```
import numpy as np #scientific computation
import pandas as pd #loading dataset file
import matplotlib.pyplot as plt #visualization
from sklearn.model_selection import train_test_split #train dataset
import nltk #preprocessing
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
```

```
# read the dataset
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

```
#Give consise  summary of the dataframe
df.info()
```

To undo cell deletion use Ctrl+M Z or the Undo option in the Edit menu ×

```
    <class 'pandas.core.frame.DataFrame'>
    RangeIndex: 5572 entries, 0 to 5571
    Data columns (total 5 columns):
     #   Column      Non-Null Count  Dtype
    ---  ------      --------------  -----
     0   label       5572 non-null   int64
     1   text        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: int64(1), object(4)
    memory usage: 217.8+ KB
```

```
#return the sum of all no values
df.isna().sum()
```

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

```
#rename the dataset
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
5570	ham	The guy did some bitching but I acted like	NaN	NaN	NaN

```
#HANDILING CATEGROICAL VALUES
from sklearn.preprocessing import LabelEncoder
```

```
le = LabelEncoder()
df['label'] = le.fit_transform(df['label'])
```

```
#CLEANING THE TEXT DATA
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")
  [illegible]                                                  opword)]
  [illegible]
```

To undo cell deletion use Ctrl+M Z or the Undo option in the Edit menu ×

```
corpus
```

```
 'hey book kb sat already... lesson go an? keep sat night free need meet confirm lodg',
 'chk ur belovd ms dict',
 'time want come?',
 'awesome, lemm know whenev around',
 'shb b ok lor... thanx...',
 'beauti truth gravity.. read carefully: \\our heart feel light someon it.. feel heavi someon leav it..\\" good
night"',
 "also rememb get dobby' bowl car",
 'filthi stori girl wait',
 "sorri c ur msg... yar lor poor thing... 4 one night... tmr u'll brand new room 2 sleep in...",
 'love decision, feeling. could decid love, then, life would much simpler, less magic',
 'welp appar retir',
 "sort code acc . bank natwest. repli confirm i'v sent right person!",
 '@',
 "u sure u can't take sick time?",
 'urgent! tri contact u. today draw show å£800 prize guaranteed. call 09050001808 land line. claim m95.
valid12hr',
 'watch cartoon, listen music &amp; eve go templ &amp; church.. u?',
 'yo chad gymnast class wanna take? site say christian class full..',
 'much buzi',
 'better still catch let ask sell &lt;#&gt; ms ',
```

```
# splting datq into train and validatiob sets using train_test_split
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
```

To undo cell deletion use Ctrl+M Z or the Undo option in the Edit menu ✕

```
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
```

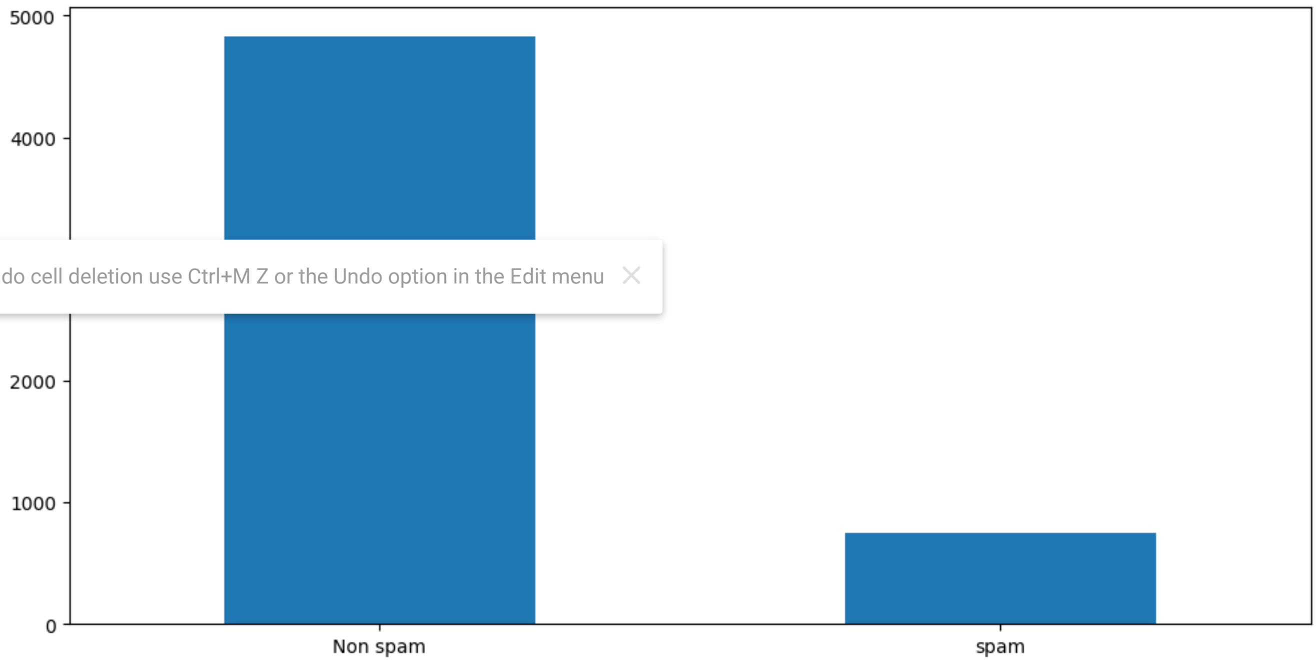
```
df.describe()
```

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

```
df.shape
```

```
(5572, 5)
```

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



To undo cell deletion use Ctrl+M Z or the Undo option in the Edit menu ✕

```
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"))
```

To undo cell deletion use Ctrl+M Z or the Undo option in the Edit menu ✕

```
="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 [==============================] - 165s 1s/step - loss: 0.6128 - accuracy: 0.9138
Epoch 2/10
121/121 [==============================] - 178s 1s/step - loss: 0.5767 - accuracy: 0.9907
Epoch 3/10
121/121 [==============================] - 167s 1s/step - loss: 0.5505 - accuracy: 0.9928
Epoch 4/10
121/121 [==============================] - 187s 2s/step - loss: 0.5260 - accuracy: 0.9933
Epoch 5/10
121/121 [==============================] - 186s 2s/step - loss: 0.5037 - accuracy: 0.9922
Epoch 6/10
121/121 [==============================] - 206s 2s/step - loss: 0.4815 - accuracy: 0.9932
Epoch 7/10
121/121 [==============================] - 171s 1s/step - loss: 0.4605 - accuracy: 0.9940
Epoch 8/10
121/121 [==============================] - 182s 2s/step - loss: 0.4405 - accuracy: 0.9940
Epoch 9/10
121/121 [==============================] - 194s 2s/step - loss: 0.4228 - accuracy: 0.9939
Epoch 10/10
111/121 [==========================>...] - ETA: 13s - loss: 0.4058 - accuracy: 0.9936WARNING:tensorflow:Your input
121/121 [==============================] - 155s 1s/step - loss: 0.4058 - accuracy: 0.9936
```

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

```
35/35 [==============================] - 7s 174ms/step
array([[0.46443313],
            [0.47209898],
            [0.39208782],
            ...,
            [0.4862801 ],
            [0.5039984 ],
            [0.4511185 ]], dtype=float32)
```

```
y_pred1=model.predict(x_train)
y_pred1
```

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

```
y_pred1 = np.where(y_pred>0.5,1,0)
```

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

```
y_test
```

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

To undo cell deletion use Ctrl+M Z or the Undo option in the Edit menu ✕

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

```
y_pred1 = np.where(y_pred>0.5,1,0)
```

```
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)
```

```
     [[824 125]
      [154  12]]
     Accuracy Score Is:-  74.97757847533633
```

```
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...how are you
     1/1 [==============================] - 0s 64ms/step
     [[0.5]]
```

```
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)
```

```
[[824 125]
 [154  12]]
Accuracy Score Is Naive Bayes:-  74.97757847533633
```

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

```
from sklearn.metrics import confusion_matrix,accuracy_score
cm1=confusion_matrix(y_test,y_pred1)
score = accuracy_score(y_test,y_pred1)
print(cm1)
print('Accuracy Score Is Naive Bayes:- ' ,score*100)
```

```
[[824 125]
 [154  12]]
Accuracy Score Is Naive Bayes:-  74.97757847533633
[[824 125]
```

To undo cell deletion use Ctrl+M Z or the Undo option in the Edit menu ×

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

```
from sklearn.svm import SVC
```

```
svm1=SVC(kernel='rbf')
```

```
svm1.fit(x_train_res, y_train_res)
```

```
▾ SVC
SVC()
```

```
y_pred4=svm1.predict(x_test)
from sklearn.metrics import accuracy_score
svm_rbf=accuracy_score(y_test,y_pred4)
svm_rbf
```

```
0.8986547085201794
```

```
svm2=SVC(kernel='sigmoid')
svm2.fit(x_train, y_train)
```

```
▾         SVC
SVC(kernel='sigmoid')
```

```
y_pred5=svm2.predict(x_test)
from sklearn.metrics import accuracy_score
svm_sig=accuracy_score(y_test,y_pred5)
svm_sig
```

```
0.9739910313901345
```

```
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier()
model.fit(x_train, y_train)
```

```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

```
y_pred6=model.predict(x_test)
from sklearn.metrics import accuracy_score
dec_tree=accuracy_score(y_test,y_pred6)
dec_tree
```

```
0.9721973094170404
```

To undo cell deletion use Ctrl+M Z or the Undo option in the Edit menu ×

✓ 0s completed at 8:28 PM ● ×