```
import numpy as np
import pandas as pd
from keras.datasets import imdb
from keras.layers import LSTM, Embedding, Dense
from tensorflow.keras import Sequential
from keras.utils import pad_sequences
```

```
np.random.seed(7)
```

```
# load the data
(x_train, y_train), (x_test, y_test) = imdb.load_data(num_words=10000)
```

```
# pad input sequences

x_train = pad_sequences(x_train,maxlen = 500)
x_test = pad_sequences(x_test,maxlen = 500)
```

```

```

```
# model

model = Sequential()
model.add(Embedding(10000,32,input_length=500)) # 6000 ,32 = batch size (32 = dense vector size)
model.add(LSTM(100)) # lstm has 100 units : output is of vector size 100 for each input sequence
model.add(Dense(1,activation = 'sigmoid'))
model.compile(loss = 'binary_crossentropy',optimizer = 'adam')
print(model.summary())
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 500, 32)           320000

 lstm (LSTM)                 (None, 100)               53200

 dense (Dense)               (None, 1)                 101

=================================================================
Total params: 373,301
Trainable params: 373,301
Non-trainable params: 0
_________________________________________________________________
None
```

```
x_test.shape
```

```
(25000, 500)
```

```
y_test.shape
```

```
(25000,)
```

```
score = model.evaluate(x_test,y_test,verbose = 0)
print(score*100)
```

```
69.32507753372192
```

```

```

```

```

```

```

```

```

```

```

```

```

```
import numpy as np
import tensorflow as tf
from keras.datasets import imdb
from keras.layers import LSTM, Embedding, Dense
from tensorflow.keras import Sequential
#from keras.preprocessing.sequence import pad_sequence
from keras.utils import pad_sequences
#from keras_preprocessing.sequence import pad_sequences
# fix random seed for reproducibility
np.random.seed(7)
# load the dataset but only keep the top 6000 words
(X_train, y_train), (X_test, y_test) = imdb.load_data(num_words=6000)
# pad input sequences
X_train = pad_sequences(X_train, maxlen=500)
X_test = pad_sequences(X_test, maxlen=500)
#model
model = Sequential()
model.add(Embedding(6000, 32, input_length=500))
model.add(LSTM(100))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
print(model.summary())
# Final evaluation of the model
scores = model.evaluate(X_test, y_test, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))
```