```
# Recalling Python Library and downloading data form google drive
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sbn
```

```
[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, 78.0MB/s]
```

```
# loading data to pandas library
df=pd.read_csv("Bank-Records.csv")
df
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Complain	Satisfaction Score
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1	1	2
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0	1	3
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1	1	3
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0	0	5
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0	0	5
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9995	9996	15606229	Obijiaku	771	France	Male	39	5	0.00	2	1	0	96270.64	0	0	1
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	101699.77	0	0	5
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	42085.58	1	1	3
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	2	1	0	92888.52	1	1	2
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1	0	38190.78	0	0	3

10000 rows × 18 columns

⇒ Given data has 10K rows and 18 columns

```
# Checking Null values & data type
df.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
```

```
# Basic Exploring Data Analysis
df['CustomerId'].nunique()
```

```
10000
```

```
df[['CustomerId','Exited']]
```

	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

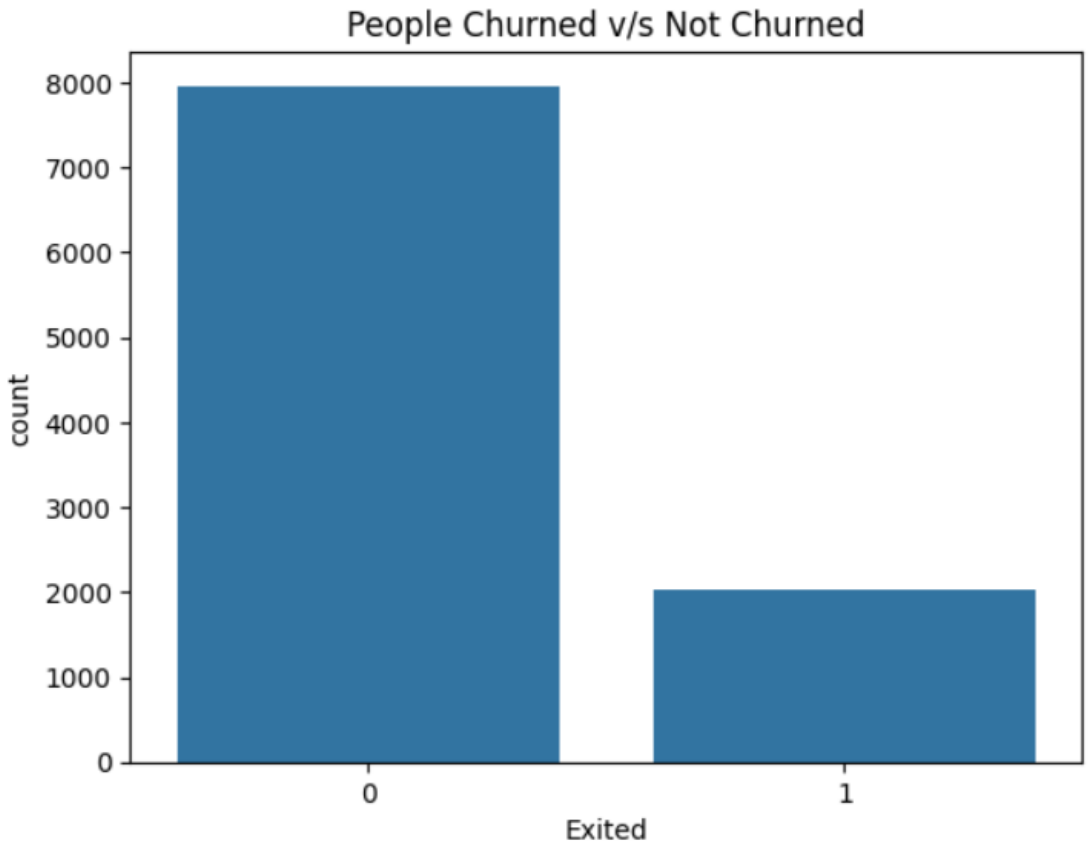
⇒ There are 10K unique CustomerID

```
# People Churned v/s Not Churned
df['Exited'].value_counts()
```

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

```
# Visualization of People Churned v/s Not churned
sbn.countplot(x=df['Exited'])
plt.title('People Churned v/s Not Churned')
```

```
Text(0.5, 1.0, 'People Churned v/s Not Churned')
```



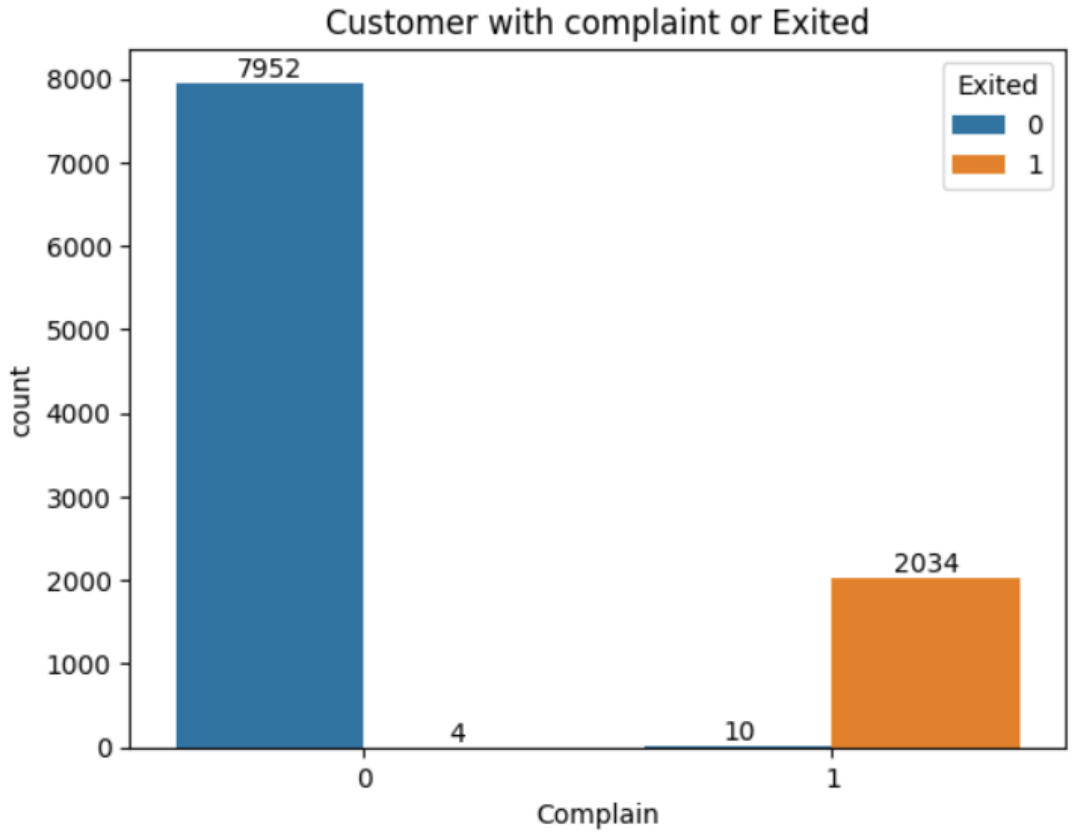
⇒ Out of 10K customers, 2038 exited from bank and 7962 are still account holder.

```
# Customers with Complaint or Exited
pd.crosstab(columns = df['Complain'],index = df['Exited'])
```

Complain	0	1
Exited		
0	7952	10
1	4	2034

```
# Visulaization to Customers with Complaint or Exited
ax1 = sbn.countplot(x=df['Complain'],hue=df['Exited'])
for container in ax1.containers:
    ax1.bar_label(container)
plt.title('Customer with complaint or Exited')
```

```
Text(0.5, 1.0, 'Customer with complaint or Exited')
```



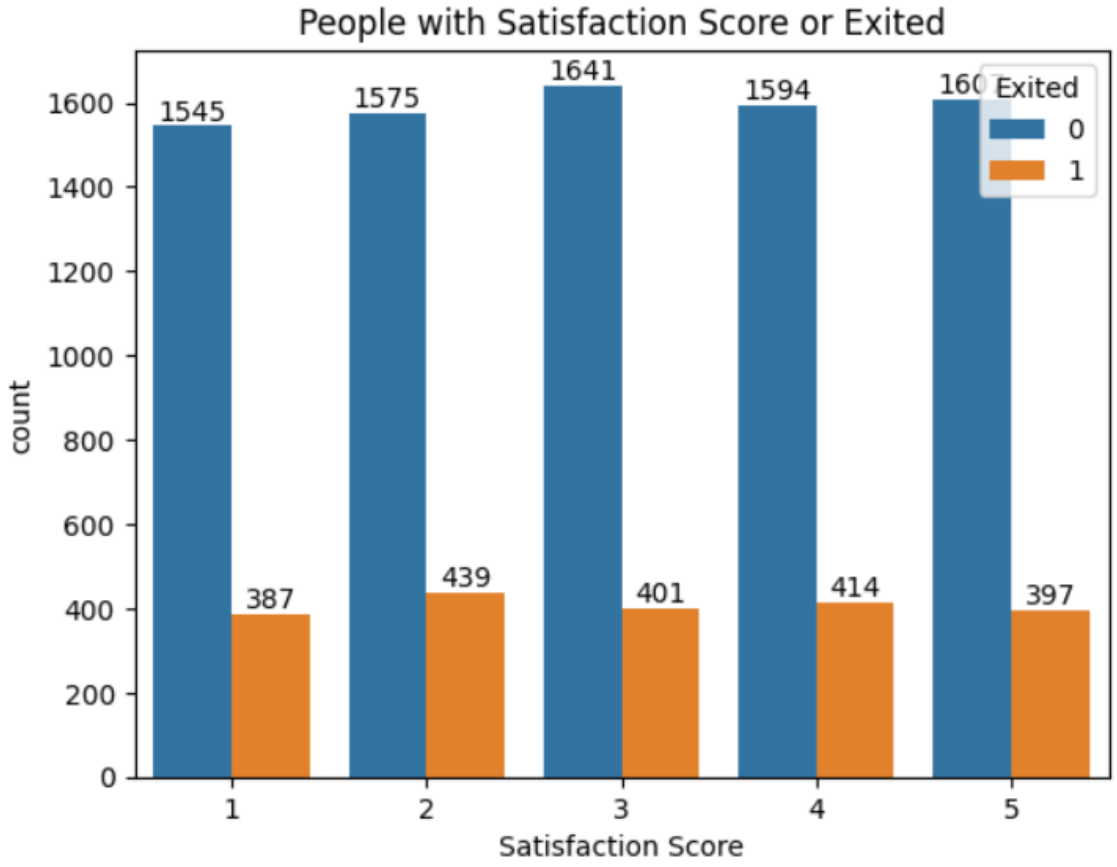
⇒ Out of 2038 customers Churned, there were 2034 customers who complained.

```
[7] # People with Satisfaction Score or Exited
    pd.crosstab(columns = df['Satisfaction Score'],index = df['Exited'])
```

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

```
# Visulaization to People with Satisfaction Score or Exited
ax2= sbn.countplot(x=df['Satisfaction Score'],hue=df['Exited'])
for container in ax2.containers:
    ax2.bar_label(container)
plt.title('People with Satisfaction Score or Exited')
```

```
Text(0.5, 1.0, 'People with Satisfaction Score or Exited')
```



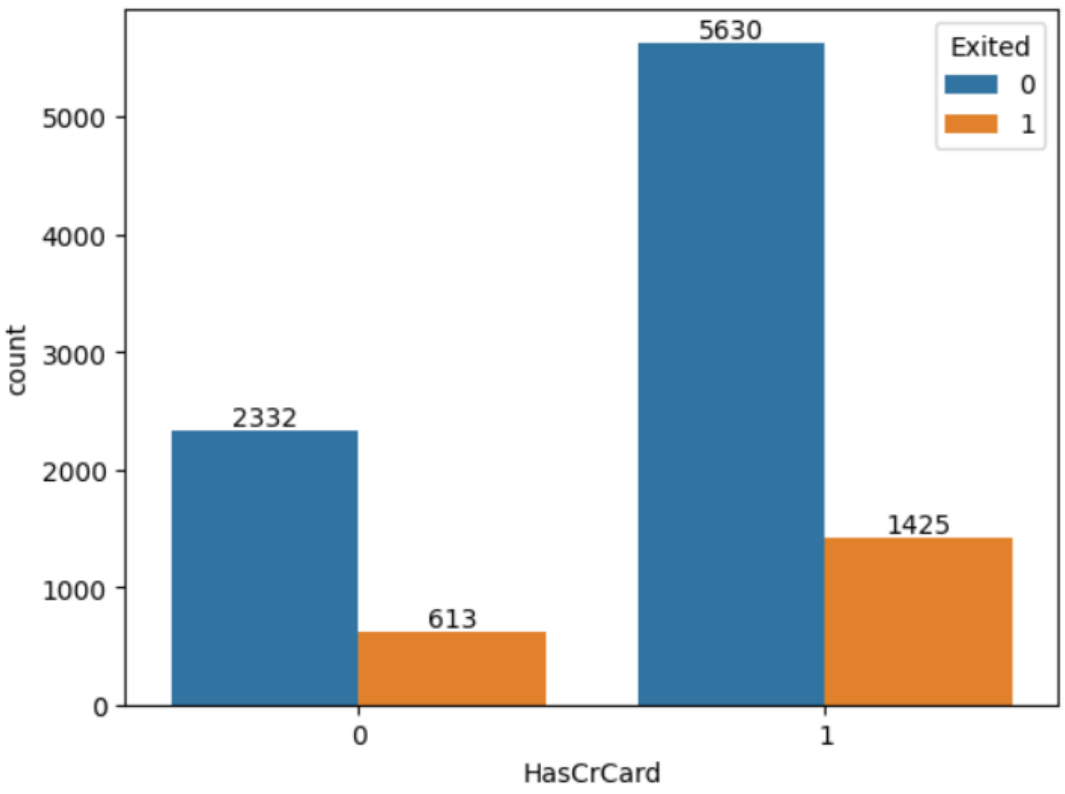
⇒ There is no insight for customer exit from Satisfaction Score.

0s

