In [42]:

```
import pandas as pd
from matplotlib import pyplot as plt
import numpy as np
%matplotlib inline
import tensorflow.python as tf
from tensorflow import keras
```

In [43]:

```
data=pd.read_csv("Customer-Churn.csv")
```

In [44]:

```
data.head(7)
print(data.shape)
```

```
(7043, 21)
```

In [45]:

```
data.drop("customerID",axis='columns',inplace=True)
```

In [46]:

```
data.sample(5)
# data.dtypes
```

Out[46]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	Online
5264	Male	1	Yes	No	69	No	No phone service	DSL	No	
3256	Male	0	Yes	Yes	61	Yes	No	DSL	Yes	
566	Male	0	Yes	Yes	15	Yes	No	Fiber optic	Yes	
6267	Female	0	No	No	1	Yes	No	Fiber optic	No	
2994	Male	0	Yes	No	62	No	No phone service	DSL	Yes	

In [47]:

```
data[pd.to_numeric(data["TotalCharges"],errors="coerce").isnull()]
```

Out[47]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	Online
488	Female	0	Yes	Yes	0	No	No phone service	DSL	Yes	
753	Male	0	No	Yes	0	Yes	No	No	No internet service	No
936	Female	0	Yes	Yes	0	Yes	No	DSL	Yes	
1082	Male	0	Yes	Yes	0	Yes	Yes	No	No internet service	No
1340	Female	0	Yes	Yes	0	No	No phone service	DSL	Yes	

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	Online
3331	Male	0	Yes	Yes	0	Yes	No	No	No internet service	No
3826	Male	0	Yes	Yes	0	Yes	Yes	No	No internet service	No
4380	Female	0	Yes	Yes	0	Yes	No	No	No internet service	No
5218	Male	0	Yes	Yes	0	Yes	No	No	No internet service	No
6670	Female	0	Yes	Yes	0	Yes	Yes	DSL	No	
6754	Male	0	No	Yes	0	Yes	Yes	DSL	Yes	

In [48]:

```
data=data[data['TotalCharges']!=" "]
data.shape
data.dtypes
```

Out[48]:

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

In [49]:

```
data["TotalCharges"]=pd.to_numeric(data["TotalCharges"])
data.dtypes
#now Total charges is coming as a float datatype
```

Out[49]:

```
gender              object
SeniorCitizen        int64
Partner             object
Dependents          object
tenure               int64
PhoneService        object
MultipleLines       object
InternetService     object
OnlineSecurity      object
OnlineBackup        object
DeviceProtection    object
TechSupport         object
StreamingTV         object
StreamingMovies     object
Contract            object
PaperlessBilling    object
PaymentMethod       object
```

```
MonthlyCharges      float64
TotalCharges        float64
Churn                object
dtype: object
```

In [50]:

```
tenure_churn_no=data[data['Churn']=='No']["tenure"]
tenure_churn_yes=data[data['Churn']=='Yes']["tenure"]
```

In [51]:

