In [53]:
```
import pandas as pd
from matplotlib import pyplot as plt
import numpy as np
%matplotlib inline
```

In [54]:
```
df=pd.read_csv('Telco-Customer-Churn.csv')
df.sample(5)
```

Out[54]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	Mul
297	6851-WEFYX	Male	1	Yes	No	35	Yes	
5090	6502-HCJTI	Male	1	Yes	No	7	Yes	
6303	6308-CQRBU	Female	0	Yes	No	71	Yes	
6421	2999-AANRQ	Female	0	No	No	21	Yes	
4824	3339-EAQNV	Male	1	Yes	No	72	Yes	

5 rows × 21 columns

In [55]:
```
df.drop('customerID',axis='columns',inplace=True)
```

In [126…
```
df.dtypes
```

```
Out[126…  gender               int64
          SeniorCitizen        int64
          Partner              int64
          Dependents           int64
          tenure               int64
          PhoneService         int64
          MultipleLines        int64
          InternetService     object
          OnlineSecurity       int64
          OnlineBackup         int64
          DeviceProtection     int64
          TechSupport          int64
          StreamingTV          int64
          StreamingMovies      int64
          Contract            object
          PaperlessBilling     int64
          PaymentMethod       object
          MonthlyCharges     float64
          TotalCharges       float64
          Churn                int64
          dtype: object
```

```
In [127…  df.TotalCharges.values
```

```
Out[127…  array([  29.85, 1889.5 ,  108.15, ...,  346.45,  306.6 , 6844.5 ])
```

```
In [128…  df.MonthlyCharges.values
```

```
Out[128…  array([ 29.85,  56.95,  53.85, ...,  29.6 ,  74.4 , 105.65])
```

```
In [129…  pd.to_numeric(df.TotalCharges,errors='coerce').isnull()
```

```
Out[129…  0       False
          1       False
          2       False
          3       False
          4       False
                  ...
          7038    False
          7039    False
          7040    False
          7041    False
          7042    False
          Name: TotalCharges, Length: 7043, dtype: bool
```

```
In [130…  df[pd.to_numeric(df.TotalCharges,errors='coerce').isnull()]
```

Out[130…

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	In
488	1	0	1	1	0	0	0	
753	0	0	0	1	0	1	0	
936	1	0	1	1	0	1	0	
1082	0	0	1	1	0	1	1	
1340	1	0	1	1	0	0	0	
3331	0	0	1	1	0	1	0	
3826	0	0	1	1	0	1	1	
4380	1	0	1	1	0	1	0	
5218	0	0	1	1	0	1	0	
6670	1	0	1	1	0	1	1	
6754	0	0	0	1	0	1	1	

In [131…

```
df.shape
```

Out[131…

```
(7043, 20)
```

In [132…

```
df.iloc[488]['TotalCharges']
```

Out[132…

```
np.float64(nan)
```

In [133…

```
df1=df[df.TotalCharges!=' ']
df1.shape
```

Out[133…

```
(7043, 20)
```

In [134…

```
df1.TotalCharges=pd.to_numeric(df1.TotalCharges)
```

In [135…

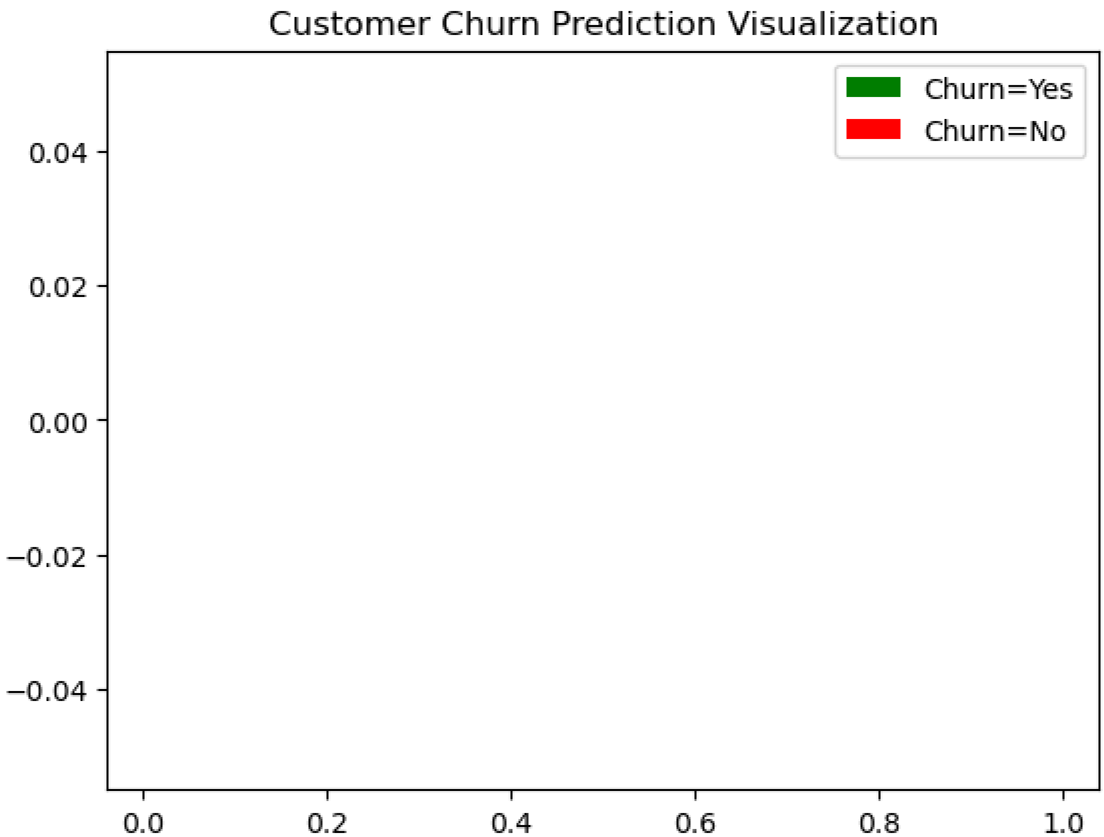
```
df1.TotalCharges.dtypes
```

Out[135…

```
dtype('float64')
```

In [136…

```
tenure_churn_no=df1[df1.Churn=='No'].tenure
tenure_churn_yes=df1[df1.Churn=='Yes'].tenure
plt.hist([tenure_churn_yes,tenure_churn_no],color=['green','red'],label=['Churn=Yes
plt.title('Customer Churn Prediction Visualization')
plt.legend()
plt.show()
```

