Assignment - 4

SMS SPAM Classification

Assignment Date	29 October 2022
Student Name	Mr. Dhinagaran . S
Student Roll Number	513419106008
Maximum Marks	2 Marks

SPAM Classifier

Importing required libraries

In [120]:

```
import pandas as pd
import numpy as np
import nltk
import re

nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import CountVectorizer
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Reading Dataset

In [121]:

```
df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/ibm/assignment_4/spam.csv', enco
ding='ISO-8859-1')
df.shape
```

Out[121]:

```
(5572, 5)
```

Analysing Dataset

In [122]:

```
df
```

Out[122]:

	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
...	...	...	...	...	...

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
5567	spam	This is the 2nd time we have tried 2 contact u...	NaN	NaN	NaN
5568	ham	Will Ì_ b going to esplanade fr home?	NaN	NaN	NaN
5569	ham	Pity, * was in mood for that. So...any other s...	NaN	NaN	NaN
5570	ham	The guy did some bitching but I acted like i'd...	NaN	NaN	NaN
5571	ham	Rofl. Its true to its name	NaN	NaN	NaN

5572 rows × 5 columns

In [123]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   v1          5572 non-null   object
 1   v2          5572 non-null   object
 2   Unnamed: 2  50 non-null     object
 3   Unnamed: 3  12 non-null     object
 4   Unnamed: 4  6 non-null      object
dtypes: object(5)
memory usage: 217.8+ KB
```

In [124]:

```
df.describe()
```

Out[124]:

	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

In [125]:

```
print(f'Checking is there any columns having null values \n{df.isnull().any()}\n')
print(f'Checking is there any columns having only null values \n{df.isnull().all()}\n')
print(f'Checking total number of null values in all columms \n{df.isnull().sum()}\n')
print(df.shape)
```

```
Checking is there any columns having null values 
v1            False
v2            False
Unnamed: 2     True
Unnamed: 3     True
Unnamed: 4     True
dtype: bool

Checking is there any columns having only null values 
v1            False
v2            False
Unnamed: 2    False
Unnamed: 3    False
Unnamed: 4    False
dtype: bool

Checking total number of null values in all columms 
v1               0
v2               0
Unnamed: 2    5522
Unnamed: 3    5560
Unnamed: 4    5566
```

```
Unnamed: 4    5566
dtype: int64

(5572, 5)
```

Pre-Processing Data to create model

In [126]:

```
# Taking a copy of dataset

df1 = df.copy()
```

In [127]:

```
# Removing those columns having very less data

df1 = df1.iloc[:,0:2]
df1.shape
```

Out[127]:

```
(5572, 2)
```

In [128]:

```
# Checking for null values

df1.isnull().sum()
```

Out[128]:

```
v1    0
v2    0
dtype: int64
```

In [129]:

```
# Seperating Independent and Dependent Columns

train_set_x = df1.iloc[:,1:2]
train_set_y = df1.iloc[:,0:1]
print(train_set_x)
print(train_set_y)
```

```
                                                     v2
0     Go until jurong point, crazy.. Available only ...
1                         Ok lar... Joking wif u oni...
2     Free entry in 2 a wkly comp to win FA Cup fina...
3     U dun say so early hor... U c already then say...
4     Nah I don't think he goes to usf, he lives aro...
...                                                 ...
5567  This is the 2nd time we have tried 2 contact u...
5568               Will Ì_ b going to esplanade fr home?
5569  Pity, * was in mood for that. So...any other s...
5570  The guy did some bitching but I acted like i'd...
5571                         Rofl. Its true to its name

[5572 rows x 1 columns]
        v1
0      ham
1      ham
2     spam
3      ham
4      ham
...    ...
5567  spam
5568   ham
5569   ham
5570   ham
5571   ham
```

```
[5572 rows x 1 columns]
```

Creating an Object for doing Pre-Processing

In [130]:

```
class SMSProcessor():

  def __init__(self,x,y):
    try:
      if len(x) == len(y):
        self.x = x
        self.y = y
        self.data = []
        self.ps = PorterStemmer()
        self.cv = CountVectorizer()
        self.re = re
        self.limit = self.x.shape[0]
    except:
      raise 'The given independent column - x  and dependent column - y   sizes are not
matching'


  def sentence_process(self,string):
    v2 = str(string)
    v2 = self.re.sub('[^a-zA-Z]',' ',v2)
    v2 = v2.lower()
    v2 = v2.split()
    v2 = [self.ps.stem(word) for word in v2 if word not in set(stopwords.words('english'
))]
    v2 = ' '.join(v2)
    return v2

  def sentence_updater(self):
    for i in range(0,self.limit):
      data = self.sentence_process(self.x.values[i])
      self.data.append(data)

  def train_process(self):
    self.x = self.cv.fit_transform(self.data).toarray()
    self.y = pd.get_dummies(self.y).drop('v1_spam', axis=1)

  def x_y_formater(self):
    self.sentence_updater()
    self.train_process()
    return self.x, self.y

  def test_process(self,string):
    string = self.sentence_process(string)
    string = self.cv.transform([string]).toarray()
    return string
```

Preprocessing Dataset

In [131]:

```
processor = SMSProcessor(train_set_x, train_set_y)

x_train,y_train = processor.x_y_formater()
print(x_train)
print(y_train)
```

```
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
      v1_ham
0          1
1          1
2          0
3          1
4          1
...      ...
5567       0
5568       1
5569       1
5570       1
5571       1

[5572 rows x 1 columns]
```

Model training

Importing required libraries for model training

In [132]:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

Creating Model Skeleton

In [133]:

```
model = Sequential()
model.add(Dense(1000, activation='relu'))
model.add(Dense(1500, activation='relu'))
model.add(Dense(3000, activation='relu'))
model.add(Dense(5000, activation='relu'))
model.add(Dense(500, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
```

Compiling Model to train

In [134]:

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

Training Model

In [135]:

```
model.fit(x_train,y_train,epochs=15)
```

```
Epoch 1/15
175/175 [==============================] - 2s 10ms/step - loss: 0.1350 - accuracy: 0.9675
Epoch 2/15
175/175 [==============================] - 2s 9ms/step - loss: 0.0126 - accuracy: 0.9964
Epoch 3/15
175/175 [==============================] - 2s 9ms/step - loss: 0.0017 - accuracy: 0.9995
Epoch 4/15
175/175 [==============================] - 2s 9ms/step - loss: 0.0030 - accuracy: 0.9993
Epoch 5/15
175/175 [==============================] - 2s 9ms/step - loss: 0.0020 - accuracy: 0.9991
Epoch 6/15
175/175 [==============================] - 2s 9ms/step - loss: 0.0035 - accuracy: 0.9996
Epoch 7/15
175/175 [==============================] - 2s 9ms/step - loss: 0.0061 - accuracy: 0.9978
Epoch 8/15
175/175 [==============================] - 2s 9ms/step - loss: 3.1365e-04 - accuracy: 0.9
```

```
998
Epoch 9/15
175/175 [==============================] - 2s 10ms/step - loss: 2.8419e-06 - accuracy: 1.
0000
Epoch 10/15
175/175 [==============================] - 2s 9ms/step - loss: 1.4639e-07 - accuracy: 1.0
000
Epoch 11/15
175/175 [==============================] - 2s 9ms/step - loss: 1.2206e-07 - accuracy: 1.0
000
Epoch 12/15
175/175 [==============================] - 2s 9ms/step - loss: 1.0303e-07 - accuracy: 1.0
000
Epoch 13/15
175/175 [==============================] - 2s 9ms/step - loss: 8.6978e-08 - accuracy: 1.0
000
Epoch 14/15
175/175 [==============================] - 2s 9ms/step - loss: 7.5572e-08 - accuracy: 1.0
000
Epoch 15/15
175/175 [==============================] - 2s 9ms/step - loss: 6.3404e-08 - accuracy: 1.0
000
```

Out[135]:

```
<keras.callbacks.History at 0x7f914a0e6c10>
```

Saving Model

In [136]:

```
model.save('sms.h5')
```

Testing Model

In [137]:

```
sample_input = input('Enter the sms here : \n')
sms = processor.test_process(sample_input)
pred = model.predict(sms)
print(f'\n\nThe prodicted binary output is : {pred[0][0]}')
print(f"The SMS is {'HAM' if pred>0.5 else 'SPAM'}")
```

```
Enter the sms here :
Will Ì_ b going to esplanade fr home?
1/1 [==============================] - 0s 64ms/step


The prodicted binary output is : 1.0
The SMS is HAM
```