Guided Project Report

Applications in Natural Language Processing

Name: Shruti Verma
Course: AI and ML
(Batch 4)
Duration: 10 months

Problem Statement: Using BoW and NLTK for processing, implement a simple spam filter that marks all the spam texts as dangerous.

Prerequisites

What things you need to install the software and how to install them:

Python 3.8 or higher versions This setup requires that your machine has latest version of python. The following url https://www.python.org/downloads/ can be referred to download python. Once you have python downloaded and installed, you will need to setup PATH variables (if you want to run python program directly, detail instructions are below in how to run software section). To do that check this: https://www.pythoncentral.io/add-python-to-path-python-is-not- recognized-as-an-internal-or-external- command/. Setting up PATH variable is optional as you can also run program without it and more instruction are given below on this topic.

Second and easier option is to download anaconda and use its anaconda prompt to run the commands. To install anaconda check this url https://www.anaconda.com/download/ You will also need to download and install below 3 packages after you install either python or anaconda from the steps above Sklearn (scikit-learn) numpy scipy if you have chosen to install python 3.8 then run below commands in command prompt/terminal to install these packages pip install -U scikit-learn pip install numpy pip install scipy if you have chosen to install anaconda then run below commands in anaconda prompt to install these packages conda install -c scikit-learn conda install -c anaconda numpy conda install -c anaconda scipy . Install nltk.

Video Link

https://drive.google.com/file/d/1yF7Gt2BxeM40db4m_wHbI4vayuF9v7cH/view?usp=sharing

Dataset used

Any paragraph

Education is a process of learning through which spam we acquire knowledge.It enlightens, empowers, and creates spam a positive development.

Education gives an individual the knowledge and skills to work with virtue.

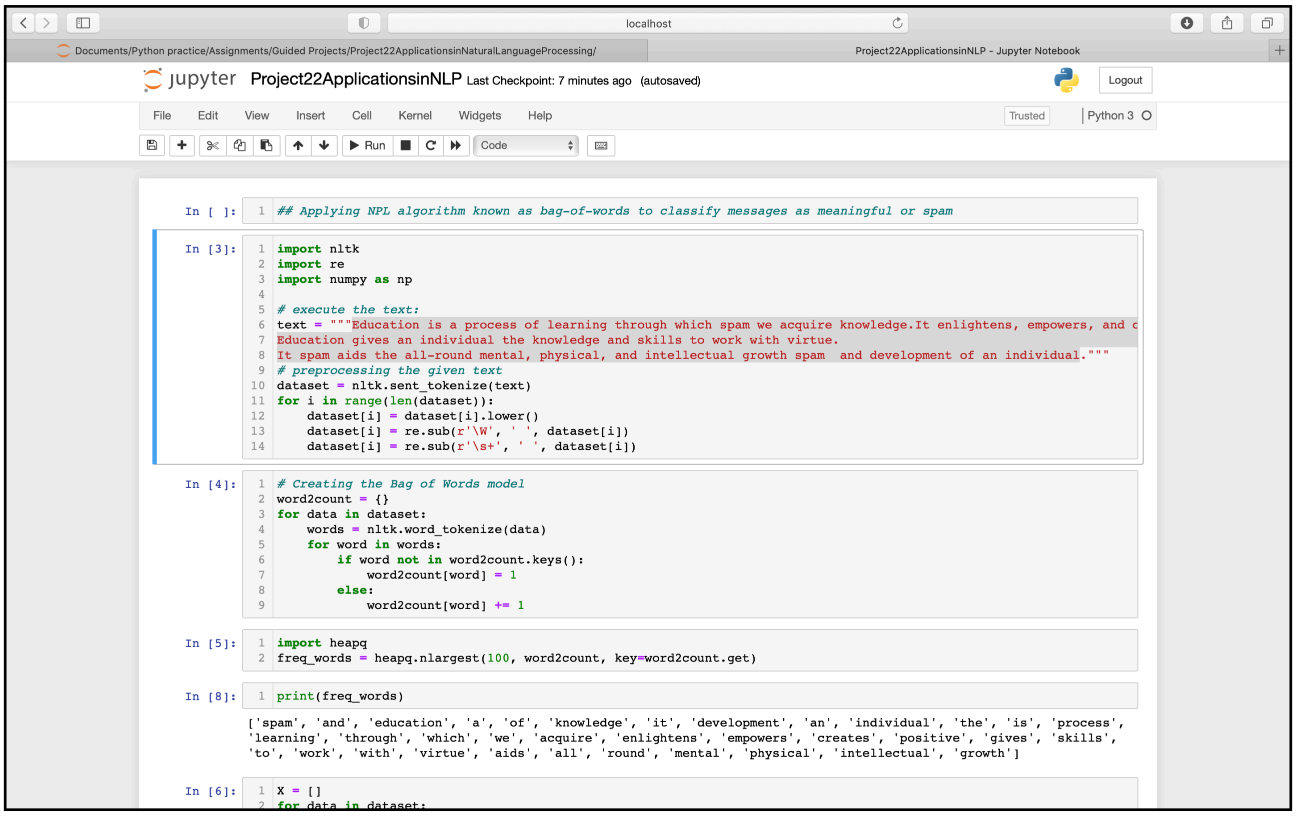
It spam aids the all-round mental, physical, and intellectual growth spam and development of an individual

Method used for detection

- Data reading
- Preprocessing
- Classification as spam

Importing the libraries and capturing images:

Importing necessary libraries and reading the store data


localhost
Documents/Python practice/Assignments/Guided Projects/Project22ApplicationsinNaturalLanguageProcessing/
Project22ApplicationsinNLP - Jupyter Notebook
jupyter Project22ApplicationsinNLP Last Checkpoint: 7 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

In [ ]:
```
## Applying NPL algorithm known as bag-of-words to classify messages as meaningful or spam
```

In [3]:
```
import nltk
import re
import numpy as np

# execute the text:
text = """Education is a process of learning through which spam we acquire knowledge.It enlightens, empowers, and c
Education gives an individual the knowledge and skills to work with virtue.
It spam aids the all-round mental, physical, and intellectual growth spam  and development of an individual."""
# preprocessing the given text
dataset = nltk.sent_tokenize(text)
for i in range(len(dataset)):
    dataset[i] = dataset[i].lower()
    dataset[i] = re.sub(r'\W', ' ', dataset[i])
    dataset[i] = re.sub(r'\s+', ' ', dataset[i])
```

In [4]:
```
# Creating the Bag of Words model
word2count = {}
for data in dataset:
    words = nltk.word_tokenize(data)
    for word in words:
        if word not in word2count.keys():
            word2count[word] = 1
        else:
            word2count[word] += 1
```

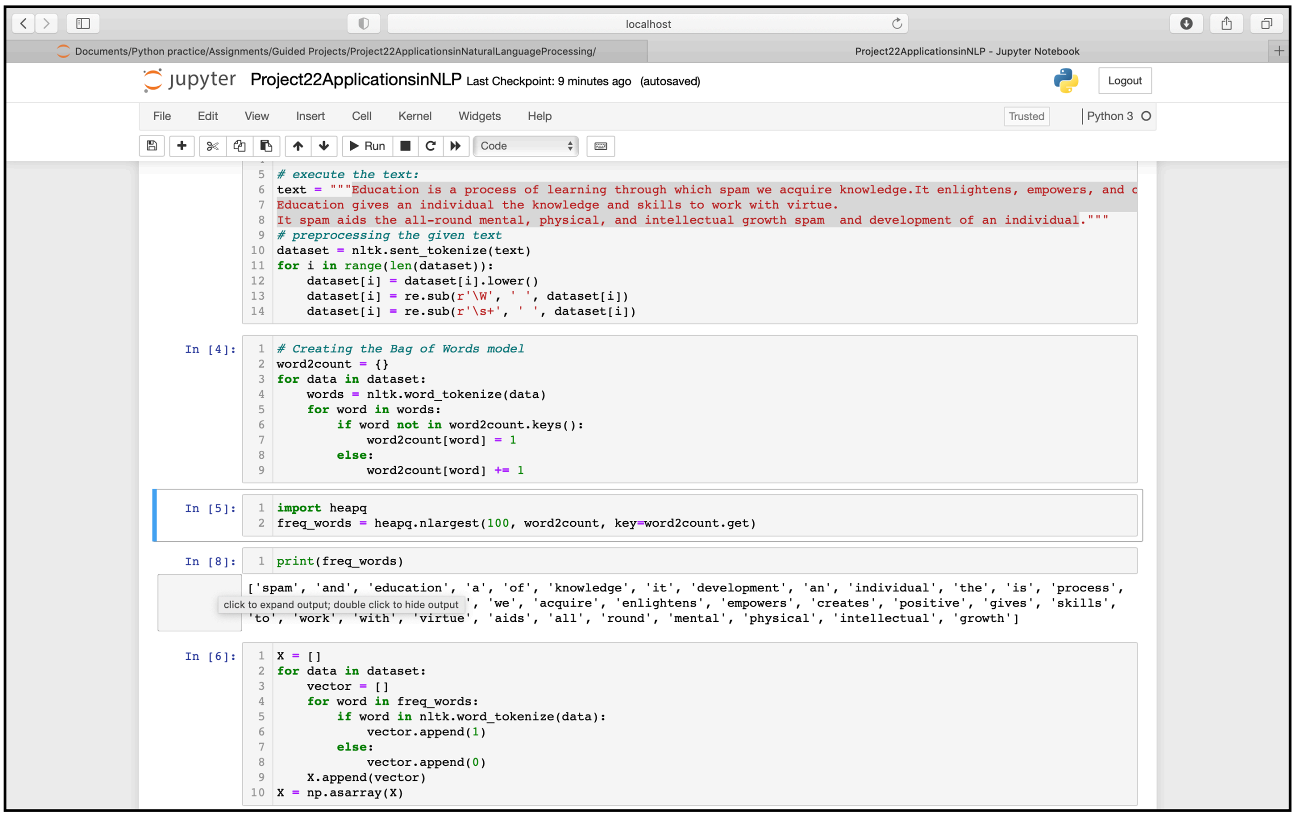
In [5]:
```
import heapq
freq_words = heapq.nlargest(100, word2count, key=word2count.get)
```

In [8]:
```
print(freq_words)
```

```
['spam', 'and', 'education', 'a', 'of', 'knowledge', 'it', 'development', 'an', 'individual', 'the', 'is', 'process',
'learning', 'through', 'which', 'we', 'acquire', 'enlightens', 'empowers', 'creates', 'positive', 'gives', 'skills',
'to', 'work', 'with', 'virtue', 'aids', 'all', 'round', 'mental', 'physical', 'intellectual', 'growth']
```