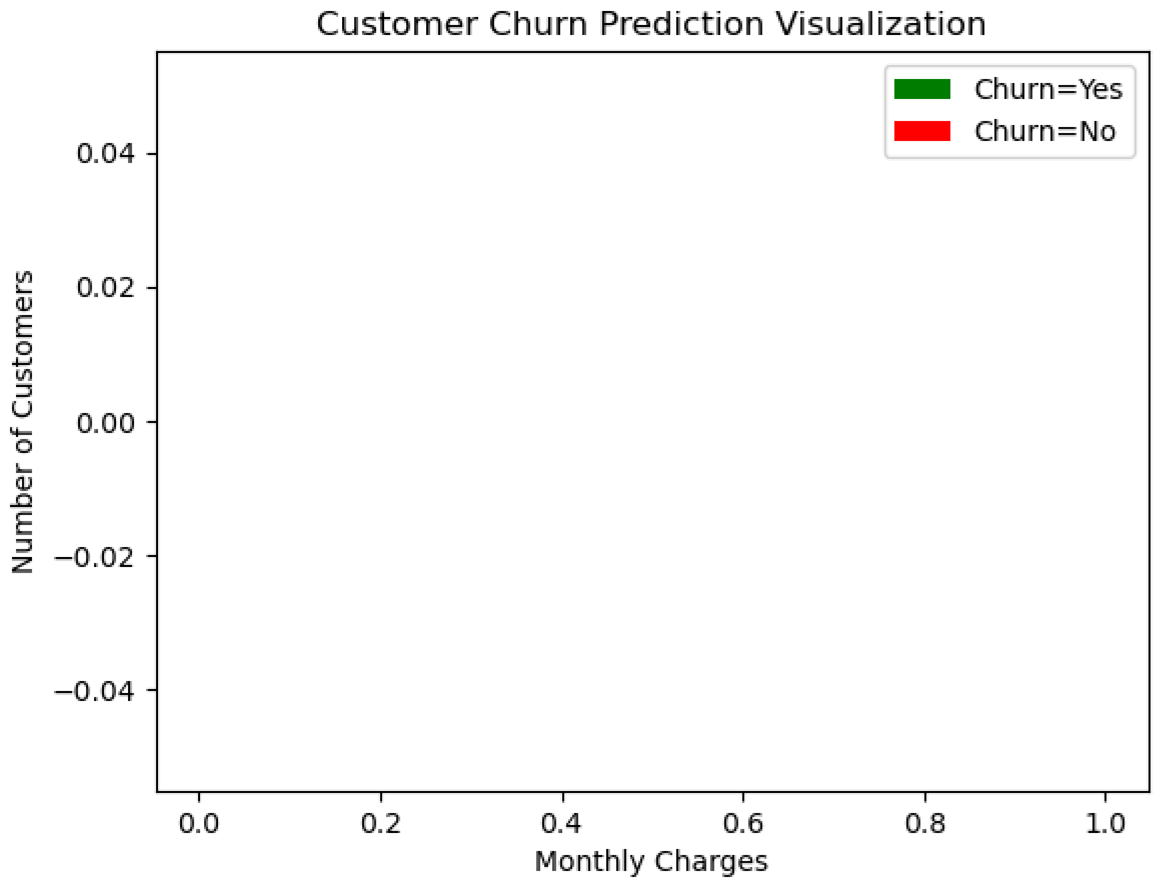


In [137…

```
mc_churn_no=df1[df1.Churn=='No'].MonthlyCharges
mc_churn_yes=df1[df1.Churn=='Yes'].MonthlyCharges
plt.xlabel('Monthly Charges')
plt.ylabel('Number of Customers')
plt.title('Customer Churn Prediction Visualization')

blood_sugar_man=[113,85,90,150,149,88,93,135,80,77,82,129]
blood_sugar_woman=[67,98,89,120,133,150,84,69,89,79,120,112,100]

plt.hist([mc_churn_yes,mc_churn_no],rwidth=0.95,color=['green','red'],label=['Churn
plt.legend()
plt.show()
```



In [138…

```
def print_unique_col_values(df):
    for column in df:
        if df[column].dtypes=='object':
            print(f'{column} : {df[column].unique()}')
```

In [139…

```
print_unique_col_values(df)
```

```
InternetService : ['DSL' 'Fiber optic' 'No']
Contract : ['Month-to-month' 'One year' 'Two year']
PaymentMethod : ['Electronic check' 'Mailed check' 'Bank transfer (automatic)'
 'Credit card (automatic)']
```

In [140…

```
df1=df.replace('No internet service','No',inplace=True)
df1=df.replace('No phone service','No',inplace=True)
```

In [141…

```
print_unique_col_values(df)
```

```
InternetService : ['DSL' 'Fiber optic' 'No']
Contract : ['Month-to-month' 'One year' 'Two year']
PaymentMethod : ['Electronic check' 'Mailed check' 'Bank transfer (automatic)'
 'Credit card (automatic)']
```

In [142…

```
for col in df:
    print(f'{col}:{df[col].unique()}')
```

```
gender:[1 0]
SeniorCitizen:[0 1]
Partner:[1 0]
Dependents:[0 1]
tenure:[ 1 34  2 45  8 22 10 28 62 13 16 58 49 25 69 52 71 21 12 30 47 72 17 27
  5 46 11 70 63 43 15 60 18 66  9  3 31 50 64 56  7 42 35 48 29 65 38 68
 32 55 37 36 41  6  4 33 67 23 57 61 14 20 53 40 59 24 44 19 54 51 26  0
 39]
PhoneService:[0 1]
MultipleLines:[0 1]
InternetService:['DSL' 'Fiber optic' 'No']
OnlineSecurity:[0 1]
OnlineBackup:[1 0]
DeviceProtection:[0 1]
TechSupport:[0 1]
StreamingTV:[0 1]
StreamingMovies:[0 1]
Contract:['Month-to-month' 'One year' 'Two year']
PaperlessBilling:[1 0]
PaymentMethod:['Electronic check' 'Mailed check' 'Bank transfer (automatic)'
 'Credit card (automatic)']
MonthlyCharges:[29.85 56.95 53.85 ... 63.1  44.2  78.7 ]
TotalCharges:[  29.85 1889.5   108.15 ...  346.45  306.6  6844.5 ]
Churn:[0 1]
```

In [143…

```python
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')
```

In [144…

```python
for col in df:
    print(f'{col}:{df[col].unique()}')
```

```
gender:[1 0]
SeniorCitizen:[0 1]
Partner:[1 0]
Dependents:[0 1]
tenure:[ 1 34  2 45  8 22 10 28 62 13 16 58 49 25 69 52 71 21 12 30 47 72 17 27
  5 46 11 70 63 43 15 60 18 66  9  3 31 50 64 56  7 42 35 48 29 65 38 68
 32 55 37 36 41  6  4 33 67 23 57 61 14 20 53 40 59 24 44 19 54 51 26  0
 39]
PhoneService:[0 1]
MultipleLines:[0 1]
InternetService:['DSL' 'Fiber optic' 'No']
OnlineSecurity:[0 1]
OnlineBackup:[1 0]
DeviceProtection:[0 1]
TechSupport:[0 1]
StreamingTV:[0 1]
StreamingMovies:[0 1]
Contract:['Month-to-month' 'One year' 'Two year']
PaperlessBilling:[1 0]
PaymentMethod:['Electronic check' 'Mailed check' 'Bank transfer (automatic)'
 'Credit card (automatic)']
MonthlyCharges:[29.85 56.95 53.85 ... 63.1  44.2  78.7 ]
TotalCharges:[  29.85 1889.5   108.15 ...  346.45  306.6  6844.5 ]
Churn:[0 1]
```

In [145…

```python
df['gender'] = df['gender'].replace({'Female': 1, 'Male': 0})
```

In [146…

```
df['gender'].unique()
```

Out[146…

```
array([1, 0])
```

In [147…

```
df2=pd.get_dummies(data=df,columns=['InternetService','Contract','PaymentMethod'])
df2.columns
```

Out[147…

```
Index(['gender', 'SeniorCitizen', 'Partner', 'Dependents', 'tenure',
       'PhoneService', 'MultipleLines', 'OnlineSecurity', 'OnlineBackup',
       'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies',
       'PaperlessBilling', 'MonthlyCharges', 'TotalCharges', 'Churn',
       'InternetService_DSL', 'InternetService_Fiber optic',
       'InternetService_No', 'Contract_Month-to-month', 'Contract_One year',
       'Contract_Two year', 'PaymentMethod_Bank transfer (automatic)',
       'PaymentMethod_Credit card (automatic)',
       'PaymentMethod_Electronic check', 'PaymentMethod_Mailed check'],
      dtype='object')
```

In [148…

```
df2.sample(4)
```

Out[148…

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	O
1882	0	0	1	1	29	1	0	
5803	0	0	1	0	5	1	1	
6770	1	0	1	1	4	1	0	
2646	0	0	0	0	56	1	0	

4 rows × 27 columns

In [149…

```
df2.dtypes
```

Out[149…

```
gender                                        int64
SeniorCitizen                                 int64
Partner                                       int64
Dependents                                    int64
tenure                                        int64
PhoneService                                  int64
MultipleLines                                 int64
OnlineSecurity                                int64
OnlineBackup                                  int64
DeviceProtection                              int64
TechSupport                                   int64
StreamingTV                                   int64
StreamingMovies                               int64
PaperlessBilling                              int64
MonthlyCharges                              float64
TotalCharges                                float64
Churn                                         int64
InternetService_DSL                            bool
InternetService_Fiber optic                    bool
InternetService_No                             bool
Contract_Month-to-month                        bool
Contract_One year                              bool
Contract_Two year                              bool
PaymentMethod_Bank transfer (automatic)        bool
PaymentMethod_Credit card (automatic)          bool
PaymentMethod_Electronic check                 bool
PaymentMethod_Mailed check                     bool
dtype: object
```

In [150…

```
df2 = df2.astype({col: int for col in df2.select_dtypes('bool').columns})
```

In [151…

```
df2.dtypes
```

Out[151…

```
gender                                      int64
SeniorCitizen                               int64
Partner                                     int64
Dependents                                  int64
tenure                                      int64
PhoneService                                int64
MultipleLines                               int64
OnlineSecurity                              int64
OnlineBackup                                int64
DeviceProtection                            int64
TechSupport                                 int64
StreamingTV                                 int64
StreamingMovies                             int64
PaperlessBilling                            int64
MonthlyCharges                            float64
TotalCharges                              float64
Churn                                       int64
InternetService_DSL                         int64
InternetService_Fiber optic                 int64
InternetService_No                          int64
Contract_Month-to-month                     int64
Contract_One year                           int64
Contract_Two year                           int64
PaymentMethod_Bank transfer (automatic)     int64
PaymentMethod_Credit card (automatic)       int64
PaymentMethod_Electronic check              int64
PaymentMethod_Mailed check                  int64
dtype: object
```

In [157…

```
cols_to_scale=['tenure','MonthlyCharges','TotalCharges']

from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()

df2[cols_to_scale]=scaler.fit_transform(df2[cols_to_scale])
```

In [158…

```
df2.sample(3)
```

Out[158…

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines
3999	1	1	0	0	0.027778	1	0
3388	1	0	1	0	0.902778	1	0
5131	1	0	1	0	0.611111	1	1

3 rows × 27 columns

In [160…

```
for col in df2:
    print(f'{col}:{df2[col].unique()}')
```

```
gender:[1 0]
SeniorCitizen:[0 1]
Partner:[1 0]
Dependents:[0 1]
tenure:[0.01388889 0.47222222 0.02777778 0.625      0.11111111 0.30555556
 0.13888889 0.38888889 0.86111111 0.18055556 0.22222222 0.80555556
 0.68055556 0.34722222 0.95833333 0.72222222 0.98611111 0.29166667
 0.16666667 0.41666667 0.65277778 1.         0.23611111 0.375
 0.06944444 0.63888889 0.15277778 0.97222222 0.875      0.59722222
 0.20833333 0.83333333 0.25       0.91666667 0.125      0.04166667
 0.43055556 0.69444444 0.88888889 0.77777778 0.09722222 0.58333333
 0.48611111 0.66666667 0.40277778 0.90277778 0.52777778 0.94444444
 0.44444444 0.76388889 0.51388889 0.5        0.56944444 0.08333333
 0.05555556 0.45833333 0.93055556 0.31944444 0.79166667 0.84722222
 0.19444444 0.27777778 0.73611111 0.55555556 0.81944444 0.33333333
 0.61111111 0.26388889 0.75       0.70833333 0.36111111 0.
 0.54166667]
PhoneService:[0 1]
MultipleLines:[0 1]
OnlineSecurity:[0 1]
OnlineBackup:[1 0]
DeviceProtection:[0 1]
TechSupport:[0 1]
StreamingTV:[0 1]
StreamingMovies:[0 1]
PaperlessBilling:[1 0]
MonthlyCharges:[0.11542289 0.38507463 0.35422886 ... 0.44626866 0.25820896 0.6014925
4]
TotalCharges:[0.00343704 0.21756402 0.01245279 ... 0.03989153 0.03530306 0.78810105]
Churn:[0 1]
InternetService_DSL:[1 0]
InternetService_Fiber optic:[0 1]
InternetService_No:[0 1]
Contract_Month-to-month:[1 0]
Contract_One year:[0 1]
Contract_Two year:[0 1]
PaymentMethod_Bank transfer (automatic):[0 1]
PaymentMethod_Credit card (automatic):[0 1]
PaymentMethod_Electronic check:[1 0]
PaymentMethod_Mailed check:[0 1]
```

In [161...

```
x=df2.drop('Churn',axis='columns')
y=df2['Churn']
```

In [163...

```
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=5)
```

In [164...

```
X_train.shape
```

Out[164...

```
(5634, 26)
```

In [165...

```
X_test.shape
```

Out[165...

```
(1409, 26)
```

In [166…

```
X_train[:10]
```

Out[166…

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines
5860	1	0	0	0	0.027778	1	0
2458	0	1	1	0	0.694444	1	1
5879	0	0	1	0	0.458333	1	0
4708	1	0	1	1	0.777778	1	0
1293	0	0	1	1	0.930556	1	1
2242	0	0	1	1	0.611111	1	1
1444	0	0	0	1	0.569444	1	0
3269	0	0	0	0	0.902778	1	1
101	1	0	1	1	0.013889	1	0
4191	1	0	1	0	0.875000	1	1

10 rows × 26 columns

In [167…

```
len(X_train.columns)
```

Out[167…

26

In [177…

```
import sys
!{sys.executable} -m pip install --upgrade tensorflow
```

```
Requirement already satisfied: tensorflow in c:\users\ramya\anaconda3\lib\site-packa
ges (2.20.0)
Requirement already satisfied: absl-py>=1.0.0 in c:\users\ramya\anaconda3\lib\site-p
ackages (from tensorflow) (2.3.1)
Requirement already satisfied: astunparse>=1.6.0 in c:\users\ramya\anaconda3\lib\sit
e-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=24.3.25 in c:\users\ramya\anaconda3\lib
\site-packages (from tensorflow) (25.12.19)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in c:\users\ramya
\anaconda3\lib\site-packages (from tensorflow) (0.7.0)
Requirement already satisfied: google_pasta>=0.1.1 in c:\users\ramya\anaconda3\lib\s
ite-packages (from tensorflow) (0.2.0)
Requirement already satisfied: libclang>=13.0.0 in c:\users\ramya\anaconda3\lib\site
-packages (from tensorflow) (18.1.1)
Requirement already satisfied: opt_einsum>=2.3.2 in c:\users\ramya\anaconda3\lib\sit
e-packages (from tensorflow) (3.4.0)
Requirement already satisfied: packaging in c:\users\ramya\anaconda3\lib\site-packag
es (from tensorflow) (24.2)
Requirement already satisfied: protobuf>=5.28.0 in c:\users\ramya\anaconda3\lib\site
-packages (from tensorflow) (5.29.3)
Requirement already satisfied: requests<3,>=2.21.0 in c:\users\ramya\anaconda3\lib\s
ite-packages (from tensorflow) (2.32.3)
Requirement already satisfied: setuptools in c:\users\ramya\anaconda3\lib\site-packa
ges (from tensorflow) (72.1.0)
Requirement already satisfied: six>=1.12.0 in c:\users\ramya\anaconda3\lib\site-pack
ages (from tensorflow) (1.17.0)
Requirement already satisfied: termcolor>=1.1.0 in c:\users\ramya\anaconda3\lib\site
-packages (from tensorflow) (3.3.0)
Requirement already satisfied: typing_extensions>=3.6.6 in c:\users\ramya\anaconda3
\lib\site-packages (from tensorflow) (4.12.2)
Requirement already satisfied: wrapt>=1.11.0 in c:\users\ramya\anaconda3\lib\site-pa
ckages (from tensorflow) (1.17.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in c:\users\ramya\anaconda3\lib\s
ite-packages (from tensorflow) (1.76.0)
Requirement already satisfied: tensorboard~=2.20.0 in c:\users\ramya\anaconda3\lib\s
ite-packages (from tensorflow) (2.20.0)
Requirement already satisfied: keras>=3.10.0 in c:\users\ramya\anaconda3\lib\site-pa
ckages (from tensorflow) (3.13.0)
Requirement already satisfied: numpy>=1.26.0 in c:\users\ramya\anaconda3\lib\site-pa
ckages (from tensorflow) (2.1.3)
Requirement already satisfied: h5py>=3.11.0 in c:\users\ramya\anaconda3\lib\site-pac
kages (from tensorflow) (3.12.1)
Requirement already satisfied: ml_dtypes<1.0.0,>=0.5.1 in c:\users\ramya\anaconda3\l
ib\site-packages (from tensorflow) (0.5.4)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\ramya\anaconda3
\lib\site-packages (from requests<3,>=2.21.0->tensorflow) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in c:\users\ramya\anaconda3\lib\site-pac
kages (from requests<3,>=2.21.0->tensorflow) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\ramya\anaconda3\lib\si
te-packages (from requests<3,>=2.21.0->tensorflow) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\ramya\anaconda3\lib\si
te-packages (from requests<3,>=2.21.0->tensorflow) (2025.11.12)
Requirement already satisfied: markdown>=2.6.8 in c:\users\ramya\anaconda3\lib\site-
packages (from tensorboard~=2.20.0->tensorflow) (3.8)
Requirement already satisfied: pillow in c:\users\ramya\anaconda3\lib\site-packages
(from tensorboard~=2.20.0->tensorflow) (11.1.0)
```

```
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in c:\users\ramya\anaconda3\lib\site-packages (from tensorboard~=2.20.0->tensorflow) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in c:\users\ramya\anaconda3\lib\site-packages (from tensorboard~=2.20.0->tensorflow) (3.1.3)
Requirement already satisfied: wheel<1.0,>=0.23.0 in c:\users\ramya\anaconda3\lib\site-packages (from astunparse>=1.6.0->tensorflow) (0.45.1)
Requirement already satisfied: rich in c:\users\ramya\anaconda3\lib\site-packages (from keras>=3.10.0->tensorflow) (13.9.4)
Requirement already satisfied: namex in c:\users\ramya\anaconda3\lib\site-packages (from keras>=3.10.0->tensorflow) (0.1.0)
Requirement already satisfied: optree in c:\users\ramya\anaconda3\lib\site-packages (from keras>=3.10.0->tensorflow) (0.18.0)
Requirement already satisfied: MarkupSafe>=2.1.1 in c:\users\ramya\anaconda3\lib\site-packages (from werkzeug>=1.0.1->tensorboard~=2.20.0->tensorflow) (3.0.2)
Requirement already satisfied: markdown-it-py>=2.2.0 in c:\users\ramya\anaconda3\lib\site-packages (from rich->keras>=3.10.0->tensorflow) (2.2.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in c:\users\ramya\anaconda3\lib\site-packages (from rich->keras>=3.10.0->tensorflow) (2.19.1)
Requirement already satisfied: mdurl~=0.1 in c:\users\ramya\anaconda3\lib\site-packages (from markdown-it-py>=2.2.0->rich->keras>=3.10.0->tensorflow) (0.1.0)
```

In [183…

```python
import tensorflow as tf
from tensorflow import keras

model = keras.Sequential([
    keras.layers.Dense(20,input_shape=(26,),activation='relu'),
    keras.layers.Dense(1,activation='sigmoid'),
])

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

model.fit(X_train,y_train, epochs=100)
```

```
Epoch 1/100
C:\Users\ramya\anaconda3\Lib\site-packages\keras\src\layers\core\dense.py:106: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

```
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.7639 - loss: 0.4730
Epoch 2/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.7961 - loss: 0.4279
Epoch 3/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.7993 - loss: 0.4224
Epoch 4/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8053 - loss: 0.4187
Epoch 5/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8072 - loss: 0.4171
Epoch 6/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8062 - loss: 0.4162
Epoch 7/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8067 - loss: 0.4156
Epoch 8/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8042 - loss: 0.4145
Epoch 9/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8085 - loss: 0.4144
Epoch 10/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8072 - loss: 0.4142
Epoch 11/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8095 - loss: 0.4130
Epoch 12/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8083 - loss: 0.4129
Epoch 13/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8095 - loss: 0.4124
Epoch 14/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8078 - loss: 0.4125
Epoch 15/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8076 - loss: 0.4113
Epoch 16/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8103 - loss: 0.4115
Epoch 17/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8095 - loss: 0.4104
Epoch 18/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8101 - loss: 0.4102
Epoch 19/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8083 - loss: 0.4098
Epoch 20/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8104 - loss: 0.4095
Epoch 21/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8110 - loss: 0.4087
Epoch 22/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8101 - loss: 0.4086
Epoch 23/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8106 - loss: 0.4090
Epoch 24/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8092 - loss: 0.4080
Epoch 25/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8135 - loss: 0.4079
Epoch 26/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8104 - loss: 0.4075
Epoch 27/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8106 - loss: 0.4072
Epoch 28/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8122 - loss: 0.4071
Epoch 29/100
```

```
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8131 - loss: 0.4066
Epoch 30/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8111 - loss: 0.4065
Epoch 31/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8142 - loss: 0.4059
Epoch 32/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8135 - loss: 0.4059
Epoch 33/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8110 - loss: 0.4058
Epoch 34/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8124 - loss: 0.4049
Epoch 35/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8127 - loss: 0.4053
Epoch 36/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8135 - loss: 0.4046
Epoch 37/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8110 - loss: 0.4043
Epoch 38/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8110 - loss: 0.4043
Epoch 39/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8120 - loss: 0.4034
Epoch 40/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8142 - loss: 0.4044
Epoch 41/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8151 - loss: 0.4036
Epoch 42/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step - accuracy: 0.8147 - loss: 0.4026
Epoch 43/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8136 - loss: 0.4023
Epoch 44/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8136 - loss: 0.4018
Epoch 45/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8138 - loss: 0.4020
Epoch 46/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8142 - loss: 0.4019
Epoch 47/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8124 - loss: 0.4010
Epoch 48/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8147 - loss: 0.4016
Epoch 49/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8140 - loss: 0.4004
Epoch 50/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8163 - loss: 0.4003
Epoch 51/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8154 - loss: 0.4000
Epoch 52/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8143 - loss: 0.4008
Epoch 53/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8154 - loss: 0.4002
Epoch 54/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8135 - loss: 0.3992
Epoch 55/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8138 - loss: 0.3998
Epoch 56/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8156 - loss: 0.3990
Epoch 57/100
```

```
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8166 - loss: 0.3987
Epoch 58/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8149 - loss: 0.3991
Epoch 59/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8166 - loss: 0.3980
Epoch 60/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8152 - loss: 0.3985
Epoch 61/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8165 - loss: 0.3986
Epoch 62/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8152 - loss: 0.3978
Epoch 63/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8159 - loss: 0.3980
Epoch 64/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8165 - loss: 0.3977
Epoch 65/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8158 - loss: 0.3975
Epoch 66/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8163 - loss: 0.3971
Epoch 67/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8161 - loss: 0.3972
Epoch 68/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8166 - loss: 0.3965
Epoch 69/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8177 - loss: 0.3966
Epoch 70/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - accuracy: 0.8156 - loss: 0.3969
Epoch 71/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8168 - loss: 0.3964
Epoch 72/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8161 - loss: 0.3964
Epoch 73/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8152 - loss: 0.3957
Epoch 74/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8181 - loss: 0.3956
Epoch 75/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8168 - loss: 0.3963
Epoch 76/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8152 - loss: 0.3955
Epoch 77/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8172 - loss: 0.3950
Epoch 78/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8188 - loss: 0.3952
Epoch 79/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8165 - loss: 0.3949
Epoch 80/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8181 - loss: 0.3946
Epoch 81/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8166 - loss: 0.3945
Epoch 82/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8158 - loss: 0.3943
Epoch 83/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8149 - loss: 0.3940
Epoch 84/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8184 - loss: 0.3936
Epoch 85/100
```

```
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8182 - loss: 0.3937
Epoch 86/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8170 - loss: 0.3933
Epoch 87/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8166 - loss: 0.3935
Epoch 88/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - accuracy: 0.8179 - loss: 0.3933
Epoch 89/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8177 - loss: 0.3930
Epoch 90/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8206 - loss: 0.3926
Epoch 91/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8165 - loss: 0.3928
Epoch 92/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - accuracy: 0.8191 - loss: 0.3929
Epoch 93/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8175 - loss: 0.3922
Epoch 94/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8182 - loss: 0.3919
Epoch 95/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8186 - loss: 0.3917
Epoch 96/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8197 - loss: 0.3918
Epoch 97/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8181 - loss: 0.3914
Epoch 98/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8207 - loss: 0.3908
Epoch 99/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8193 - loss: 0.3912
Epoch 100/100
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8193 - loss: 0.3914
```

Out[183…

```
<keras.src.callbacks.history.History at 0x183a6041350>
```

In [184…

```python
model.evaluate(X_test,y_test)
```

```
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.7991 - loss: 0.4272
```

Out[184…

```
[0.4272060692310333, 0.7991483211517334]
```

In [185…

```python
yp = model.predict(X_test)
yp[:5]
```

```
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step
```

Out[185…

```
array([[0.2592207 ],
       [0.4044013 ],
       [0.34134796],
       [0.9068077 ],
       [0.0928181 ]], dtype=float32)
```

In [189…

```python
y_test[:10]
```

Out[189…
```
4213    1
5035    0
3713    1
1720    0
234     0
4558    1
40      0
3455    1
5944    1
1089    0
Name: Churn, dtype: int64
```

In [190…
```
y_pred = []
for element in yp:
    if element > 0.5:
        y_pred.append(1)
    else:
        y_pred.append(0)
```

In [191…
```
y_pred[:10]
```

Out[191…
```
[0, 0, 0, 1, 0, 1, 0, 1, 0, 0]
```

In [194…
```
from sklearn.metrics import confusion_matrix , classification_report
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.84      0.89      0.87      1023
           1       0.66      0.56      0.61       386

