ASSIGNMENT 4 - SPAM CLASSIFICATION

Team Lead - Chadalavada Gautham (Roll No:310619104018)

IMPORT LIBRARIES

```
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
```

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

LOAD DATASET

```
a = pd.read_csv('/content/spam.csv',encoding='ISO-8859-1')
a.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

```
a=a[['v1','v2']]
a.head()
```

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

```
a.shape
```

```
(5572, 2)
```

Text processing (NLP)

```
ps=PorterStemmer()
message=[]
for i in range(0,5572):
  msg=a['v2'][i]
  msg=re.sub('[^a-zA-Z]',' ',msg)
  msg=msg.lower()
  msg=msg.split(' ')
  msg = [ps.stem(word) for word in msg if word not in set(stopwords.words('english'))]
  msg=' '.join(msg)
  message.append(msg)
```

```
message[:6]
```

```
['go jurong point  crazi   avail bugi n great world la e buffet    cine got amor wat
',
 'ok lar    joke wif u oni   ',
 'free entri   wkli comp win fa cup final tkt   st may       text fa       receiv entri
question std txt rate c appli              ',
 'u dun say earli hor    u c alreadi say   ',
 'nah think goe usf  live around though',
 'freemsg hey darl   week word back  like fun still  tb ok  xxx std chg send
rcv']
```

```
from sklearn.feature_extraction.text import CountVectorizer
```

```
cv = CountVectorizer()
x = cv.fit_transform(message).toarray()
x
```

```
array([[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]])
```

```
#LABEL ENCODING

from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()


a['v1']=le.fit_transform(a['v1'])
y = a['v1'].values
y
```

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

MODEL BUILDIND

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

model = Sequential()
model.add(Dense(1550,activation='relu'))
model.add(Dense(3000,activation='relu'))
model.add(Dense(1,activation='sigmoid'))

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

```
model.fit(x,y,epochs=10)
```

```
     Epoch 1/10
     175/175 [==============================] - 19s 105ms/step - loss: 0.1128 - accuracy: 0.9
     Epoch 2/10
     175/175 [==============================] - 18s 105ms/step - loss: 0.0131 - accuracy: 0.9
     Epoch 3/10
     175/175 [==============================] - 18s 105ms/step - loss: 0.0013 - accuracy: 0.9
     Epoch 4/10
     175/175 [==============================] - 18s 104ms/step - loss: 1.9955e-04 - accuracy
     Epoch 5/10
     175/175 [==============================] - 18s 104ms/step - loss: 8.9791e-05 - accuracy
     Epoch 6/10
     175/175 [==============================] - 18s 105ms/step - loss: 5.2074e-05 - accuracy
     Epoch 7/10
     175/175 [==============================] - 18s 105ms/step - loss: 3.3522e-05 - accuracy
     Epoch 8/10
     175/175 [==============================] - 18s 105ms/step - loss: 2.3012e-05 - accuracy
     Epoch 9/10
```

```
175/175 [==============================] - 18s 105ms/step - loss: 1.6572e-05 - accuracy
Epoch 10/10
175/175 [==============================] - 18s 105ms/step - loss: 1.2497e-05 - accuracy
<keras.callbacks.History at 0x7f71945e6090>
```

SAVE THE MODEL

```
model.save('spam-NLP.h5')
```

TEST THE MODEL

```
msg='FREE MESSAGE Activate your 500 FREE Text Messages by replying to this message with the w
print('THE ORIGINAL MESSAGE IS:  ',msg)
msg=re.sub('[^a-zA-Z]',' ',msg)
msg=msg.lower()
msg=msg.split(' ')
msg = [ps.stem(word) for word in msg if word not in set(stopwords.words('english'))]
msg=' '.join(msg)
print('THE STEMMED MESSAGE IS:  ',msg)

predict = model.predict(cv.transform([msg]))
if predict > 0.5:
  pred='SPAM'
else: pred='NOT SPAM'
print('THE MESSAGE IS PREDICTED AS:  ',pred)
```

```
THE ORIGINAL MESSAGE IS:   FREE MESSAGE Activate your 500 FREE Text Messages by replying
THE STEMMED MESSAGE IS:   free messag activ     free text messag repli messag word free
1/1 [==============================] - 0s 158ms/step
THE MESSAGE IS PREDICTED AS:   SPAM
```

```
msg='Wishing you and your family Merry \X\" mas and HAPPY NEW Year in advance.."'
print('THE ORIGINAL MESSAGE IS:  ',msg)
msg=re.sub('[^a-zA-Z]',' ',msg)
msg=msg.lower()
msg=msg.split(' ')
msg = [ps.stem(word) for word in msg if word not in set(stopwords.words('english'))]
msg=' '.join(msg)
print('THE STEMMED MESSAGE IS:  ',msg)

predict = model.predict(cv.transform([msg]))
if predict > 0.5:
  pred='spam'
```

```
else: pred='NOT SPAM'
print('THE MESSAGE IS PREDICTED AS:  ',pred)
```

```
THE ORIGINAL MESSAGE IS:   Wishing you and your family Merry \X" mas and HAPPY NEW Year
THE STEMMED MESSAGE IS:   wish famili merri  x  ma happi new year advanc
1/1 [==============================] - 0s 9ms/step
THE MESSAGE IS PREDICTED AS:   NOT SPAM
```