In [6]:
```
X = []
for data in dataset:
```

Creating the BoW model and checking the frequency of words


localhost
Documents/Python practice/Assignments/Guided Projects/Project22ApplicationsinNaturalLanguageProcessing/
Project22ApplicationsinNLP - Jupyter Notebook
jupyter Project22ApplicationsinNLP Last Checkpoint: 9 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
click to expand output; double click to hide output

```
# execute the text:
text = """Education is a process of learning through which spam we acquire knowledge.It enlightens, empowers, and c
Education gives an individual the knowledge and skills to work with virtue.
It spam aids the all-round mental, physical, and intellectual growth spam  and development of an individual."""
# preprocessing the given text
dataset = nltk.sent_tokenize(text)
for i in range(len(dataset)):
    dataset[i] = dataset[i].lower()
    dataset[i] = re.sub(r'\W', ' ', dataset[i])
    dataset[i] = re.sub(r'\s+', ' ', dataset[i])
```

In [4]:

```
# Creating the Bag of Words model
word2count = {}
for data in dataset:
    words = nltk.word_tokenize(data)
    for word in words:
        if word not in word2count.keys():
            word2count[word] = 1
        else:
            word2count[word] += 1
```

In [5]:

```
import heapq
freq_words = heapq.nlargest(100, word2count, key=word2count.get)
```

In [8]:

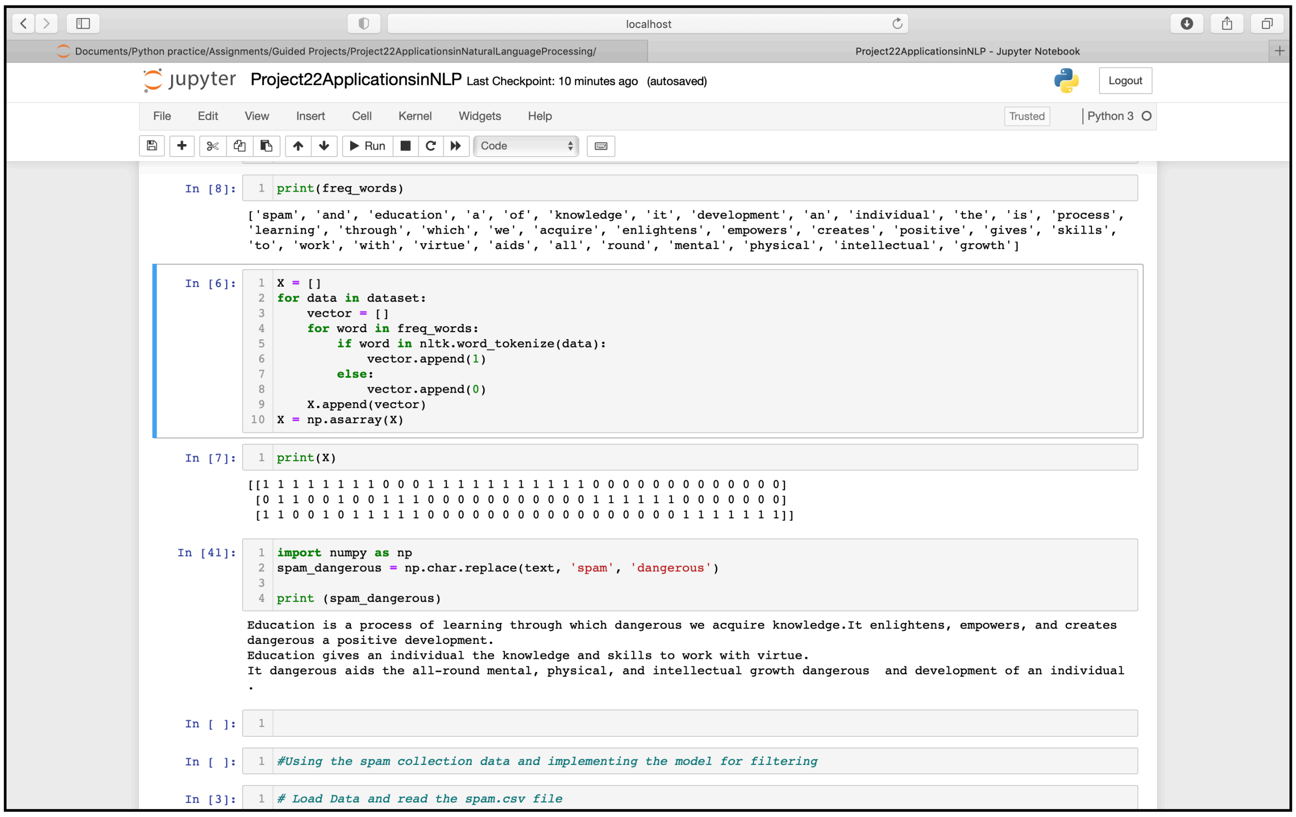
```
print(freq_words)
```

```
['spam', 'and', 'education', 'a', 'of', 'knowledge', 'it', 'development', 'an', 'individual', 'the', 'is', 'process',
                                          ', 'we', 'acquire', 'enlightens', 'empowers', 'creates', 'positive', 'gives', 'skills',
'to', 'work', 'with', 'virtue', 'aids', 'all', 'round', 'mental', 'physical', 'intellectual', 'growth']
```

In [6]:

```
X = []
for data in dataset:
    vector = []
    for word in freq_words:
        if word in nltk.word_tokenize(data):
            vector.append(1)
        else:
            vector.append(0)
    X.append(vector)
X = np.asarray(X)
```

Replacing the spam word with dangerous


localhost
Documents/Python practice/Assignments/Guided Projects/Project22ApplicationsinNaturalLanguageProcessing/
Project22ApplicationsinNLP - Jupyter Notebook
jupyter Project22ApplicationsinNLP Last Checkpoint: 10 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

In [8]:

```
print(freq_words)
```

```
['spam', 'and', 'education', 'a', 'of', 'knowledge', 'it', 'development', 'an', 'individual', 'the', 'is', 'process',
'learning', 'through', 'which', 'we', 'acquire', 'enlightens', 'empowers', 'creates', 'positive', 'gives', 'skills',
'to', 'work', 'with', 'virtue', 'aids', 'all', 'round', 'mental', 'physical', 'intellectual', 'growth']
```

In [6]:

```
X = []
for data in dataset:
    vector = []
    for word in freq_words:
        if word in nltk.word_tokenize(data):
            vector.append(1)
        else:
            vector.append(0)
    X.append(vector)
X = np.asarray(X)
```

In [7]:

```
print(X)
```

```
[[1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 1 1 0 0 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0]
 [1 1 0 0 1 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1]]
```

In [41]:

```
import numpy as np
spam_dangerous = np.char.replace(text, 'spam', 'dangerous')

print (spam_dangerous)
```

```
Education is a process of learning through which dangerous we acquire knowledge.It enlightens, empowers, and creates
dangerous a positive development.
Education gives an individual the knowledge and skills to work with virtue.
It dangerous aids the all-round mental, physical, and intellectual growth dangerous  and development of an individual
.
```

In [ ]:

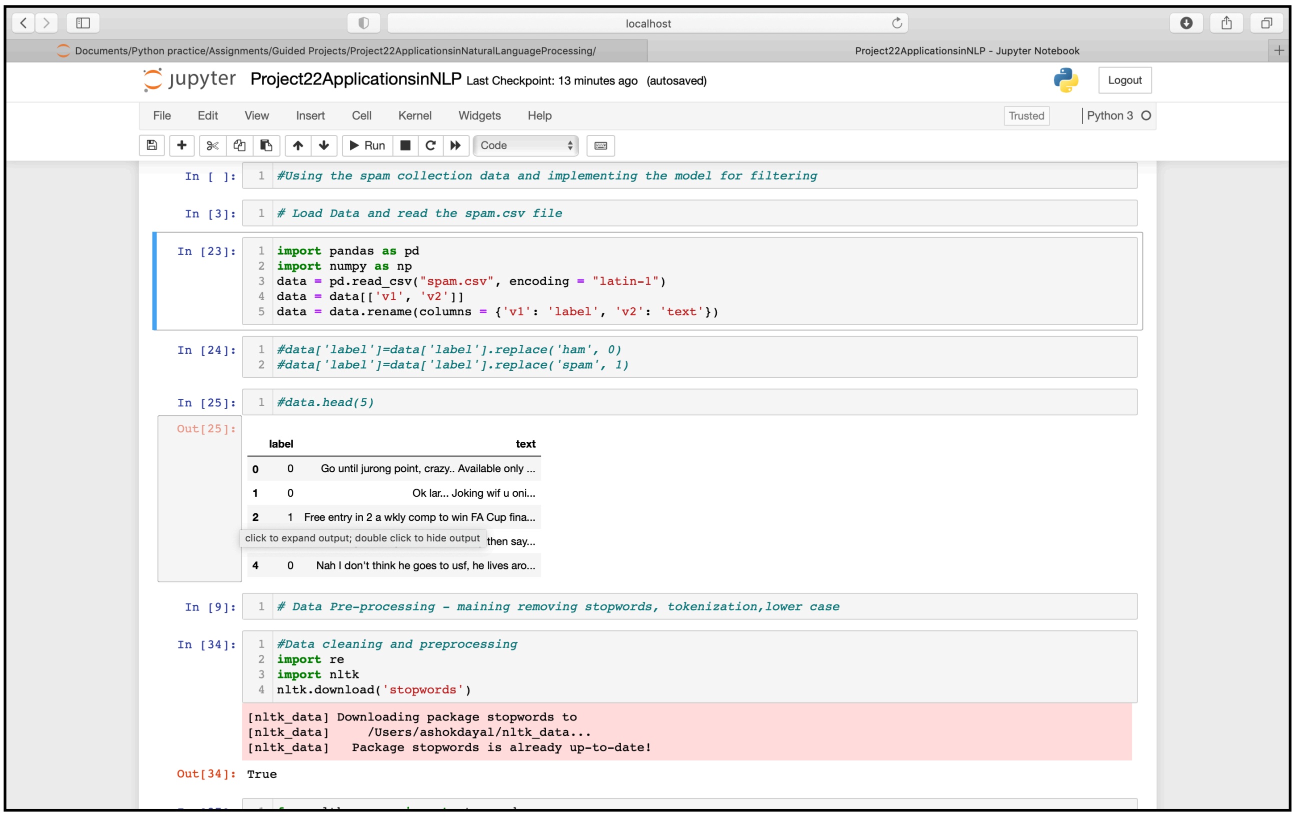
In [ ]:

```
#Using the spam collection data and implementing the model for filtering
```

