capstone-project-churn-analysis

October 5, 2024

#Losing bank customers

• Every bank wants to hold their customers for sustaining their business and thus this Anonymous Multinational bank. You have customer data of account holders at Anonymous Multinational Bank with the aim of understanding • exploring the correlation between variables such as credit score, age, tenure, balance, and geography with customer churn. Assess the impact of demographic factors like gender and the presence of credit cards on churn rates. • Additionally, analyze customer satisfaction scores and complaint resolutions to identify areas for service improvement. Utilize your analytics skills to find factors contributing to potential churn based. This project provides an opportunity to enhance customer retention strategies by uncovering patterns and insights within the dataset.

Losing bank customers

Data description

RowNumber—corresponds to the record (row) number and has no effect on the output.

CustomerId—contains random values and has no effect on customer leaving the bank.

Surname—the surname of a customer has no impact on their decision to leave the bank.

CreditScore—can have an effect on customer churn, since a customer with a higher credit score is less likely to leave the bank.

Geography—a customer's location can affect their decision to leave the bank.

Gender—it's interesting to explore whether gender plays a role in a customer leaving the bank.

Age—this is certainly relevant, since older customers are less likely to leave their bank than younger ones.

Tenure—refers to the number of years that the customer has been a client of the bank. Normally, older clients are more loyal and less likely to leave a bank.

Balance—also a very good indicator of customer churn, as people with a higher balance in their accounts are less likely to leave the bank compared to those with lower balances.

NumOfProducts—refers to the number of products that a customer has purchased through the bank.

HasCrCard—denotes whether or not a customer has a credit card. This column is also relevant, since people with a credit card are less likely to leave the bank.

IsActiveMember—active customers are less likely to leave the bank.

EstimatedSalary—as with balance, people with lower salaries are more likely to leave the bank compared to those with higher salaries.

Exited—whether or not the customer left the bank.

Complain—customer has complaint or not.

Satisfaction Score—Score provided by the customer for their complaint resolution.

Card Type—type of card hold by the customer.

Points Earned—the points earned by the customer for using credit card.

[2]:
```
!gdown 1q1Mh3Mm4kv1LitxWcdY6--gNHVmuAfPP
```

```
Downloading…
From: https://drive.google.com/uc?id=1q1Mh3Mm4kv1LitxWcdY6--gNHVmuAfPP
To: /content/Bank-Records.csv
100% 837k/837k [00:00<00:00, 85.0MB/s]
```

[3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
```

[4]:
```
data = pd.read_csv('Bank-Records.csv')
data
```

[4]:
```
      RowNumber  CustomerId    Surname  CreditScore Geography  Gender  Age  \
0             1    15634602   Hargrave          619    France  Female   42
1             2    15647311       Hill          608     Spain  Female   41
2             3    15619304       Onio          502    France  Female   42
3             4    15701354       Boni          699    France  Female   39
4             5    15737888   Mitchell          850     Spain  Female   43
…           …         …         …          …       …      …   …
9995       9996    15606229   Obijiaku          771    France    Male   39
9996       9997    15569892  Johnstone          516    France    Male   35
9997       9998    15584532        Liu          709    France  Female   36
9998       9999    15682355  Sabbatini          772   Germany    Male   42
9999      10000    15628319     Walker          792    France  Female   28

      Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
0          2       0.00              1          1               1
1          1   83807.86              1          0               1
2          8  159660.80              3          1               0
3          1       0.00              2          0               0
4          2  125510.82              1          1               1
…        …        …            …         …              …
9995       5       0.00              2          1               0
```

```
9996      10   57369.61              1          1               1
9997       7       0.00              1          0               1
9998       3   75075.31              2          1               0
9999       4  130142.79              1          1               0

      EstimatedSalary  Exited  Complain  Satisfaction Score Card Type  \
0           101348.88       1         1                   2   DIAMOND
1           112542.58       0         1                   3   DIAMOND
2           113931.57       1         1                   3   DIAMOND
3            93826.63       0         0                   5      GOLD
4            79084.10       0         0                   5      GOLD
…                 …     …       …                   …       …
9995         96270.64       0         0                   1   DIAMOND
9996        101699.77       0         0                   5  PLATINUM
9997         42085.58       1         1                   3    SILVER
9998         92888.52       1         1                   2      GOLD
9999         38190.78       0         0                   3   DIAMOND

      Point Earned
0              464
1              456
2              377
3              350
4              425
…             …
9995           300
9996           771
9997           564
9998           339
9999           911

[10000 rows x 18 columns]
```

[5]:
```
data.shape
```

[5]:
```
(10000, 18)
```

[6]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 18 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   RowNumber           10000 non-null  int64
 1   CustomerId          10000 non-null  int64
 2   Surname             10000 non-null  object
```

```
 3   CreditScore         10000 non-null  int64
 4   Geography           10000 non-null  object
 5   Gender              10000 non-null  object
 6   Age                 10000 non-null  int64
 7   Tenure              10000 non-null  int64
 8   Balance             10000 non-null  float64
 9   NumOfProducts       10000 non-null  int64
 10  HasCrCard           10000 non-null  int64
 11  IsActiveMember      10000 non-null  int64
 12  EstimatedSalary     10000 non-null  float64
 13  Exited              10000 non-null  int64
 14  Complain            10000 non-null  int64
 15  Satisfaction Score  10000 non-null  int64
 16  Card Type           10000 non-null  object
 17  Point Earned        10000 non-null  int64
dtypes: float64(2), int64(12), object(4)
memory usage: 1.4+ MB
```

[7]:
```
data['CustomerId'].nunique()
```

[7]: 10000

1 Performing Basic Exploring data analysis

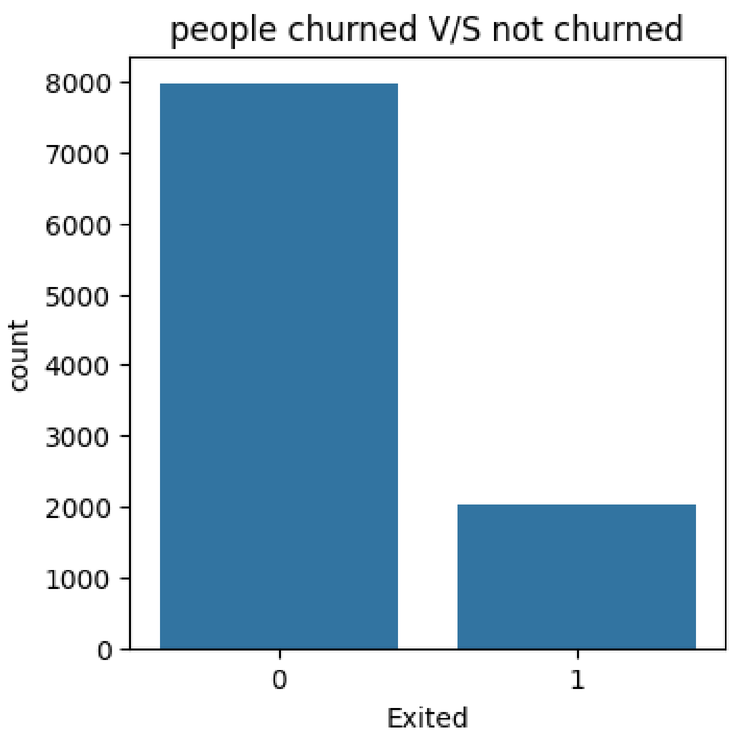
[8]:
```
data[['CustomerId','Exited']]
```

[8]:

	CustomerId	Exited
0	15634602	1
1	15647311	0
2	15619304	1
3	15701354	0
4	15737888	0
…	…	…
9995	15606229	0
9996	15569892	0
9997	15584532	1
9998	15682355	1
9999	15628319	0

[10000 rows x 2 columns]

[9]:
```
plt.figure(figsize=(4,4))
sns.countplot(x = data['Exited'])
plt.title("people churned V/S not churned")
plt.show()
```



[10]:
```
data['Exited'].value_counts()
```

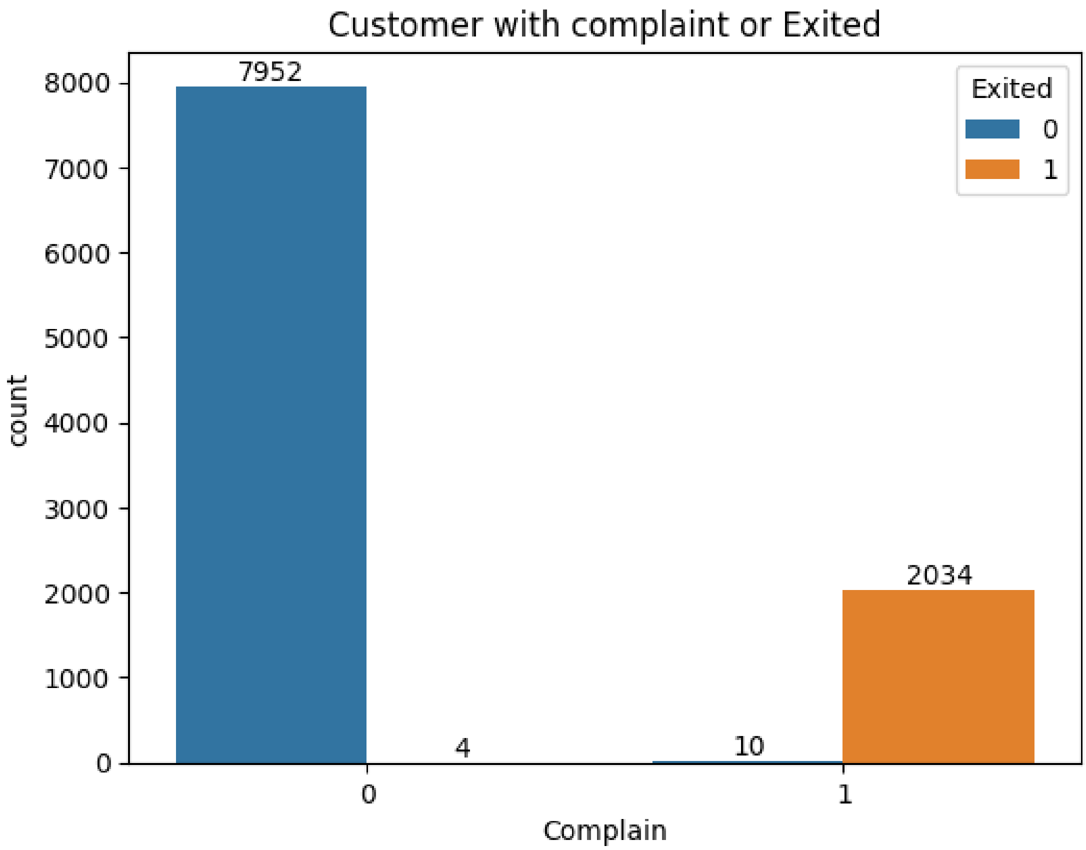
[10]:
```
Exited
0    7962
1    2038
Name: count, dtype: int64
```

- from above observation it is clear that 2038 people exited from bank and 7962 are still account holder at the bank out of 10000

[11]:
```
pd.crosstab(columns = data['Complain'],index = data['Exited'])
```

[11]:
```
Complain     0     1
Exited
0         7952    10
1            4  2034
```

[104]:
```
warnings.simplefilter(action='ignore', category=FutureWarning)

ax1 = sns.countplot(x=data['Complain'],hue=data['Exited'])
for container in ax1.containers:
    ax1.bar_label(container)
plt.title('Customer with complaint or Exited')
plt.show()
```

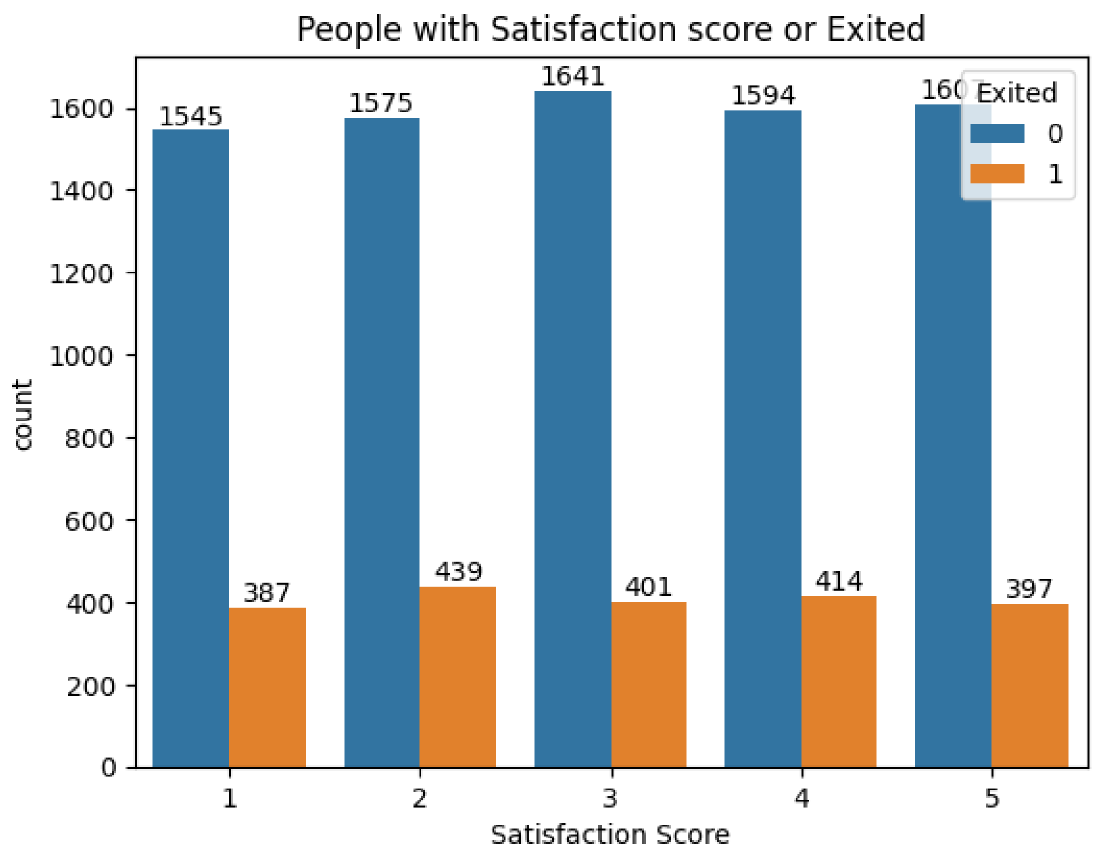


out of 2038 customer churned there were 2034 customer who complained

[13]:
```
pd.crosstab(columns = data['Satisfaction Score'],index = data['Exited'])
```

[13]:
```
Satisfaction Score     1     2     3     4     5
Exited
0                   1545  1575  1641  1594  1607
1                    387   439   401   414   397
```

[103]:
```
warnings.simplefilter(action='ignore', category=FutureWarning)

ax2 = sns.countplot(x=data['Satisfaction Score'],hue=data['Exited'])
for container in ax2.containers:
    ax2.bar_label(container)
plt.title('People with Satisfaction score or Exited')

plt.show()
```

