­

**A PROJECT REPORT ON**

**SOCIAL MEDIA SENTIMENT ANALYSIS**

**SUBMITTED BY:**

Shivani Chauhan

(BTech (CSE) – 2nd Year student at GLA University, Mathura)

**UNDER THE GUIDANCE OF:**

Mr. Bipul Shahi

(CTO, Diginique TechLabs)

**JUNE, 2020**

**ABSTRACT**

Growth in the area of opinion mining and sentiment analysis has been rapid and aims to explore the opinions or text present on different platforms of social media through machine-learning techniques with sentiment, subjectivity analysis or polarity calculations. Social media contain huge amount of the sentiment data in the form of tweets, blogs, and updates on the status, posts, etc. In this project, the most popular micro blogging platform – Twitter, is used. Twitter sentiment analysis is an application of sentiment analysis on data from Twitter (tweets) to extract user’s opinions and sentiments. The main goal is to explore how text analysis techniques can be used to dig into some of the data in a series of posts focusing on different trends of tweet’s languages, tweet’s volumes on twitter. Experimental evaluations show that the logistic regression classifier is efficient and performs better in terms of accuracy.

**ACKNOWLEDGEMENT**

The satisfaction which accompanies the successful completion of any project is incomplete without the mention of the names of those who made it possible.

I would like to express my sincere gratitude to **Mr. Bipul Shahi**, CTO of Diginique TechLabs, whose role as project guide was invaluable for the project. I am extremely thankful for the keen interest he took in advising me, for the books, the reference materials and for providing the moral support to me. I convey my gratitude to all the teachers for providing me the technical skill that will always remain as my asset. Lastly, I thank my family and friends for their support and encouragement.

**INTRODUCTION**

In recent years, a huge number of people have been attracted to social-networking platforms like Facebook, Twitter and Instagram. Micro blogging websites are one of the most important sources of varied kind of information. This is due to the fact that every people post their opinions on a variety of topics, discusses current issues, complains and expresses positive sentiment for products they use in daily life.

Sentimental analysis is the process of deriving the quality information from the text. In other words, it is the process of deriving the structured data from unstructured data. This is used to measure opinions of the customer, feedback, product reviews.

Unstructured data not only refers to the tables, figures from the organization but also consists of information from the internet i.e. chats, E-mail, pdfs, word files, E-Commerce websites and social networking sites. On structured data, analytics operation can be easily performed and the result can be obtained easily. But in case of unstructured data from E-mail, Twitter etc., it is quite difficult to conclude the output because of various problems such as virtual noise effect and unspecific data. We look at one such popular micro blog called Twitter.

Data mining is another name for sentimental analysis. In many fields like business,

politics and public actions, determining the sentimental analysis is very important. The Sentiment analysis is part of natural language processing.

Natural language Processing is used for data analytics purpose, to extract meaningful information from lots of data. This is one of the methods to get information about current trend in the market of what people are thinking or talking on social media. There are so many practical applications present in the current world like in election which party is favourable or gaining popularity or a customer watching for reviews before actually buying something online. These are few of the applications which are getting harder to solve as size of data keeps on increasing.

The project would heavily rely on techniques of Natural Language Processing in extracting significant patterns and features from the large data set of tweets and on Machine Learning techniques for accurately classifying individual unlabelled data samples (tweets) according to whichever pattern model best describes them.

**PROBLEM STATEMENT**

Cyber bullying and hate speeches have been a menace for quite a long time. The objective of this project is to detect hate speech in tweets. For the sake of simplicity, we say a tweet contains hate speech if it has a racist or sexist sentiment associated with it. So, the task is to classify racist or sexist tweets from other tweets.

Formally, given a training sample of tweets and labels, where label '1' denotes the tweet is racist/sexist and label '0' denotes the tweet is not racist/sexist, our objective is to predict the labels on the test dataset.

With the given Kaggle dataset consisting of train.csv and test.csv files where we have 31962 labelled tweets and 17191 unlabelled tweets, we train and validate on the train.csv file and then test our best possible model on the test.csv file.

**BACKGROUND AND RELATED WORK**

**DATA PRE-PROCESSING:**

Raw tweets scraped from twitter generally result in a noisy dataset. This is due to the casual nature of people’s usage of social media. Tweets have certain special characteristics such as retweets, emoticons, user mentions, etc. which have to be suitably extracted. Therefore, raw twitter data has to be normalized to create a dataset which can be easily learned by various classifiers. An extensive number of pre-processing steps have been applied to standardize the dataset and reduce its size. Some general pre-processing on tweets are as follows:

* Removing twitter handles (@user)
* Removing punctuation, numbers, special characters
* Removing short words i.e. words with length<3
* Tokenization: It is the process of splitting a string of text into tokens.
* Stemming: It is a rule-based process of stripping the suffixes from a word.

**DATA VISUALIZATION:**

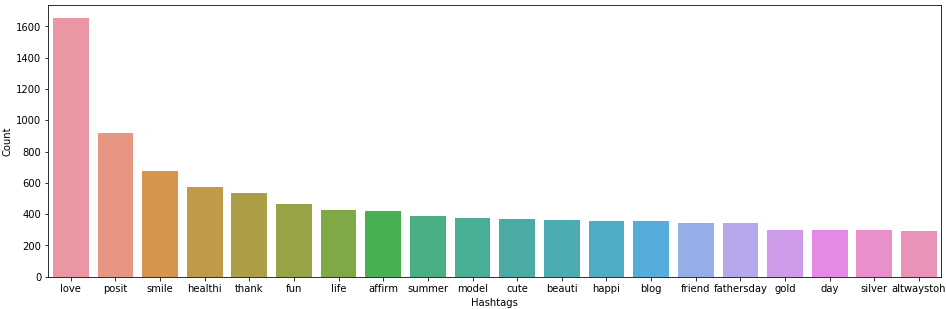
After pre-processing, we visualize the data with the following techniques:

### Word Cloud: A word cloud is a visualization where the most frequent words appear in large size and the less frequent words appear in smaller sizes.

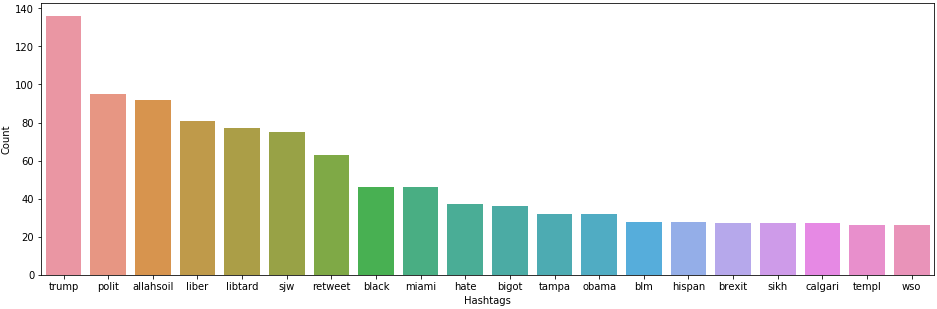
### 

* **Bar Plot:** A bar plot presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent.

For positive tweets:



For negative tweets:



**FEATURE EXTRACTION:**

* **Bag-of-Words:** Bag of Words is a method to extract features from text documents. These features can be used for training machine learning algorithms. It creates a vocabulary of all the unique words occurring in all the documents in the training set.

Consider a corpus (collection of texts) called C, of D documents {d1, d2,…,dD} and N unique tokens extracted out of the corpus C. The N tokens (words) will form a list, and the size of the bag-of-words matrix M will be given by D X N. Each row in the matrix M contains the frequency of tokens in document D(i).

For example, if you have 2 documents-

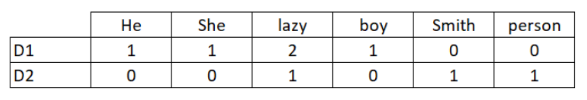
1. D1 - He is a lazy boy. She is also lazy.
2. D2 - Smith is a lazy person.

First, it creates a vocabulary using unique words from all the documents

#### [‘He’, ’She’, ’lazy’, 'boy’, 'Smith’, ’person’]

Here, D=2, N=6.

The matrix M of size 2 X 6 will be represented as:



The table depicts the training features containing term frequencies of each word in each document. This is called bag-of-words approach since the number of occurrence and not sequence or order of words matters in this approach.

* **TF-IDF:** It stands for Term Frequency-Inverse Document Frequency, and the tf-idf weight is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus.

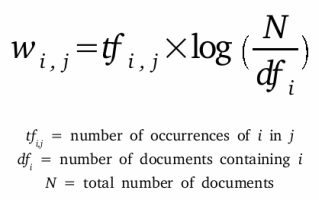
Typically, the tf-idf weight is composed by two terms: the first computes the normalized Term Frequency (TF), the number of times a word appears in a document, divided by the total number of words in that document; the second term is the Inverse Document Frequency (IDF), computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears.

**TF**: It measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length (the total number of terms in the document) as a way of normalization.

**TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document).**

**IDF**: It measures how important a term is. While computing TF, all terms are considered equally important. However, it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus, we need to weigh down the frequent terms while scale up the rare ones, by computing the following:

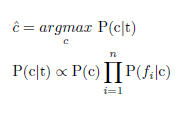
**IDF(t) = loge (Total number of documents / Number of documents with term t in it).**



**DESCRIPTION OF ALGORITHMS**

**NAÏVE BAYES:**

It is a simple model which can be used for text classification. In this model, the class is assigned to a tweet *t*, where



In the formula above, fi represents the *i*th feature of total *n* features. P(c) and P(fi|c) can be obtained through maximum likelihood estimates.

MultinomialNB from sklearn.naive\_bayes package of scikit-learn is used for Naive Bayes classification.

**SVM:**

SVM, also known as support vector machines, is a non-probabilistic binary linear classifier. For a training set of points (xi, yi) where *x* is the feature vector and *y* is the class, we want to find the maximum-margin hyperplane that divides the points with yi = 1 and yi = -1.

The equation of the hyperplane is as follows:

w ‧ x - b = 0

We want to maximize the margin, denoted by, as follows:



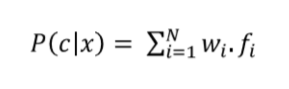
in order to separate the points well.

The SVM classifier available in sklearn is utilized to predict the sentiment.

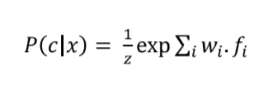
**LOGISTIC REGRESSION:**

It uses a Logistic function, for instance, a sigmoid function to estimate probabilities between positive or negative label and data features.

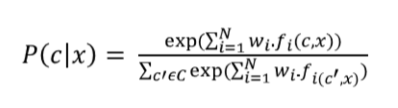
Logistic regression is a discriminative model which means computing P(y|x) by discriminating among different possible values of the class y based on the given input x. The equation is:



To generate a value of P(y|x) of an output that is in between value 0 and 1, the following expression is used:

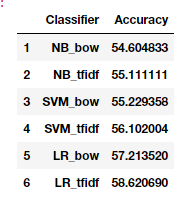


The final equation for computing the probability of y being of class c given x is:

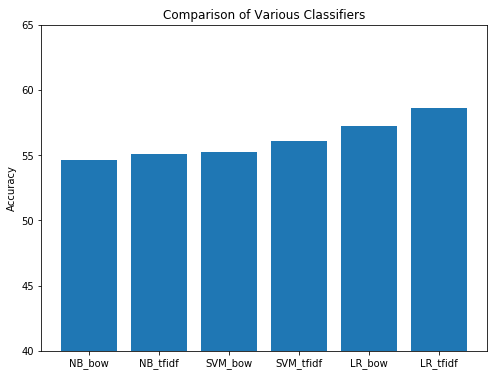


**ALGORITHM COMPARISON RESULT**

Based on the experiment results**,** the findings of the classifiers are tabulated based on its accuracy.



Logistic regression model for TF-IDF feature gives the best accuracy of 58.62% compared to all the classifiers. The following bar plot gives an understanding of the performance effect.



**REFERENCES**

* [https://www.kaggle.com](https://www.kaggle.com/)
* [https://www.analyticsvidhya.com](https://www.analyticsvidhya.com/)
* [https://www.medium.com](https://www.medium.com/)
* [https://scikit-learn.org](https://scikit-learn.org/stable/)
* <https://github.com/>

**CONCLUSION AND FUTURE WORK**

The project goal is to analyse the sentiments of texts which are extracted from the Twitter and determine its nature as positive or negative. After data pre-processing and visualization, various machine learning models were applied to predict the sentiment of tweets, such as Naïve Bayes, Support Vector Machine and Logistic Regression. Out of all the classifiers, accuracy of Logistic regression was highest. The test dataset was then predicted using Logistic Regression to obtain the final result.

The identified algorithms are implemented on a specified size of test data and the obtained results are subjective only to the used dataset. Thus, presenting scope for future research of validating the above-stated models with different sizes of the dataset and evaluating the results. Additionally, this can also pave a way for future research on analysing the impact of the size of the dataset on the performance of the machine learning algorithms.

We can improve and train our models to handle a range of sentiments. Tweets don’t always have positive or negative sentiment. At times they may have no sentiment i.e. neutral. Sentiment can also have gradations like the sentence – ‘This is good’, is positive but the sentence – ‘This is extraordinary’ is somewhat more positive than the first. We can, therefore, classify the sentiment in ranges, say, from -2 to +2.

During our pre-processing, we discard most of the symbols like commas, full-stops, and exclamation mark. These symbols may be helpful in assigning sentiment to a sentence.

Still, there is a huge scope of improvement of these existing sentiment analysis models.