```
# People with Credit Card v/s  Churn
pd.crosstab(columns = df['HasCrCard'],index = df['Exited'])
```

HasCrCard	0	1
Exited		
0	2332	5630
1	613	1425

```
# Visualization of People with Credit Card v/s Churn
ax3= sbn.countplot(x=df['HasCrCard'],hue=df['Exited'])
for container in ax3.containers:
    ax3.bar_label(container)
```

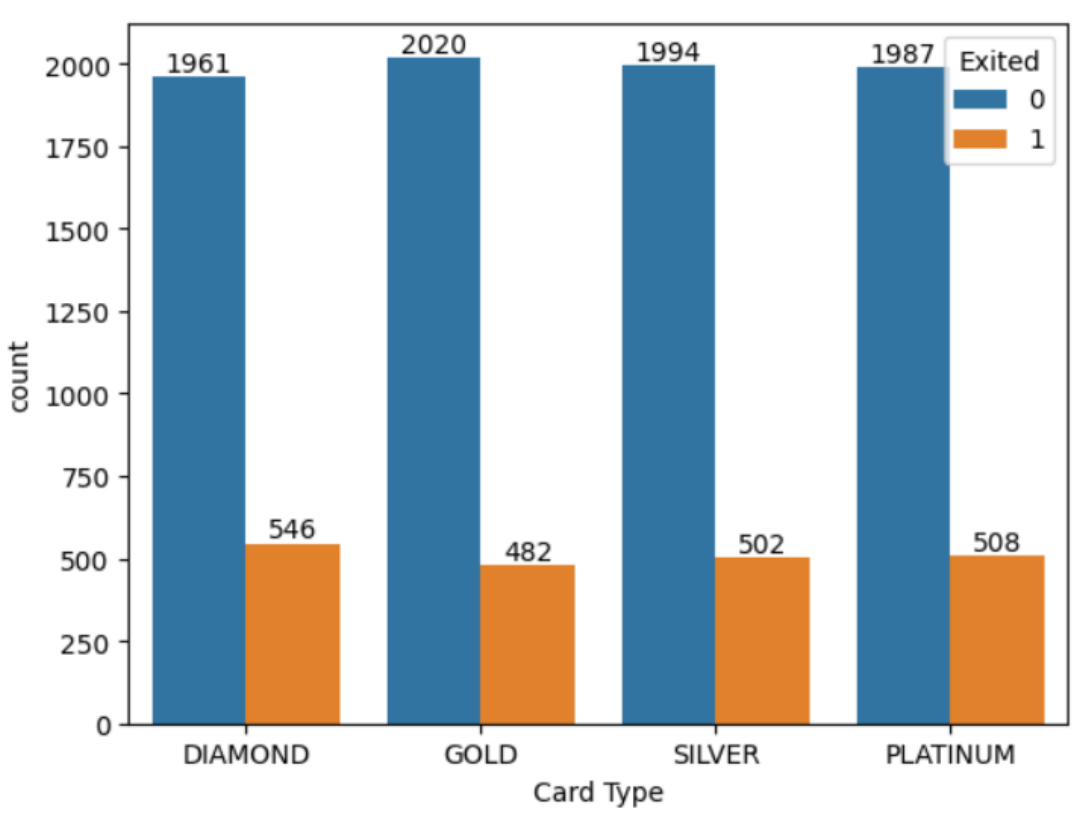


⇒ People having Credit Card has exited more than that having no Credit Card.

```
# People with different Card Type v/s Exit
pd.crosstab(columns = df['Card Type'],index = df['Exited'])
```

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

```
[5] # Visualization of People with different Card Type v/s Exit
    ax4= sbn.countplot(x=df['Card Type'],hue=df['Exited'])
    for container in ax4.containers:
        ax4.bar_label(container)
```



⇒ Customer Churn is independent on type of card which they had.

```
# People Churned v/s CreditScore
df[df['Exited']==1]['CreditScore'].max()
```

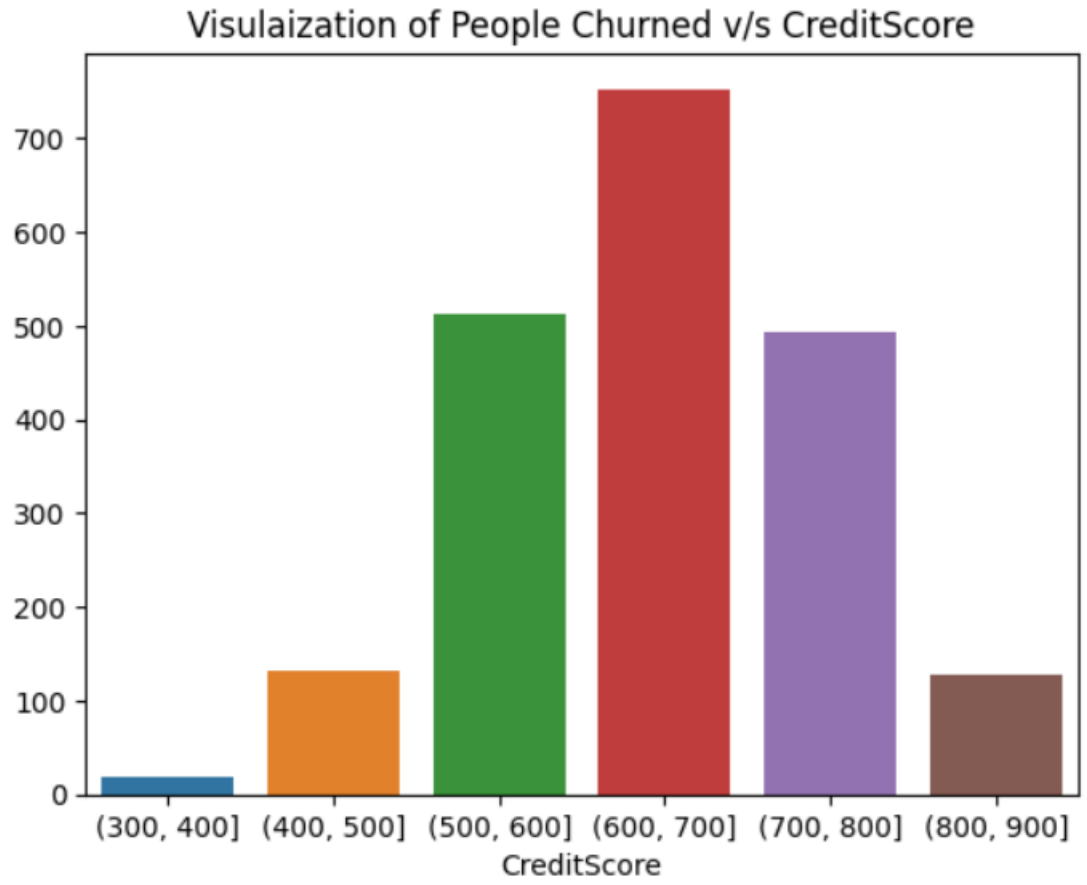
```
850
```

```
[12] bins=[300,400,500,600,700,800,900]
     credit_bin=pd.cut(df[df['Exited']==1]['CreditScore'],bins)
     pd.crosstab(columns=credit_bin, index=df['Exited'])
```

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

```
# Visulaization of People Churned v/s CreditScore
sbn.barplot(pd.crosstab(columns=credit_bin, index=df['Exited']))
plt.title('Visulaization of People Churned v/s CreditScore')
```

```
Text(0.5, 1.0, 'Visulaization of People Churned v/s CreditScore')
```



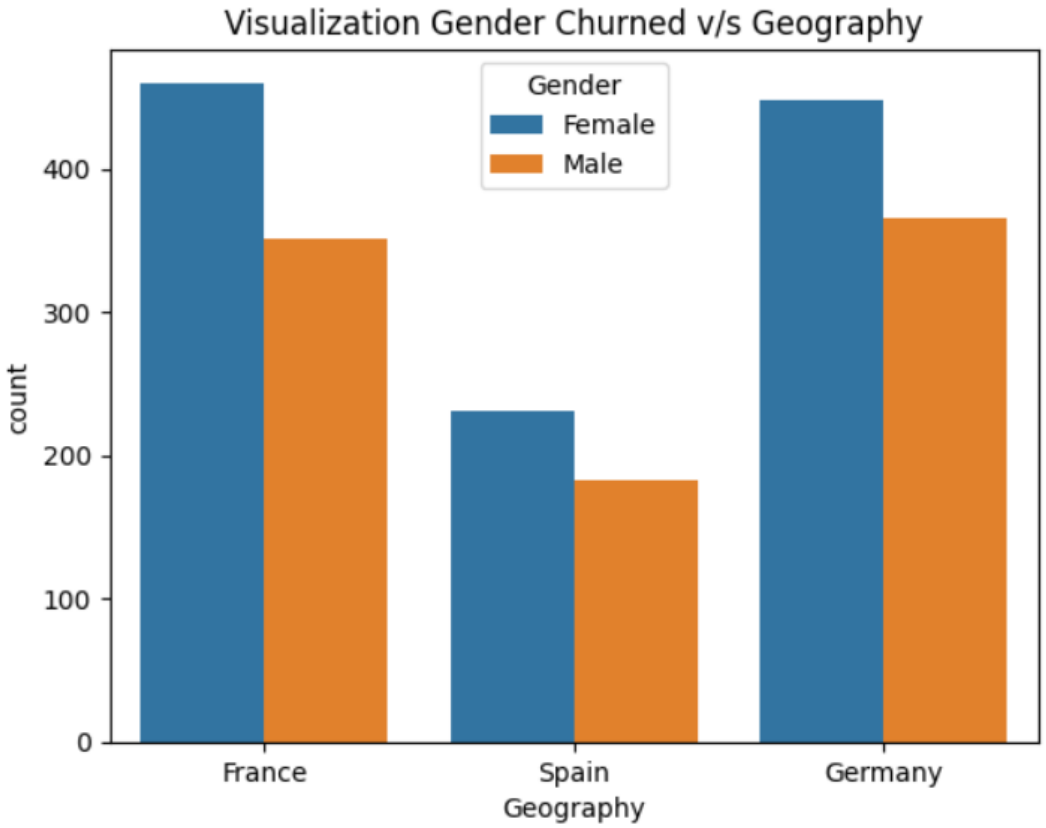
⇒ People with credit score in (500-600, 600-700, 700-800) left the banking service most.

```
# Gender Churned v/s Geography
pd.crosstab(columns = [df['Geography'], df['Gender']], index = df['Exited'])
```

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

```
# Visualization Gender Churned v/s Geography
sbn.countplot(x=df[df['Exited']==1]['Geography'],hue=df[df['Exited']==1]['Gender'])
plt.title('Visualization Gender Churned v/s Geography')
```

```
Text(0.5, 1.0, 'Visualization Gender Churned v/s Geography')
```

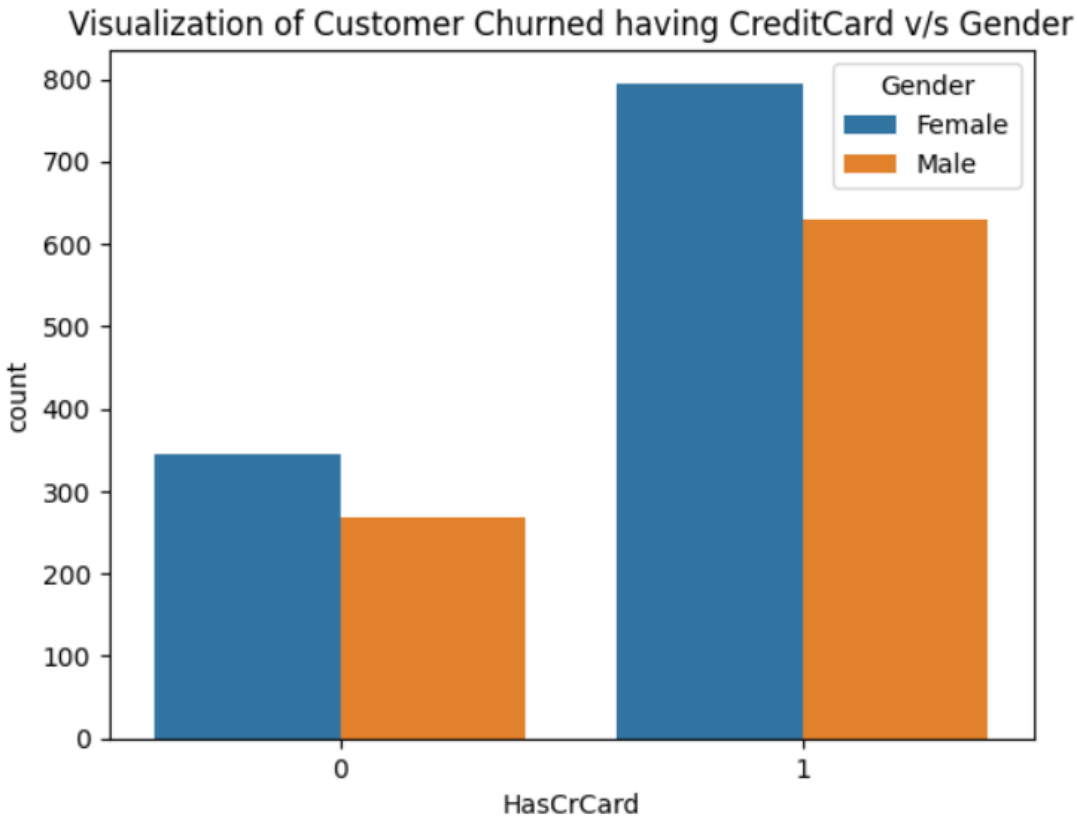



```
[28] # Customer Churned having CreditCard v/s Gender
     pd.crosstab(columns = [df['HasCrCard'], df['Gender']], index = df['Exited'])
```

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

```
# Visualization of Customer Churned having CreditCard v/s Gende
sbn.countplot(x=df[df['Exited']==1]['HasCrCard'],hue=df[df['Exited']==1]['Gender'])
plt.title('Visualization of Customer Churned having CreditCard v/s Gender')
```

```
Text(0.5, 1.0, 'Visualization of Customer Churned having CreditCard v/s Gender')
```

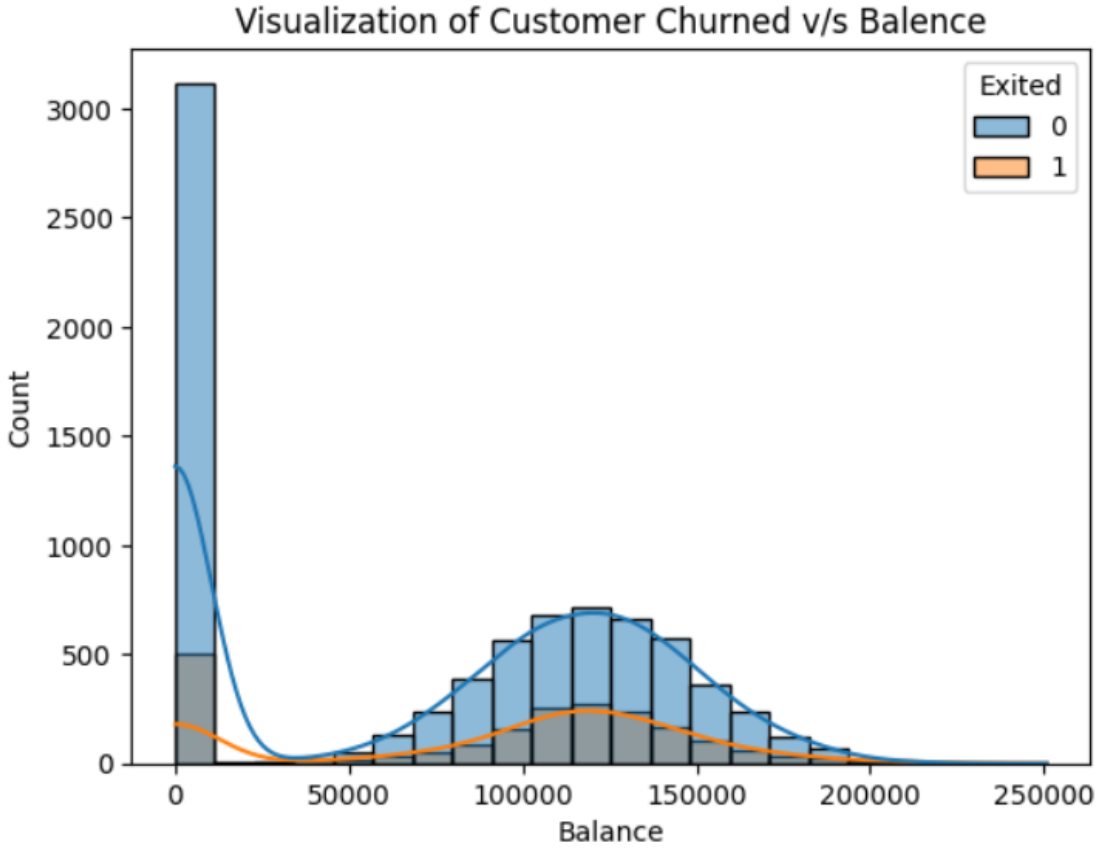


```
pd.crosstab(columns = df['Tenure'],index = df['Exited'])
```

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

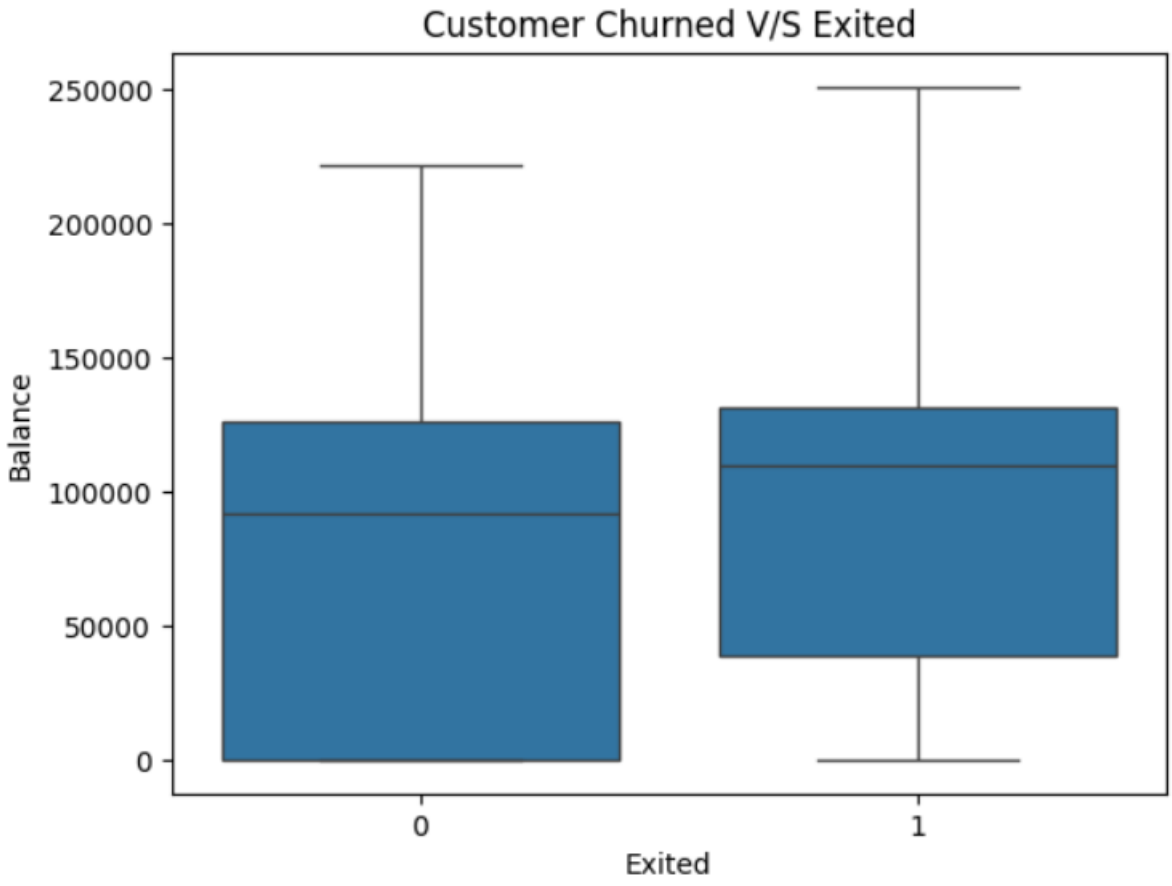
```
# Visualization of Customer Churned v/s Balence
sbn.histplot(x= df['Balance'],hue =df['Exited'],kde =True)
plt.title('Visualization of Customer Churned v/s Balence')
```

```
Text(0.5, 1.0, 'Visualization of Customer Churned v/s Balence')
```



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

```
Text(0.5, 1.0, 'Customer Churned V/S Exited')
```

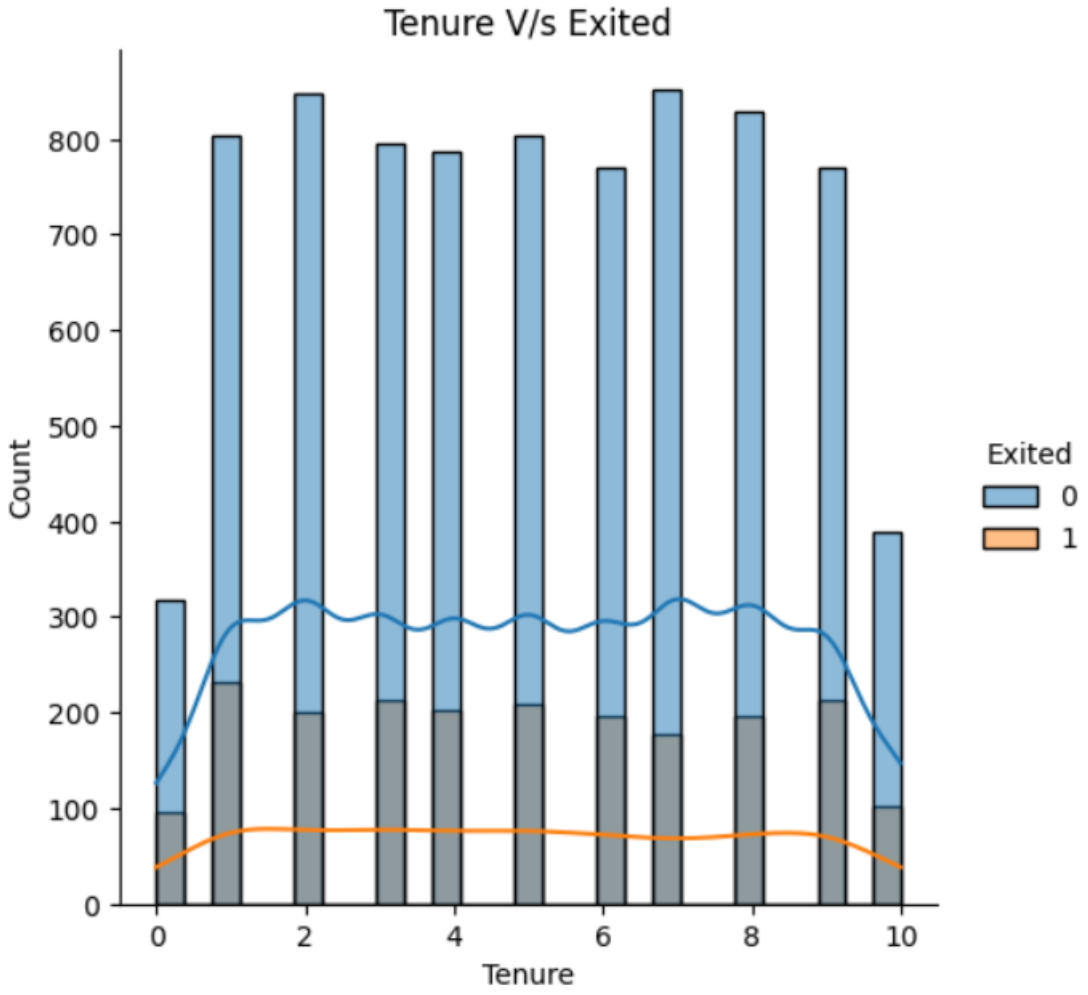



```
# Tenure V/S Exited
pd.crosstab(columns = df['Tenure'],index = df['Exited'])
```

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

```
# Visualization of Tenure V/S Exited
sbn.displot(x = df['Tenure'],hue = df['Exited'],kde =True)
plt.title('Tenure V/s Exited')
```

```
Text(0.5, 1.0, 'Tenure V/s Exited')
```

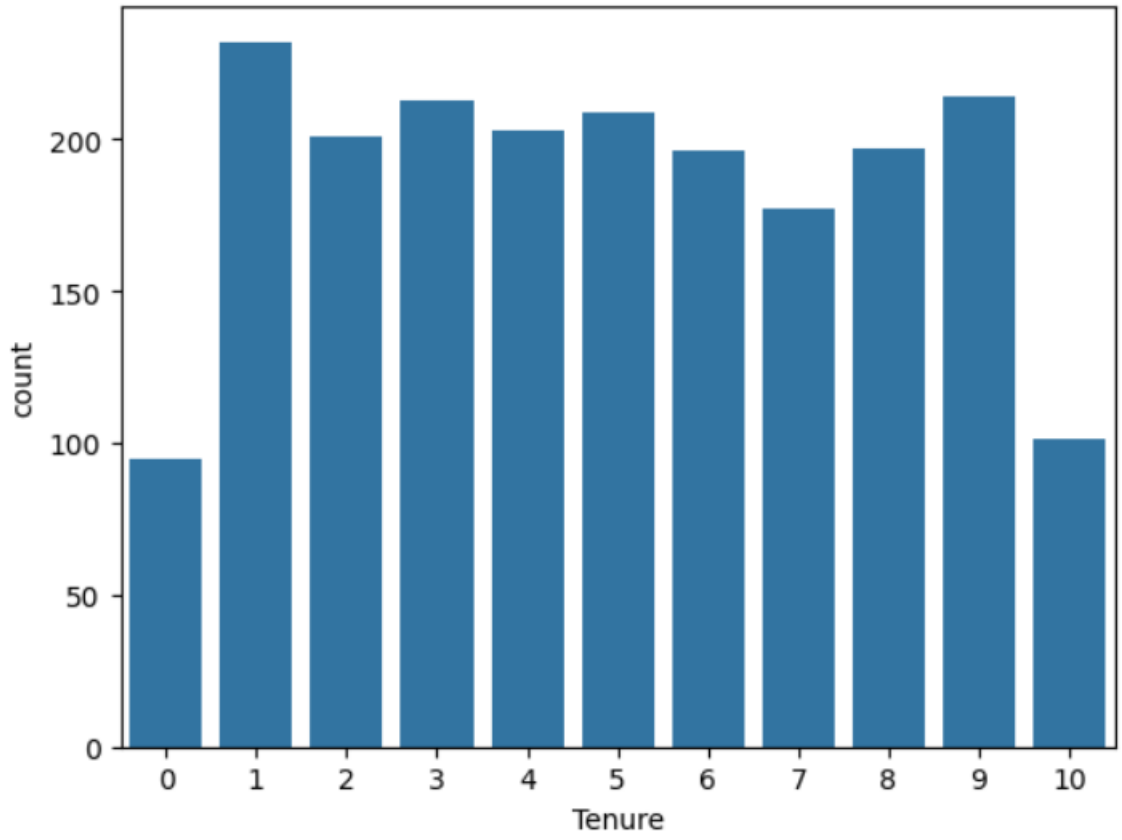



```
# Tenure V/S Exited
df[df['Exited']==1]['Tenure'].value_counts().reset_index().T
```

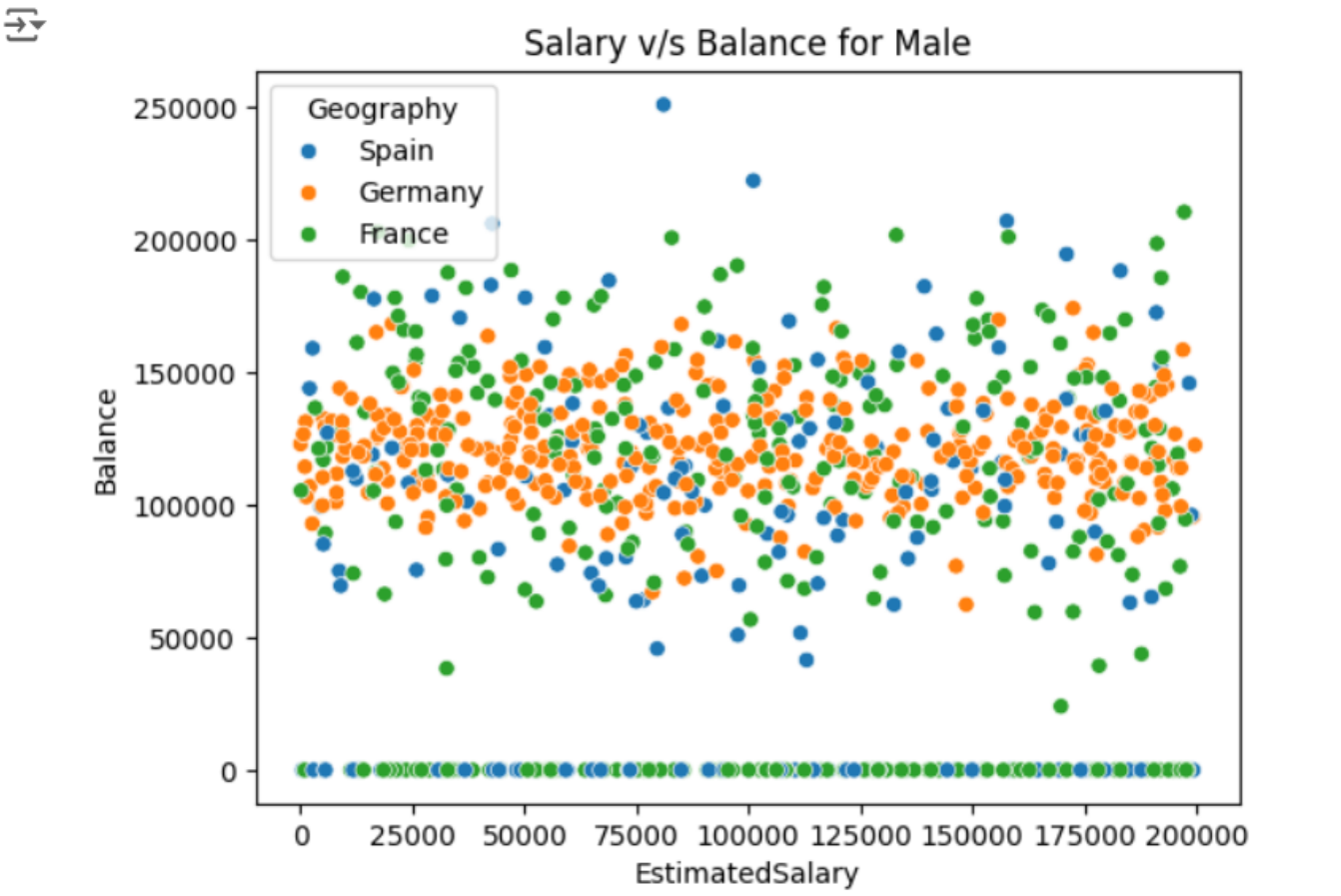
	0	1	2	3	4	5	6	7	8	9	10
Tenure	1	9	3	5	4	2	8	6	7	10	0
count	232	214	213	209	203	201	197	196	177	101	95

```python
# Visualization of count of Exited people v/s Tenure
sbn.countplot(x =df[df['Exited']==1]['Tenure'])
```

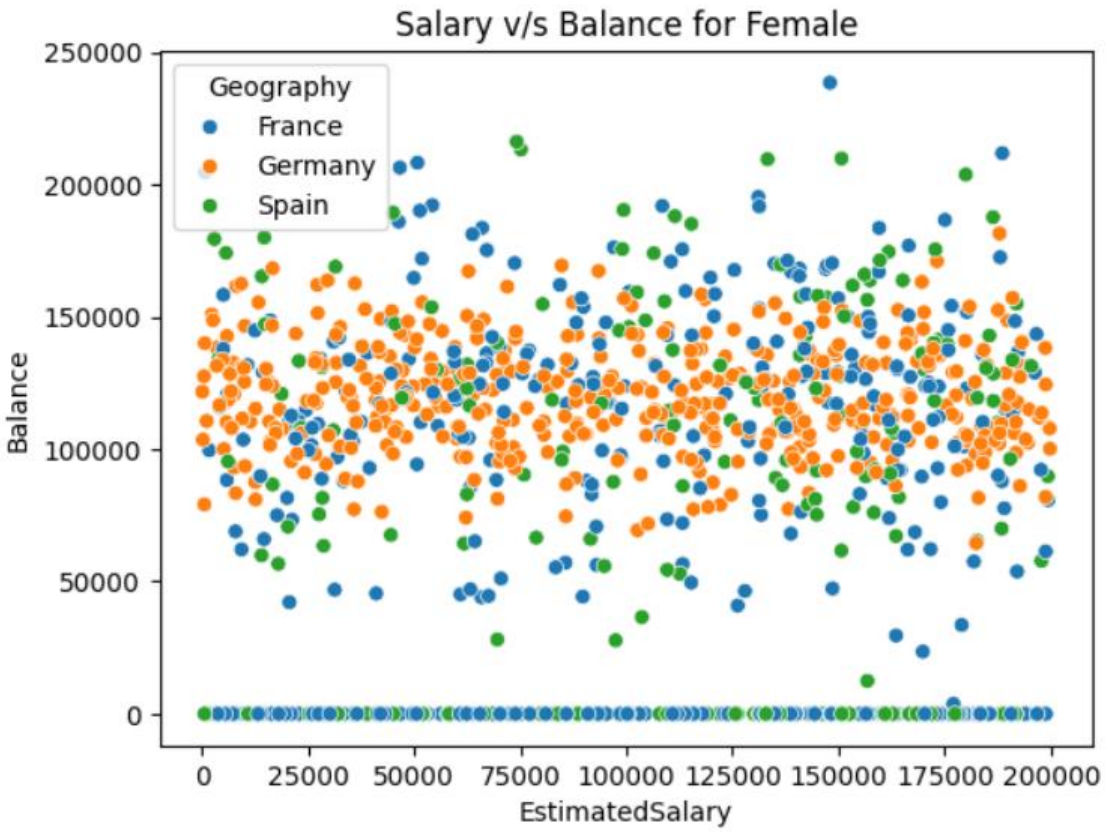
```
<Axes: xlabel='Tenure', ylabel='count'>
```



```
# Checking estimated salary v/s balance of people w.r.t to Geography for
# different genders who left the bank
ax = sbn.scatterplot(x="EstimatedSalary", y="Balance",
                     hue="Geography",
                     data=df[(df['Exited']==1) & (df['Gender'] =='Male')])
ax.set_title('Salary v/s Balance for Male')
plt.show()
```



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




```
from scipy.stats import f_oneway,kruskal,ttest_ind,chi2_contingency
```

```
[22] t_stats, p_value = ttest_ind(df[df['Exited'] == 0]['CreditScore'],df[df['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
```

```
[25] # Age vs Customer churn
     df[['Age','Exited']].T
```

	0	1	2	3	4	5	6	7	8	9	...	9990	9991	9992	9993	9994	9995	9996	9997	9998	9999
Age	42	41	42	39	43	44	50	29	44	27	...	33	53	36	28	29	39	35	36	42	28
Exited	1	0	1	0	0	1	0	1	0	0	...	0	1	0	0	0	0	0	1	1	0

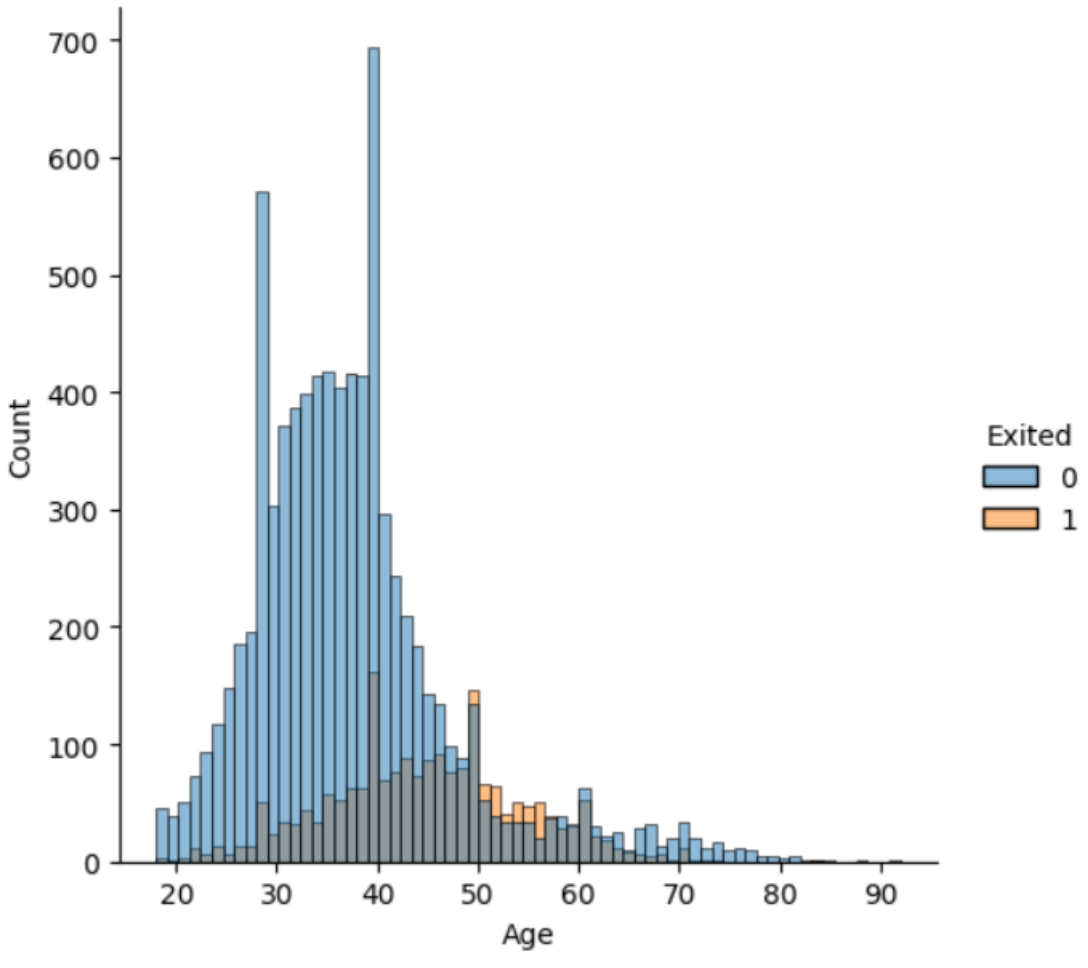
2 rows × 10000 columns

```
t_stats, p_value = ttest_ind(df[df['Exited'] == 0]['Age'],df[df['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
```

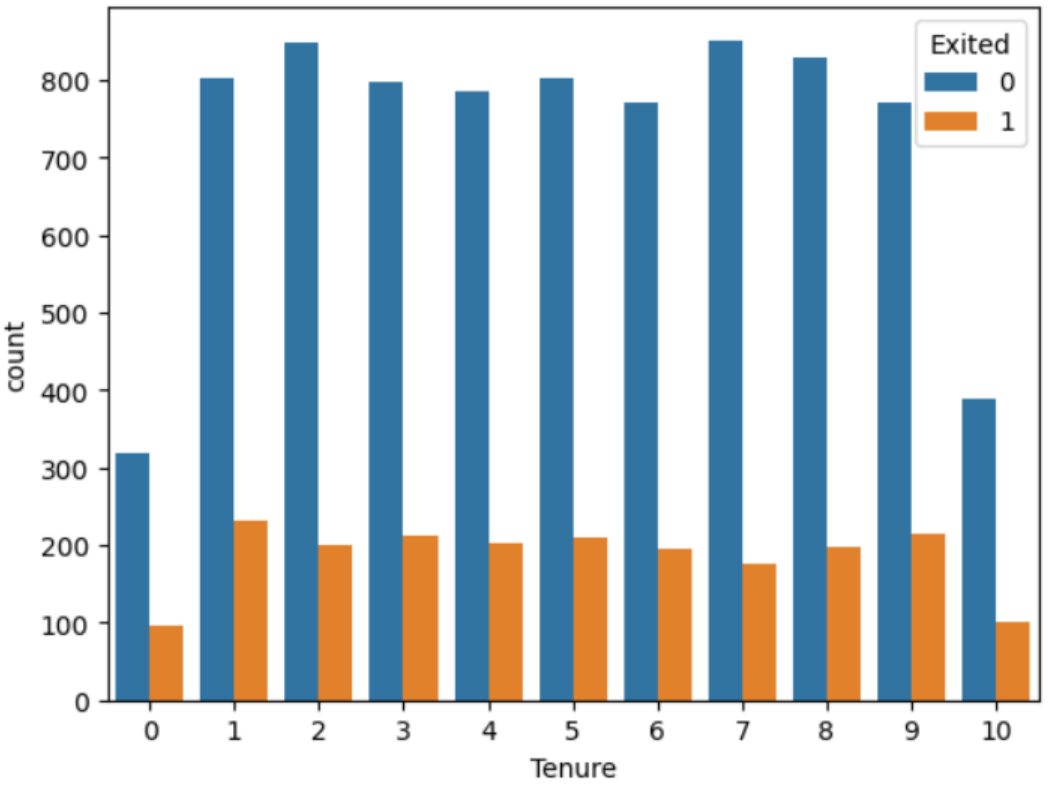
```
# Visualization of Age v/s Exited
plt.figure(figsize=(5, 5))
sbn.displot(data=df, x="Age", hue="Exited")
```

```
<seaborn.axisgrid.FacetGrid at 0x7b4f49fad150>
<Figure size 500x500 with 0 Axes>
```



```
# Tenure v/s Customer Churn
sbn.countplot(x = df['Tenure'],hue = df['Exited'])
```

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

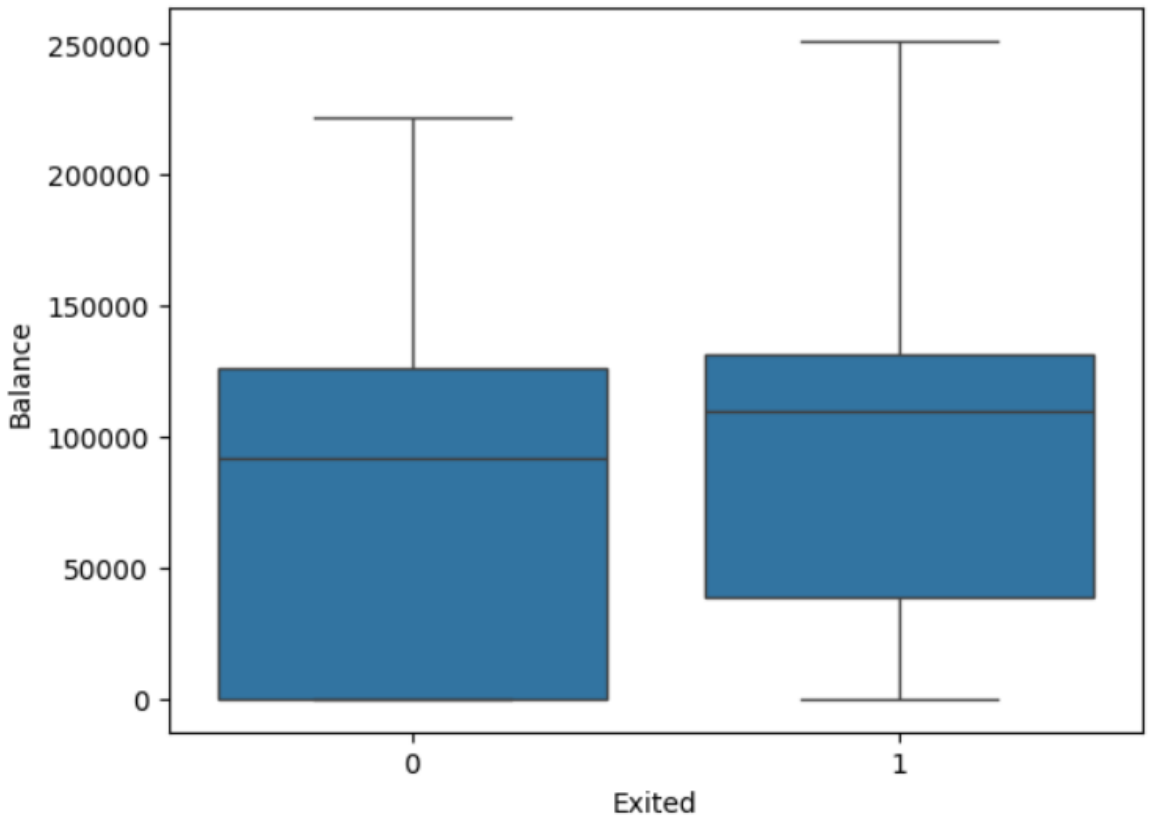



```
t_stats, p_value = ttest_ind(df[df['Exited'] == 0]['Tenure'],df[df['Exited'] == 1]['Tenure'])
print("t_stats :",t_stats)
print("p_value",p_value)
if p_value < 0.05:
  print("Null hypothesis is rejected")
else:
  print("Null hypothesis is accepted")
```

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

```
# Balance v/s Customer Churn
sbn.boxplot(y = df['Balance'], x= df['Exited'])
```

```
<Axes: xlabel='Exited', ylabel='Balance'>
```




```
[35] # Balence v/s Customer Churn
     print(" max Balance of person who churned ", df[df['Exited'] == 1]['Balance'].max())
     print(" min Balance of person who churned ",df[df['Exited'] == 1]['Balance'].min())
     print(" max Balance of person who didn't churned ", df[df['Exited'] == 0]['Balance'].max())
     print(" min Balance of person who  didn't churned ",df[df['Exited'] == 0]['Balance'].min())
```

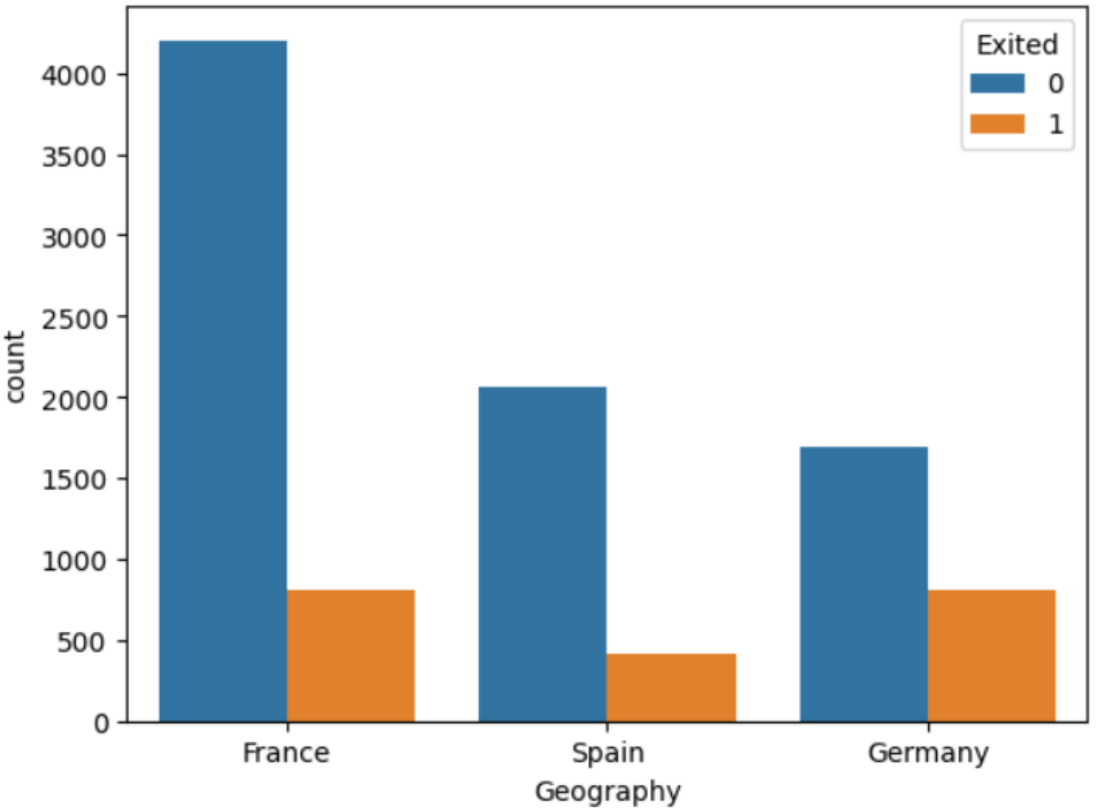
```
 max Balance of person who churned  250898.09
 min Balance of person who churned  0.0
 max Balance of person who didn't churned  221532.8
 min Balance of person who  didn't churned  0.0
```

```
# Geography v/s Customer Churn
GC = pd.crosstab(columns = df['Geography'],index = df['Exited'])
GC
```

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

```
sbn.countplot(x=df['Geography'],hue=df['Exited'])
```

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




```
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
```

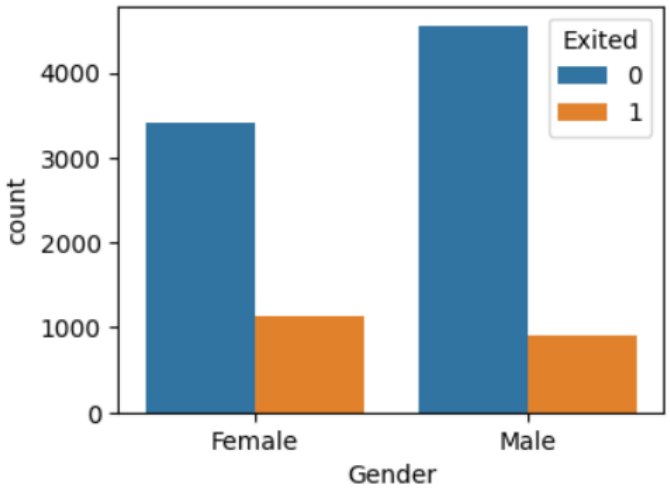
```
# Gende & Customer Churn
Gec = pd.crosstab(columns = df['Gender'],index = df['Exited'])
Gec
```

Gender	Female	Male
Exited		
0	3404	4558
1	1139	899

Next steps: Generate code with Gec | View recommended plots | New interactive sheet

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

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




```
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
```

```
# Imapct of Credit card on Churn rate
Cc = pd.crosstab(columns = df['Card Type'],index = df['Exited'])
Cc
```

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

Next steps: Generate code with Cc | View recommended plots | New interactive sheet

```
[48] 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")
```

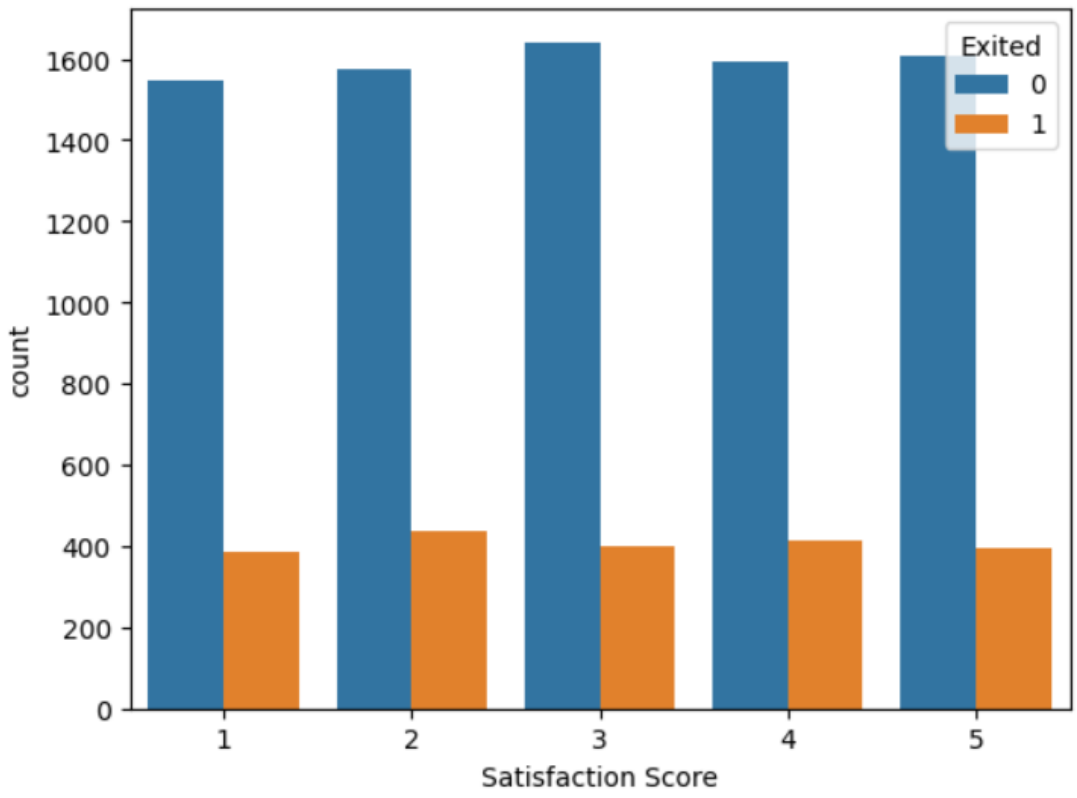
```
Result: Chi2ContingencyResult(statistic=112.39655374778587, pvalue=2.9253677618642e-26, dof=1, expected_freq=array([[3617.1366, 4344.8634
       [ 925.8634, 1112.1366]]))
t_stats : 112.39655374778587
p_value 2.9253677618642e-26
Null hypothesis is rejected
Credit Card and Customer churn are dependent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

```
0s
# Analayze Area for service improvement
pd.crosstab(columns = [df['Complain'],df['Satisfaction Score']],index = df['Exited'])
```

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

```
sbn.countplot(x=df['Satisfaction Score'],hue= df['Exited'])
```

```
<Axes: xlabel='Satisfaction Score', ylabel='count'>
```




```
# Strategies for customer retenion strategies
data_banking_behaviour = df.loc[df['Exited'] ==1,['CustomerId','Tenure','NumOfProducts','EstimatedSalary','Balance']]
data_banking_behaviour
```

	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

Next steps: Generate code with data_banking_behaviour | View recommended plots | New interactive sheet

```
data_banking_behaviour['Spent'] = data_banking_behaviour['EstimatedSalary']* data_banking_behaviour['Tenure']- data_banking_behaviour['Balance']
data_banking_behaviour
```

	CustomerId	Tenure	NumOfProducts	EstimatedSalary	Balance	Spent
0	15634602	2	1	101348.88	0.00	202697.76
2	15619304	8	3	113931.57	159660.80	751791.76
5	15574012	8	2	149756.71	113755.78	1084297.90
7	15656148	4	4	119346.88	115046.74	362340.78
16	15737452	1	1	5097.67	132602.88	-127505.21
...	...	...	...	...	...	...
9981	15672754	3	1	53445.17	152039.70	8295.81
9982	15768163	7	1	115146.40	137145.12	668879.68
9991	15769959	4	1	69384.71	88381.21	189157.63
9997	15584532	7	1	42085.58	0.00	294599.06
9998	15682355	3	2	92888.52	75075.31	203590.25

2038 rows × 6 columns

Insight:

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

```
data_banking_behaviour[data_banking_behaviour['Spent'] < 0 ]
```

	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 × 6 columns

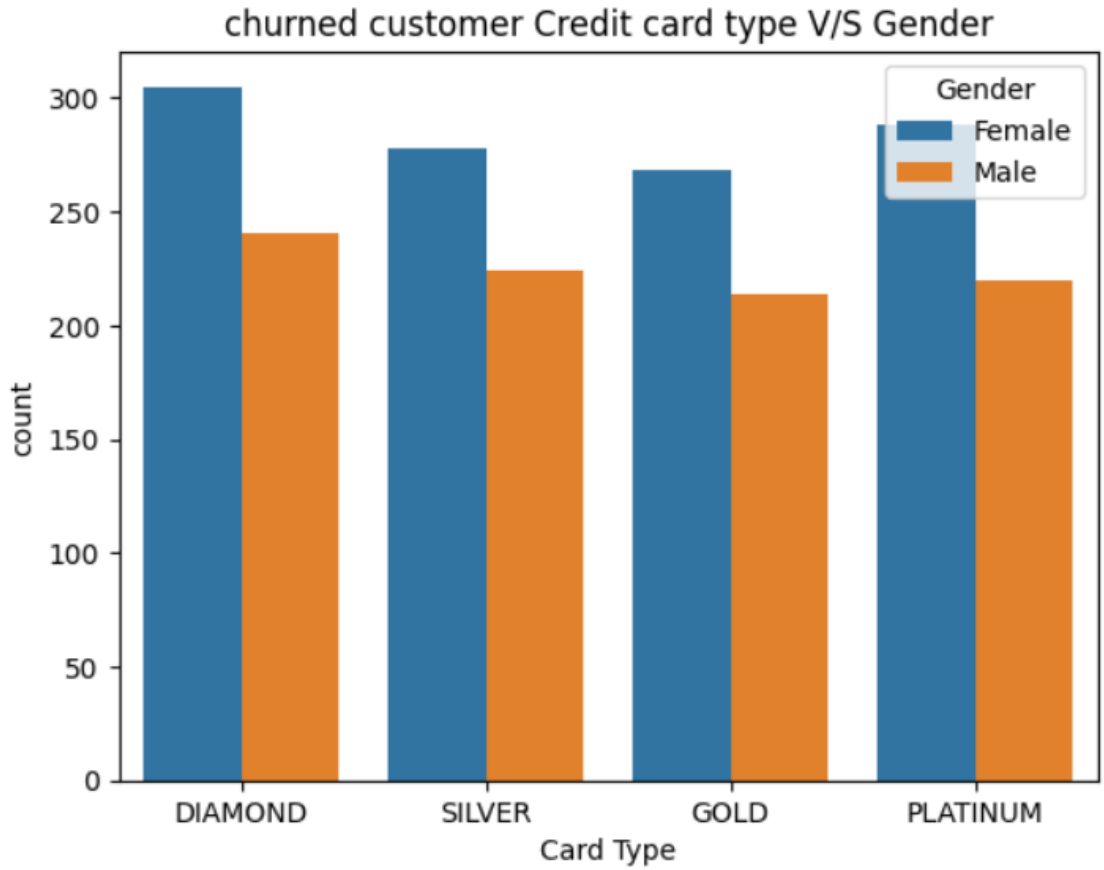
Insight:

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.

```
# Lets check the people whose balance were not zero or less but have complaint and
# churned out of the bank with different credit card
sbn.countplot(x = df[df['Exited'] == 1]['Card Type'],hue = df['Gender'])
plt.title("churned customer Credit card type V/S Gender")
```

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



```
df.loc[df['Exited']== 1,['Balance','Complain','Card Type','Satisfaction Score']]
```

	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

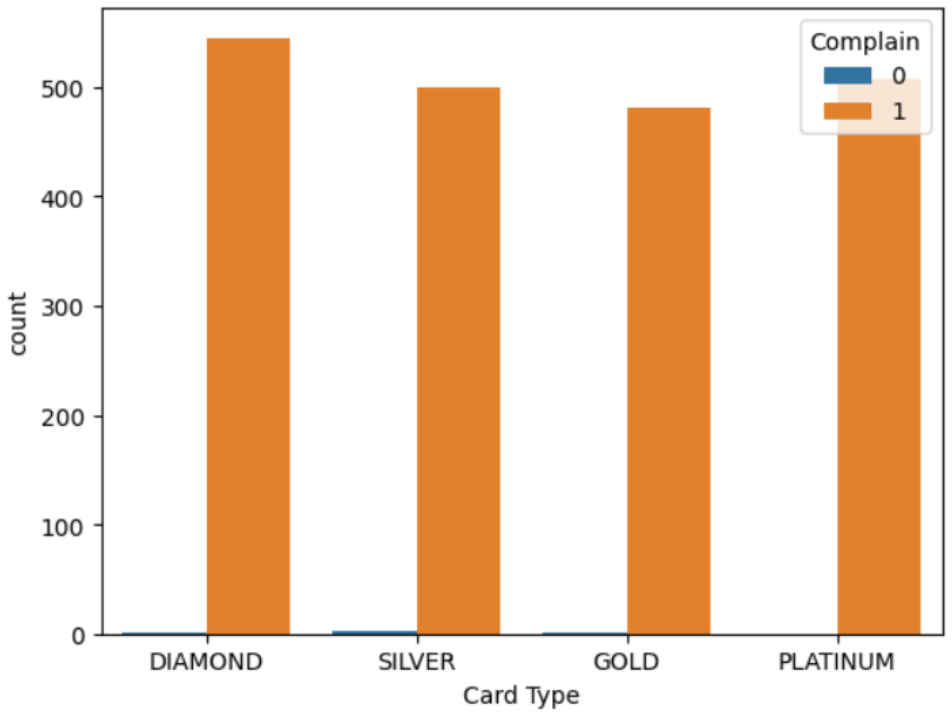
2038 rows × 4 columns

```
pd.crosstab(index = df[df['Exited'] == 1]['Card Type'],columns = df[df['Exited'] == 1]['Complain'],margins=True).reset_index()
```

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

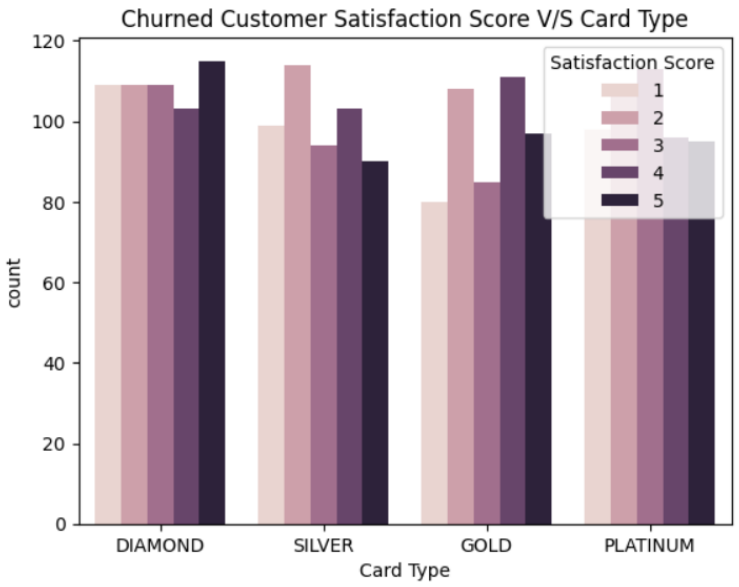
```
sbn.countplot(x = df[df['Exited'] == 1]['Card Type'],hue = df[df['Exited'] == 1]['Complain'])
```

