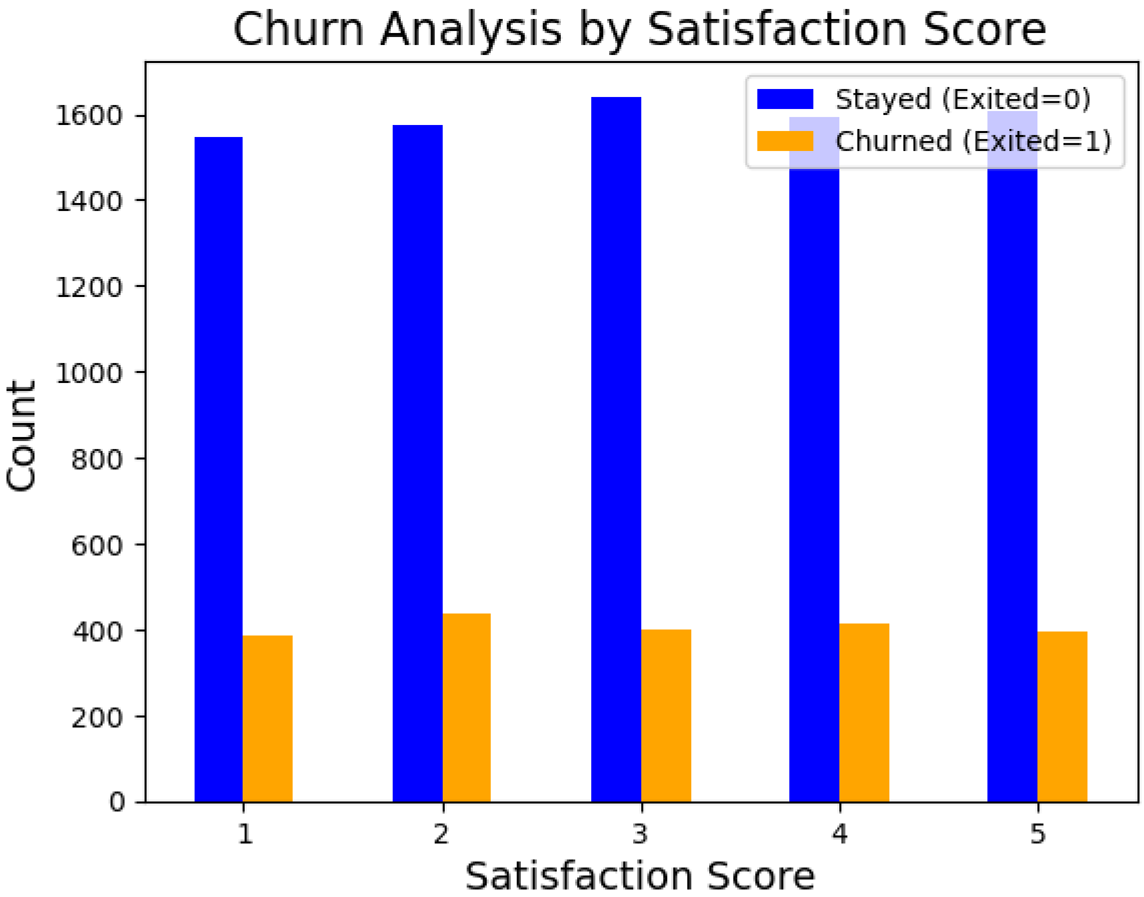
Customer Churn Analysis by Satisfaction Score

In [332...

```
satisfaction_churn = df.groupby(['Satisfaction Score', 'Exited']).size().unstack(fill_value=0).reset_index()

plt.figure(figsize=(10, 6))
satisfaction_churn.plot(kind='bar', x='Satisfaction Score', stacked=False, color=['blue', 'orange'])
plt.title('Churn Analysis by Satisfaction Score', fontsize=16)
plt.xlabel('Satisfaction Score', fontsize=14)
plt.ylabel('Count', fontsize=14)
plt.legend(['Stayed (Exited=0)', 'Churned (Exited=1)'])
plt.xticks(rotation=0)
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```



- Satisfaction scores almost have similar churn level.

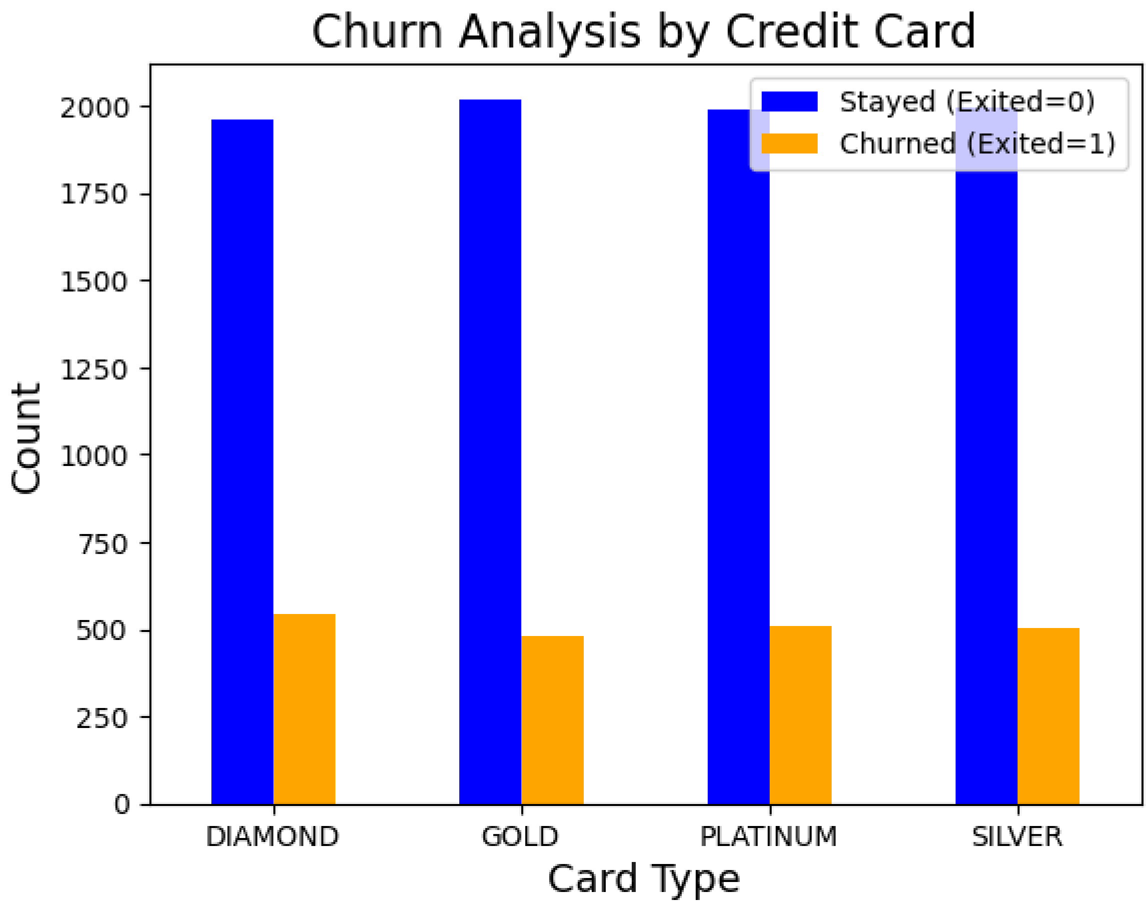
Customer Churn Analysis by Card Type

In [333…

```
card_type_churn = df.groupby(['Card Type', 'Exited']).size().unstack(fill_value=0).reset_index()

plt.figure(figsize=(10, 6))
card_type_churn.plot(kind='bar', x='Card Type', stacked=False, color=['blue', 'orange'])
plt.title('Churn Analysis by Credit Card', fontsize=16)
plt.xlabel('Card Type', fontsize=14)
plt.ylabel('Count', fontsize=14)
plt.legend(['Stayed (Exited=0)', 'Churned (Exited=1)'])
plt.xticks(rotation=0)
plt.show()
```

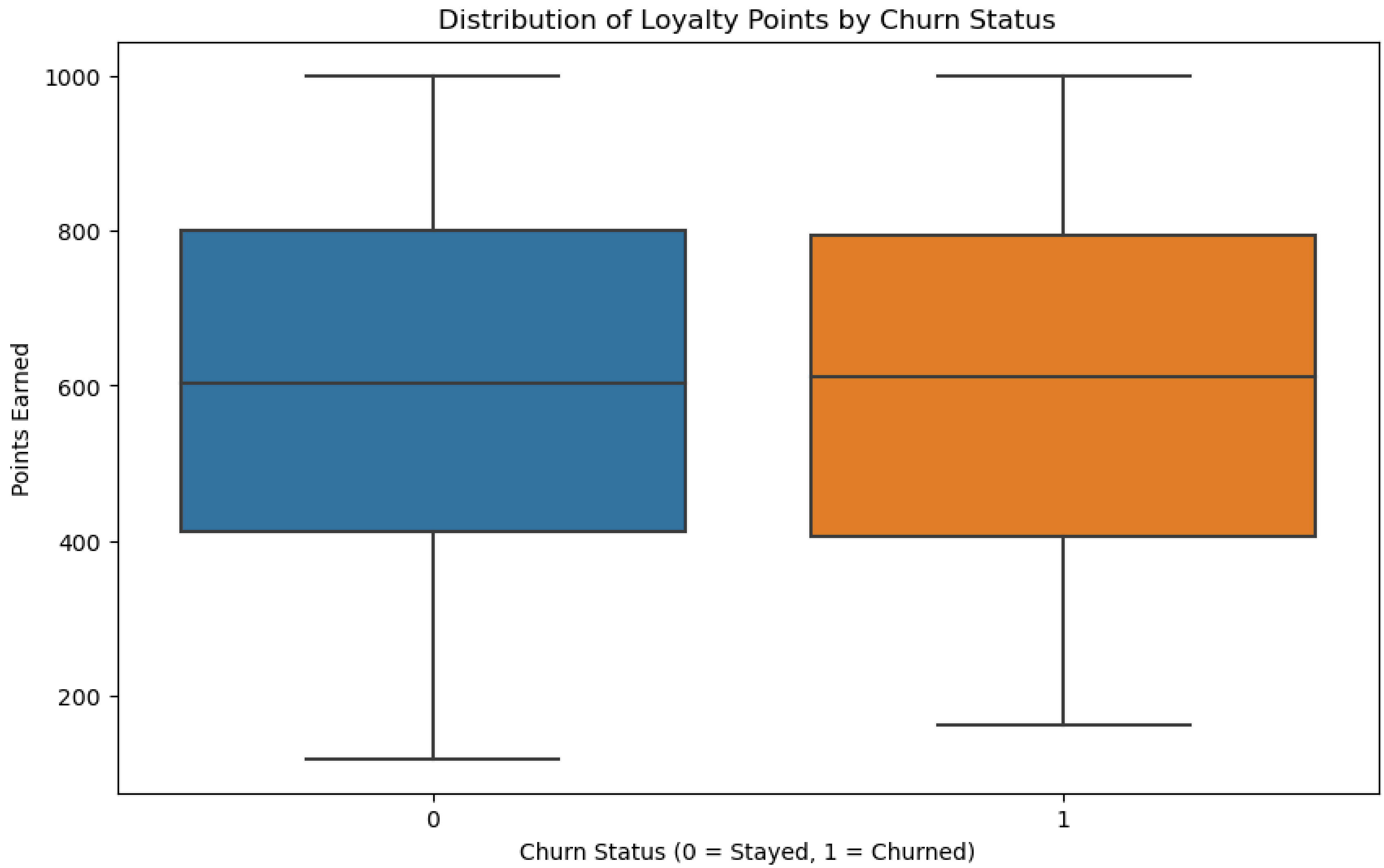
```
<Figure size 1000x600 with 0 Axes>
```



Customer Churn Analysis by Loyalty Points

In [334…

```
plt.figure(figsize=(10, 6))
sns.boxplot(x='Exited', y='Point Earned', data=df)
plt.title('Distribution of Loyalty Points by Churn Status')
plt.xlabel('Churn Status (0 = Stayed, 1 = Churned)')
plt.ylabel('Points Earned')
plt.show()
```



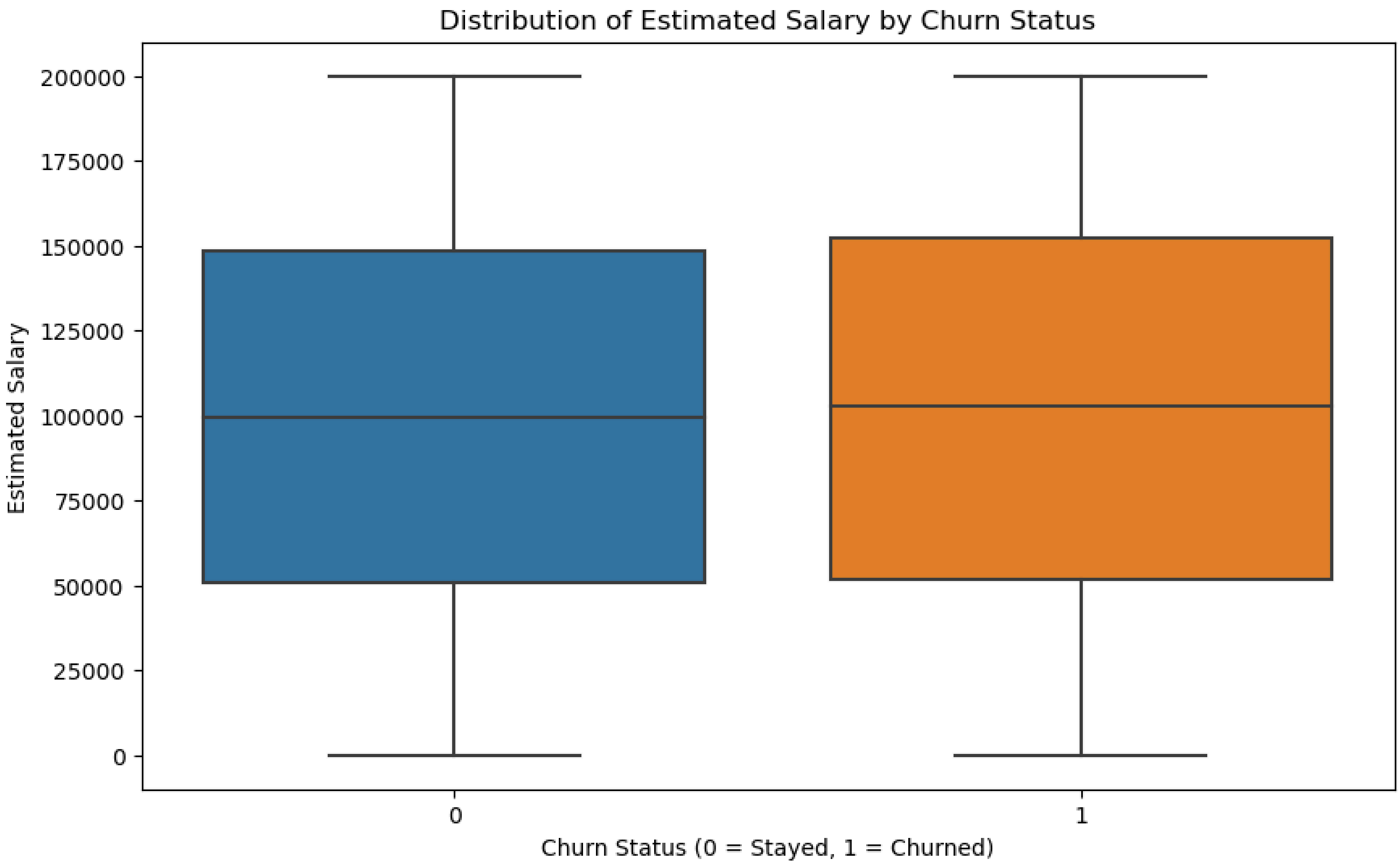
- Based on the box plot, we can say that the points customers earn might not influence whether they stay or leave, since both churned and retained customers have similar point ranges: 180 to 1000 for churned customers and 100 to 1000 for retained customers.

Customer Churn Analysis by Estimated Salary

In [335…

```
plt.figure(figsize=(10, 6))
sns.boxplot(x='Exited', y='EstimatedSalary', data=df)
```

```python
plt.title('Distribution of Estimated Salary by Churn Status')
plt.xlabel('Churn Status (0 = Stayed, 1 = Churned)')
plt.ylabel('Estimated Salary')
plt.show()
```



- Estimated salary might not influence whether they stay or leave, since both churned and retained customers have similar point ranges.

Hypothesis Testing

In [336...

```
for col in numerical_cols:
    churned = df[df['Exited'] == 1][col]
    not_churned = df[df['Exited'] == 0][col]
    t_stat, p_val = stats.ttest_ind(churned, not_churned)
    print(f'T-test for {col}: t-statistic = {t_stat}, p-value = {p_val}')
```

```
T-test for CreditScore: t-statistic = -2.6778368664704235, p-value = 0.0074220372427342435
T-test for Age: t-statistic = 29.76379695489027, p-value = 1.3467162476197306e-186
T-test for Balance: t-statistic = 11.940747722508185, p-value = 1.2092076077156017e-32
T-test for NumOfProducts: t-statistic = -4.765990052595114, p-value = 1.905776990458955e-06
T-test for EstimatedSalary: t-statistic = 1.2489445044833742, p-value = 0.2117146135149097
T-test for Point Earned: t-statistic = -0.4627759848070133, p-value = 0.6435350184288993
```

- Customers who left the bank have lower credit scores than those who stayed.
- Customers who left the bank are generally older than those who stayed.
- Customers who left the bank tend to have higher account balances.
- Customers who left the bank have fewer products with the bank.
- Salary doesn't make a significant difference in whether customers stay or leave.
- The number of points earned doesn't significantly affect whether customers stay or leave.

In [337...

```
categorical_cols = ['Geography', 'Gender', 'HasCrCard', 'IsActiveMember', 'Card Type']
for col in categorical_cols:
    contingency_table = pd.crosstab(df[col], df['Exited'])
    chi2, p, dof, ex = stats.chi2_contingency(contingency_table)
    print(f'Chi-square test for {col}: chi2 = {chi2}, p-value = {p}')
```

```
Chi-square test for Geography: chi2 = 300.6264011211942, p-value = 5.245736109572763e-66
Chi-square test for Gender: chi2 = 112.39655374778587, p-value = 2.9253677618642e-26
Chi-square test for HasCrCard: chi2 = 0.4494039375253385, p-value = 0.5026181509009862
Chi-square test for IsActiveMember: chi2 = 243.6948024819593, p-value = 6.1531674381134086e-55
Chi-square test for Card Type: chi2 = 5.053223027060927, p-value = 0.16794112067810177
```

- The region where customers live affects whether they are likely to stay or leave.
- There is a significant difference in the likelihood of staying or leaving between male and female customers.
- Having a credit card doesn't significantly affect whether customers stay or leave.
- Active members are much less likely to leave compared to inactive members.

- The type of cards doesn't significantly affect whether customers stay or leave.

Recommendation

- We noticed that more women are leaving than men. So, let's think about what might make them unhappy and try to fix those things. Maybe we could offer them special deals or make sure they feel heard when they have a problem.
- It seems like customers from Germany are leaving more than others. Offering them services that suit their needs better or train our staff to understand their culture.
- People who aren't using our services much are more likely to leave. Let's come up with ways to get them interested again. Maybe we could offer them a discount or show them how our services can help them.
- Older customers seem to be leaving more than younger ones. To make sure they feel valued, offering them with advice on managing their money or make our services easier for them to use.
- Customers who complain often end up leaving. We need to be quick to respond to their complaints and fix any problems they have. Making sure they feel heard and valued.
- Customers with lower credit scores tend to leave more. Let's offer them resources and advice on how to improve their credit. Maybe we could give them tips on managing their money better or offer them services to help them build their credit back up.
- Customers with a lot of money in their accounts are leaving more. Offering them special perks or benefits for keeping their money with us.
- Most of our customers are between 30 and 40 years old. Analysing what they might need at this stage in their lives and offer them services that match. Helping them with things like buying a house or starting a family.
- Customers who use our services a lot are less likely to leave. Offering them discounts or rewards for staying active with us.