

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

df = pd.read_csv('Churn_Modelling.csv')
df.head()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age
0	1	15634602	Hargrave	619	France	Female	42
1	2	15647311	Hill	608	Spain	Female	41
2	3	15619304	Onio	502	France	Female	42
3	4	15701354	Boni	699	France	Female	39
4	5	15737888	Mitchell	850	Spain	Female	43

	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	
0	2	0.00	1	1	1	
1	1	83807.86	1	0	1	
2	8	159660.80	3	1	0	
3	1	0.00	2	0	0	
4	2	125510.82	1	1	1	

	EstimatedSalary	Exited
0	101348.88	1
1	112542.58	0
2	113931.57	1
3	93826.63	0
4	79084.10	0

```
df.info()
```

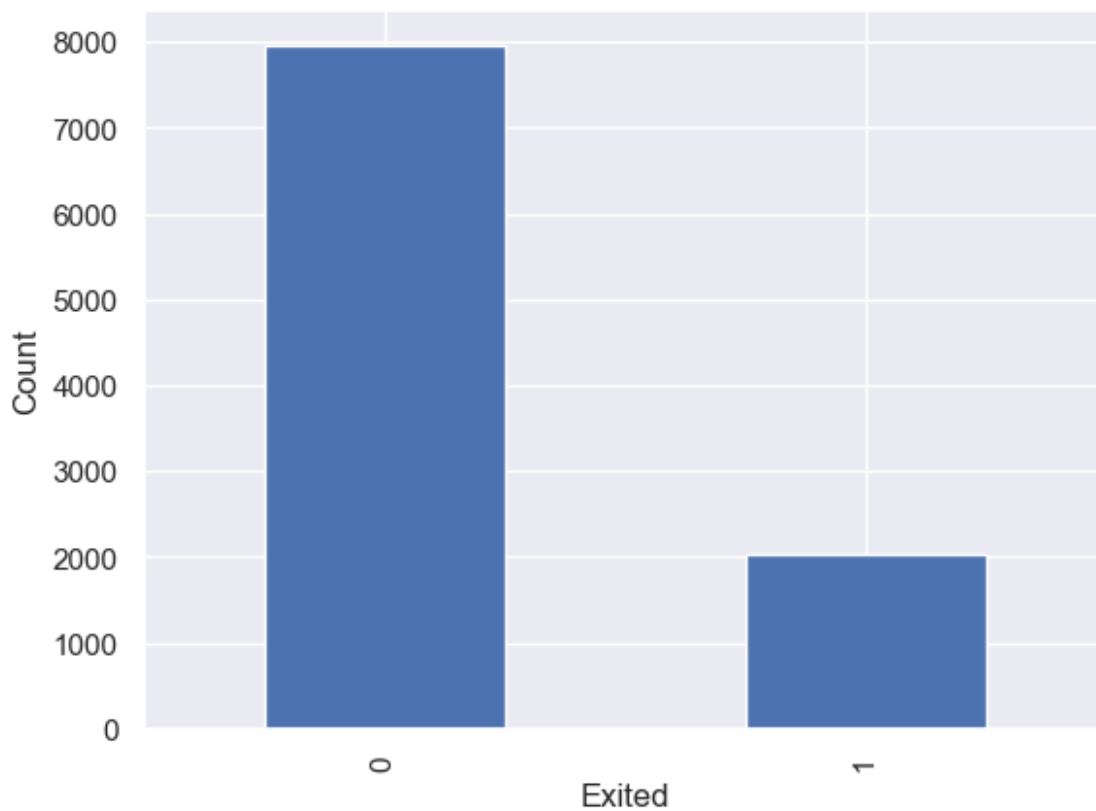
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---  -
0   RowNumber              10000 non-null  int64
1   CustomerId             10000 non-null  int64
2   Surname                10000 non-null  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
dtypes: float64(2), int64(9), object(3)
memory usage: 1.1+ MB

plt.xlabel('Exited')
plt.ylabel('Count')
df['Exited'].value_counts().plot.bar()
plt.show()

```



```

df['Geography'].value_counts()

Geography
France      5014
Germany     2509
Spain       2477
Name: count, dtype: int64

df =
pd.concat([df, pd.get_dummies(df['Geography'], prefix='Geo')], axis=1)

```

```
df = pd.concat([df,pd.get_dummies(df['Gender'])],axis=1)
df.drop(columns=['RowNumber','CustomerId','Surname','Geography','Gender'],inplace=True)
```

```
df.head()
```

	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	\
0	619	42	2	0.00	1	1	
1	608	41	1	83807.86	1	0	
2	502	42	8	159660.80	3	1	
3	699	39	1	0.00	2	0	
4	850	43	2	125510.82	1	1	

	IsActiveMember	EstimatedSalary	Exited	Geo_France	Geo_Germany	\
0	1	101348.88	1	True	False	
1	1	112542.58	0	False	False	
2	0	113931.57	1	True	False	
3	0	93826.63	0	True	False	
4	1	79084.10	0	False	False	

	Geo_Spain	Female	Male
0	False	True	False
1	True	True	False
2	False	True	False
3	False	True	False
4	True	True	False

```
y = df['Exited'].values
x = df.loc[:,df.columns != 'Exited'].values

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test =
train_test_split(x,y,random_state=20,test_size=0.25)
```

```
from sklearn.metrics import accuracy_score
from sklearn.neural_network import MLPClassifier

model = MLPClassifier(random_state=1, max_iter=300)
model.fit(x_train, y_train)
```

```
MLPClassifier(max_iter=300, random_state=1)
```

```
y_pred = model.predict(x_test)
```

```
from sklearn.metrics import confusion_matrix,ConfusionMatrixDisplay
accuracy_score(y_test,y_pred)
```

```
0.8072
```

```
cm = confusion_matrix(y_test,y_pred)
display = ConfusionMatrixDisplay(cm)
display.plot()
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at
0x1893f6f1a60>
```