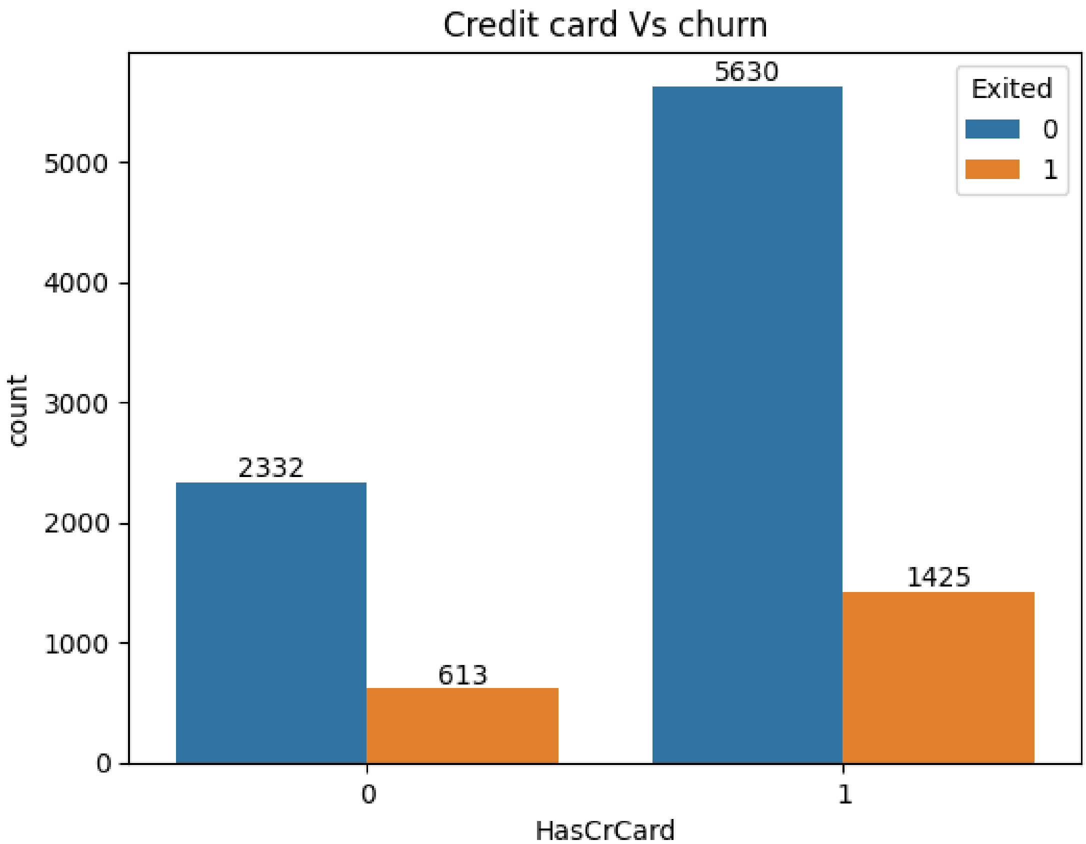


[15]:
```
pd.crosstab(columns = data['HasCrCard'],index = data['Exited'])
```

[15]:
```
HasCrCard     0     1
Exited
0          2332  5630
1           613  1425
```

from above observation it is cleared that people who have no card and exited were 613 and people with card and exited were 1425 which shows people having card exited more than who have no cards

[102]:
```
warnings.simplefilter(action='ignore', category=FutureWarning)

ax3 = sns.countplot(x = data['HasCrCard'],hue=data['Exited'])
for container in ax3.containers:
    ax3.bar_label(container)
plt.title("Credit card Vs churn")
plt.show()
```



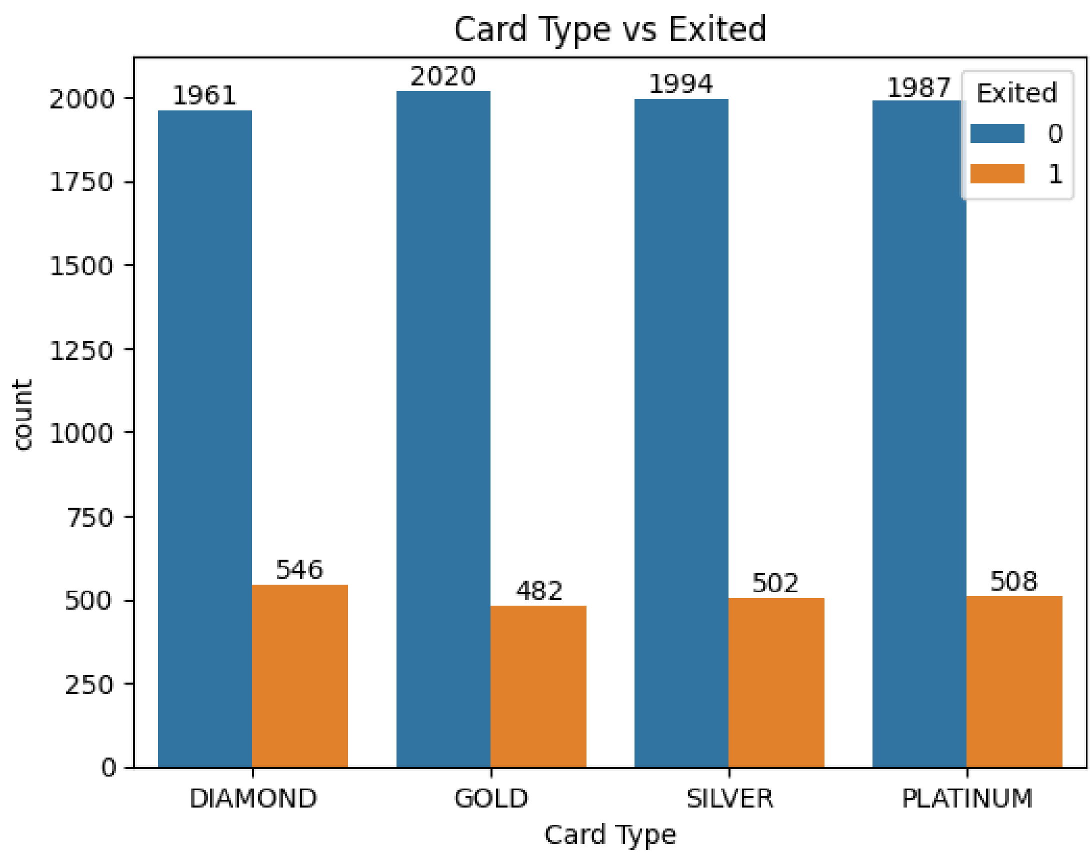
[17]:
```
pd.crosstab(columns = data['Card Type'],index = data['Exited'])
```

[17]:

Card Type	DIAMOND	GOLD	PLATINUM	SILVER
Exited				
0	1961	2020	1987	1994
1	546	482	508	502

from above observation we can see almost all different type of Card Type holders have Equally churned out

[101]:
```
warnings.simplefilter(action='ignore', category=FutureWarning)

ax4 = sns.countplot(x=data['Card Type'],hue=data['Exited'])
for container in ax4.containers:
    ax4.bar_label(container)
plt.title('Card Type vs Exited')
plt.show()
```



[19]:
```
data[data['Exited']== 1]['CreditScore'].max()
```

[19]: 850

[20]:
```
bins =[300,400,500,600,700,800,900]
```

[21]:
```
credit_bin = pd.cut(data[data['Exited']== 1]['CreditScore'],bins)
```

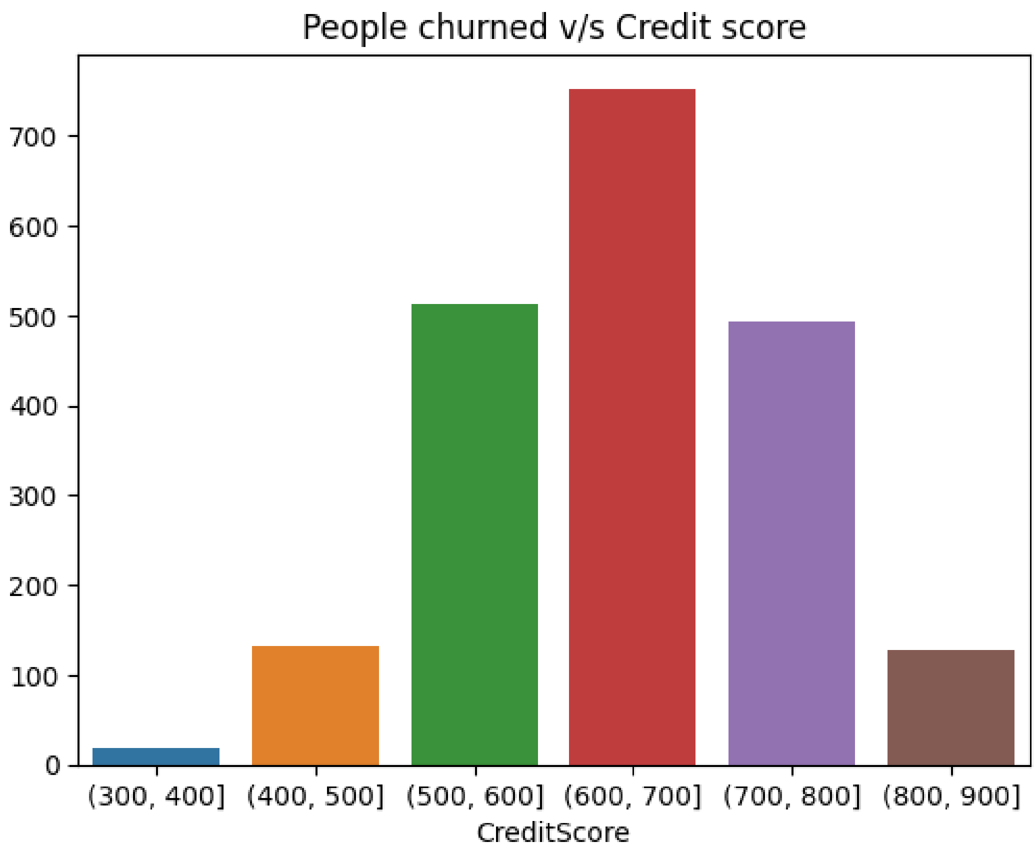
[22]:
```
pd.crosstab(columns = credit_bin ,index = data['Exited'])
```

[22]:
```
CreditScore  (300, 400]  (400, 500]  (500, 600]  (600, 700]  (700, 800]  \
Exited
1                    19         133         513         753         493

CreditScore  (800, 900]
Exited
1                   127
```

people with credit score in between 500 - 600 and 600-700 left the banking service the most

[100]:
```
warnings.simplefilter(action='ignore', category=FutureWarning)

sns.barplot(pd.crosstab(columns = credit_bin ,index = data['Exited']))
plt.title('People churned v/s Credit score')
```

[100]:
```
Text(0.5, 1.0, 'People churned v/s Credit score')
```



[24]:
```
pd.crosstab(columns = data['Gender'],index = data['Exited'])
```

[24]:

Gender	Female	Male
Exited		
0	3404	4558
1	1139	899

[25]:
```
pd.crosstab(columns = data['Geography'],index = data['Exited'])
```

[25]:

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

[26]:
```
pd.crosstab(columns = data['Geography'],index = data['Gender'])
```

[26]:

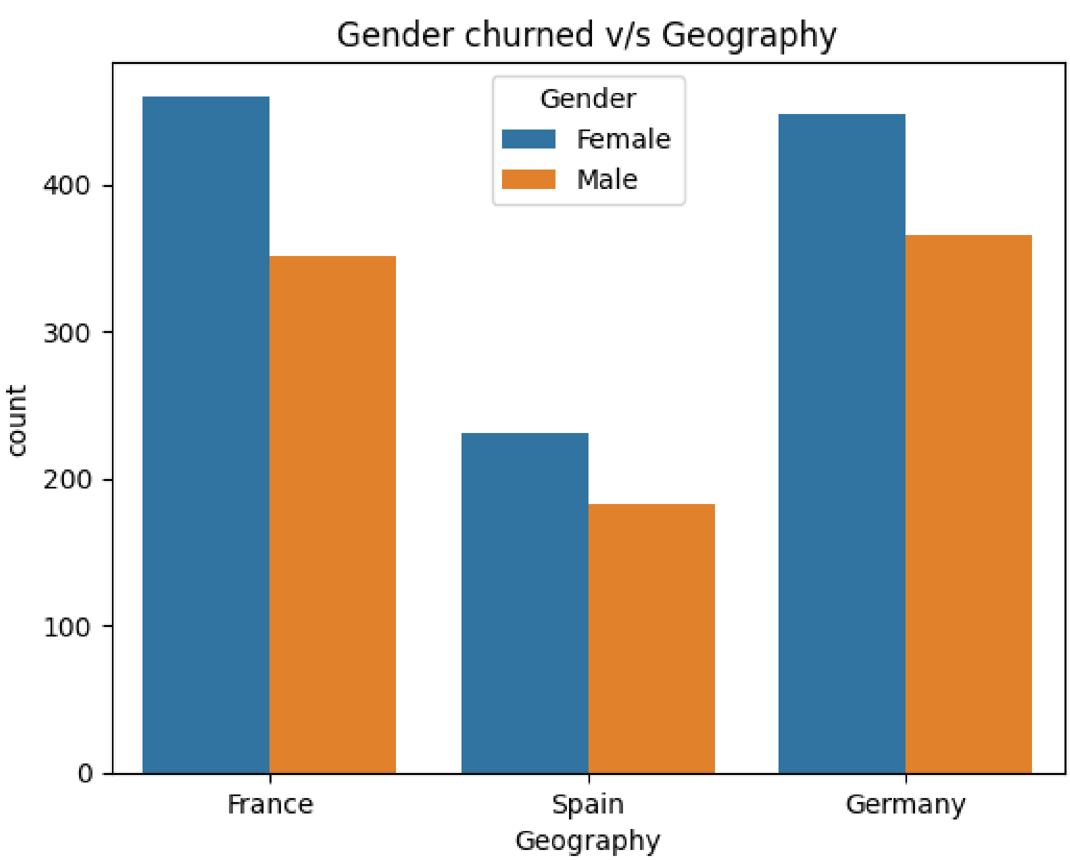
Geography	France	Germany	Spain
Gender			
Female	2261	1193	1089
Male	2753	1316	1388

[27]:
```
pd.crosstab(columns = [data['Geography'],data['Gender']],index = data['Exited'])
```

[27]:

Geography	France		Germany		Spain	
Gender	Female	Male	Female	Male	Female	Male
Exited						
0	1801	2402	745	950	858	1206
1	460	351	448	366	231	182

[99]:
```
warnings.simplefilter(action='ignore', category=FutureWarning)

sns.countplot(x=
 ↪data[data['Exited']==1]['Geography'],hue=data[data['Exited']==1]['Gender'])
plt.title("Gender churned v/s Geography")
plt.show()
```

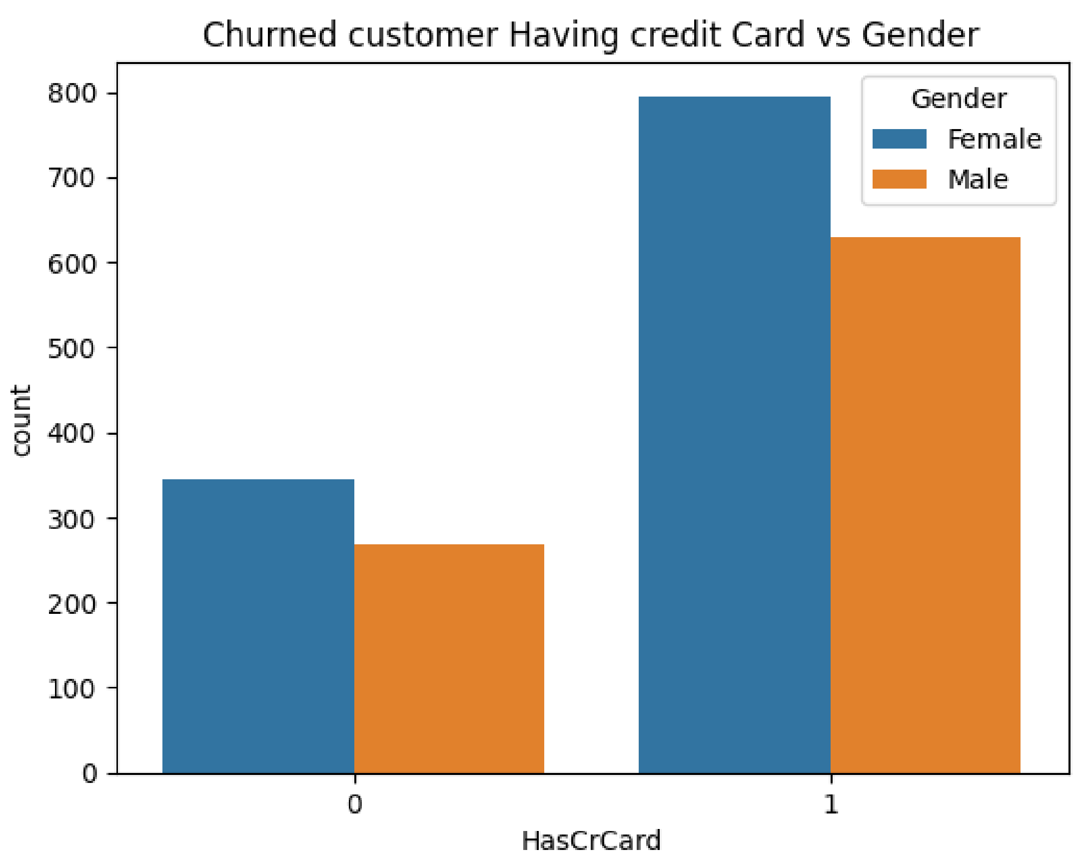


[30]:
```
pd.crosstab(columns = [data['HasCrCard'],data['Gender']],index = data['Exited'])
```

[30]:
```
HasCrCard      0             1
Gender    Female  Male Female  Male
Exited
0           1007  1325   2397  3233
1            344   269    795   630
```

[98]:
```
warnings.simplefilter(action='ignore', category=FutureWarning)

sns.countplot(x = data[data['Exited'] == 1]['HasCrCard'] ,hue =␣
 ↪data[data['Exited'] == 1]['Gender'])
plt.title('Churned customer Having credit Card vs Gender')
```

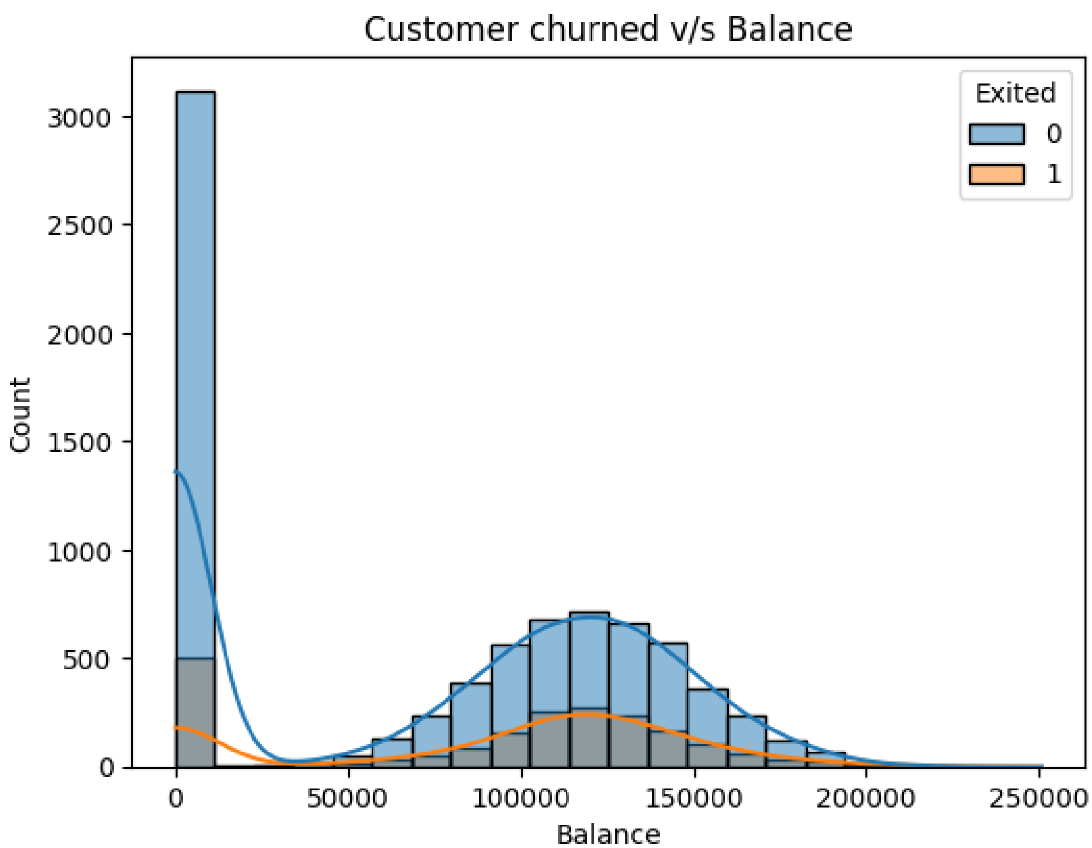
[98]:
```
Text(0.5, 1.0, 'Churned customer Having credit Card vs Gender')
```



[97]:
```
warnings.simplefilter(action='ignore', category=FutureWarning)
```

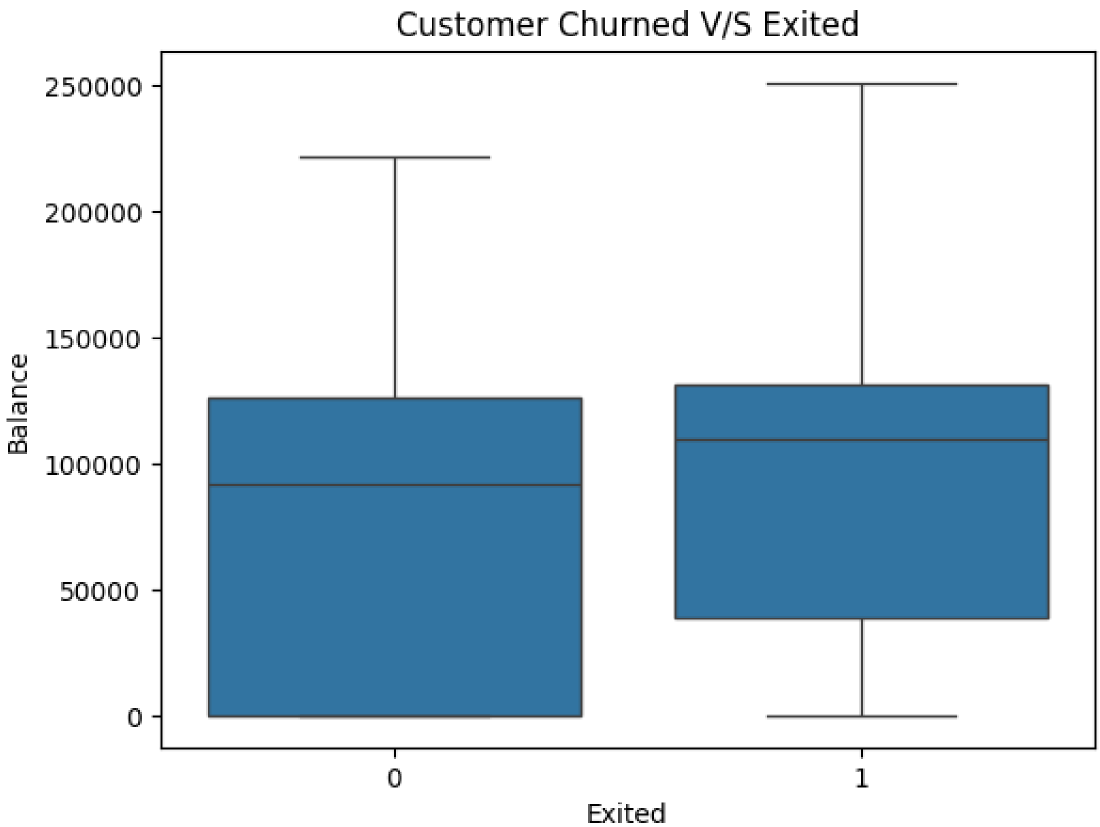
```python
sns.histplot(data = data, x= data['Balance'],hue =data['Exited'],kde =True)
plt.title('Customer churned v/s Balance')
```

[97]: Text(0.5, 1.0, 'Customer churned v/s Balance')



[96]:
```python
warnings.simplefilter(action='ignore', category=FutureWarning)

sns.boxplot(data=data,x=data['Exited'],y = data['Balance'])
plt.title("Customer Churned V/S Exited")
```

[96]: Text(0.5, 1.0, 'Customer Churned V/S Exited')



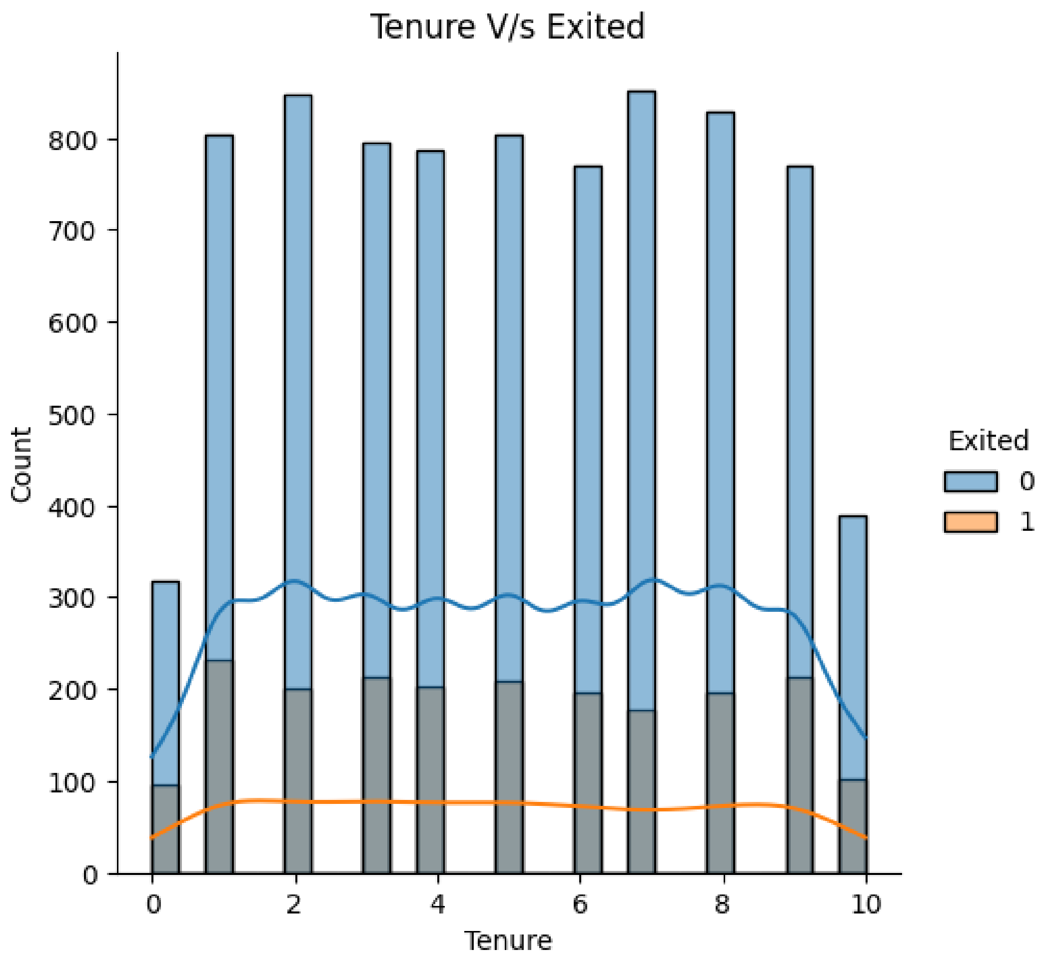
[34]:
```
pd.crosstab(columns = data['Tenure'],index = data['Exited'])
```

[34]:

Tenure	0	1	2	3	4	5	6	7	8	9	10
Exited											
0	318	803	847	796	786	803	771	851	828	770	389
1	95	232	201	213	203	209	196	177	197	214	101

[95]:
```
warnings.simplefilter(action='ignore', category=FutureWarning)

sns.displot(x = data['Tenure'],hue = data['Exited'],kde =True)
plt.title('Tenure V/s Exited')
```

[95]:
```
Text(0.5, 1.0, 'Tenure V/s Exited')
```



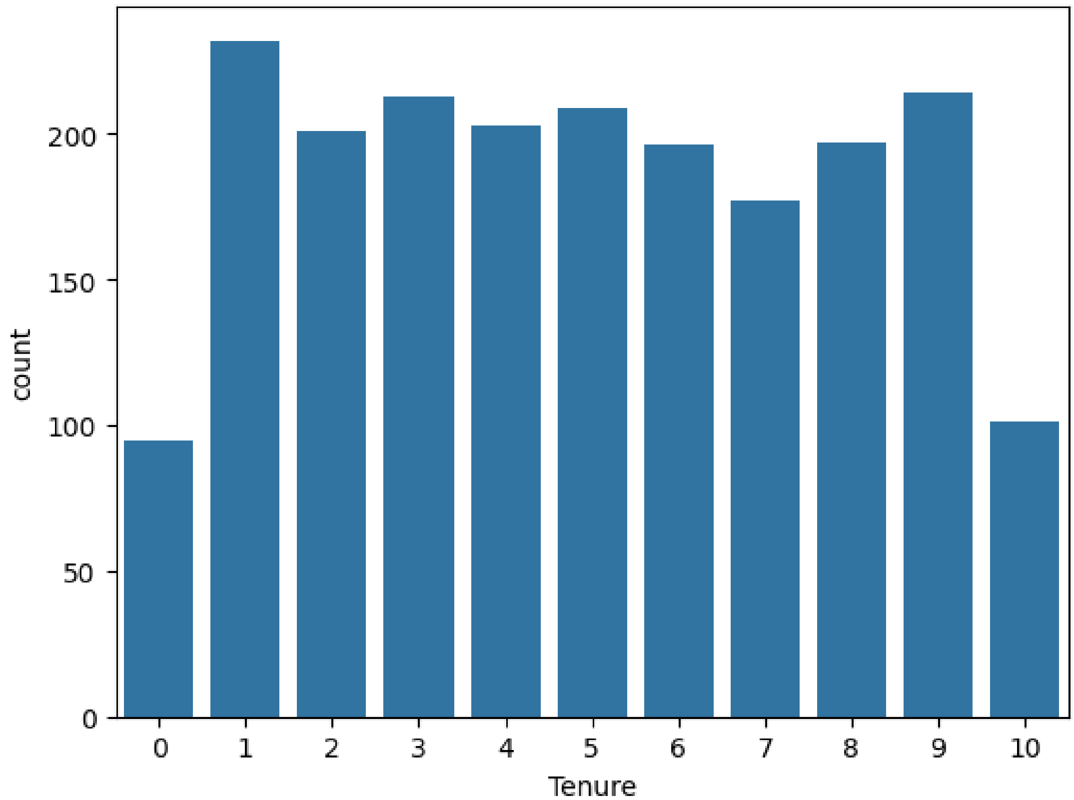
[36]:
```
data[data['Exited']==1]['Tenure'].value_counts().reset_index()
```

[36]:

	Tenure	count
0	1	232
1	9	214
2	3	213
3	5	209
4	4	203
5	2	201
6	8	197
7	6	196
8	7	177
9	10	101
10	0	95

