```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
import nltk
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
```

```
df = pd.read_csv("/content/spam.csv",encoding="latin")
df.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 sav...	NaN	NaN	NaN

```
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
```

```
df.isna().sum()
```

```
v1               0
v2               0
Unnamed: 2    5522
Unnamed: 3    5560
Unnamed: 4    5566
dtype: int64
```

```
df.rename({"v1":"label","v2":"text"},inplace=True,axis=1)
```

```
df.tail()
```

	label	text	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
[illegible]	[illegible]	The guy did some bitching but I acted like	[illegible]	[illegible]	[illegible]

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['label'] = le.fit_transform(df['label'])
```

```
nltk.download("stopwords")
```

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

```
import nltk
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
```

```
import re
corpus = []
length = len(df)
```

```
for i in range(0,length):
  text = re.sub("^a-Za-Z0-9]"," " ,df["text"][i])
  text = text.lower()
  text = text.split()
  pe = PorterStemmer()
  stopword = stopwords.words("english")
  text = [pe.stem(word) for word in text if not word in set(stopword)]
  text = " ".join(text)
  corpus.append(text)
```

```
corpus
```

```
['go jurong point, crazy.. avail bugi n great world la e buffet... cine got amor wat...',
 'ok lar... joke wif u oni...',
 "free entri 2 wkli comp win fa cup final tkt 21st may 2005. text fa 87121 receiv entri question(std txt
rate)t&c' appli 08452810075over18'",
 'u dun say earli hor... u c alreadi say...',
 'nah think goe usf, live around though',
 "freemsg hey darl 3 week' word back! i'd like fun still? tb ok! xxx std chg send, å£1.50 rcv",
 'even brother like speak me. treat like aid patent.',
 "per request 'mell mell (oru minnaminungint nurungu vettam)' set callertun callers. press *9 copi friend
callertun",
 'winner!! valu network custom select receivea å£900 prize reward! claim call 09061701461. claim code kl341.
valid 12 hour only.',
 'mobil 11 month more? u r entitl updat latest colour mobil camera free! call mobil updat co free 08002986030',
 "i'm gonna home soon want talk stuff anymor tonight, k? i'v cri enough today.",
 'six chanc win cash! 100 20,000 pound txt> csh11 send 87575. cost 150p/day, 6days, 16+ tsandc appli repli hl 4
info',
 'urgent! 1 week free membership å£100,000 prize jackpot! txt word: claim no: 81010 t&c www.dbuk.net lccltd
pobox 4403ldnw1a7rw18',
 "i'v search right word thank breather. promis wont take help grant fulfil promise. wonder bless times.",
 'date sunday will!!',
 'xxxmobilemovieclub: use credit, click wap link next txt messag click here>> http://wap.
xxxmobilemovieclub.com?n=qjkgighjjgcbl',
 "oh k...i'm watch here:)",
 'eh u rememb 2 spell name... ye did. v naughti make v wet.',
 'fine thatåõ way u feel. thatåõ way gota b',
 'england v macedonia - dont miss goals/team news. txt ur nation team 87077 eg england 87077 try:wales,
scotland 4txt/ì¼1.20 poboxox36504w45wq 16+',
 'serious spell name?',
 'i\x89û÷m go tri 2 month ha ha joke',
 'ì_ pay first lar... da stock comin...',
 'aft finish lunch go str lor. ard 3 smth lor. u finish ur lunch already?',
 'ffffffffff. alright way meet sooner?',
 "forc eat slice. i'm realli hungri tho. sucks. mark get worried. know i'm sick turn pizza. lol",
 'lol alway convincing.',
 "catch bu ? fri egg ? make tea? eat mom' left dinner ? feel love ?",
 "i'm back &amp; we'r pack car now, i'll let know there' room",
 'ahhh. work. vagu rememb that! feel like? lol',
 "wait that' still clear, sure sarcast that' x want live us",
 'yeah got 2 v apologetic. n fallen actin like spoilt child got caught that. till 2! go there! badli cheers.
you?',
 'k tell anyth you.',
 'fear faint housework did? quick cuppa',
 'thank subscript rington uk mobil charg å£5/month pleas confirm repli ye no. repli charg',
 'yup... ok go home look time msg ì_ again... xuhui go learn 2nd may lesson 8am',
 "oops, i'll let know roommate' done",
 'see letter b car',
 'anyth lor... u decide...',
 "hello! how' saturday go? text see decid anyth tomo. i'm tri invit anything!",
 'pl go ahead watts. want sure. great weekend. abiola',
 'forget tell ? want , need you, crave ... ... love sweet arabian steed ... mmmmmm ... yummi',
 '07732584351 - rodger burn - msg = tri call repli sm free nokia mobil + free camcorder. pleas call 08000930705
deliveri tomorrow',
 'seeing?',
 'great! hope like man well endowed. &lt;#&gt; inches...',
 'calls..messages..miss call',
 'get hep b immunis nigeria.',
 'fair enough, anyth go on?',
 "yeah hopefully, tyler can't could mayb ask around bit",
```

```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features=35000)
x =cv.fit_transform(corpus).toarray()
```

```
y = pd.get_dummies(df['label'])
y = y.iloc[:, 1].values
```

```
import pickle
pickle.dump(cv, open('cv1.pkl','wb'))
```

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.20, random_state=0)
```

```
print("Before OverSampling, counts of label '1': {}".format(sum(y_train == 1)))
print("Before OverSampling, counts of label '0': {}  \n".format(sum(y_train == 0)))

from imblearn.over_sampling import SMOTE
sm = SMOTE(random_state = 2)
x_train_res, y_train_res = sm.fit_resample(x_train, y_train.ravel())

print('After OverSampling, the shape of train_x: {}'.format(x_train_res.shape))
print('After OverSampling, the shape of train_y: {} \n'.format(y_train_res.shape))

print("After OverSampling, counts of label '1': {}".format(sum(y_train == 1)))
print("After OverSampling, counts of label '0': {}".format(sum(y_train == 0)))
```

```
Before OverSampling, counts of label '1': 581
Before OverSampling, counts of label '0': 3876

After OverSampling, the shape of train_x: (7752, 8194)
After OverSampling, the shape of train_y: (7752,)

After OverSampling, counts of label '1': 581
After OverSampling, counts of label '0': 3876
```

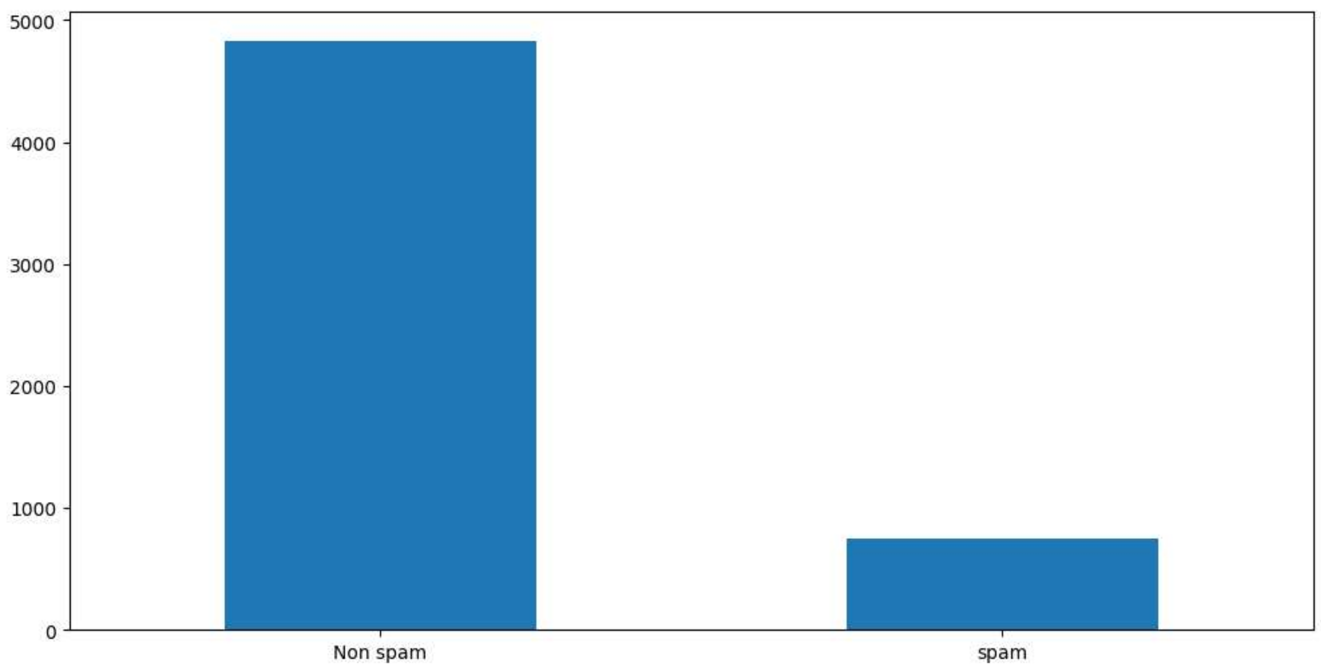
```
df.describe()
```

	label
count	5572.000000
mean	0.134063
std	0.340751
min	0.000000
25%	0.000000
50%	0.000000
75%	0.000000
max	1.000000

```
df.shape
```

```
(5572, 5)
```

```
df["label"].value_counts().plot(kind="bar",figsize=(12,6))
plt.xticks(np.arange(2),  ('Non spam', 'spam'),rotation=0);
```



```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.20, random_state=0)
```

```
from sklearn.tree import DecisionTreeClassifier
```

```
model = DecisionTreeClassifier()
model.fit(x_train_res, y_train_res)
```

```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

```
from sklearn.ensemble import  RandomForestClassifier
```

```
model = RandomForestClassifier()
model.fit(x_train_res, y_train_res)
```

```
▾ RandomForestClassifier
RandomForestClassifier()
```

```
from sklearn.naive_bayes import MultinomialNB
model = MultinomialNB()
```

```
model.fit(x_train_res, y_train_res)
```

```
▾ MultinomialNB
MultinomialNB()
```

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

```
model =  Sequential()
```

```
x_train.shape

    (4457, 8194)
```

```
model.add(Dense(units = x_train_res.shape[1],activation="relu",kernel_initializer="random_uniform"))
```

```
model.add(Dense(units =100,activation="relu",kernel_initializer="random_uniform"))
```

```
model.add(Dense(units=1,activation="sigmoid"))
```

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

```
generator = model.fit(x_train_res,y_train_res,epochs=10,steps_per_epoch=len(x_train_res)//64)

    Epoch 1/10
    121/121 [==============================] - 139s 1s/step - loss: 0.1178 - accuracy: 0.9675
    Epoch 2/10
    121/121 [==============================] - 126s 1s/step - loss: 0.0164 - accuracy: 0.9959
    Epoch 3/10
    121/121 [==============================] - 126s 1s/step - loss: 0.0089 - accuracy: 0.9982
    Epoch 4/10
    121/121 [==============================] - 123s 1s/step - loss: 0.0083 - accuracy: 0.9985
    Epoch 5/10
    121/121 [==============================] - 114s 945ms/step - loss: 0.0078 - accuracy: 0.9985
    Epoch 6/10
    121/121 [==============================] - 112s 926ms/step - loss: 0.0072 - accuracy: 0.9985
    Epoch 7/10
    121/121 [==============================] - 112s 928ms/step - loss: 0.0076 - accuracy: 0.9987
    Epoch 8/10
    121/121 [==============================] - 113s 935ms/step - loss: 0.0065 - accuracy: 0.9985
    Epoch 9/10
    121/121 [==============================] - 123s 1s/step - loss: 0.0089 - accuracy: 0.9987
    Epoch 10/10
    111/121 [==========================>...] - ETA: 9s - loss: 0.0063 - accuracy: 0.9989 WARNING:tensorflow:Your input
    121/121 [==============================] - 103s 853ms/step - loss: 0.0063 - accuracy: 0.9989
```

```
y_pred=model.predict(x_test)
y_pred

    35/35 [==============================] - 3s 92ms/step
    array([[1.9948400e-10],
           [1.7848400e-03],
           [3.4924572e-13],
           ...,
           [3.0376623e-05],
           [3.5189529e-15],
           [1.3390812e-13]], dtype=float32)
```

```
y_pr = np.where(y_pred>0.5,1,0)
```

```
y_test

    array([0, 0, 0, ..., 0, 0, 0], dtype=uint8)
```

```
from sklearn.metrics import confusion_matrix,accuracy_score
cm = confusion_matrix(y_test, y_pr)
score = accuracy_score(y_test,y_pr)
print(cm)
print('Accuracy Score Is:- ' ,score*100)

    [[933  16]
     [ 14 152]]
    Accuracy Score Is:-  97.30941704035875
```

```
def new_review(new_review):
   new_review = new_review
   new_review = re.sub('[^a-zA-Z]', ' ', new_review)
   new_review = new_review.lower()
```

```
    new_review = new_review.split()
    ps = PorterStemmer()
    all_stopwords = stopwords.words('english')
    all_stopwords.remove('not')
    new_review = [ps.stem(word) for word in new_review if not word in   set(all_stopwords)]
    new_review = ' '.join(new_review)
    new_corpus = [new_review]
    new_X_test = cv.transform(new_corpus).toarray()
    new_y_pred = model.predict(new_X_test)
    print(new_y_pred)
    new_X_pred = np.where(new_y_pred>0.5,1,0)
    return new_review
new_review = new_review(str(input("Enter new review...")))
```

```
Enter new review...hello
1/1 [==============================] - 0s 44ms/step
[[0.98991776]]
```

```
from sklearn.metrics import confusion_matrix,accuracy_score
cm=confusion_matrix(y_test,y_pr)
score = accuracy_score(y_test,y_pr)
print(cm)
print('Accuracy Score Is Naive Bayes:- ' ,score*100)
```

```
[[933  16]
 [ 14 152]]
Accuracy Score Is Naive Bayes:-  97.30941704035875
```

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

```
!pip install nbconvert
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: nbconvert in /usr/local/lib/python3.9/dist-packages (6.5.4)
Requirement already satisfied: entrypoints>=0.2.2 in /usr/local/lib/python3.9/dist-packages (from nbconvert) (0.4)
Requirement already satisfied: tinycss2 in /usr/local/lib/python3.9/dist-packages (from nbconvert) (1.2.1)
Requirement already satisfied: bleach in /usr/local/lib/python3.9/dist-packages (from nbconvert) (6.0.0)
Requirement already satisfied: jupyter-core>=4.7 in /usr/local/lib/python3.9/dist-packages (from nbconvert) (5.3.0
Requirement already satisfied: pygments>=2.4.1 in /usr/local/lib/python3.9/dist-packages (from nbconvert) (2.14.0)
Requirement already satisfied: lxml in /usr/local/lib/python3.9/dist-packages (from nbconvert) (4.9.2)
Requirement already satisfied: traitlets>=5.0 in /usr/local/lib/python3.9/dist-packages (from nbconvert) (5.7.1)
Requirement already satisfied: packaging in /usr/local/lib/python3.9/dist-packages (from nbconvert) (23.0)
Requirement already satisfied: jinja2>=3.0 in /usr/local/lib/python3.9/dist-packages (from nbconvert) (3.1.2)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.9/dist-packages (from nbconvert) (2.1.2)
Requirement already satisfied: jupyterlab-pygments in /usr/local/lib/python3.9/dist-packages (from nbconvert) (0.2
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.9/dist-packages (from nbconvert) (4.11.2)
Requirement already satisfied: nbformat>=5.1 in /usr/local/lib/python3.9/dist-packages (from nbconvert) (5.8.0)
Requirement already satisfied: nbclient>=0.5.0 in /usr/local/lib/python3.9/dist-packages (from nbconvert) (0.7.3)
Requirement already satisfied: mistune<2,>=0.8.1 in /usr/local/lib/python3.9/dist-packages (from nbconvert) (0.8.4
Requirement already satisfied: pandocfilters>=1.4.1 in /usr/local/lib/python3.9/dist-packages (from nbconvert) (1.
Requirement already satisfied: defusedxml in /usr/local/lib/python3.9/dist-packages (from nbconvert) (0.7.1)
Requirement already satisfied: platformdirs>=2.5 in /usr/local/lib/python3.9/dist-packages (from jupyter-core>=4.7
Requirement already satisfied: jupyter-client>=6.1.12 in /usr/local/lib/python3.9/dist-packages (from nbclient>=0.
Requirement already satisfied: jsonschema>=2.6 in /usr/local/lib/python3.9/dist-packages (from nbformat>=5.1->nbco
Requirement already satisfied: fastjsonschema in /usr/local/lib/python3.9/dist-packages (from nbformat>=5.1->nbcon
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.9/dist-packages (from beautifulsoup4->nbcon
Requirement already satisfied: webencodings in /usr/local/lib/python3.9/dist-packages (from bleach->nbconvert) (0.
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.9/dist-packages (from bleach->nbconvert) (1.16
Requirement already satisfied: attrs>=17.4.0 in /usr/local/lib/python3.9/dist-packages (from jsonschema>=2.6->nbfo
Requirement already satisfied: pyrsistent!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0 in /usr/local/lib/python3.9/dist-pack
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.9/dist-packages (from jupyter-client
Requirement already satisfied: tornado>=4.1 in /usr/local/lib/python3.9/dist-packages (from jupyter-client>=6.1.12
Requirement already satisfied: pyzmq>=13 in /usr/local/lib/python3.9/dist-packages (from jupyter-client>=6.1.12->n
```

```
!jupyter nbconvert --to html spam.ipynb
```

```
[NbConvertApp] Converting notebook spam.ipynb to html
[NbConvertApp] Writing 750496 bytes to spam.html
```

```
!pip install flask-ngrok
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: flask-ngrok in /usr/local/lib/python3.9/dist-packages (0.0.25)
Requirement already satisfied: requests in /usr/local/lib/python3.9/dist-packages (from flask-ngrok) (2.27.1)
Requirement already satisfied: Flask>=0.8 in /usr/local/lib/python3.9/dist-packages (from flask-ngrok) (2.2.3)
Requirement already satisfied: Jinja2>=3.0 in /usr/local/lib/python3.9/dist-packages (from Flask>=0.8->flask-ngrok
Requirement already satisfied: importlib-metadata>=3.6.0 in /usr/local/lib/python3.9/dist-packages (from Flask>=0.
Requirement already satisfied: Werkzeug>=2.2.2 in /usr/local/lib/python3.9/dist-packages (from Flask>=0.8->flask-n
Requirement already satisfied: click>=8.0 in /usr/local/lib/python3.9/dist-packages (from Flask>=0.8->flask-ngrok)
Requirement already satisfied: itsdangerous>=2.0 in /usr/local/lib/python3.9/dist-packages (from Flask>=0.8->flask
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.9/dist-packages (from requests->flask-
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.9/dist-packages (from requests-
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.9/dist-packages (from requests->fla
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.9/dist-packages (from requests->flask-ngrok)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.9/dist-packages (from importlib-metadata>=3.6.0
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.9/dist-packages (from Jinja2>=3.0->Flask>
```

```python
from flask import Flask, render_template, request
import pickle
import numpy as np
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from tensorflow.keras.models import load_model


loaded_model = load_model('spam.h5')
cv = pickle.load(open('cv1.pkl','rb'))
app = Flask(__name__)


@app.route('/')
def home():
    return render_template('home.html')


@app.route('/Spam',methods=['POST','GET'])
def prediction():
    return render_template('spam.html')


@app.route('/predict',methods=['POST'])
def predict():
    if request.method == 'POST':
       message = request.form['message']
       data = message

    new_review = str(data)
    print(new_review)
    new_review = re.sub('[^a-zA-Z]', ' ',new_review)
    new_review = new_review.lower()
    new_review = new_review.split()
    ps = PorterStemmer()
    all_stopwords = stopwords.words('english')
    all_stopwords.remove('not')
    new_review = [ps.stem(word) for word in new_review if not word in   set(all_stopwords)]
    new_review = ' '.join(new_review)
    new_corpus = [new_review]
    new_X_test = cv.transform(new_corpus).toarray()
    new_y_pred = model.predict(new_X_test)
    print(new_y_pred)
    new_X_pred = np.where(new_y_pred>0.5,1,0)
    print(new_X_pred)
    if new_review[0][0]==1:
       return render_template('result.html', prediction="Spam")
    else :
       return render_template('result.html', prediction="Not a Spam")


import os


if __name__=="__main__":
```

```
port=int(os.environ.get('PORT',5000))
app.run(debug=False)
```

```
 * Serving Flask app '__main__'
 * Debug mode: off
INFO:werkzeug:WARNING: This is a development server. Do not use it in a production deployment. Use a production WS
 * Running on http://127.0.0.1:5000
INFO:werkzeug:Press CTRL+C to quit
```

✓ 1s completed at 15:19