In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
data=pd.read_csv('Churn_Modelling.csv')
data
```

Out[2]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfI
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	
2	3	15619304	Onio	502	France	Female	42	8	159660.80	
3	4	15701354	Boni	699	France	Female	39	1	0.00	
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	
...	...	...	...	...	...	...	...	...	...	
9995	9996	15606229	Obijiaku	771	France	Male	39	5	0.00	
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	

10000 rows × 14 columns

In [3]:
```
data=pd.get_dummies(columns=['Geography','Gender'],data=data)
data
```

Out[3]:

	RowNumber	CustomerId	Surname	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard
0	1	15634602	Hargrave	619	42	2	0.00	1	1
1	2	15647311	Hill	608	41	1	83807.86	1	0
2	3	15619304	Onio	502	42	8	159660.80	3	1
3	4	15701354	Boni	699	39	1	0.00	2	0
4	5	15737888	Mitchell	850	43	2	125510.82	1	1
...	...	...	...	...	...	...	...	...	...
9995	9996	15606229	Obijiaku	771	39	5	0.00	2	1
9996	9997	15569892	Johnstone	516	35	10	57369.61	1	1
9997	9998	15584532	Liu	709	36	7	0.00	1	0
9998	9999	15682355	Sabbatini	772	42	3	75075.31	2	1
9999	10000	15628319	Walker	792	28	4	130142.79	1	1

10000 rows × 17 columns

In [4]:
```
X = data.drop(columns=['CustomerId', 'Exited','RowNumber','Surname'])  # Exclude unneces
y = data['Exited']
```

In [5]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42
```

In [6]:
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

In [16]:
```
from sklearn.linear_model import LogisticRegression
logistic_regression = LogisticRegression()
# Train the model
logistic_regression.fit(X_train, y_train)
```

Out[16]:
```
▾ LogisticRegression

LogisticRegression()
```

In [15]:
```
y_pred = logistic_regression.predict(X_test)

# Evaluate the model
from sklearn.metrics import accuracy_score , confusion_matrix
```

In [9]:
```
accuracy = accuracy_score(y_test, y_pred)
accuracy
```

Out[9]:
```
0.811
```

In [20]:
```
cons = confusion_matrix(y_test, y_pred)
cons
```

Out[20]:
```
array([[1543,   64],
       [ 314,   79]], dtype=int64)
```

In [ ]:
```
```