**Project Documentation**

**Twitter Sentiment Analysis**

**Submitted By**

Himanshu Goyal

Ravindra Dhakar

Pushpa Sushma Aitha

Ishan Kotian

Abhinav Bhardwaj

**INDEX:**

**Contents Page**

1. Introduction 3

2. Problem Definition 5

3. Literature Survey 6

4. Theory 7

5. Dataset Used 10

6. Libraries 11

7. Implementation 12

8. Result 14

9. Conclusion 15

**Introduction: -**

In the past few years, there has been a huge growth in the use of micro blogging platforms such as Twitter. Spurred by that growth, companies and media organizations are increasingly seeking ways to mine Twitter for information about what people think and feel about their products and services. Companies such as Twitratr (twitrratr.com), tweetfeel (www.tweetfeel.com), and Social Mention (www.socialmention.com) are just a few who advertise Twitter sentiment analysis as one of their services.

While there has been a fair amount of research on how sentiments are expressed in genres such as online reviews and news articles, how sentiments are expressed given the informal language and message-length constraints of micro blogging has been much less studied. Features such as automatic part-of-speech tags and resources such as sentiment lexicons have proved useful for sentiment analysis in other domains, but will they also prove useful for sentiment analysis in Twitter? In this paper, we begin to investigate this question.

Sentiment analysis refers to the broad area of natural language processing which deals with the computational study of opinions, sentiments and emotions expressed in text. Sentiment Analysis (SA) or Opinion Mining (OM) aims at learning people’s opinions, attitudes and emotions towards an entity. The entity can represent individuals, events or topics. An immense amount of research has been performed in the area of sentiment analysis. But most of them focused on classifying formal and larger pieces of text data like reviews. With the wide popularity of social networking and micro blogging websites and an immense amount of data available from these resources, research projects on sentiment analysis have witnessed a gradual domain shift. The past few years have witnessed a huge growth in the use of micro blogging platforms. Popular micro blogging websites like Twitter have evolved to become a source of varied information. This diversity in the information owes to such micro blogs being elevated as platforms where people post real time messages about their opinions on a wide variety of topics, discuss current affairs and share their experience on products and services they use in daily life.

Twitter is an innovative microblogging service aired in 2006 with currently more than 550 million users 1. The user created status messages are termed tweets by this service. The public timeline of twitter service displays tweets of all users worldwide and is an extensive source of real-time information. The original concept behind microblogging was to provide personal status updates. But the current scenario surprisingly witnesses tweets covering everything under the world, ranging from current political affairs to personal experiences. Movie reviews, travel experiences, current events etc. add to the list. Tweets (and microblogs in general) are different from reviews in their basic structure. While reviews are characterized by formal text patterns and are summarized thoughts of authors, tweets are more casual and restricted to 140 characters of text. Tweets offer companies an additional avenue to gather feedback. Sentiment analysis to research products, movie reviews etc. aid customers in decision making before making a purchase or planning for a movie. Enterprises find this area useful to research public opinion of their company and products, or to analyze customer satisfaction. Organizations utilize this information to gather feedback about newly released products which supplements in improving further design. Different approaches which include machine learning (ML) techniques, sentiment lexicons, hybrid approaches etc. have been proved useful for sentiment analysis on formal texts.

**Problem Definition: -**

Sentiment analysis of in the domain of micro-blogging is a relatively new research topic so there is still a lot of room for further research in this area. Decent amount of related prior work has been done on sentiment analysis of user reviews, documents, web blogs/articles and general phrase level sentiment analysis. These differ from twitter mainly because of the limit of 140 characters per tweet which forces the user to express opinion compressed in very short text. The best results reached in sentiment classification use supervised learning techniques such as Naive Bayes and Support Vector Machines, but the manual labelling required for the supervised approach is very expensive. Some work has been done on unsupervised and semi-supervised approaches, and there is a lot of room of improvement. Various researchers testing new features and classification techniques often just compare their results to base-line performance. Hate Speech in the form of racism and sexism has become a nuisance on twitter and it is important to segregate these sorts of tweets from the rest.

**Literature Survey**: -

Sentiment analysis is a growing area of Natural Language Processing with research ranging from document level classification (Pang and Lee 2008) to learning the polarity of words and phrases (e.g., (Hatzivassiloglou and McKeown 1997; Esuli and Sebastiani 2006)). Given the character limitations on tweets, classifying the sentiment of Twitter messages is most similar to sentencelevel sentiment analysis (e.g., (Yu and Hatzivassiloglou 2003; Kim and Hovy 2004)); however, the informal and specialized language used in tweets, as well as the very nature of the micro blogging domain make Twitter sentiment analysis a very different task. It’s an open question how well the features and techniques used on more well-formed data will transfer to the micro blogging domain.

Researchers have also begun to investigate various ways of automatically collecting training data. Several researchers rely on emoticons for defining their training data (Pak and Paroubek 2010; Bifet and Frank 2010). (Barbosa and Feng 2010) exploit existing Twitter sentiment sites for collecting training data. (Davidov, Tsur, and Rappoport 2010) also use hashtags for creating training data, but they limit their experiments to sentiment/non-sentiment classification, rather than 3-way polarity classification, as we do. We use WEKA and apply the following Machine Learning algorithms for this second classification to arrive at the best result:

• K-Means Clustering

• Support Vector Machine

• Logistic Regression

• K Nearest Neighbours

• Naive Bayes

• Rule Based Classifiers

**Theory: -**

**Naive Bayes:-**

Many language processing tasks are tasks of classification, although luckily our classes are much easier to define than those of Borges. In this classification we present the naive Bayes algorithms classification, demonstrated on an important classification problem: text categorization, the task of classifying an entire text by assigning it a text categorization label drawn from some set of labels.

We focus on one common text categorization task, sentiment analysis, the ex-sentiment analysis traction of sentiment, the positive or negative orientation that a writer expresses toward some object. Are view of a movie, book, or product on the web expresses the author’s sentiment toward the product, while an editorial or political text expresses sentiment toward a candidate or political action. Automatically extracting consumer sentiment is important for marketing of any sort of product, while measuring public sentiment is important for politics and also for market prediction. The simplest version of sentiment analysis is a binary classification, and the words of the review provide excellent cues. Consider, for example, the following phrases extracted from positive and negative reviews of movies and restaurants. Words like great, richly, awesome, and pathetic, and awful and ridiculously are very informative cues:

+ ...zany characters and richly applied satire, and some great plot twists

− It was pathetic. The worst part about it was the boxing scenes...

+ ...awesome caramel sauce and sweet toasty almonds. I love this place!

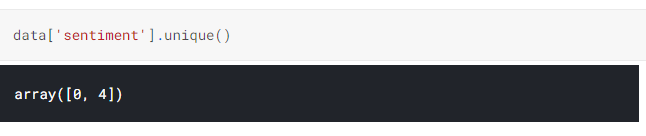
− ...awful pizza and ridiculously overpriced...

Naive Bayes is a probabilistic classifier, meaning that for a document d, out of all classes c∈C the classifier returns the class ˆ c which has the maximum posterior probability given the document. In Eq. 1 we use the hat notation to mean “our estimate of the correct class”

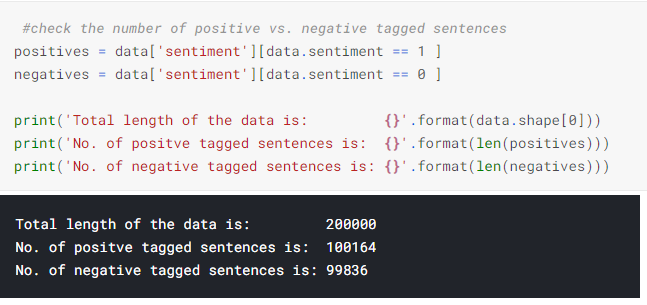
. c = argmax P(c|d) where c∈C

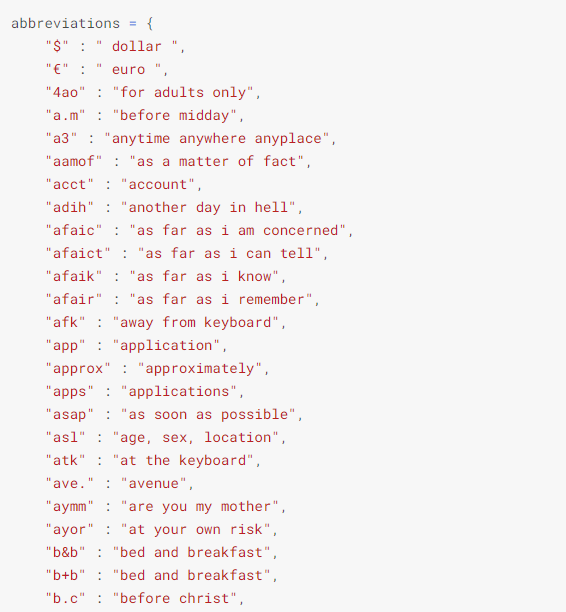
Naive Bayes is a generative model that make the bag of words assumption (position doesn’t matter) and the conditional in dependence assumption (words are conditionally independent of each other given the class).

Naive Bayes with binary features seems to work better for many text classification tasks.

But before segregating the data from dataset, we need to do some analysis operation like: checking null values, unique values, and remove some character like @,%,$ and many more to make our analysis and accuracy reliable.

To segregate the data, we need to initialize the sentiment of tweets into 1 and 0. For ex. Positive tweets refers as 1 and Negative tweets refers as 0.

Some abbreviations are used in our project, which we need to clarify. Because these are not easily understandable by computer, so we initialize all abbreviation in dictionary form to a variable abbreviation**.**



**Dataset Used**: -

The dataset used in this project is taken from Kaggle.com.

<https://www.kaggle.com/kazanova/sentiment140>

This is the direct link of dataset.

This is the sentiment140 dataset.  
It contains 1,600,000 tweets extracted using the twitter api. The tweets have been annotated (0 = negative, 2 = neutral, 4 = positive) and they can be used to detect sentiment.  
It contains the following 6 fields:

1. target: the polarity of the tweet (*0* = negative, *2* = neutral, *4* = positive)
2. ids: The id of the tweet ( *2087*)
3. date: the date of the tweet (*Sat May 16 23:58:44 UTC 2009*)
4. flag: The query (*lyx*). If there is no query, then this value is NO\_QUERY.
5. user: the user that tweeted (*robotickilldozr*)
6. text: the text of the tweet (*Lyx is cool*)

**Libraries**: -

* + 1. **Scikit-Learn: -** Scikit-Learn library used in this project. **Scikit-learn** is a Python module for machine learning built on top of SciPy and is distributed under the 3-Clause BSD license.

Scikit-learn provide a range of supervised and unsupervised learning algorithms via a consistent interface in Python.

The library is built upon the SciPy (Scientific Python) that must be installed before you can use scikit-learn. This stack that includes:

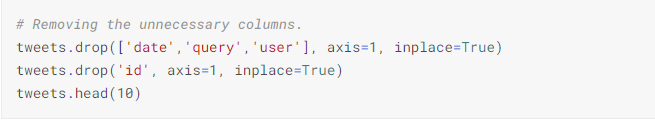
* **NumPy**: Base n-dimensional array package
* **SciPy**: Fundamental library for scientific computing
* **Matplotlib**: Comprehensive 2D/3D plotting
* **IPython**: Enhanced interactive console
* **Sympy**: Symbolic mathematics
* **Pandas**: Data structures and analysis
  + 1. **Pandas:** - Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license.
    2. **NumPy: -** NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
    3. **Matplotlib: -** Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.
    4. **Seaborn:** - Seaborn is a Python data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. Seaborn helps you explore and understand your data. Its plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots.

**Implementation: -**

To implement this project first we need to install all the libraries in the Jupyter Notebook or whatever open-source software we are using. After loading dataset, we need to do some analysis on dataset like: checking null values, unique values and then we will drop or remove those null values and unnecessary column from our dataset.



Now we need to remove the unnecessary columns:

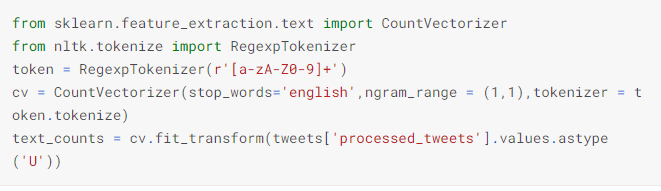


There are lot more words here in form of URL, Usernames, and Punctuation marks that need to be removed:

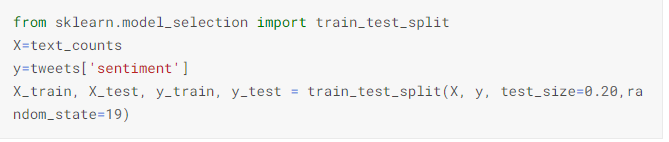


Scikit-learn's **CountVectorizer** is used to convert a collection of text documents to a vector of term/token counts. It also enables the ​pre-processing of text data prior to generating the vector representation.

A **RegexpTokenizer** splits a string into substrings using a regular expression.

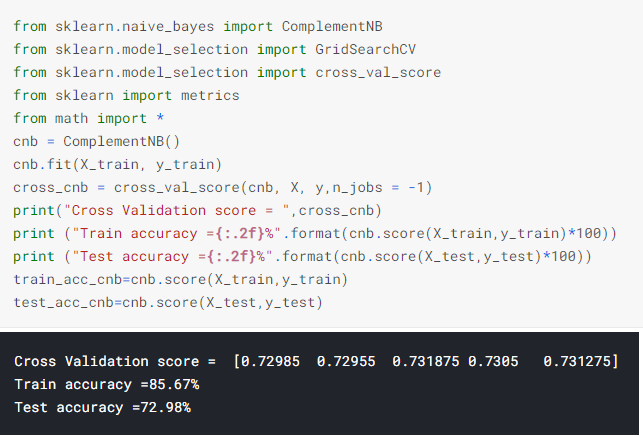


**train\_test\_split** is a function in Sklearn model selection for splitting data arrays into two subsets: for training data and for testing data.



**Result: -**

After performing all these operations on dataset, we reached to final result. We have achieved 85% accuracy, which is very good in this case because you need to perform lot of task on unstructured data and make the dataset easy to read.



**Conclusion**

This document presented a study of various types of abuse on Twitter. We analyzed 1.6 million tweets with embedded URLs and found that, during a period of high spam activity, many of them were spam or malicious in nature.

We examined the response rates for various types of Twitter spam and found that they widely varied, depending on the spam’s content and other factors. We therefore conclude that quoting a single response rate for Twitter spam is inadequate; it is important to quote response rates for each type of spam instead.

Nowadays, sentiment analysis or opinion mining is a hot topic in machine learning. We are still far to detect the sentiments of s corpus of texts very accurately because of the complexity in the English language and even more if we consider other languages such as Chinese.

In this project we tried to show the basic way of classifying tweets into positive or negative category using Naive Bayes as baseline and how language models are related to the Naïve Bayes and can produce better results. We could further improve our classifier by trying to extract more features from the tweets, trying different kinds of features.