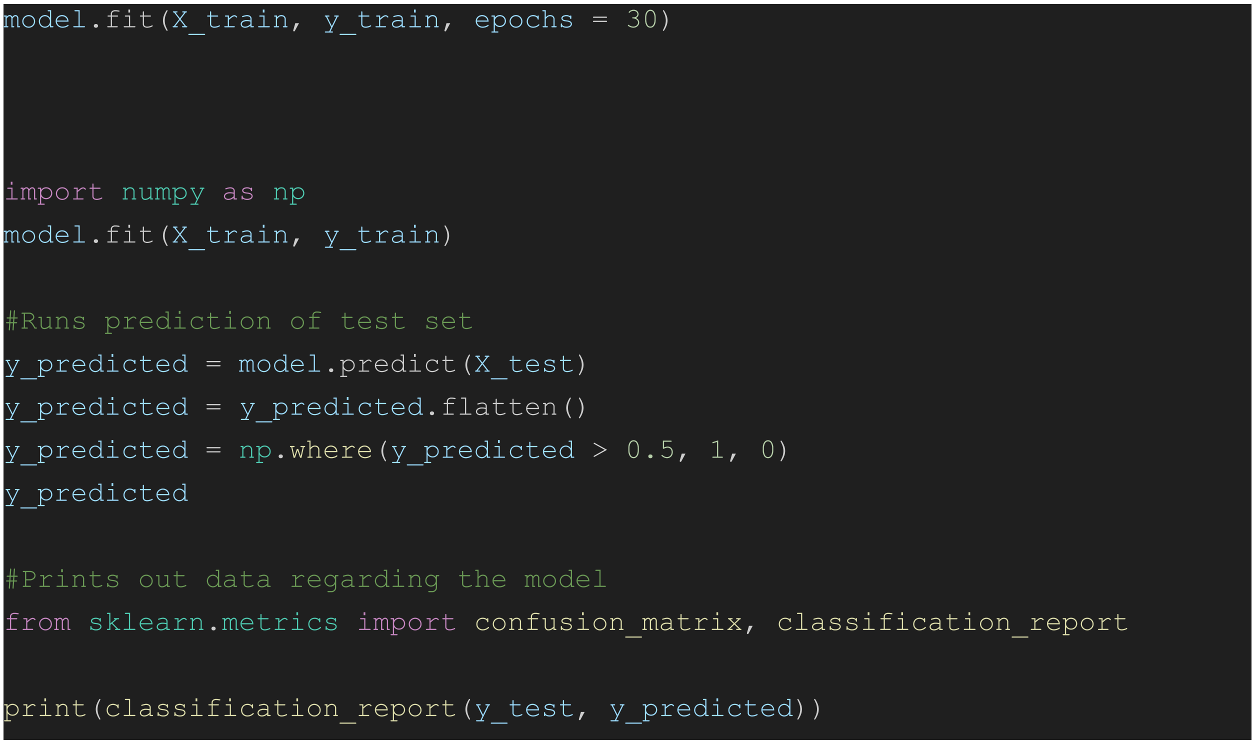
```
#This model was adjusted from a model created by codebasics. Neural
network layers and K Fold Validation in addition to alterations to the
BERT layer were done by Gustavo Rodriguez with the help of Joseph Yu.

import tensorflow as tf
import tensorflow_hub as hub
import tensorflow_text as text
import pandas as pd
df = pd.read_csv("/mnt/c/Users/ger81/Desktop/DEMENTIA/RedoDatas3.csv")
df_positive = df[df['Diagnosis']==0]
df_negative = df[df['Diagnosis']==1]
df_balanced = pd.concat([df_positive, df_negative])
from sklearn.model_selection import train_test_split
#Train Test Split
X_train, X_test, y_train, y_test =
train_test_split(df_balanced['Text'],df_balanced['Diagnosis'],
stratify=df_balanced['Diagnosis'], train_size = .9, test_size = .1,
random_state=50 )

#K Fold Validation - Was used to carry out T-Tests on the model and get
reliable accuracy numbers.

# box1 = df[0:14]
# box2 = df[14:28]
# box3= df[28:42]
# box4 = df[42:56]
# box5 = df[56:70]
# box6 = df[70:84]
# box7 = df[84:98]
# box8 = df[98:112]
# box9 = df[112:126]
# box10 = df[126:140]
# X_train = pd.concat([box2['Text'], box3['Text'], box4['Text'],
box5['Text'], box6['Text'], box7['Text'], box8['Text'], box9['Text'],
box10['Text']])
# X_test = pd.concat ([box1['Text']])
# y_train = pd.concat([box2['Diagnosis'], box3['Diagnosis'],
box4['Diagnosis'], box5['Diagnosis'], box6['Diagnosis'],
box7['Diagnosis'], box8['Diagnosis'], box9['Diagnosis'],
box10['Diagnosis']])
```

```
# y_test = pd.concat([box1['Diagnosis']])
# print(box1)

#Loads BERT
bert_preprocess = hub.KerasLayer("https://kaggle.com/models/tensorflow/bert/frameworks/TensorFlow2/variations/en-uncased-preprocess/versions/3")
bert_encoder = hub.KerasLayer("https://www.kaggle.com/models/tensorflow/bert/frameworks/TensorFlow2/variations/bert-en-uncased-l-8-h-256-a-4/versions/2")

# Bert layers - Converts text sequences into array
text_input = tf.keras.layers.Input(shape=(), dtype=tf.string, name='text')
preprocessed_text = bert_preprocess(text_input)
outputs = bert_encoder(preprocessed_text)
outputs = outputs['pooled_output']
# Neural network layers - Binary Classifier
l = tf.keras.layers.Dropout(0.1, name="dropout")(outputs)
l = tf.keras.layers.Dense(256, activation='relu', name="den1")(l)
l = tf.keras.layers.Dense(128, activation='relu', name="den2")(l)
l = tf.keras.layers.Dense(64, activation='relu', name="den3")(l)
l = tf.keras.layers.Dense(16, activation='relu', name="den4")(l)
l = tf.keras.layers.Dense(1, activation='sigmoid', name="MAMS")(l)

# Use inputs and outputs to construct a final model
model = tf.keras.Model(inputs=[text_input], outputs = [l])

model.summary()

METRICS = [
      tf.keras.metrics.BinaryAccuracy(name='accuracy'),
      tf.keras.metrics.Precision(name='precision'),
      tf.keras.metrics.Recall(name='recall')
]

model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=METRICS)

#Trains the model
```



```
model.fit(X_train, y_train, epochs = 30)



import numpy as np
model.fit(X_train, y_train)

#Runs prediction of test set
y_predicted = model.predict(X_test)
y_predicted = y_predicted.flatten()
y_predicted = np.where(y_predicted > 0.5, 1, 0)
y_predicted

#Prints out data regarding the model
from sklearn.metrics import confusion_matrix, classification_report

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