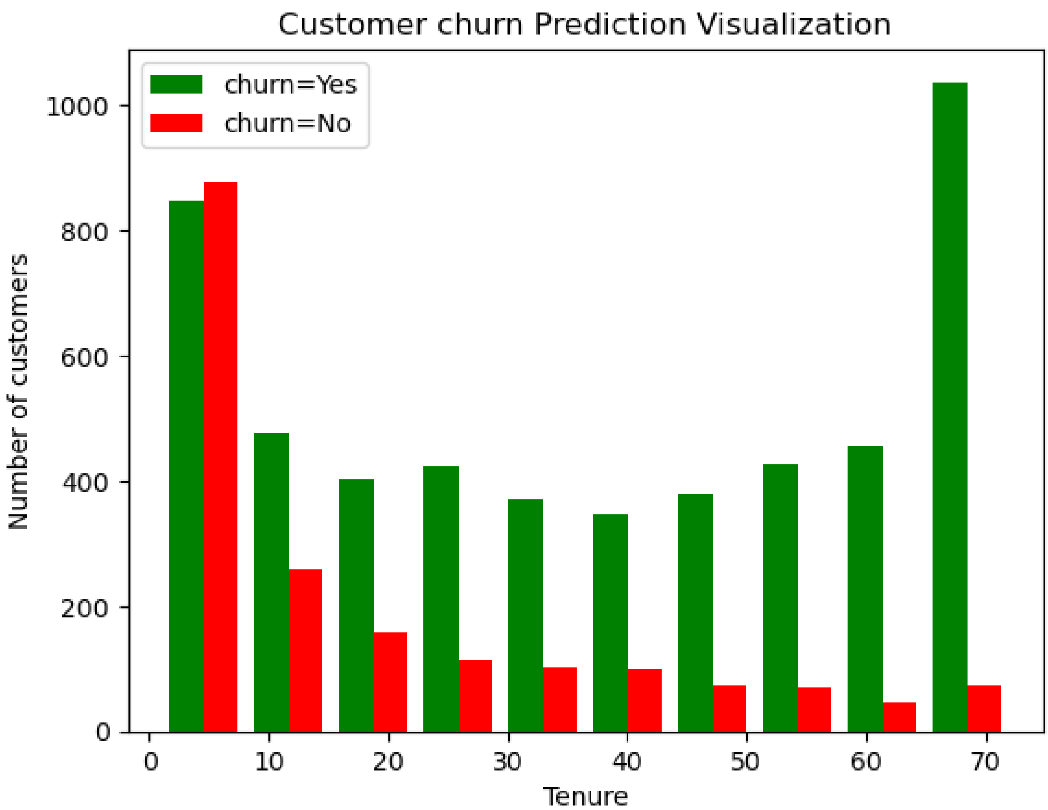
```
import matplotlib.pyplot as plt
```

In [52]:

```
plt.title("Customer churn Prediction Visualization")
plt.xlabel("Tenure")
plt.ylabel("Number of customers")
plt.hist([tenure_churn_no,tenure_churn_yes],color=["green","red"],label=["churn=Yes","ch
urn=No"])
plt.legend()
```

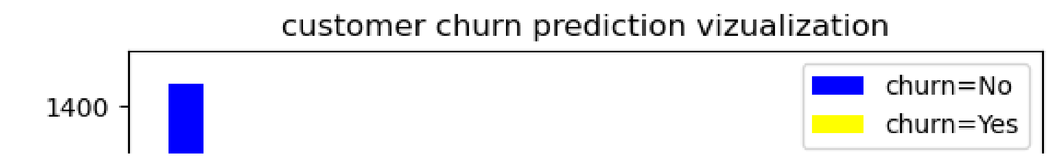
Out[52]:

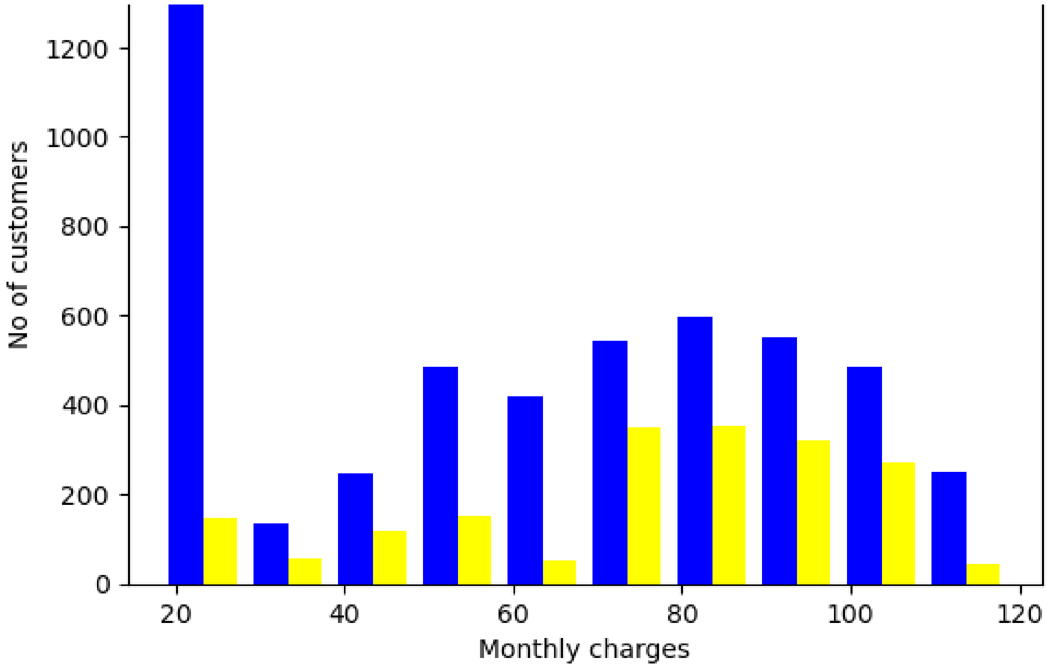
```
<matplotlib.legend.Legend at 0x1e16a87e210>
```



In [53]:

```
mc_churn_no=data[data["Churn"]=="No"]["MonthlyCharges"]
mc_churn_yes=data[data["Churn"]=="Yes"]["MonthlyCharges"]
plt.hist([mc_churn_no,mc_churn_yes],color=["blue","yellow"],label=["churn=No","churn=Yes
"])
plt.title("customer churn prediction vizualization")
plt.xlabel("Monthly charges")
plt.ylabel("No of customers")
plt.legend()
plt.show()
```





In [54]:

```
for column in data:
    print(f'{column} : {data[column].unique()}')
```

```
gender : ['Female' 'Male']
SeniorCitizen : [0 1]
Partner : ['Yes' 'No']
Dependents : ['No' 'Yes']
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 39]
PhoneService : ['No' 'Yes']
MultipleLines : ['No phone service' 'No' 'Yes']
InternetService : ['DSL' 'Fiber optic' 'No']
OnlineSecurity : ['No' 'Yes' 'No internet service']
OnlineBackup : ['Yes' 'No' 'No internet service']
DeviceProtection : ['No' 'Yes' 'No internet service']
TechSupport : ['No' 'Yes' 'No internet service']
StreamingTV : ['No' 'Yes' 'No internet service']
StreamingMovies : ['No' 'Yes' 'No internet service']
Contract : ['Month-to-month' 'One year' 'Two year']
PaperlessBilling : ['Yes' 'No']
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 : ['No' 'Yes']
```

In [55]:

```
data.replace("No internet service","No",inplace=True)
data.replace("No phone service","No",inplace=True)
```

In [56]:

```
for column in data:
    print(f'{column} : {data[column].unique()}')
```

```
gender : ['Female' 'Male']
SeniorCitizen : [0 1]
Partner : ['Yes' 'No']
Dependents : ['No' 'Yes']
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 39]
PhoneService : ['No' 'Yes']
MultipleLines : ['No' 'Yes']
InternetService : ['DSL' 'Fiber optic' 'No']
```

```
OnlineSecurity : ['No' 'Yes']
OnlineBackup : ['Yes' 'No']
DeviceProtection : ['No' 'Yes']
TechSupport : ['No' 'Yes']
StreamingTV : ['No' 'Yes']
StreamingMovies : ['No' 'Yes']
Contract : ['Month-to-month' 'One year' 'Two year']
PaperlessBilling : ['Yes' 'No']
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 : ['No' 'Yes']
```

In [57]:

```
yes_no_columns=[
"SeniorCitizen",
"Partner",
"Dependents",
"PhoneService",
"MultipleLines",
"OnlineSecurity",
"OnlineBackup",
"DeviceProtection",
"TechSupport",
"StreamingTV",
"StreamingMovies",
"PaperlessBilling",
"Churn"]
data["PaperlessBilling"].replace("Yes",1,inplace=True)
data["PaperlessBilling"].replace("No",0,inplace=True)
for col in yes_no_columns:
    data[col].replace("Yes",1,inplace=True)
    data[col].replace("No",0,inplace=True)
```

In [58]:

```
for column in data:
    print(f'{column} : {data[column].unique()}')
```

```
gender : ['Female' 'Male']
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 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 [59]:

```
data["gender"].replace("Female",0,inplace=True)
data["gender"].replace("Male",1,inplace=True)
```

In [60]:

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

```
gender:[0 1]
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 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 [61]:

```python
data
```

Out[61]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	Online
0	0	0	1	0	1	0	0	DSL	0	
1	1	0	0	0	34	1	0	DSL	1	
2	1	0	0	0	2	1	0	DSL	1	
3	1	0	0	0	45	0	0	DSL	1	
4	0	0	0	0	2	1	0	Fiber optic	0	
...	...	...	...	...	...	...	...	...	...	
7038	1	0	1	1	24	1	1	DSL	1	
7039	0	0	1	1	72	1	1	Fiber optic	0	
7040	0	0	1	1	11	0	0	DSL	1	
7041	1	1	1	0	4	1	1	Fiber optic	0	
7042	1	0	0	0	66	1	0	Fiber optic	1	

7032 rows × 20 columns

In [62]:

```
df=pd.get_dummies(data=data,columns=["InternetService","Contract","PaymentMethod"])
df.replace(True,1,inplace=True)
df.replace(False,0,inplace=True)
```

In [63]:

```
df.dtypes
```

Out[63]:

```
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 [64]:

```
df.sample(3)
```

Out[64]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	OnlineSecurity	OnlineBackup	DeviceF
6090	1	0	1	1	68	1	0	1	1	
6421	0	0	0	0	21	1	0	1	1	
891	1	0	1	1	50	1	1	0	1	

3 rows × 27 columns

In [65]:

```
#data scaling
cols_to_scale=["tenure","MonthlyCharges","TotalCharges"]
```

In [66]:

```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
#converting values present in tenure,monthlycharges,totalcharges into range 0 to 1
df[cols_to_scale]=scaler.fit_transform(df[cols_to_scale])
```

In [67]:

```
df.sample(5)
```

Out[67]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	OnlineSecurity	OnlineBackup	Devic
1500	0	0	0	0	0.507042	0	0	0	0	
5922	1	0	1	1	1.000000	1	1	1	0	
2948	1	1	1	0	0.563380	1	1	0	0	
1116	0	0	0	0	0.521127	1	1	0	0	
2301	0	0	1	0	0.098592	1	1	0	0	

5 rows × 27 columns

In [68]:

```
for col in df:
    print (f'{col}: {df[col].unique()}')
#we can se walue scale between 0 to 1
```

```
gender: [0 1]
SeniorCitizen: [0 1]
Partner: [1 0]
Dependents: [0 1]
tenure: [0.         0.46478873 0.01408451 0.61971831 0.09859155 0.29577465
 0.12676056 0.38028169 0.85915493 0.16901408 0.21126761 0.8028169
 0.67605634 0.33802817 0.95774648 0.71830986 0.98591549 0.28169014
 0.15492958 0.4084507  0.64788732 1.         0.22535211 0.36619718
 0.05633803 0.63380282 0.14084507 0.97183099 0.87323944 0.5915493
 0.1971831  0.83098592 0.23943662 0.91549296 0.11267606 0.02816901
 0.42253521 0.69014085 0.88732394 0.77464789 0.08450704 0.57746479
 0.47887324 0.66197183 0.3943662  0.90140845 0.52112676 0.94366197
 0.43661972 0.76056338 0.50704225 0.49295775 0.56338028 0.07042254
 0.04225352 0.45070423 0.92957746 0.30985915 0.78873239 0.84507042
 0.18309859 0.26760563 0.73239437 0.54929577 0.81690141 0.32394366
 0.6056338  0.25352113 0.74647887 0.70422535 0.35211268 0.53521127]
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 0.60149254]
TotalCharges: [0.0012751  0.21586661 0.01031041 ... 0.03780868 0.03321025 0.78764136]
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 [69]:

```
x=df.drop("Churn",axis=1)
y=df["Churn"]
```

In [70]:

```
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 [71]:

```
import tensorflow as tf
from tensorflow import keras
model=keras.Sequential([
    keras.layers.Dense(20,input_shape=(26,),activation="relu"),
    keras.layers.Dense(20,activation="relu"),
    keras.layers.Dense(1,activation="sigmoid")
])
```

In [72]:

```
model.compile(optimizer="adam",
             loss="binary_crossentropy",#log loss function
             metrics=["accuracy"])#number of currectly classified examples/total number
of examples
```

In [73]:

```
model.fit(x_train,y_train,epochs=200)
```

```
Epoch 1/200
176/176 [==============================] - 1s 2ms/step - loss: 0.5260 - accuracy: 0.7157
Epoch 2/200
176/176 [==============================] - 0s 2ms/step - loss: 0.4330 - accuracy: 0.7899
Epoch 3/200
176/176 [==============================] - 0s 2ms/step - loss: 0.4238 - accuracy: 0.7956
Epoch 4/200
176/176 [==============================] - 0s 1ms/step - loss: 0.4189 - accuracy: 0.8014
Epoch 5/200
176/176 [==============================] - 0s 1ms/step - loss: 0.4156 - accuracy: 0.8032
Epoch 6/200
176/176 [==============================] - 0s 2ms/step - loss: 0.4129 - accuracy: 0.8002
Epoch 7/200
176/176 [==============================] - 0s 2ms/step - loss: 0.4110 - accuracy: 0.8053
Epoch 8/200
176/176 [==============================] - 0s 1ms/step - loss: 0.4091 - accuracy: 0.8050
Epoch 9/200
176/176 [==============================] - 0s 2ms/step - loss: 0.4075 - accuracy: 0.8057
Epoch 10/200
176/176 [==============================] - 0s 2ms/step - loss: 0.4060 - accuracy: 0.8105
Epoch 11/200
176/176 [==============================] - 0s 2ms/step - loss: 0.4049 - accuracy: 0.8101
Epoch 12/200
176/176 [==============================] - 0s 1ms/step - loss: 0.4033 - accuracy: 0.8121
Epoch 13/200
176/176 [==============================] - 0s 2ms/step - loss: 0.4028 - accuracy: 0.8121
Epoch 14/200
176/176 [==============================] - 0s 2ms/step - loss: 0.4010 - accuracy: 0.8137
Epoch 15/200
176/176 [==============================] - 0s 1ms/step - loss: 0.4006 - accuracy: 0.8121
Epoch 16/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3980 - accuracy: 0.8148
Epoch 17/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3977 - accuracy: 0.8133
Epoch 18/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3955 - accuracy: 0.8137
Epoch 19/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3950 - accuracy: 0.8132
Epoch 20/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3936 - accuracy: 0.8158
Epoch 21/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3930 - accuracy: 0.8151
Epoch 22/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3918 - accuracy: 0.8153
Epoch 23/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3910 - accuracy: 0.8188
Epoch 24/200
```

```
176/176 [==============================] - 0s 2ms/step - loss: 0.3902 - accuracy: 0.8171
Epoch 25/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3889 - accuracy: 0.8169
Epoch 26/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3877 - accuracy: 0.8180
Epoch 27/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3873 - accuracy: 0.8187
Epoch 28/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3855 - accuracy: 0.8206
Epoch 29/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3856 - accuracy: 0.8201
Epoch 30/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3849 - accuracy: 0.8190
Epoch 31/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3853 - accuracy: 0.8178
Epoch 32/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3846 - accuracy: 0.8192
Epoch 33/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3830 - accuracy: 0.8210
Epoch 34/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3820 - accuracy: 0.8210
Epoch 35/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3809 - accuracy: 0.8204
Epoch 36/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3800 - accuracy: 0.8228
Epoch 37/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3804 - accuracy: 0.8204
Epoch 38/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3814 - accuracy: 0.8222
Epoch 39/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3799 - accuracy: 0.8242
Epoch 40/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3790 - accuracy: 0.8215
Epoch 41/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3784 - accuracy: 0.8220
Epoch 42/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3771 - accuracy: 0.8236
Epoch 43/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3765 - accuracy: 0.8233
Epoch 44/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3769 - accuracy: 0.8233
Epoch 45/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3767 - accuracy: 0.8212
Epoch 46/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3757 - accuracy: 0.8244
Epoch 47/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3744 - accuracy: 0.8245
Epoch 48/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3751 - accuracy: 0.8267
Epoch 49/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3732 - accuracy: 0.8245
Epoch 50/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3743 - accuracy: 0.8244
Epoch 51/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3726 - accuracy: 0.8245
Epoch 52/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3716 - accuracy: 0.8258
Epoch 53/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3719 - accuracy: 0.8240
Epoch 54/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3717 - accuracy: 0.8247
Epoch 55/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3706 - accuracy: 0.8256
Epoch 56/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3703 - accuracy: 0.8277
Epoch 57/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3694 - accuracy: 0.8251
Epoch 58/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3693 - accuracy: 0.8268
Epoch 59/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3688 - accuracy: 0.8297
Epoch 60/200
```

```
176/176 [==============================] - 0s 1ms/step - loss: 0.3681 - accuracy: 0.8276
Epoch 61/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3675 - accuracy: 0.8286
Epoch 62/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3687 - accuracy: 0.8245
Epoch 63/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3684 - accuracy: 0.8256
Epoch 64/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3679 - accuracy: 0.8300
Epoch 65/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3676 - accuracy: 0.8288
Epoch 66/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3664 - accuracy: 0.8249
Epoch 67/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3660 - accuracy: 0.8274
Epoch 68/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3650 - accuracy: 0.8274
Epoch 69/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3654 - accuracy: 0.8295
Epoch 70/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3647 - accuracy: 0.8251
Epoch 71/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3650 - accuracy: 0.8277
Epoch 72/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3640 - accuracy: 0.8295
Epoch 73/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3635 - accuracy: 0.8309
Epoch 74/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3641 - accuracy: 0.8276
Epoch 75/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3635 - accuracy: 0.8295
Epoch 76/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3623 - accuracy: 0.8283
Epoch 77/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3629 - accuracy: 0.8297
Epoch 78/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3625 - accuracy: 0.8304
Epoch 79/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3620 - accuracy: 0.8309
Epoch 80/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3618 - accuracy: 0.8290
Epoch 81/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3606 - accuracy: 0.8299
Epoch 82/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3606 - accuracy: 0.8286
Epoch 83/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3605 - accuracy: 0.8293
Epoch 84/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3606 - accuracy: 0.8300
Epoch 85/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3593 - accuracy: 0.8299
Epoch 86/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3601 - accuracy: 0.8308
Epoch 87/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3583 - accuracy: 0.8316
Epoch 88/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3600 - accuracy: 0.8316
Epoch 89/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3575 - accuracy: 0.8313
Epoch 90/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3565 - accuracy: 0.8313
Epoch 91/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3569 - accuracy: 0.8329
Epoch 92/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3587 - accuracy: 0.8315
Epoch 93/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3576 - accuracy: 0.8322
Epoch 94/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3560 - accuracy: 0.8322
Epoch 95/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3554 - accuracy: 0.8327
Epoch 96/200
```

```
176/176 [==============================] - 0s 2ms/step - loss: 0.3555 - accuracy: 0.8332
Epoch 97/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3547 - accuracy: 0.8382
Epoch 98/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3555 - accuracy: 0.8325
Epoch 99/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3539 - accuracy: 0.8334
Epoch 100/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3540 - accuracy: 0.8361
Epoch 101/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3544 - accuracy: 0.8329
Epoch 102/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3545 - accuracy: 0.8336
Epoch 103/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3537 - accuracy: 0.8354
Epoch 104/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3530 - accuracy: 0.8331
Epoch 105/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3527 - accuracy: 0.8341
Epoch 106/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3532 - accuracy: 0.8329
Epoch 107/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3519 - accuracy: 0.8366
Epoch 108/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3529 - accuracy: 0.8343
Epoch 109/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3512 - accuracy: 0.8356
Epoch 110/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3510 - accuracy: 0.8332
Epoch 111/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3514 - accuracy: 0.8318
Epoch 112/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3512 - accuracy: 0.8354
Epoch 113/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3493 - accuracy: 0.8356
Epoch 114/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3497 - accuracy: 0.8340
Epoch 115/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3498 - accuracy: 0.8364
Epoch 116/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3494 - accuracy: 0.8332
Epoch 117/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3492 - accuracy: 0.8347
Epoch 118/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3493 - accuracy: 0.8357
Epoch 119/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3497 - accuracy: 0.8373
Epoch 120/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3490 - accuracy: 0.8379
Epoch 121/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3484 - accuracy: 0.8359
Epoch 122/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3474 - accuracy: 0.8347
Epoch 123/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3476 - accuracy: 0.8395
Epoch 124/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3482 - accuracy: 0.8373
Epoch 125/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3471 - accuracy: 0.8380
Epoch 126/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3486 - accuracy: 0.8379
Epoch 127/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3474 - accuracy: 0.8341
Epoch 128/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3461 - accuracy: 0.8368
Epoch 129/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3483 - accuracy: 0.8395
Epoch 130/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3470 - accuracy: 0.8364
Epoch 131/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3471 - accuracy: 0.8341
Epoch 132/200
```

```
176/176 [==============================] - 0s 1ms/step - loss: 0.3458 - accuracy: 0.8352
Epoch 133/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3462 - accuracy: 0.8368
Epoch 134/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3460 - accuracy: 0.8388
Epoch 135/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3448 - accuracy: 0.8368
Epoch 136/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3457 - accuracy: 0.8366
Epoch 137/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3455 - accuracy: 0.8372
Epoch 138/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3448 - accuracy: 0.8398
Epoch 139/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3436 - accuracy: 0.8375
Epoch 140/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3437 - accuracy: 0.8379
Epoch 141/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3433 - accuracy: 0.8372
Epoch 142/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3425 - accuracy: 0.8409
Epoch 143/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3416 - accuracy: 0.8420
Epoch 144/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3431 - accuracy: 0.8372
Epoch 145/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3427 - accuracy: 0.8370
Epoch 146/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3420 - accuracy: 0.8398
Epoch 147/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3421 - accuracy: 0.8396
Epoch 148/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3426 - accuracy: 0.8388
Epoch 149/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3413 - accuracy: 0.8380
Epoch 150/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3426 - accuracy: 0.8368
Epoch 151/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3418 - accuracy: 0.8391
Epoch 152/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3412 - accuracy: 0.8373
Epoch 153/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3403 - accuracy: 0.8404
Epoch 154/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3397 - accuracy: 0.8382
Epoch 155/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3407 - accuracy: 0.8421
Epoch 156/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3404 - accuracy: 0.8370
Epoch 157/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3396 - accuracy: 0.8386
Epoch 158/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3405 - accuracy: 0.8398
Epoch 159/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3383 - accuracy: 0.8372
Epoch 160/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3396 - accuracy: 0.8373
Epoch 161/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3390 - accuracy: 0.8402
Epoch 162/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3381 - accuracy: 0.8380
Epoch 163/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3403 - accuracy: 0.8368
Epoch 164/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3372 - accuracy: 0.8398
Epoch 165/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3386 - accuracy: 0.8372
Epoch 166/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3377 - accuracy: 0.8402
Epoch 167/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3368 - accuracy: 0.8405
Epoch 168/200
```

```
176/176 [==============================] - 0s 1ms/step - loss: 0.3363 - accuracy: 0.8441
Epoch 169/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3369 - accuracy: 0.8386
Epoch 170/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3366 - accuracy: 0.8428
Epoch 171/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3370 - accuracy: 0.8380
Epoch 172/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3361 - accuracy: 0.8411
Epoch 173/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3359 - accuracy: 0.8411
Epoch 174/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3373 - accuracy: 0.8416
Epoch 175/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3359 - accuracy: 0.8393
Epoch 176/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3359 - accuracy: 0.8372
Epoch 177/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3353 - accuracy: 0.8453
Epoch 178/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3339 - accuracy: 0.8430
Epoch 179/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3340 - accuracy: 0.8425
Epoch 180/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3351 - accuracy: 0.8425
Epoch 181/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3350 - accuracy: 0.8402
Epoch 182/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3338 - accuracy: 0.8412
Epoch 183/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3336 - accuracy: 0.8423
Epoch 184/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3344 - accuracy: 0.8411
Epoch 185/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3344 - accuracy: 0.8416
Epoch 186/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3332 - accuracy: 0.8402
Epoch 187/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3328 - accuracy: 0.8441
Epoch 188/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3318 - accuracy: 0.8414
Epoch 189/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3345 - accuracy: 0.8430
Epoch 190/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3333 - accuracy: 0.8425
Epoch 191/200
176/176 [==============================] - 0s 2ms/step - loss: 0.3314 - accuracy: 0.8448
Epoch 192/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3323 - accuracy: 0.8436
Epoch 193/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3347 - accuracy: 0.8407
Epoch 194/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3316 - accuracy: 0.8475
Epoch 195/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3328 - accuracy: 0.8407
Epoch 196/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3328 - accuracy: 0.8439
Epoch 197/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3316 - accuracy: 0.8443
Epoch 198/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3316 - accuracy: 0.8466
Epoch 199/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3304 - accuracy: 0.8432
Epoch 200/200
176/176 [==============================] - 0s 1ms/step - loss: 0.3304 - accuracy: 0.8428
```

Out[73]:

```
<keras.src.callbacks.History at 0x1e16c8049d0>
```

In [81]:

```
model.evaluate(x_test,y_test)
```

```
44/44 [==============================] - 0s 2ms/step - loss: 0.5268 - accuracy: 0.7683
```

Out[81]:

```
[0.5267581939697266, 0.7683013677597046]
```

In [82]:

```
y_predicted=model.predict(x_test)
```

```
44/44 [==============================] - 0s 1ms/step
```

In [83]:

```
y_predicted=y_predicted.reshape(1407,)
y_predicted
```

Out[83]:

```
array([0.13091475, 0.3511366 , 0.00095845, ..., 0.739014  , 0.7248184 ,
       0.7483483 ], dtype=float32)
```

In [84]:

```
list=[]
for i in y_predicted:
    if i<0.5:
        list.append(0)
    else:
        list.append(1)
```

In [85]:

```
y_predicted=list
y_predicted[:10]
```

Out[85]:

```
[0, 0, 0, 1, 1, 1, 0, 0, 0, 0]
```

In [86]:

```
from sklearn.metrics import confusion_matrix,classification_report
```

In [89]:

```
print (classification_report(y_test,y_predicted))
```

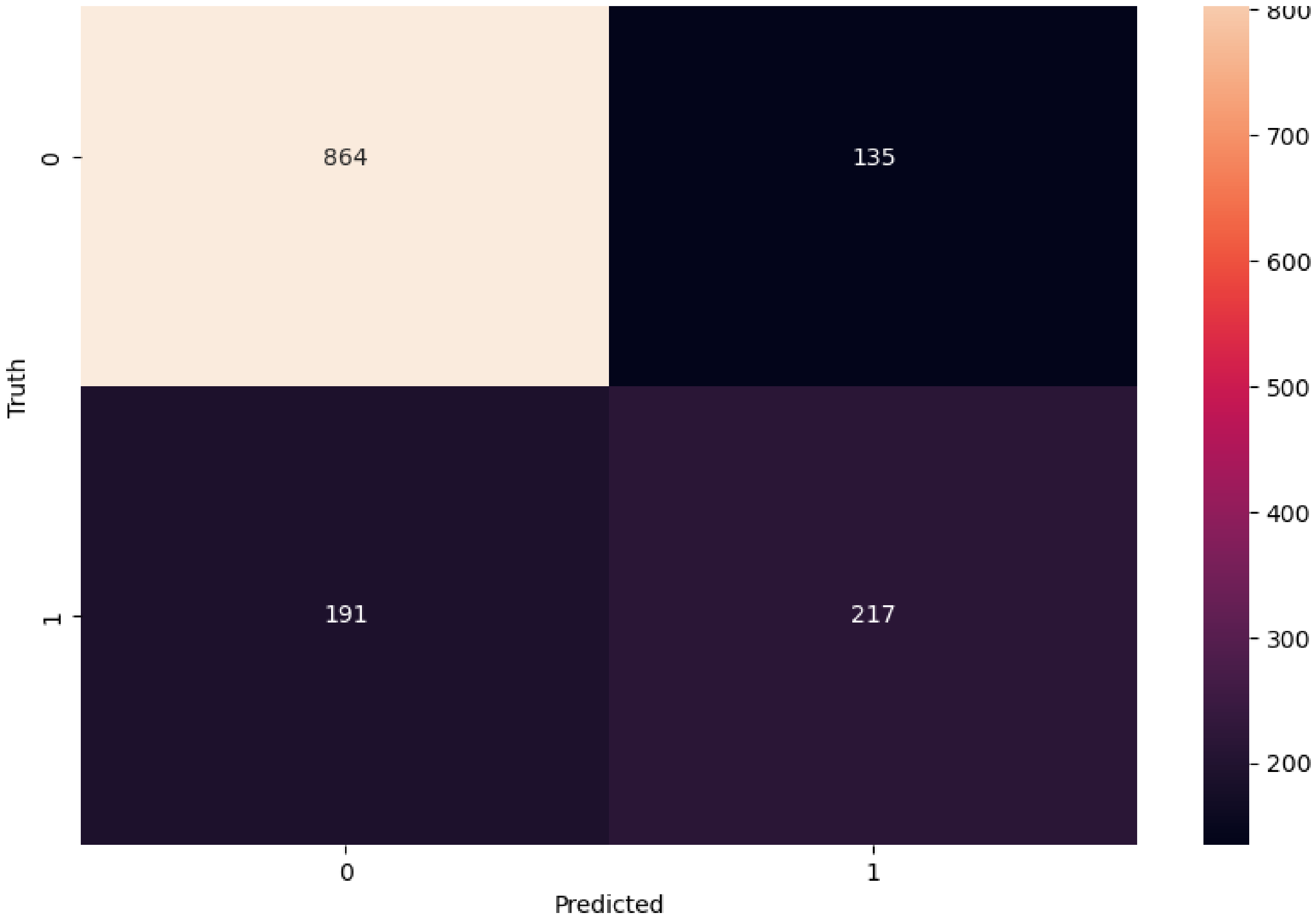
```
              precision    recall  f1-score   support

           0       0.82      0.86      0.84       999
           1       0.62      0.53      0.57       408

    accuracy                           0.77      1407
   macro avg       0.72      0.70      0.71      1407
weighted avg       0.76      0.77      0.76      1407
```

In [90]:

```
import seaborn as sb
cm=tf.math.confusion_matrix(y_test,y_predicted)
plt.figure(figsize=(10,7))
sb.heatmap(cm,annot=True,fmt="d")
plt.xlabel("Predicted")
plt.ylabel("Truth")
plt.show()
```


864
135
191
217
0
1
Truth
0
1
Predicted
800
700
600
500
400
300
200

In [ ]: