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

```
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 building

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
model = Sequential()
model.add(Dense(1500,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 108ms/step - loss: 0.1166 - accuracy:
Epoch 2/10
175/175 [==============================] - 18s 102ms/step - loss: 0.0082 - accuracy:
Epoch 3/10
175/175 [==============================] - 18s 101ms/step - loss: 0.0017 - accuracy:
Epoch 4/10
175/175 [==============================] - 19s 108ms/step - loss: 3.2305e-04 - accur
Epoch 5/10
175/175 [==============================] - 19s 109ms/step - loss: 9.4276e-05 - accur
Epoch 6/10
175/175 [==============================] - 18s 102ms/step - loss: 5.1171e-05 - accur
Epoch 7/10
175/175 [==============================] - 19s 108ms/step - loss: 3.2042e-05 - accur
Epoch 8/10
175/175 [==============================] - 18s 102ms/step - loss: 2.1643e-05 - accur
Epoch 9/10
175/175 [==============================] - 18s 102ms/step - loss: 1.5546e-05 - accur
Epoch 10/10
175/175 [==============================] - 19s 108ms/step - loss: 1.1369e-05 - accur
<keras.callbacks.History at 0x7fe636fefd90>
```

Save the model

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

Testing model

```
msg='URGENT! You have won a 1 week FREE membership in our å£100,000 Prize Jackpot! Txt the
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:   URGENT! You have won a 1 week FREE membership in our å£10
THE STEMMED MESSAGE IS:   urgent    week free membership           prize jackpot  tx
1/1 [==============================] - 0s 165ms/step
THE MESSAGE IS PREDICTED AS:   SPAM
```

```
msg='My sister in law, hope you are having a great month. Just saying hey. Abiola'
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:   My sister in law, hope you are having a great month. Just
THE STEMMED MESSAGE IS:   sister law  hope great month  say hey  abiola
1/1 [==============================] - 0s 9ms/step
THE MESSAGE IS PREDICTED AS:   NOT SPAM
```