Problem to be solved: Given a dataset containing some text related to a movie, the problem is to predict the sentiment behind the statement in the form of 0 and 1 label (0 for negative and 1 for positive)

Importing Libraries

```
!pip install tensorflow_text
```

```
Requirement already satisfied: tensorflow_text in /usr/local/lib/python3.10/dist-packages (2.14.0)
Requirement already satisfied: tensorflow-hub>=0.13.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow_text) (0.15.0)
Requirement already satisfied: tensorflow<2.15,>=2.14.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow_text) (2.14.0)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.15,>=2.14.0->tensorflow_
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.15,>=2.14.0->tensorfl
Requirement already satisfied: flatbuffers>=23.5.26 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.15,>=2.14.0->tensc
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.15,
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.15,>=2.14.0->tensor
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.15,>=2.14.0->tensorflow_te
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.15,>=2.14.0->tensorflo
Requirement already satisfied: ml-dtypes==0.2.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.15,>=2.14.0->tensorflo
Requirement already satisfied: numpy>=1.23.5 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.15,>=2.14.0->tensorflow_t
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.15,>=2.14.0->tensorfl
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.15,>=2.14.0->tensorflow_text
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3 in /usr/local/lib/py
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.15,>=2.14.0->tensorflow_text
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.15,>=2.14.0->tensorflow_te
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.15,>=2.14.0->tensorflo
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.15,>=2.14.0->t
Requirement already satisfied: wrapt<1.15,>=1.11.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.15,>=2.14.0->tensor
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.15
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.15,>=2.14.0->tensor
Requirement already satisfied: tensorboard<2.15,>=2.14 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.15,>=2.14.0->te
Requirement already satisfied: tensorflow-estimator<2.15,>=2.14.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.15,>
Requirement already satisfied: keras<2.15,>=2.14.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.15,>=2.14.0->tensor
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.10/dist-packages (from astunparse>=1.6.0->tensorflow<2.1
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.15,>=2.14->tensc
Requirement already satisfied: google-auth-oauthlib<1.1,>=0.5 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.15,>=2
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.15,>=2.14->tensorflow<
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.15,>=2.14->tensorf
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.15,>=2.14->tensorflow<
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3->tensor
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3->tensorb
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3->tensorboard<2.1
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.10/dist-packages (from google-auth-oauthlib<1.1,>=
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensor
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorboard<2.15,>
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorboard<
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorboard<
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.10/dist-packages (from werkzeug>=1.0.1->tensorboard<2.15,
Requirement already satisfied: pyasn1<0.6.0,>=0.4.6 in /usr/local/lib/python3.10/dist-packages (from pyasn1-modules>=0.2.1->google-a
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.10/dist-packages (from requests-oauthlib>=0.7.0->google-aut
```

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

Reading Dataset

```
df = pd.read_csv('/content/drive/MyDrive/Train.csv')
```

```
df.head()
```

	text	label
0	I grew up (b. 1965) watching and loving the Th...	0
1	When I put this movie in my DVD player, and sa...	0
2	Why do people who do not know what a particula...	0

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40000 entries, 0 to 39999
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   text    40000 non-null  object
 1   label   40000 non-null  int64 
dtypes: int64(1), object(1)
memory usage: 625.1+ KB
```

Checking for NAN values

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

```
text     0
label    0
dtype: int64
```

Checking the count of lables

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

```
0    20019
1    19981
Name: label, dtype: int64
```

```
df = df.sample(10000)
```

To reduce the training time, I'm taking 10k samples

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

```
1    5042
0    4958
Name: label, dtype: int64
```

Importing Tensorflow libraries

```
import tensorflow_hub as hub
import tensorflow_text as text
```

```
df.head()
```

	text	label
32823	The central theme in this movie seems to be co...	0
16298	An excellent example of "cowboy noir", as it's...	1
28505	The ending made my heart jump up into my throa...	0
6689	Only the chosen ones will appreciate the quali...	1
26893	This is a really funny film, especially the se...	1

Separatind target feature and other features

```
x = df.drop('label',axis=1)
y = df['label']
```

Splitting dataset into training and testing

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y, random_state=42)
```

```
x_train.shape
```

```
(7500, 1)
```

```
x_train.head()
```

	text
31627	Ossessione is in very bad state but is now und...
31534	Martin Weisz, who directed the solid "Rohtenbu...
26717	... but had to see just how bad it could get. ...
1458	The Late Shift is a great book, I read the boo...
30084	First of all I've got to give it to the people...