[37]:
```
sns.countplot(x =data[data['Exited']==1]['Tenure'])
```

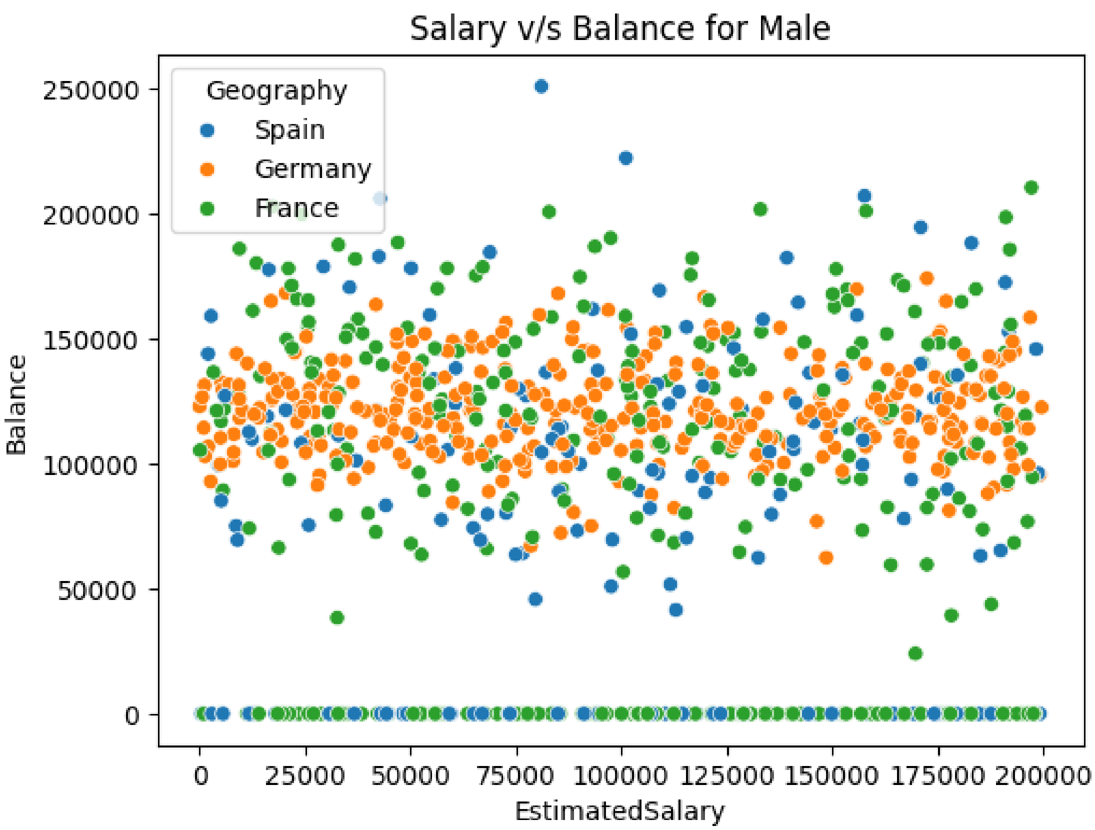
[37]: <Axes: xlabel='Tenure', ylabel='count'>



2 Lets check Estimated salary v/s balance of people w.r.t to Geography for different genders who left the bank.

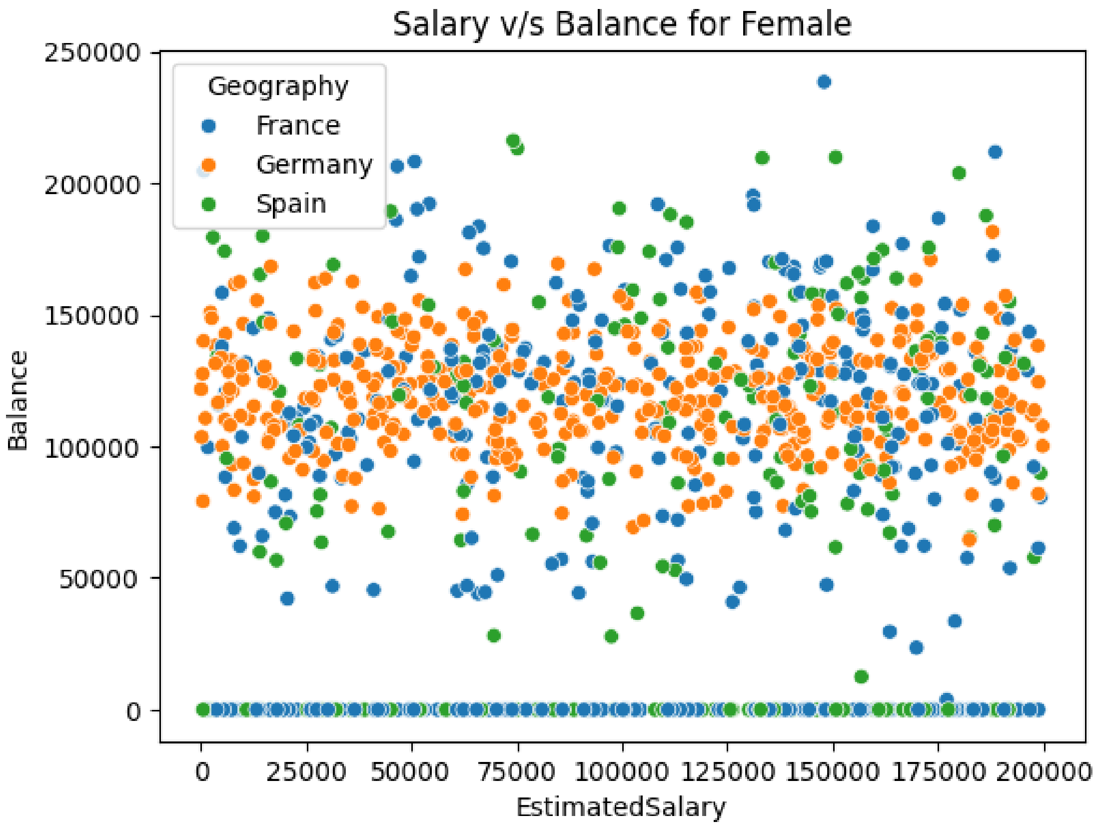
3 Male

[38]:
```
ax = sns.scatterplot(x="EstimatedSalary", y="Balance",
                     hue="Geography",
                     data=data[(data['Exited']==1) & (data['Gender'] =='Male')])
ax.set_title('Salary v/s Balance for Male')
plt.show()
```



4 Female

[39]:
```
ax = sns.scatterplot(x="EstimatedSalary", y="Balance",
                     hue="Geography",
                     data=data[(data['Exited']==1) & (data['Gender']␣
 ↪=='Female')])
ax.set_title('Salary v/s Balance for Female')
plt.show()
```



5 lets create functions for our Hypothesis test inorder to check correlations

5.1 Credit score vs Customer churn.

Credit score vs Customer churn we will use ANOVA for our hypothesis testing

[40]:
```
d1 = data [['CreditScore','Exited']]
d1
```

[40]:

	CreditScore	Exited
0	619	1
1	608	0
2	502	1
3	699	0
4	850	0
…	…	…
9995	771	0
9996	516	0
9997	709	1

```
9998           772       1
9999           792       0

[10000 rows x 2 columns]
```

[42]:
```
from scipy.stats import f_oneway,kruskal,ttest_ind,chi2_contingency
```

Ho: Customer churn is independent of Credit score

Ha: customer churn is dependent on Credit score

[43]:
```
t_stats, p_value = ttest_ind(data[data['Exited'] ==␣
 ↪0]['CreditScore'],data[data['Exited'] == 1]['CreditScore'])
print("t_stats :",t_stats)
print("p_value",p_value)
if p_value < 0.05:
  print("Null hypothesis is rejected")
else:
  print("Null hypothesis is accepted")
```

```
t_stats : 2.6778368664704235
p_value 0.0074220372427342435
Null hypothesis is rejected
```

6 Age vs Customer churn

we will use ttest_ind

[44]:
```
data[['Age','Exited']]
```

[44]:
```
      Age  Exited
0      42       1
1      41       0
2      42       1
3      39       0
4      43       0
…    …      …
9995   39       0
9996   35       0
9997   36       1
9998   42       1
9999   28       0

[10000 rows x 2 columns]
```

H0: Customer churn is independent of Age

Ha: Customer churn is dependent of Age

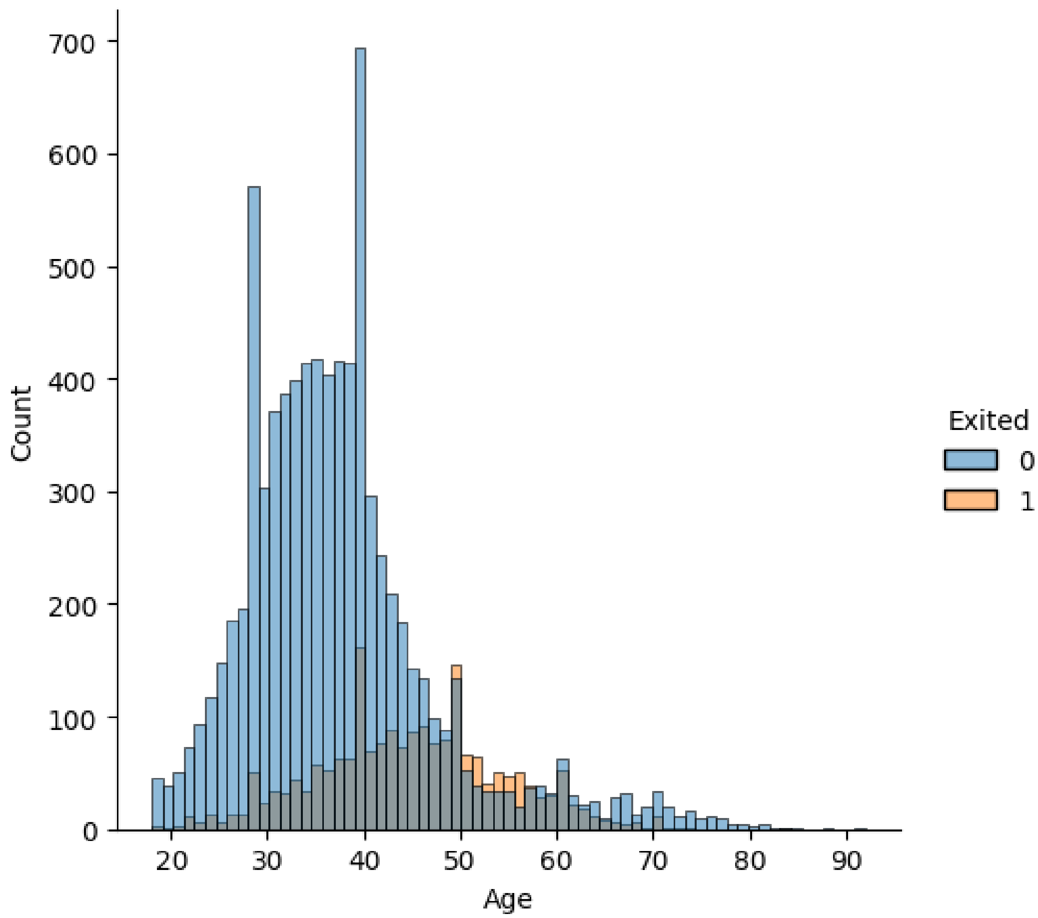
[45]:
```
t_stats, p_value = ttest_ind(data[data['Exited'] ==␣
 ↪0]['Age'],data[data['Exited'] == 1]['Age'])
print("t_stats :",t_stats)
print("p_value",p_value)
if p_value < 0.05:
  print("Null hypothesis is rejected")
else:
  print("Null hypothesis is accepted")
```

```
t_stats : -29.76379695489027
p_value 1.3467162476197306e-186
Null hypothesis is rejected
```

[94]:
```
warnings.simplefilter(action='ignore', category=FutureWarning)

plt.figure(figsize=(5, 5))
sns.displot(data=data, x="Age", hue="Exited")
```

[94]: <seaborn.axisgrid.FacetGrid at 0x7869ecc597e0>

```
<Figure size 500x500 with 0 Axes>
```



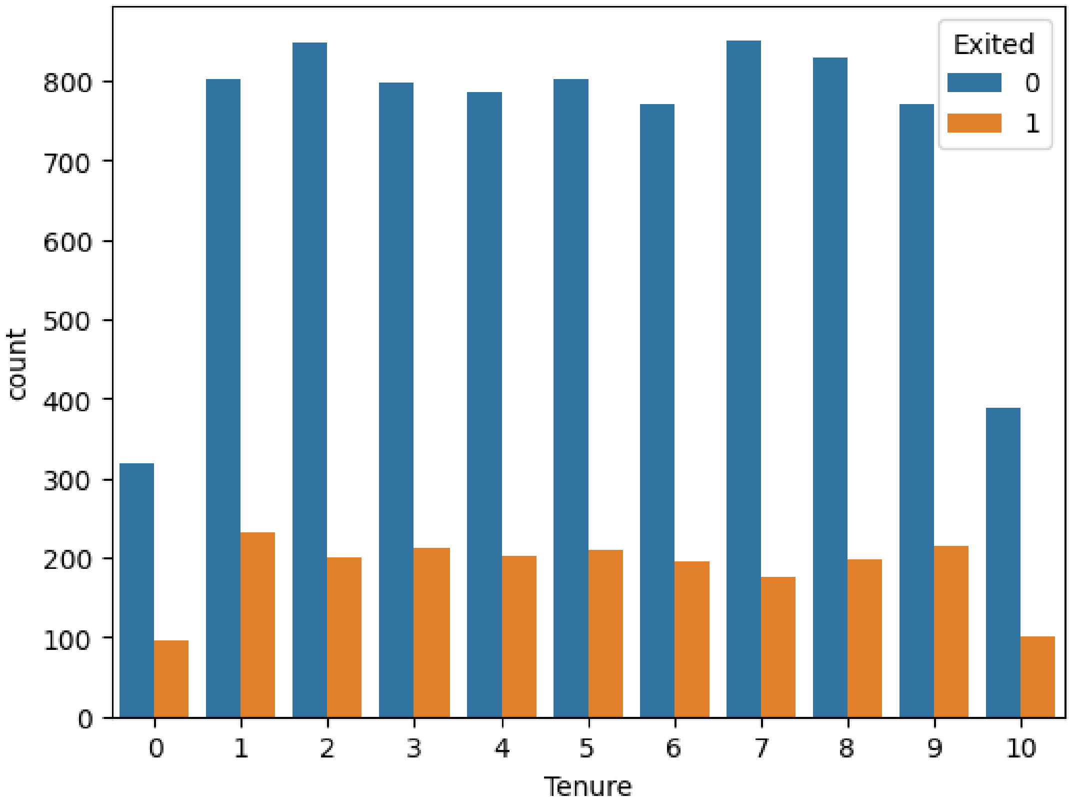
7 Tenure V/s Customer churn

[47]:
```
data[['Tenure','Exited']]
```

[47]:
```
      Tenure  Exited
0          2       1
1          1       0
2          8       1
3          1       0
4          2       0
…        …      …
9995       5       0
9996      10       0
9997       7       1
9998       3       1
9999       4       0

[10000 rows x 2 columns]
```

[93]:
```
warnings.simplefilter(action='ignore', category=FutureWarning)

sns.countplot(x = data['Tenure'],hue = data['Exited'])
```

[93]:
```
<Axes: xlabel='Tenure', ylabel='count'>
```



H0: Customer churn is independent of tenure

Ha: Customer churn is dependent of tenure

[49]:
```
t_stats, p_value = ttest_ind(data[data['Exited'] ==
 0]['Tenure'],data[data['Exited'] == 1]['Tenure'])
print("t_stats :",t_stats)
print("p_value",p_value)
if p_value < 0.05:
  print("Null hypothesis is rejected")
else:
  print("Null hypothesis is accepted")
```

```
t_stats : 1.365570678788837
p_value 0.1721044754880606
Null hypothesis is accepted
```

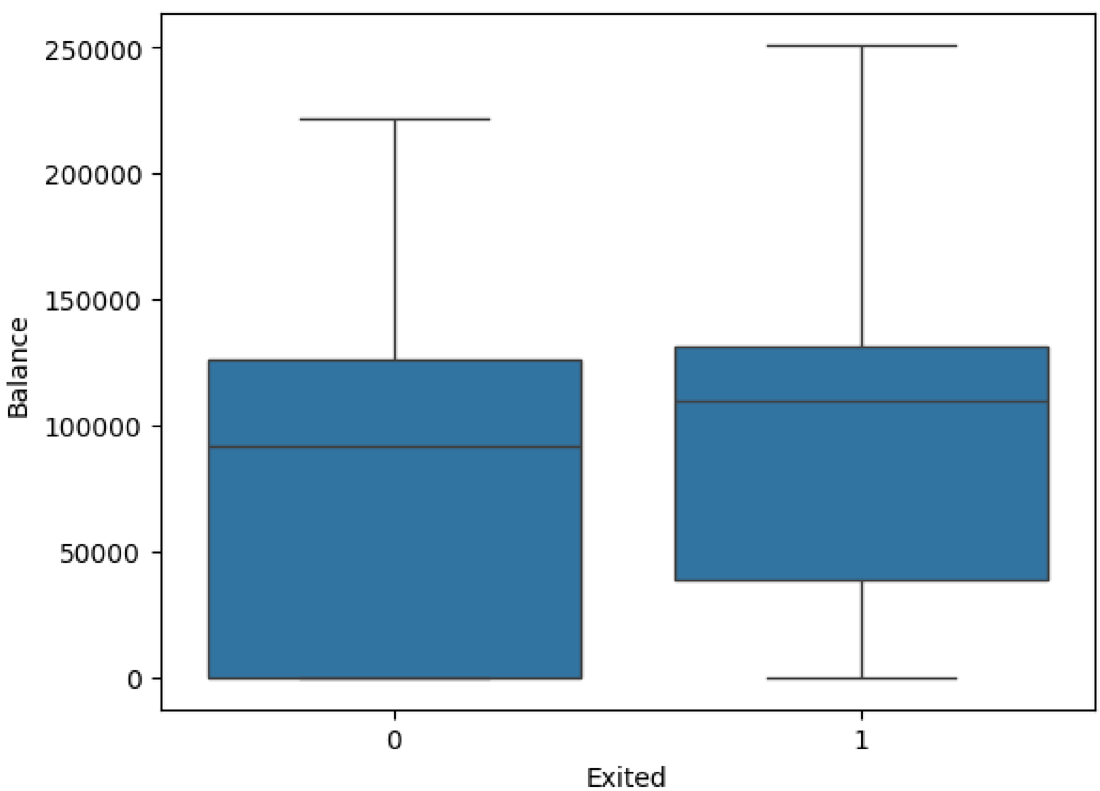
8 Balance vs Customer Churn

[50]:
```
print(" max Balance of person who churned ", data[data['Exited'] ==␣
↪1]['Balance'].max())
print(" min Balance of person who churned ",data[data['Exited'] ==␣
↪1]['Balance'].min())
print(" max Balance of person who didn't churned ", data[data['Exited'] ==␣
↪0]['Balance'].max())
print(" min Balance of person who  didn't churned ",data[data['Exited'] ==␣
↪0]['Balance'].min())
```

```
max Balance of person who churned  250898.09
min Balance of person who churned  0.0
max Balance of person who didn't churned  221532.8
min Balance of person who  didn't churned  0.0
```

[92]:
```
warnings.simplefilter(action='ignore', category=FutureWarning)

sns.boxplot(y = data['Balance'], x= data['Exited'])
```

[92]: <Axes: xlabel='Exited', ylabel='Balance'>



from graphical observation it is Difficult to conclude about correlation of customer churn and their balance in account

Ho: Customer Churn is independent of Balance

Ha: Customer Churn is dependent of Balance

[52]:
```
t_stats, p_value = ttest_ind(data[data['Exited'] ==␣
 ↪0]['Balance'],data[data['Exited'] == 1]['Balance'])
print("t_stats :",t_stats)
print("p_value",p_value)
if p_value < 0.05:
  print("Null hypothesis is rejected")
else:
  print("Null hypothesis is accepted")
```

```
t_stats : -11.940747722508185
p_value 1.2092076077156017e-32
Null hypothesis is rejected
```

9 Geogrpahy v/s customer churn

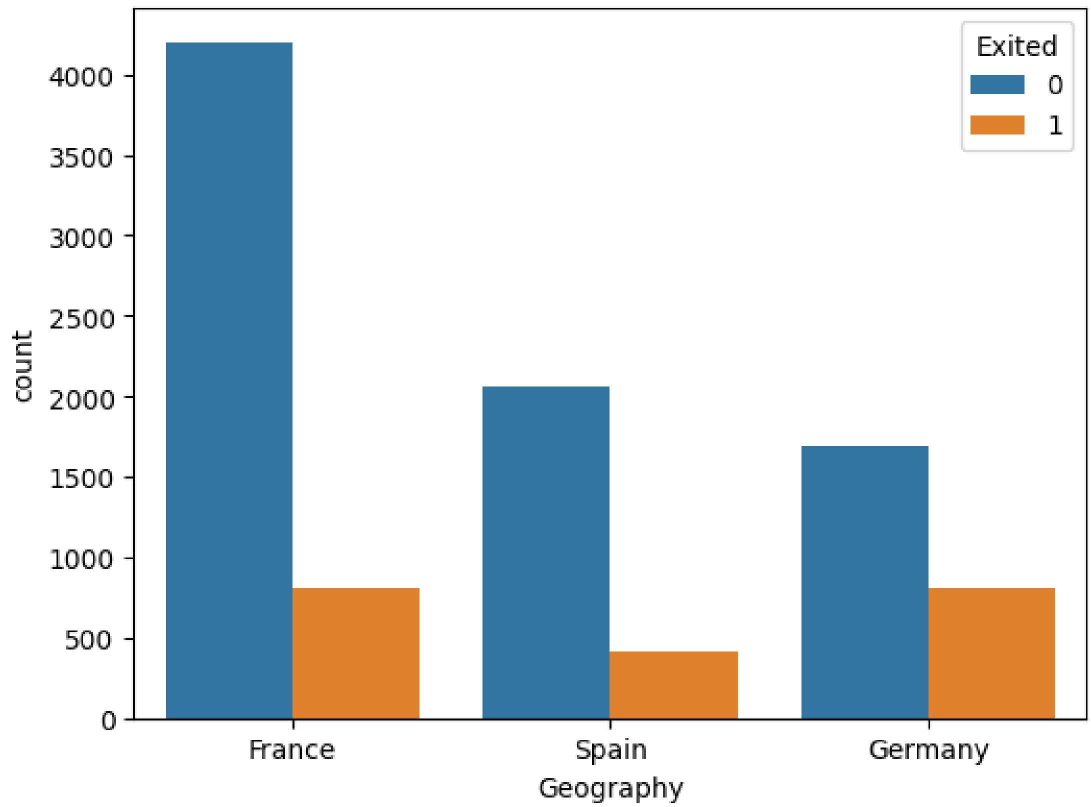
[53]:
```
GC = pd.crosstab(columns = data['Geography'],index = data['Exited'])
GC
```

[53]:

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

[91]:
```
warnings.simplefilter(action='ignore', category=FutureWarning)

sns.countplot(x=data['Geography'],hue=data['Exited'])
```

[91]: <Axes: xlabel='Geography', ylabel='count'>



Since this is a case of categorical - categorical we would apply chi2_contingency or Chi_square test of independence

H0: Geography and Customer churn are independent

Ha : Geography and Customer churn are dependent

[55]:
```
t_stats, p_value, dof, array = chi2_contingency (GC)
print("Result:",chi2_contingency (GC))
print("t_stats :",t_stats)
print("p_value",p_value)
if p_value < 0.05:
  print("Null hypothesis is rejected")
  print("Geography and Customer churn are dependent")

else:
  print("Null hypothesis is accepted")
  print("Geography and Customer churn are Independent")
```

```
Result: Chi2ContingencyResult(statistic=300.6264011211942,
pvalue=5.245736109572763e-66, dof=2, expected_freq=array([[3992.1468, 1997.6658,
1972.1874],
       [1021.8532,  511.3342,  504.8126]]))
```

```
t_stats : 300.6264011211942
p_value 5.245736109572763e-66
Null hypothesis is rejected
Geography and Customer churn are dependent
```

10 Impact assessement of different features on Customer churn

11 Gender and Customer Churn

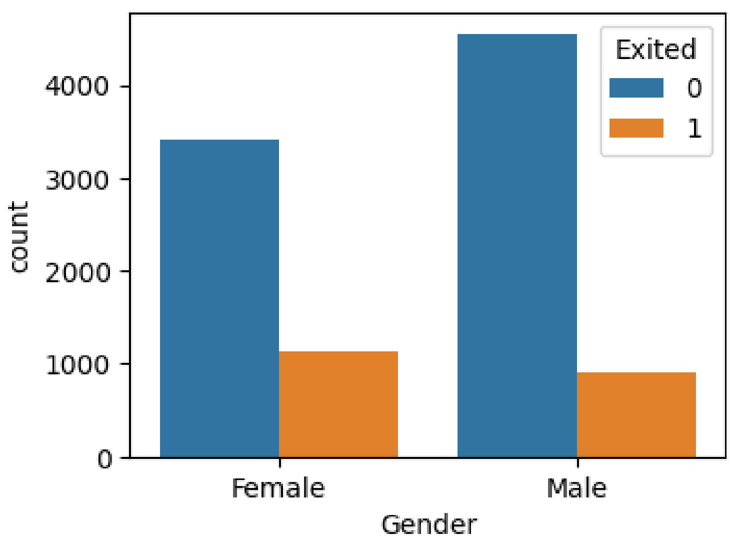
[56]:
```
Gec = pd.crosstab(columns = data['Gender'],index = data['Exited'])
Gec
```

[56]:

Gender	Female	Male
Exited		
0	3404	4558
1	1139	899

[90]:
```
warnings.simplefilter(action='ignore', category=FutureWarning)

plt.figure(figsize=(4,3))
sns.countplot(x=data['Gender'],hue=data['Exited'])
```

[90]:
```
<Axes: xlabel='Gender', ylabel='count'>
```



H0: Gender and Customer churn are independent

Ha : Gender and Customer churn are dependent

[58]:
```
t_stats, p_value, dof, array = chi2_contingency (Gec)
print("Result:",chi2_contingency (Gec))
print("t_stats :",t_stats)
print("p_value",p_value)
if p_value < 0.05:
  print("Null hypothesis is rejected")
  print("Gender and Customer churn are dependent")

else:
  print("Null hypothesis is accepted")
  print("Gender and Customer churn are Independent")
```

```
Result: Chi2ContingencyResult(statistic=112.39655374778587,
pvalue=2.9253677618642e-26, dof=1, expected_freq=array([[3617.1366, 4344.8634],
       [ 925.8634, 1112.1366]]))
t_stats : 112.39655374778587
p_value 2.9253677618642e-26
Null hypothesis is rejected
Gender and Customer churn are dependent
```

12 Impact of Credit Card on Churn rate

[59]:
```
Cc = pd.crosstab(columns = data['Card Type'],index = data['Exited'])
Cc
```

[59]:

Card Type	DIAMOND	GOLD	PLATINUM	SILVER
Exited				
0	1961	2020	1987	1994
1	546	482	508	502

H0: Credit Card and Customer churn are independent

Ha : Credit Card and Customer churn are dependent

[60]:
```
t_stats, p_value, dof, array = chi2_contingency (Gec)
print("Result:",chi2_contingency (Gec))
print("t_stats :",t_stats)
print("p_value",p_value)
if p_value < 0.05:
  print("Null hypothesis is rejected")
  print("Credit Card and Customer churn are dependent")

else:
  print("Null hypothesis is accepted")
  print("Credit Card and Customer churn are Independent")
```

```
Result: Chi2ContingencyResult(statistic=112.39655374778587,
pvalue=2.9253677618642e-26, dof=1, expected_freq=array([[3617.1366, 4344.8634],
```

```
[ 925.8634, 1112.1366]]))
t_stats : 112.39655374778587
p_value 2.9253677618642e-26
Null hypothesis is rejected
Credit Card and Customer churn are dependent
```

13 Analayze Area for service improvement

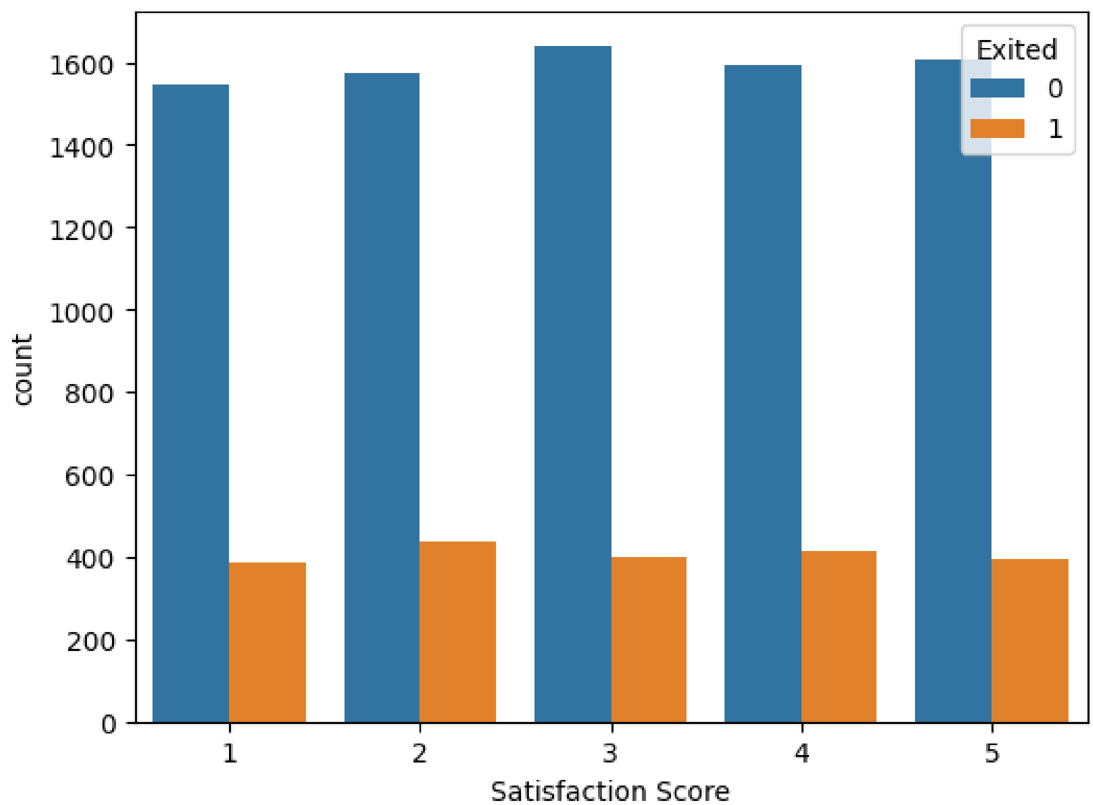
[61]:
```python
pd.crosstab(columns = [data['Complain'],data['Satisfaction Score']],index =
    data['Exited'])
```

