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.

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
df = pd.read_csv(r"C:\Users\SYEDA TAYABA\OneDrive\Documents\csv dataset
df.head(4)
```

Out[2]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure
0	1	15634602	Hargrave	619	France	Female	42	2
1	2	15647311	Hill	608	Spain	Female	41	1
2	3	15619304	Onio	502	France	Female	42	8
3	4	15701354	Boni	699	France	Female	39	1

In [3]:

```
df.shape
```

Out[3]:

```
(10000, 18)
```

In [4]:
```
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
```

In [5]:
```
df['CustomerId'].nunique()
```

Out[5]:
```
10000
```

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
RowNumber             0
CustomerId            0
Surname               0
CreditScore           0
Geography             0
Gender                0
Age                   0
Tenure                0
Balance               0
NumOfProducts         0
HasCrCard             0
IsActiveMember        0
EstimatedSalary       0
Exited                0
Complain              0
Satisfaction Score    0
Card Type             0
Point Earned          0
dtype: int64
```

- No missing values are present in dataset.

In [7]:
```
df.duplicated()
```

Out[7]:
```
0       False
1       False
2       False
3       False
4       False
        ...
9995    False
9996    False
9997    False
9998    False
9999    False
Length: 10000, dtype: bool
```

- No duplicate values are present, all are unique.

Exploratory Data Analysis

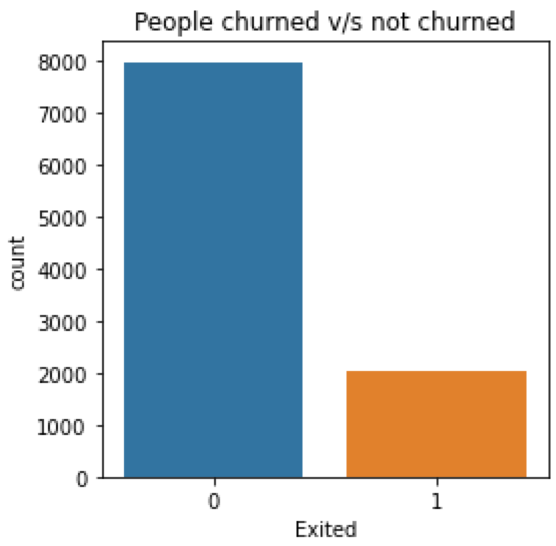
In [8]:
```
df[['CustomerId','Exited']].head(5)
```

Out[8]:

	CustomerId	Exited
0	15634602	1
1	15647311	0
2	15619304	1
3	15701354	0
4	15737888	0

Calculating No of Customers Exited

In [9]:
```
plt.figure(figsize=(4,4))
sns.countplot(x = df['Exited'])
plt.title("People churned v/s not churned")
plt.show()
```



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

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

- from above observation it is clear that 2038 people have exited from bank and 87962 are still holder at bank out of 10000.

No of Customers Complained & Exited

In [11]:
```
df['Complain'].value_counts()
```

Out[11]:
```
0    7956
1    2044
Name: Complain, dtype: int64
```

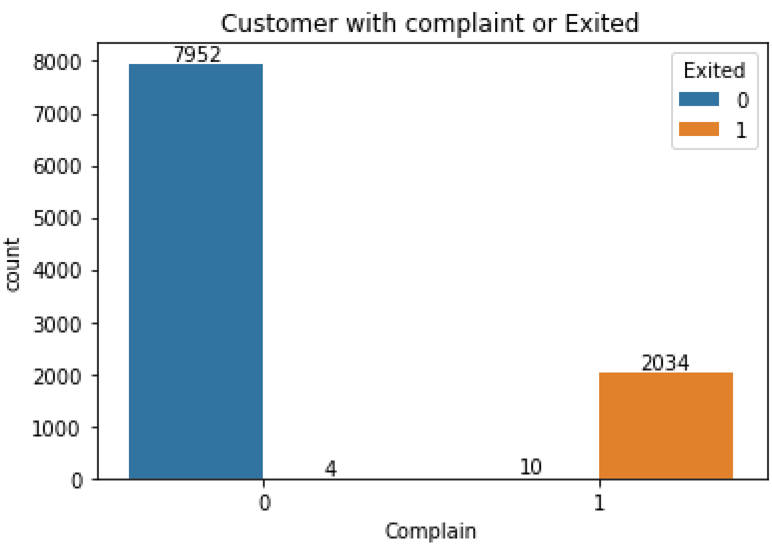
In [12]:
```
pd.crosstab(columns = df['Complain'], index = df['Exited'])
```

Out[12]:

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

- 2034 people complained & exited, 4 people did not complain but exited.

In [13]:
```
ax1 = sns.countplot(x=df['Complain'],hue=df['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.

Customers Satisfied

In [14]:
```
df['Satisfaction Score'].value_counts().sort_index()
```

Out[14]:
```
1    1932
2    2014
3    2042
4    2008
5    2004
Name: Satisfaction Score, dtype: int64
```

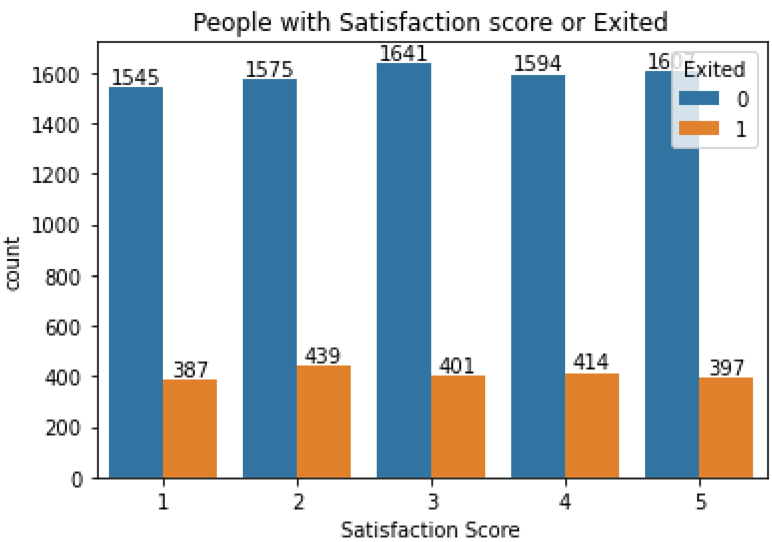
In [15]:
```
pd.crosstab(columns = df['Satisfaction Score'], index = df['Exited'])
```

Out[15]:

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

- 5 stars = 1607 people 397 exited
- 4 stars = 1594 people 414 exited
- 3 stars = 1641 people 401 exited
- 2 stars = 1575 people 439 exited
- 1 stars = 1545 people 387 exited

In [16]:
```
ax2 = sns.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')

plt.show()
```



Credited Card holder Exited

In [17]:
```
df['HasCrCard'].value_counts().sort_index()
```

Out[17]:
```
0    2945
1    7055
Name: HasCrCard, dtype: int64
```

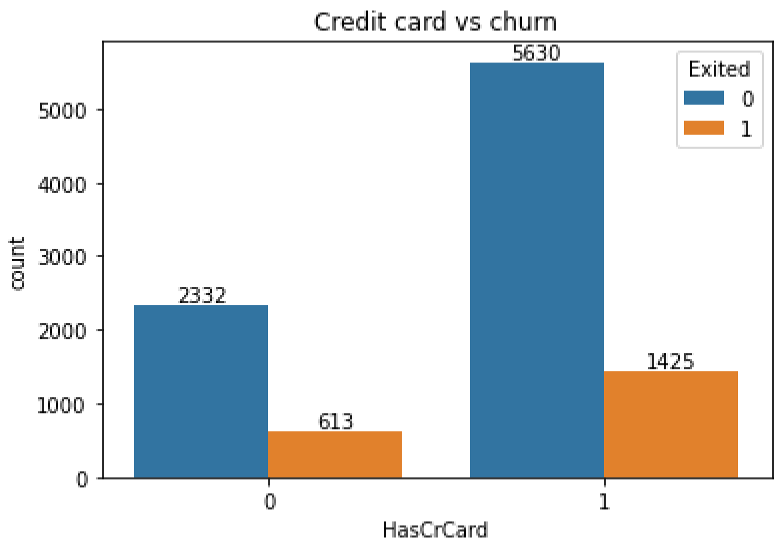
In [18]:
```
pd.crosstab(columns = df['HasCrCard'], index = df['Exited'])
```

Out[18]:

HasCrCard	0	1
Exited		
0	2332	5630
1	613	1425

- No card, 613 people exited.
- card, 1425 people exited.
- This shows card holder exited more than who have no cards.

In [19]:
```
ax3 = sns.countplot(x = df['HasCrCard'], hue = df['Exited'])
for container in ax3.containers:
    ax3.bar_label(container)
plt.title('Credit card vs churn')
plt.show()
```



Card Type

In [20]:
```
df['Card Type'].value_counts()
```

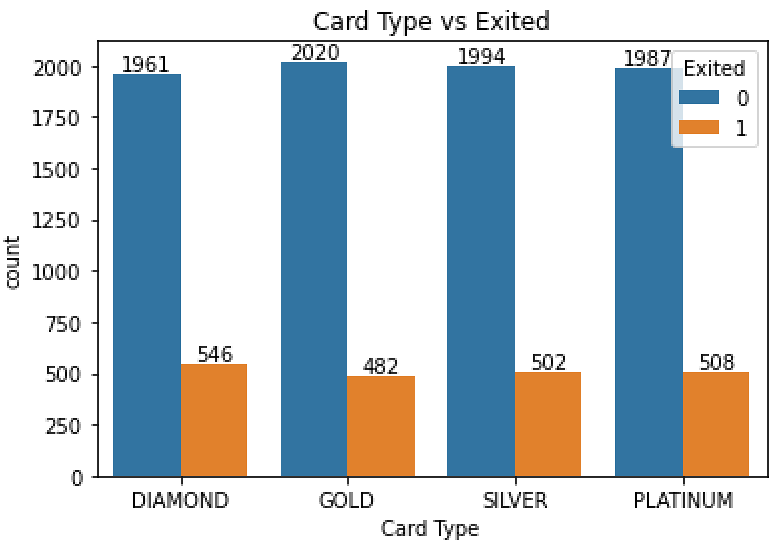
Out[20]:
```
DIAMOND     2507
GOLD        2502
SILVER      2496
PLATINUM    2495
Name: Card Type, dtype: int64
```

In [21]:
```
pd.crosstab(columns = df['Card Type'], index = df['Exited'])
```

Out[21]:

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

In [22]:
```
ax4 = sns.countplot(x = df['Card Type'], hue = df['Exited'])
for container in ax4.containers:
    ax4.bar_label(container)
plt.title('Card Type vs Exited')
plt.show()
```



Credit Score

In [23]:
```
df['CreditScore'].nunique()
```

Out[23]: 460

In [24]:
```
df[df['Exited']==1]['CreditScore'].max()
```

Out[24]: 850

In [25]:
```
bins = [300,400,500,600,700,800,900]
```

In [26]:
```
credit_bin = pd.cut(df[df['Exited']== 1]['CreditScore'], bins)
```

In [27]:
```
pd.crosstab(columns = credit_bin, index = df['Exited'])
```

Out[27]:

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 [28]:
```
#sns.barplot(pd.crosstab(columns = credit_bin ,index = df['Exited']))
#plt.title('People churned v/s Credit score')
```

Gender Customer

In [29]:
```
df['Gender'].value_counts()
```

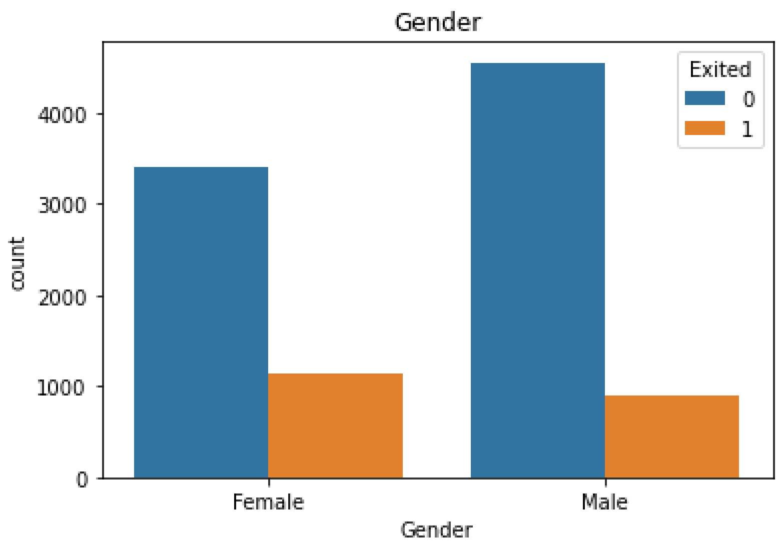
Out[29]:
```
Male      5457
Female    4543
Name: Gender, dtype: int64
```

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

Out[30]:

Gender	Female	Male
Exited		
0	3404	4558
1	1139	899

In [31]:
```
sns.countplot(x = df['Gender'], hue = df['Exited'])
plt.title('Gender')
plt.show()
```



Geography

In [32]:
```
df['Geography'].value_counts()
```

Out[32]:
```
France     5014
Germany    2509
Spain      2477
Name: Geography, dtype: int64
```

In [33]:
```
pd.crosstab(columns = df['Geography'], index = df['Exited'])
```

Out[33]:

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

In [34]:
```
pd.crosstab(columns = df['Geography'], index = df['Gender'])
```

Out[34]:

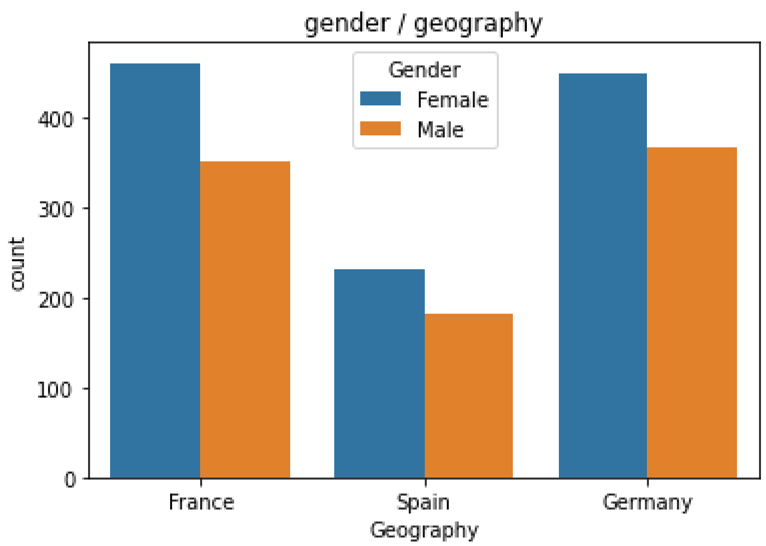
Geography	France	Germany	Spain
Gender			
Female	2261	1193	1089
Male	2753	1316	1388

In [35]:
```
pd.crosstab(columns = [df['Geography'], df['Gender']], index = df['Exit
```

Out[35]:

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 [36]:
```
sns.countplot(x= df[df['Exited'] == 1]['Geography'],hue = df[df['Exited
plt.title('gender / geography')
plt.show()
```



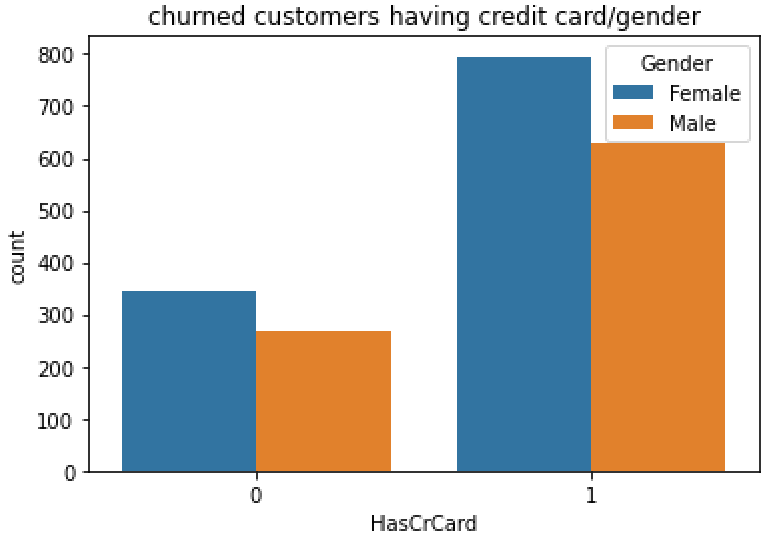
In [37]:
```
pd.crosstab(columns = [df['HasCrCard'], df['Gender']], index = df['Exit
```

Out[37]:

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

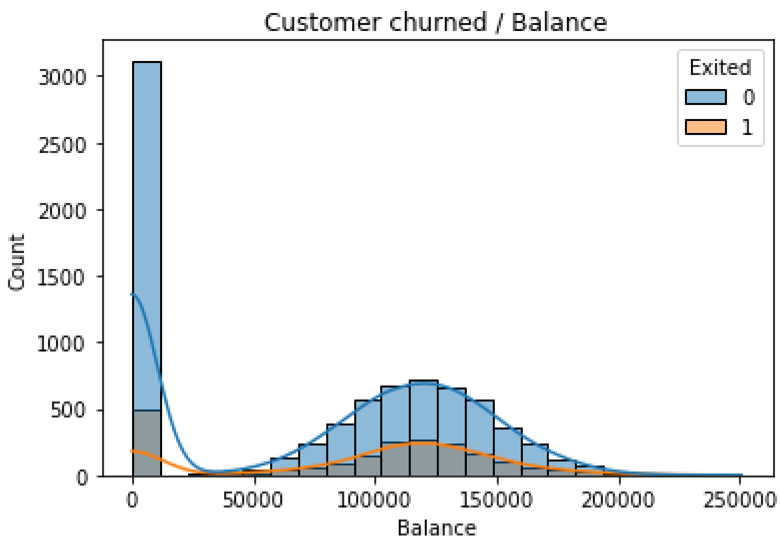
In [38]:
```
sns.countplot(x = df[df['Exited']==1] ['HasCrCard'], hue = df[df['Exite
plt.title('churned customers having credit card/gender')
```

Out[38]: `Text(0.5, 1.0, 'churned customers having credit card/gender')`



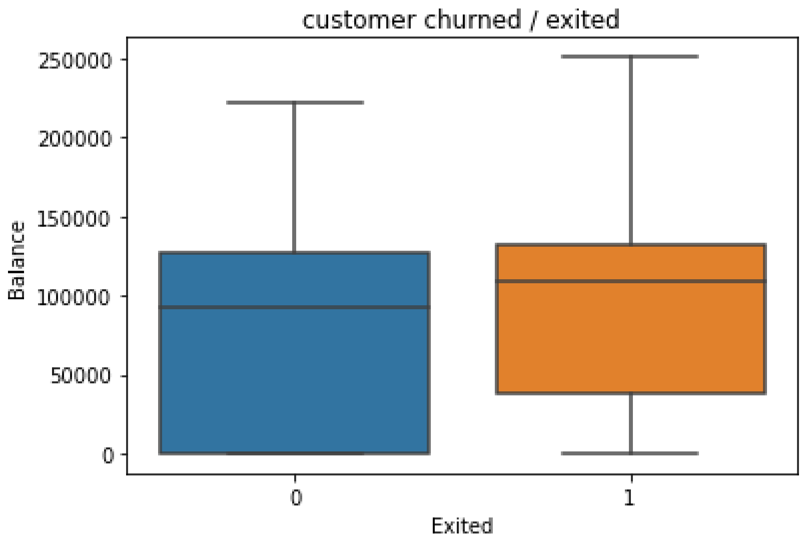
In [39]:
```
sns.histplot(data = df, x=df['Balance'], hue = df['Exited'], kde= True)
plt.title('Customer churned / Balance')
```

Out[39]: `Text(0.5, 1.0, 'Customer churned / Balance')`



In [40]:
```
sns.boxplot(data = df, x =df['Exited'], y = df['Balance'])
plt.title("customer churned / exited")
```

Out[40]:
```
Text(0.5, 1.0, 'customer churned / exited')
```



In [41]:
```
# tenure = years
df['Tenure'].value_counts().sort_index()
```

Out[41]:
```
0      413
1     1035
2     1048
3     1009
4      989
5     1012
6      967
7     1028
8     1025
9      984
10     490
Name: Tenure, dtype: int64
```

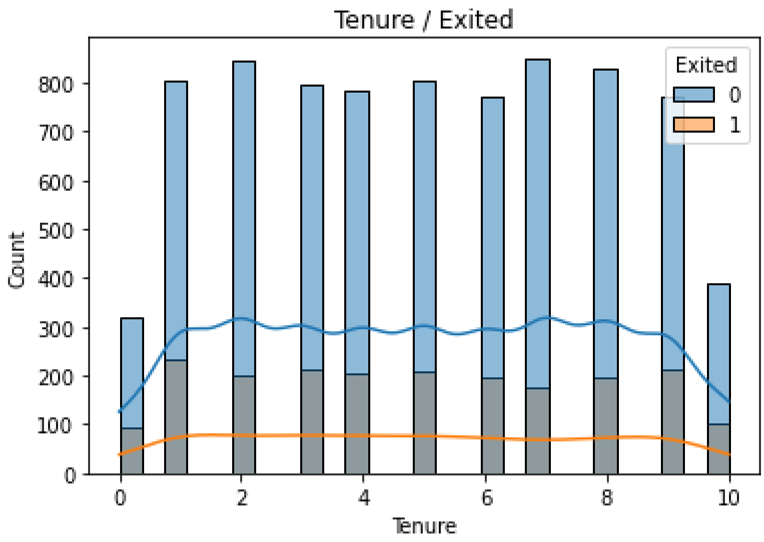
In [42]:
```
pd.crosstab(columns =  df['Tenure'], index = df['Exited'])
```

Out[42]:

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 [43]:
```
sns.histplot(x= df['Tenure'], hue = df['Exited'], kde = True)
plt.title('Tenure / Exited')
```

Out[43]: Text(0.5, 1.0, 'Tenure / Exited')



In [44]:
```
df[df['Exited']==1]['Tenure'].value_counts().reset_index()
```

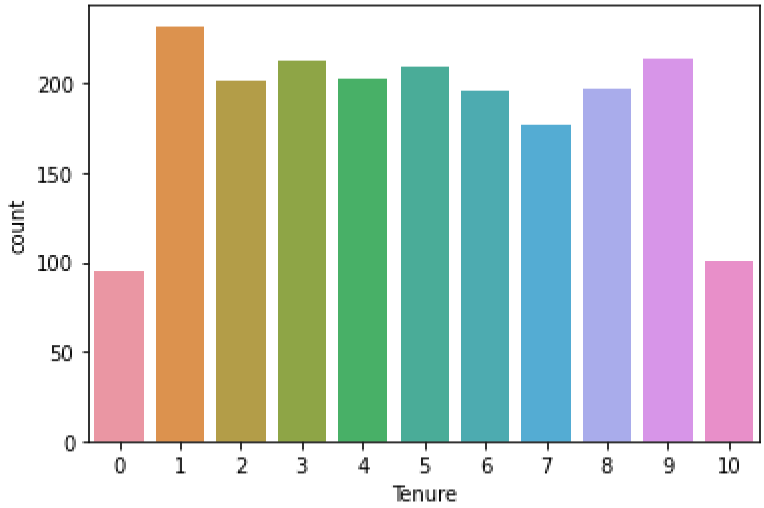
Out[44]:

	index	Tenure
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 [45]:

```
sns.countplot(x = df[df['Exited']==1]['Tenure'])
```

Out[45]: <AxesSubplot:xlabel='Tenure', ylabel='count'>

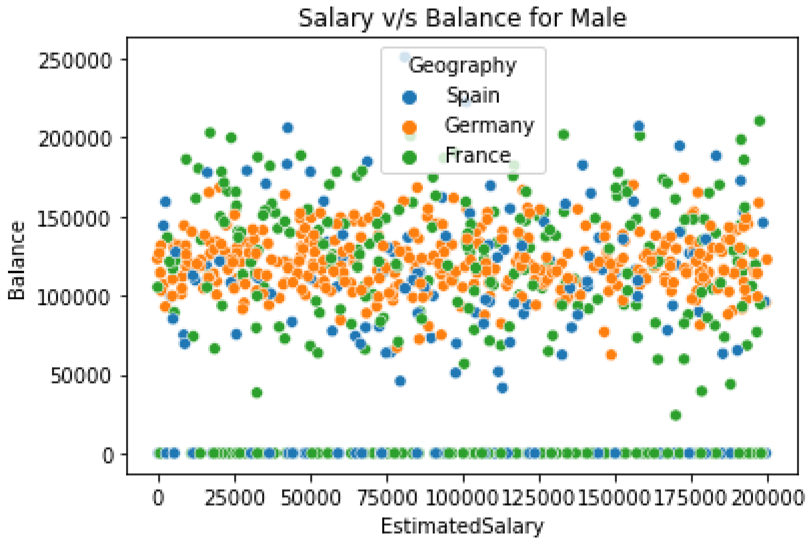


Lets check Estimated salary v/s balance of people w.r.t to Geography for different genders who left the bank.

Male

In [47]:

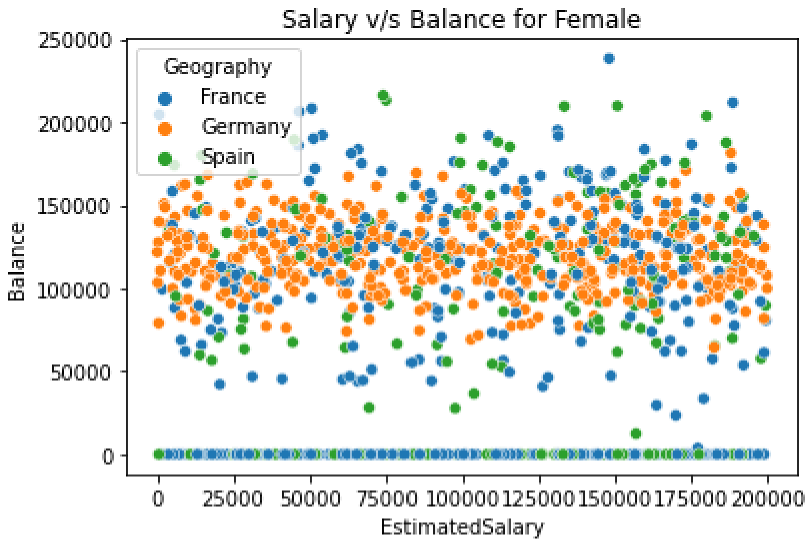
```
ax = sns.scatterplot(x="EstimatedSalary", y="Balance",
                     hue="Geography",
                     data=df[(df['Exited']==1) & (df['Gender'] =='Male'
ax.set_title('Salary v/s Balance for Male')
plt.show()
```



Female

In [48]:

```
ax = sns.scatterplot(x="EstimatedSalary", y="Balance",
                     hue="Geography",
                     data=df[(df['Exited']==1) & (df['Gender'] =='Femal
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
- we will use ANOVA for our hypothesis testing

In [51]:

```
d1 = df[['CreditScore','Exited']]
d1
```

Out[51]:

	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 [52]:

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

- H0 = Customer churn is independent of credit score.
- H1 = Customer churn is dependent of Credit score.

In [56]:

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

Age vs Customer churn

we will use ttest_ind

In [57]:
```python
df[['Age','Exited']]
```

Out[57]:

	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

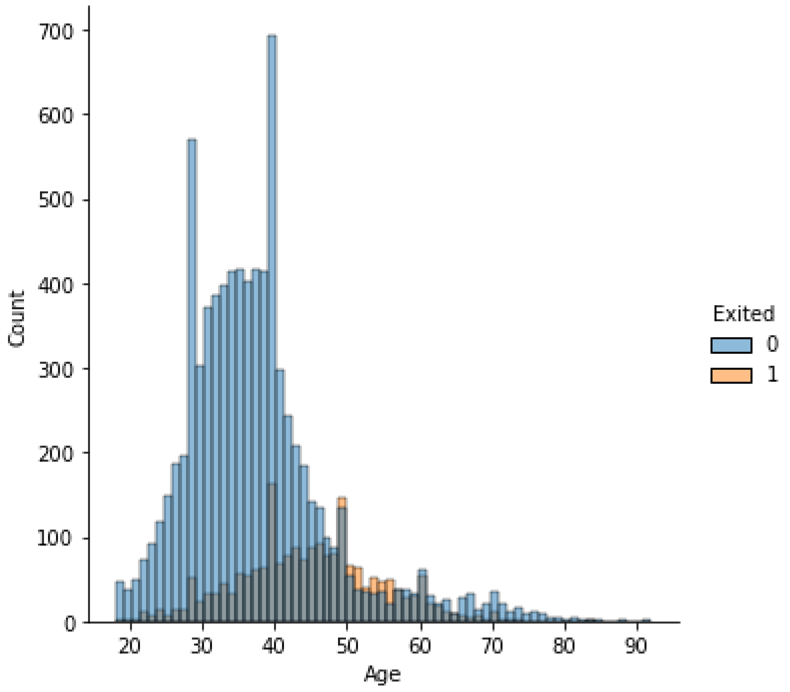
In [59]:
```python
t_stats, p_value = ttest_ind(df[df['Exited'] == 0]['Age'],df[df['Exited
print("t_stats :",t_stats)
print("p_value",p_value)
if p_value < 0.05:
  print("Null hypothesis is rejected")
else:
  print("Null hypothesis is accepted")
```

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

In [62]:
```python
plt.figure(figsize=(5, 5))
sns.displot(data=df, x="Age", hue="Exited")
```

Out[62]: <seaborn.axisgrid.FacetGrid at 0x276000729a0>

<Figure size 360x360 with 0 Axes>



Tenure V/s Customer churn

In [63]:
```python
df[['Tenure','Exited']]
```

Out[63]:

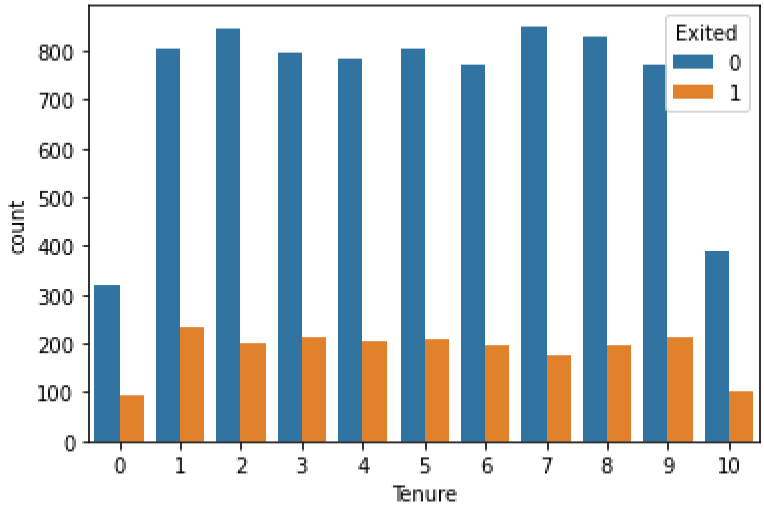
	Tenure	Exited
0	2	1
1	1	0
2	8	1
3	1	0
4	2	0
...	...	...
9995	5	0
9996	10	0
9997	7	1
9998	3	1
9999	4	0

10000 rows × 2 columns

In [64]:

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

Out[64]: <AxesSubplot:xlabel='Tenure', ylabel='count'>



- H0: Customer churn is independent of tenure
- Ha: Customer churn is dependent of tenure

In [66]:

```
t_stats, p_value = ttest_ind(df[df['Exited'] == 0]['Tenure'],df[df['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 : 1.365570678788837
p_value 0.1721044754880606
Null hypothesis is accepted
```

Balance vs Customer Churn

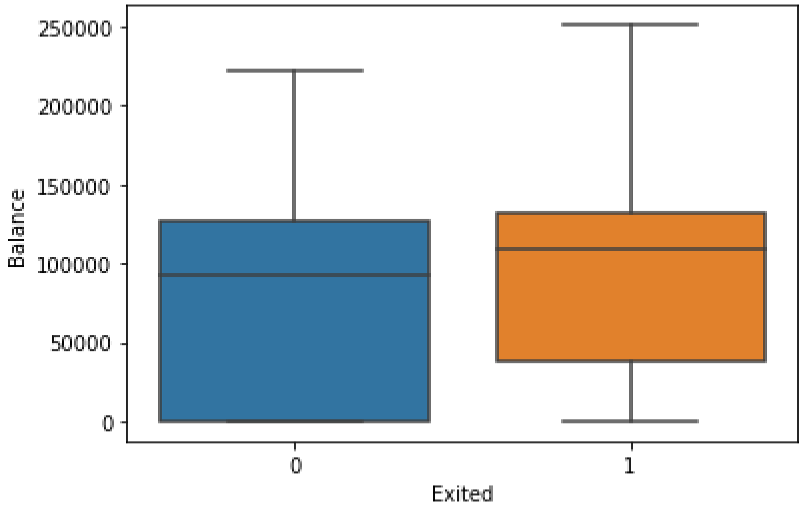
In [68]:

```
print(" max Balance of person who churned ", df[df['Exited'] == 1]['Bal
print(" min Balance of person who churned ",df[df['Exited'] == 1]['Bala
print(" max Balance of person who didn't churned ", df[df['Exited'] ==
print(" min Balance of person who  didn't churned ",df[df['Exited'] ==
```

```
 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 [70]:
```
sns.boxplot(y = df['Balance'], x =df['Exited'])
```

Out[70]: <AxesSubplot: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 [72]:
```
t_stats, p_value = ttest_ind(df[df['Exited'] == 0]['Balance'],df[df['Ex
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 [75]:
```
GC = pd.crosstab(columns = df['Geography'],index = df['Exited'])
GC
```

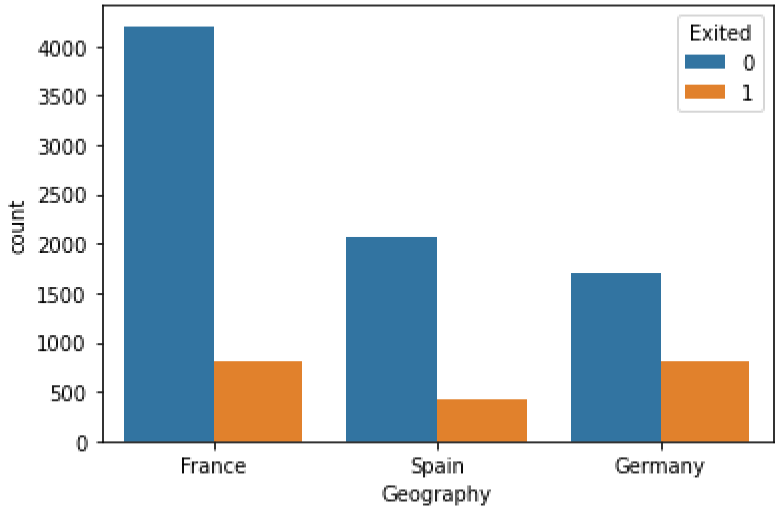
Out[75]:

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

In [77]:

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

Out[77]: <AxesSubplot: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 [78]:

```
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: (300.6264011211942, 5.245736109572763e-66, 2, array([[3992.146
8, 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 [80]:

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

Out[80]:

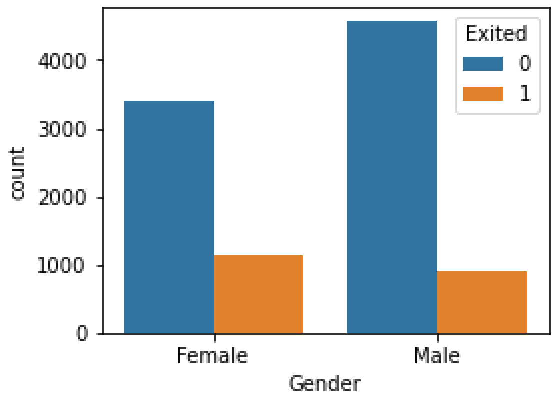
Gender	Female	Male
Exited		
0	3404	4558
1	1139	899

In [82]:

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

Out[82]:

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



- H0: Gender and Customer churn are independent
- Ha : Gender and Customer churn are dependent

In [83]:

```
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: (112.39655374778587, 2.9253677618642e-26, 1, array([[3617.136
6, 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 [85]:

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

Out[85]:

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 [86]:

```
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: (112.39655374778587, 2.9253677618642e-26, 1, array([[3617.136
6, 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 [87]:

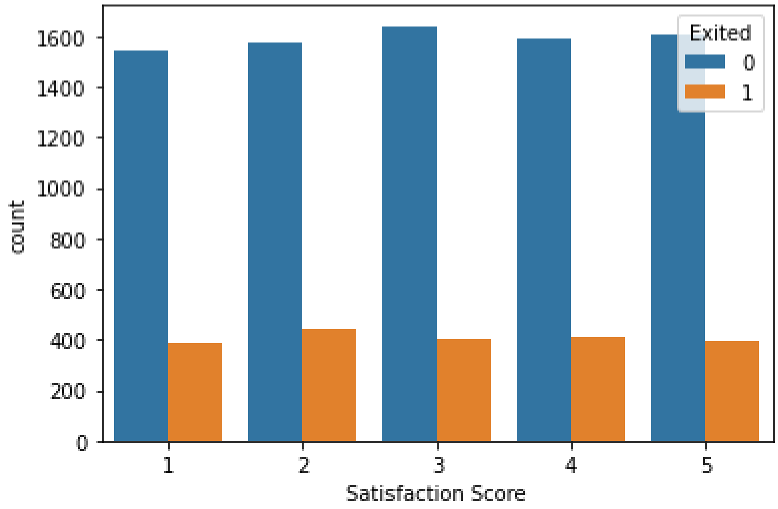
```
pd.crosstab(columns = [df['Complain'], df['Satisfaction Score']], index
```

Out[87]:

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

Out[89]: <AxesSubplot: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 [92]:
```
data_banking_behaviour = df.loc[df['Exited'] ==1,['CustomerId','Tenure'
data_banking_behaviour
```

Out[92]:

	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 [93]:

```
data_banking_behaviour['Spent'] = data_banking_behaviour['EstimatedSala
data_banking_behaviour
```

Out[93]:

	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

In [94]:

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

Out[94]:

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 [95]:

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

Out[95]:

	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

- 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

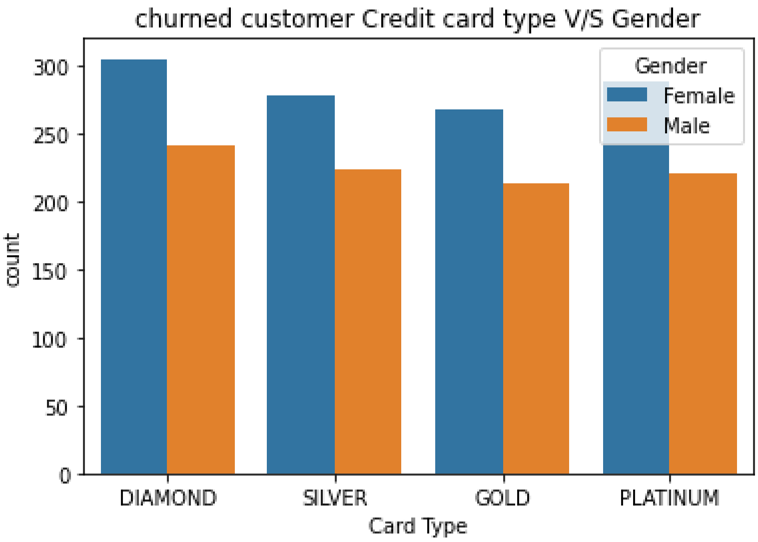
Lets check the people whose balance were not zero or less but have complaint and churned out of the bank with different credit card.

In [97]:

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

Out[97]:

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



In [98]:

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

Out[98]:

	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

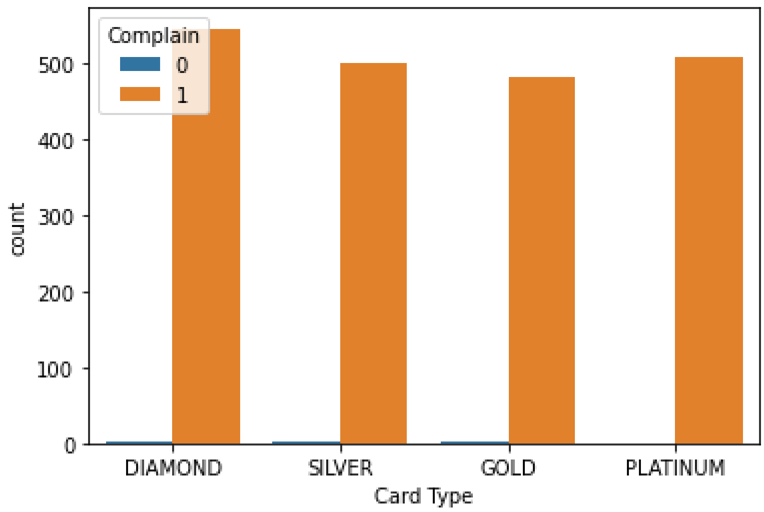
In [100]:
```
pd.crosstab(index = df[df['Exited'] == 1]['Card Type'],columns = df[df[
```

Out[100]:

	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 [102]:
```
sns.countplot(x = df[df['Exited'] == 1]['Card Type'],hue = df[df['Exite
```

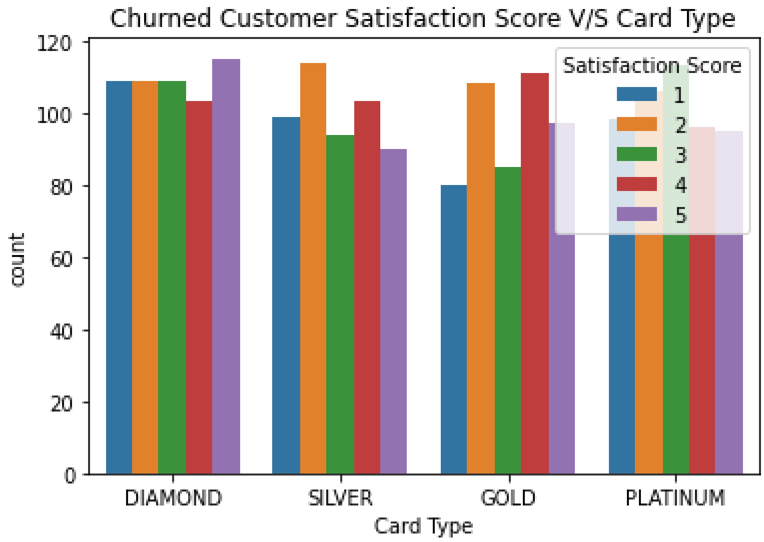
Out[102]: <AxesSubplot:xlabel='Card Type', ylabel='count'>



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

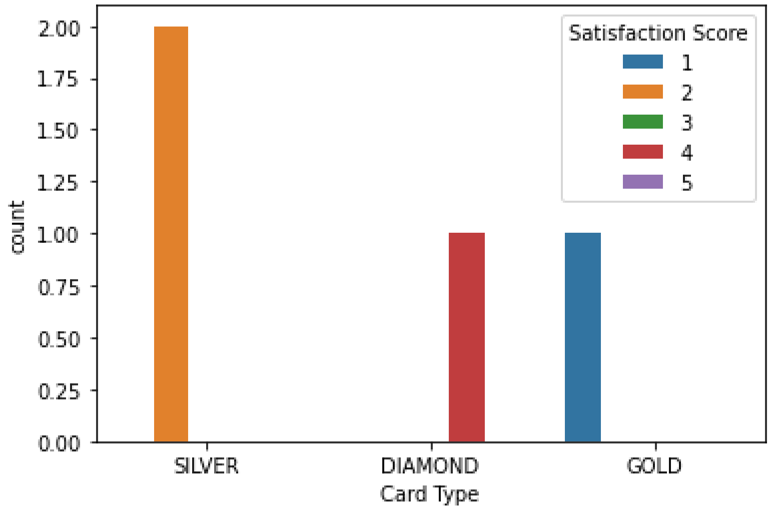
In [106]:
```
sns.countplot(x = df[(df['Exited'] ==1) & (df['Complain']==1)]['Card Ty
plt.title('Churned Customer Satisfaction Score V/S Card Type')
```

Out[106]: Text(0.5, 1.0, 'Churned Customer Satisfaction Score V/S Card Type')



In [108]:
```
sns.countplot(x = df[(df['Exited'] ==1) & (df['Complain']==0)]['Card Ty
```

Out[108]: <AxesSubplot:xlabel='Card Type', ylabel='count'>



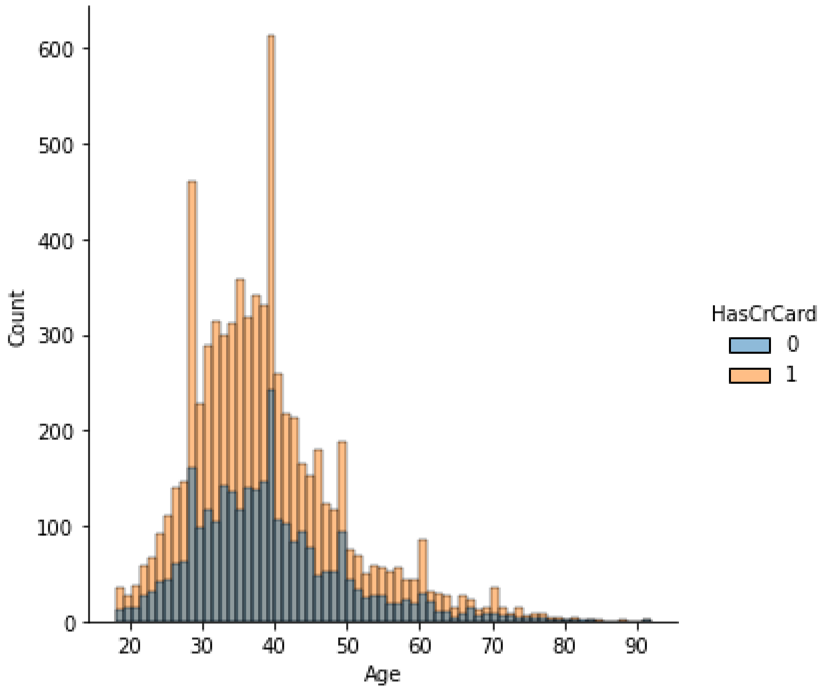
Checking Credit card Age wise

In [111]:

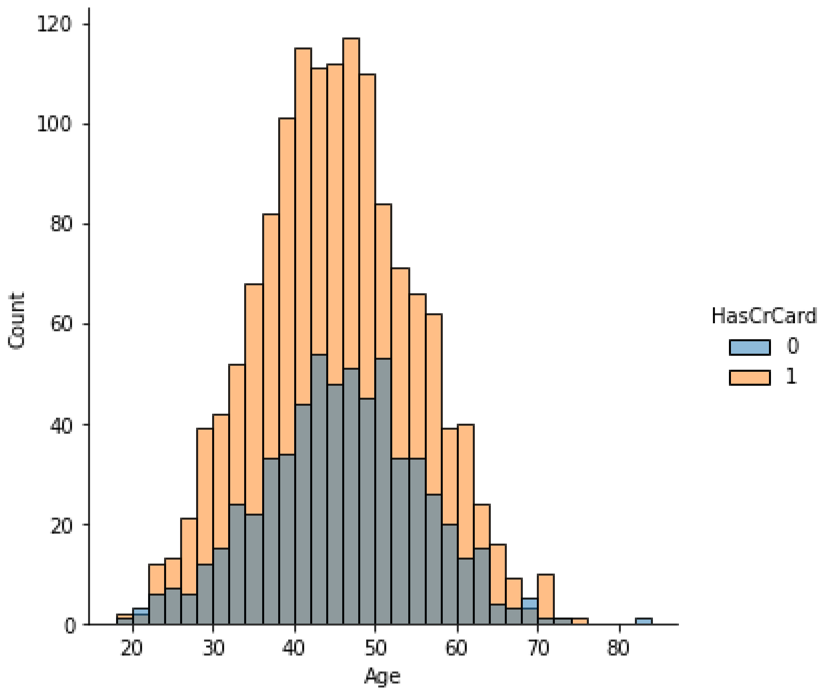
```
plt.figure(figsize=(5, 5))
sns.displot(data=df, x="Age", hue="HasCrCard")
plt.figure(figsize=(5, 5))  # Create a new figure
sns.displot(data=df[df["Exited"] == 1], x="Age", hue="HasCrCard")
plt.figure(figsize=(5, 5))
sns.displot(data=df[df["Exited"] == 1], x="Age", hue="IsActiveMember")
```

Out[111]:

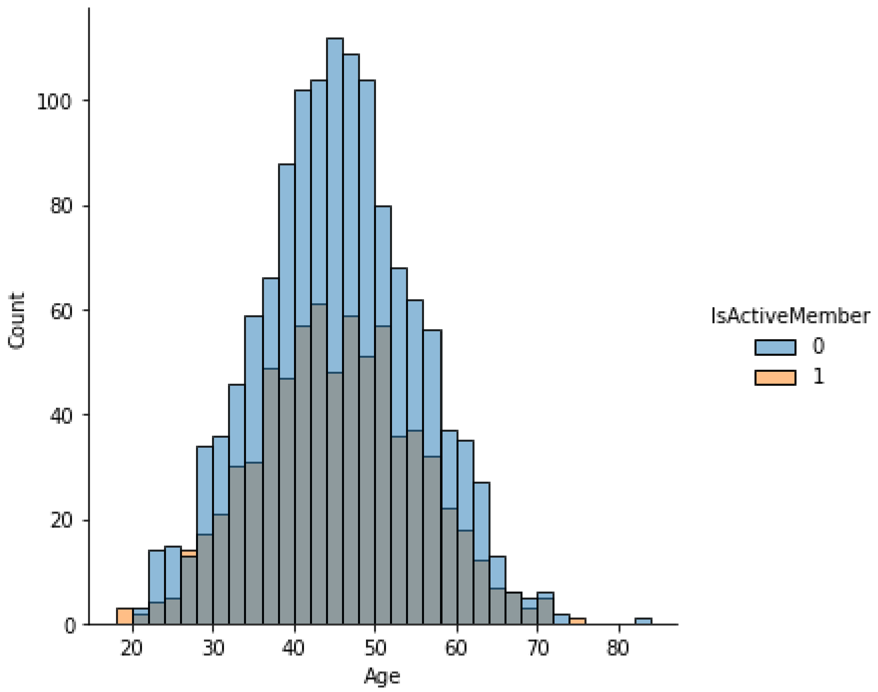
```
<seaborn.axisgrid.FacetGrid at 0x27604e9fd90>

