IMPORT LIBRARIES

In [97]:

```
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]   Package stopwords is already up-to-date!
```

LOAD DATASET

In [98]:

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

Out[98]:

	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

In [99]:

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

Out[99]:

	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...

In [100]:

```
a.shape
```

Out[100]:

```
(5572, 2)
```

Text processing (NLP)

In [101]:

```
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]
```

Out[101]:

```
['go jurong point  crazi   avail bugi n great world la e buffet    cine got a
mor wat   ',
 'ok lar    joke wif u oni   ',
 'free entri   wkli comp win fa cup final tkt   st may        text fa        re
ceiv 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']
```

In [102]:

```
from sklearn.feature_extraction.text
import CountVectorizer

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

Out[102]:

```
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]])
```

In [103]:

```
#LABEL ENCODING
 from sklearn.preprocessing import
LabelEncoder le = LabelEncoder()

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

Out[103]:

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

MODEL BUILDIND

In [104]:

```
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 [==============================] - 20s 110ms/step - loss: 0.1053 - ac
curacy: 0.9684
Epoch 2/10
175/175 [==============================] - 19s 109ms/step - loss: 0.0079 - ac
curacy: 0.9978
Epoch 3/10
175/175 [==============================] - 19s 109ms/step - loss: 0.0019 - ac
curacy: 0.9993
Epoch 4/10
175/175 [==============================] - 19s 109ms/step - loss: 2.5750e-04
- accuracy: 1.0000
Epoch 5/10
175/175 [==============================] - 20s 112ms/step - loss: 8.1691e-05
- accuracy: 1.0000
Epoch 6/10
175/175 [==============================] - 20s 115ms/step - loss: 4.8099e-05
- accuracy: 1.0000
```

```
Epoch 7/10
175/175 [==============================] - 19s 109ms/step - loss: 3.0980e-05
- accuracy: 1.0000
Epoch 8/10
175/175 [==============================] - 19s 108ms/step - loss: 2.1764e-05
- accuracy: 1.0000
Epoch 9/10
175/175 [==============================] - 19s 107ms/step - loss: 1.5995e-05
- accuracy: 1.0000
Epoch 10/10
175/175 [==============================] - 19s 108ms/step - loss: 1.2088e-05
- accuracy: 1.0000
```

Out[104]:

SAVE THE MODEL

In [105]:

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

TEST THE MODEL

In [106]:

```
msg='FREE MESSAGE Activate your 500 FREE Text Messages by replying to
this message with the word FREE' 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 to this message with the word FREE
THE STEMMED MESSAGE IS:   free messag activ     free text messag repli messag
word free
1/1 [==============================] - 0s 87ms/step
THE MESSAGE IS PREDICTED AS:   SPAM
```

In [107]:

```
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
ORIGINAL 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 HAPP
Y NEW Year in advance.."
THE ORIGINAL MESSAGE IS:   wish famili merri  x  ma happi new year advanc
1/1 [==============================] - 0s 8ms/step
THE MESSAGE IS PREDICTED AS:   NOT SPAM
```