

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

```
data=pd.read_csv(r"C:\Users\CTTC\Downloads\Churn_Modelling (2).csv")
```

```
data
```



	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	1
3	4	15701354	Boni	699	France	Female	39	1	
4	5	15737888	Mitchell	850	Spain	Female	43	2	1
...
9995	9996	15606229	Obijaku	771	France	Male	39	5	
9996	9997	15569892	Johnstone	516	France	Male	35	10	
9997	9998	15584532	Liu	709	France	Female	36	7	
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	
9999	10000	15628319	Walker	792	France	Female	28	4	1

10000 rows × 14 columns

```
data.isnull().sum()
```



```
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
dtype: int64
```

```
data.dtypes
```

```
➞ RowNumber      int64
   CustomerId     int64
   Surname        object
   CreditScore     int64
   Geography      object
   Gender         object
   Age            int64
   Tenure         int64
   Balance        float64
   NumOfProducts  int64
   HasCrCard      int64
   IsActiveMember int64
   EstimatedSalary float64
   Exited         int64
   dtype: object
```

```
for i in ['Surname', 'Geography', 'Gender']:
    print(i, ': ', '\n', data[i].unique())
```

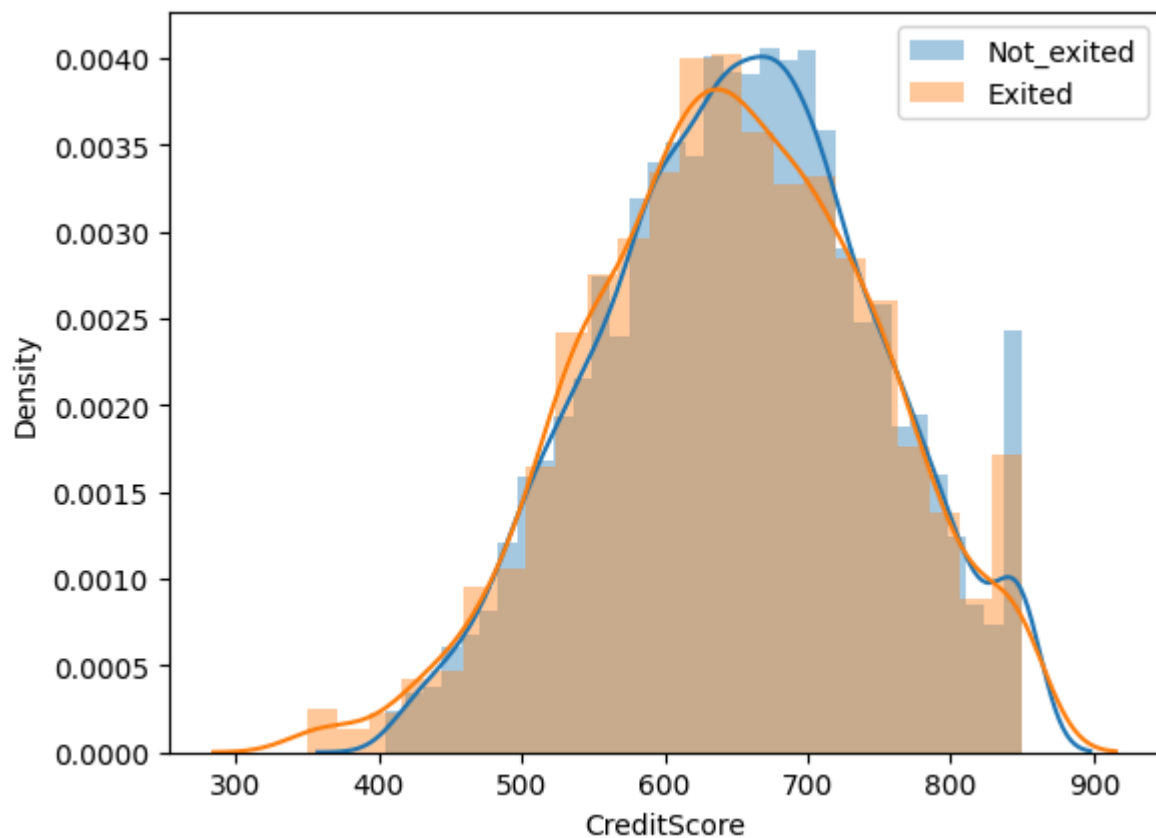
```
➞ Surname :
   ['Hargrave' 'Hill' 'Onio' ... 'Kashiwagi' 'Aldridge' 'Burbidge']
Geography :
   ['France' 'Spain' 'Germany']
Gender :
   ['Female' 'Male']
```

```
data.shape
```

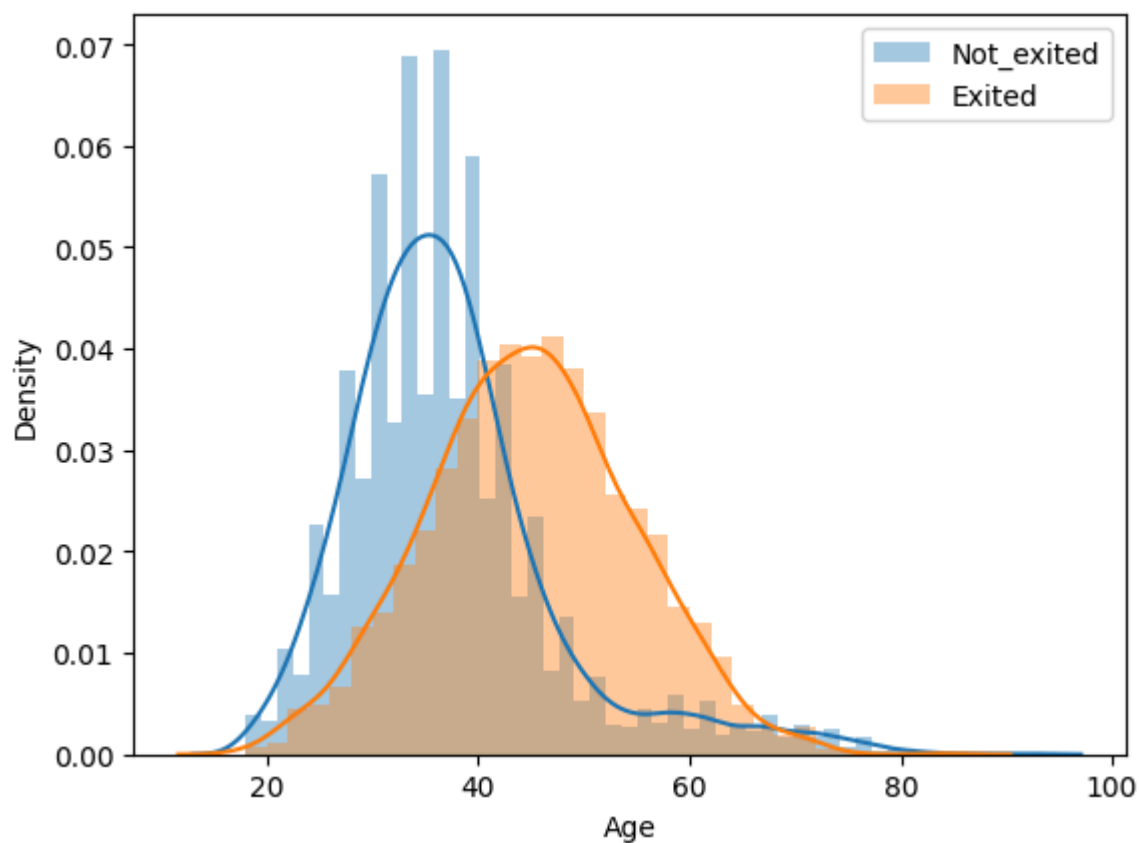
```
➞ (10000, 14)
```

```
import warnings
warnings.filterwarnings('ignore')
```

```
sns.distplot(data.CreditScore[data.Exited==0], label='Not_exited')
sns.distplot(data.CreditScore[data.Exited==1], label='Exited')
plt.legend()
plt.show()
```

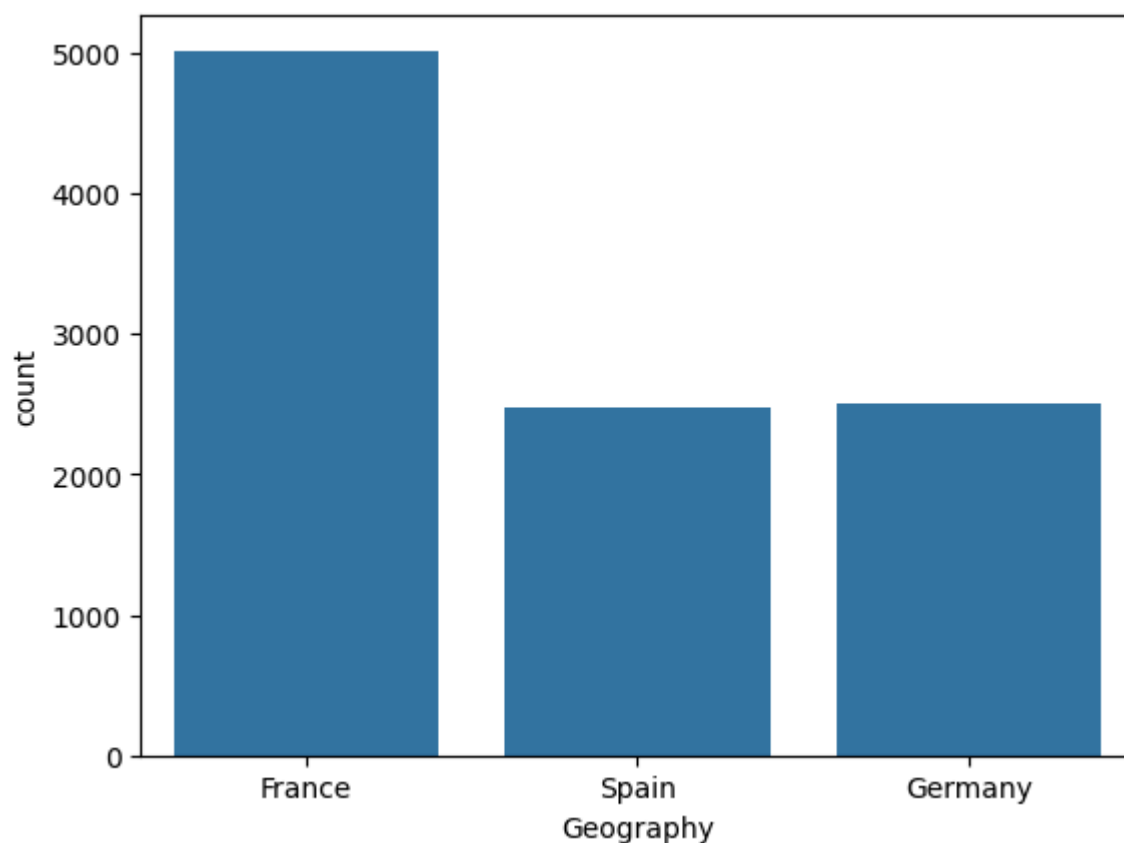


```
sns.distplot(data.Age[data.Exited==0],label='Not_exited')  
sns.distplot(data.Age[data.Exited==1],label='Exited')  
plt.legend()  
plt.show()
```



