```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
import tensorflow as tf
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense
from sklearn.metrics import classification_report
```

```
df=df=pd.read_csv("airlines_delay.csv")
```

```
df.head()
```

	Flight	Time	Length	Airline	AirportFrom	AirportTo	DayOfWeek	Class
0	2313.0	1296.0	141.0	DL	ATL	HOU	1	0
1	6948.0	360.0	146.0	OO	COS	ORD	4	0
2	1247.0	1170.0	143.0	B6	BOS	CLT	3	0
3	31.0	1410.0	344.0	US	OGG	PHX	6	0
4	563.0	692.0	98.0	FL	BMI	ATL	4	0

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 539382 entries, 0 to 539381
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype  
---  ------       --------------   -----  
 0   Flight       539382 non-null  float64
 1   Time         539382 non-null  float64
 2   Length       539382 non-null  float64
 3   Airline      539382 non-null  object 
 4   AirportFrom  539382 non-null  object 
 5   AirportTo    539382 non-null  object 
 6   DayOfWeek    539382 non-null  int64  
 7   Class        539382 non-null  int64  
dtypes: float64(3), int64(2), object(3)
memory usage: 32.9+ MB
```

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

```
Flight         0
Time           0
Length         0
Airline        0
AirportFrom    0
AirportTo      0
DayOfWeek      0
Class          0
dtype: int64
```

```
df['Class'].value_counts()
```

```
0    299118
1    240264
Name: Class, dtype: int64
```

```
df.drop("Flight",axis=1,inplace=True)

df.head()
```

	Time	Length	Airline	AirportFrom	AirportTo	DayOfWeek	Class
0	1296.0	141.0	DL	ATL	HOU	1	0
1	360.0	146.0	OO	COS	ORD	4	0
2	1170.0	143.0	B6	BOS	CLT	3	0
3	1410.0	344.0	US	OGG	PHX	6	0
4	692.0	98.0	FL	BMI	ATL	4	0

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()


df['Airline']=le.fit_transform(df['Airline'])
df['AirportFrom']=le.fit_transform(df['AirportFrom'])
df['AirportTo']=le.fit_transform(df['AirportTo'])


df.head()
```

	Time	Length	Airline	AirportFrom	AirportTo	DayOfWeek	Class
0	1296.0	141.0	5	16	129	1	0
1	360.0	146.0	12	65	208	4	0
2	1170.0	143.0	3	35	60	3	0
3	1410.0	344.0	14	203	217	6	0
4	692.0	98.0	8	32	16	4	0

```
x=df.iloc[:,:-1].values
x
```

```
array([[1.296e+03, 1.410e+02, 5.000e+00, 1.600e+01, 1.290e+02, 1.000e+00],
       [3.600e+02, 1.460e+02, 1.200e+01, 6.500e+01, 2.080e+02, 4.000e+00],
       [1.170e+03, 1.430e+02, 3.000e+00, 3.500e+01, 6.000e+01, 3.000e+00],
       ...,
       [8.270e+02, 7.400e+01, 6.000e+00, 4.700e+01, 1.600e+01, 2.000e+00],
       [7.150e+02, 6.500e+01, 1.500e+01, 4.500e+01, 4.300e+01, 4.000e+00],
       [7.700e+02, 5.500e+01, 1.200e+01, 6.700e+01, 7.900e+01, 2.000e+00]])
```

```
y=df['Class'].values
y
```

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

```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.30,random_state=1)


from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
xtrain=sc.fit_transform(xtrain)
xtest=sc.transform(xtest)


ann=Sequential()
ann.add(Dense(units=6,activation="relu"))
ann.add(Dense(units=6,activation="relu"))
```

```
ann.add(Dense(units=1,activation="sigmoid"))


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


ann.fit(xtrain,ytrain,epochs=50,batch_size=60)
```

```
6293/6293 [==============================] - 15s 2ms/step - loss: 0.6342 - accuracy: 0.6380
Epoch 23/50
6293/6293 [==============================] - 15s 2ms/step - loss: 0.6341 - accuracy: 0.6382
Epoch 24/50
6293/6293 [==============================] - 16s 3ms/step - loss: 0.6342 - accuracy: 0.6377
Epoch 25/50
6293/6293 [==============================] - 15s 2ms/step - loss: 0.6342 - accuracy: 0.6381
Epoch 26/50
6293/6293 [==============================] - 15s 2ms/step - loss: 0.6341 - accuracy: 0.6381
Epoch 27/50
6293/6293 [==============================] - 15s 2ms/step - loss: 0.6341 - accuracy: 0.6380
Epoch 28/50
6293/6293 [==============================] - 15s 2ms/step - loss: 0.6342 - accuracy: 0.6373
Epoch 29/50
6293/6293 [==============================] - 16s 3ms/step - loss: 0.6342 - accuracy: 0.6378
Epoch 30/50
6293/6293 [==============================] - 15s 2ms/step - loss: 0.6341 - accuracy: 0.6382
Epoch 31/50
6293/6293 [==============================] - 15s 2ms/step - loss: 0.6342 - accuracy: 0.6377
Epoch 32/50
6293/6293 [==============================] - 15s 2ms/step - loss: 0.6341 - accuracy: 0.6383
Epoch 33/50
6293/6293 [==============================] - 16s 3ms/step - loss: 0.6342 - accuracy: 0.6379
Epoch 34/50
6293/6293 [==============================] - 16s 3ms/step - loss: 0.6340 - accuracy: 0.6378
Epoch 35/50
6293/6293 [==============================] - 16s 3ms/step - loss: 0.6341 - accuracy: 0.6381
Epoch 36/50
6293/6293 [==============================] - 16s 2ms/step - loss: 0.6340 - accuracy: 0.6383
Epoch 37/50
6293/6293 [==============================] - 17s 3ms/step - loss: 0.6341 - accuracy: 0.6380
Epoch 38/50
6293/6293 [==============================] - 15s 2ms/step - loss: 0.6341 - accuracy: 0.6378
Epoch 39/50
6293/6293 [==============================] - 16s 3ms/step - loss: 0.6341 - accuracy: 0.6376
Epoch 40/50
6293/6293 [==============================] - 16s 2ms/step - loss: 0.6340 - accuracy: 0.6379
Epoch 41/50
6293/6293 [==============================] - 17s 3ms/step - loss: 0.6340 - accuracy: 0.6374
Epoch 42/50
6293/6293 [==============================] - 16s 3ms/step - loss: 0.6341 - accuracy: 0.6376
Epoch 43/50
6293/6293 [==============================] - 22s 4ms/step - loss: 0.6340 - accuracy: 0.6381
Epoch 44/50
6293/6293 [==============================] - 23s 4ms/step - loss: 0.6340 - accuracy: 0.6379
Epoch 45/50
6293/6293 [==============================] - 18s 3ms/step - loss: 0.6341 - accuracy: 0.6380
Epoch 46/50
6293/6293 [==============================] - 16s 3ms/step - loss: 0.6340 - accuracy: 0.6374
Epoch 47/50
6293/6293 [==============================] - 17s 3ms/step - loss: 0.6340 - accuracy: 0.6379
Epoch 48/50
6293/6293 [==============================] - 21s 3ms/step - loss: 0.6340 - accuracy: 0.6379
Epoch 49/50
6293/6293 [==============================] - 17s 3ms/step - loss: 0.6340 - accuracy: 0.6378
Epoch 50/50
6293/6293 [==============================] - 17s 3ms/step - loss: 0.6341 - accuracy: 0.6378
<keras.callbacks.History at 0x7f94f47170d0>
```

```
ypred=ann.predict(xtest)
ypred=ypred>0.5
from sklearn.metrics import classification_report
```

```
5057/5057 [==============================] - 9s 2ms/step
```

```
print(classification_report(ypred,ytest))
```

```
                  precision    recall  f1-score   support

           False       0.84      0.63      0.72    118385
            True       0.40      0.66      0.50     43430

        accuracy                           0.64    161815
       macro avg       0.62      0.65      0.61    161815
    weighted avg       0.72      0.64      0.66    161815
```

```
from tensorflow.keras.callbacks import EarlyStopping
early_stop = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=25)
```

```
ann.fit(xtrain,ytrain,epochs=100,validation_data=(xtest, ytest),verbose=1,batch_size=30,callbacks=[early_stop])
```

```
Epoch 1/100
12586/12586 [==============================] - 42s 3ms/step - loss: 0.6344 - accuracy: 0.6370 - val_loss: 0.6327 -
Epoch 2/100
12586/12586 [==============================] - 43s 3ms/step - loss: 0.6345 - accuracy: 0.6372 - val_loss: 0.6341 -
Epoch 3/100
12586/12586 [==============================] - 43s 3ms/step - loss: 0.6343 - accuracy: 0.6380 - val_loss: 0.6339 -
Epoch 4/100
12586/12586 [==============================] - 42s 3ms/step - loss: 0.6344 - accuracy: 0.6376 - val_loss: 0.6334 -
Epoch 5/100
12586/12586 [==============================] - 43s 3ms/step - loss: 0.6345 - accuracy: 0.6377 - val_loss: 0.6333 -
Epoch 6/100
12586/12586 [==============================] - 40s 3ms/step - loss: 0.6344 - accuracy: 0.6372 - val_loss: 0.6339 -
Epoch 7/100
12586/12586 [==============================] - 40s 3ms/step - loss: 0.6343 - accuracy: 0.6375 - val_loss: 0.6336 -
Epoch 8/100
12586/12586 [==============================] - 43s 3ms/step - loss: 0.6344 - accuracy: 0.6376 - val_loss: 0.6341 -
Epoch 9/100
12586/12586 [==============================] - 43s 3ms/step - loss: 0.6344 - accuracy: 0.6378 - val_loss: 0.6349 -
Epoch 10/100
12586/12586 [==============================] - 42s 3ms/step - loss: 0.6344 - accuracy: 0.6375 - val_loss: 0.6328 -
Epoch 11/100
12586/12586 [==============================] - 42s 3ms/step - loss: 0.6344 - accuracy: 0.6379 - val_loss: 0.6335 -
Epoch 12/100
12586/12586 [==============================] - 40s 3ms/step - loss: 0.6343 - accuracy: 0.6377 - val_loss: 0.6330 -
Epoch 13/100
12586/12586 [==============================] - 43s 3ms/step - loss: 0.6345 - accuracy: 0.6376 - val_loss: 0.6334 -
Epoch 14/100
12586/12586 [==============================] - 42s 3ms/step - loss: 0.6343 - accuracy: 0.6378 - val_loss: 0.6333 -
Epoch 15/100
12586/12586 [==============================] - 43s 3ms/step - loss: 0.6344 - accuracy: 0.6378 - val_loss: 0.6334 -
Epoch 16/100
12586/12586 [==============================] - 41s 3ms/step - loss: 0.6343 - accuracy: 0.6379 - val_loss: 0.6329 -
Epoch 17/100
12586/12586 [==============================] - 42s 3ms/step - loss: 0.6344 - accuracy: 0.6374 - val_loss: 0.6328 -
Epoch 18/100
12586/12586 [==============================] - 43s 3ms/step - loss: 0.6344 - accuracy: 0.6376 - val_loss: 0.6367 -
Epoch 19/100
12586/12586 [==============================] - 42s 3ms/step - loss: 0.6343 - accuracy: 0.6379 - val_loss: 0.6344 -
Epoch 20/100
12586/12586 [==============================] - 39s 3ms/step - loss: 0.6343 - accuracy: 0.6377 - val_loss: 0.6337 -
Epoch 21/100
12586/12586 [==============================] - 39s 3ms/step - loss: 0.6343 - accuracy: 0.6379 - val_loss: 0.6340 -
Epoch 22/100
12586/12586 [==============================] - 40s 3ms/step - loss: 0.6342 - accuracy: 0.6377 - val_loss: 0.6331 -
Epoch 23/100
12586/12586 [==============================] - 42s 3ms/step - loss: 0.6342 - accuracy: 0.6380 - val_loss: 0.6338 -
Epoch 24/100
12586/12586 [==============================] - 39s 3ms/step - loss: 0.6343 - accuracy: 0.6377 - val_loss: 0.6329 -
Epoch 25/100
12586/12586 [==============================] - 39s 3ms/step - loss: 0.6343 - accuracy: 0.6376 - val_loss: 0.6331 -
```

```
Epoch 26/100
12586/12586 [==============================] - 41s 3ms/step - loss: 0.6344 - accuracy: 0.6382 - val_loss: 0.6343 -
Epoch 26: early stopping
<keras.callbacks.History at 0x7f94f4689e80>
```

```
ypred=ann.predict(xtest)
ypred=ypred>0.5
```

```
5057/5057 [==============================] - 11s 2ms/step
```

```
print(classification_report(ypred,ytest))
```

```
                  precision    recall  f1-score   support

           False       0.87      0.62      0.73    125308
            True       0.35      0.69      0.46     36507

        accuracy                           0.64    161815
       macro avg       0.61      0.66      0.60    161815
    weighted avg       0.75      0.64      0.67    161815
```