In [3]:

```
# Load Data and read the spam.csv file
```

Load and read spam.csv



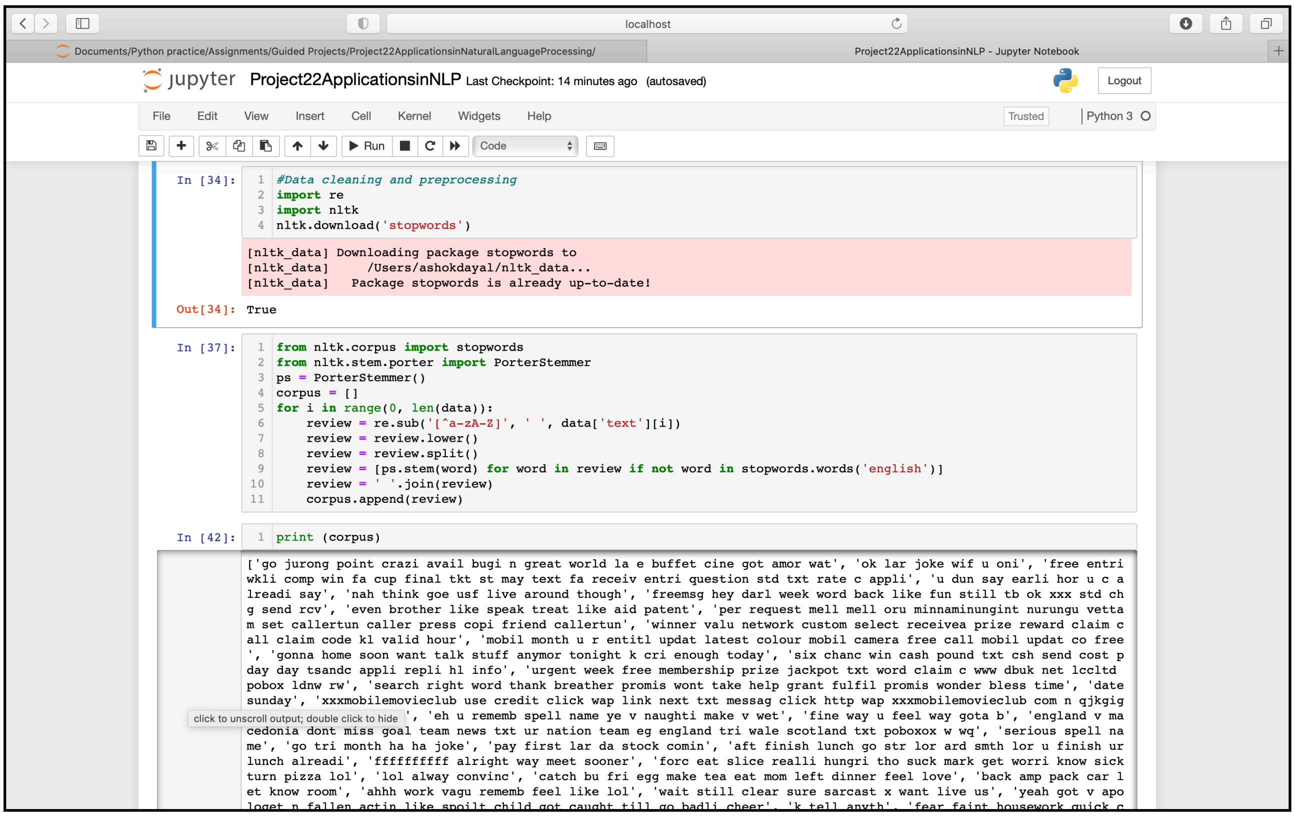
```
In [ ]:  #Using the spam collection data and implementing the model for filtering

In [3]:  # Load Data and read the spam.csv file

In [23]: import pandas as pd
         import numpy as np
         data = pd.read_csv("spam.csv", encoding = "latin-1")
         data = data[['v1', 'v2']]
         data = data.rename(columns = {'v1': 'label', 'v2': 'text'})

In [24]: #data['label']=data['label'].replace('ham', 0)
         #data['label']=data['label'].replace('spam', 1)

In [25]: #data.head(5)
```

Out[25]:

	label	text
0	0	Go until jurong point, crazy.. Available only ...
1	0	Ok lar... Joking wif u oni...
2	1	Free entry in 2 a wkly comp to win FA Cup fina...
		then say...
4	0	Nah I don't think he goes to usf, he lives aro...

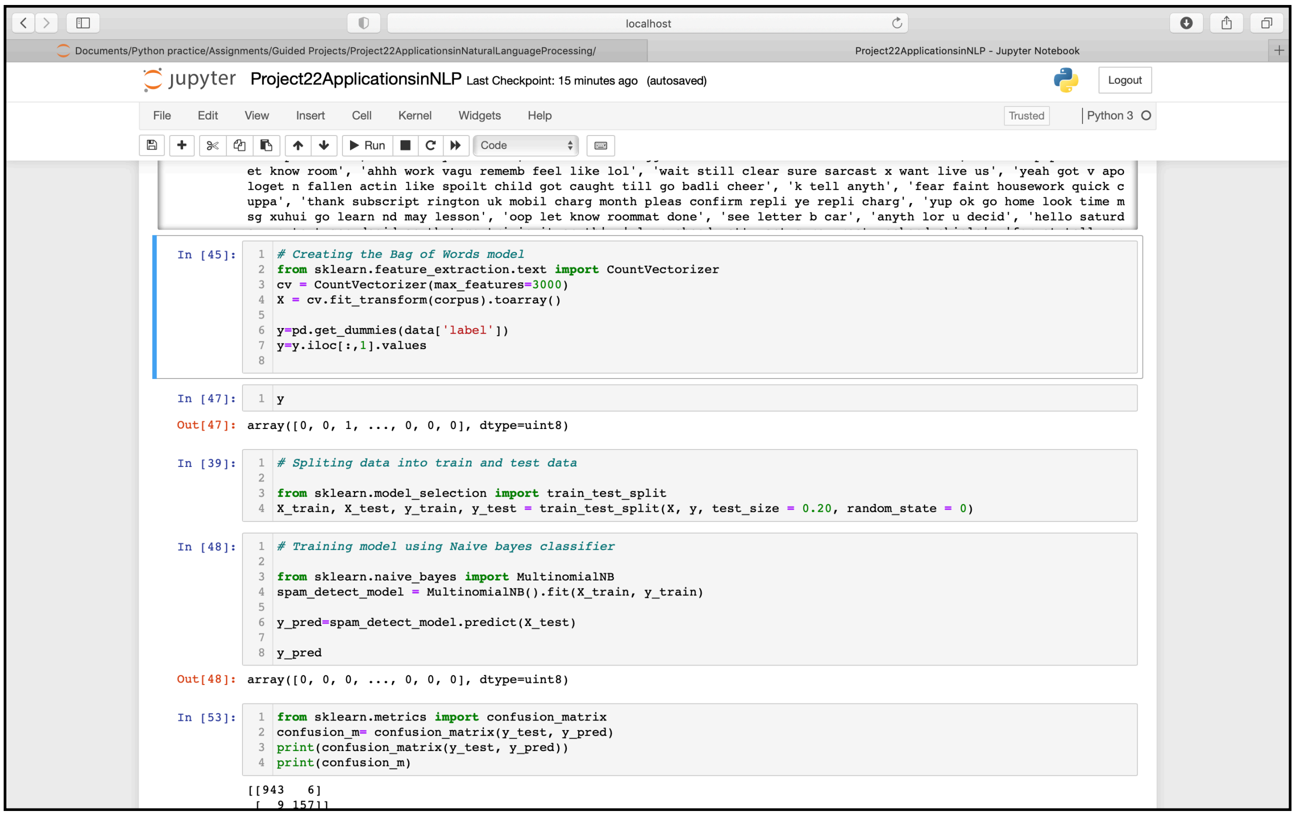
```
In [9]:  # Data Pre-processing - maining removing stopwords, tokenization,lower case

In [34]: #Data cleaning and preprocessing
         import re
         import nltk
         nltk.download('stopwords')

[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/ashokdayal/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!

Out[34]: True
```

Data cleaning and preprocessing



```
In [34]: #Data cleaning and preprocessing
         import re
         import nltk
         nltk.download('stopwords')

[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/ashokdayal/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!

Out[34]: True

In [37]: from nltk.corpus import stopwords
         from nltk.stem.porter import PorterStemmer
         ps = PorterStemmer()
         corpus = []
         for i in range(0, len(data)):
             review = re.sub('[^a-zA-Z]', ' ', data['text'][i])
             review = review.lower()
             review = review.split()
             review = [ps.stem(word) for word in review if not word in stopwords.words('english')]
             review = ' '.join(review)
             corpus.append(review)

In [42]: print (corpus)

['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 a
lreadi say', 'nah think goe usf live around though', 'freemsg hey darl week word back like fun still tb ok xxx std ch
g send rcv', 'even brother like speak treat like aid patent', 'per request mell mell oru minnaminungint nurungu vetta
m set callertun caller press copi friend callertun', 'winner valu network custom select receivea prize reward claim c
all 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 qjkgig
', 'eh u rememb spell name ye v naughti make v wet', 'fine way u feel way gota b', 'england v ma
cedonia dont miss goal team news txt ur nation team eg england tri wale scotland txt poboxox w wq', 'serious spell na
me', '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', 'fffffffff 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 l
et know room', 'ahhh work vagu rememb feel like lol', 'wait still clear sure sarcast x want live us', 'yeah got v apo
loget n fallen actin like spoilt child got caught till go badli cheer', 'k tell anyth', 'fear faint housework quick c
```

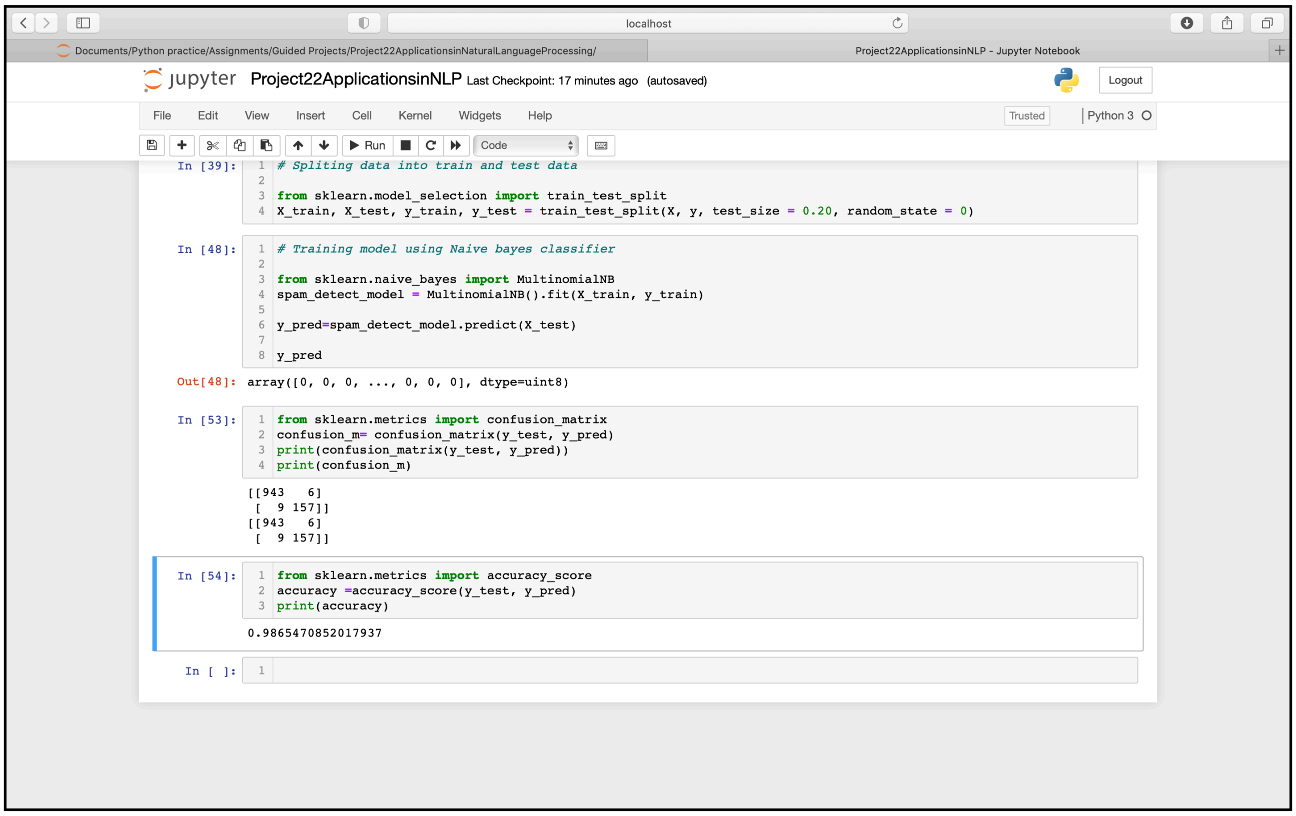
Applying BoW model and splitting data into train and test


localhost
Documents/Python practice/Assignments/Guided Projects/Project22ApplicationsinNaturalLanguageProcessing/
Project22ApplicationsinNLP - Jupyter Notebook
jupyter Project22ApplicationsinNLP Last Checkpoint: 15 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
et know room', 'ahhh work vagu rememb feel like lol', 'wait still clear sure sarcast x want live us', 'yeah got v apo
loget n fallen actin like spoilt child got caught till go badli cheer', 'k tell anyth', 'fear faint housework quick c
uppa', 'thank subscript rington uk mobil charg month pleas confirm repli ye repli charg', 'yup ok go home look time m
sg xuhui go learn nd may lesson', 'oop let know roommat done', 'see letter b car', 'anyth lor u decid', 'hello saturd
```

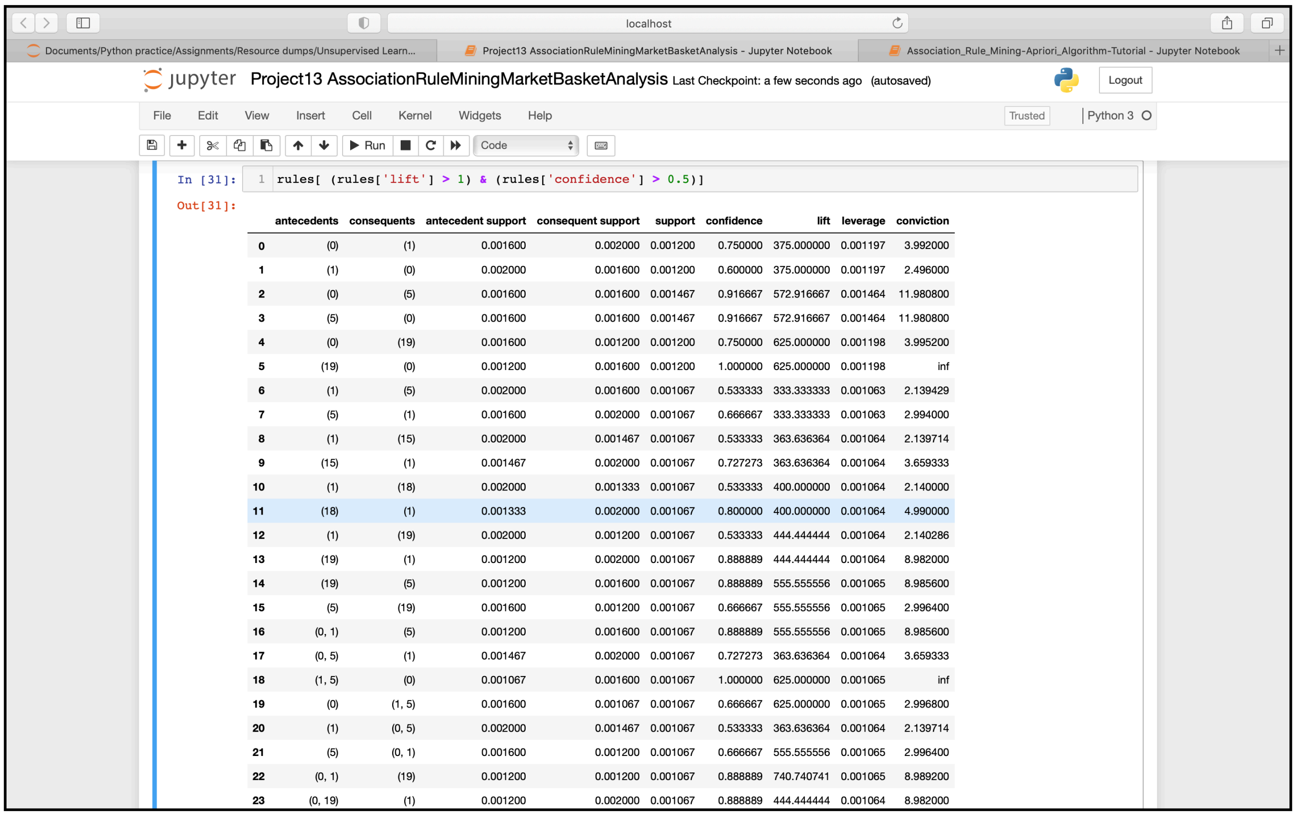
```
In [45]:
# Creating the Bag of Words model
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features=3000)
X = cv.fit_transform(corpus).toarray()

