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

```
[1]  import numpy as np
     import pandas as pd
```

2. Read Dataset Do Pre-processing

```
[2]  df=pd.read_csv(r"spam.csv",encoding='Windows-1252')
```

```
[3]  df.head()
```

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN

```
[4]  df.describe()
```

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
count	5572	5572	50	12	6
unique	2	5169	43	10	5
top	ham	Sorry, I'll call later	bt not his girlfrnd... G o o d n i g h t . . .@"	MK17 92H. 450Ppw 16"	GNT:-)"
freq	4825	30	3	2	2

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=re.sub('[^a-z]',' ',message)
         message=message.split()
         message=[ps.stem(word) for word in message if not word in set(stopwords.words("english"))]
         message=' '.join(message)
         data.append(message)
```

```
[9]  data
```

```
'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 kb 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',
'',
'u 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 omw back',
'yup thk e shop close lor',
'account number',
```

```
[10] from sklearn.feature_extraction.text import CountVectorizer
```

```
[11] cv=CountVectorizer(max_features=7000)
     x=cv.fit_transform(data).toarray()
     x.shape
```

```
(5572, 6221)
```

```
[12] df["v1"].loc[df["v1"]=="spam"]=0.0
     df["v1"].loc[df["v1"]=="ham"]=1.0
     df["v1"]
```

```
0       1.0
1       1.0
2       0.0
3       1.0
4       1.0
       ...
5567    0.0
5568    1.0
5569    1.0
5570    1.0
5571    1.0
Name: v1, Length: 5572, dtype: object
```

```
[13] y=df.iloc[:,0:1].values
     y=np.asarray(y).astype("float64")
```

```
[14] y
```

```
array([[1.],
       [1.],
       [0.],
       ...,
       [1.],
       [1.],
       [1.]])
```

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'))
     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 [==============================] - 258s 2s/step - loss: 1.8936 - accuracy: 0.9457
Epoch 2/10
140/140 [==============================] - 251s 2s/step - loss: 0.0357 - accuracy: 0.9946
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
140/140 [==============================] - 241s 2s/step - loss: 1.2959 - accuracy: 0.9693
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
140/140 [==============================] - 240s 2s/step - loss: 1.9029 - accuracy: 0.9834
```

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.        ],
            [1.        ]], dtype=float32)
```

8. Test the Model

```
[23] y_test
```

```
array([[1.],
       [1.],
       [1.],
       ...,
       [1.],
       [1.],
       [1.]])
```

```
[24] text=model.predict(cv.transform(["Wishing you and your family a very happy and Prosperous Diwali!"]))
     text>0.5
```

```
1/1 [==============================] - 1s 595ms/step
array([[ True]])
```

```
class_name=["ham","spam"]
pred_id=text.argmax(axis=1)[0]
pred_id
print(str(class_name[pred_id]))
```

```
ham
```