

```
In [1]: 1 import pandas as pd
        2 df=pd.read_csv("Churn_Modelling.csv")
```

```
In [2]: 1 df
```

Out[2]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance
	0	1	15634602	Hargrave	619	France	Female	42	2
	1	2	15647311	Hill	608	Spain	Female	41	83807.86
	2	3	15619304	Onio	502	France	Female	42	159660.80
	3	4	15701354	Boni	699	France	Female	39	0.00
	4	5	15737888	Mitchell	850	Spain	Female	43	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

0000 rows × 14 columns



```
In [5]: 1 x=df[['CreditScore','Age','Tenure','Balance','NumOfProducts','HasCrCard']]
        2 x
```

Out[5]:

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

10000 rows × 8 columns

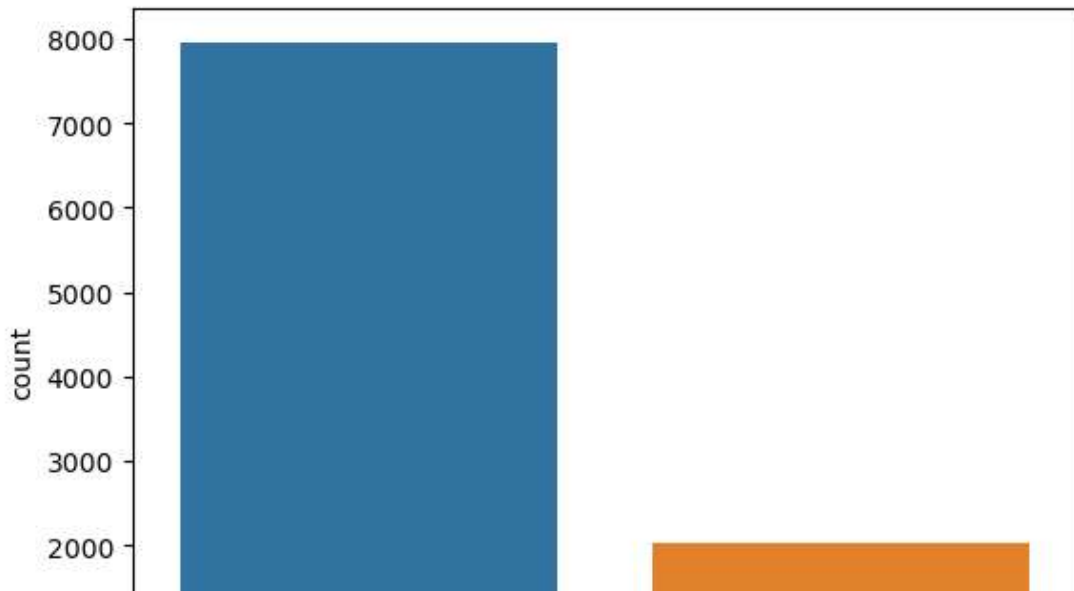


```
In [8]: 1 y=df['Exited']  
2 y.shape
```

Out[8]: (10000,)

```
In [10]: 1 import seaborn as sns  
2 sns.countplot(x=y)
```

Out[10]: <AxesSubplot:xlabel='Exited', ylabel='count'>



```
In [12]: 1 y.value_counts()
```

Out[12]: 0 7963
1 2037
Name: Exited, dtype: int64

```
In [41]: 1 !pip install imbalanced-learn
```

Collecting imbalanced-learn

Using cached imbalanced_learn-0.11.0-py3-none-any.whl (235 kB)

Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\vishw\anaconda3\lib\site-packages (from imbalanced-learn) (2.2.0)

Requirement already satisfied: numpy>=1.17.3 in c:\users\vishw\anaconda3\lib\site-packages (from imbalanced-learn) (1.24.4)

Collecting joblib>=1.1.1

Using cached joblib-1.3.2-py3-none-any.whl (302 kB)

Requirement already satisfied: scikit-learn>=1.0.2 in c:\users\vishw\anaconda3\lib\site-packages (from imbalanced-learn) (1.0.2)

Requirement already satisfied: scipy>=1.5.0 in c:\users\vishw\anaconda3\lib\site-packages (from imbalanced-learn) (1.9.1)

Installing collected packages: joblib, imbalanced-learn

Attempting uninstall: joblib

Found existing installation: joblib 1.1.0

Uninstalling joblib-1.1.0:

Successfully uninstalled joblib-1.1.0

Successfully installed imbalanced-learn-0.11.0 joblib-1.3.2

```
In [42]: 1 from imblearn.over_sampling import RandomOverSampler
```

```
In [44]: 1 ros=RandomOverSampler(random_state=0)
```

```
In [45]: 1 x_res,y_res=ros.fit_resample(x,y)
```

```
In [46]: 1 y_res.value_counts()
```

```
Out[46]: 1    7963  
        0    7963  
        Name: Exited, dtype: int64
```

```
In [47]: 1 #Normalize  
        2 from sklearn.preprocessing import StandardScaler  
        3 scaler=StandardScaler()  
        4 x_scaled=scaler.fit_transform(x_res)
```

```
In [48]: 1 x_scaled
```

```
Out[48]: array([[ -0.29877723,  0.08418894, -1.01840607, ...,  0.6512495 ,  
                  1.08223556,  0.00817382],  
               [ -0.4103938 , -0.01032629, -1.36135608, ..., -1.53550983,  
                  1.08223556,  0.20261687],  
               [-1.48597169,  0.08418894,  1.03929402, ...,  0.6512495 ,  
                 -0.92401325,  0.22674468],  
               ...,  
               [-0.84671313,  1.02934128,  0.01044398, ...,  0.6512495 ,  
                 -0.92401325,  1.28878188],  
               [-0.96847667,  0.65128034, -0.67545605, ..., -1.53550983,  
                  1.08223556, -1.21851316],  
               [-1.5874413 ,  0.74579558,  1.03929402, ...,  0.6512495 ,  
                  1.08223556,  1.42417217]])
```

```
In [49]: 1 #cross validation  
        2 from sklearn.model_selection import train_test_split  
        3
```

```
In [50]: 1 x_train,x_test,y_train,y_test=train_test_split(x_scaled,y_res,random_st
```

```
In [51]: 1 x_res.shape
```

```
Out[51]: (15926, 8)
```

```
In [52]: 1 x_test.shape
```

```
Out[52]: (3982, 8)
```

```
In [53]: 1 x_train.shape
```

```
Out[53]: (11944, 8)
```

```
In [54]: 1 from sklearn.neural_network import MLPClassifier
```

```
In [55]: 1 ann=MLPClassifier(hidden_layer_sizes=(100,100,100),random_state=0,max_iter=1000)
```

```
In [56]: 1 ann.fit(x_train,y_train)
```

```
C:\Users\vishw\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:692: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and the optimization hasn't converged yet.  
warnings.warn(
```

```
Out[56]: MLPClassifier(hidden_layer_sizes=(100, 100, 100), max_iter=100, random_state=0)
```

```
In [57]: 1 y_pred=ann.predict(x_test)
```

```
In [58]: 1 y_pred
```

```
Out[58]: array([1, 1, 1, ..., 1, 1, 1], dtype=int64)
```

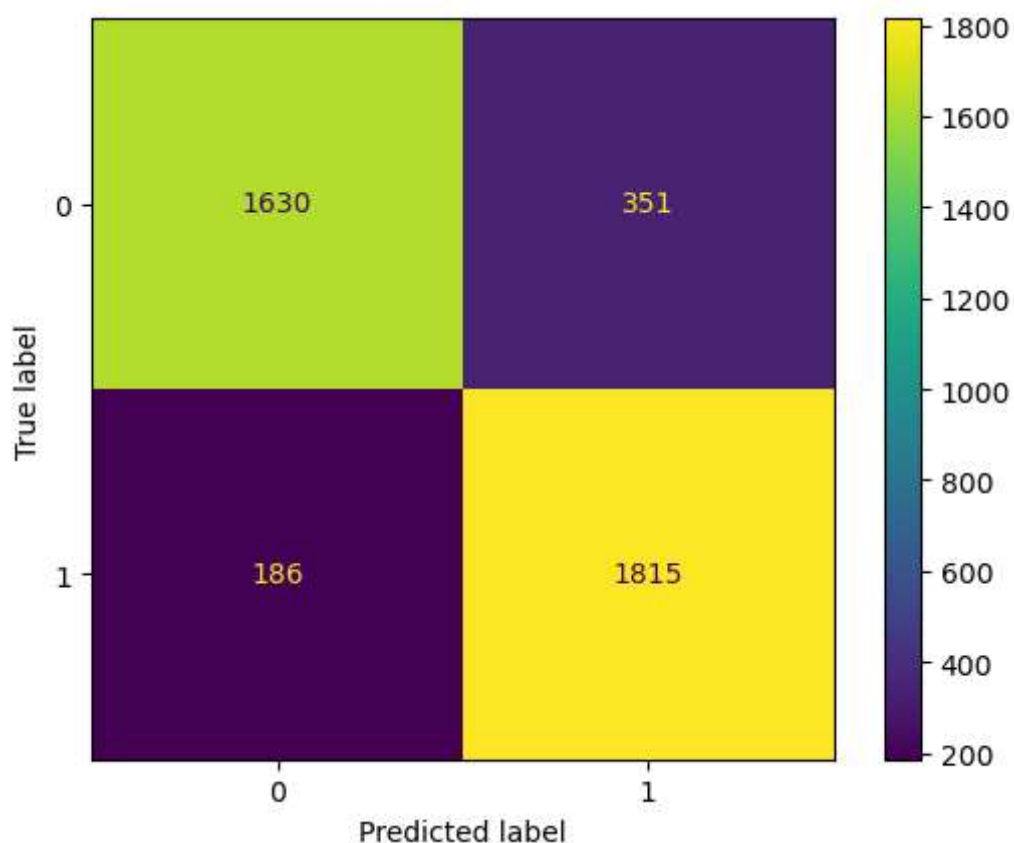
```
In [59]: 1 from sklearn.metrics import ConfusionMatrixDisplay, classification_report  
2 from sklearn.metrics import accuracy_score
```

```
In [60]: 1 y_test.value_counts()
```

```
Out[60]: 1    2001  
0    1981  
Name: Exited, dtype: int64
```

In [61]: 1 ConfusionMatrixDisplay.from_predictions(y_test,y_pred)

Out[61]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x1cec9ca61f0>



In [62]: 1 print(classification_report(y_test,y_pred))

	precision	recall	f1-score	support
0	0.90	0.82	0.86	1981
1	0.84	0.91	0.87	2001
accuracy			0.87	3982
macro avg	0.87	0.86	0.86	3982
weighted avg	0.87	0.87	0.86	3982

In []: 1