**SENTIMENT ANALYSIS OF TWITTER**

Submitted in partial fulfillment of the requirements for the award of degree of

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE & ENGINEERING**



**Submitted to:**

**Miss ANUSHREE MAM**

**Submitted By:**

**MUDIT JAIN**

**18BCS6529**

**ARSHDEEP SINGH**

**18BCS3409**

**ANAND KUMAR SETH**

**18BCS6693**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**Chandigarh University, Gharuan**

**APRIL 2020**

**TABLE OF CONTENT**

**1. Implementation**

**2. Output validation and comparison**

**3. Team Work**

**1. IMPLEMENTATION**

We use different feature sets and machine learning classifiers to determine the best combination for sentiment analysis of twitter.

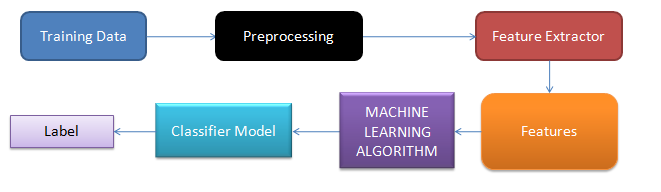
We investigated the following features –

Unigrams, bigrams, trigrams and negation detection.

We finally train our classifier using various machine-learning algorithms –

Naive Bayes, Decision Trees and Maximum Entropy.

We use a modularized approach with feature extractor and classification algorithm as two independent components.



**1.1 Datasets**

One of the major challenges in Sentiment Analysis of Twitter is to collect a labelled dataset. Researchers have made public the various datasets for training and testing classifiers.

Dataset used in this project is publicly available and can be found at: http://cs.stanford.edu/people/alecmgo/traini ng andtestdata.zip

It consists of 80,000 positive and 80,000 negatively classified tweets based on the emoticons used by the user.

**1.2 Pre Processing**

User-generated content on the web is seldom present in a form usable for learning. It becomes important to normalize the text by applying a series of pre-processing steps. We have applied an extensive set of pre-processing steps to decrease the size of the feature set to make it suitable for learning algorithms.

**1.2.1 Hashtags**

A hashtag is a word or an un-spaced phrase prefixed with the hash symbol (#). These are used to both naming subjects and phrases that are currently in trending topics. For example, #iPad, #news

Regular Expression: #(\w+)

Replace Expression: HASH\_\1

**1.2.2 URLs**

Users often share hyperlinks in their tweets. Twitter shortens them using its in-house URL shortening service, like http://t.co/FCWXoUd8 - such links also enables Twitter to alert users if the link leads out of its domain. From the point of view of text classification, a particular URL is not important. However, presence of a URL can be an important feature. Regular expression for detecting a URL is fairly complex because of different types of URLs that can be there, but because of Twitter’s shortening service, we can use a relatively simple regular expression.

Regular Expression: (http|https|ftp)://[a-zA-Z0-9\\./]+

Replace Expression: URL

**1.2.3 Punctuations**

Although not all Punctuations are important from the point of view of classification but some of these, like question mark, exclamation mark can also provide information about the sentiments of the text. We replace every word boundary by a list of relevant punctuations present at that point.

**1.2.4 Repeating Characters**

People often use repeating characters while using colloquial language, like "I’m in a hurrryyyyy", "We won, yaaayyyyy!" As our final pre-processing step, we replace characters repeating more than twice as two characters.

**1.3 Features**

A wide variety of features can be used to build a classifier for tweets. However, there's a lot of domain specific information present in tweets that can also be used for classifying them. We have experimented with two sets of features:

**1.3.1 Unigrams**

Unigrams are the simplest features that can be used for text classification. A Tweet can be represented by a multiset of words present in it. We, however, have used the presence of unigrams in a tweet as a feature set. Presence of a word is more important than how many times it is repeated. Pang et al. found that presence of unigrams yields better results than repetition. This also helps us to avoid having to scale the data, which can considerably decrease training time.

