

ML_Pract_3

October 31, 2023

Aim: Given a bank customer, build a neural network-based classifier that can determine whether they will leave or not in the next 6 months. Dataset Description: The case study is from an open-source dataset from Kaggle. The dataset contains 10,000 sample points with 14 distinct features such as CustomerId, CreditScore, Geography, Gender, Age, Tenure, Balance, etc.

Name: Yash Ghorpade

Div: BE-A

Roll No: B211046

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

```
[2]: df = pd.read_csv("Churn_Modelling.csv")
```

```
[3]: df.describe()
```

```
[3]:
```

	RowNumber	CustomerId	CreditScore	Age	Tenure \
count	10000.00000	1.000000e+04	10000.000000	10000.000000	10000.000000
mean	5000.50000	1.569094e+07	650.528800	38.921800	5.012800
std	2886.89568	7.193619e+04	96.653299	10.487806	2.892174
min	1.00000	1.556570e+07	350.000000	18.000000	0.000000
25%	2500.75000	1.562853e+07	584.000000	32.000000	3.000000
50%	5000.50000	1.569074e+07	652.000000	37.000000	5.000000
75%	7500.25000	1.575323e+07	718.000000	44.000000	7.000000
max	10000.00000	1.581569e+07	850.000000	92.000000	10.000000

	Balance	NumOfProducts	HasCrCard	IsActiveMember \
count	10000.000000	10000.000000	10000.00000	10000.000000
mean	76485.889288	1.530200	0.70550	0.515100
std	62397.405202	0.581654	0.45584	0.499797
min	0.000000	1.000000	0.00000	0.000000
25%	0.000000	1.000000	0.00000	0.000000
50%	97198.540000	1.000000	1.00000	1.000000
75%	127644.240000	2.000000	1.00000	1.000000
max	250898.090000	4.000000	1.00000	1.000000

	EstimatedSalary	Exited
count	10000.000000	10000.000000
mean	100090.239881	0.203700
std	57510.492818	0.402769
min	11.580000	0.000000
25%	51002.110000	0.000000
50%	100193.915000	0.000000
75%	149388.247500	0.000000

max 199992.480000 1.000000

```
[4]: df.head()
```

```
[4]:
```

	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	

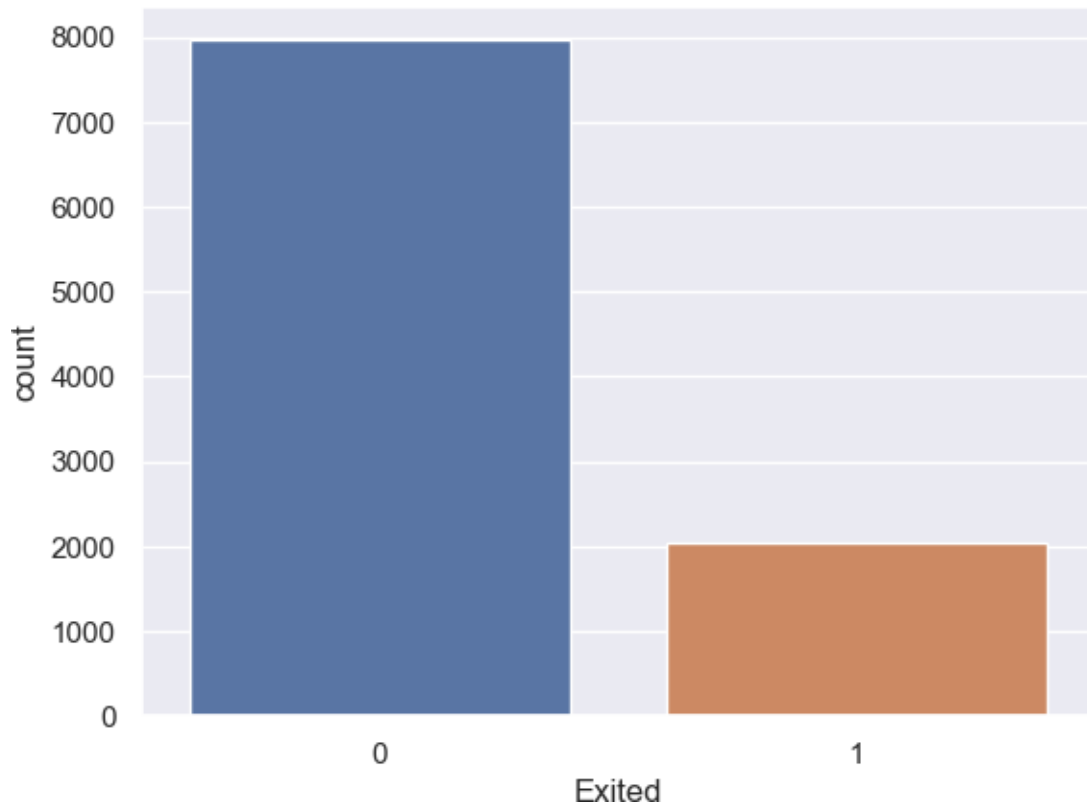
	EstimatedSalary	Exited
0	101348.88	1
1	112542.58	0
2	113931.57	1
3	93826.63	0
4	79084.10	0

```
[5]: x = df.drop(["Surname", "Geography", "RowNumber", "Gender", "Exited"], axis = 1)
```

```
[6]: y = df["Exited"]
```

```
[7]: sns.countplot(x=y)
```

```
[7]: <Axes: xlabel='Exited', ylabel='count'>
```



1 Scaling

```
[8]: from sklearn.preprocessing import StandardScaler  
     scalar = StandardScaler()  
     x_scaled = scalar.fit_transform(x)
```

```
[9]: x_scaled
```

```
[9]: array([[ -0.78321342, -0.32622142,  0.29351742, ...,  0.64609167,  
            0.97024255,  0.02188649],  
        [ -0.60653412, -0.44003595,  0.19816383, ..., -1.54776799,  
            0.97024255,  0.21653375],  
        [ -0.99588476, -1.53679418,  0.29351742, ...,  0.64609167,  
        -1.03067011,  0.2406869 ],  
        ...,  
        [ -1.47928179,  0.60498839, -0.27860412, ..., -1.54776799,  
            0.97024255, -1.00864308],  
        [ -0.11935577,  1.25683526,  0.29351742, ...,  0.64609167,  
        -1.03067011, -0.12523071],  
        [ -0.87055909,  1.46377078, -1.04143285, ...,  0.64609167,
```

```
-1.03067011, -1.07636976]])
```

2 Cross validation

```
[10]: from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x_scaled,y,random_state = 46,
test_size = 0.46)
```

```
[11]: x_train.shape
```

```
[11]: (5400, 9)
```

```
[12]: x_test.shape
```

```
[12]: (4600, 9)
```

```
[13]: from sklearn.neural_network import MLPClassifier
ann = MLPClassifier(hidden_layer_sizes=(100,100,100),random_state=2,max_iter=100,activation_
= "relu")
```

```
[14]: ann.fit(x_train,y_train)
```

```
C:\ProgramData\anaconda3\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:684:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
warnings.warn(
```

```
[14]: MLPClassifier(hidden_layer_sizes=(100, 100, 100), max_iter=100, random_state=2)
```

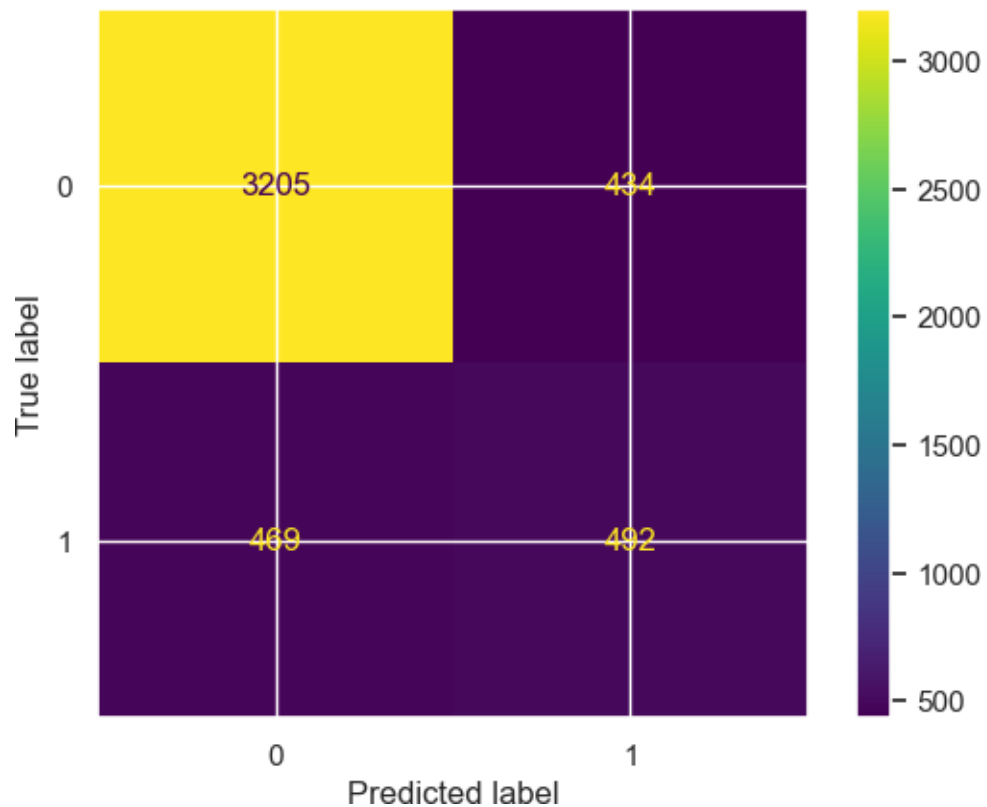
```
[15]: y_pred = ann.predict(x_test)
```

```
[16]: y_pred
```

```
[16]: array([0, 1, 0, ..., 0, 1, 0], dtype=int64)
```

```
[17]: from sklearn.metrics import
ConfusionMatrixDisplay,accuracy_score,classification_report
ConfusionMatrixDisplay.from_predictions(y_test,y_pred)
```

```
[17]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x1f68f02ec20>
```



```
[18]: accuracy_score(y_test,y_pred)
```

```
[18]: 0.803695652173913
```

```
[19]: print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.87	0.88	0.88	3639
1	0.53	0.51	0.52	961
accuracy			0.80	4600
macro avg	0.70	0.70	0.70	4600
weighted avg	0.80	0.80	0.80	4600

```
[ ]:
```