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

In [47]:
y
Out[47]: array([0, 0, 1, ..., 0, 0, 0], dtype=uint8)

In [39]:
# Spliting data into train and test data

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)

In [48]:
# Training model using Naive bayes classifier

from sklearn.naive_bayes import MultinomialNB
spam_detect_model = MultinomialNB().fit(X_train, y_train)

y_pred=spam_detect_model.predict(X_test)

y_pred
Out[48]: array([0, 0, 0, ..., 0, 0, 0], dtype=uint8)

In [53]:
from sklearn.metrics import confusion_matrix
confusion_m= confusion_matrix(y_test, y_pred)
print(confusion_matrix(y_test, y_pred))
print(confusion_m)

[[943   6]
 [  9 157]]
```

Using Classifier and checking the accuracy


localhost
Documents/Python practice/Assignments/Guided Projects/Project22ApplicationsinNaturalLanguageProcessing/
Project22ApplicationsinNLP - Jupyter Notebook
jupyter Project22ApplicationsinNLP Last Checkpoint: 17 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
In [39]:
# Spliting data into train and test data

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)

In [48]:
# Training model using Naive bayes classifier

from sklearn.naive_bayes import MultinomialNB
spam_detect_model = MultinomialNB().fit(X_train, y_train)

y_pred=spam_detect_model.predict(X_test)

y_pred
Out[48]: array([0, 0, 0, ..., 0, 0, 0], dtype=uint8)

In [53]:
from sklearn.metrics import confusion_matrix
confusion_m= confusion_matrix(y_test, y_pred)
print(confusion_matrix(y_test, y_pred))
print(confusion_m)
```

```
[[943   6]
 [  9 157]]
[[943   6]
 [  9 157]]

In [54]:
from sklearn.metrics import accuracy_score
accuracy =accuracy_score(y_test, y_pred)
print(accuracy)

0.9865470852017937

In [ ]:
```



```
In [31]: rules[ (rules['lift'] > 1) & (rules['confidence'] > 0.5)]
Out[31]:
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(0)	(1)	0.001600	0.002000	0.001200	0.750000	375.000000	0.001197	3.992000
1	(1)	(0)	0.002000	0.001600	0.001200	0.600000	375.000000	0.001197	2.496000
2	(0)	(5)	0.001600	0.001600	0.001467	0.916667	572.916667	0.001464	11.980800
3	(5)	(0)	0.001600	0.001600	0.001467	0.916667	572.916667	0.001464	11.980800
4	(0)	(19)	0.001600	0.001200	0.001200	0.750000	625.000000	0.001198	3.995200
5	(19)	(0)	0.001200	0.001600	0.001200	1.000000	625.000000	0.001198	inf
6	(1)	(5)	0.002000	0.001600	0.001067	0.533333	333.333333	0.001063	2.139429
7	(5)	(1)	0.001600	0.002000	0.001067	0.666667	333.333333	0.001063	2.994000
8	(1)	(15)	0.002000	0.001467	0.001067	0.533333	363.636364	0.001064	2.139714
9	(15)	(1)	0.001467	0.002000	0.001067	0.727273	363.636364	0.001064	3.659333
10	(1)	(18)	0.002000	0.001333	0.001067	0.533333	400.000000	0.001064	2.140000
11	(18)	(1)	0.001333	0.002000	0.001067	0.800000	400.000000	0.001064	4.990000
12	(1)	(19)	0.002000	0.001200	0.001067	0.533333	444.444444	0.001064	2.140286
13	(19)	(1)	0.001200	0.002000	0.001067	0.888889	444.444444	0.001064	8.982000
14	(19)	(5)	0.001200	0.001600	0.001067	0.888889	555.555556	0.001065	8.985600
15	(5)	(19)	0.001600	0.001200	0.001067	0.666667	555.555556	0.001065	2.996400
16	(0, 1)	(5)	0.001200	0.001600	0.001067	0.888889	555.555556	0.001065	8.985600
17	(0, 5)	(1)	0.001467	0.002000	0.001067	0.727273	363.636364	0.001064	3.659333
18	(1, 5)	(0)	0.001067	0.001600	0.001067	1.000000	625.000000	0.001065	inf
19	(0)	(1, 5)	0.001600	0.001067	0.001067	0.666667	625.000000	0.001065	2.996800
20	(1)	(0, 5)	0.002000	0.001467	0.001067	0.533333	363.636364	0.001064	2.139714
21	(5)	(0, 1)	0.001600	0.001200	0.001067	0.666667	555.555556	0.001065	2.996400
22	(0, 1)	(19)	0.001200	0.001200	0.001067	0.888889	740.740741	0.001065	8.989200
23	(0, 19)	(1)	0.001200	0.002000	0.001067	0.888889	444.444444	0.001064	8.982000