TWITTER SENTIMENT ANALYSIS

GROUP 5

FINAL REPORT

NLP - 15 CSE 358

CB.EN.U4CSE17014	CH. SAI PHANI JASWANTH
CB.EN.U4CSE17043	NANDITHA MENON
CB.EN.U4CSE17065	T. SAI SRIHITHA REDDY

INTRODUCTION

Social media today has become a very popular communication tool among Internet users. Public and private opinion about a wide variety of subjects are expressed and spread persistently by means of various social media. Twitter is one of the social media platforms that is picking up popularity.

Twitter is an American microblogging and social networking service on which users post and interact with messages known as "tweets". Registered users can post, like and retweet tweets, but unregistered users can only read them. Users access Twitter through its website interface. Twitter has become increasingly popular with academics as well as students, policymakers, politicians and the general public.

Sentiment analysis (or opinion mining) uses natural language processing and machine learning to interpret and classify emotions in subjective data. Sentiment analysis is often used in business to detect sentiment in social data, gauge brand reputation, and understand customers. Sentiment analysis models focus on polarity (positive, negative, neutral) but also on feelings and emotions (angry, happy, sad, etc), and even on intentions (e.g. interested v. not interested). Sentiment Analysis also helps organisations look far beyond just the number of likes/shares/comments they get on an ad campaign, blog post, released product, or anything of that nature. With everything shifting online, Brands have started giving utmost importance to Sentiment Analysis. Social Media listening can help organisations from any domain understand the grievances and concerns of their customers – which eventually helps the organisations scale up their services. Sentiment Analysis helps brands tackle the exact problems or concerns of their customers. Research shows that news articles and social media can hugely influence the stock market. News with overall positive sentiment has been observed to relate to a large increase in price albeit for a short period of time. On the other hand, negative news is seen to be linked to a decrease in price – but with more prolonged effects.
Sentiment analysis gives an organisation the much-needed insights on their customers. Organisations can now adjust their marketing strategies depending on how the customers are responding to it. Sentiment Analysis also helps organisations measure the ROI of their marketing campaigns and improve their customer service. Since sentiment analysis gives the organisations a sneak peek into their customer's emotions, they can be aware of any crisis that's to come well in time – and manage it accordingly.

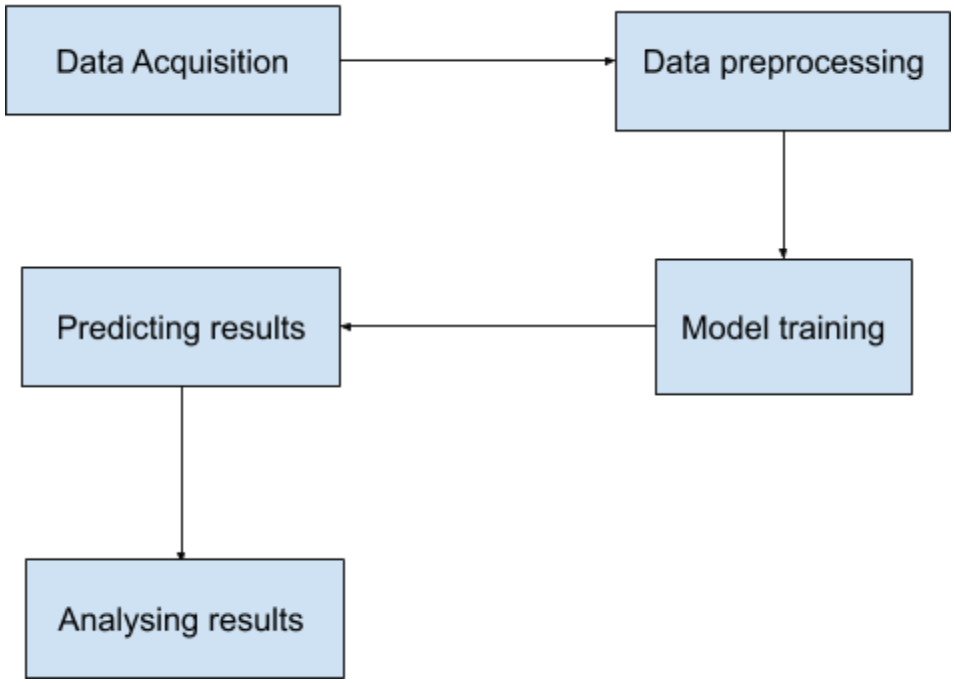
OBJECTIVE

- To implement an algorithm incorporating Naive Bayes classifier multi-label classification for automatic classification of tweet into ‘positive’, ‘negative’, ‘neutral’ and ‘irrelevant’.
- To observe the graphical representation of hashtags grouped according to sentiment.

PROBLEM STATEMENT

The problem in sentiment analysis is classifying the polarity of a given tweet as ‘positive’, ‘negative’, ‘neutral’ or ‘irrelevant’ at the feature/aspect level.

ARCHITECTURE



RELATED WORK

1. https://www.irjet.net/archives/V6/i3/IRJET-V6I393.pdf
2. https://arxiv.org/ftp/arxiv/papers/1711/1711.10377.pdf

NOVELTY OF THE WORK

An aspect of social media data such as twitter messages is that it includes rich structured information about the individuals involved in the communication. It can lead to more accurate tools for extracting semantic information. It provides means for empirically studying properties of social interactions.

Sentiment analysis is extremely useful in social media monitoring as it allows us to gain an overview of the wider public opinion behind certain topics. The applications of sentiment analysis are broad and powerful. The ability to extract insights from social data is a practice that is being widely adopted by organisations across the world. Shifts in sentiment on social media

have been shown to correlate with shifts in the stock market. It can also be an essential part of your market research and customer service approach. Not only can you see what people think of your own products or services, you can see what they think about your competitors too. The overall customer experience of users can be revealed quickly with sentiment analysis.

DATA COLLECTION AND PREPARATION

The corpus used in this case study is Niek Sanders' Corpus of over 5000 hand-classified tweets. Being hand-classified makes this corpus quite reliable to evaluate the training models.

```
tweetDataFile
```

	Topic	Sentiment	TweetId	TweetDate	TweetText
0	apple	positive	126415614616154112	Tue Oct 18 21:53:25 +0000 2011	Now all @Apple has to do is get swype on the i...
1	apple	positive	126404574230740992	Tue Oct 18 21:09:33 +0000 2011	@Apple will be adding more carrier support to ...
2	apple	positive	126402758403305474	Tue Oct 18 21:02:20 +0000 2011	Hilarious @youtube video - guy does a duet wit...
3	apple	positive	126397179614068736	Tue Oct 18 20:40:10 +0000 2011	@RIM you made it too easy for me to switch to ...
4	apple	positive	126395626979196928	Tue Oct 18 20:34:00 +0000 2011	I just realized that the reason I got into twi...
...	...	...	...	...	...
5108	twitter	irrelevant	126855687060987904	Thu Oct 20 03:02:07 +0000 2011	me re copè con #twitter
5109	twitter	irrelevant	126855171702661120	Thu Oct 20 03:00:04 +0000 2011	Buenas noches genteeee :) #twitter los quieroo...
5110	twitter	irrelevant	126854999442587648	Thu Oct 20 02:59:23 +0000 2011	#twitter tiene la mala costumbre de ponerce bn...
5111	twitter	irrelevant	126854818101858304	Thu Oct 20 02:58:40 +0000 2011	Oi @flaviasansi. Muito bem vinda ao meu #Twitt...
5112	twitter	irrelevant	126854423317188608	Thu Oct 20 02:57:06 +0000 2011	Eles arrastaram os barcos para a praia, deixar...

Sample dataset snippet

IMPLEMENTATION

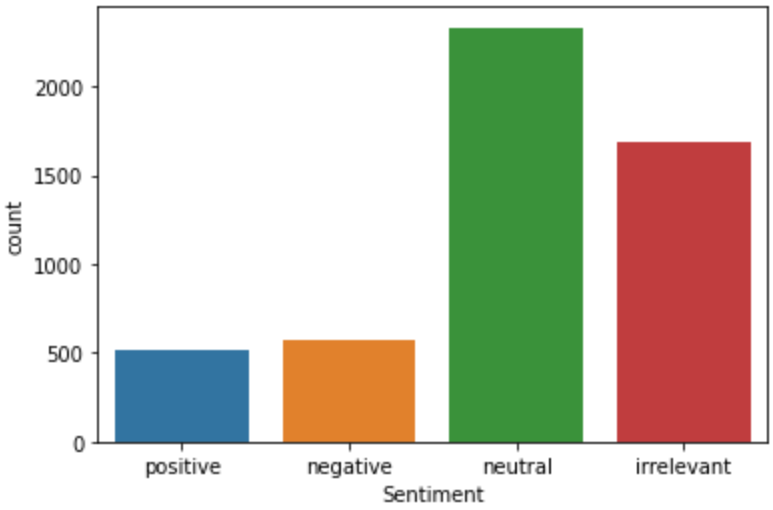
VISUALISATION

```
[93] tweetDataFile.Sentiment.unique()

     array(['positive', 'negative', 'neutral', 'irrelevant'], dtype=object)
```

The corpus has 4 sentiment labels - positive, negative, neutral, irrelevant

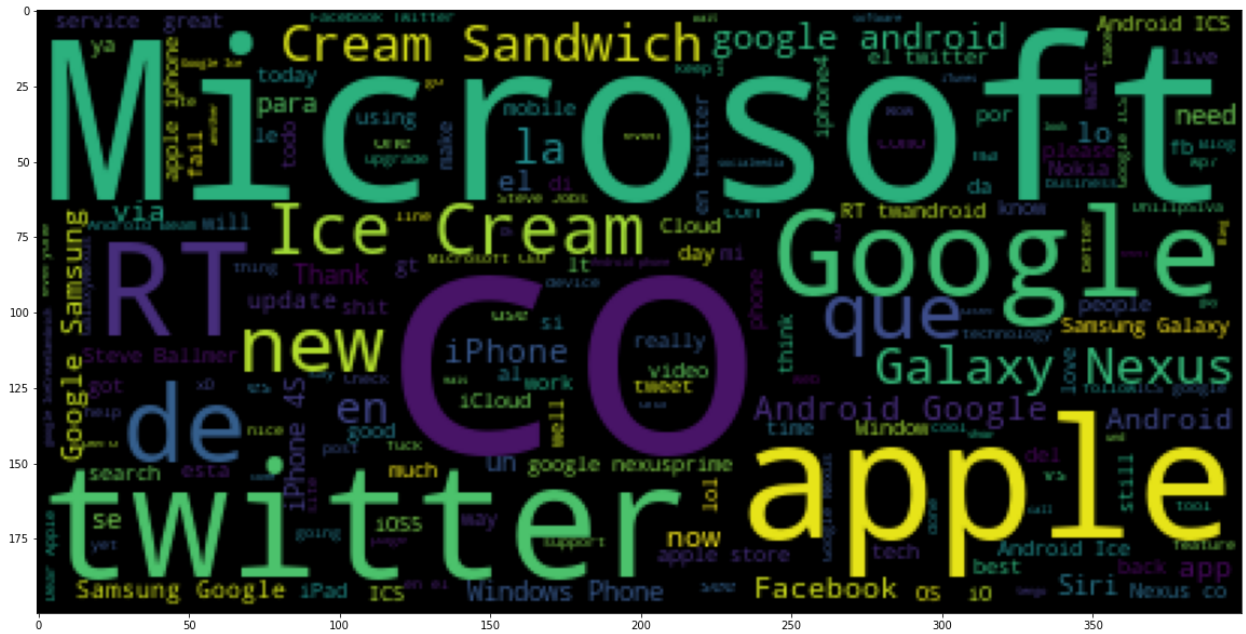
COUNT PLOT ON SENTIMENT COLUMN



WORD CLOUD VISUALISATION

A Wordcloud (or Tag cloud) is a visual representation of text data. It displays a list of words, the importance of each being shown with font size or color.

```
[132] from wordcloud import WordCloud
      sentences = tweetDataFile['TweetText'].tolist()
      sentences_as_one_string = " ".join(sentences)
      plt.figure(figsize=(20,20))
      plt.imshow(WordCloud().generate(sentences_as_one_string))
```

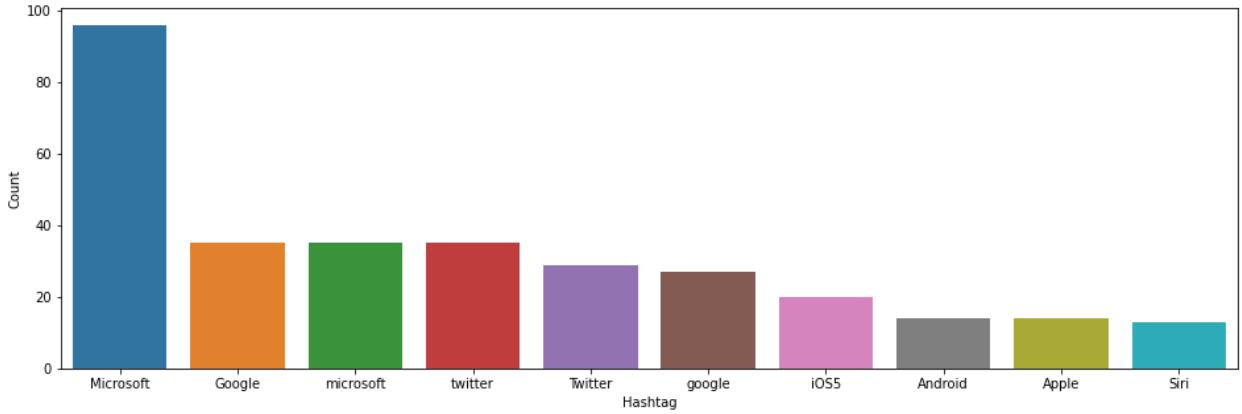


