In [2]:

```
import pandas as pd
import re
import string
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import CountVectorizer
from sklearn import feature_extraction, linear_model, model_selection, preprocessing
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.pipeline import Pipeline
from sklearn.svm import LinearSVC
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.naive_bayes import BernoulliNB
from sklearn.naive_bayes import MultinomialNB
from sklearn.naive_bayes import MultinomialNB

from sklearn.ensemble import GradientBoostingClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import SGDClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.gaussian_process import GaussianProcessClassifier
from sklearn.metrics import accuracy_score, confusion_matrix,classification_report

from nltk.stem.porter import *
import gensim
import string
from nltk. tokenize import word_tokenize
from nltk.corpus import stopwords
import nltk
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
tweets_train = pd.read_csv('train1.csv')
```

In [3]:

```
import warnings
warnings.filterwarnings('ignore')
```

In [4]:

```
tweets_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27448 entries, 0 to 27447
Data columns (total 2 columns):
text         27447 non-null object
sentiment    27448 non-null object
dtypes: object(2)
memory usage: 429.0+ KB
```

In [5]:

```
#We can see an extra textID, to make consistent dropping it
tweets_train.dropna(inplace=True)
```

In [6]:

```
tweets_train.head()
```

Out[6]:

	text	sentiment
0	oh Marly, I`m so sorry!! I hope you find her...	neutral
1	Playing Ghost Online is really interesting. Th...	positive
2	is cleaning the house for her family who is co...	neutral
3	gotta restart my computer .. I thought Win7 wa...	neutral
4	SEe waT I Mean bOuT FoLL0w fRiiDaYs... It`S cA...	neutral

In [7]:

```
tweets_train['sentiment'].value_counts()
```

Out[7]:

```
neutral     11105
positive     8575
negative     7767
Name: sentiment, dtype: int64
```

In [8]:

```
#Citation: Borrowed a few regex'es from Google
def process_tweets(text):
    text = str(text).lower() #lower
    text = re.sub('\[.*?\]', '', text) #Remove text in square brackets
    text = re.sub('https?://\S+|www\.\S+', '', text) #Hyperlinks removal
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text) #punctuations
    text = re.sub('\n', '', text) #newlines
    text = re.sub('\w*\d\w*', '', text) #word containing numbers
    return text
```

In [9]:

```
#Pre-process the tweets
tweets_train['text'] = tweets_train['text'].apply(lambda x:process_tweets(x))
```

In [10]:

```
tweets_train.head()
```

Out[10]:

	text	sentiment
0	oh marly im so sorry i hope you find her soon	neutral
1	playing ghost online is really interesting the...	positive
2	is cleaning the house for her family who is co...	neutral
3	gotta restart my computer i thought was supp...	neutral
4	see wat i mean bout friidays its called lose ...	neutral

In [11]:

```
#Stemming
stemmer = PorterStemmer()
tokenized_tweet = tweets_train['text'].apply(lambda x: x.split()) #split on tokens
tokenized_tweet = tokenized_tweet.apply(lambda x: [stemmer.stem(i) for i in x]) # s
print(tokenized_tweet.head())
tweets_training_set = []
for item in tokenized_tweet:
    tweets_training_set.append(' '.join(item))
print (len(tweets_training_set))
```

```
0    [oh, marli, im, so, sorri, i, hope, you, find,...
1    [play, ghost, onlin, is, realli, interest, the...
2    [is, clean, the, hous, for, her, famili, who, ...
3    [gotta, restart, my, comput, i, thought, wa, s...
4    [see, wat, i, mean, bout, friiday, it, call, l...
Name: text, dtype: object
27447
```

In [12]:

```
tweets_train['Analyzed_Tweet'] = tweets_training_set
tweets_train.head()
```

Out[12]:

	text	sentiment	Analyzed_Tweet
0	oh marly im so sorry i hope you find her soon	neutral	oh marli im so sorri i hope you find her soon
1	playing ghost online is really interesting the...	positive	play ghost onlin is realli interest the new up...
2	is cleaning the house for her family who is co...	neutral	is clean the hous for her famili who is com la...
3	gotta restart my computer i thought was supp...	neutral	gotta restart my comput i thought wa suppos to...
4	see wat i mean bout friidays its called lose ...	neutral	see wat i mean bout friiday it call lose frida...

In [13]:

```
_test,y_train,y_test = train_test_split(tweets_train['Analyzed_Tweet'],
                                        tweets_train.sentiment, test_size=0.2, rando
```

In [14]:

```
print (x_train.shape)
print (y_train.shape)
print (x_test.shape)
print (y_test.shape)
```

```
(21957,)
(21957,)
(5490,)
(5490,)
```

In [17]:

```
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import ShuffleSplit
cv = ShuffleSplit(n_splits=5, test_size=0.3, random_state=0)
print ("Model : GENERATIVE MODEL BASED ON NAIVE BAYES")

pipe = Pipeline([('vect', CountVectorizer(stop_words='english')),
                 ('tfidf', TfidfTransformer()),
                 ('model', MultinomialNB())])

scores = cross_val_score(pipe, x_train, y_train, cv=cv)
print ("\n Cross Validation Scores on the training set: ", scores)


modelNB = pipe.fit(x_train, y_train)
prediction = modelNB.predict(x_test)
print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
print("\nCONFUSION MATRIX: \n\n", confusion_matrix(y_test, prediction))
print("\nCLASSIFICATION REPORT: \n\n", classification_report(y_test, prediction))
```

```
Model : GENERATIVE MODEL BASED ON NAIVE BAYES

 Cross Validation Scores on the training set:  [0.62128112 0.61369156
0.61885246 0.61020036 0.61566485]
accuracy: 61.38%
accuracy: 61.38%

CONFUSION MATRIX: 

 [[ 635  866   68]
 [ 145 1771  287]
 [  55  699  964]]

CLASSIFICATION REPORT: 

               precision    recall  f1-score   support

    negative       0.76      0.40      0.53      1569
     neutral       0.53      0.80      0.64      2203
    positive       0.73      0.56      0.63      1718

    accuracy                           0.61      5490
   macro avg       0.67      0.59      0.60      5490
weighted avg       0.66      0.61      0.61      5490
```

In [ ]: