

ML_3

November 11, 2025

```
[1]: import pandas as pd  
import numpy as np
```

```
[2]: df = pd.read_csv(r"C:  
    ↴\Users\suraj\OneDrive\Desktop\LP3-master\ML\datasets\Churn_Modelling.csv")
```

```
[3]: 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
```

```
[4]: df.describe()  
df.info()  
df.shape  
df.isna()  
df.isna().sum()
```

```
<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 |
| 1 | CustomerId | 10000 | non-null |
| 2 | Surname | 10000 | non-null |
| 3 | CreditScore | 10000 | non-null |
| 4 | Geography | 10000 | non-null |
| 5 | Gender | 10000 | non-null |
| 6 | Age | 10000 | non-null |
| 7 | Tenure | 10000 | non-null |
| 8 | Balance | 10000 | non-null |
| 9 | NumOfProducts | 10000 | non-null |
| 10 | HasCrCard | 10000 | non-null |
| 11 | IsActiveMember | 10000 | non-null |
| 12 | EstimatedSalary | 10000 | non-null |
| 13 | Exited | 10000 | non-null |

dtypes: float64(2), int64(9), object(3)
memory usage: 1.1+ MB

```
[4]: 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
```

```
[5]: df = df.drop(['RowNumber', 'CustomerId', 'Surname'], axis=1)
```

```
[6]: df.shape
```

```
[6]: (10000, 11)
```

```
[7]: X = df.drop('Exited', axis=1)
y=df['Exited']
```

```
[8]: X = pd.get_dummies(X, drop_first=True)
```

```
[9]: print(X.dtypes)
```

```
CreditScore           int64
Age                  int64
Tenure               int64
Balance              float64
NumOfProducts        int64
HasCrCard            int64
IsActiveMember       int64
EstimatedSalary      float64
Geography_Germany   bool
Geography_Spain      bool
Gender_Male          bool
dtype: object
```

```
[10]: X=X.astype(int)
print(X.dtypes)
```

```
CreditScore           int32
Age                  int32
Tenure               int32
Balance              int32
NumOfProducts        int32
HasCrCard            int32
IsActiveMember       int32
EstimatedSalary      int32
Geography_Germany   int32
Geography_Spain      int32
Gender_Male          int32
dtype: object
```

```
[11]: #divide into training and testing set
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)
#3 - Normalize and test data
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
#4 Build Neural NETWORK
from tensorflow.keras.models import Sequential #represent a linear stack of
                                                #layers meaning
                                                #each layer feeds its output to next one

from tensorflow.keras.layers import Dense, Dropout #every neuron in
                                                #the current layer connects to every neuron in next layer
```

```

#initialize the model
model = Sequential()

# add hidden layers
model.add(Dense(units=16, activation='relu'))

model.add(Dense(units=8, activation='relu'))

model.add(Dense(units=1, activation='sigmoid'))

model.compile(optimizer='adam', loss='binary_crossentropy',
              metrics=['accuracy'])

history = model.fit(X_train, y_train, epochs=50, batch_size=25, verbose=1)

```

ModuleNotFoundError Traceback (most recent call last)

Cell In[11], line 13

```

11 X_test = scaler.transform(X_test)
12 #4 Build Neural NETWORK
---> 13 from tensorflow.keras.models import Sequential #represent a linear stack
      ↵of layers meaning
     14 #each layer feeds its output to next one
     15 from tensorflow.keras.layers import Dense, Dropout #every neuron in

```

ModuleNotFoundError: No module named 'tensorflow'

```

[12]: from sklearn.metrics import accuracy_score, confusion_matrix

y_pred = model.predict(X_test)
y_pred = (y_pred > 0.5)

acc = accuracy_score(y_test, y_pred)
cm = confusion_matrix(y_test, y_pred)

print("Accuracy Score:", acc)
print("Confusion Matrix:\n", cm)

```

NameError Traceback (most recent call last)

Cell In[12], line 3

```

1 from sklearn.metrics import accuracy_score, confusion_matrix
---> 3 y_pred = model.predict(X_test)
     4 y_pred = (y_pred > 0.5)
     5 acc = accuracy_score(y_test, y_pred)

```

NameError: name 'model' is not defined

[]: