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.

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

```
Downloading...
From: https://drive.google.com/uc?id=1q1Mh3Mm4kv1LitxWcdY6--gNHVmuAfPP
To: /content/Bank-Records.csv

  0% 0.00/837k [00:00<?, ?B/s]
100% 837k/837k [00:00<00:00, 85.0MB/s]
```

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

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

Out[4]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	A
0	1	15634602	Hargrave	619	France	Female	
1	2	15647311	Hill	608	Spain	Female	
2	3	15619304	Onio	502	France	Female	
3	4	15701354	Boni	699	France	Female	
4	5	15737888	Mitchell	850	Spain	Female	
...	...	...	...	...	...	...	
9995	9996	15606229	Obijiaku	771	France	Male	
9996	9997	15569892	Johnstone	516	France	Male	
9997	9998	15584532	Liu	709	France	Female	
9998	9999	15682355	Sabbatini	772	Germany	Male	
9999	10000	15628319	Walker	792	France	Female	

10000 rows × 18 columns

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

Out[5]: (10000, 18)

In [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
```

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

Out[7]:
```
10000
```

Performing Basic Exploring data analysis

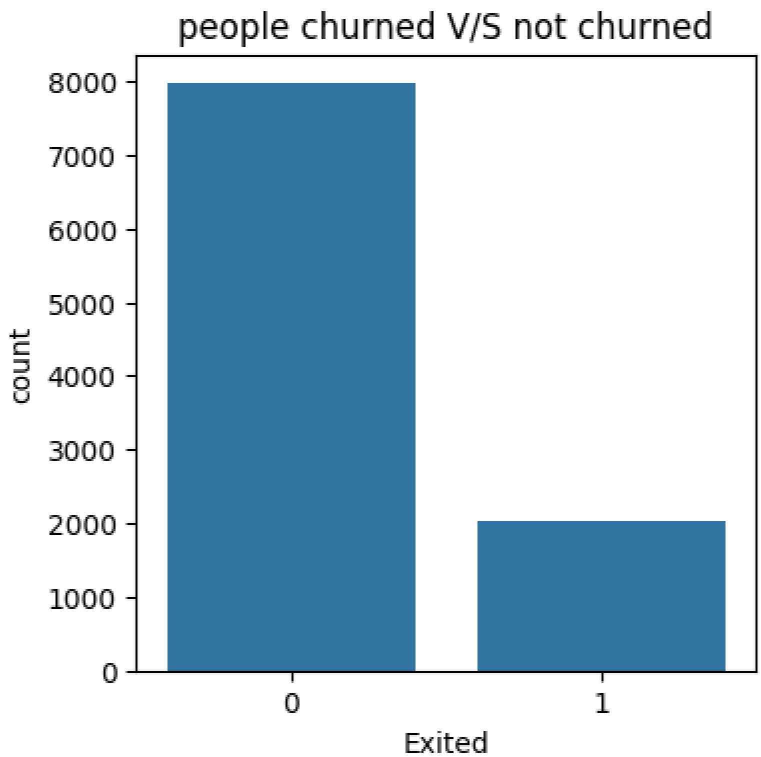
In [8]:
```
data[['CustomerId','Exited']]
```

Out[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 × 2 columns

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



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

Out[10]:

	count
Exited	
0	7962
1	2038

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

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

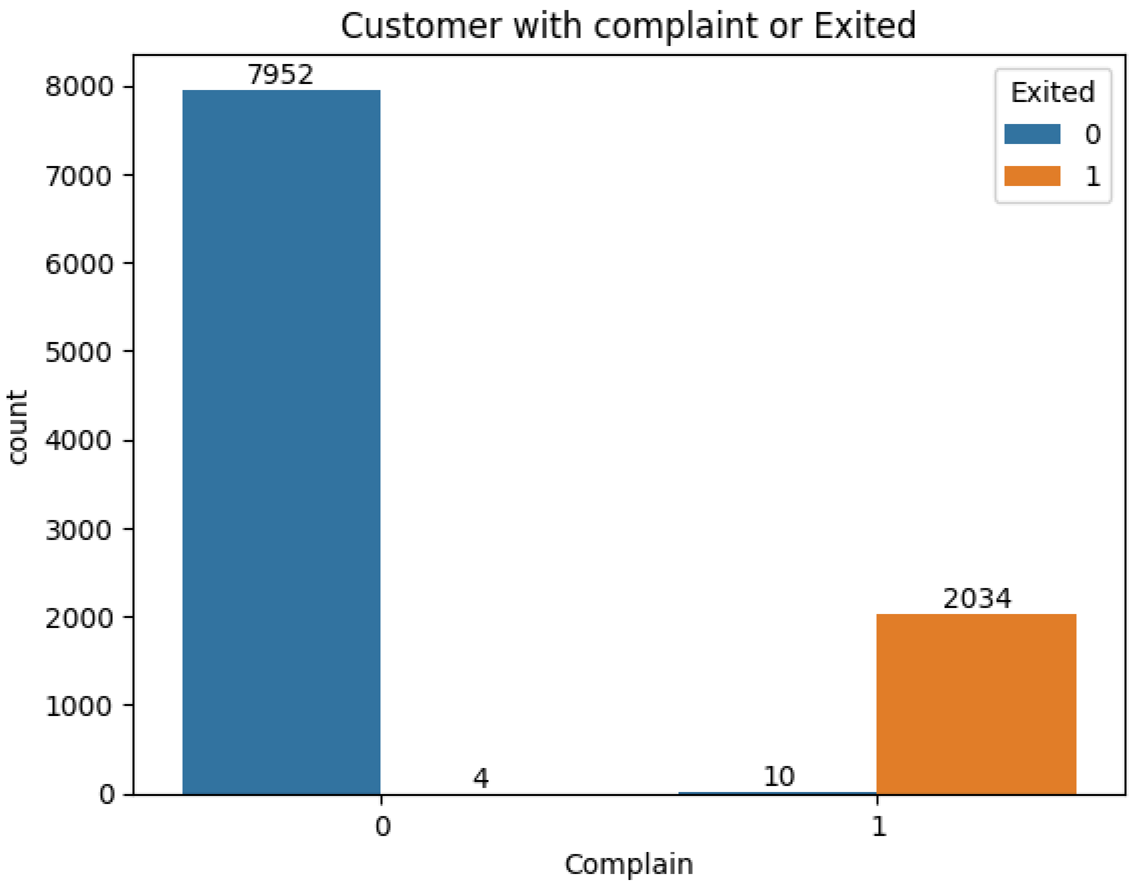
Out[11]:

Complain	**0**	**1**
Exited		
0	7952	10
1	4	2034

In [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

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

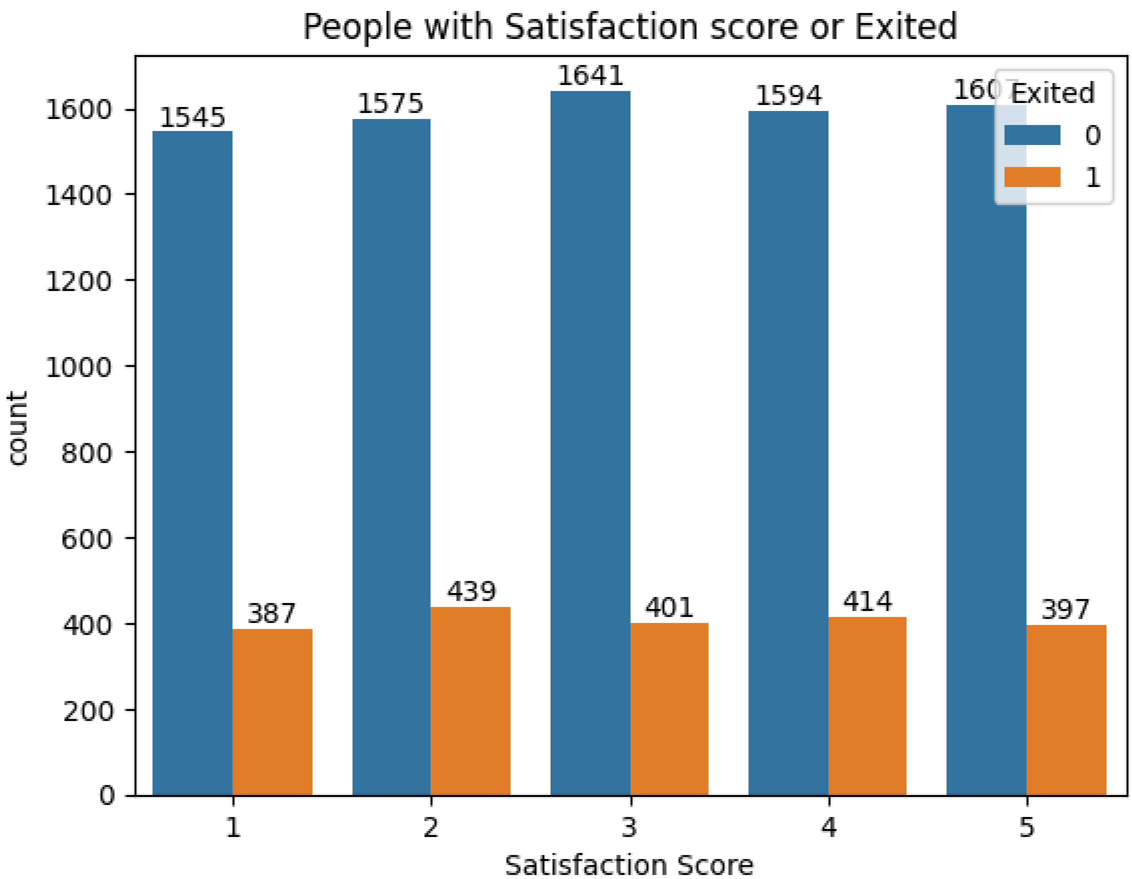
Out[13]:

Satisfaction Score	1	2	3	4	5
Exited					
0	1545	1575	1641	1594	1607
1	387	439	401	414	397

In [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()
```



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

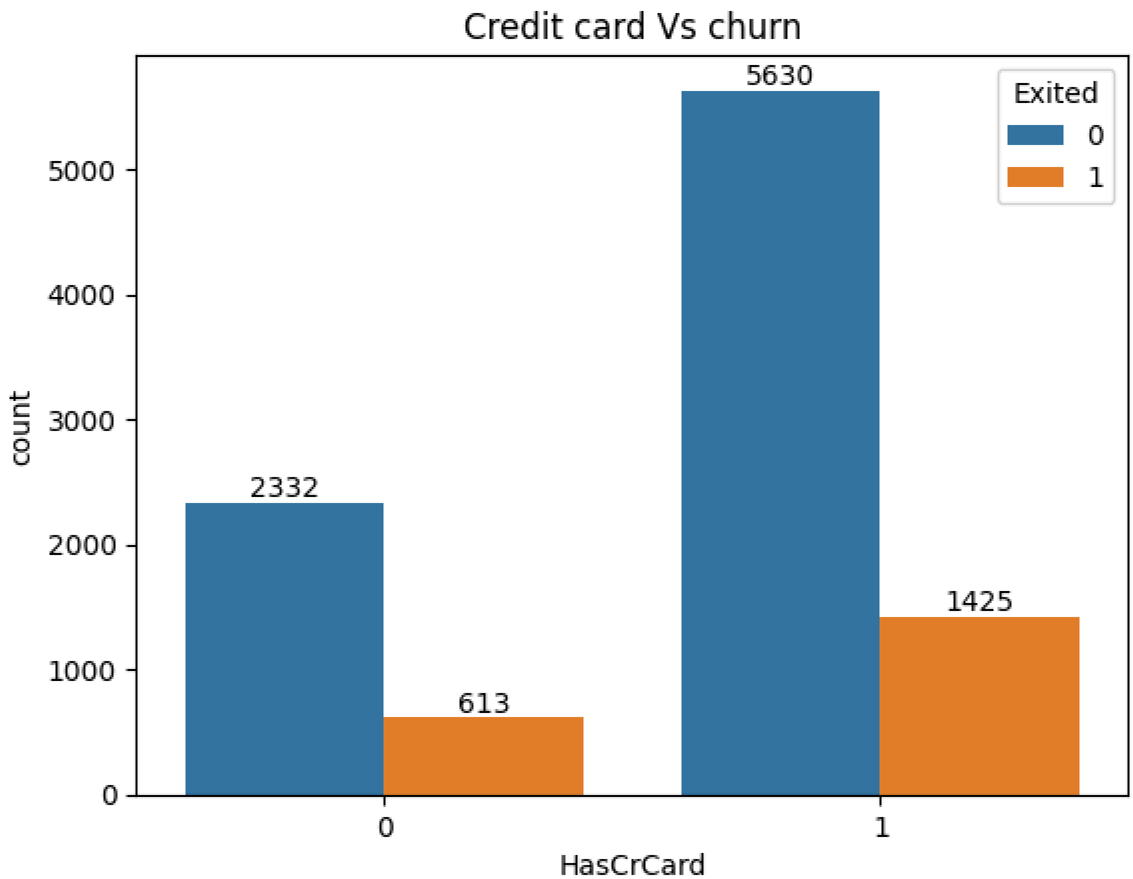
Out[15]:

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

In [102…
```python
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()
```



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

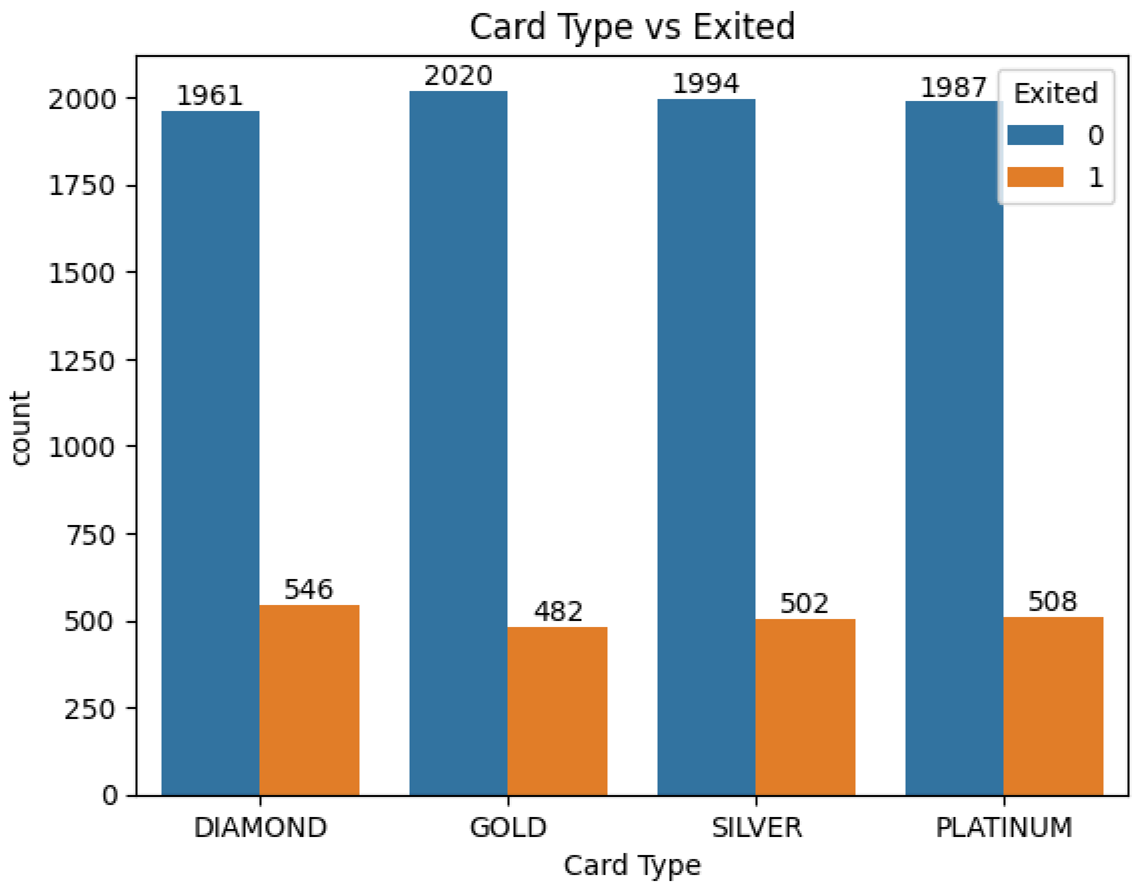
Out[17]:

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

In [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()
```



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

Out[19]: 850

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

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

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

Out[22]:

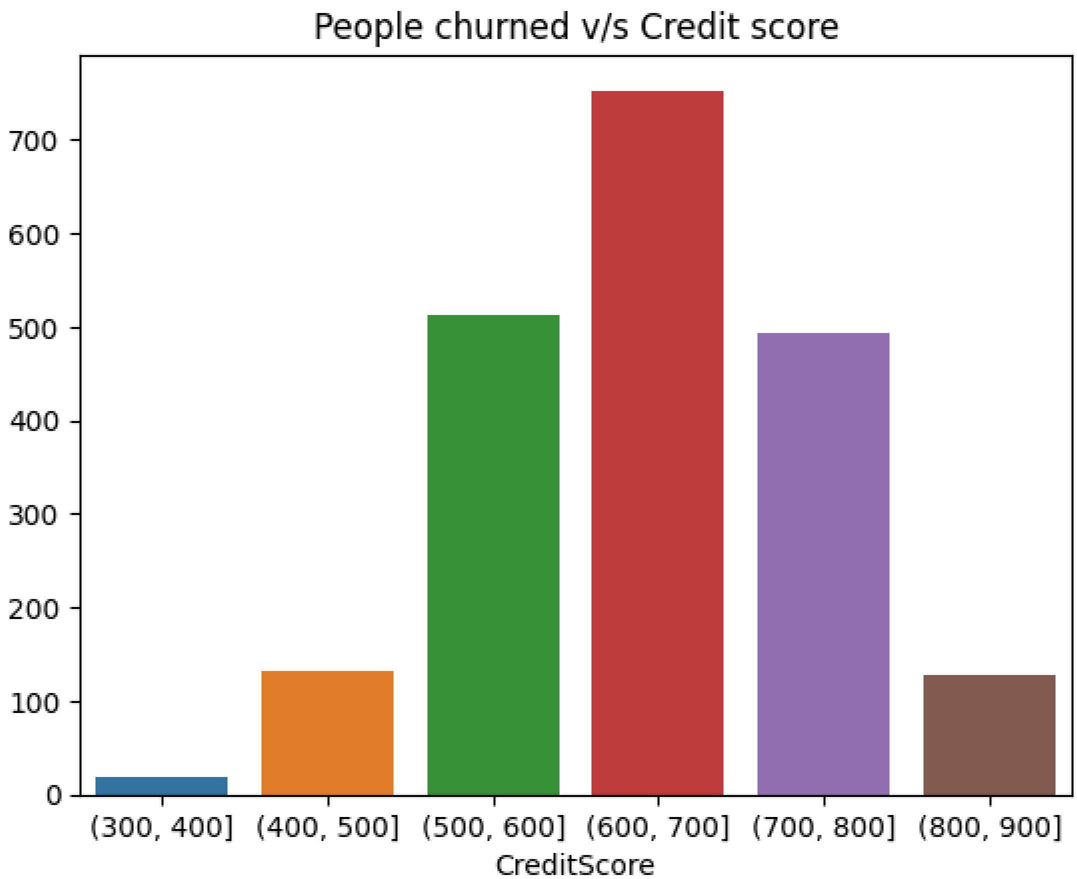
CreditScore	**(300, 400]**	**(400, 500]**	**(500, 600]**	**(600, 700]**	**(700, 800]**	**(800, 900]**
Exited						
1	19	133	513	753	493	127

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

In [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')
```

Out[100… Text(0.5, 1.0, 'People churned v/s Credit score')



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

Out[24]:

Gender	Female	Male
Exited		
0	3404	4558
1	1139	899

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

Out[25]:

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

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

Out[26]:

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

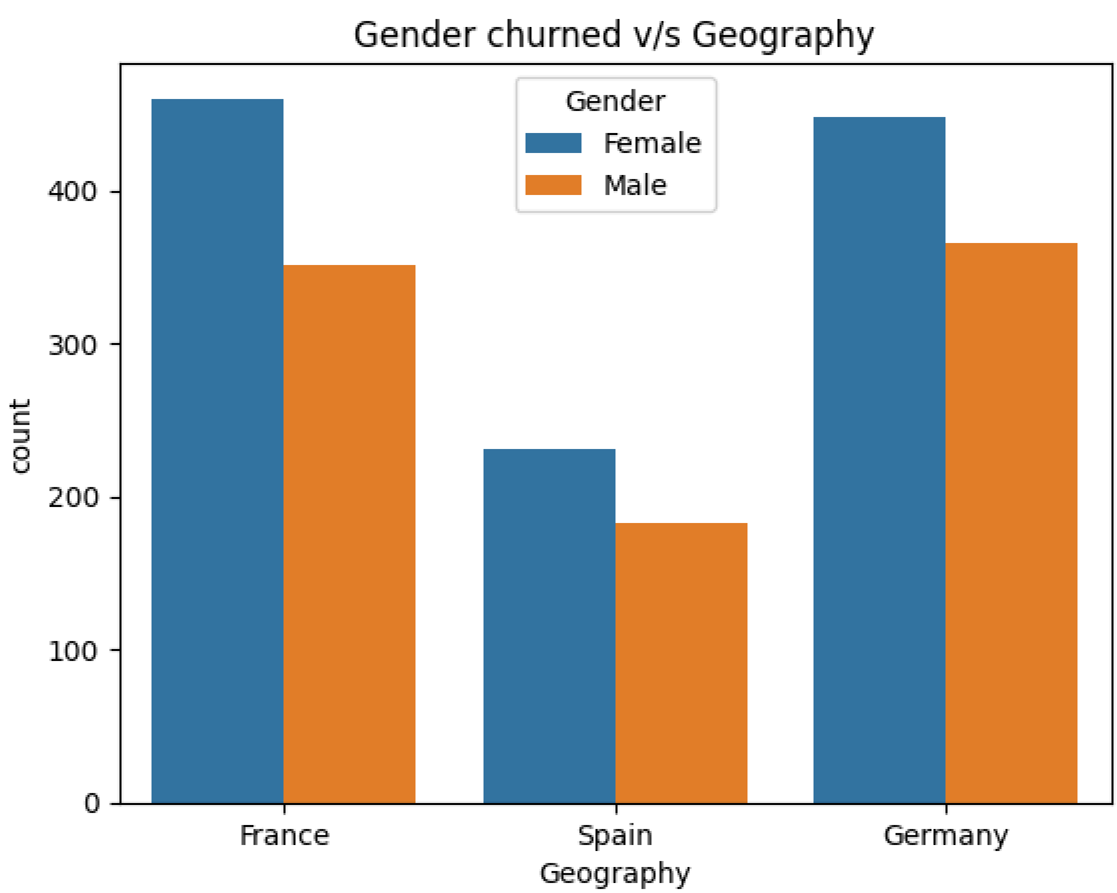
In [27]:
```
pd.crosstab(columns = [data['Geography'],data['Gender']],index = data['Exite
```

Out[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

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

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



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

Out[30]:

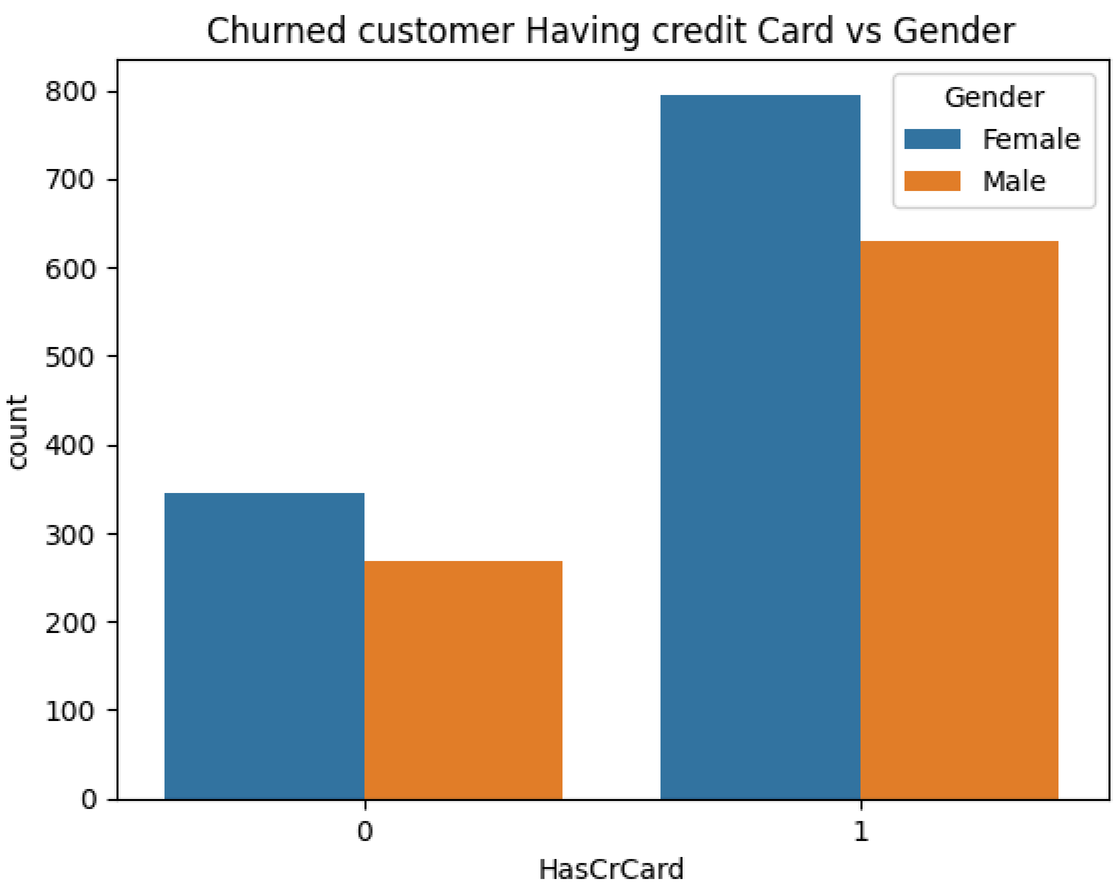
HasCrCard	0		1	
Gender	**Female**	**Male**	**Female**	**Male**
Exited				
0	1007	1325	2397	3233
1	344	269	795	630

In [98]:

```
warnings.simplefilter(action='ignore', category=FutureWarning)

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

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

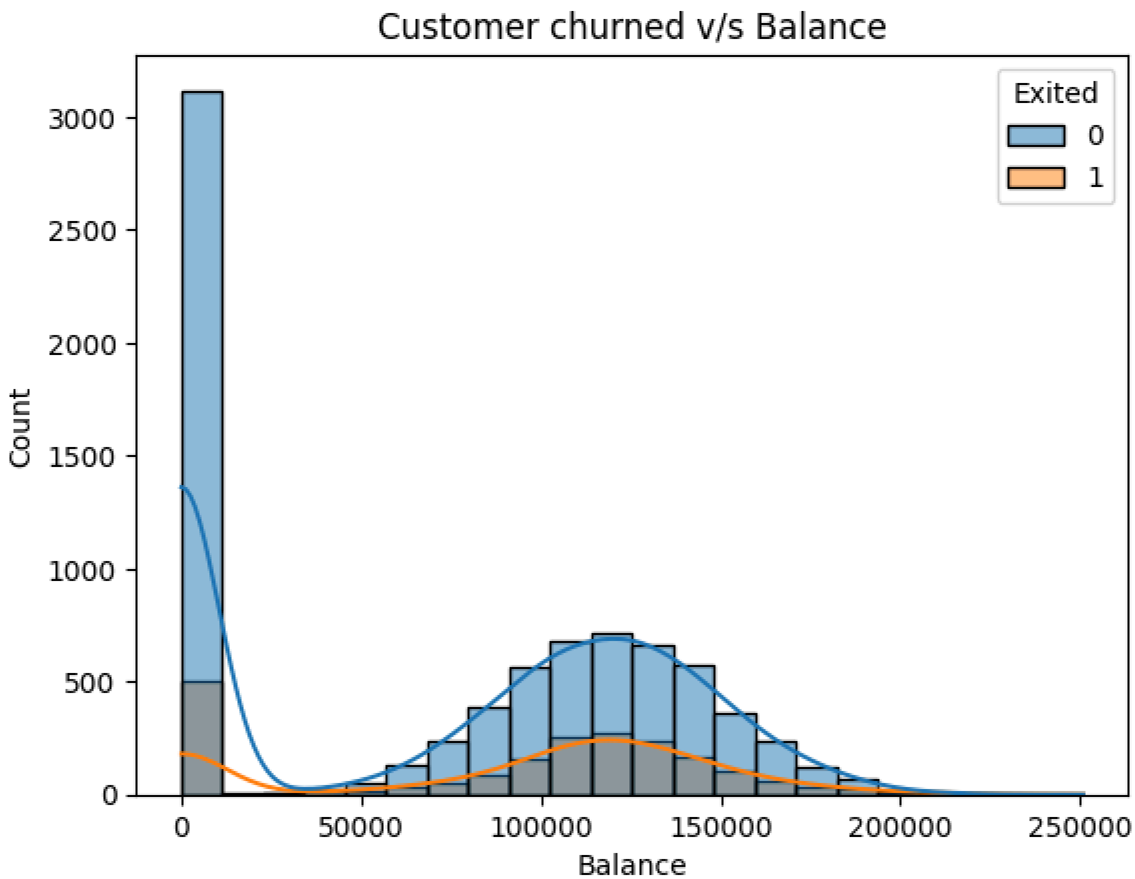


In [97]:

```
warnings.simplefilter(action='ignore', category=FutureWarning)

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

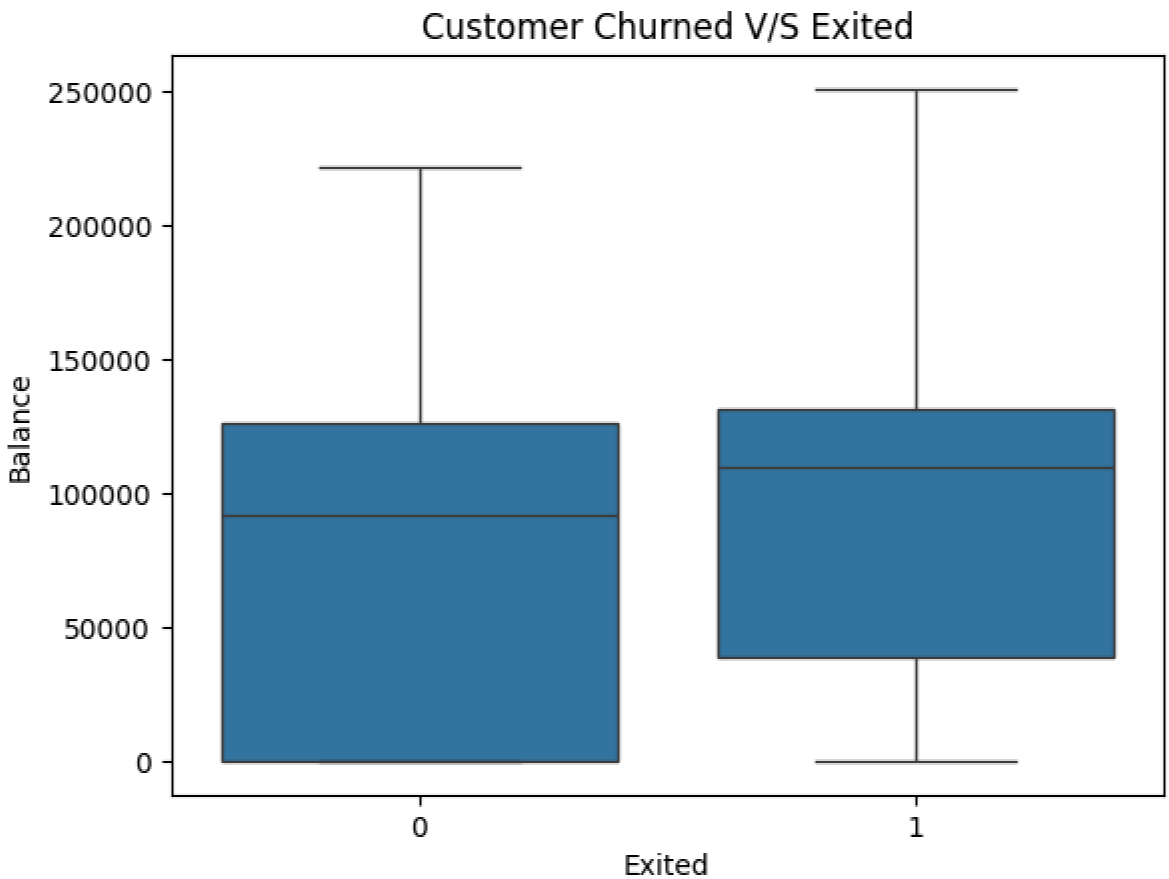
Out[97]: Text(0.5, 1.0, 'Customer churned v/s Balance')



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

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

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



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

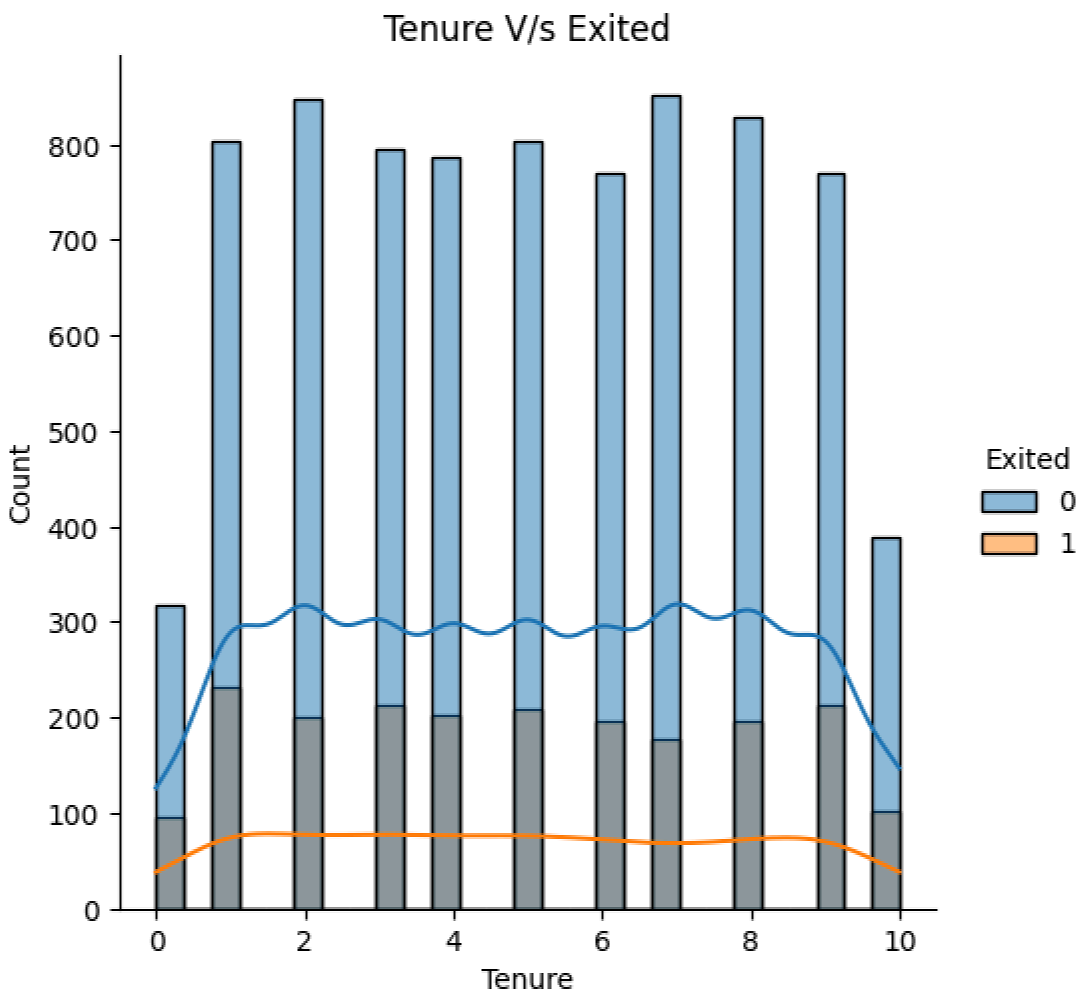
Out[34]:

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

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

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

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



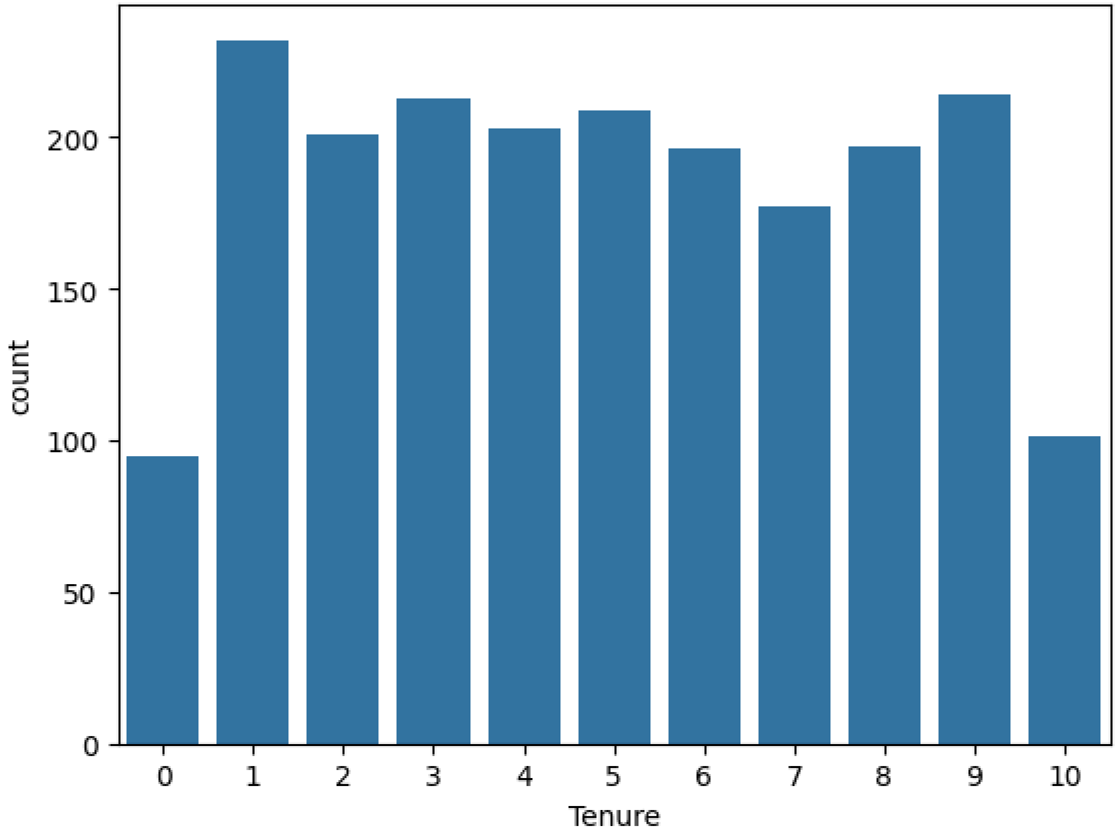
In [36]:
```
data[data['Exited']==1]['Tenure'].value_counts().reset_index()
```

Out[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

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

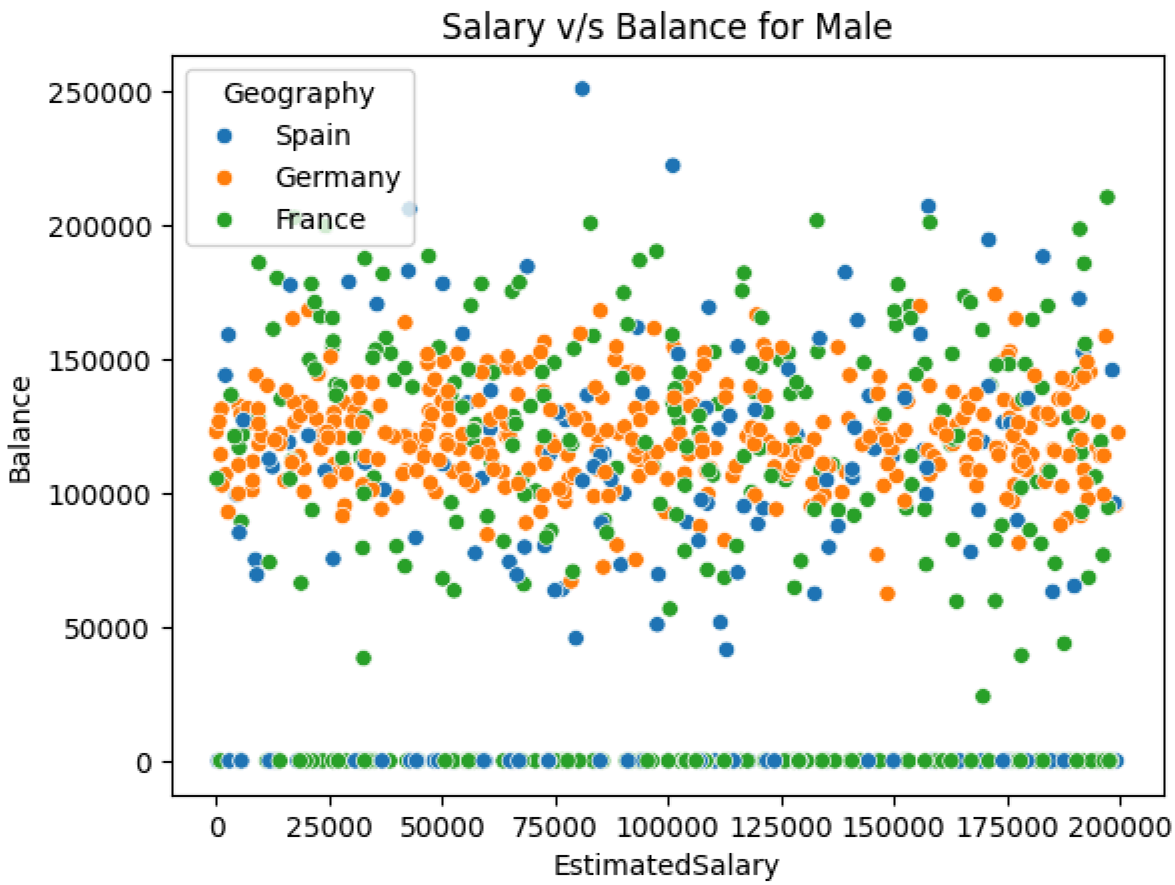
Out[37]: <Axes: xlabel='Tenure', ylabel='count'>



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

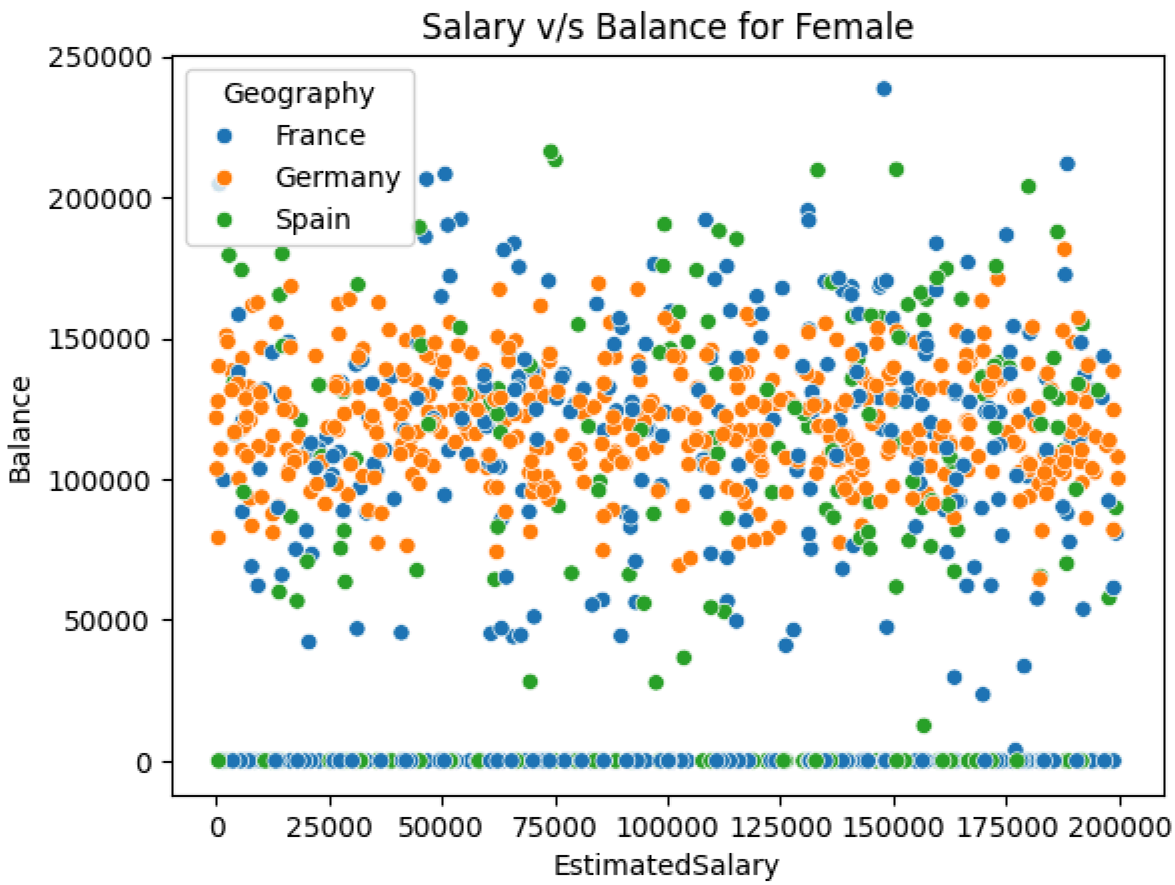
Male

In [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()
```



Female

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



lets create functions for our Hypothesis test inorder to check correlations

Credit score vs Customer churn.

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

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

Out[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 × 2 columns

In [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

In [43]:
```
t_stats, p_value = ttest_ind(data[data['Exited'] == 0]['CreditScore'],data[c
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 0.007422037242734243
Null hypothesis is rejected
```

Age vs Customer churn

we will use ttest_ind

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

Out[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 × 2 columns

H0: Customer churn is independent of Age

Ha: Customer churn is dependent of Age

In [45]:
```
t_stats, p_value = ttest_ind(data[data['Exited'] == 0]['Age'],data[data['Exi
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")
```

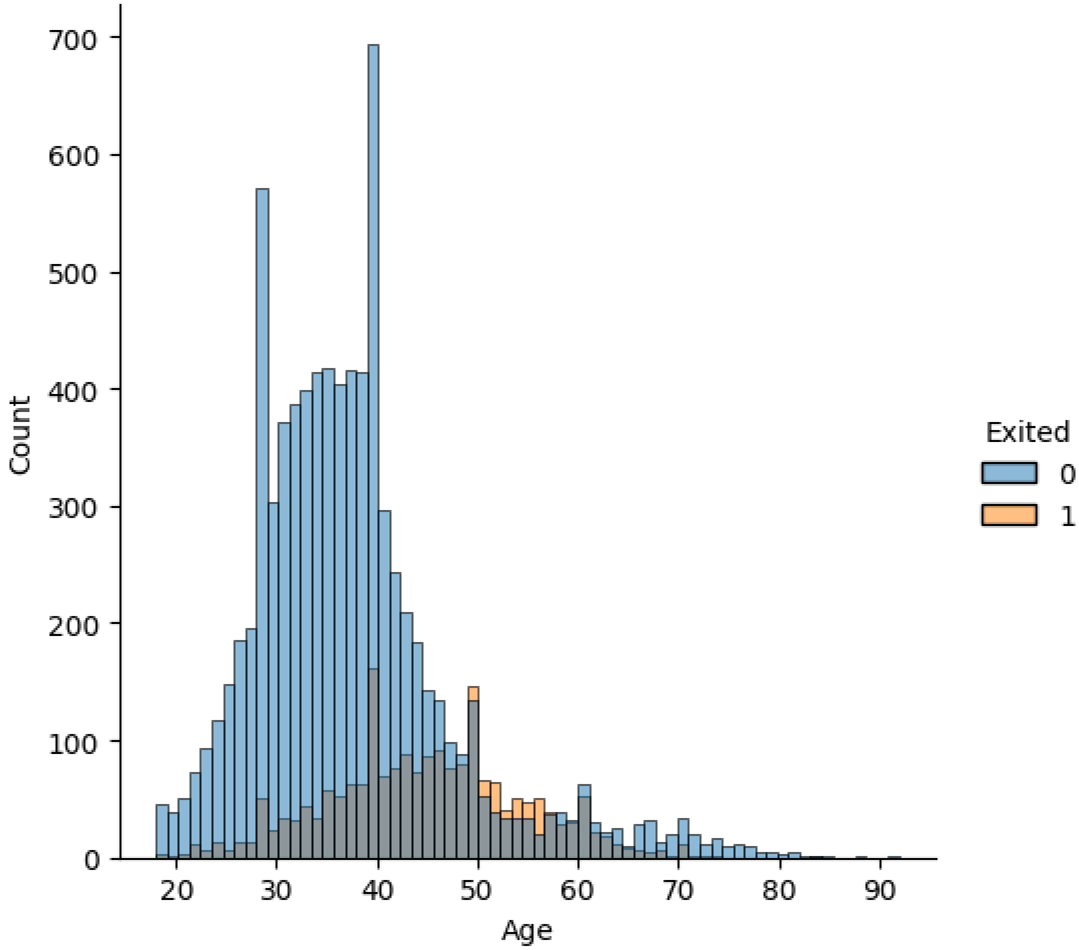
```
t_stats : -29.76379695489027
p_value 1.3467162476197306e-186
Null hypothesis is rejected
```

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

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

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

<Figure size 500x500 with 0 Axes>



Tenure V/s Customer churn

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

Out[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

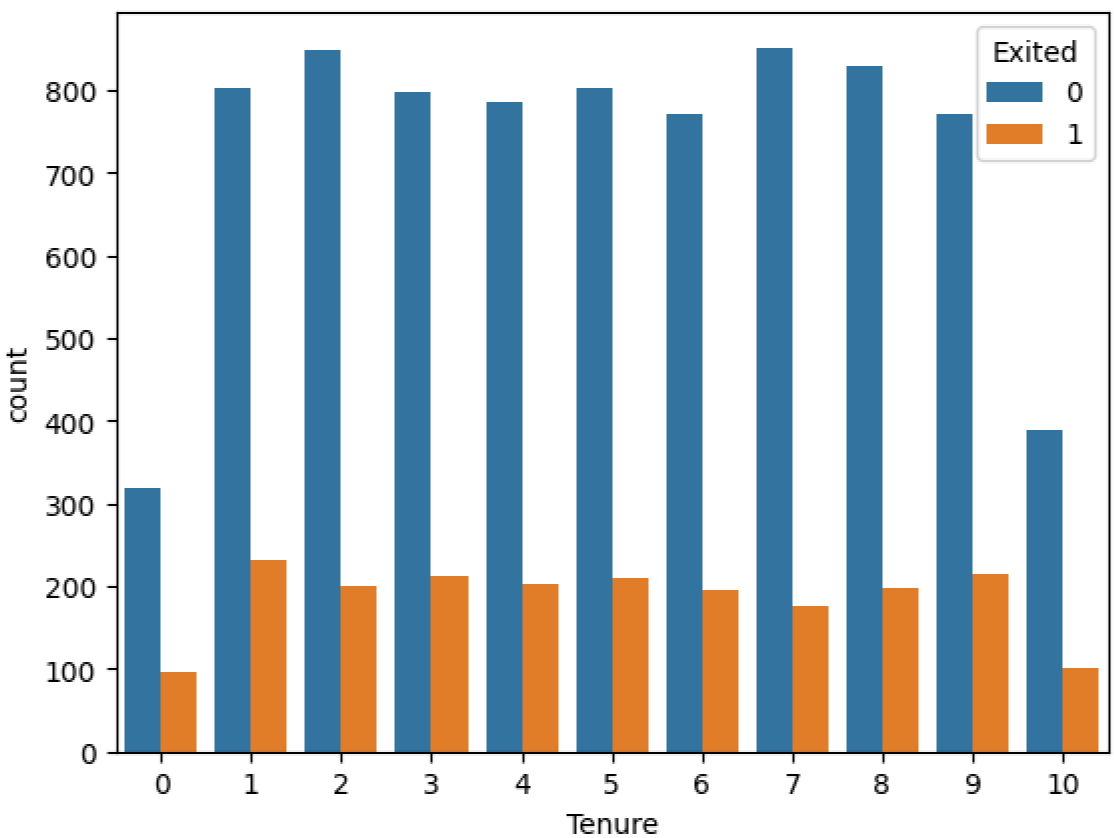
10000 rows × 2 columns

In [93]:

```
warnings.simplefilter(action='ignore', category=FutureWarning)

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

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



H0: Customer churn is independent of tenure

Ha: Customer churn is dependent of tenure

In [49]:
```
t_stats, p_value = ttest_ind(data[data['Exited'] == 0]['Tenure'],data[data['
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
```

Balance vs Customer Churn

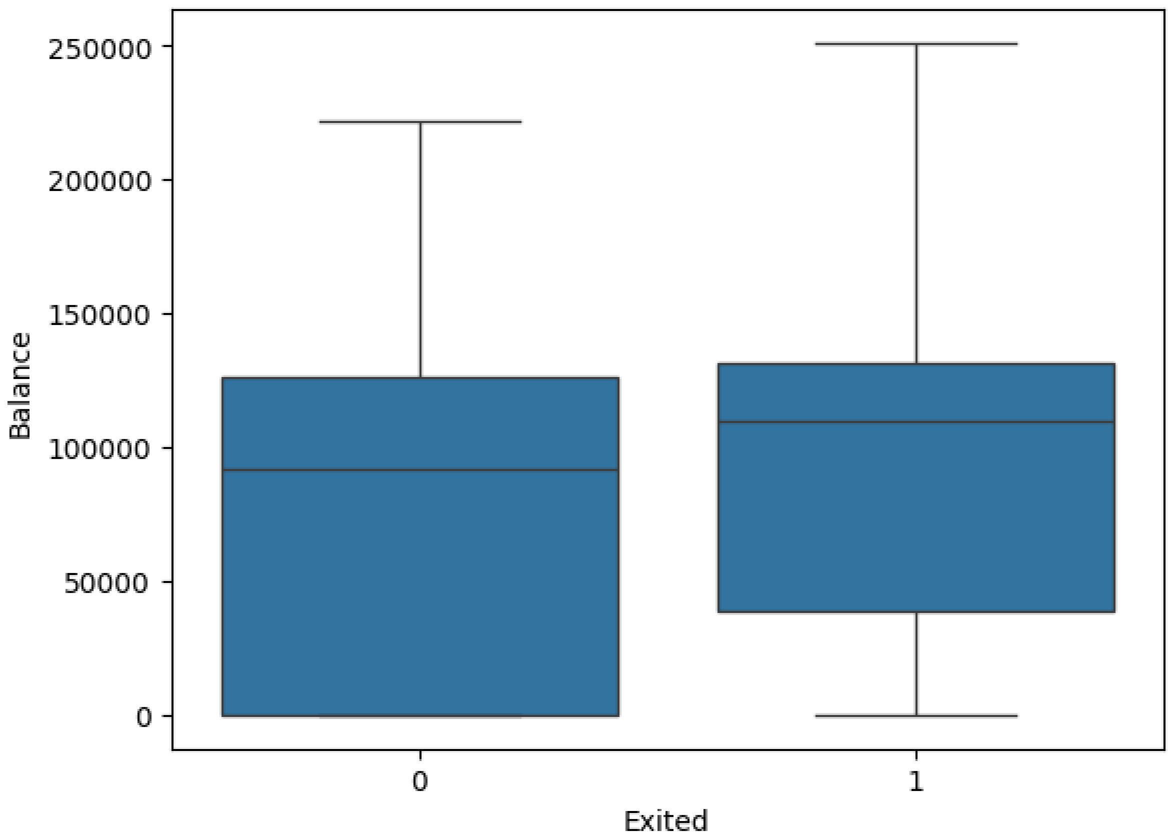
In [50]:
```
print(" max Balance of person who churned ", data[data['Exited'] == 1]['Bala
print(" min Balance of person who churned ",data[data['Exited'] == 1]['Balan
print(" max Balance of person who didn't churned ", data[data['Exited'] == 0
print(" min Balance of person who  didn't churned ",data[data['Exited'] == 0
```

```
 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
```

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

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

Out[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

In [52]:
```
t_stats, p_value = ttest_ind(data[data['Exited'] == 0]['Balance'],data[data[
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
```

Geogrpahy v/s customer churn

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

Out[53]:

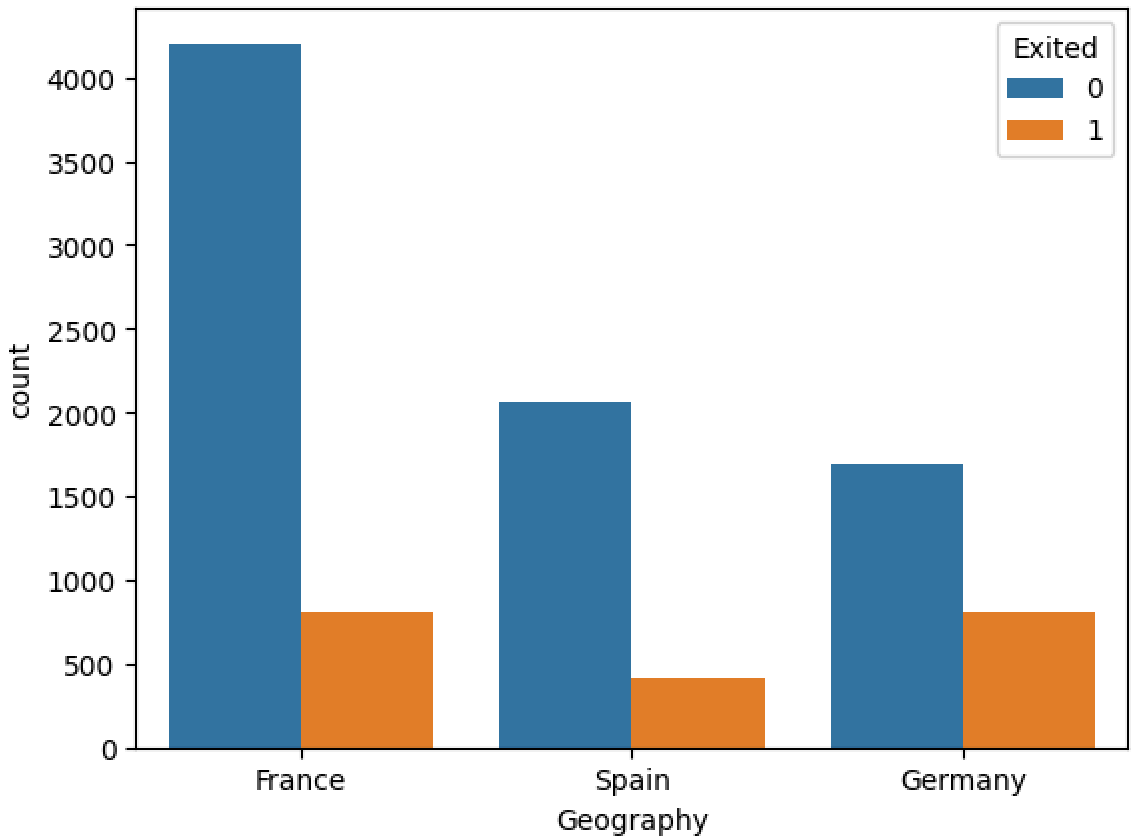
Geography	France	Germany	Spain
Exited			
0	4203	1695	2064
1	811	814	413

In [91]:

```
warnings.simplefilter(action='ignore', category=FutureWarning)

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

Out[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

In [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=5.24573610
9572763e-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
```

Impact assessement of different features on Customer churn

Gender and Customer Churn

In [56]:

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

Out[56]:

Gender	Female	Male
Exited		
0	3404	4558
1	1139	899

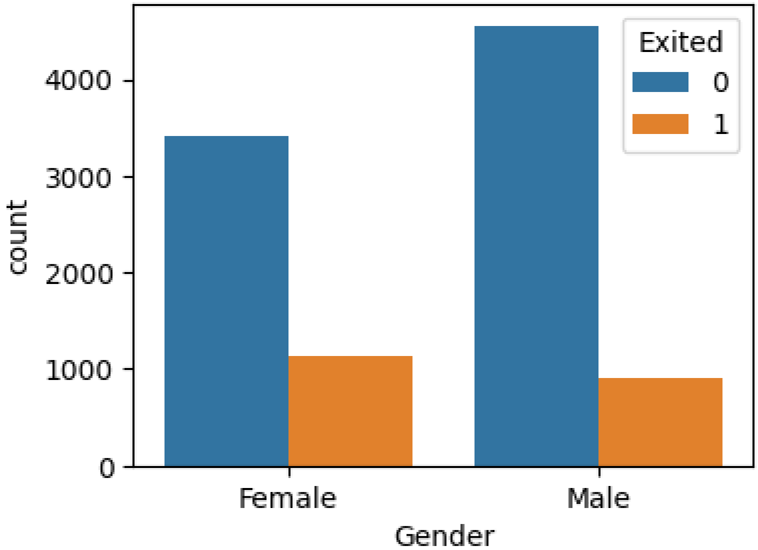
In [90]:

```
warnings.simplefilter(action='ignore', category=FutureWarning)

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

Out[90]:

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



H0: Gender and Customer churn are independent

Ha : Gender and Customer churn are dependent

In [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=2.9253677
618642e-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
```

Impact of Credit Card on Churn rate

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

Out[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

In [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=2.9253677
618642e-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
```

Analayze Area for service improvement

In [61]:

```
pd.crosstab(columns = [data['Complain'],data['Satisfaction Score']],index =
```

Out[61]:

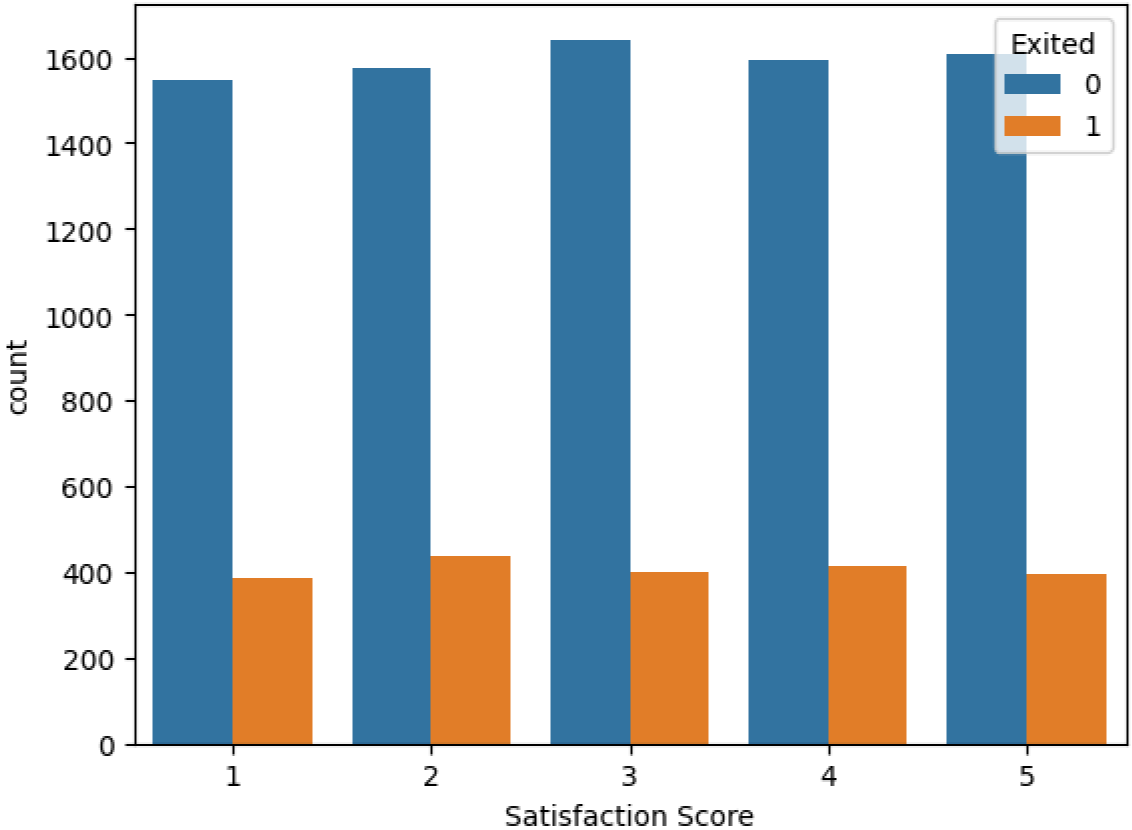
Complain	0					1				
Satisfaction Score	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
Exited										
0	1544	1574	1636	1594	1604	1	1	5	0	3
1	1	2	0	1	0	386	437	401	413	397

In [89]:

```
warnings.simplefilter(action='ignore', category=FutureWarning)

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

Out[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

Strategies for customer retenion strategies

In [63]:
```
data_banking_behaviour = data.loc[data['Exited'] ==1,['CustomerId','Tenure',
data_banking_behaviour
```

Out[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 × 5 columns

In [64]:

```
data_banking_behaviour['Spent'] = data_banking_behaviour['EstimatedSalary']*
data_banking_behaviour
```

Out[64]:

	CustomerId	Tenure	NumOfProducts	EstimatedSalary	Balance	
0	15634602	2	1	101348.88	0.00	2026
2	15619304	8	3	113931.57	159660.80	7517
5	15574012	8	2	149756.71	113755.78	10842
7	15656148	4	4	119346.88	115046.74	3623
16	15737452	1	1	5097.67	132602.88	-1275
...	...	...	...	...	...	
9981	15672754	3	1	53445.17	152039.70	82
9982	15768163	7	1	115146.40	137145.12	6688
9991	15769959	4	1	69384.71	88381.21	1891
9997	15584532	7	1	42085.58	0.00	2945
9998	15682355	3	2	92888.52	75075.31	2035

2038 rows × 6 columns

In [65]:

```
data_banking_behaviour[data_banking_behaviour['Balance'] < 0 ]
```

Out[65]:

CustomerId	Tenure	NumOfProducts	EstimatedSalary	Balance	Spent

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

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

Out[66]:

	CustomerId	Tenure	NumOfProducts	EstimatedSalary	Balance	S
16	15737452	1	1	5097.67	132602.88	-1275
35	15794171	0	1	27822.99	134264.04	-1342
54	15569590	1	1	40014.76	98495.72	-584
70	15703793	2	4	28373.86	133745.44	-769
127	15782688	0	1	46824.08	148507.24	-1485
...	...	...	...	...	...	
9863	15726179	5	2	3497.43	131433.33	-1139
9882	15785490	3	1	16281.68	105229.72	-563
9920	15673020	3	1	738.88	204510.94	-2022
9924	15578865	5	1	6985.34	107959.39	-730
9947	15732202	1	2	73124.53	83503.11	-103

350 rows × 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

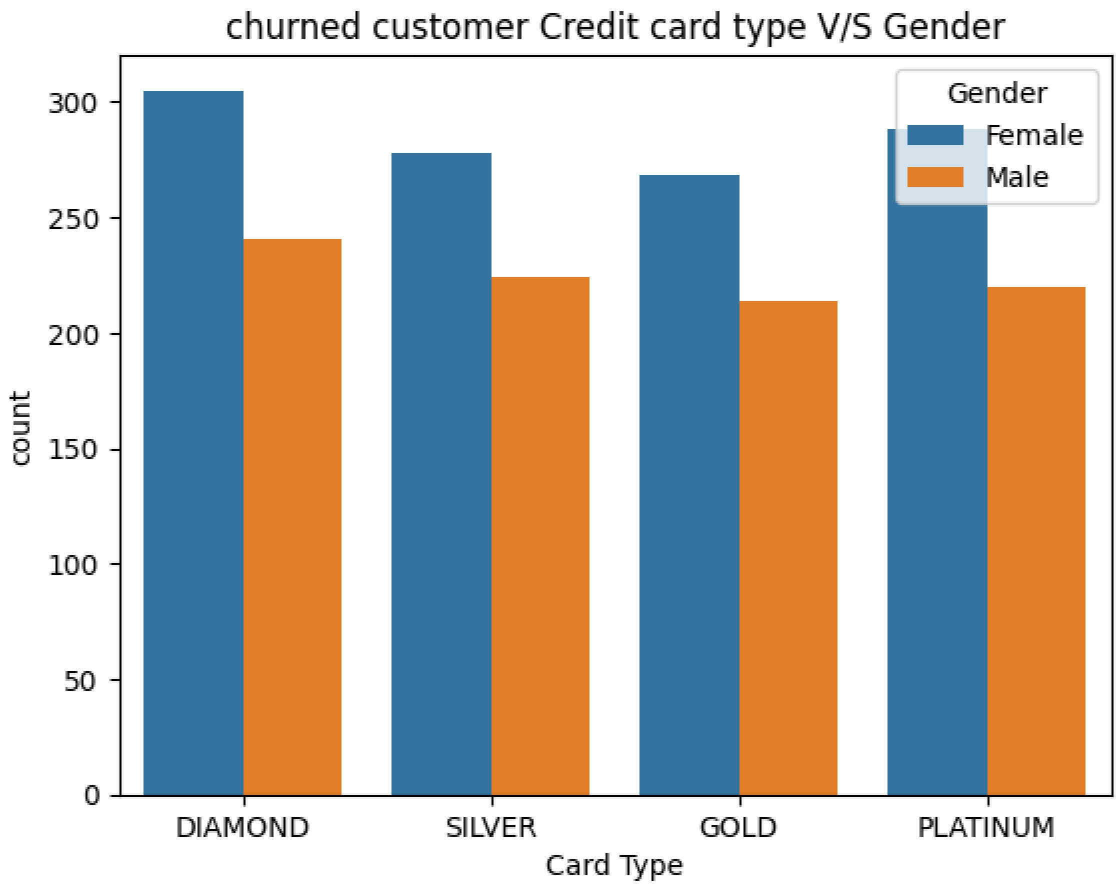
In [ ]:

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

In [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")
```

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



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

Out[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 × 4 columns

In [69]:
```
pd.crosstab(index = data[data['Exited'] == 1]['Card Type'],columns = data[da
```

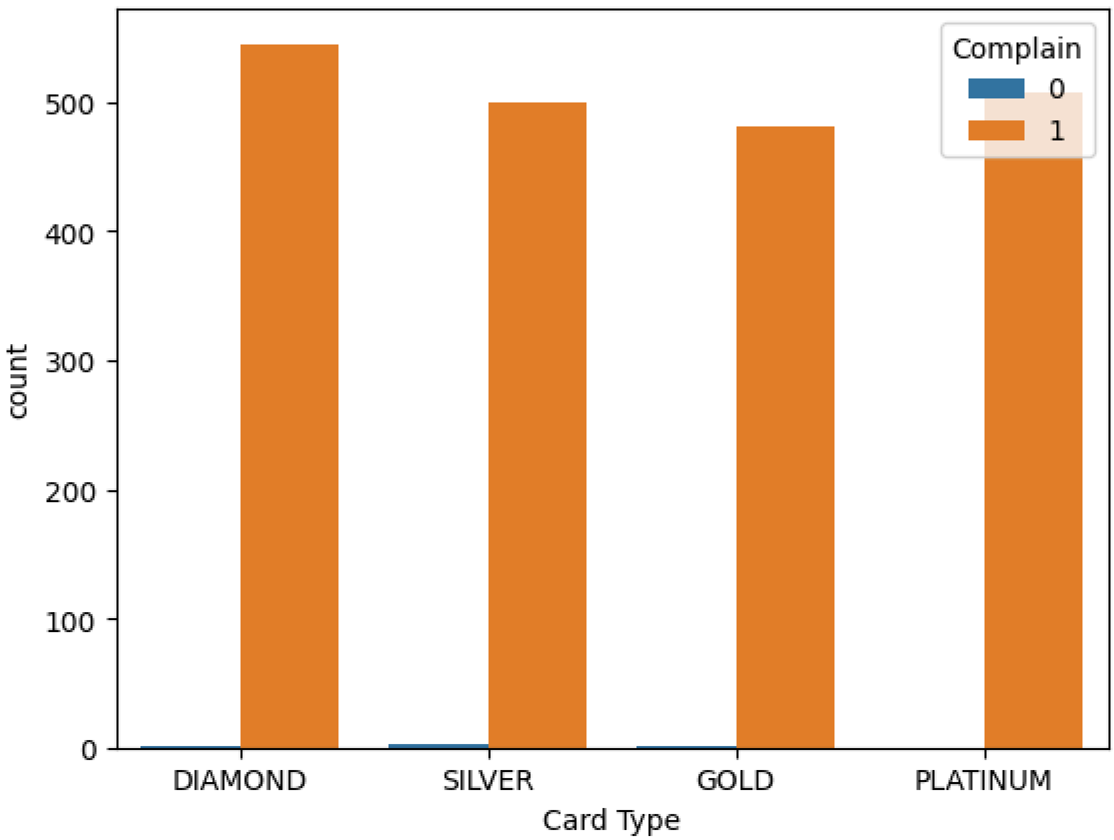
Out[69]:

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

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

sns.countplot(x = data[data['Exited'] == 1]['Card Type'],hue = data[data['Ex
```

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

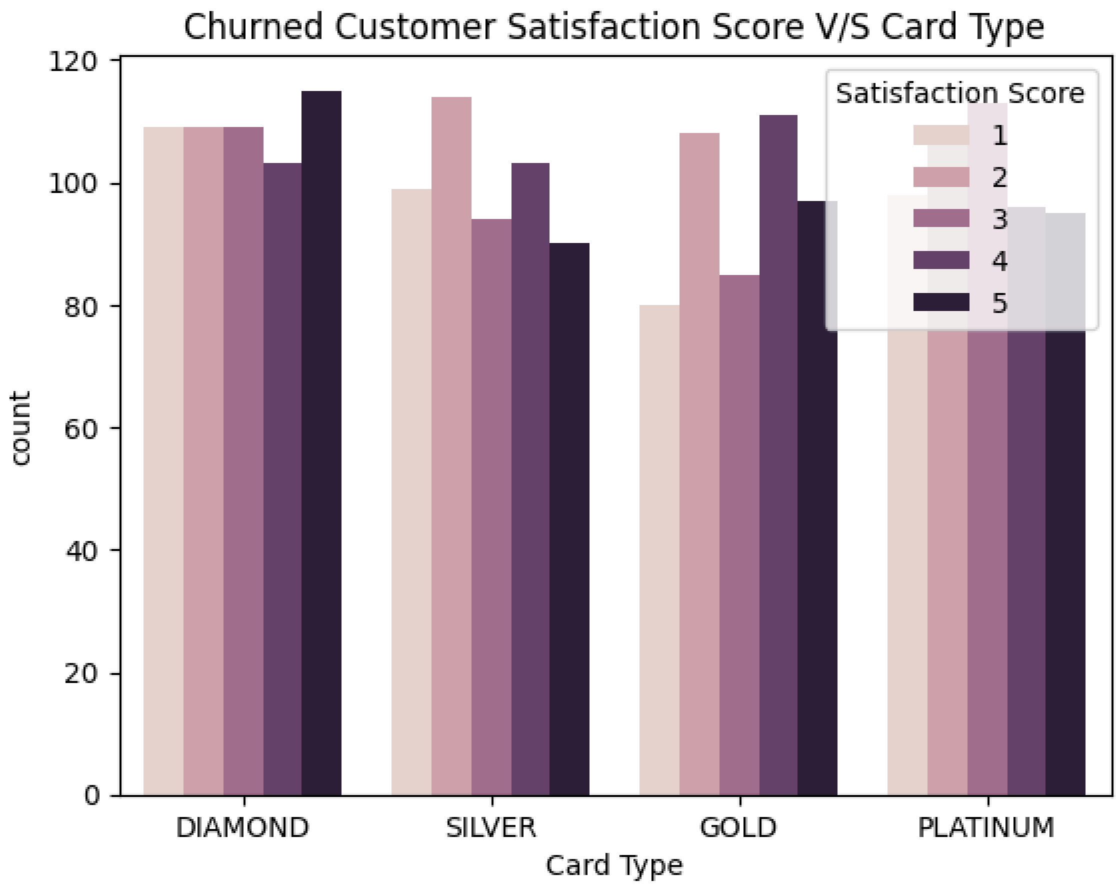


satisfaction score for Customer who churned out and have complained to banking services were visualize as below shown

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

sns.countplot(x = data[(data['Exited'] ==1) & (data['Complain']==1)]['Card T
plt.title('Churned Customer Satisfaction Score V/S Card Type')
```

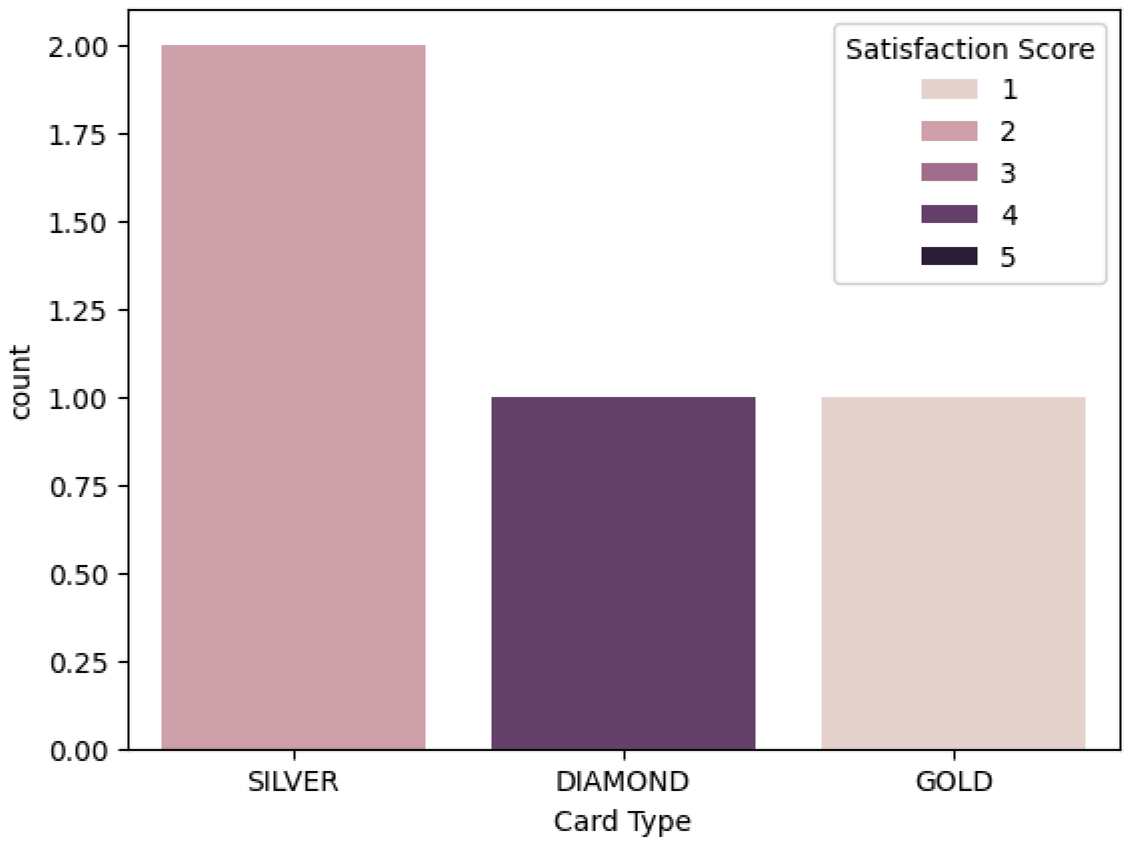
Out[86]: Text(0.5, 1.0, 'Churned Customer Satisfaction Score V/S Card Type')



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

sns.countplot(x = data[(data['Exited'] ==1) & (data['Complain']==0)]['Card T
```