HASHTAG ANALYSIS

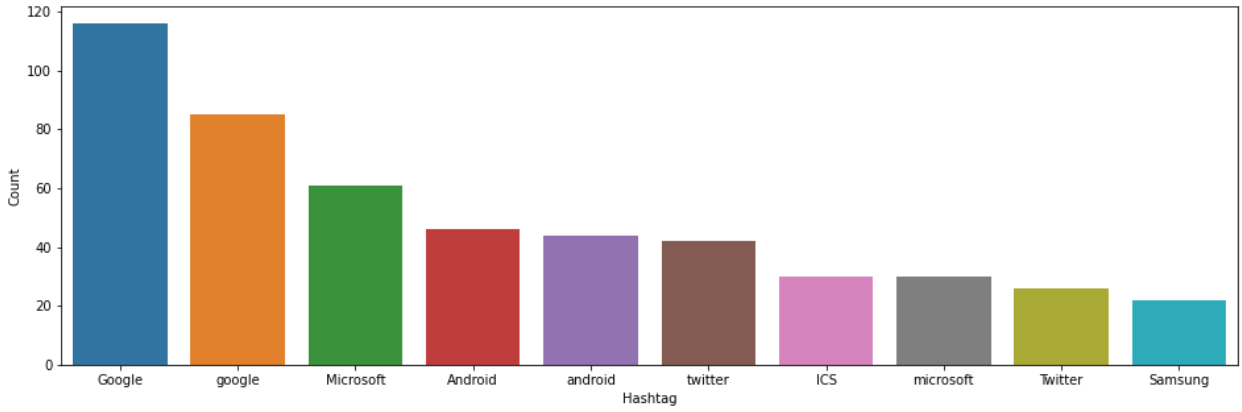
Separating the hashtags of each tweet based on the sentiment of the tweet and plotting it in the form of couplot gives the following visualisations. This form of analysis can be used to see which hashtags are associated with positive sentiment or negative sentiment and how much are they associated at a quick glance.

```
def hashtag_extract(x):
    hashtags = []
    # Loop over the words in the tweet
    for i in x:
        #print(i)
        ht = re.findall(r"#(\w+)", i)
        hashtags.append(ht)
    return hashtags
```

