```
pip install tensorflow
```

```
Requirement already satisfied: tensorflow in d:\anaconda\lib\site-
packages (2.20.0)
Requirement already satisfied: absl-py>=1.0.0 in d:\anaconda\lib\site-
packages (from tensorflow) (2.3.1)
Requirement already satisfied: astunparse>=1.6.0 in d:\anaconda\lib\
site-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=24.3.25 in d:\anaconda\
lib\site-packages (from tensorflow) (25.9.23)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in
d:\anaconda\lib\site-packages (from tensorflow) (0.6.0)
Requirement already satisfied: google_pasta>=0.1.1 in d:\anaconda\lib\
site-packages (from tensorflow) (0.2.0)
Requirement already satisfied: libclang>=13.0.0 in d:\anaconda\lib\
site-packages (from tensorflow) (18.1.1)
Requirement already satisfied: opt_einsum>=2.3.2 in d:\anaconda\lib\
site-packages (from tensorflow) (3.4.0)
Requirement already satisfied: packaging in d:\anaconda\lib\site-
packages (from tensorflow) (24.1)
Requirement already satisfied: protobuf>=5.28.0 in d:\anaconda\lib\
site-packages (from tensorflow) (6.32.1)
Requirement already satisfied: requests<3,>=2.21.0 in d:\anaconda\lib\
site-packages (from tensorflow) (2.32.3)
Requirement already satisfied: setuptools in d:\anaconda\lib\site-
packages (from tensorflow) (75.1.0)
Requirement already satisfied: six>=1.12.0 in d:\anaconda\lib\site-
packages (from tensorflow) (1.16.0)
Requirement already satisfied: termcolor>=1.1.0 in d:\anaconda\lib\
site-packages (from tensorflow) (3.1.0)
Requirement already satisfied: typing_extensions>=3.6.6 in d:\
anaconda\lib\site-packages (from tensorflow) (4.15.0)
Requirement already satisfied: wrapt>=1.11.0 in d:\anaconda\lib\site-
packages (from tensorflow) (1.14.1)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in d:\anaconda\lib\
site-packages (from tensorflow) (1.75.0)
Requirement already satisfied: tensorboard~=2.20.0 in d:\anaconda\lib\
site-packages (from tensorflow) (2.20.0)
Requirement already satisfied: keras>=3.10.0 in d:\anaconda\lib\site-
packages (from tensorflow) (3.11.3)
Requirement already satisfied: numpy>=1.26.0 in d:\anaconda\lib\site-
packages (from tensorflow) (1.26.4)
Requirement already satisfied: h5py>=3.11.0 in d:\anaconda\lib\site-
packages (from tensorflow) (3.11.0)
Requirement already satisfied: ml_dtypes<1.0.0,>=0.5.1 in d:\anaconda\
lib\site-packages (from tensorflow) (0.5.3)
Requirement already satisfied: wheel<1.0,>=0.23.0 in d:\anaconda\lib\
site-packages (from astunparse>=1.6.0->tensorflow) (0.44.0)
Requirement already satisfied: rich in d:\anaconda\lib\site-packages
(from keras>=3.10.0->tensorflow) (13.7.1)
```

```
Requirement already satisfied: namex in d:\anaconda\lib\site-packages
(from keras>=3.10.0->tensorflow) (0.1.0)
Requirement already satisfied: optree in d:\anaconda\lib\site-packages
(from keras>=3.10.0->tensorflow) (0.17.0)
Requirement already satisfied: charset-normalizer<4,>=2 in d:\
anaconda\lib\site-packages (from requests<3,>=2.21.0->tensorflow)
(3.3.2)
Requirement already satisfied: idna<4,>=2.5 in d:\anaconda\lib\site-
packages (from requests<3,>=2.21.0->tensorflow) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in d:\anaconda\lib\
site-packages (from requests<3,>=2.21.0->tensorflow) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in d:\anaconda\lib\
site-packages (from requests<3,>=2.21.0->tensorflow) (2024.8.30)
Requirement already satisfied: markdown>=2.6.8 in d:\anaconda\lib\
site-packages (from tensorboard~=2.20.0->tensorflow) (3.4.1)
Requirement already satisfied: pillow in d:\anaconda\lib\site-packages
(from tensorboard~=2.20.0->tensorflow) (10.4.0)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0
in d:\anaconda\lib\site-packages (from tensorboard~=2.20.0-
>tensorflow) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in d:\anaconda\lib\
site-packages (from tensorboard~=2.20.0->tensorflow) (3.0.3)
Requirement already satisfied: MarkupSafe>=2.1.1 in d:\anaconda\lib\
site-packages (from werkzeug>=1.0.1->tensorboard~=2.20.0->tensorflow)
(2.1.3)
Requirement already satisfied: markdown-it-py>=2.2.0 in d:\anaconda\
lib\site-packages (from rich->keras>=3.10.0->tensorflow) (2.2.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in d:\anaconda\
lib\site-packages (from rich->keras>=3.10.0->tensorflow) (2.15.1)
Requirement already satisfied: mdurl~=0.1 in d:\anaconda\lib\site-
packages (from markdown-it-py>=2.2.0->rich->keras>=3.10.0->tensorflow)
(0.1.0)
Note: you may need to restart the kernel to use updated packages.
```

```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from keras.models import Sequential
from keras.layers import Dense, Input
from sklearn.metrics import confusion_matrix, accuracy_score,
classification_report
```

```python
df = pd.read_csv('churn_modelling.csv')
df.head()
```

```
   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
```

```
df.shape
```

```
(10000, 14)
```

```
df.describe
```

```
<bound method NDFrame.describe of       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
...         ...         ...        ...          ...       ...     ...
...
9995       9996    15606229   Obijiaku          771    France    Male
39
9996       9997    15569892  Johnstone          516    France    Male
35
9997       9998    15584532        Liu          709    France  Female
36
```

```
9998       9999  15682355  Sabbatini          772   Germany    Male
42
9999      10000  15628319     Walker          792    France  Female
28

      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
...      ...        ...            ...        ...             ...
9995       5       0.00              2          1               0
9996      10   57369.61              1          1               1
9997       7       0.00              1          0               1
9998       3   75075.31              2          1               0
9999       4  130142.79              1          1               0

      EstimatedSalary  Exited
0           101348.88       1
1           112542.58       0
2           113931.57       1
3            93826.63       0
4            79084.10       0
...               ...     ...
9995         96270.64       0
9996        101699.77       0
9997         42085.58       1
9998         92888.52       1
9999         38190.78       0

[10000 rows x 14 columns]>
```

```
df.isnull()
df.isnull().sum()
```

```
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
```

```
df.info()
```

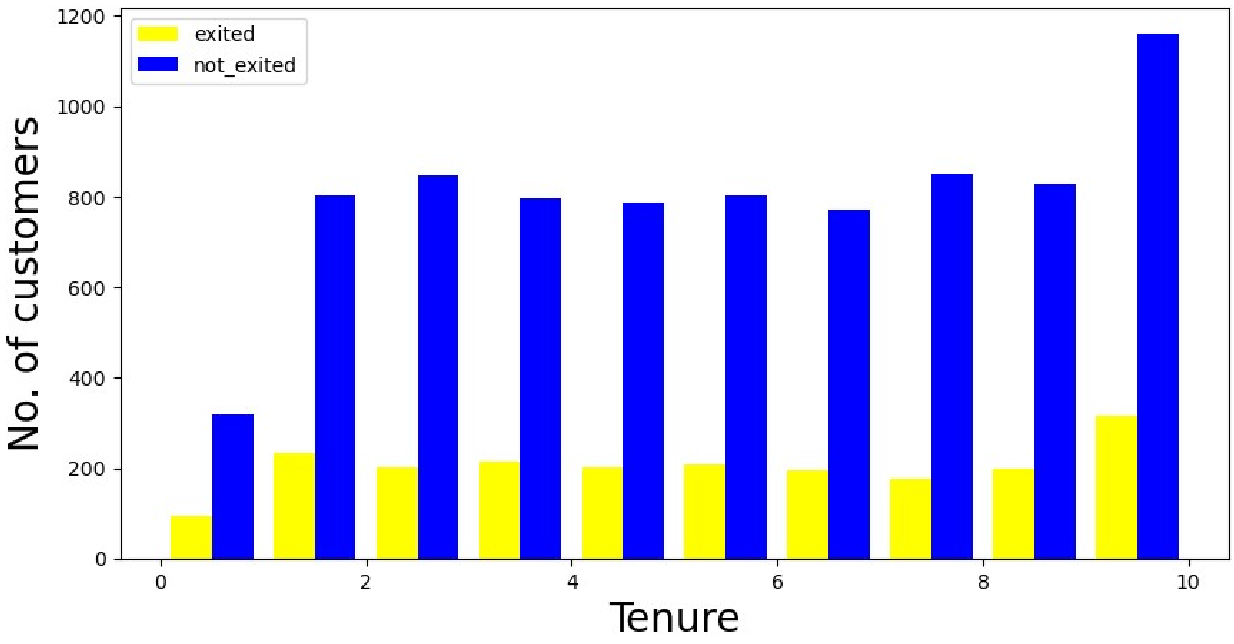
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   RowNumber        10000 non-null  int64
 1   CustomerId       10000 non-null  int64
 2   Surname          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
```