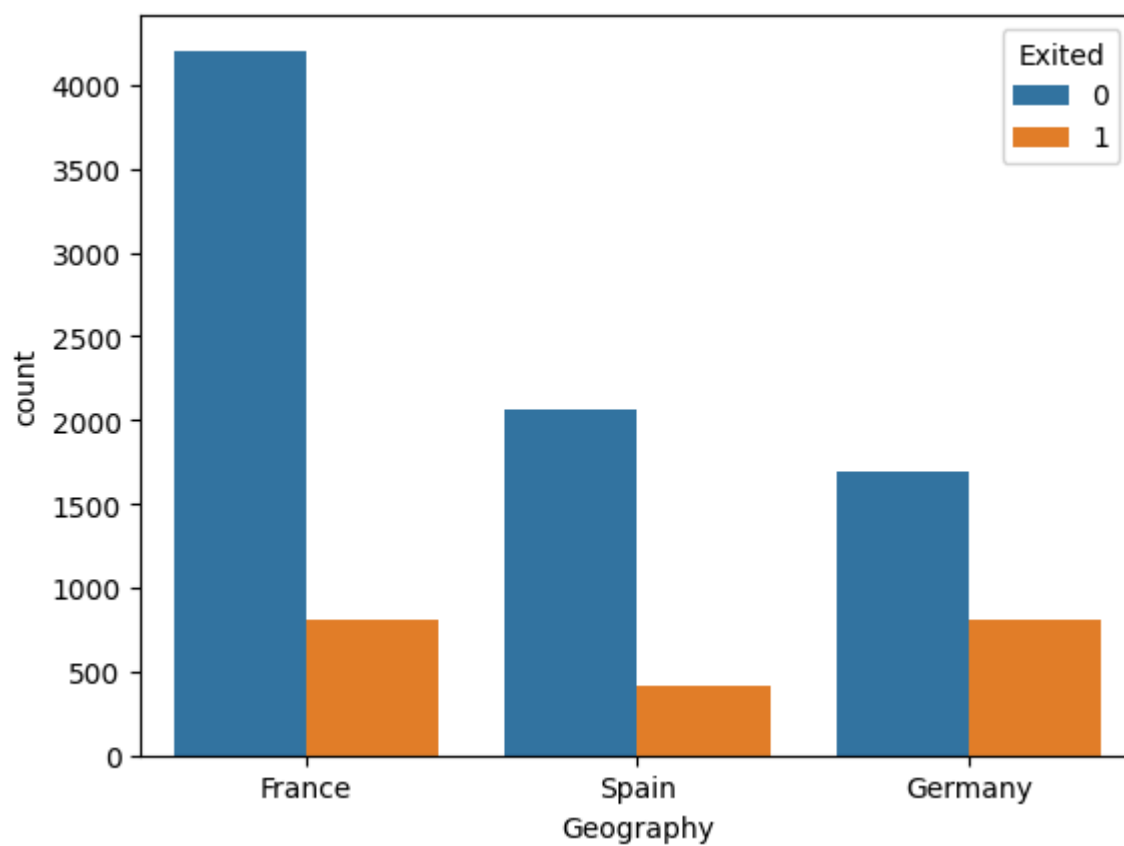
```
sns.countplot(x=data.Geography)
```

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



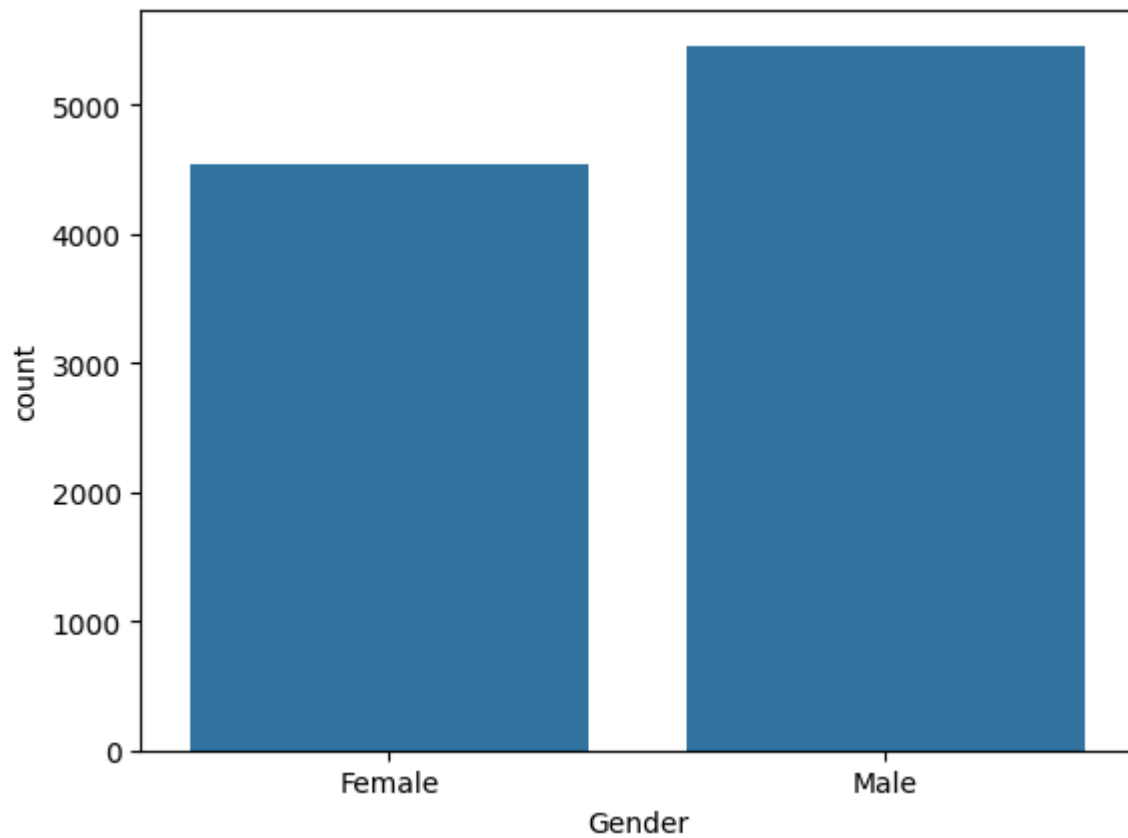
```
sns.countplot(x=data.Geography, hue=data.Exited)
```