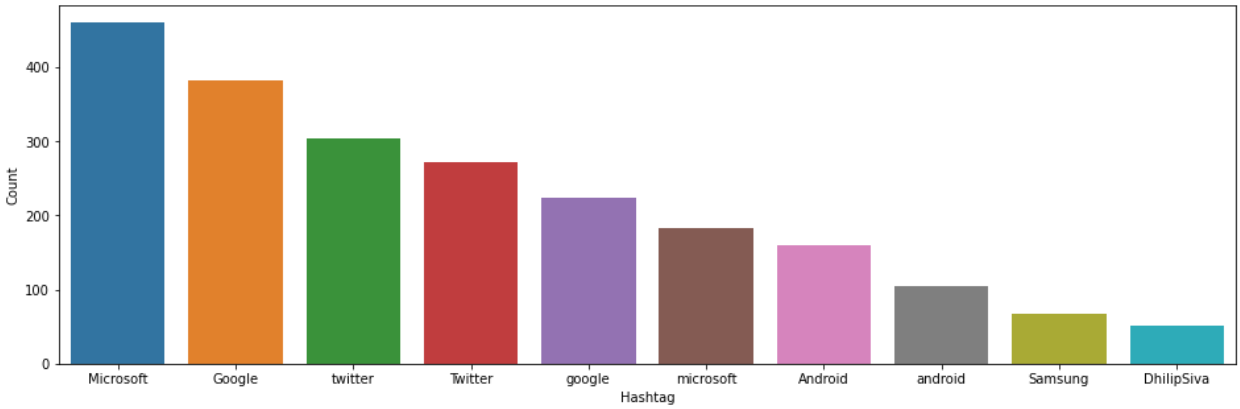
Positive sentiment hashtag countplot



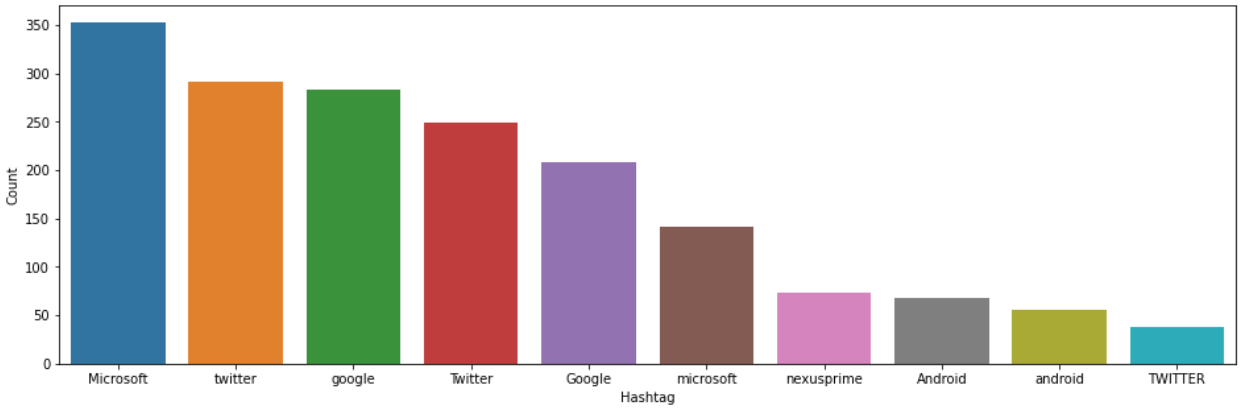
Negative sentiment hashtag analysis



Neutral sentiment hashtag analysis



Irrelevant sentiment hashtag analysis



PREPROCESSING THE DATA

The dataset is split into a training set and a test set using train_test_split function.
Then the tweets are preprocessed and tokenized with the help of the below function.
The stop words are downloaded from nltk.corpus. And we ignore any user tagged or url in the tweet.

```
self._stopwords = set(stopwords.words('english') + list(punctuation) + ['AT_USER','URL'])
```

```
def _processTweet(self, tweet):
    tweet = tweet.lower() # convert text to lower-case
    tweet = emoji.demojize(tweet) #converts emojis into text
    tweet = re.sub('((www\.[^\s]+)|(https?://[^\s]+))', 'URL', tweet) # remove URLs
    tweet = re.sub('@[^\s]+', 'AT_USER', tweet) # remove usernames
    tweet = re.sub(r'#([^\s]+)', r'\1', tweet) # remove the # in #hashtag
    tweet = word_tokenize(tweet) # remove repeated characters (helloooooooo into hello)
    return  [word for word in tweet if word not in self._stopwords]
```

For example: The following tweet could be present in the data set:

"@person1 retweeted @person2: Yippee with corn is the mooooostttt delicious!!!! 😛 #corn #yippe #yummy ..."

The pre-processor will result in the tweet looking like:

“AT_USER rt AT_USER Yippee with corn is the most delicious! :drooling_face: corn yippee yummy”

And finally, the tokenization will result in:

{“yippee”, “with”, “corn”, “most”, “delicious”, “:drooling_face:”, “corn”, “yippee”, “yummy”}

BUILDING VOCAB AND EXTRACTING FEATURES

A vocabulary is a list of all speech segments available for the model. This includes all the words in the Training set. This is just creating a list of all_words we have in the Training set, breaking it into word features. These word_features are a list of distinct words, each of which has its frequency as a key.

```
[119] import nltk

      def buildVocabulary(preprocessedTrainingData):
          all_words = []

          for (words, sentiment) in preprocessedTrainingData:
              all_words.extend(words)

          wordlist = nltk.FreqDist(all_words)
          word_features = wordlist.keys()

          return word_features
```

This function matches the tweets against the developed vocabulary. For every word in the word_features, we will have a key ‘contains word X’, where X is the word. Every key of those will have the value True/False, — True for ‘present’ and False for ‘absent’.

```
[120] def extract_features(tweet):
          tweet_words = set(tweet)
          features = {}
          for word in word_features:
              features['contains(%s)' % word] = (word in tweet_words)
          return features
```

Finally,

```
[148] word_features = buildVocabulary(preprocessedTrainingData)
      TrainingFeatures = nltk.classify.apply_features(extract_features, preprocessedTrainingData)
```

The NTLK built-in function apply_features does the actual feature extraction from our lists. Applying nltk.classify.apply_features provides word feature vectors which can be plugged into the Naives Bayes Classifier.

TRAINING THE CLASSIFIER

The word feature vectors produced are passed to the inbuilt NLTK Naives Bayes Classifier.

