```
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
import ember
from sklearn.preprocessing import StandardScaler


from google.colab import drive

drive.mount('/content/gdrive')

dir = "/content/gdrive/MyDrive/DataScience/MidtermDataset"
```

```
Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mou
```

```
X_train, y_train, X_test, y_test = ember.read_vectorized_features(dir)
metadata_dataframe = ember.read_metadata(dir)
```

```
WARNING: EMBER feature version 2 were computed using lief version 0.9.0-
WARNING:   lief version 0.11.5-37bc2c9 found instead. There may be slight inconsistencie
WARNING:   in the feature calculations.
/usr/local/lib/python3.7/dist-packages/numpy/lib/arraysetops.py:580: FutureWarning: elem
  mask |= (ar1 == a)
```

```
labelrows = (y_train != -1)
X_train = X_train[labelrows]
y_train = y_train[labelrows]


def make_model(feature_size):

  tf.compat.v1.disable_eager_execution()

  keras.backend.clear_session()

  model = tf.keras.Sequential()
  model.add(layers.InputLayer(input_shape=(1,feature_size)))
  model.add(layers.Dropout(0.4))
  model.add(layers.Dense(3500, activation='relu'))
  model.add(layers.Dropout(0.4))
  model.add(layers.Dense(1, activation='sigmoid'))

  model.compile(tf.keras.optimizers.Adam(learning_rate=0.001),
            loss='binary_crossentropy',
            metrics=['accuracy',tf.keras.metrics.AUC(),tf.keras.metrics.Precision()])
```

```
  model.fit(X_train, y_train,batch_size=128,epochs=1000,validation_split=0.2)

  model_json = model.to_json()

  model.save_weights (dir+'/weights1.h5')

# Save the model architecture #
  model_json = model.to_json()
  with open(dir+"/model1.json", "w") as json_file:
      json_file.write(model_json)

  print(model.summary())

  return model

def prepare_data(X,Y,feature_size,train):
    mms = StandardScaler()
    if(train):
       for x in range(0,600000,100000):
           mms.partial_fit(X[x:x+100000])
    else:
       X_test = mms.fit_transform(X)
    X = mms.transform(X)
    X = np.reshape(X,(-1,1,feature_size))
    Y = np.reshape(Y,(-1,1,1))
    return X,Y



X1 = X_train
Y1 = y_train


feature_size = 2381

X_train,y_train = prepare_data(X1,Y1,feature_size,True)



model = make_model(feature_size)


    Train on 720 samples, validate on 180 samples
    720/720 [==============================] - 0s 477us/sample - loss: 0.3110 - accuracy: 0
    /usr/local/lib/python3.7/dist-packages/keras/engine/training.py:2470: UserWarning: `Mode
      warnings.warn('`Model.state_updates` will be removed in a future version. '
    Model: "sequential"
    _________________________________________________________________
    Layer (type)                 Output Shape              Param #
    =================================================================
```

```
dropout (Dropout)            (None, 1, 238)            0
_________________________________________________________________
dense (Dense)                (None, 1, 3500)           836500
_________________________________________________________________
dropout_1 (Dropout)          (None, 1, 3500)           0
_________________________________________________________________
dense_1 (Dense)              (None, 1, 1)              3501
=================================================================
Total params: 840,001
Trainable params: 840,001
Non-trainable params: 0
_________________________________________________________________
None
```

```
X_test,y_test = prepare_data(X_test,y_test,feature_size,False)

results =model.evaluate(X_test,y_test)
```

✓ 0s completed at 12:20 AM