You can learn more about the bert architecture and working here

Loading Model and Encoder

```
bert_preprocess = hub.KerasLayer('https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3')
bert_encoder = hub.KerasLayer("https://tfhub.dev/tensorflow/bert_en_uncased_L-12_H-768_A-12/4")
```

```
from keras.src.layers.attention.multi_head_attention import regularization
```

Bulding Model

```
keras.utils.set_random_seed(42)

text_input = tf.keras.layers.Input(shape=(),  name='text',dtype=tf.string)

preprocessed_text = bert_preprocess(text_input)

output = bert_encoder(preprocessed_text)

l = tf.keras.layers.Dropout(.1, name='dropout1')(output['pooled_output'])
l = tf.keras.layers.Dense(1, activation='sigmoid',name='output')(l)
```

```
model = tf.keras.Model(inputs=[text_input], outputs=[l])
```

```
print(model.summary())
```

```
Model: "model_1"
__________________________________________________________________________________________________
 Layer (type)                Output Shape                 Param #   Connected to
==================================================================================================
 text (InputLayer)           [(None,)]                    0         []

 keras_layer_2 (KerasLayer)  {'input_word_ids': (None,    0         ['text[0][0]']
                             128),
                              'input_type_ids': (None,
                             128),
                              'input_mask': (None, 128)
                             }

 keras_layer_3 (KerasLayer)  {'encoder_outputs': [(None   1094822   ['keras_layer_2[0][0]',
                             , 128, 768),                 41          'keras_layer_2[0][1]',
                              (None, 128, 768),                       'keras_layer_2[0][2]']
                              (None, 128, 768),
                              (None, 128, 768),
                              (None, 128, 768),
                              (None, 128, 768),
                              (None, 128, 768),
                              (None, 128, 768),
                              (None, 128, 768),
```

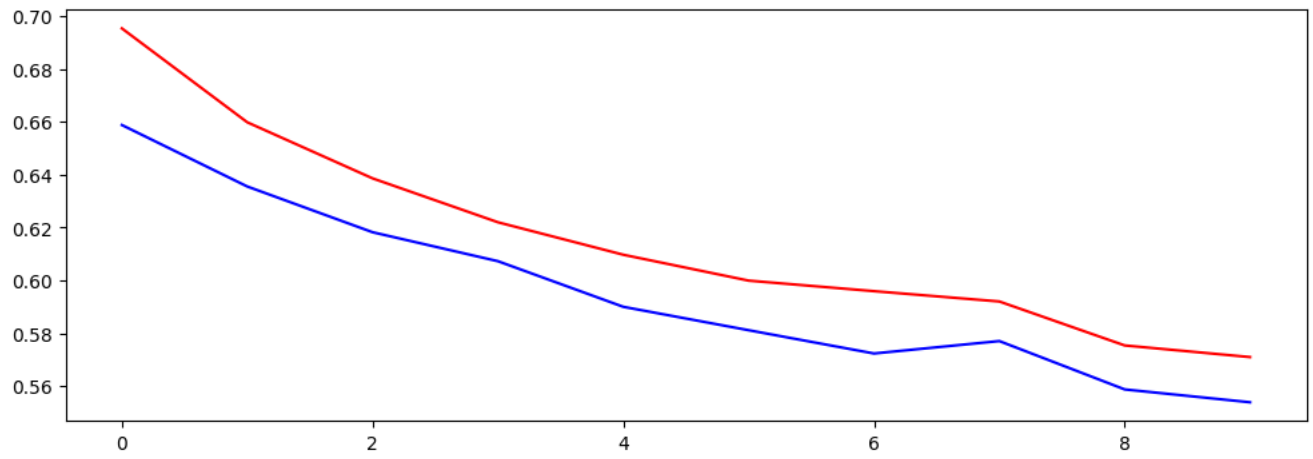
```
                                (None, 128, 768),
                                (None, 128, 768),
                                (None, 128, 768)],
                                'default': (None, 768),
                                'pooled_output': (None, 7
                               68),
                                'sequence_output': (None,
                                128, 768)}

 dropout1 (Dropout)          (None, 768)                 0         ['keras_layer_3[0][13]']

 output (Dense)              (None, 1)                   769       ['dropout1[0][0]']

==================================================================================================
Total params: 109483010 (417.64 MB)
Trainable params: 769 (3.00 KB)
Non-trainable params: 109482241 (417.64 MB)
__________________________________________________________________________________________________
None
```

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

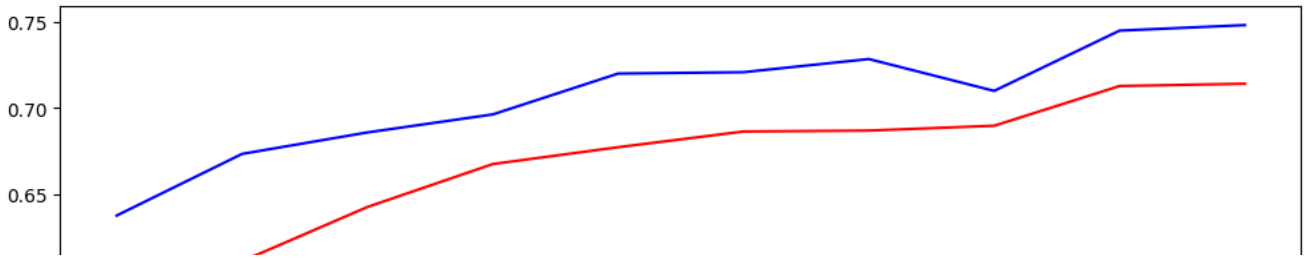
```
fit = model.fit(x_train, y_train, epochs=10, validation_data=(x_test, y_test),batch_size=128)
```

```
Epoch 1/10
59/59 [==============================] - 156s 2s/step - loss: 0.6953 - accuracy: 0.5319 - val_loss: 0.6588 - val_accuracy: 0.6380
Epoch 2/10
59/59 [==============================] - 145s 2s/step - loss: 0.6598 - accuracy: 0.6116 - val_loss: 0.6355 - val_accuracy: 0.6736
Epoch 3/10
59/59 [==============================] - 145s 2s/step - loss: 0.6386 - accuracy: 0.6429 - val_loss: 0.6183 - val_accuracy: 0.6860
Epoch 4/10
59/59 [==============================] - 145s 2s/step - loss: 0.6220 - accuracy: 0.6677 - val_loss: 0.6073 - val_accuracy: 0.6964
Epoch 5/10
59/59 [==============================] - 145s 2s/step - loss: 0.6097 - accuracy: 0.6775 - val_loss: 0.5901 - val_accuracy: 0.7200
Epoch 6/10
59/59 [==============================] - 144s 2s/step - loss: 0.6000 - accuracy: 0.6865 - val_loss: 0.5812 - val_accuracy: 0.7208
Epoch 7/10
59/59 [==============================] - 145s 2s/step - loss: 0.5960 - accuracy: 0.6871 - val_loss: 0.5724 - val_accuracy: 0.7284
Epoch 8/10
59/59 [==============================] - 145s 2s/step - loss: 0.5921 - accuracy: 0.6899 - val_loss: 0.5771 - val_accuracy: 0.7100
Epoch 9/10
59/59 [==============================] - 145s 2s/step - loss: 0.5755 - accuracy: 0.7128 - val_loss: 0.5588 - val_accuracy: 0.7448
Epoch 10/10
59/59 [==============================] - 145s 2s/step - loss: 0.5711 - accuracy: 0.7141 - val_loss: 0.5540 - val_accuracy: 0.7480
```

```
plt.figure(figsize=(12,4))
plt.plot(fit.history['loss'], label='training loss', color='r')
plt.plot(fit.history['val_loss'],label='validation loss',color='b')
plt.show()
```




```
plt.figure(figsize=(12,4))
plt.plot(fit.history['accuracy'], label='training accuracy', color='r')
plt.plot(fit.history['val_accuracy'],label='validation accuracy',color='b')
plt.show()
```



Making Predictions

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

```
79/79 [==============================] - 28s 352ms/step
```

```
pred[:10]
```

```
array([[0.5883882 ],
       [0.26047358],
       [0.38230047],
       [0.51511014],
       [0.30832118],
       [0.2700232 ],
       [0.1878159 ],
       [0.1257992 ],
       [0.4433875 ],
       [0.7639161 ]], dtype=float32)
```

```
# Converting probabilities into 1 and 0
pred = [1 if x>=.5 else 0 for x in pred]
```

Checking performance of our model

```
from sklearn.metrics import classification_report
```

```
print(classification_report(y_test, pred))
```

```
                  precision    recall  f1-score   support

               0       0.73      0.76      0.75      1222
               1       0.76      0.73      0.75      1278

        accuracy                           0.75      2500
       macro avg       0.75      0.75      0.75      2500
    weighted avg       0.75      0.75      0.75      2500
```

Overall model is giving a good performance.