1. Importing the required library

```
import pandas as pd
import numpy as np
import nltk
import re  # Remove unwanted char.

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

# ANN Model

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

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Dell\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\stopwords.zip.
```

2. Read dataset and do pre-processing

```
df = pd.read_csv('spam.csv',encoding="ISO-8859-1")
df
```

```
     v1                                                  v2      Unnamed: 2  Unnamed: 3  Unnamed: 4
```

```
# df.loc[:, 'v2']
port = PorterStemmer()
data = []

for i in range(len(df)):
  review = df['v2'][i] # Reading data
  review = re.sub('[^a-zA-Z]',' ',review) # Removing special char.
  review = review.lower() # Convert capital letters into small letters
  review = review.split() # Split the input
  review = [port.stem(w) for w in review if w not in set(stopwords.words('english'))] # St
  review = ' '.join(review) # join words
  data.append(review)

data
```

```
['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',
 'even brother like speak treat like aid patent',
 'per request mell mell oru minnaminungint nurungu vettam set callertun caller
press copi friend callertun',
 'winner valu network custom select receivea prize reward claim call claim code
kl valid hour',
 'mobil month u r entitl updat latest colour mobil camera free call mobil updat
co free',
 'gonna home soon want talk stuff anymor tonight k cri enough today',
 'six chanc win cash pound txt csh send cost p day day tsandc appli repli hl
info',
 'urgent week free membership prize jackpot txt word claim c www dbuk net lccltd
pobox ldnw rw',
 'search right word thank breather promis wont take help grant fulfil promis
wonder bless time',
 'date sunday',
 'xxxmobilemovieclub use credit click wap link next txt messag click http wap
xxxmobilemovieclub com n qjkgighjjgcbl',
 'oh k watch',
 'eh u rememb spell name ye v naughti make v wet',
 'fine way u feel way gota b',
 'england v macedonia dont miss goal team news txt ur nation team eg england tri
wale scotland txt poboxox w wq',
 'serious spell name',
 'go tri month ha ha joke',
 'pay first lar da stock comin',
 'aft finish lunch go str lor ard smth lor u finish ur lunch alreadi',
 'fffffffffff alright way meet sooner',
 'forc eat slice realli hungri tho suck mark get worri know sick turn pizza lol',
 'lol alway convinc',
 'catch bu fri egg make tea eat mom left dinner feel love',
 'back amp pack car let know room',
 'ahhh work vagu rememb feel like lol',
 'wait still clear sure sarcast x want live us',
 'yeah got v apologet n fallen actin like spoilt child got caught till go badli
```

```
      cheer',
       'k tell anyth',
       'fear faint housework quick cuppa',
       'thank subscript rington uk mobil charg month pleas confirm repli ye repli
      charg',
       'yup ok go home look time msg xuhui go learn nd may lesson',
       'oop let know roommat done',
       'see letter b car',
       'anyth lor u decid',
       'hello saturday go text see decid anyth tomo tri invit anyth',
       'pl go ahead watt want sure great weekend abiola',
       'forget tell want need crave love sweet arabian steed mmmmmm yummi',
       'rodger burn msg tri call repli sm free nokia mobil free camcord pleas call
      deliveri tomorrow',
       'see',
       'great hope like man well endow lt gt inch',
       'call messag miss call',
```

```
cv = CountVectorizer()

x = cv.fit_transform(data).toarray()
x
len(x)
```

```
5572
```

```
# label_encoder object knows how to understand word labels.
label_encoder = preprocessing.LabelEncoder()

# Encode labels in column 'species'.
y = label_encoder.fit_transform(df['v1'])
y
```

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

3. Create Model and Add Layers

```
# ANN Block

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

4. Compile the model

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

5. Fit the model

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

```
Epoch 1/10
175/175 [==============================] - 16s 87ms/step - loss: 0.1198 - accuracy: 0
Epoch 2/10
175/175 [==============================] - 15s 86ms/step - loss: 0.0101 - accuracy: 0
Epoch 3/10
175/175 [==============================] - 15s 85ms/step - loss: 0.0018 - accuracy: 0
Epoch 4/10
175/175 [==============================] - 15s 87ms/step - loss: 2.4143e-04 - accurac
Epoch 5/10
175/175 [==============================] - 15s 87ms/step - loss: 1.0326e-04 - accurac
Epoch 6/10
175/175 [==============================] - 15s 85ms/step - loss: 4.9540e-05 - accurac
Epoch 7/10
175/175 [==============================] - 15s 85ms/step - loss: 3.1108e-05 - accurac
Epoch 8/10
175/175 [==============================] - 15s 85ms/step - loss: 2.0914e-05 - accurac
Epoch 9/10
175/175 [==============================] - 15s 86ms/step - loss: 1.5043e-05 - accurac
Epoch 10/10
175/175 [==============================] - 15s 86ms/step - loss: 1.1091e-05 - accurac
<keras.callbacks.History at 0x2b0eba99cc0>
```

6. Save The Model

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

7. Test The Model

```
# Test 1

text =  'free entri wkli comp win fa cup final tkt st may text fa receiv entri question st
print(text)
print('*'*50)
text = re.sub('[^a-zA-Z]',' ',text)
print(text)
print('*'*50)
text = text.lower()
print(text)
print('*'*50)
text = text.split()
print(text)
print('*'*50)
```

```
text = [port.stem(w) for w in text if w not in set(stopwords.words('english'))]
print(text)
print('*'*50)
text = ' '.join(text)
print(text)
print('*'*50)
text = cv.transform([text]).toarray()
print(text)
print('*'*50)
pred = model.predict(text)
print(pred)
print('*'*50)
if pred>0.5:
  print('Positive')
else: print('Negative')
```

```
free entri wkli comp win fa cup final tkt st may text fa receiv entri question std t
**************************************************
free entri wkli comp win fa cup final tkt st may text fa receiv entri question std t
**************************************************
free entri wkli comp win fa cup final tkt st may text fa receiv entri question std t
**************************************************
['free', 'entri', 'wkli', 'comp', 'win', 'fa', 'cup', 'final', 'tkt', 'st', 'may', 't
**************************************************
['free', 'entri', 'wkli', 'comp', 'win', 'fa', 'cup', 'final', 'tkt', 'st', 'may', 't
**************************************************
free entri wkli comp win fa cup final tkt st may text fa receiv entri question std t
**************************************************
[[0 0 0 ... 0 0 0]]
**************************************************
1/1 [==============================] - 0s 151ms/step
[[1.]]
**************************************************
Positive
```

```
# Test 2

text =  'ok lar doubl check wif da hair dresser alreadi said wun cut v short said cut look
print(text)
print('*'*50)
text = re.sub('[^a-zA-Z]',' ',text)
print(text)
print('*'*50)
text = text.lower()
print(text)
print('*'*50)
text = text.split()
print(text)
print('*'*50)
text = [port.stem(w) for w in text if w not in set(stopwords.words('english'))]
print(text)
print('*'*50)
text = ' '.join(text)
print(text)
```

```
print('*'*50)
text = cv.transform([text]).toarray()
print(text)
print('*'*50)
pred = model.predict(text)
print(pred)
print('*'*50)
if pred>0.5:
  print('Positive')
else: print('Negative')
```

```
ok lar doubl check wif da hair dresser alreadi said wun cut v short said cut look nic
**************************************************
ok lar doubl check wif da hair dresser alreadi said wun cut v short said cut look nic
**************************************************
ok lar doubl check wif da hair dresser alreadi said wun cut v short said cut look nic
**************************************************
['ok', 'lar', 'doubl', 'check', 'wif', 'da', 'hair', 'dresser', 'alreadi', 'said', 'v
**************************************************
['ok', 'lar', 'doubl', 'check', 'wif', 'da', 'hair', 'dresser', 'alreadi', 'said', 'v
**************************************************
ok lar doubl check wif da hair dresser alreadi said wun cut v short said cut look nic
**************************************************
[[0 0 0 ... 0 0 0]]
**************************************************
1/1 [==============================] - 0s 22ms/step
[[1.3491409e-32]]
**************************************************
Negative
```