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



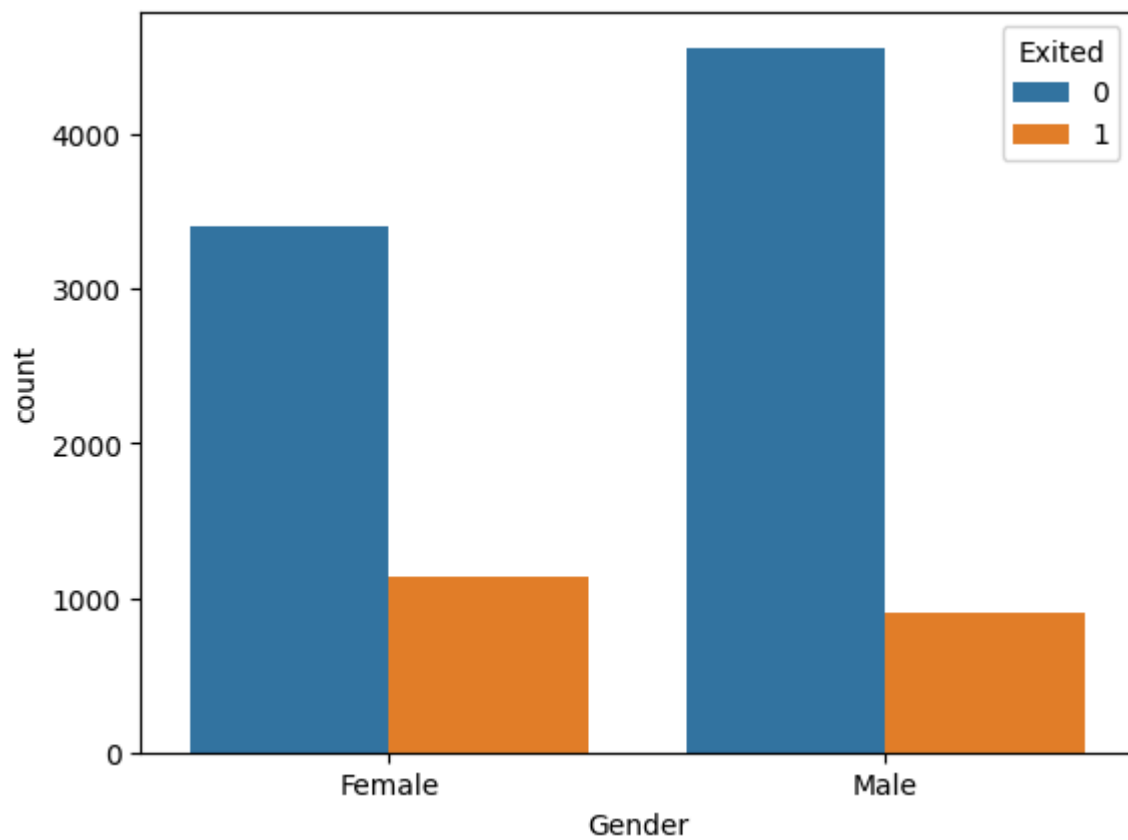
```
sns.countplot(x=data.Gender)
```

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



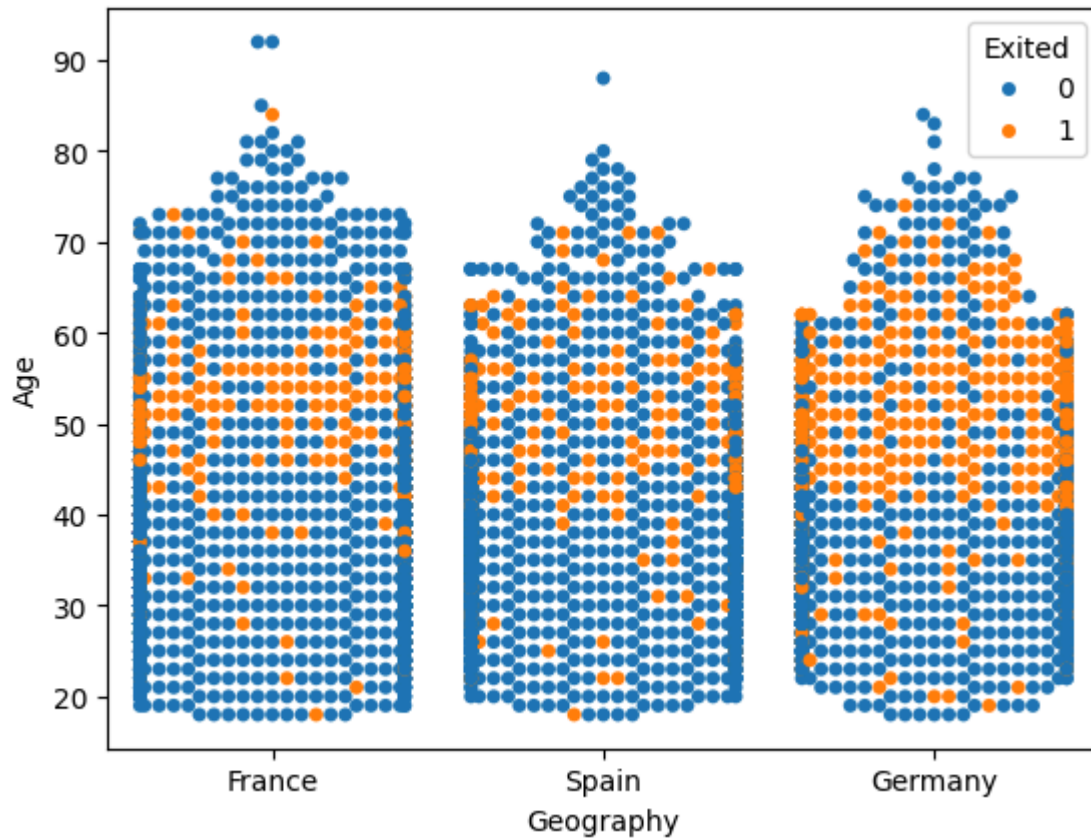
```
sns.countplot(x=data.Gender, hue=data.Exited)
```

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

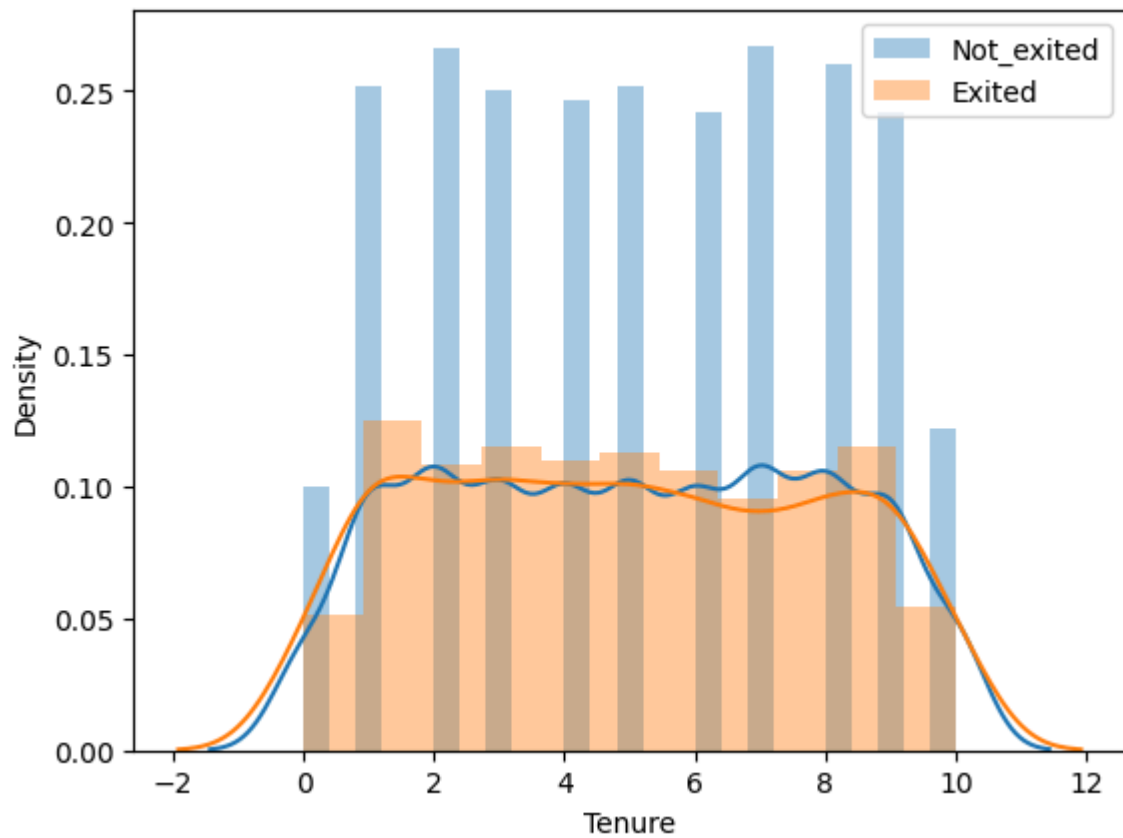


```
sns.swarmplot(x=data.Geography,y=data.Age,hue=data.Exited)
```

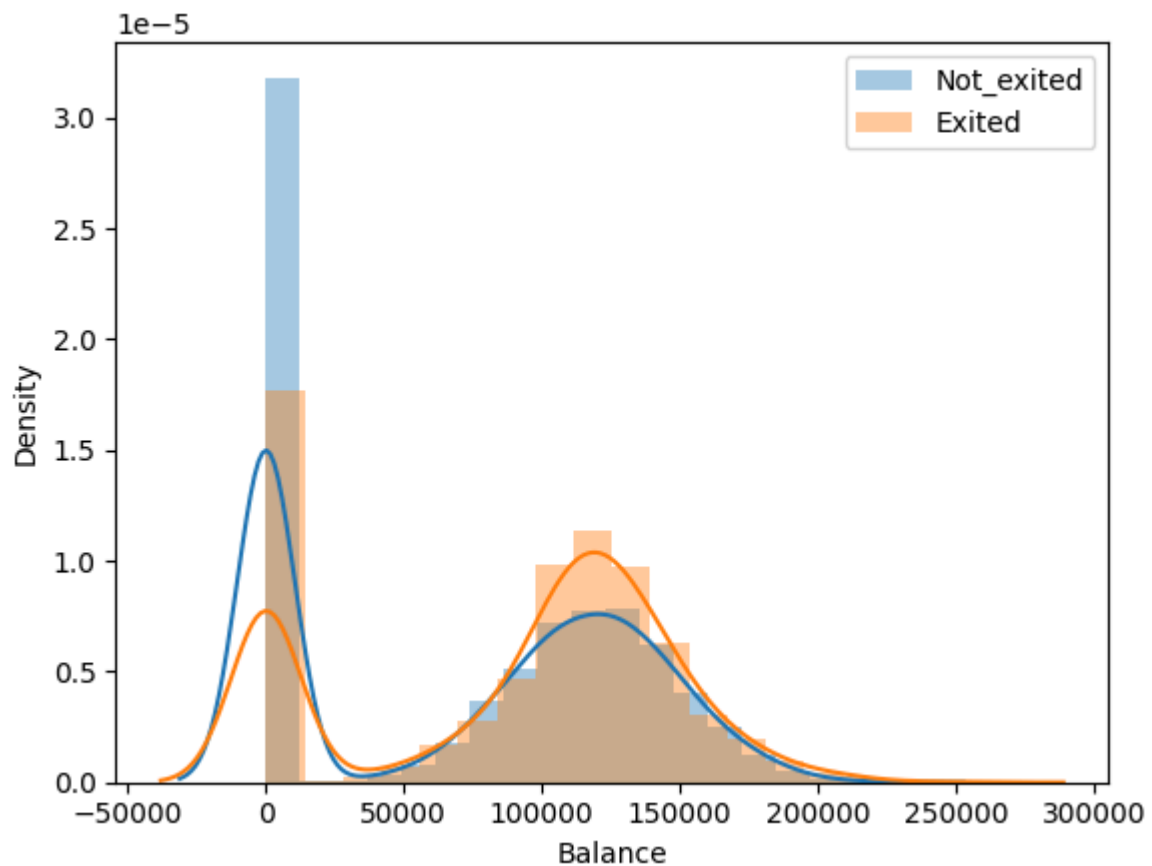
```
<Axes: xlabel='Geography', ylabel='Age'>
```