**1.3.2 N-grams**

N-gram refers to an n-long sequence of words. Probabilistic Language Models based on Unigrams, Bigrams and Trigrams can be successfully used to predict the next word given a current context of words. In the domain of sentiment analysis, the performance of N-grams is unclear. According to Pang et al., some researchers report that unigrams alone are better than bigrams for classification movie reviews, while some others report that bigrams and trigrams yield better product-review polarity classification

**1.3.3 Negation Handling**

The need negation detection in sentiment analysis can be illustrated by the difference in the meaning of the phrases, "This is good" vs. "This is not good" However, the negations occurring in natural language are seldom so simple. Handling the negation consists of two tasks – Detection of explicit negation cues and the scope of negation of these words.

**1.4 Classifier Model**

**1.4.1 Naive Bayes**

Naive Bayes classifier is the simplest and the fastest classifier. Many researchers [2], [4] claim to have gotten best results using this classifier.

For a given tweet, if we need to find the label for it, we find the probabilities of all the labels, given that feature and then select the label with maximum probability.

The results from training the Naive Bayes classifier are shown below in Figure 8 . The accuracy of Unigrams is the lowest at 79.67%. The accuracy increases if we also use Negation detection (81.66%) or higher order n-grams (86.68%). We see that if we use both Negation detection and higher order n-grams, the accuracy is marginally less than just using higher order n-grams (85.92%). We can also note that accuracies for double step classifier are lesser than those for corresponding single step.

**MODULES**

1. **PYMATLAB**

This package lets Python users interface and communicate with MATLAB from Python. Pymatlab makes it easier for users to integrate a project with a large MATLAB codebase into python scripts by using MATLAB scripts as a part of the python program.

The basic functionality of this package is to send data from Python to MATLAB’s workspace to be able to run Matlab function on the data. After processing you retrieve back data to python. This enables you to process data with Mathlab’s built in functions, toolboxes or Matlab-scripts. It is also possible to use MATLAB’s to generate plots or other graphics

1. **NUMPY**

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

* a powerful N-dimensional array object
* sophisticated (broadcasting) functions
* tools for integrating C/C++ and Fortran code
* useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

1. **SCIKIT-LEARN**

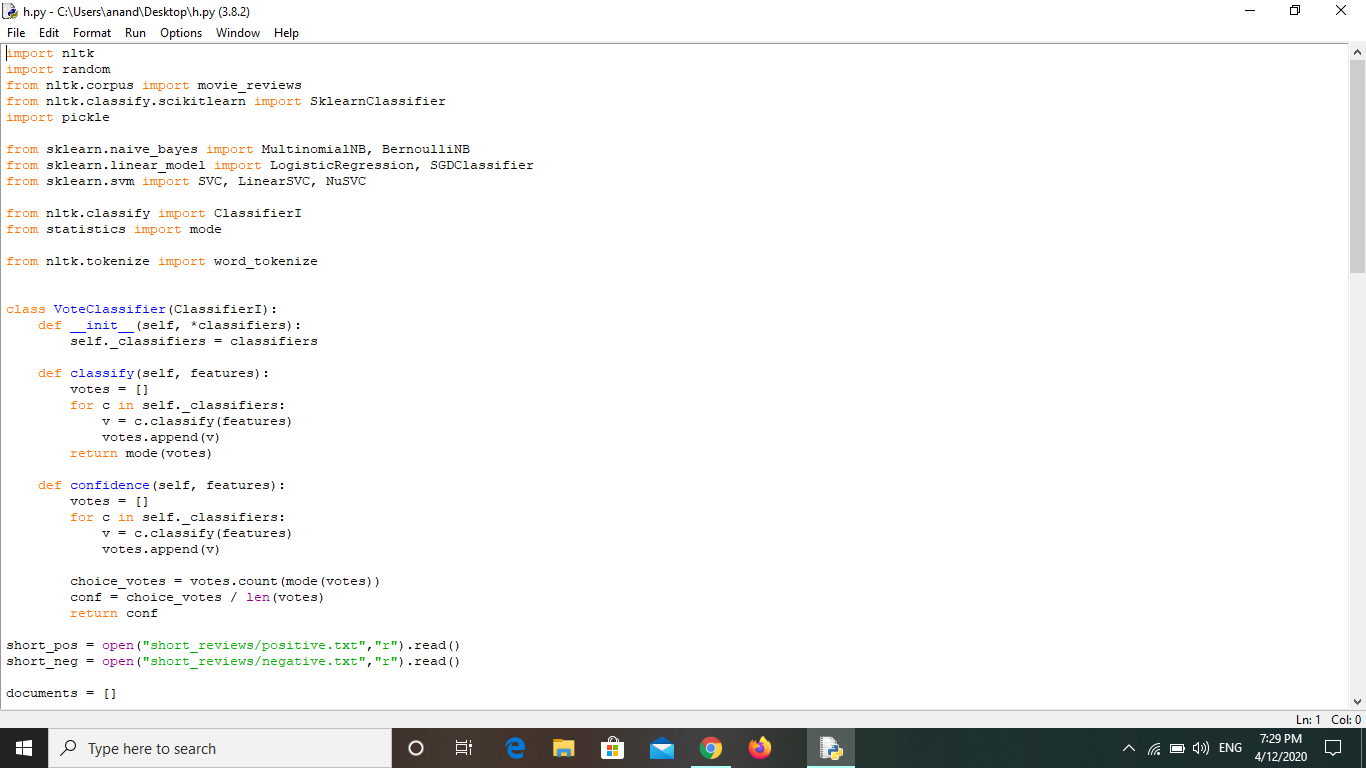
scikit-learn is a Python module for machine learning built on top of SciPy and is distributed under the 3-Clause BSD license.

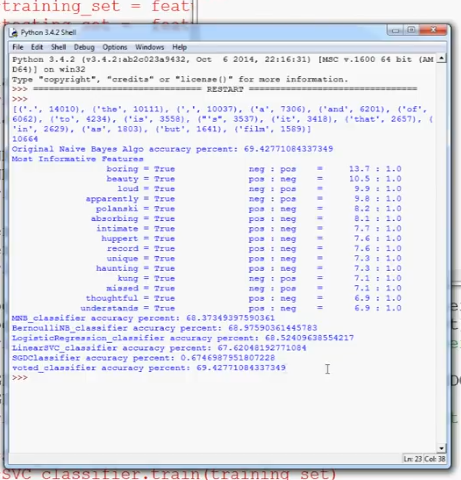
1. **TWEEPY**

An easy-to-use Python library for accessing the Twitter API.