[61]:

Complain	0					1				
Satisfaction Score	1	2	3	4	5	1	2	3	4	5
Exited										
0	1544	1574	1636	1594	1604	1	1	5	0	3
1	1	2	0	1	0	386	437	401	413	397

[89]:
```python
warnings.simplefilter(action='ignore', category=FutureWarning)

sns.countplot(x=data['Satisfaction Score'],hue= data['Exited'])
```

[89]: <Axes: xlabel='Satisfaction Score', ylabel='count'>



people who raised the complaint and churned = 1 and their satisfaction score were 1 ,2 3, 4, 5

14 Strategies for customer retenion strategies

[63]:
```
data_banking_behaviour = data.loc[data['Exited']␣
↪==1,['CustomerId','Tenure','NumOfProducts','EstimatedSalary','Balance']]
data_banking_behaviour
```

[63]:
```
      CustomerId  Tenure  NumOfProducts  EstimatedSalary    Balance
0       15634602       2              1        101348.88       0.00
2       15619304       8              3        113931.57  159660.80
5       15574012       8              2        149756.71  113755.78
7       15656148       4              4        119346.88  115046.74
16      15737452       1              1          5097.67  132602.88
…              …     …              …                …          …
9981    15672754       3              1         53445.17  152039.70
9982    15768163       7              1        115146.40  137145.12
9991    15769959       4              1         69384.71   88381.21
9997    15584532       7              1         42085.58       0.00
9998    15682355       3              2         92888.52   75075.31

[2038 rows x 5 columns]
```

[64]:
```
data_banking_behaviour['Spent'] = data_banking_behaviour['EstimatedSalary']*␣
↪data_banking_behaviour['Tenure']- data_banking_behaviour['Balance']
data_banking_behaviour
```

[64]:
```
      CustomerId  Tenure  NumOfProducts  EstimatedSalary    Balance  \
0       15634602       2              1        101348.88       0.00
2       15619304       8              3        113931.57  159660.80
5       15574012       8              2        149756.71  113755.78
7       15656148       4              4        119346.88  115046.74
16      15737452       1              1          5097.67  132602.88
…              …     …              …                …          …
9981    15672754       3              1         53445.17  152039.70
9982    15768163       7              1        115146.40  137145.12
9991    15769959       4              1         69384.71   88381.21
9997    15584532       7              1         42085.58       0.00
9998    15682355       3              2         92888.52   75075.31

           Spent
0      202697.76
2      751791.76
5     1084297.90
```

```
7      362340.78
16    -127505.21
…            …
9981     8295.81
9982   668879.68
9991   189157.63
9997   294599.06
9998   203590.25

[2038 rows x 6 columns]
```

[65]:
```python
data_banking_behaviour[data_banking_behaviour['Balance'] < 0 ]
```

[65]:
```
Empty DataFrame
Columns: [CustomerId, Tenure, NumOfProducts, EstimatedSalary, Balance, Spent]
Index: []
```

we don't have any negative balance account it shows we have no customer who have dfaulted while exiting the bank after using its service

[66]:
```python
data_banking_behaviour[data_banking_behaviour['Spent'] < 0 ]
```

[66]:

	CustomerId	Tenure	NumOfProducts	EstimatedSalary	Balance	Spent
16	15737452	1	1	5097.67	132602.88	-127505.21
35	15794171	0	1	27822.99	134264.04	-134264.04
54	15569590	1	1	40014.76	98495.72	-58480.96
70	15703793	2	4	28373.86	133745.44	-76997.72
127	15782688	0	1	46824.08	148507.24	-148507.24
…	…	…	…	…	…	…
9863	15726179	5	2	3497.43	131433.33	-113946.18
9882	15785490	3	1	16281.68	105229.72	-56384.68
9920	15673020	3	1	738.88	204510.94	-202294.30
9924	15578865	5	1	6985.34	107959.39	-73032.69
9947	15732202	1	2	73124.53	83503.11	-10378.58

[350 rows x 6 columns]

The above analysis shows the out of total people who left 350 are of people whose balance were more than their estimated salary according to Their bank tenure usage which speaks that apart from their estimated salary they have had more balance not from salary but from other assets

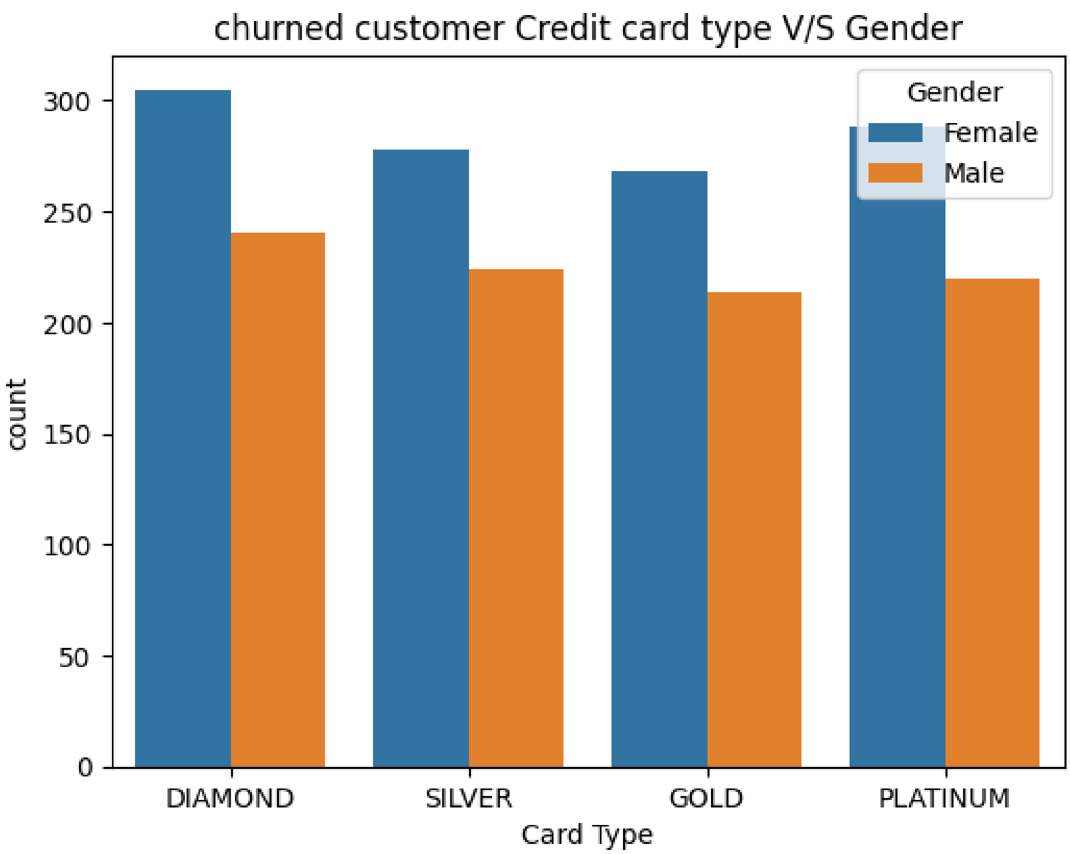
bank is at loss for loosing such customers

[ ]:

15 Lets check the people whose balance were not zero or less but have complaint and churned out of the bank with different credit card

[88]:
```
warnings.simplefilter(action='ignore', category=FutureWarning)

sns.countplot(x = data[data['Exited'] == 1]['Card Type'],hue = data['Gender'])
plt.title("churned customer Credit card type V/S Gender")
```

[88]:
```
Text(0.5, 1.0, 'churned customer Credit card type V/S Gender')
```



[68]:
```
data.loc[data['Exited']== 1,['Balance','Complain','Card Type','Satisfaction␣
↪Score']]
```

[68]:

	Balance	Complain	Card Type	Satisfaction Score
0	0.00	1	DIAMOND	2
2	159660.80	1	DIAMOND	3
5	113755.78	1	DIAMOND	5
7	115046.74	1	DIAMOND	2

```
16     132602.88         0    SILVER                 2
…          …        …       …                 …
9981   152039.70         1      GOLD                 3
9982   137145.12         1      GOLD                 4
9991    88381.21         1      GOLD                 3
9997        0.00         1    SILVER                 3
9998    75075.31         1      GOLD                 2

[2038 rows x 4 columns]
```

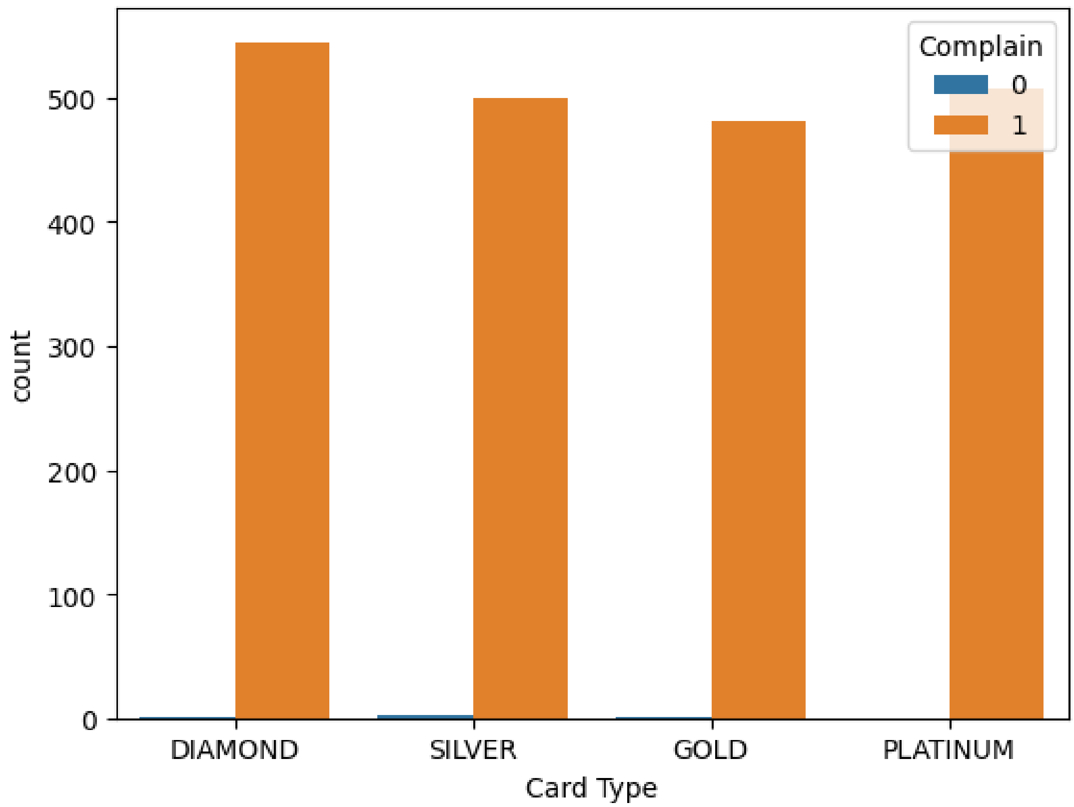
[69]:
```python
pd.crosstab(index = data[data['Exited'] == 1]['Card Type'],columns =␣
↪data[data['Exited'] == 1]['Complain'],margins=True).reset_index()
```

[69]:

Complain	Card Type	0	1	All
0	DIAMOND	1	545	546
1	GOLD	1	481	482
2	PLATINUM	0	508	508
3	SILVER	2	500	502
4	All	4	2034	2038

[87]:
```python
warnings.simplefilter(action='ignore', category=FutureWarning)

sns.countplot(x = data[data['Exited'] == 1]['Card Type'],hue =␣
↪data[data['Exited'] == 1]['Complain'])
```

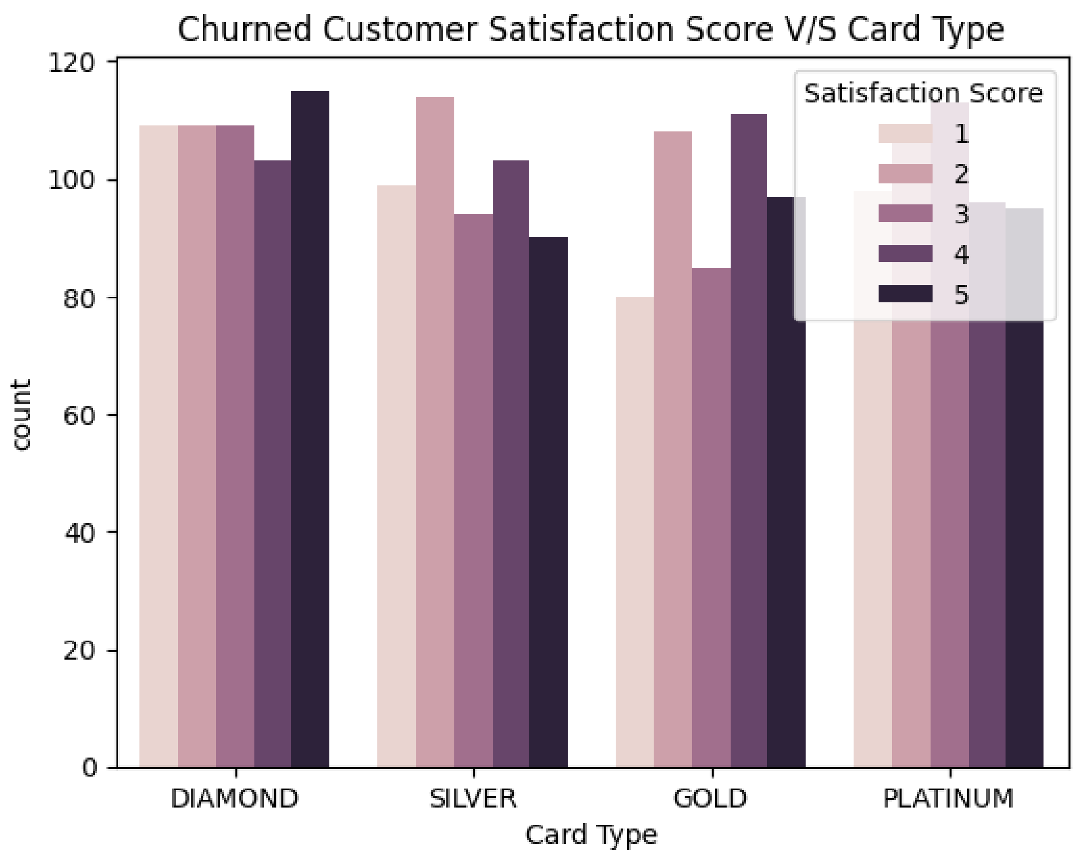
[87]:
```
<Axes: xlabel='Card Type', ylabel='count'>
```



satisfaction score for Customer who churned out and have complained to banking services were visualize as below shown