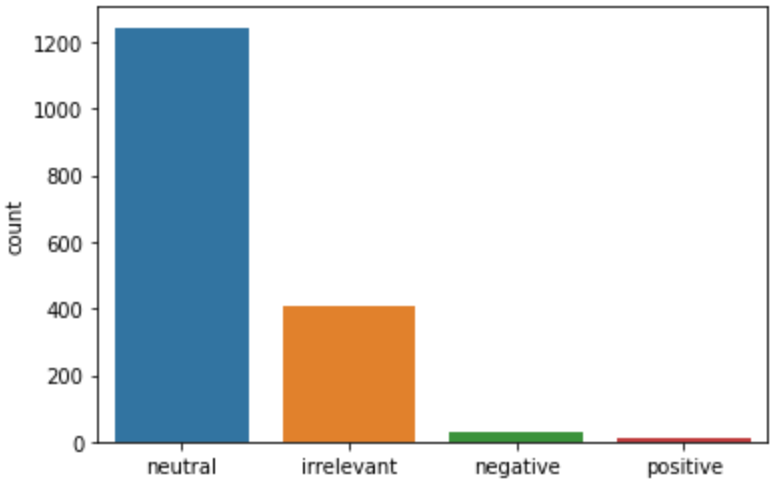
```
[149] NBayesClassifier = nltk.NaiveBayesClassifier.train(TrainingFeatures)
```

RESULTS

Running the classifier and calling the classify function with the test data set, gives the predicted labels.

```
[150] NBResultLabels = [NBayesClassifier.classify(extract_features(tweet[0])) for tweet in preprocessedTestingData]
```

Plotting the predicted sentiments

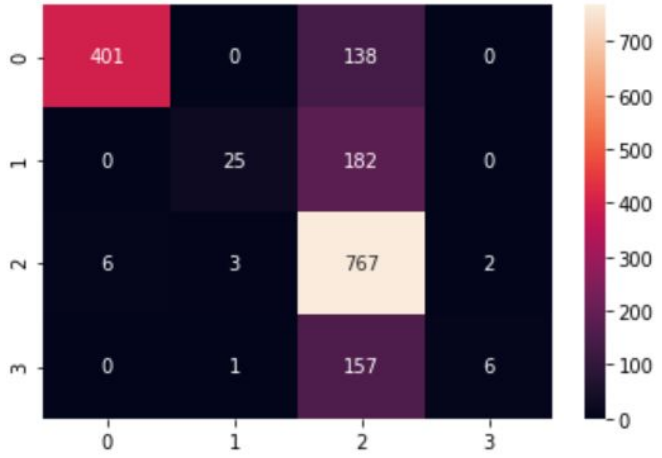


PERFORMANCE EVALUATION AND DISCUSSION

Confusion matrix is used to visualise the predictions. Accuracy, precision, recall and f1-score are used to evaluate the model using the predicted labels and the original labels.

```
[154] from sklearn.metrics import classification_report, confusion_matrix
      cm = confusion_matrix(tweetDataFileTest.Sentiment, NBResultLabels)
      sns.heatmap(cm, annot=True,fmt="d")
      print(classification_report(tweetDataFileTest.Sentiment, NBResultLabels))
```

	precision	recall	f1-score	support
irrelevant	0.99	0.74	0.85	539
negative	0.86	0.12	0.21	207
neutral	0.62	0.99	0.76	778
positive	0.75	0.04	0.07	164
accuracy			0.71	1688
macro avg	0.80	0.47	0.47	1688
weighted avg	0.78	0.71	0.65	1688



CONCLUSION

Sentiment analysis has been used to analyse the tweets to find out the sentiment of the twitter users. The Naive Bayes classifier has been evaluated with common metrics like confusion matrix. The hashtags and resulting(predicted) and true sentiments have been represented graphically.

FUTURE ENHANCEMENTS

Instead of trying to predict the sentiments of tweets from the dataset, real time tweets can be pulled from twitter with the help of tweepy package. Predicting the sentiment of such tweets and analysing them can provide us real time insights on the hashtags or trends in twitter.

REFERENCES

1. https://www.coursera.org/learn/twitter-sentiment-analysis/home/welcome
2. https://towardsdatascience.com/creating-the-twitter-sentiment-analysis-program-in-python-with-naive-bayes-classification-672e5589a7ed
3. https://www.earthdatascience.org/courses/use-data-open-source-python/intro-to-apis/analyze-tweet-sentiment-in-python/