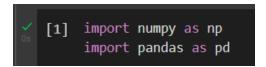
Assignment 4

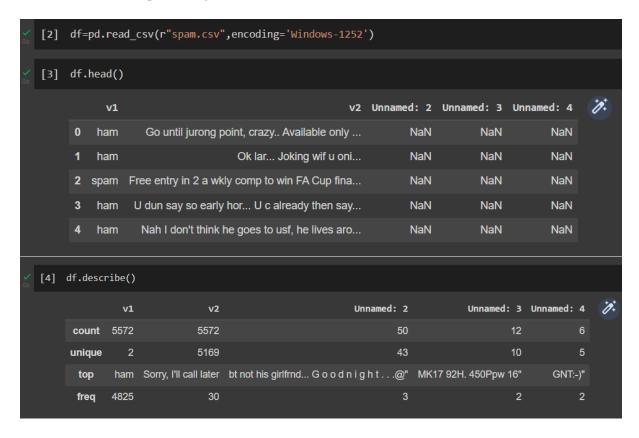
SMS SPAM Classification

Assignment Date	28 October 2022
Student Name	Santhosh M
Student Roll Number	195002309
Maximum Marks	2 Marks

1. Import Required Libraries



2. Read Dataset Do Pre-processing



3. Create Model

```
[5] import re import nltk from nltk.corpus import stopwords from nltk.stem.porter import PorterStemmer

[6] ps=PorterStemmer()

[7] import nltk nltk.download('stopwords')

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.

True

[8] data=[]

for i in range(0,5572):

    message=df["v2"][i]
    message=message.lower()
    message=message.split()
    message=nessage.split()
    message=[ps.stem(word) for word in message if not word in set(stopwords.words("english"))]
    message=' '.join(message)

data.append(message)
```

```
'also sorta blown coupl time recent id rather text blue look weed',
'sent score sopha secondari applic school think think appli research cost also contact joke ogunrind school one less expens one',
'cant wait see photo use',
'ur cash balanc current pound maxim ur cash send go p msg cc po box tcr w',
'hey book bh sat alreadi lesson go ah keep sat night free need meet confirm lodg',
'chk ur belovd ms dict',
'time want come',
'awesom lemm know whenev around',
'shb b ok lor thanx',
'beauti truth graviti read care heart feel light someon feel heavi someon leav good night',
'also rememb get dobbi bowl car',
'filthi stori girl wait',
'sorri c ur msg yar lor poor thing one night tmr u brand new room sleep',
'love decis feel could decid love life would much simpler less magic',
'welp appar retir',
'sort code acc bank natwest repli confirm sent right person',
'',
'' ur sure u take sick time',
'urgent tri contact u today draw show prize guarante call land line claim valid hr',
'watch cartoon listen music amp eve go templ amp church u',
'yo chad gymnast class wanna take site say christian class full',
'much buzi',
'better still catch let ask sell lt gt',
'sure night menu know noon menu',
'u want come back beauti necklac token heart that give wife like see one give dont call wait till come',
'will go aptitud class',
'wont b tri sort hous ok',
'yar lor wan go c hors race today mah eat earlier lor ate chicken rice u',
'haha awesom omme back',
'yar lor wan go c hors race today mah eat earlier lor ate chicken rice u',
'haha awesom omme back',
'yar lor wan go c hors race today mah eat earlier lor ate chicken rice u',
'haha awesom omme back',
'yar lor wan go c hors race today mah eat earlier lor ate chicken rice u',
'haha ewesom omme back',
'yar lor wan go c hors race today mah eat earlier lor ate chicken rice u',
'haha ewesom omme back',
'yar lor wan go c hors race today mah eat earlier lor ate chicken rice u',
'haha ewesom omme back'.
```

```
[10] from sklearn.feature extraction.text import CountVectorizer
[11] cv=CountVectorizer(max_features=7000)
      x=cv.fit_transform(data).toarray()
     (5572, 6221)
[12] df["v1"].loc[df["v1"]=="spam"]=0.0
     df["v1"].loc[df["v1"]=="ham"]=1.0
df["v1"]
             1.0
             0.0
              1.0
     4
              1.0
             0.0
     5567
     5568
              1.0
              1.0
     5570
     Name: v1, Length: 5572, dtype: object
[13] y=df.iloc[:,0:1].values
      y=np.asarray(y).astype("float64")
```

4. Add Layers

```
[16] from sklearn.model_selection import train_test_split
    x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)

[17] from tensorflow.keras.models import Sequential
    from tensorflow.keras.layers import Dense
    model=Sequential()

[18] #input layer
    model.add(Dense(units=5572,activation='relu',kernel_initializer='random_uniform'))
    #hidden layer
    model.add(Dense(units=6000,activation='relu',kernel_initializer='random_uniform'))
    model.add(Dense(units=6000,activation='relu',kernel_initializer='random_uniform'))
    model.add(Dense(units=6000,activation='relu',kernel_initializer='random_uniform'))
    model.add(Dense(units=6000,activation='relu',kernel_initializer='random_uniform'))
    #output layer
    model.add(Dense(units=1,activation='sigmoid',kernel_initializer='random_uniform'))
```

5. Compile the model

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

6. Fit the Model

```
[20] tr=model.fit(x_train,y_train,epochs=10,batch_size=32)
    Epoch 1/10
                          140/140 [==
    Epoch 2/10
                              ========] - 251s 2s/step - loss: 0.0357 - accuracy: 0.9946
    140/140 [==
    Epoch 3/10
    140/140 [==
                                    ====] - 261s 2s/step - loss: 0.0187 - accuracy: 0.9971
    Epoch 4/10
    140/140 [==
                             ========] - 247s 2s/step - loss: 0.0055 - accuracy: 0.9989
    Epoch 5/10
    140/140 [==
                                      ==] - 248s 2s/step - loss: 5.3765e-06 - accuracy: 1.0000
    Epoch 6/10
    140/140 [==
                                   =====] - 244s 2s/step - loss: 3.4380 - accuracy: 0.9569
    Epoch 7/10
                                  ======] - 241s 2s/step - loss: 1.2959 - accuracy: 0.9693
    140/140 [==
    Epoch 8/10
    140/140 [==
                             =======] - 237s 2s/step - loss: 6.5073 - accuracy: 0.9868
    Epoch 9/10
    140/140 [==
                               =======] - 236s 2s/step - loss: 0.0281 - accuracy: 0.9971
    Epoch 10/10
                            =======] - 240s 2s/step - loss: 1.9029 - accuracy: 0.9834
    140/140 [=====
```

7. Save the Model

```
[21] # Save The Model
     model.save("sms.h5")
[22] # Test The Model
     ypred=model.predict(x test)
     ypred
     35/35 [============ ] - 14s 383ms/step
     array([[1.
            [0.99999934],
            [1.
                       ],
            . . . ,
            [1.
                       ],
            [1.
                       ]], dtype=float32)
            [1.
```

8. Test the Model