<Figure size 360x360 with 0 Axes>
```




```
<Figure size 360x360 with 0 Axes>
```




```
<Figure size 360x360 with 0 Axes>
```



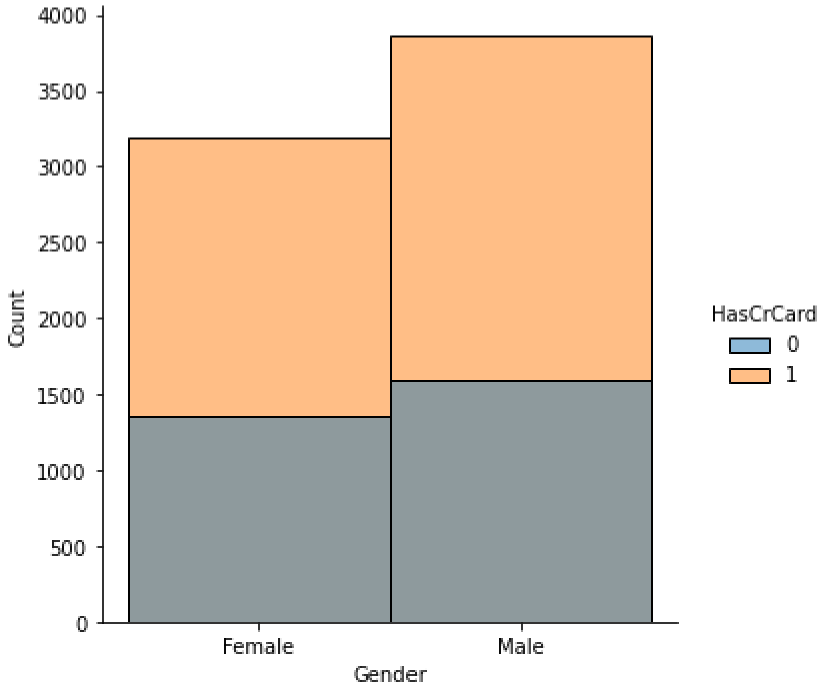
- the people who churned were more active member in age group of 30-55.

In [114]:

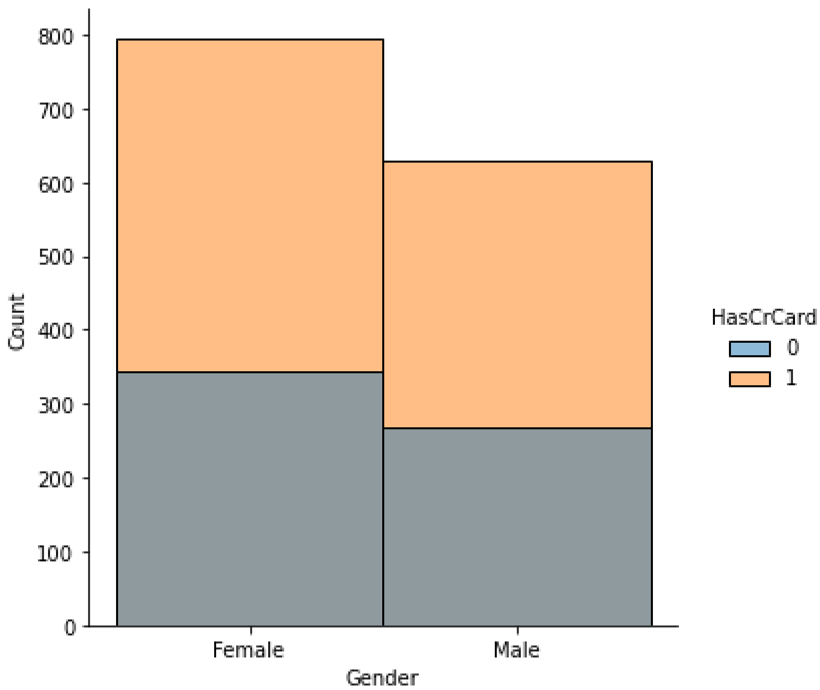
```
plt.figure(figsize=(5, 5))
sns.displot(data=df, x="Gender", hue="HasCrCard")
plt.figure(figsize=(5, 5))  # Create a new figure
sns.displot(data=df[df["Exited"] == 1], x="Gender", hue="HasCrCard")
plt.figure(figsize=(5, 5))
sns.displot(data=df[df["Exited"] == 1], x="Gender", hue="IsActiveMember
```

Out[114]: <seaborn.axisgrid.FacetGrid at 0x27606260c70>

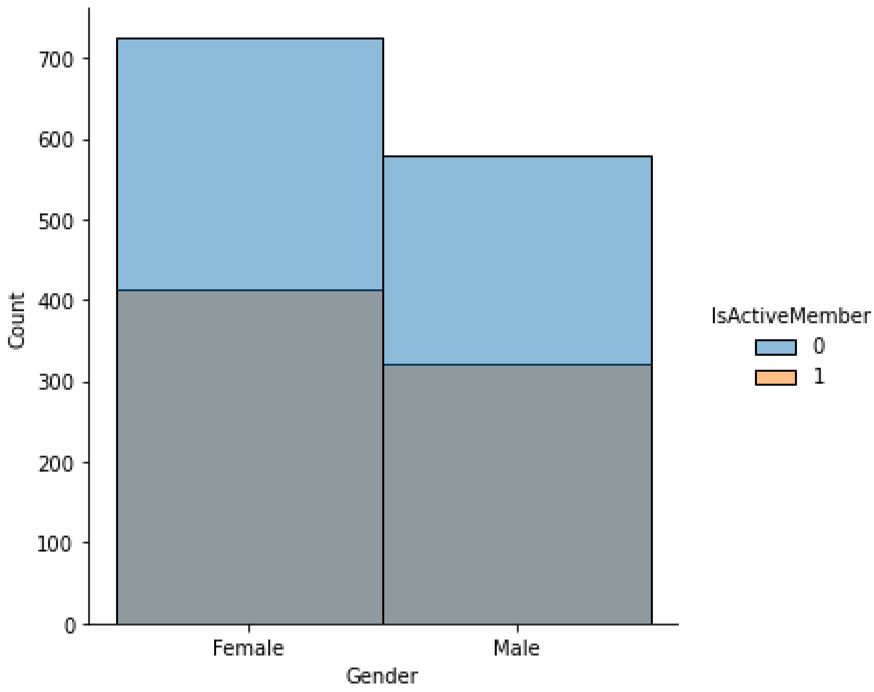
<Figure size 360x360 with 0 Axes>



<Figure size 360x360 with 0 Axes>



<Figure size 360x360 with 0 Axes>

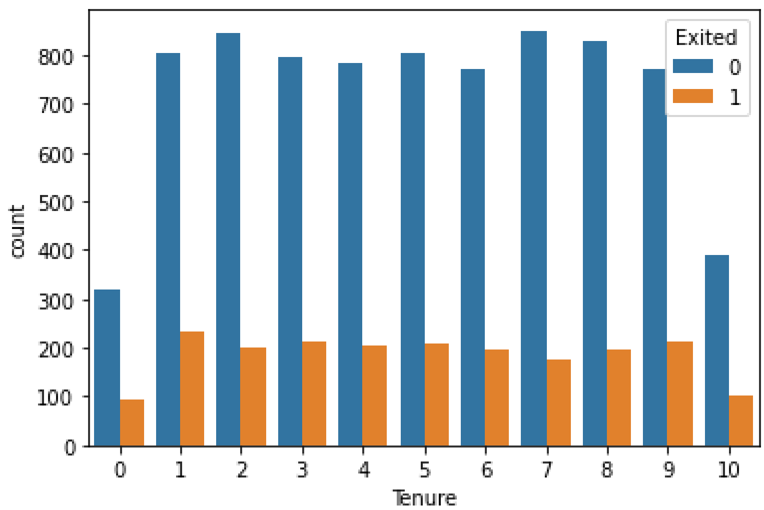


Descriptive analysis

Churn rate for different type of tenures

In [115]:
```
sns.countplot(x=df['Tenure'],hue= df['Exited'])
```

Out[115]: <AxesSubplot:xlabel='Tenure', ylabel='count'>



In [118]:
```
pd.crosstab(columns = df['Tenure'],index= df['Exited'],margins = True)
```

Out[118]:

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 [120]:
```python
churn_data = pd.crosstab(columns = df['Tenure'],index= df['Exited'],nor
churn_data
```

Out[120]:

Tenure	0	1	2	3	4	5	6	7	
Exited									
0	0.769976	0.775845	0.808206	0.7889	0.794742	0.793478	0.797311	0.827821	0
1	0.230024	0.224155	0.191794	0.2111	0.205258	0.206522	0.202689	0.172179	0

In [121]:
```python
churn_data[1:2].reset_index()
```

Out[121]:

	Tenure	Exited	0	1	2	3	4	5	6	
0		1	0.230024	0.224155	0.191794	0.2111	0.205258	0.206522	0.202689	0.17

In [123]:
```python
#sns.barplot(churn_data[1:2].reset_index().drop('Exited',axis = 1))
```

- 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

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.