```
df.dtypes
```

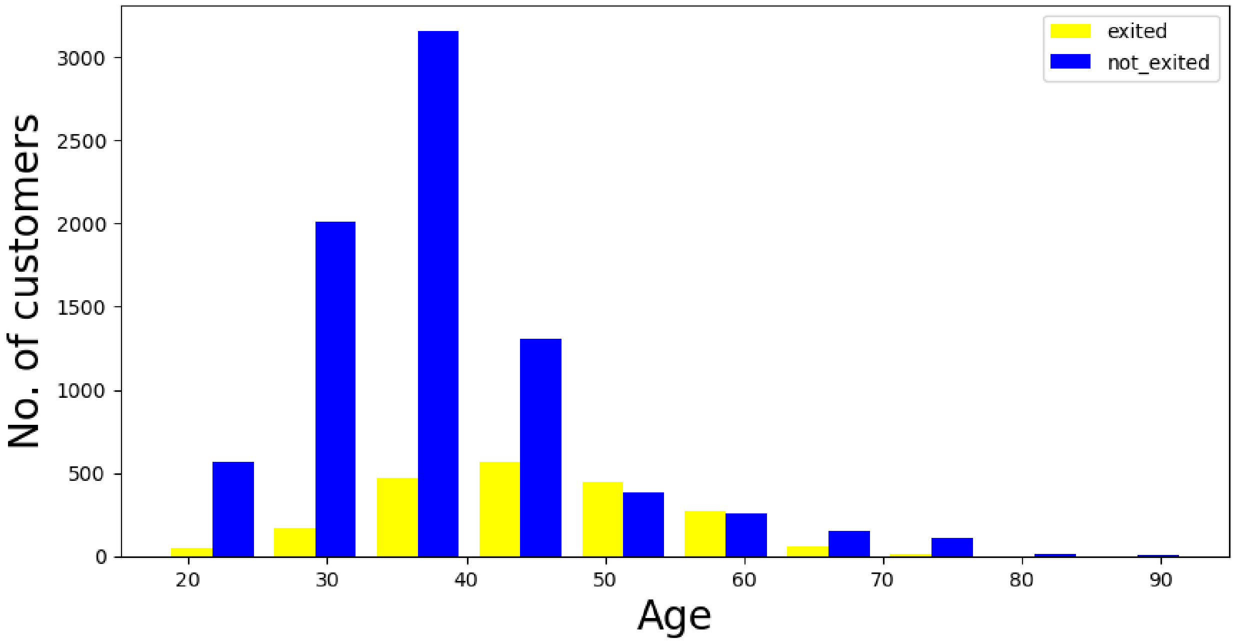
```
RowNumber            int64
CustomerId           int64
Surname             object
CreditScore          int64
Geography           object
Gender              object
Age                  int64
Tenure               int64
Balance            float64
NumOfProducts        int64
HasCrCard            int64
IsActiveMember       int64
EstimatedSalary    float64
Exited               int64
dtype: object
```

```
df.columns
```

```
Index(['RowNumber', 'CustomerId', 'Surname', 'CreditScore',
'Geography',
       'Gender', 'Age', 'Tenure', 'Balance', 'NumOfProducts',
'HasCrCard',
       'IsActiveMember', 'EstimatedSalary', 'Exited'],
      dtype='object')

df = df.drop(['RowNumber', 'Surname', 'CustomerId'], axis = 1)
df.head()

   CreditScore Geography  Gender  Age  Tenure    Balance  
NumOfProducts  \
0          619    France  Female   42       2       0.00  
1   
1          608     Spain  Female   41       1   83807.86  
1   
2          502    France  Female   42       8  159660.80  
3   
3          699    France  Female   39       1       0.00  
2   
4          850     Spain  Female   43       2  125510.82  
1   

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

def visualization(x, y, xlabel):
    plt.figure(figsize=(10,5))
    plt.hist([x, y], color=['yellow', 'blue'], label = ['exited', 
'not_exited'])
    plt.xlabel(xlabel,fontsize=20)
    plt.ylabel('No. of customers', fontsize=20)
    plt.legend()

df_churn_exited = df[df['Exited']==1]['Tenure']
df_churn_not_exited = df[df['Exited']==0]['Tenure']

visualization(df_churn_exited, df_churn_not_exited, 'Tenure')
```



```
df_churn_exited2 = df[df['Exited']==1]['Age']
df_churn_not_exited2 = df[df['Exited']==0]['Age']

visualization(df_churn_exited2, df_churn_not_exited2, 'Age')
```



```
x = df[['CreditScore', 'Gender', 'Age', 'Tenure', 'Balance',
'NumOfProducts',
        'HasCrCard', 'IsActiveMember', 'EstimatedSalary']]
```

```
states = pd.get_dummies(df['Geography'], drop_first = True)
gender = pd.get_dummies(df['Gender'], drop_first = True)

df = pd.concat([df,gender,states], axis = 1)
df.head()

   CreditScore Geography  Gender  Age  Tenure    Balance
NumOfProducts  \
0          619    France  Female   42       2       0.00
1
1          608     Spain  Female   41       1   83807.86
1
2          502    France  Female   42       8  159660.80
3
3          699    France  Female   39       1       0.00
2
4          850     Spain  Female   43       2  125510.82
1

   HasCrCard  IsActiveMember  EstimatedSalary  Exited   Male  Germany
Spain
0          1               1        101348.88       1  False    False
False
1          0               1        112542.58       0  False    False
True
2          1               0        113931.57       1  False    False
False
3          0               0         93826.63       0  False    False
False
4          1               1         79084.10       0  False    False
True

x = df[['CreditScore', 'Age', 'Tenure', 'Balance', 'NumOfProducts',
'IsActiveMember', 'EstimatedSalary', 'Male', 'Germany', 'Spain']]
y = df['Exited']

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size =
0.3)

sc = StandardScaler()

x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)

print('Training Data:\n',x_train)

Training Data:
 [[ 0.71510573 -0.57403373 -1.04508635 ...  0.92216229 -0.57910982
  -0.57185157]
 [-0.59340941  0.18697782  0.33073846 ... -1.08440782 -0.57910982
  -0.57185157]
 [ 0.05569653 -0.47890728  0.33073846 ... -1.08440782 -0.57910982
   1.74870553]
 ...
 [ 1.44663782 -0.57403373  1.70656326 ... -1.08440782  1.7267882
  -0.57185157]
 [ 0.31327825 -0.09840151 -1.04508635 ...  0.92216229 -0.57910982
   1.74870553]
 [-0.3976473  -1.23991883 -0.35717395 ... -1.08440782 -0.57910982
   1.74870553]]
```

```
print('Testing Data:\n',x_test)
```

```
Testing Data:
 [[-1.34554803 -0.09840151 -0.35717395 ... -1.08440782 -0.57910982
  -0.57185157]
 [-0.53158979 -1.33504527 -0.70113015 ...  0.92216229 -0.57910982
   1.74870553]
 [-0.68613883  0.75773648  0.67469466 ...  0.92216229  1.7267882
  -0.57185157]
 ...
 [-0.91281074 -0.09840151 -0.01321775 ...  0.92216229 -0.57910982
  -0.57185157]
 [ 1.09632668 -0.66916017  1.36260706 ...  0.92216229 -0.57910982
  -0.57185157]
 [ 2.05453067 -0.66916017  1.36260706 ...  0.92216229  1.7267882
  -0.57185157]]
```

```
classifier = Sequential()

classifier.add(Input(shape=(10,)))

classifier.add(Dense(units = 6, kernel_initializer = 'he_uniform',
activation = 'relu'))
classifier.add(Dense(units = 6, kernel_initializer = 'he_uniform',
activation = 'relu'))
classifier.add(Dense(units = 1, kernel_initializer = 'glorot_uniform',
activation = 'sigmoid'))
classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy',
metrics = ['accuracy'])

classifier.summary()
```

```
Model: "sequential"

┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━
━━━━━┓
┃ Layer (type)                     ┃ Output Shape            ┃
Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━
━━━━━┩
│ dense (Dense)                    │ (None, 6)               │
```

```
66 │
├─────────────────────────────────┼────────────────────────┼──────────
─────┤
│ dense_1 (Dense)                 │ (None, 6)              │
42 │
├─────────────────────────────────┼────────────────────────┼──────────
─────┤
│ dense_2 (Dense)                 │ (None, 1)              │
7 │
└─────────────────────────────────┴────────────────────────┴──────────
─────┘

 Total params: 115 (460.00 B)

 Trainable params: 115 (460.00 B)

 Non-trainable params: 0 (0.00 B)
```

```
classifier.fit(x_train, y_train, batch_size = 10, epochs = 50)
```

```
Epoch 1/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 5s 2ms/step - accuracy: 0.7366 - loss:
0.5844
Epoch 2/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8060 - loss:
0.4604
Epoch 3/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8200 - loss:
0.4184
Epoch 4/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8311 - loss:
0.3982
Epoch 5/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8389 - loss:
0.3830
Epoch 6/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8451 - loss:
0.3715
Epoch 7/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8479 - loss:
0.3627
Epoch 8/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8499 - loss:
0.3560
Epoch 9/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8530 - loss:
0.3524
Epoch 10/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8559 - loss:
0.3486
```

```
Epoch 11/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8551 - loss:
0.3470
Epoch 12/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8533 - loss:
0.3446
Epoch 13/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8554 - loss:
0.3436
Epoch 14/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8571 - loss:
0.3423
Epoch 15/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8584 - loss:
0.3406
Epoch 16/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8563 - loss:
0.3401
Epoch 17/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8567 - loss:
0.3393
Epoch 18/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8563 - loss:
0.3381
Epoch 19/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8590 - loss:
0.3381
Epoch 20/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 2s 2ms/step - accuracy: 0.8589 - loss:
0.3374
Epoch 21/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8613 - loss:
0.3368
Epoch 22/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8600 - loss:
0.3366
Epoch 23/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8599 - loss:
0.3361
Epoch 24/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8573 - loss:
0.3353
Epoch 25/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8613 - loss:
0.3355
Epoch 26/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8607 - loss:
0.3352
Epoch 27/50
```

```
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8600 - loss:
0.3353
Epoch 28/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8603 - loss:
0.3347
Epoch 29/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8619 - loss:
0.3341
Epoch 30/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8610 - loss:
0.3341
Epoch 31/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8610 - loss:
0.3341
Epoch 32/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 2s 2ms/step - accuracy: 0.8619 - loss:
0.3331
Epoch 33/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8611 - loss:
0.3340
Epoch 34/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8629 - loss:
0.3337
Epoch 35/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8613 - loss:
0.3335
Epoch 36/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - accuracy: 0.8610 - loss:
0.3330
Epoch 37/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8599 - loss:
0.3328
Epoch 38/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8603 - loss:
0.3328
Epoch 39/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8600 - loss:
0.3327
Epoch 40/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 2s 2ms/step - accuracy: 0.8607 - loss:
0.3331
Epoch 41/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 2s 2ms/step - accuracy: 0.8617 - loss:
0.3336
Epoch 42/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.8623 - loss:
0.3329
Epoch 43/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8613 - loss:
```

```
0.3328
Epoch 44/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8593 - loss:
0.3332
Epoch 45/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.8620 - loss:
0.3319
Epoch 46/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.8629 - loss:
0.3324
Epoch 47/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8616 - loss:
0.3316
Epoch 48/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8619 - loss:
0.3328
Epoch 49/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8613 - loss:
0.3323
Epoch 50/50
700/700 ━━━━━━━━━━━━━━━━━━━━ 2s 2ms/step - accuracy: 0.8629 - loss:
0.3318
```