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

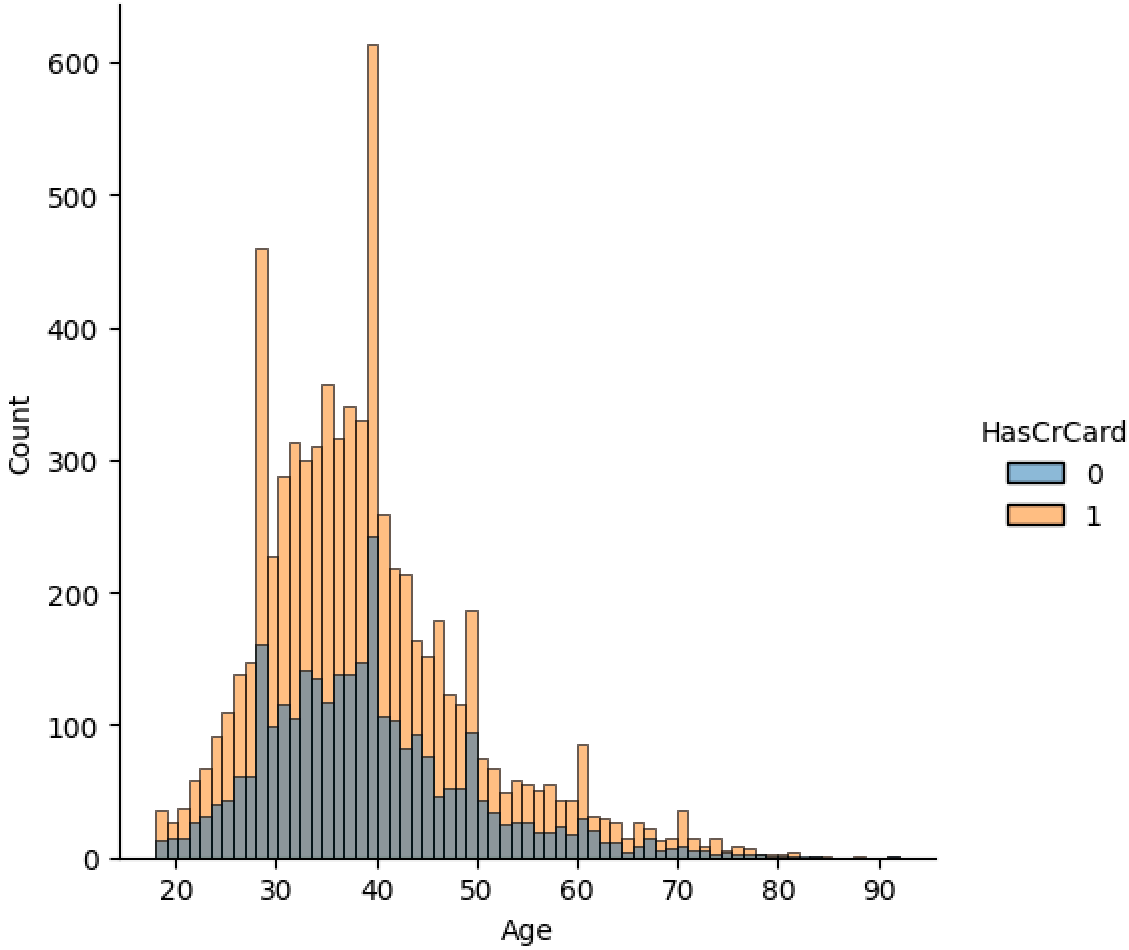


Checking Credit card Age wise

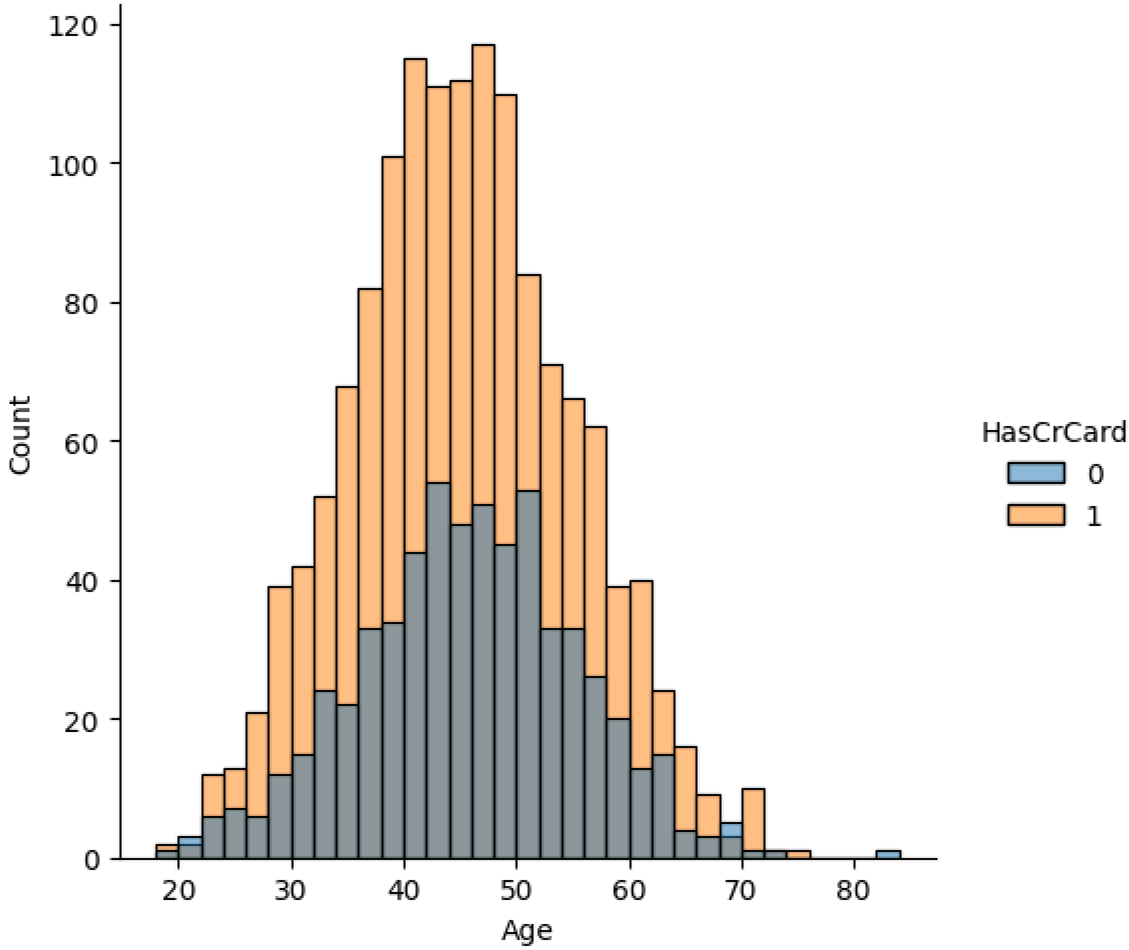
In [84]:
```
warnings.simplefilter(action='ignore', category=FutureWarning)

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")
```

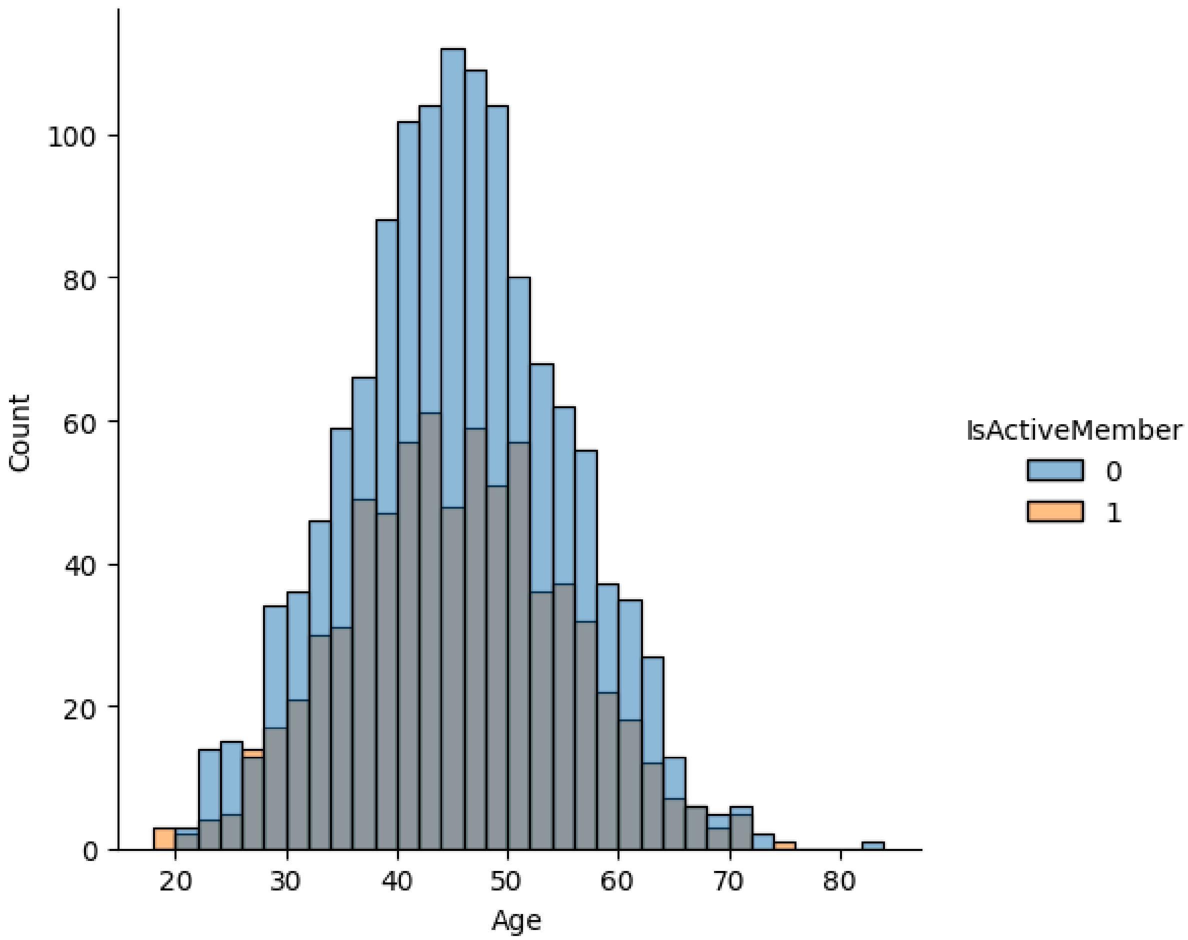
Out[84]:
```
<seaborn.axisgrid.FacetGrid at 0x7869f1cafd60>
<Figure size 500x500 with 0 Axes>
```



<Figure size 500x500 with 0 Axes>



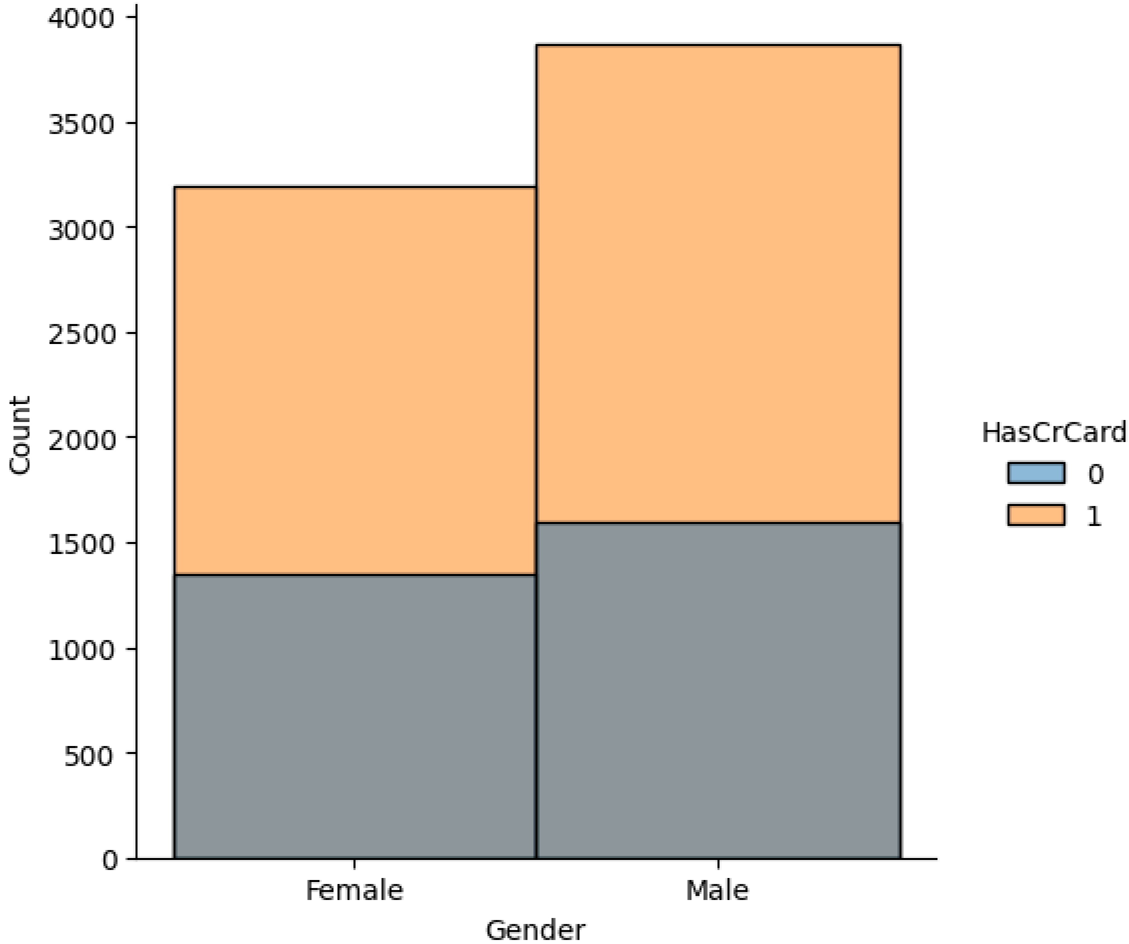
<Figure size 500x500 with 0 Axes>



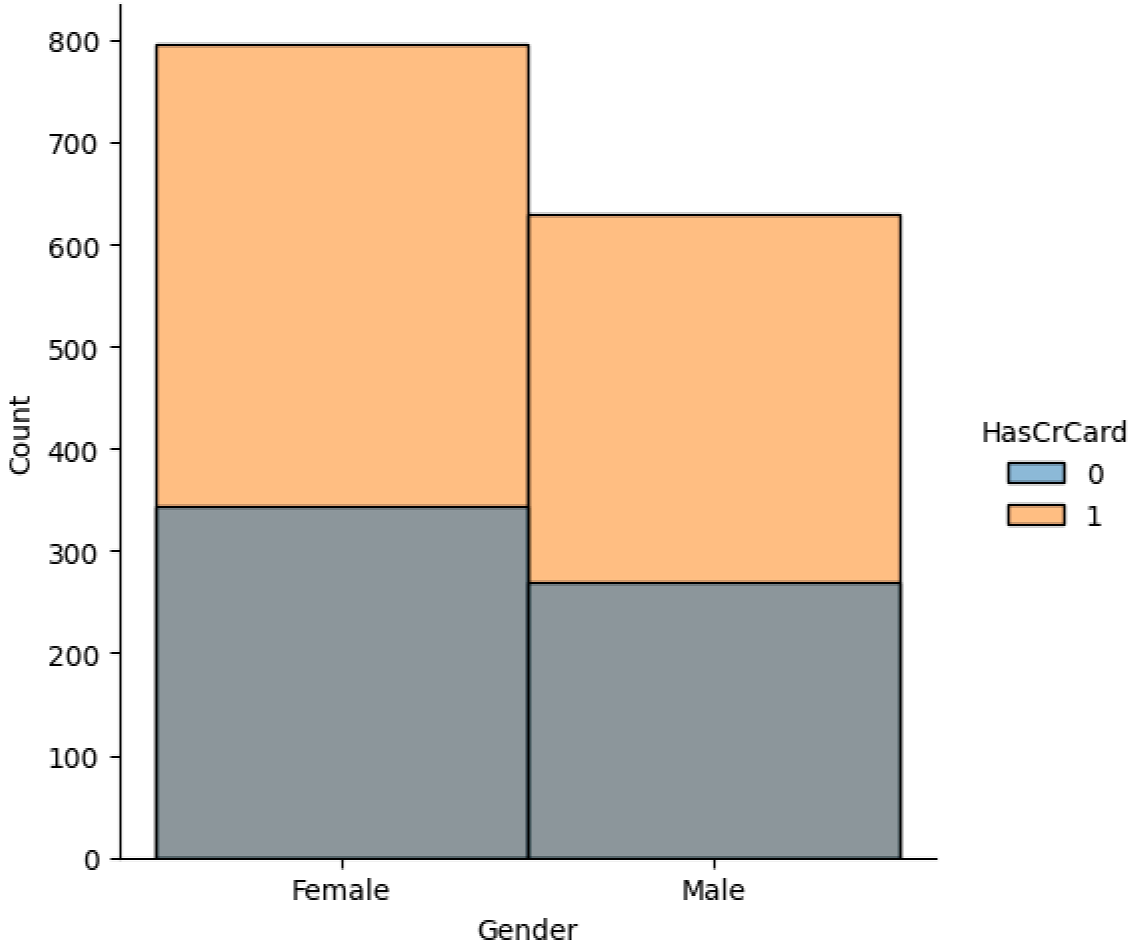
In [83]:
```
warnings.simplefilter(action='ignore', category=FutureWarning)

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"
```

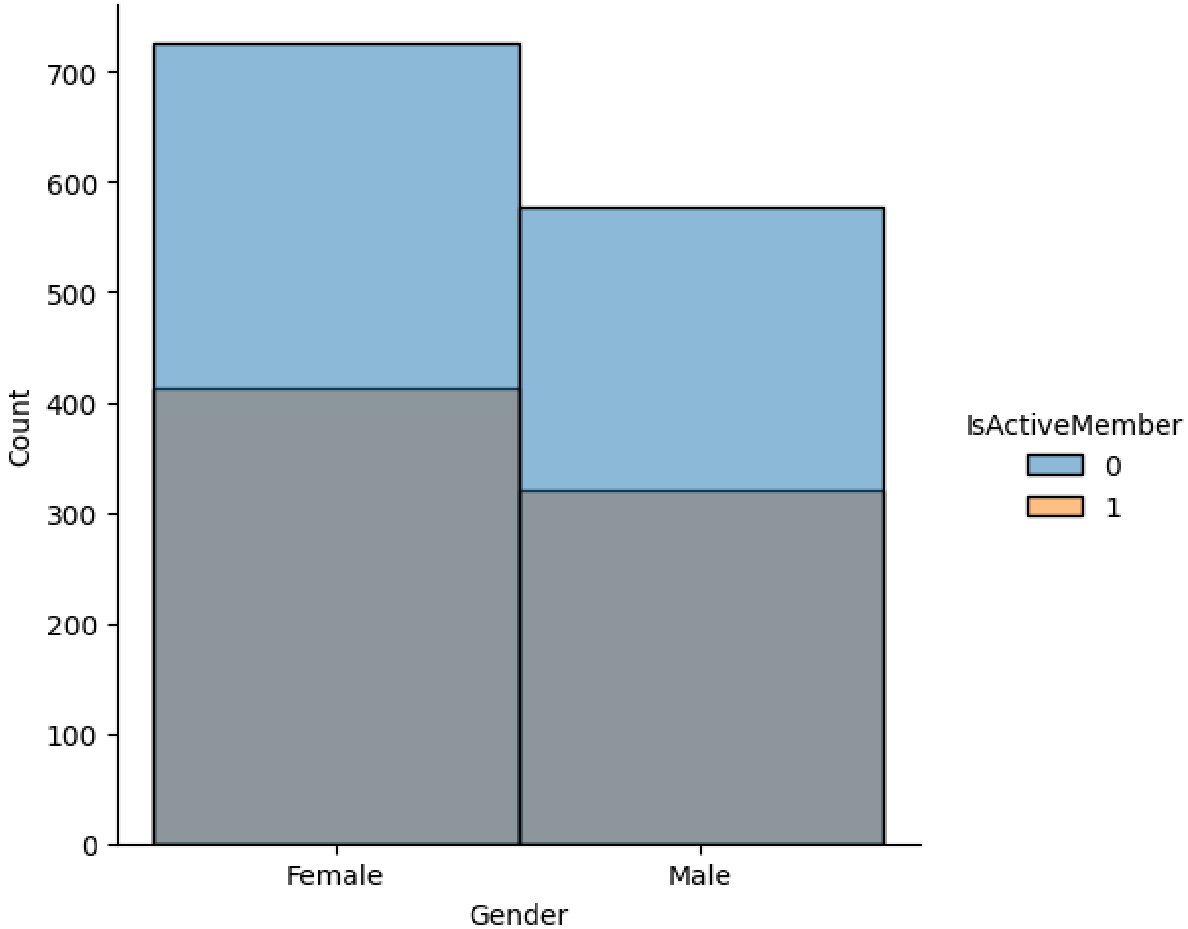
Out[83]:
```
<seaborn.axisgrid.FacetGrid at 0x7869ed5027d0>
<Figure size 500x500 with 0 Axes>
```



<Figure size 500x500 with 0 Axes>




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



Descriptive analysis

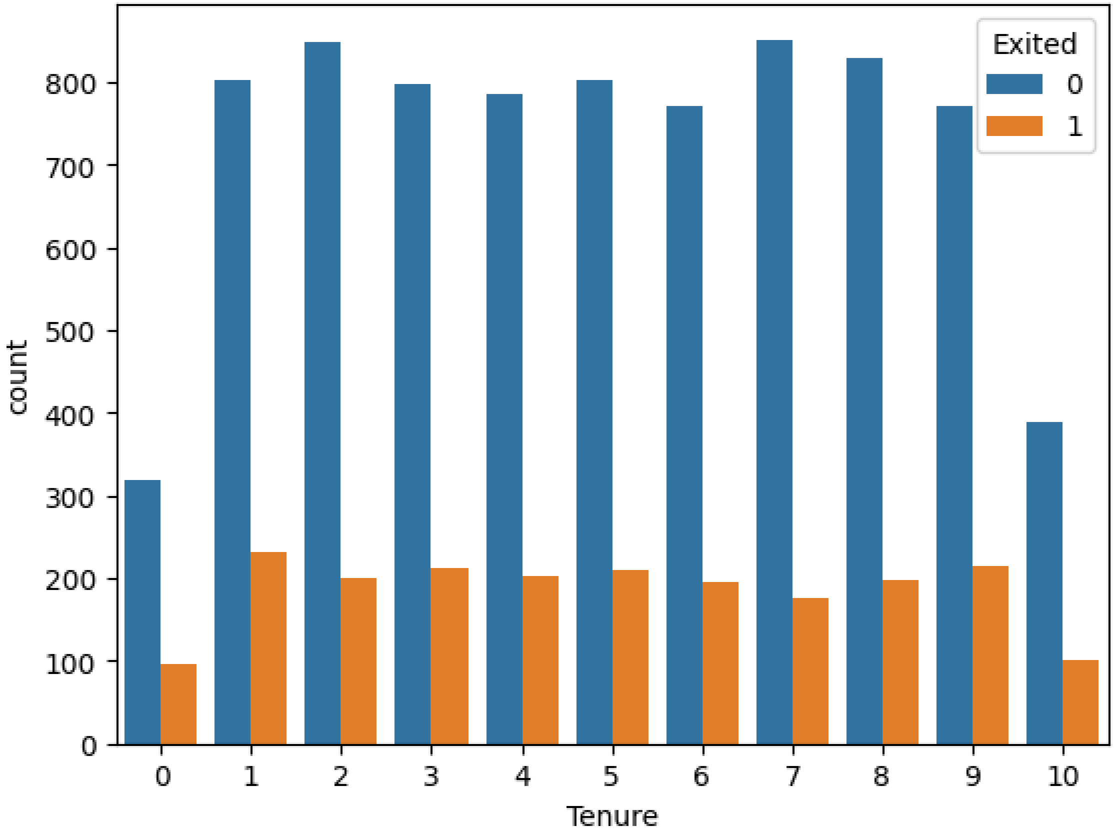
Churn rate

for different type of tenures

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

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

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



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

Out[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

In [77]:
```
churn_data = pd.crosstab(columns = data['Tenure'],index= data['Exited'],norm
churn_data
```

Out[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	0.8
1	0.230024	0.224155	0.191794	0.2111	0.205258	0.206522	0.202689	0.1

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

Out[78]:

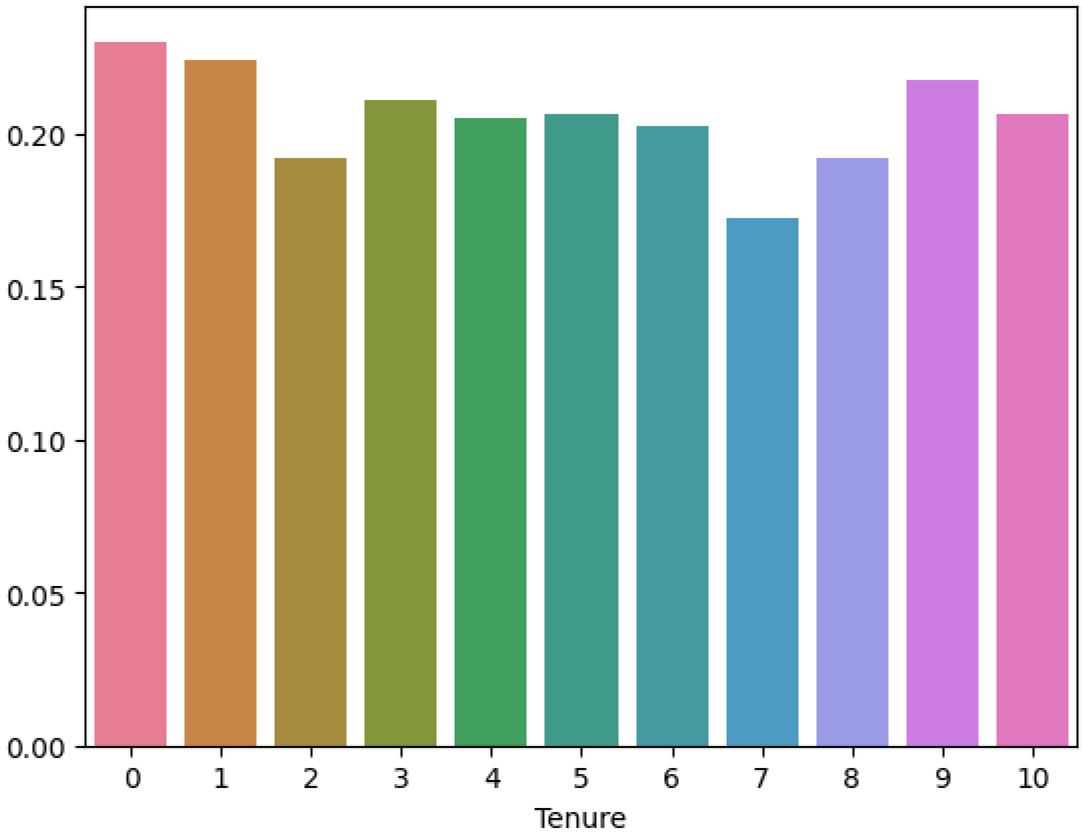
	Tenure	Exited	0	1	2	3	4	5
0	1	0.230024	0.224155	0.191794	0.2111	0.205258	0.206522	0.202

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

In [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

In [ ]:

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

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

In [ ]:

```

```