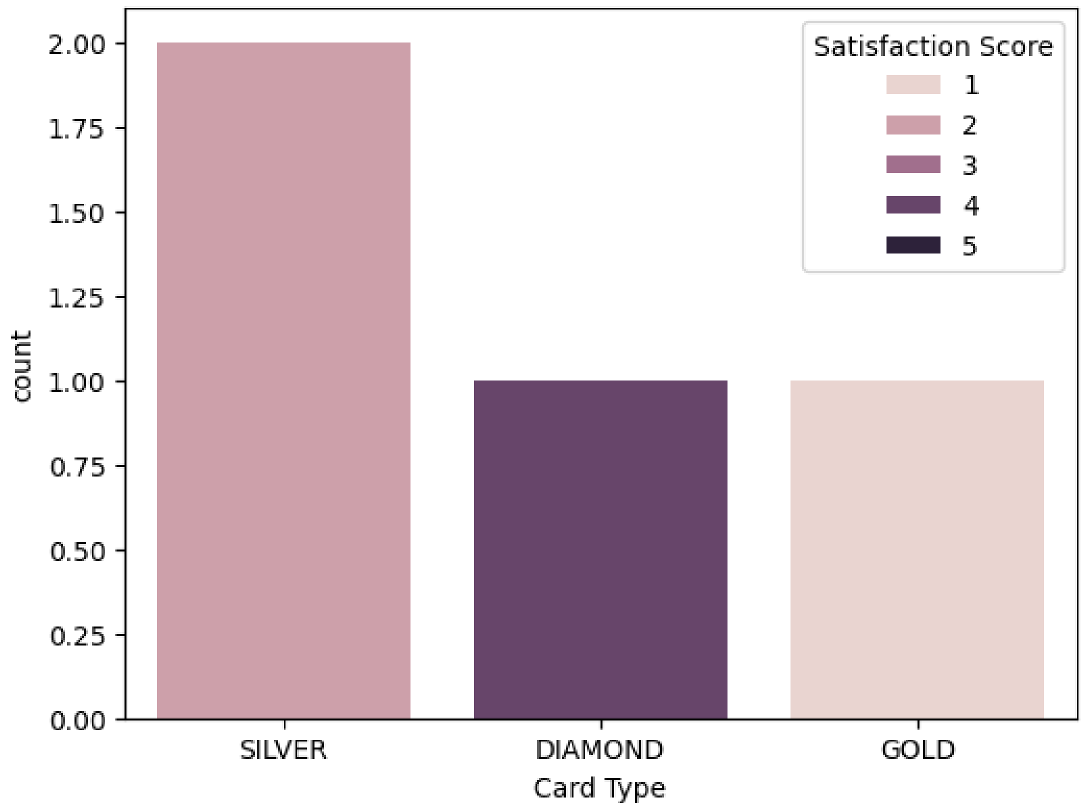
[86]:
```
warnings.simplefilter(action='ignore', category=FutureWarning)

sns.countplot(x = data[(data['Exited'] ==1) & (data['Complain']==1)]['Card␣
↪Type'],hue = data[data['Exited'] == 1]['Satisfaction Score'])
plt.title('Churned Customer Satisfaction Score V/S Card Type')
```

[86]:
```
Text(0.5, 1.0, 'Churned Customer Satisfaction Score V/S Card Type')
```



[85]:
```
warnings.simplefilter(action='ignore', category=FutureWarning)

sns.countplot(x = data[(data['Exited'] ==1) & (data['Complain']==0)]['Card␣
↪Type'],hue = data[data['Exited'] == 1]['Satisfaction Score'])
```

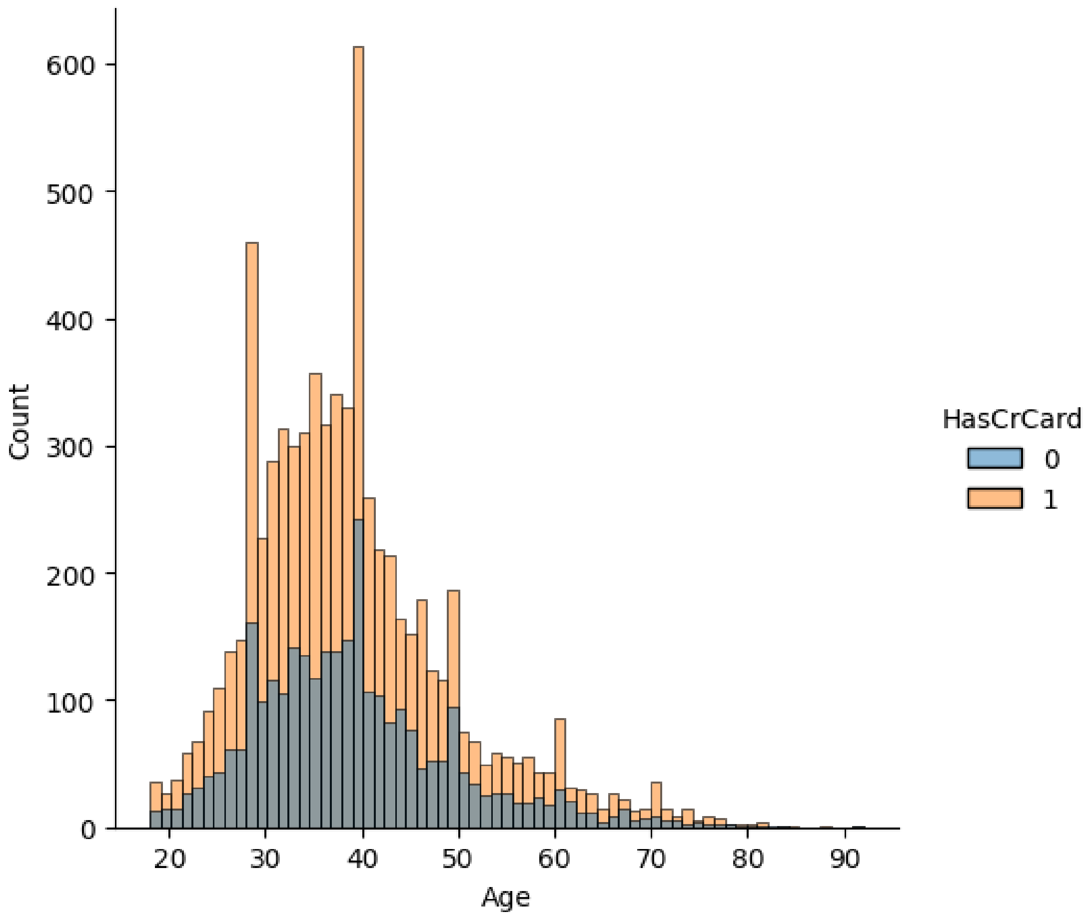
[85]: <Axes: xlabel='Card Type', ylabel='count'>



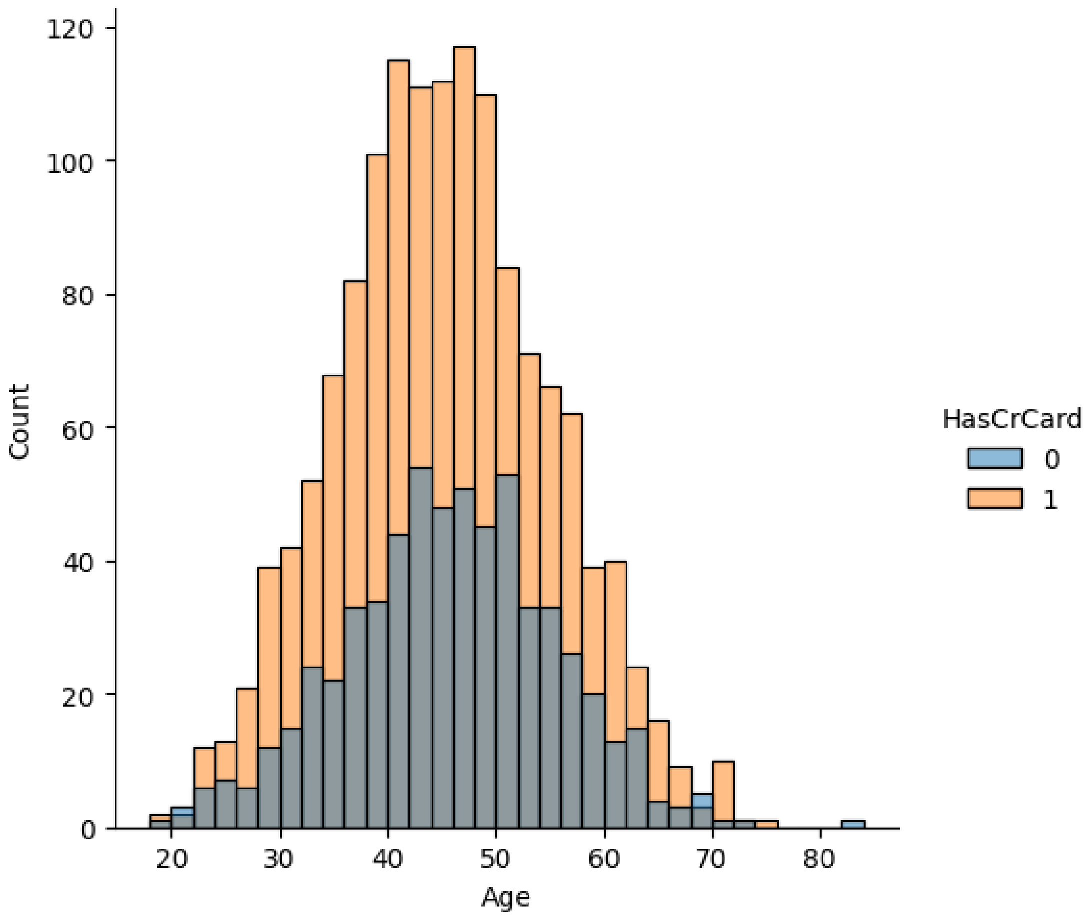
16 Checking Credit card Age wise

[84]:
```
warnings.simplefilter(action='ignore', category=FutureWarning)

plt.figure(figsize=(5, 5))
sns.displot(data=data, x="Age", hue="HasCrCard")
plt.figure(figsize=(5, 5))  # Create a new figure
sns.displot(data=data[data["Exited"] == 1], x="Age", hue="HasCrCard")
plt.figure(figsize=(5, 5))
sns.displot(data=data[data["Exited"] == 1], x="Age", hue="IsActiveMember")
```

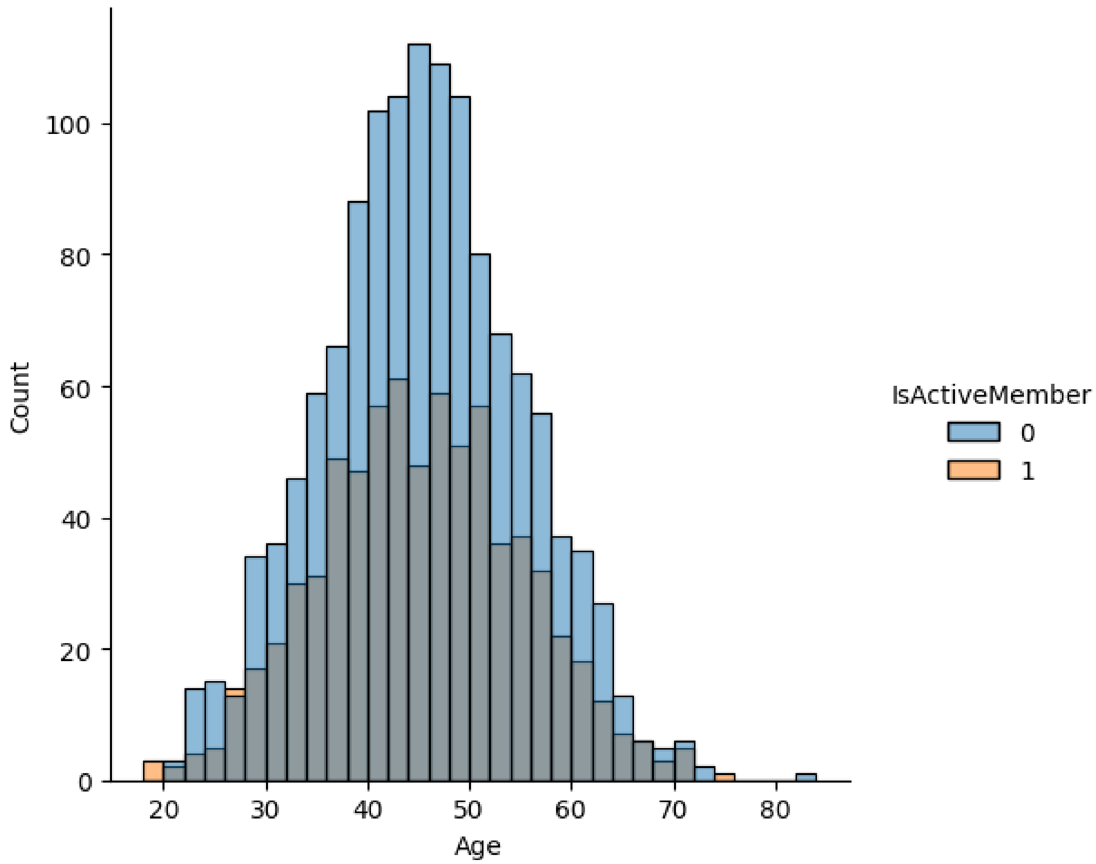
[84]:
```
<seaborn.axisgrid.FacetGrid at 0x7869f1cafd60>

<Figure size 500x500 with 0 Axes>
```




```
<Figure size 500x500 with 0 Axes>
```

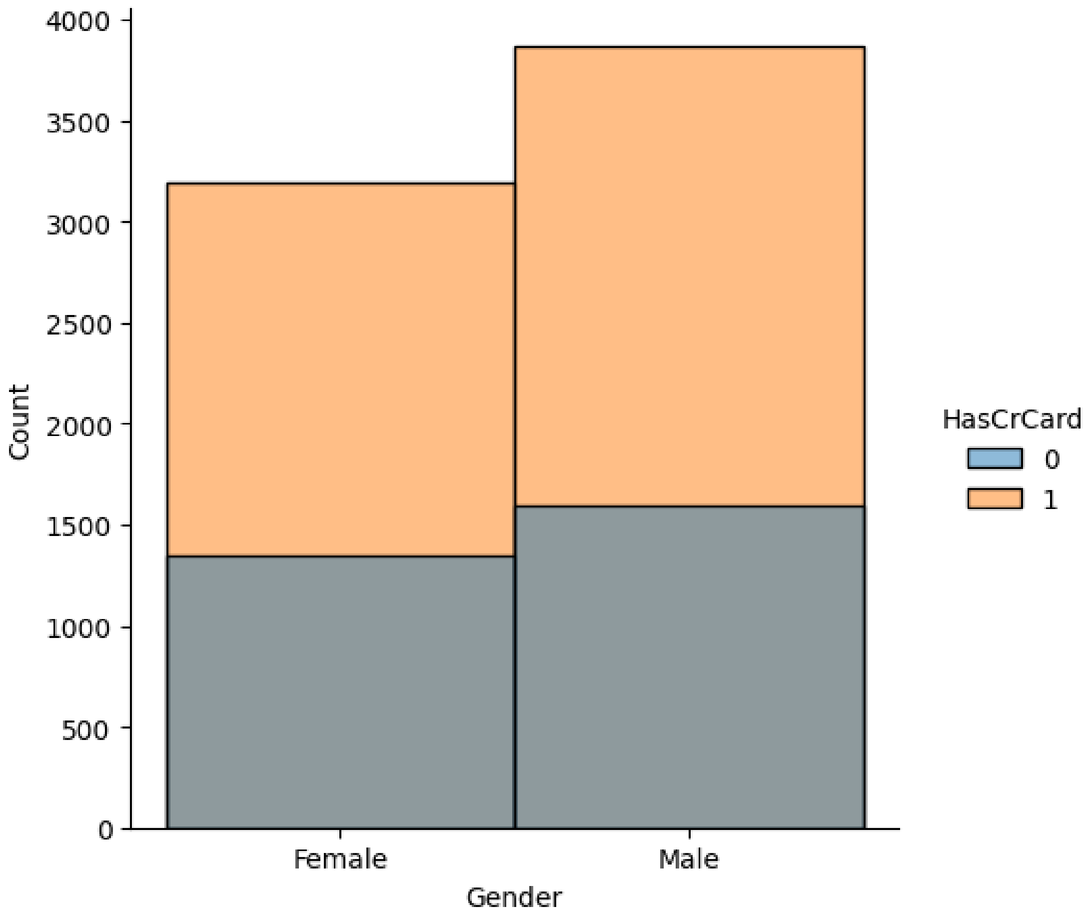



```
<Figure size 500x500 with 0 Axes>
```

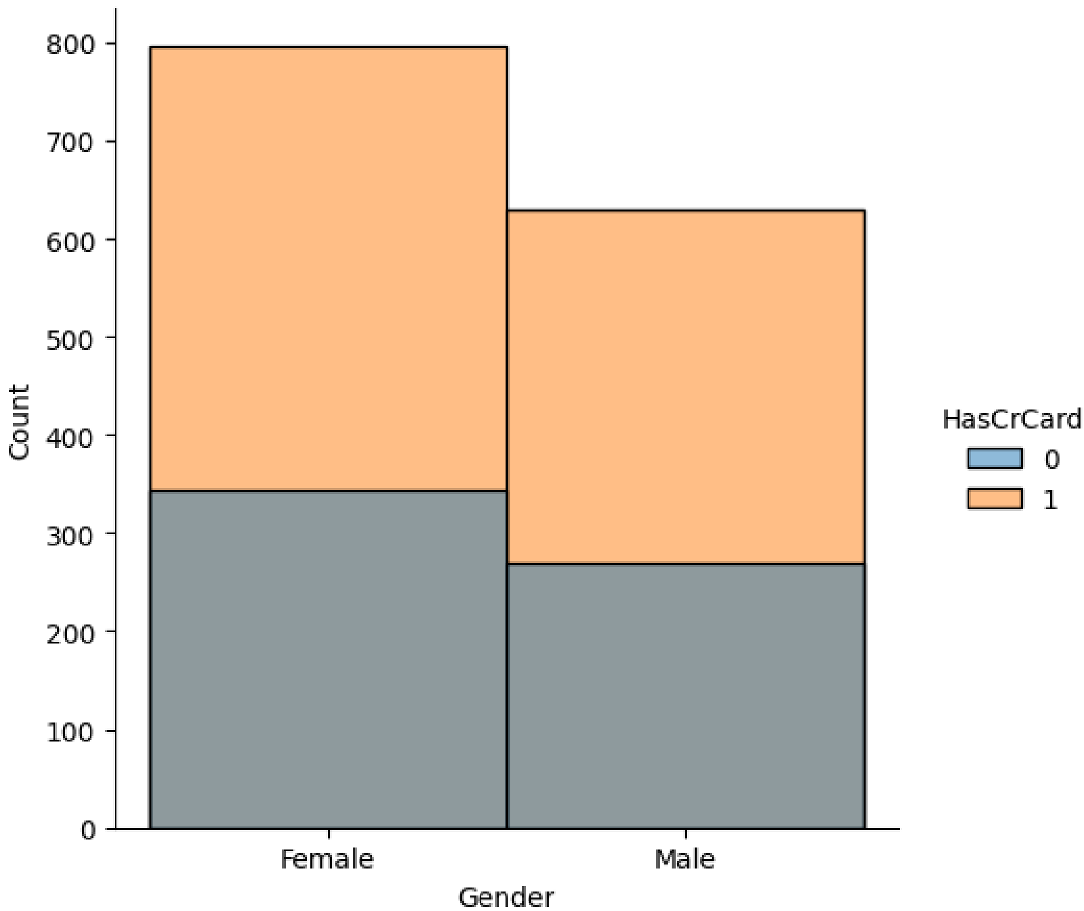


[83]:
```
warnings.simplefilter(action='ignore', category=FutureWarning)

plt.figure(figsize=(5, 5))
sns.displot(data=data, x="Gender", hue="HasCrCard")
plt.figure(figsize=(5, 5))  # Create a new figure
sns.displot(data=data[data["Exited"] == 1], x="Gender", hue="HasCrCard")
plt.figure(figsize=(5, 5))
sns.displot(data=data[data["Exited"] == 1], x="Gender", hue="IsActiveMember")
```

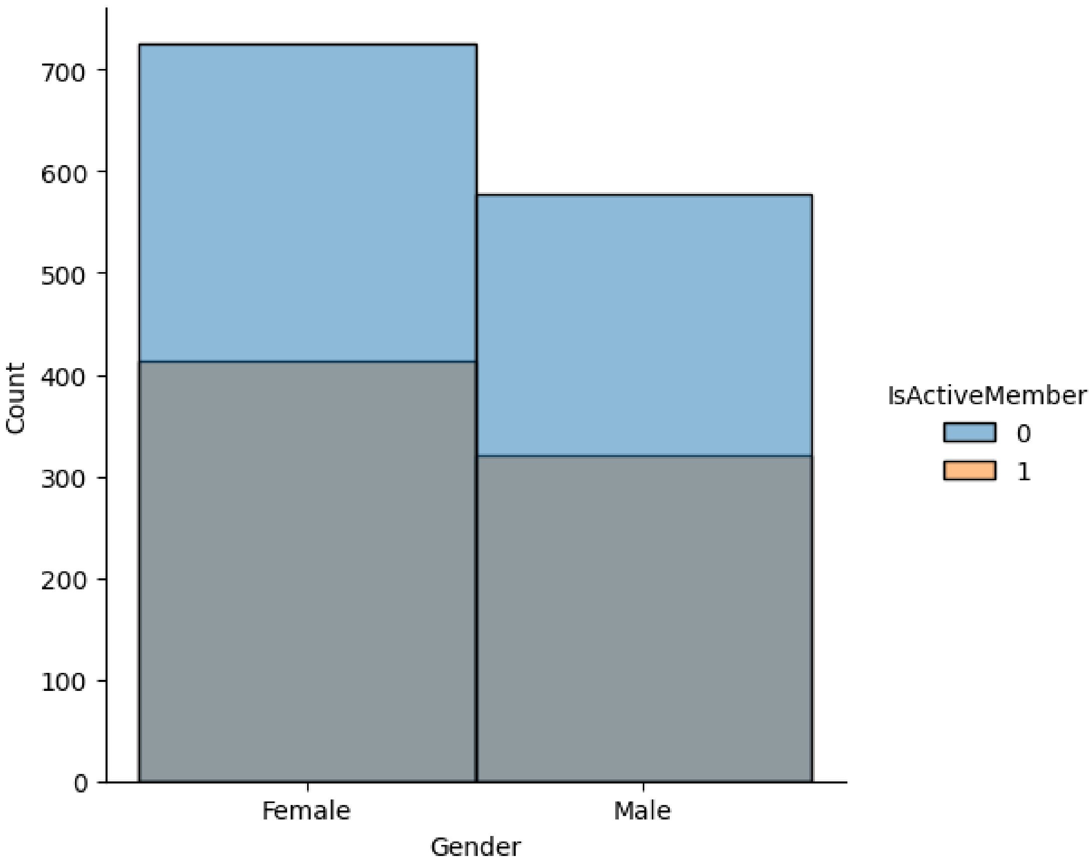
[83]:
```
<seaborn.axisgrid.FacetGrid at 0x7869ed5027d0>

<Figure size 500x500 with 0 Axes>
```




```
<Figure size 500x500 with 0 Axes>
```



<Figure size 500x500 with 0 Axes>



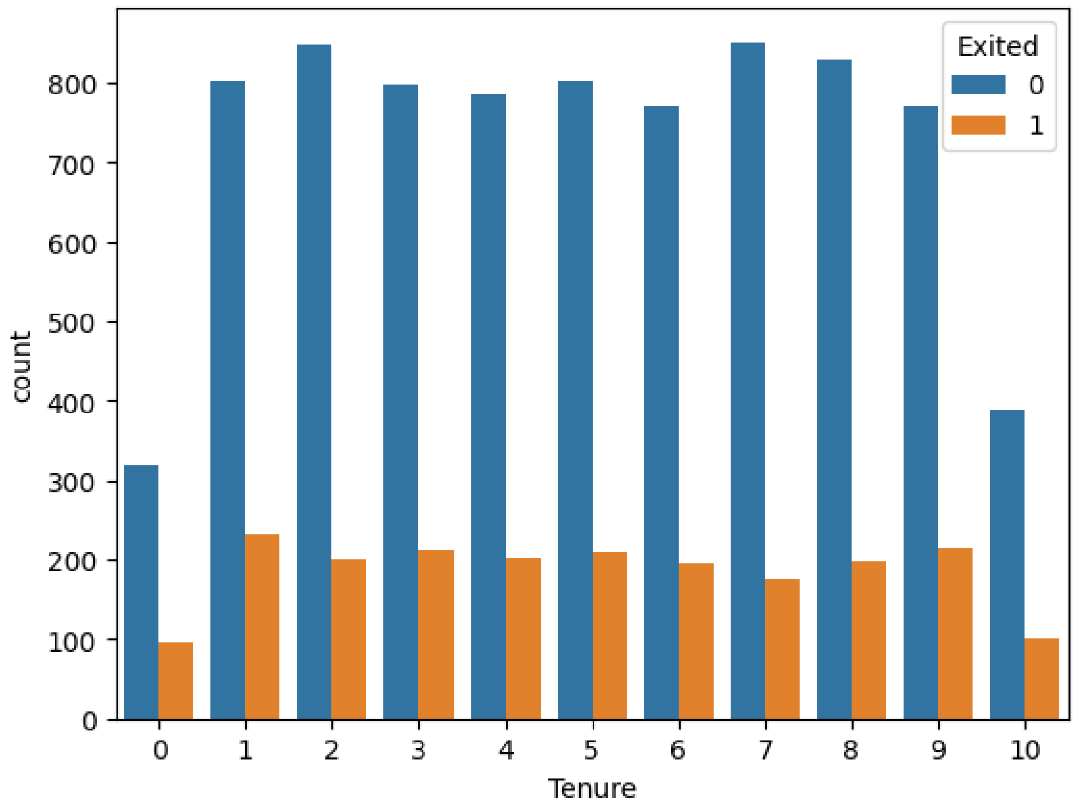
17 Descriptive analysis

18 Churn rate

for different type of tenures

[82]:
```
warnings.simplefilter(action='ignore', category=FutureWarning)

sns.countplot(x=data['Tenure'],hue= data['Exited'])
```

[82]: `<Axes: xlabel='Tenure', ylabel='count'>`



[76]:
```
pd.crosstab(columns = data['Tenure'],index= data['Exited'],margins = True)
```

[76]:
```
Tenure    0     1     2     3    4     5    6     7     8    9   10    All
Exited
0       318   803   847   796  786   803  771   851   828  770  389   7962
1        95   232   201   213  203   209  196   177   197  214  101   2038
All     413  1035  1048  1009  989  1012  967  1028  1025  984  490  10000
```

[77]:
```
churn_data = pd.crosstab(columns = data['Tenure'],index=␣
 ↪data['Exited'],normalize = 'columns')
churn_data
```

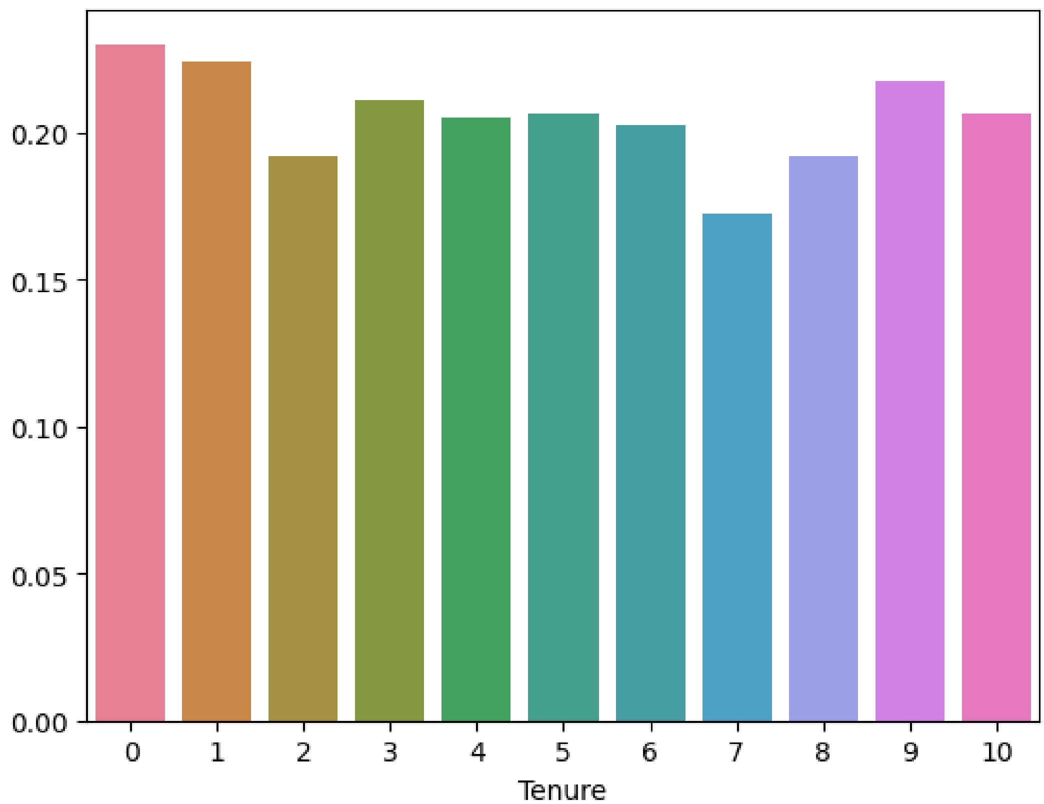
[77]:
```
Tenure         0         1         2       3         4         5         6   \
Exited
0       0.769976  0.775845  0.808206  0.7889  0.794742  0.793478  0.797311
1       0.230024  0.224155  0.191794  0.2111  0.205258  0.206522  0.202689

Tenure         7         8        9        10
Exited
0       0.827821  0.807805  0.78252  0.793878
1       0.172179  0.192195  0.21748  0.206122
```

[78]:
```
churn_data[1:2].reset_index()
```

[78]:
```
Tenure  Exited         0         1         2       3         4         5  \
0            1  0.230024  0.224155  0.191794  0.2111  0.205258  0.206522

Tenure         6         7         8        9        10
0       0.202689  0.172179  0.192195  0.21748  0.206122
```

from above table the 2nd rows show the churning rate for every different tenure

[81]:
```
warnings.simplefilter(action='ignore', category=FutureWarning)

sns.barplot(churn_data[1:2].reset_index().drop('Exited',axis = 1))
plt.show()
```



The Customer churning are dependent on Variables like Credit Score ,Age and Geography Tenure has no relation with customer who churned

Recommendation:

Focus on Customer with Credit score between 600-700 as they are more likely to churn. Keep a guard rail check on the 30-40 year of age people as they are loyal customers the Age from 40 – 50 were the mostly who churned so incentivize them too so they not churned in future Gender has an impact on churning so and incentives for gender can benefits the customer Focus on credit card service and bring innovation as people who left were most of who have credit card with them

[ ]:

#Observation & Recommendation:

The Customer churning are dependent on Variables like Credit Score ,Age and Geography, Balance Tenure has no relation with customer who churned

Recommendation Focus on Customer with Credit score between 600-700 as they are more likely to churn.

Keep a guard rail check on the 30-40 year of age people as they are loyal customers ,the Age from 40 – 50 were the mostly who churned so incentivize them too so they not churned in future

Gender has an impact on churning so an incentives for both gender can benefits the customer

Focus on credit card service and bring innovation as people who left were most of who have credit card with them

Geography especially France as most customer centric and Balance should be considered for predicting the next possible churn

19 Conclusion

Customer leaving the bank makes a significant impact on firm reputation and leads to financial loss and in order to deal with this crisis a comprehensive data analysis needed for making an informed decision by decision makers

[ ]: