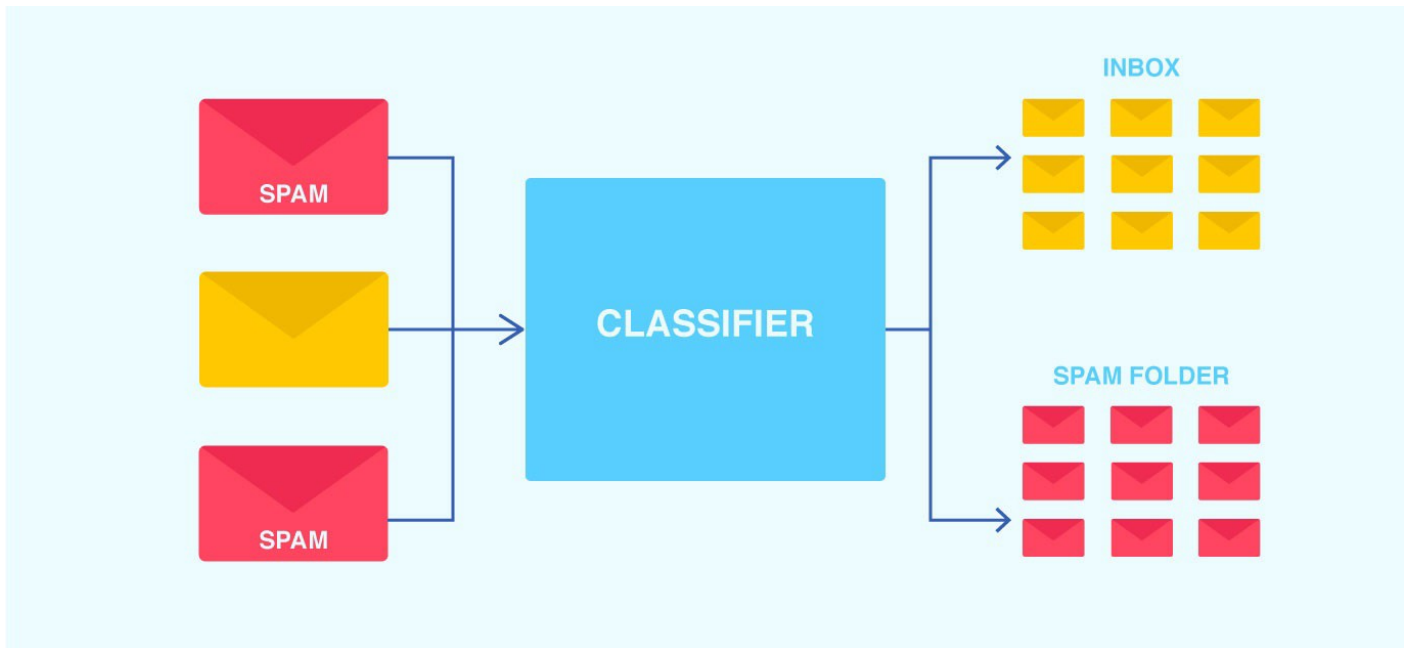


SMS Spam Classification

Submitted by :- Ambarish Singh



In [47]:

```
1 ## Comment
2 ## Observation
```

In [1]:

```
1 ## Importing Pandas Library
2 import pandas as pd
```

In [2]:

```
1 ## Importing Dataset
2 data = pd.read_csv("https://raw.githubusercontent.com/sunnysavita10/Naive-Bayes/main/SpamClassifier-with-ML/sm")
```

In [3]:

```
1 ## Checking Top 5 Rows
2 data.head()
```

Out[3]:

	label	message
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 [4]:

```
1 ## Checking Message Column Meassage first Rows.
2 data["message"][0]
```

Out[4]:

```
'Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...'
```

In [5]:

```
## Checking Message Column Meassage 40th Rows.  
data["message"][40]
```

Out[5]:

'Pls go ahead with watts. I just wanted to be sure. Do have a great weekend. Abiola'

In [6]:

```
## Checking Shape of a dataset  
data.shape
```

Out[6]:

(5572, 2)

In [7]:

```
## Importing NLTK (Natural Language Toolkit)  
import nltk
```

In [8]:

```
## Importing Regular Expression  
import re
```

In [9]:

```
## Downloading Soppwords from NLTK Library  
nltk.download("stopwords")
```

[nltk_data] Downloading package stopwords to
[nltk_data] C:\Users\Lenovo\AppData\Roaming\nltk_data...
[nltk_data] Package stopwords is already up-to-date!

Out[9]:

True

In [10]:

```
## Importing stopwords  
from nltk.corpus import stopwords
```

In [11]:

```
## Importing PorterStemmer  
from nltk.stem.porter import PorterStemmer
```

In [12]:

```
ps = PorterStemmer()
```

In [13]:

```
## Checking all Stopworks present in NLTK
stopwords.words('english')
```

Out[13]:

```
['i',
 'me',
 'my',
 'myself',
 'we',
 'our',
 'ours',
 'ourselves',
 'you',
 "you're",
 "you've",
 "you'll",
 "you'd",
 'your',
 'yours',
 'yourself',
 'yourselves',
 'he']
```

In [14]:

```
##Checking Sub-string
rev = re.sub('[^a-zA-Z]', ' ', data['message'][0])
```

In [15]:

```
rev
```

Out[15]:

```
'Go until jurong point  crazy   Available only in bugis n great world la e buffet   Cine there got amore wat '
```

In [16]:

```
## Ching all sub-string into lower case
rev.lower()
```

Out[16]:

```
'go until jurong point  crazy   available only in bugis n great world la e buffet   cine there got amore wat '
```

In [17]:

```
## Splitting every sub-string
rev.split()
```

Out[17]:

```
['Go',
 'until',
 'jurong',
 'point',
 'crazy',
 'Available',
 'only',
 'in',
 'bugis',
 'n',
 'great',
 'world',
 'la',
 'e',
 'buffet',
 'Cine',
 'there',
 'got',
 'amore',
 'wat']
```

In [18]:

```
## Seperating every sub-string which is not present in stopwords  
rev = [ps.stem(word) for word in rev if not word in stopwords.words('english')]
```

In [19]:

```
rev
```

Out[19]:

```
['g',  
,  
'u',  
'n',  
'l',  
In [20]:
```

```
['u' ## creating black list as corpus  
'r' corpus = []  
'n',  
,  
'g',
```

```
In [21]:
```

```
['p' ## doing for Loop and extracting all sentences which ignore stopwords and return in Lower case.  
'n' for i in range(0,len(data)):  
, review= re.sub('[^a-zA-Z]', ' ',data['message'][i])  
, review= review.lower()  
'c' review= review.split()  
'r',  
,  
'z' review= [ps.stem(word) for word in review if not word in stopwords.words('english')]  
, review = ' '. join(review)  
, corpus.append(review)  
,  
'a',  
'v',  
'l',  
'b',  
'l',
```

```
In [22]:
```

```
['n' ## Returning 1st sentences which are without Stopwords present in Lower cases  
'l' corpus[0]  
'l',
```

```
Out [22]:
```

```
['n',  
'go'jurong point crazi avail bugi n great world la e buffet cine got amor wat'  
'b',
```

```
In [23]:
```

```
['g',  
'n' ## Returning all sentences which are without Stopwords present in Lower cases  
'n' corpus  
,
```

```
Out [23]:
```

```
['n',  
'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 ap  
p1',  
'u' dun say earli hor u c already 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 c  
allertun',  
'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',  
'u'onna 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',  
'xxx'mobilemovieclub use credit click wan link next txt messag click httpn wan xxxmobilemovieclub c  
,
```

```
In [24]:
```

```
['n' ## Importing CountVectorizer from sklearn  
'e' from sklearn.feature_extraction.text import CountVectorizer  
,
```

```
In [25]:
```

```
['n' cv = CountVectorizer(max_features = 2500)  
'e',  
,  
'g',  
,  
'n',  
'e',  
,  
'w',
```

```
In [26]:
```

```
## Fitting the model  
X = cv.fit_transform(corpus).toarray()
```

```
In [27]:
```

```
#1 set of unique words  
#2 finally it is creating a vector  
  
X.shape
```

```
Out[27]:
```

```
(5572, 2500)
```

```
In [28]:
```

```
X[0]
```

```
Out[28]:
```

```
array([0, 0, 0, ..., 0, 0, 0], dtype=int64)
```

```
In [29]:
```

```
len(X[0])
```

```
Out[29]:
```

```
2500
```

```
In [30]:
```

```
data['label']
```

```
Out[30]:
```

```
0      ham  
1      ham  
2      spam  
3      ham  
4      ham  
...  
5567    spam  
5568    ham  
5569    ham  
5570    ham  
5571    ham
```

```
Name: label, Length: 5572, dtype: object
```

```
In [31]:
```

```
y = pd.get_dummies(data['label'], drop_first=True)
```

```
In [32]:
```

```
X
```

```
Out[32]:
```

```
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]], dtype=int64)
```


In [33]:

```
y
```

Out[33]:

	spam
0	0
1	0
2	1
3	0
4	0
...	...
5567	1
5568	0
5569	0
5570	0
5571	0

5572 rows × 1 columns

Splitting Train_Test_

In [34]:

```
## Splitting Train_Test_  
from sklearn.model_selection import train_test_split
```

In [35]:

```
X_train, X_test, y_train, y_test=train_test_split(X,y,test_size=0.25,random_state=10)
```

Building Model using GaussianNB

In [36]:

```
from sklearn.naive_bayes import GaussianNB
```

In [37]:

```
model=GaussianNB()
```

In [38]:

```
## Fitting a GaussianNB Model.  
model.fit(X_train,y_train)
```

C:\Users\Lenovo\anaconda3\lib\site-packages\sklearn\utils\validation.py:1111: DataConversionWarning:
A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

Out[38]:

```
▼ GaussianNB  
GaussianNB()
```

In [39]:

```
## Predicting the model
y_pred=model.predict(X_test)
```

In [40]:

```
## Importing accuracy_score from sklearn
from sklearn.metrics import accuracy_score
```

In [41]:

```
## Checkig Accuracy of the Model
accuracy_score(y_test,y_pred)
```

Out[41]:

0.8628858578607322

Building Model using MultinomialNB

In [42]:

```
from sklearn.naive_bayes import MultinomialNB
```

In [43]:

```
model2=MultinomialNB()
```

In [44]:

```
## Fitting the MultinomialNB Model
model2.fit(X_train,y_train)
```

C:\Users\Lenovo\anaconda3\lib\site-packages\sklearn\utils\validation.py:1111: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

Out[44]:

```
▼ MultinomialNB
MultinomialNB()
```

In [45]:

```
## Predicting the Model
y_pred2=model2.predict(X_test)
```

In [46]:

```
## Checking the accuracy_score of the model.
accuracy_score(y_test,y_pred2)
```

Out[46]:

0.9770279971284996

Thank You