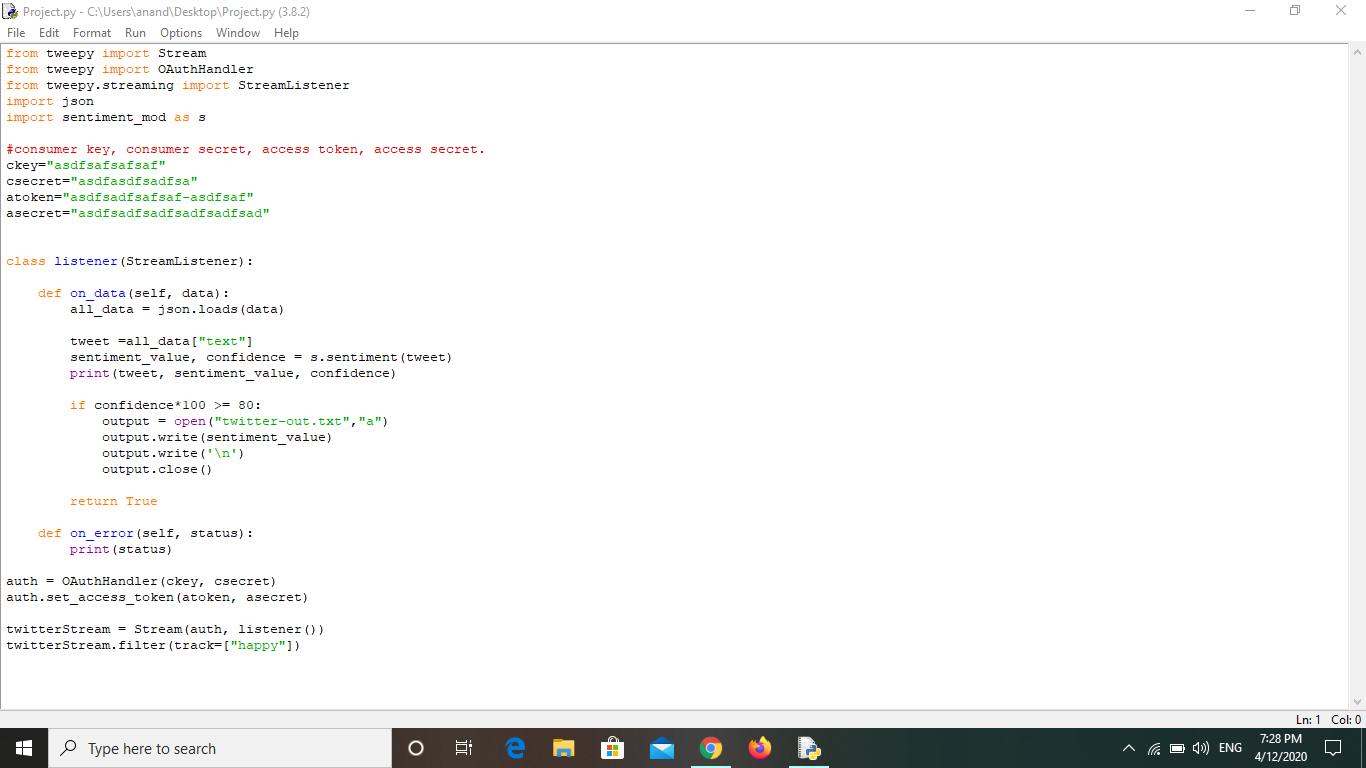
**2. OUTPUT VALIDATION AND COMPARISION**

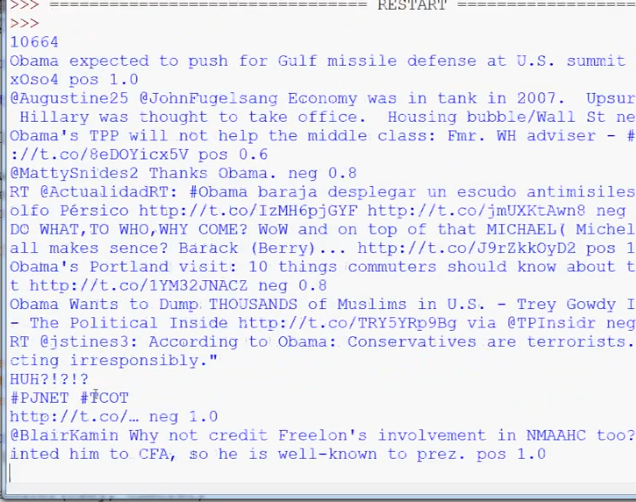
**Naive Bayes classifier**



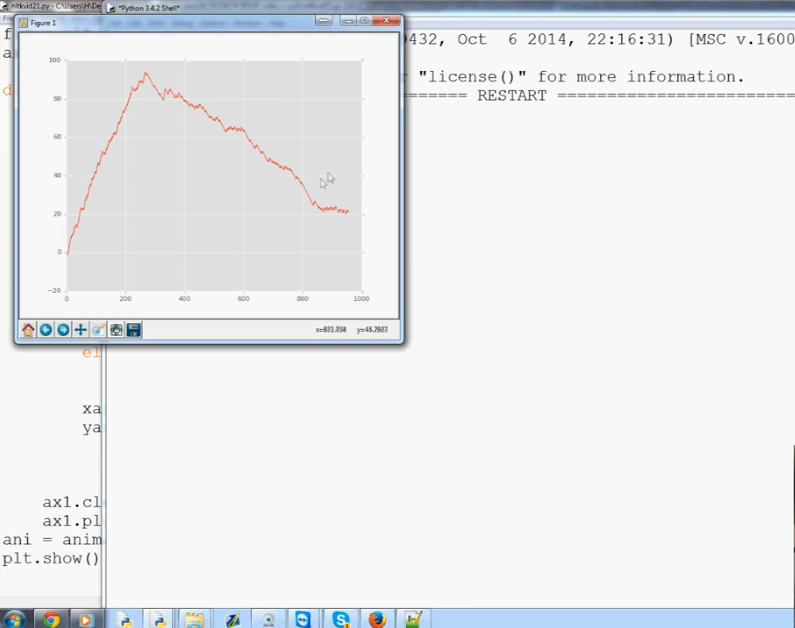


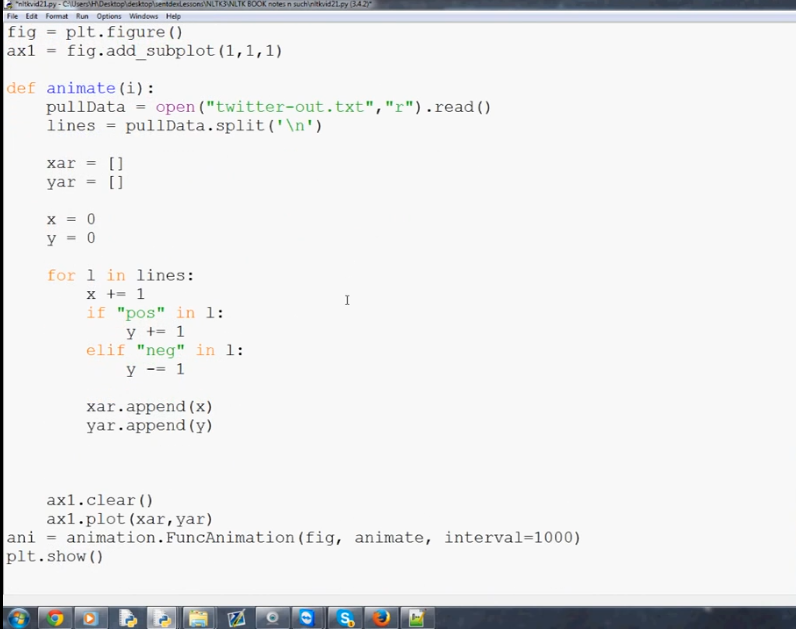
**TWITTER API**





**GRAPHICAL REPRESENTATION**





**3. TEAM WORK**

The project was divided into several parts that were to be completed by different members of the team.

Under the leadership of Anand seth, discussion about the project, schedule and deadlines were discussed. Team members separately studied overall idea of project and it’s implementation. Team discussed the process to complete the project. Arsh and anand made Entity Relationship graph and anand, mudit and arsh made SRS document. Mudit made the timeline. When the whole planning part was done, the coding part started. Mudit and arsh did dataset collection , formatting and preprocessing module coding. Anand completed the rest of the coding including application of feature extractor to application of machine learning algorithm on features till the prediction part.

We have learnt a lot of things while making the project which include the importance of leadership, planning , scheduling, deadlines. Also the distribution of work and responsibility is essential which prevents disputes from occurring in the later parts of the project.

Also we learnt the importance of proper communication between the team members and also between our mentor. We gained several good things like trust and commandrie.

This project has been beneficial for us in helping us improve in various areas.