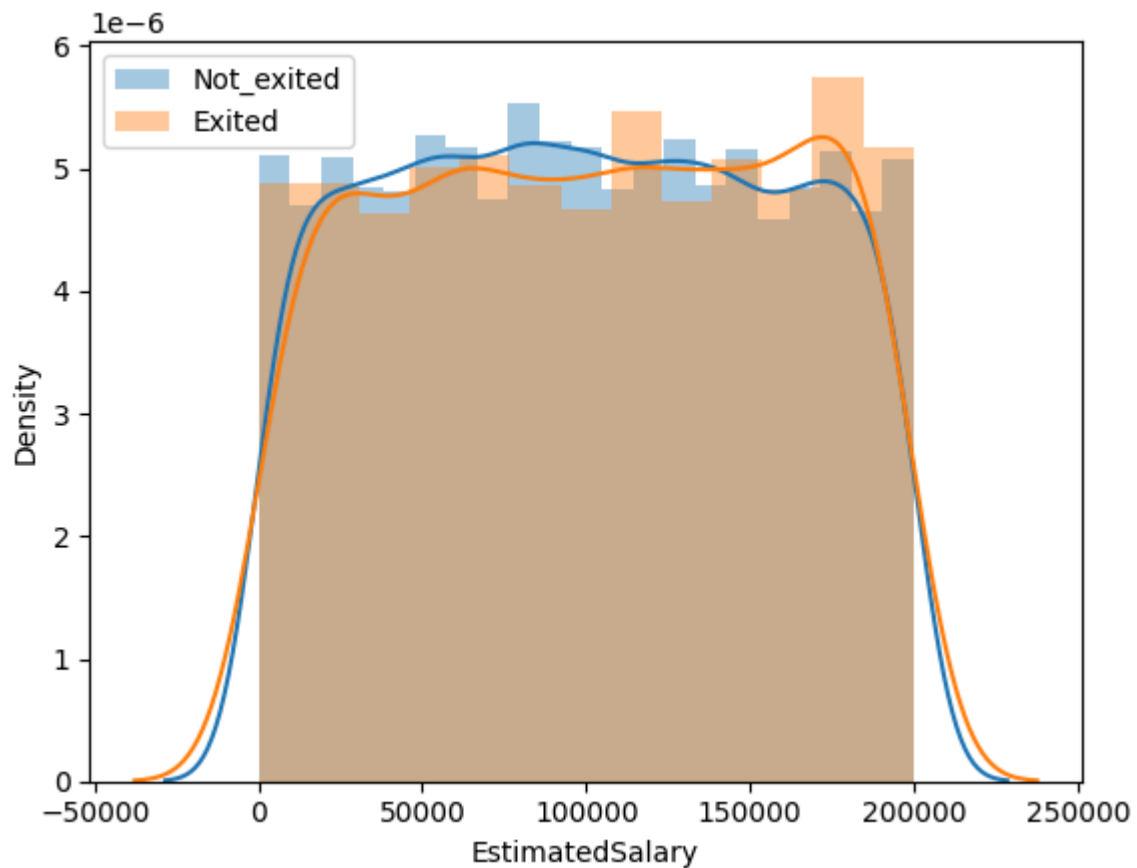
```
sns.distplot(data.Tenure[data.Exited==0],label='Not_exited')  
sns.distplot(data.Tenure[data.Exited==1],label='Exited')  
plt.legend()  
plt.show()
```



```
sns.distplot(data.Balance[data.Exited==0],label='Not_exited')
sns.distplot(data.Balance[data.Exited==1],label='Exited')
plt.legend()
plt.show()
```

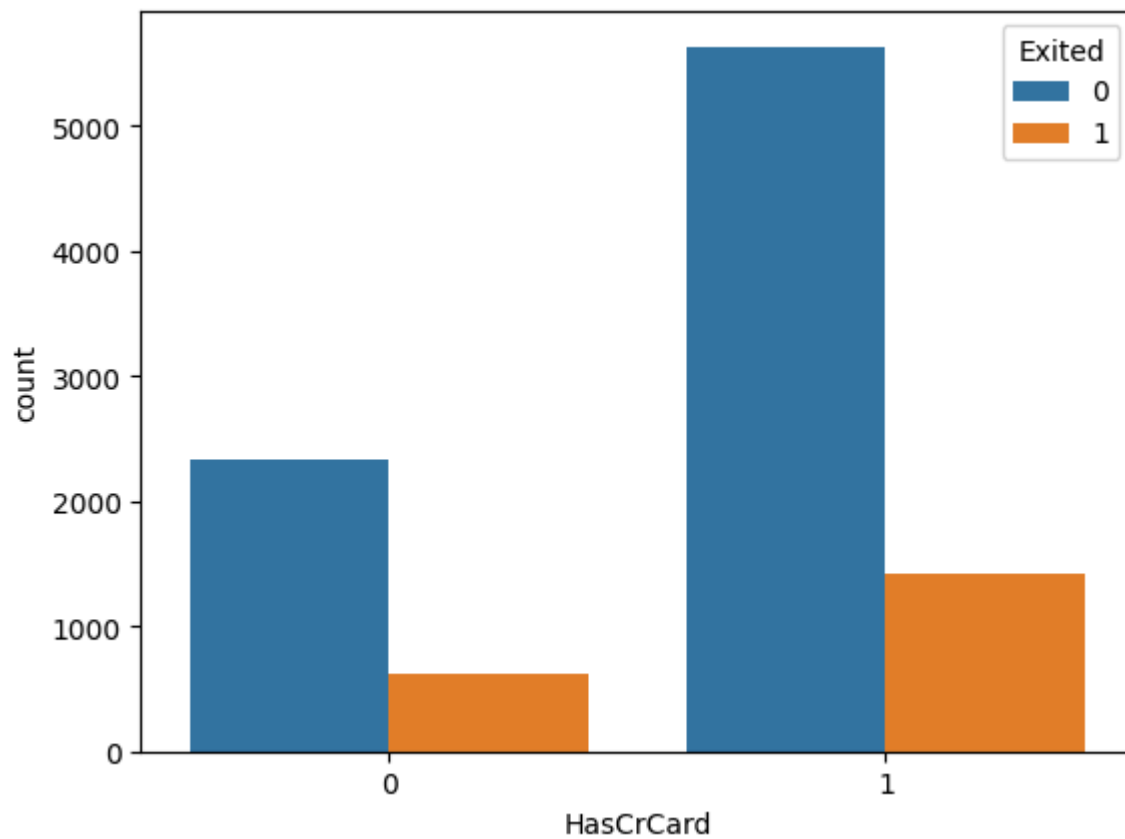


```
sns.distplot(data.EstimatedSalary[data.Exited==0],label='Not_exited')  
sns.distplot(data.EstimatedSalary[data.Exited==1],label='Exited')  
plt.legend()  
plt.show()
```