```
<keras.src.callbacks.history.History at 0x261805573e0>
```

```
y_pred =classifier.predict(x_test)
y_pred = (y_pred > 0.5)
```

```
94/94 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step
```

```
cm = confusion_matrix(y_test, y_pred)
print('Confusion Matrix:\n', cm)
```

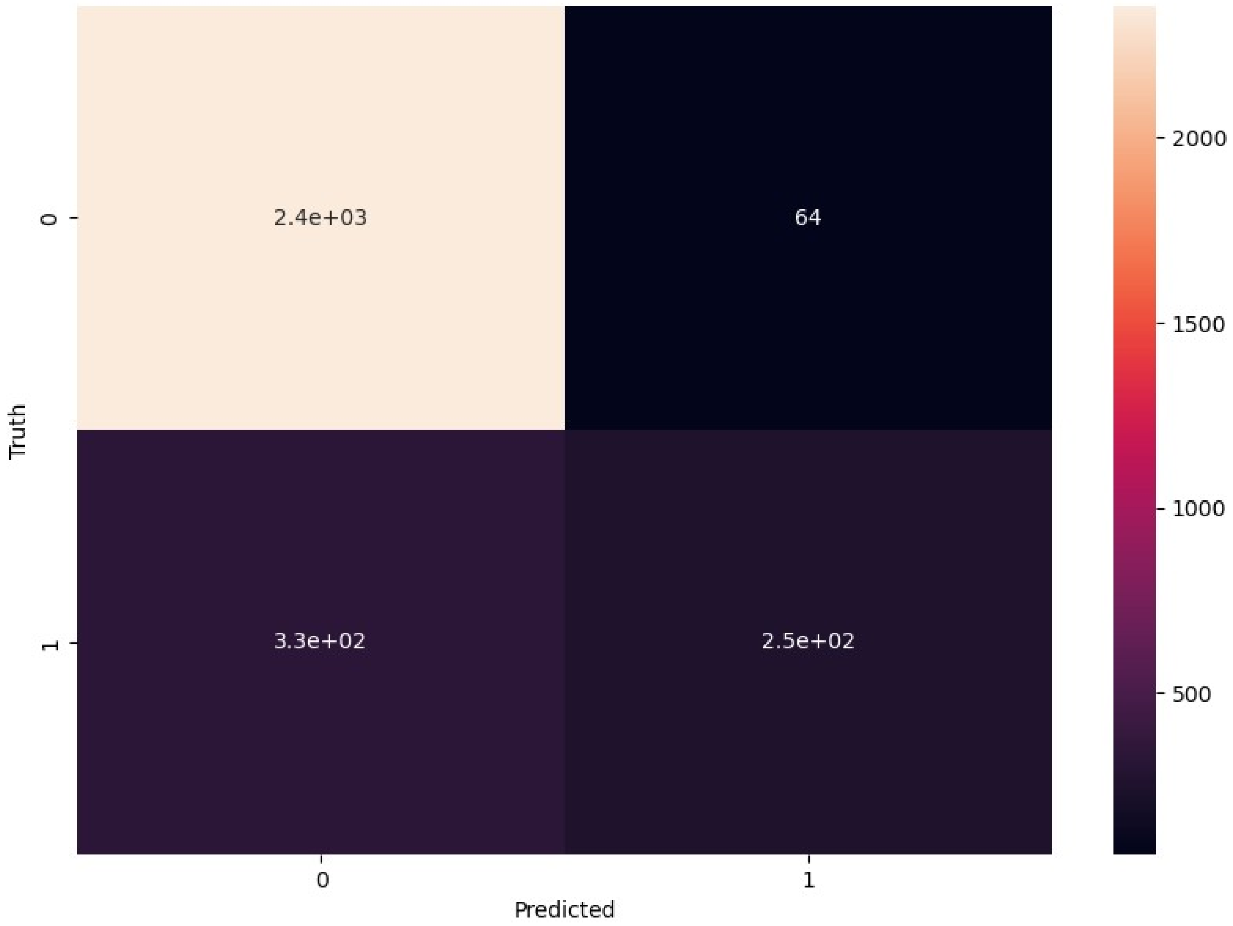
```
Confusion Matrix:
 [[2355   64]
 [ 330  251]]
```

```
acc = accuracy_score(y_test, y_pred)
print('Accuracy Score:', acc)
```

```
Accuracy Score: 0.8686666666666667
```

```
plt.figure(figsize=(10,7))
sns.heatmap(cm, annot = True)
plt.xlabel('Predicted')
plt.ylabel('Truth')
```

```
Text(95.72222222222221, 0.5, 'Truth')
```



```
print('Classification Report:\n',classification_report(y_test,
y_pred))
```

```
Classification Report:
               precision    recall  f1-score   support

           0       0.88      0.97      0.92      2419
           1       0.80      0.43      0.56       581

    accuracy                           0.87      3000
   macro avg       0.84      0.70      0.74      3000
weighted avg       0.86      0.87      0.85      3000
```