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



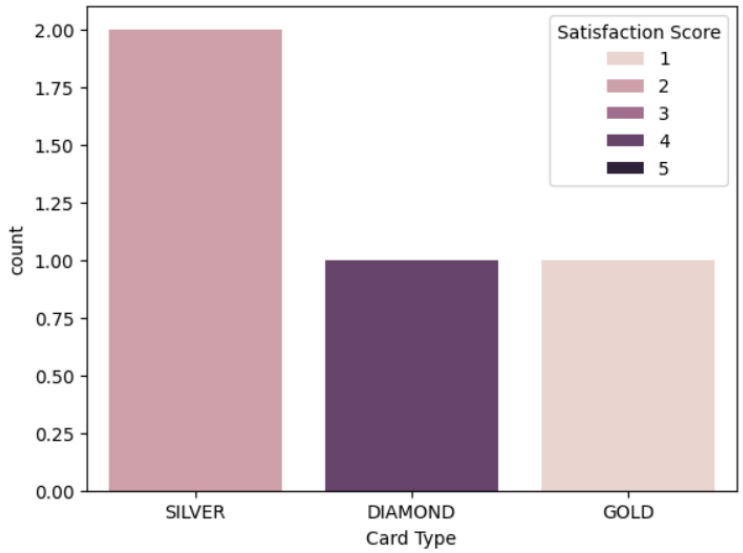

```
# Satisfaction score for Customer who churned out and have complained
# to banking services were visualize as shown below

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

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



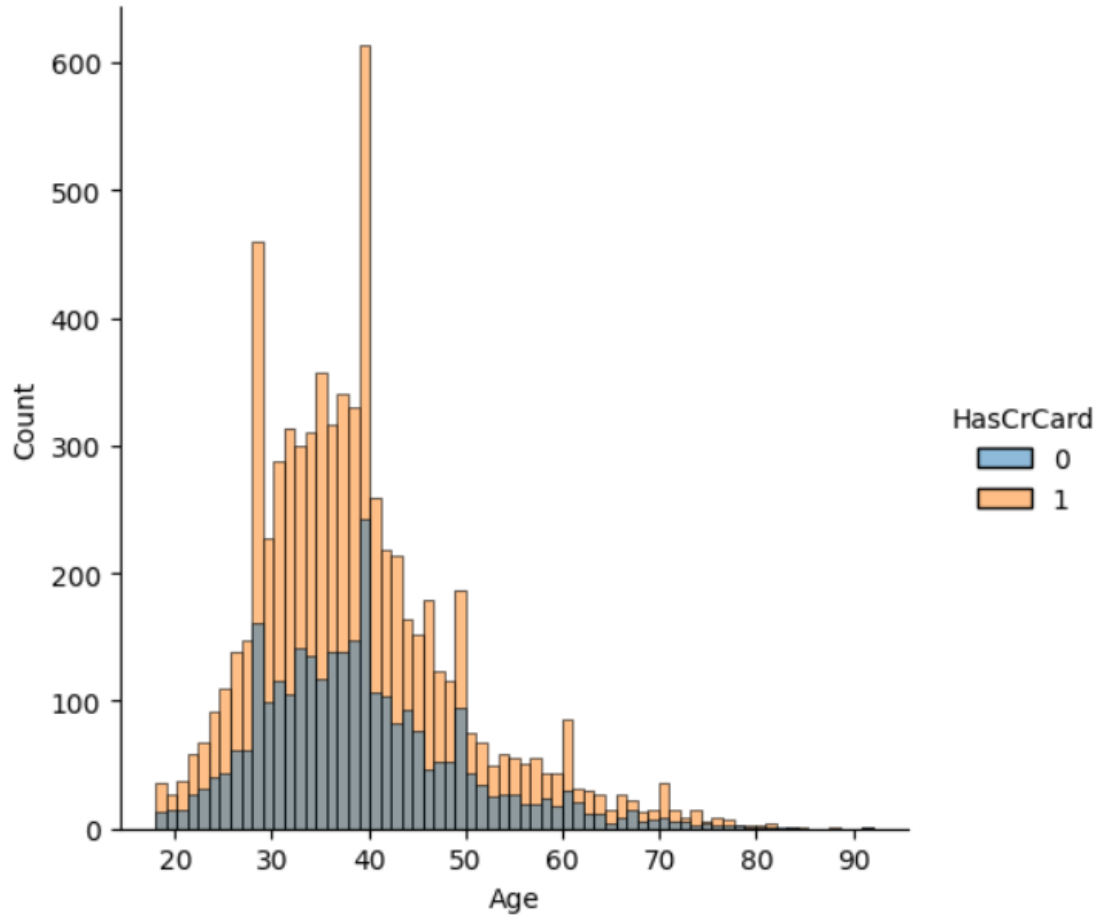
```
sbn.countplot(x = df[(df['Exited'] ==1) & (df['Complain']==0)]['Card Type'],hue = df[df['Exited'] == 1]['Satisfaction Score'])
```

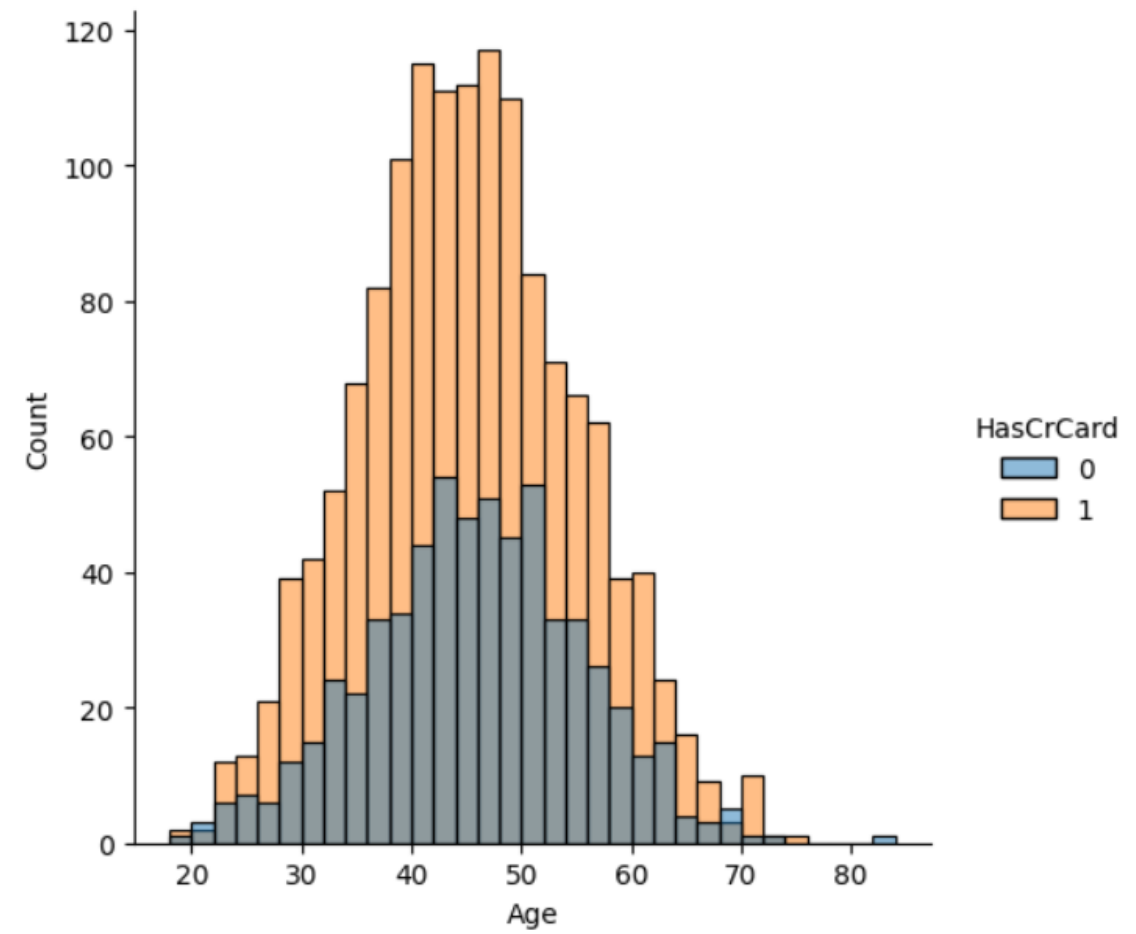
```
<Axes: xlabel='Card Type', ylabel='count'>
```



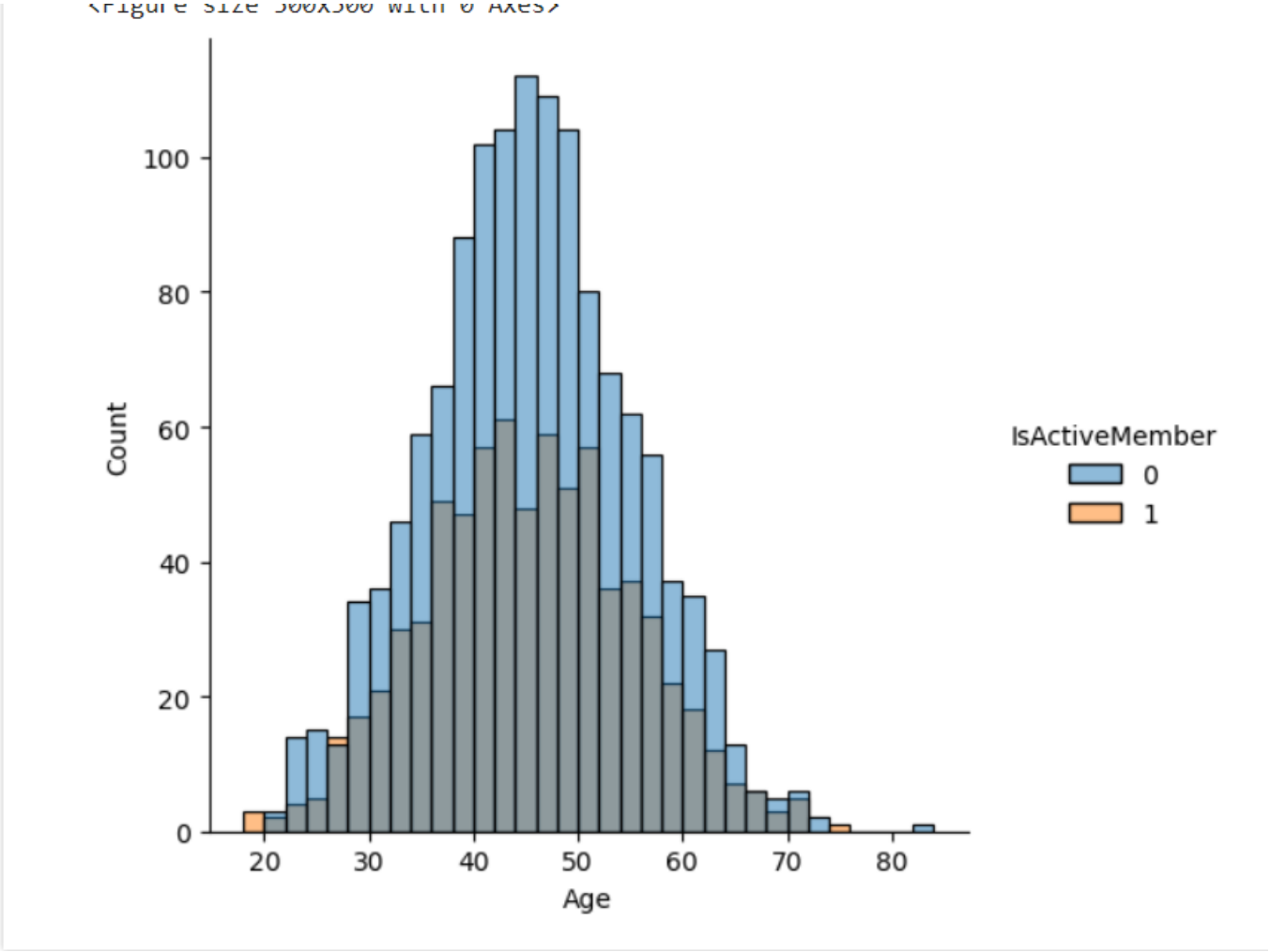
```python
# Checking Credit card Age wise
plt.figure(figsize=(5, 5))
sbn.displot(data=df, x="Age", hue="HasCrCard")
plt.figure(figsize=(5, 5))  # Create a new figure
sbn.displot(data=df[df["Exited"] == 1], x="Age", hue="HasCrCard")
plt.figure(figsize=(5, 5))
sbn.displot(data=df[df["Exited"] == 1], x="Age", hue="IsActiveMember")
```

```
<seaborn.axisgrid.FacetGrid at 0x7cb9eaa08f70>
<Figure size 500x500 with 0 Axes>
```






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

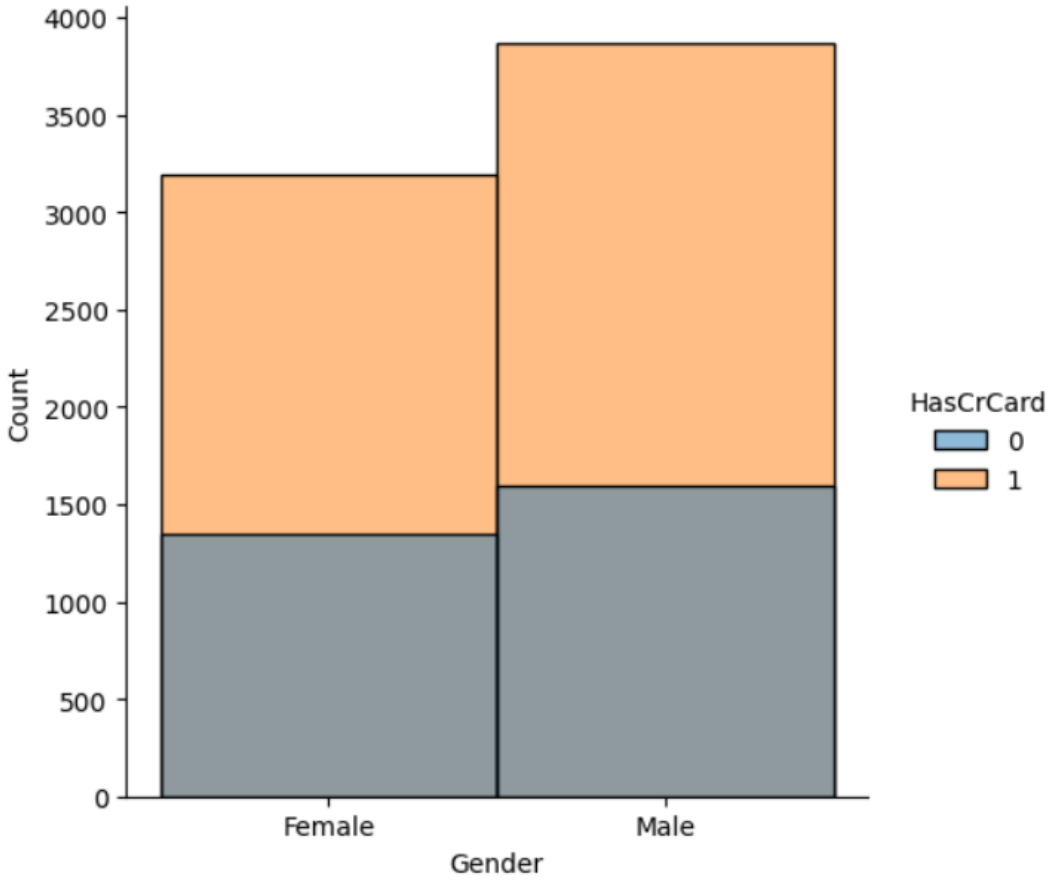


Insight:

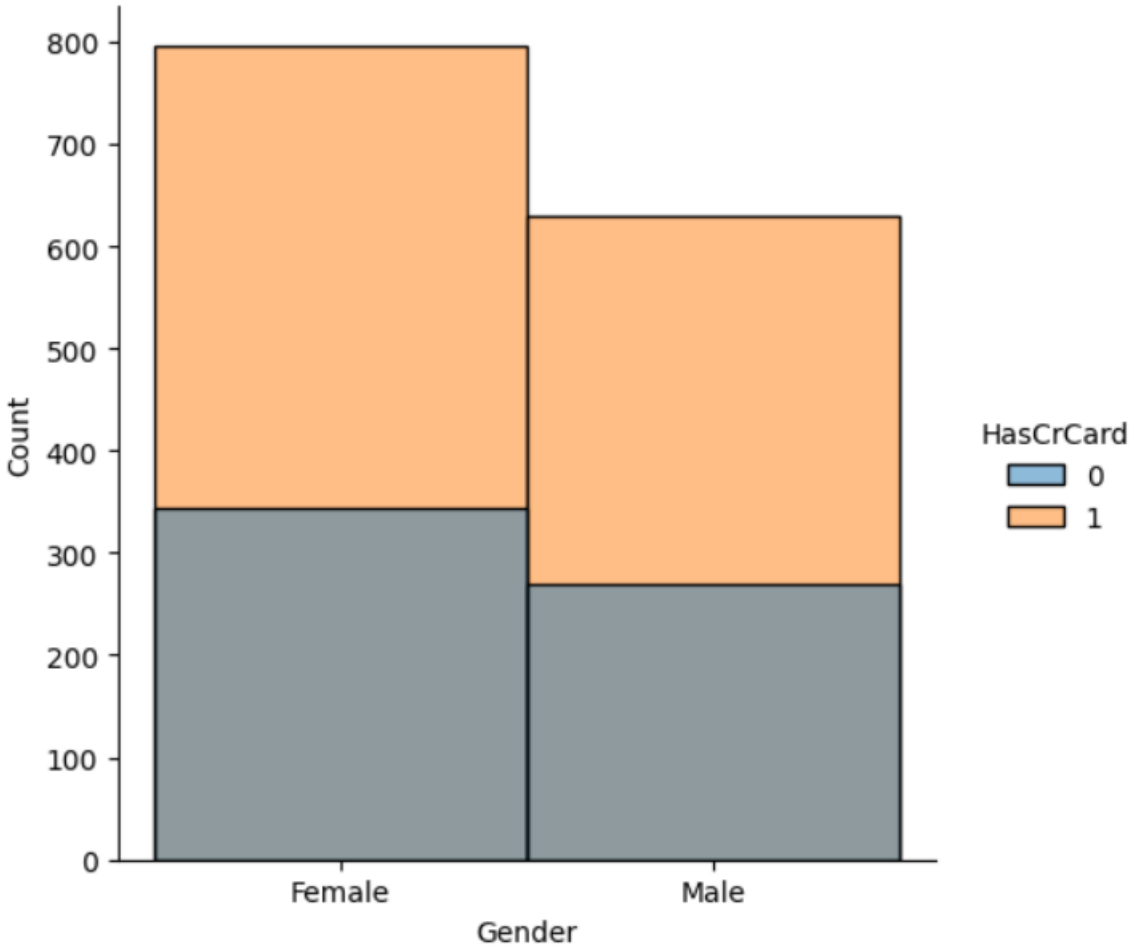
The people who churned were more active member in age group of 30-55. these are set of people who are customer of the bank now we will analyze customers who were churned were of.

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

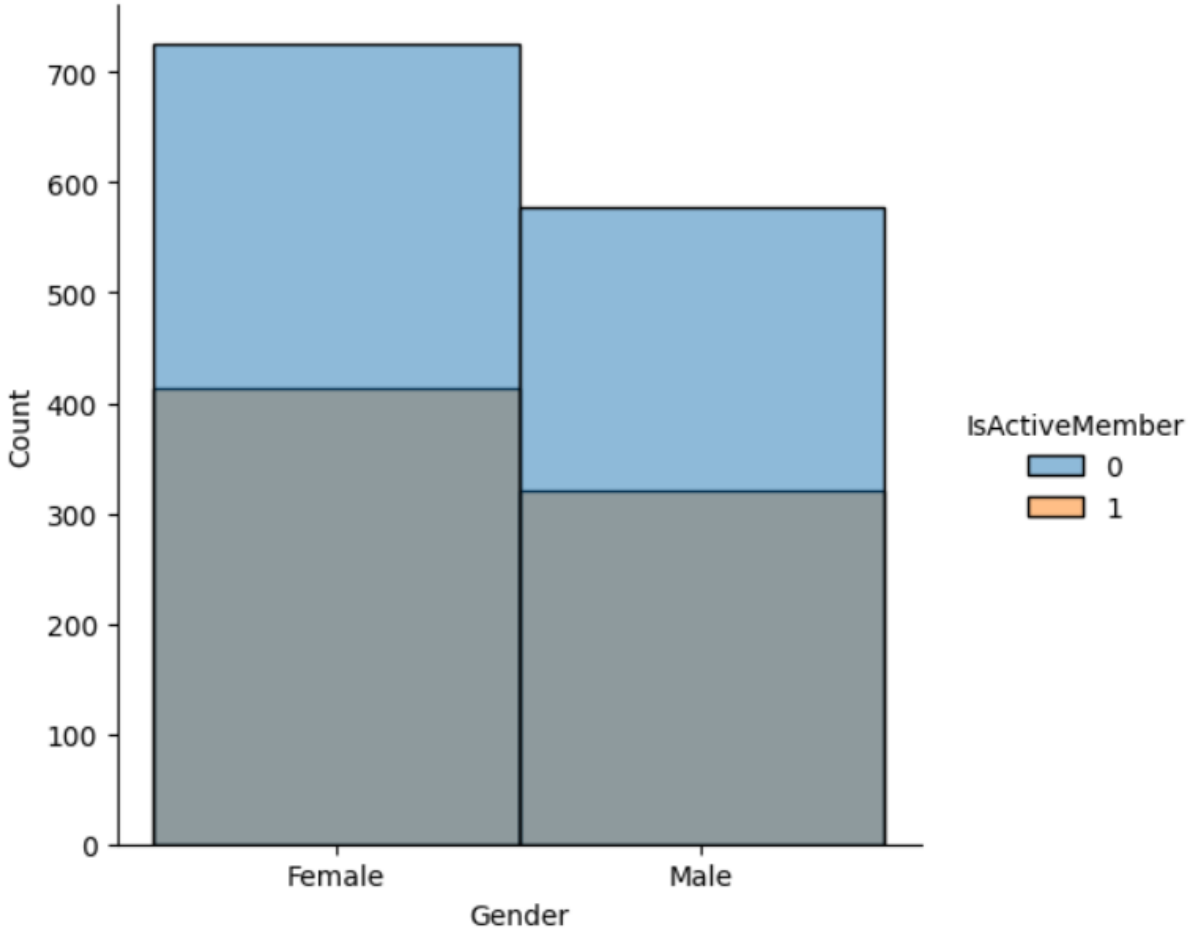
```
<seaborn.axisgrid.FacetGrid at 0x7cb9e9e01780>
<Figure size 500x500 with 0 Axes>
```



<Figure size 500x500 with 0 Axes>

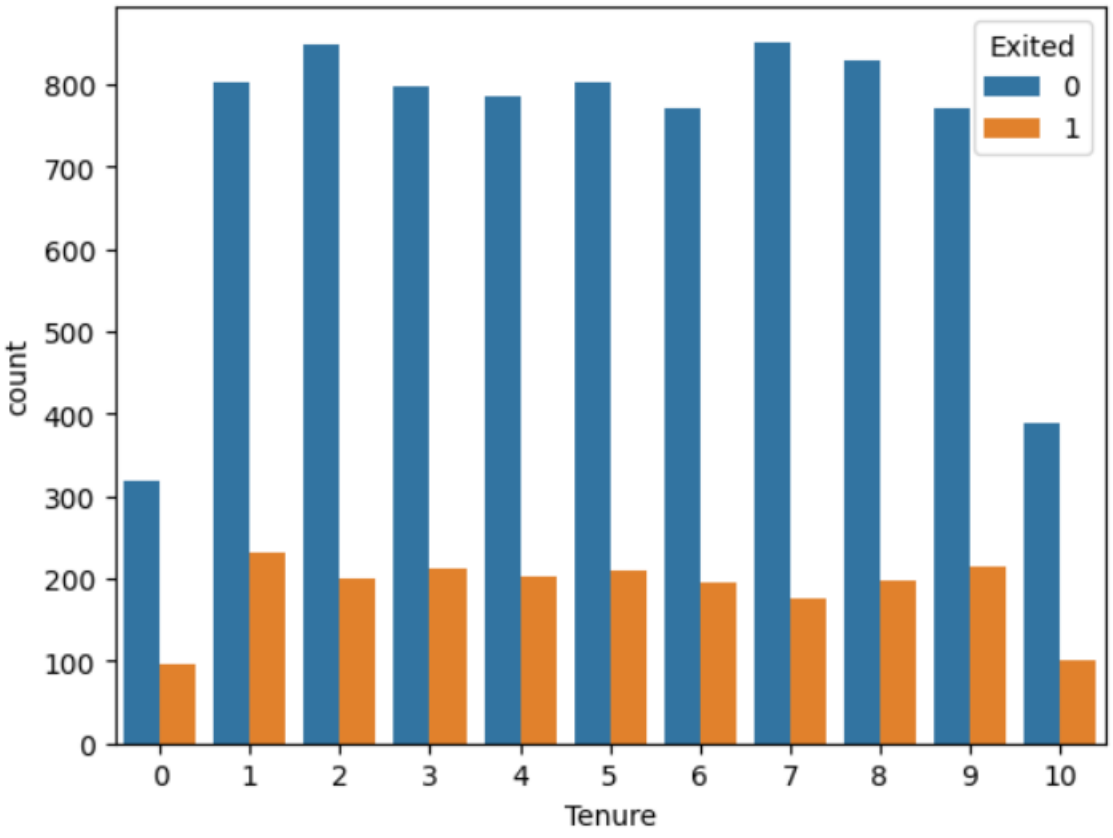


<Figure size 500x500 with 0 Axes>



```
# Descriptive analysis
# Churn rate
sbn.countplot(x=df['Tenure'],hue= df['Exited'])
```

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




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

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

```
churn_data = pd.crosstab(columns = df['Tenure'],index= df['Exited'],normalize = 'columns')
churn_data
```

Tenure	0	1	2	3	4	5	6	7	8	9	10
Exited											
0	0.769976	0.775845	0.808206	0.7889	0.794742	0.793478	0.797311	0.827821	0.807805	0.78252	0.793878
1	0.230024	0.224155	0.191794	0.2111	0.205258	0.206522	0.202689	0.172179	0.192195	0.21748	0.206122

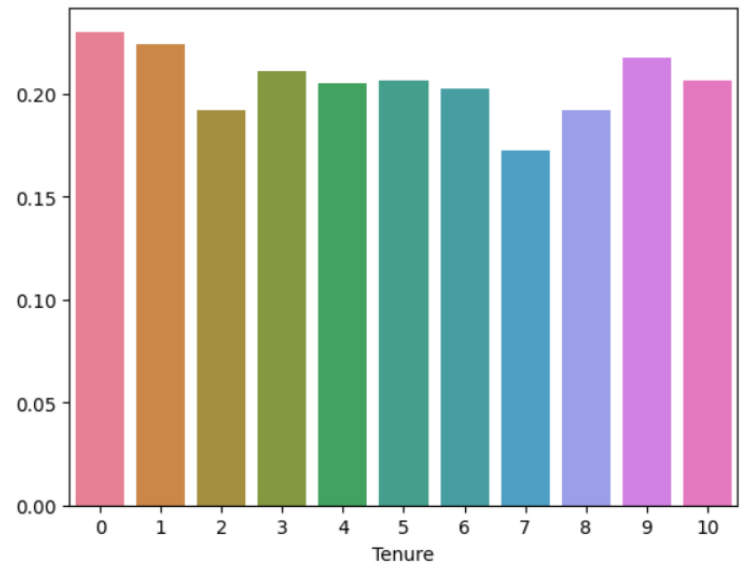
Next steps: Generate code with churn_data | View recommended plots | New interactive sheet

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

Tenure	Exited	0	1	2	3	4	5	6	7	8	9	10
0	1	0.230024	0.224155	0.191794	0.2111	0.205258	0.206522	0.202689	0.172179	0.192195	0.21748	0.206122

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

```
<Axes: xlabel='Tenure'>
```



Insight:
The Customer churns 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 mor 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 churns 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 incentive for both genders 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.