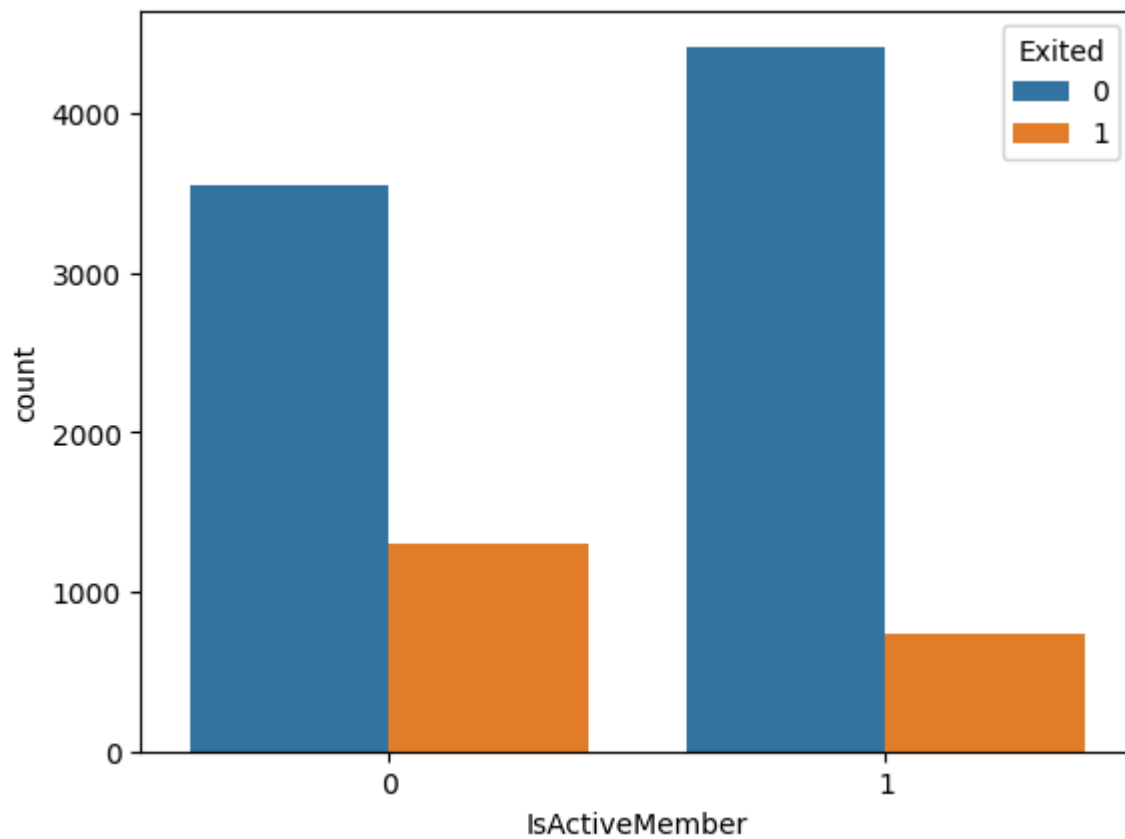
```
sns.countplot(x=data.HasCrCard,hue=data.Exited)
```


 <Axes: xlabel='HasCrCard', ylabel='count'>




```
sns.countplot(x=data.IsActiveMember,hue=data.Exited)
```

 <Axes: xlabel='IsActiveMember', ylabel='count'>



```
data1=data.drop(['Surname','RowNumber','CustomerId'],axis=1)
```

data1




	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard
0	619	France	Female	42	2	0.00	1	1
1	608	Spain	Female	41	1	83807.86	1	0
2	502	France	Female	42	8	159660.80	3	1
3	699	France	Female	39	1	0.00	2	0
4	850	Spain	Female	43	2	125510.82	1	1
...
9995	771	France	Male	39	5	0.00	2	1
9996	516	France	Male	35	10	57369.61	1	1
9997	709	France	Female	36	7	0.00	1	0
9998	772	Germany	Male	42	3	75075.31	2	1
9999	792	France	Female	28	4	130142.79	1	1

10000 rows × 11 columns

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
data1.Geography=le.fit_transform(data1.Geography)
le1=LabelEncoder()
data1.Gender=le1.fit_transform(data1.Gender)
```

data1



	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard
0	619	0	0	42	2	0.00	1	1
1	608	2	0	41	1	83807.86	1	0
2	502	0	0	42	8	159660.80	3	1
3	699	0	0	39	1	0.00	2	0