    accuracy                           0.80      1409
   macro avg       0.75      0.73      0.74      1409
weighted avg       0.79      0.80      0.79      1409
```

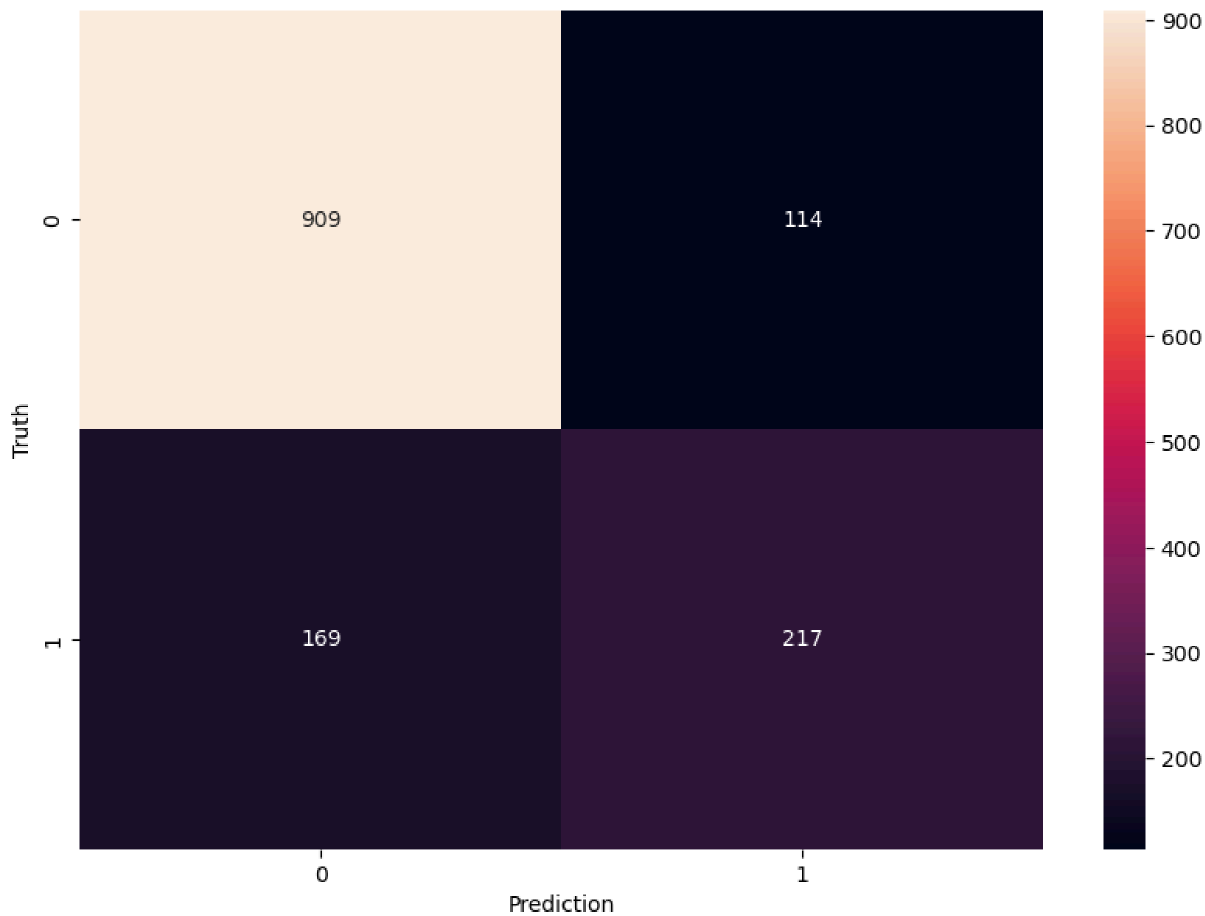
In [195…
```
import seaborn as sns

# Step 1: Make confusion matrix using TensorFlow
cm = tf.math.confusion_matrix(labels=y_test, predictions=y_pred)

# Step 2: Make a nice heatmap to see it clearly
plt.figure(figsize=(10,7))
sns.heatmap(cm, annot=True, fmt='d')  # annot=True puts numbers on cells, fmt='d' m

# Step 3: Label axes
plt.xlabel('Prediction')  # x-axis = predicted labels
plt.ylabel('Truth')       # y-axis = actual labels

# Step 4: Show the plot
plt.show()
```



Accuracy

In [197…
```
round((909+217)/(909+217+169+114),2)
```

Out[197… 0.8

Precision for 0 class i.e. Precision for customers who did not churned

In [198…
```
round(909/(909+169),2)
```

Out[198… 0.84

Precision for 1 class i.e. Precision for customers actually churned

In [200…
```
round(217/(217+114),2)
```

Out[200… 0.66

Recall for 0 class

In [201…
```
round(909/(909+114),2)
```
Out[201…
```
0.89
```
In [202…
```
round(217/(217+169),2)
```
Out[202…
```
0.56
```
In [ ]:
```

```
In [ ]:
```

```