



R & D Project Report

Academic Year- 2021-22

On

**Sentiment Analysis of tweets under supervised learning
algorithms**

Swattik Chakrabarty BT18GCS092

Aniket Chakraborty BT18GCS081

Shrey Bhardwaj BT18GCS092



DECLARATION BY STUDENT(S)

I/We hereby declare that the project report entitled Sentiment analysis of tweets which is being submitted for the partial fulfilment of the Degree of Bachelor of Technology, at NIIT University, Neemrana, is an authentic record of my/our original work under the guidance of Dr. Ratna Sanyal. Due acknowledgements have been given in the project report to all other related work used. This has previously not formed the basis for the award of any degree, diploma, associate/fellowship or any other similar title or recognition in NIIT University or elsewhere.

Place: NIIT University

Date: 16/05/2021

Swattik Chakrabarty BT18GCS085 B. tech CSE

Shrey Bhardwaj BT18GCS092 B. tech CSE

Aniket Chakraborty BT18GCS081 B. tech CSE



CERTIFICATE BY SUPERVISOR

This is to certify that the present R&D project entitle Sentiment analysis of tweets being submitted to NIIT University, Neemrana, in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology, in the area of CSE, embodies faithful record of original research carried out by Shrey Bhardwaj , Aniket Chakraborty, Swattik Chakrabarty They have worked under my guidance and supervision and that this work has not been submitted, in part or full, for any other degree or diploma of NIIT or any other University.

Place: NIIT University

Dr. Ratna Sanyal

Date: 16/05/2021

Acknowledgement

We would like to express our acknowledgement and appreciation to everyone who supported us throughout our research and made it possible.

The researcher on “Sentiment Analysis Of Tweets” has been completed with the support and invaluable assistance of our supervisor Prof. Ratna Sanyal . We would like to express our gratitude to her for giving us a finer grasp on our research topic by the reason of which made the completion of this research possible.

Last but just as importantly a special thanks to all panel members for improving our research with their comments and suggestions.

Introduction

In the past few years microblogging sites have become platforms for individuals or organizations across the world to express their opinions and sentiment in the form of tweets, status updates, blog posts, etc. These platforms have no political and economic restrictions. One of these platforms is Twitter. It has become increasingly popular with the global community. Users write short messages, also known as Tweets, where the word limit is up to 140 characters. Users give their personal opinions on many subjects, discuss current topics and write about ongoing live events in the world through tweets. According to the latest report there are 192 million active twitter users.

Sentiment analysis alludes to the utilization of natural language processing (NLP), text mining and machine learning techniques to determine the intentions of a speaker. The basic idea of sentiment investigation is to detect the polarity of text documents, tweets etc. and to classify them on this premise. Sentiment polarity is categorized as positive, negative or neutral.

Problem Statement & Objective

In the past few years it has been observed that different researchers use different algorithms in supervised learning approaches to analyse sentiments in tweets without giving preference to any single algorithm. This leads to confusion among researchers as it is not apparent which algorithm is best for classification of polarity of any given data.

Our main objective is to study the existing sentiment analysis methods of Twitter data (tweets). In this paper we are trying to determine which of the three widely accepted supervised learning algorithms, i.e, SVM, Maximum Entropy and Naive Bayes, is the most efficient, in terms of accuracy and resource consumption.

Along with our main objective we will also determine the effect of emoticons in the analysis of tweets and its effect in the accuracy of the prediction model.

We will also consider the positive or negative impact of some specific words/adjectives, and try to determine the magnitude of the sentiment reflected.

Literature Survey

Sentiment analysis is a field where many researchers are doing research, because of different challenges and multiple approaches. In general there are four kinds of approaches such as supervised ML, unsupervised ML, Text mining and deep learning approach and in each approach there are numerous algorithms through which you can calculate the accuracy, precision and the recall of your data set. There was also a research paper[1] written by Azzouza, Nouredine In 2015 from Russia which was based on Unsupervised Machine learning technique by using POS features to find the accuracy and it got only 55.5% accuracy. A model[2] developed by Andrey and Anton to extract the polarity from twitter data. The features extracted were words containing n-grams and emoticons. The investigation indicated that the Naive Bayes was overshadowed by the performance of the SVM with a precision accuracy of 81% and recall accuracy of 74%. Back in 2009[3] Go and L. Huang introduced a method to classify the sentiments of tweets by utilizing the twitter API to extract tweets that had emoticons and then they were further used to classify them as positive or negative. Numerous classifiers like Naive Bayes, MaxEnt and SVM were employed to classify those tweets. The best result obtained by them was by using MaxEnt classifier in conjunction with unigram and bigram which achieved an accuracy of 83% compared to Naive Bayes which got the classification accuracy of 82.7%. After a few years in 2017 Emil R. Kaburuan who is from Bina Nusantara University, Jakarta, Indonesia, used text mining techniques such as Decision Tree, K-NN, and Naïve Bayes Classifier for sentiment analysis of twitter data and he got accuracy between 75%-78% in each of the three classifiers.

One another group of researchers have done the same research. The three classifiers predict the labels in the dataset. The results show the Accuracy of the Decision Tree,

K-NN, and Naïve Bayes of 80%, 78%, and 77%. Results for Precision of Decision Tree, K-NN, and Naïve Bayes amounted to 79.96%, 85.67%, and 88.50%. The results also show that Recall from Decision Tree, K-NN, and Naïve Bayes is 84%, 70%, and 64%. So it can be concluded that the Naïve Bayes classifier is the best classifier for use with social media datasets because it provides more accurate and precise predictions.

[5] At the moment the research is still going on in supervised learning and deep learning approaches. There is one more technique called emotion mining (comes under deep learning) which helps to find the emotion behind the sentence. This model is developed by psychologist two of these are popular in computer recognition of emotions.i) Eckman's model that differentiates the six basic emotions: anger, disgust, fear, happiness, sadness and surprise ii) The other is Plutchik's wheel of emotions with eight basic emotions: anger, anticipation, joy, trust, fear, surprise, sadness and disgust."

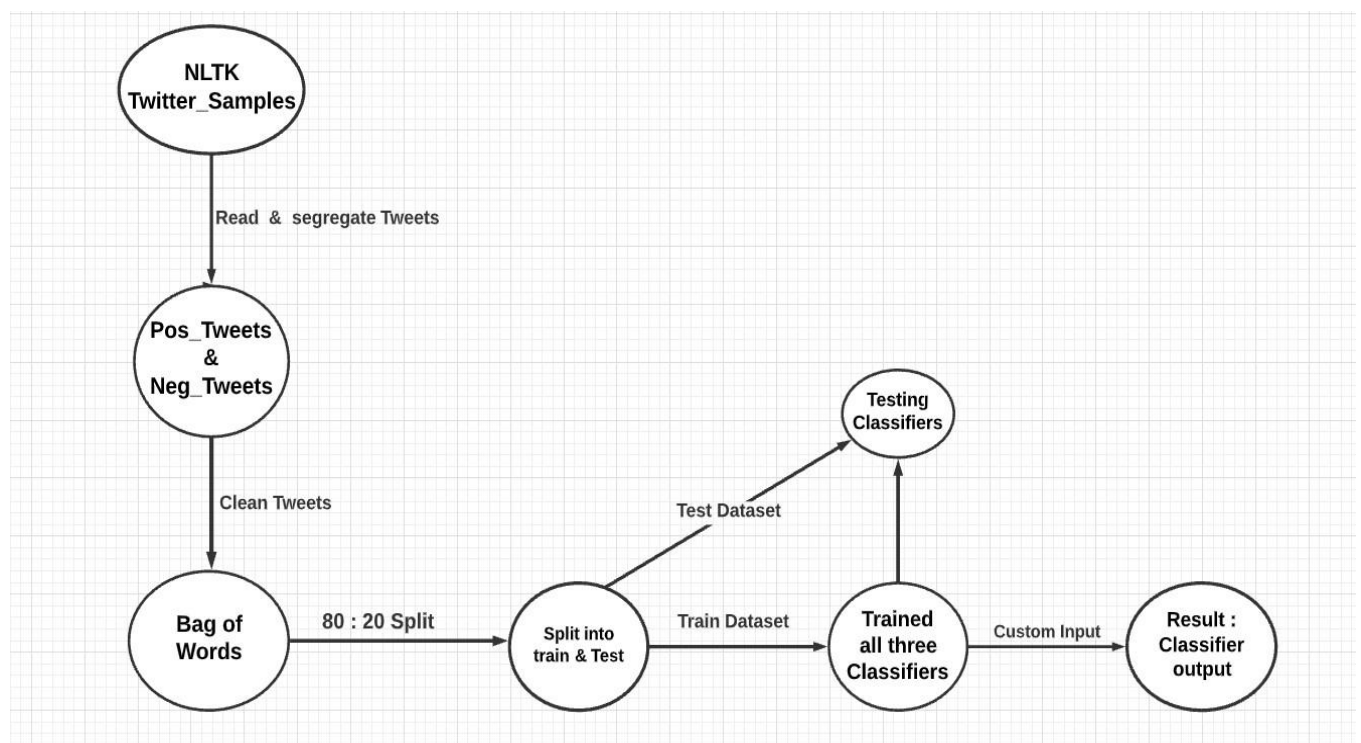
Challenges

[6] Social network users are writing more freely and often do not care about writing proper words. Sometimes they deliberately write the words incorrectly to point out the thought (for example, they write the word baaaad, whose meanings are very bad) or they replace the letters by number (2day instead of today). Depending on the context, one and the same word can be both positive and negative and also neutral. For example, word long is basically neutral (long hair) but becomes negative in the example of long queue and positive in long battery. The word context is often far-off from the word it depends on in the sentence ,so it is difficult to understand when looking at the phrases in which words stand next to each other.

Proposed methodology to specific gap

As we see from the above approaches, in the text mining approach and unsupervised approach the accuracy is 55 % and 75 % which currently seems to have quite a room for improvement, right now we are aiming to use supervised learning techniques to achieve a higher level of accuracy.

Workflow :



NLTK Twitter_Sample : This is a pre-existing data set inside nltk module with pre classified positive and negative tweets. There are total 10,000 datasets out of which 5000 are positive and 5000 are negative

POS_tweets & NEG_tweets : Tweets are segregated in positive and negative with predefined functions

Bag of words : After the tweets have been clean and stopwords are removed ,they are appended to python dictionary so that they can qualify as a viable feature set



- Removing URLs : Hyperlinks in tweets do not play much role in sentiment classification hence they have been removed.
- Removing Unicode Characters: Unicode characters are used to represent emoticons and many other complex symbols. So to avoid complexity for preprocessing we will remove these characters.
- Removing newline characters : These character are just to indicate a newline represented by “\n”, so it is not required for sentiment analysis

Split into train and test : Dataset was split in 80 : 20 ratio with training set having 8000 tweets and the test set was left with 2000 tweets.

Trained all three classifier : A training data set of 4000 positive and 4000 negative tweets have been used to train the three classifiers i.e Naive bayes ,Maximum Entropy and Support Vector Machine in order to categorize them because different models provide different accuracy and we choose the model with higher accuracy.

Testing classifier : A testing data set of 1000 positive and 1000 negative tweets have been used to train the three classifiers i.e Naive bayes ,Maximum Entropy and Support Vector Machine

Classifier output : The result of the above step is classifying tweets which may belong to two categories as mentioned above as Positive & Negative. At last we also calculate the accuracy of all three ML models. We get the predicted sentiment as the output along with the magnitude of the sentiment that is computed based on the words present in the tweets.

We are briefly explaining 3 methodologies below that we are going to implement in our ML model.

Naïve Bayes

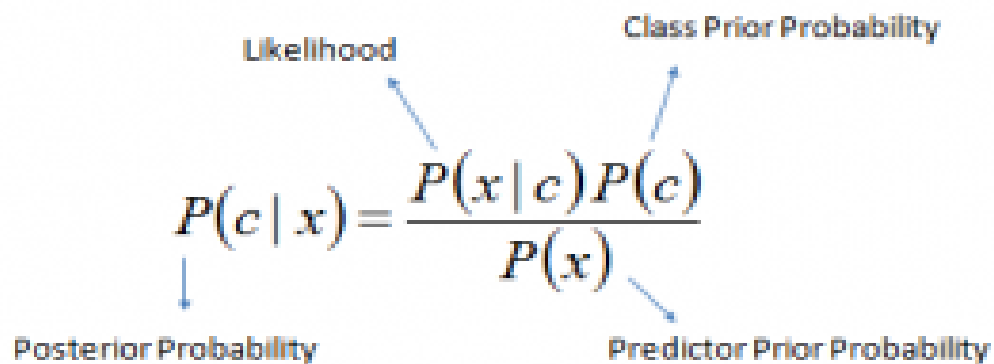
This is a grouping strategy that depends on Bayes' Theorem with strong autonomous presumptions between the features. A Naive Bayes classifier anticipates that the closeness of a particular (component) in a class is detached to the closeness of some different components. For example, a natural organic product may be viewed as an apple if its tone is red, its shape is round and it estimates roughly three crawls in expansiveness. Whether or not these highlights are reliant upon each other or upon the presence of different highlights, a Naïve Bayes classifier would consider these properties free because of the probability that this regular natural product is an apple. The Naive Bayes is known to out-perform even incredibly current strategies..

Algorithm of Naive Bayes

For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability $P(c|x)$ from $P(c)$, $P(x)$ and $P(x|c)$. Look at the equation below:



The diagram shows the equation $P(c | x) = \frac{P(x | c) P(c)}{P(x)}$ with four labels and arrows pointing to specific parts: 'Likelihood' points to $P(x | c)$, 'Class Prior Probability' points to $P(c)$, 'Posterior Probability' points to $P(c | x)$, and 'Predictor Prior Probability' points to $P(x)$.

$$P(c | x) = \frac{P(x | c) P(c)}{P(x)}$$

$$P(c | X) = P(x_1 | c) \times P(x_2 | c) \times \dots \times P(x_n | c) \times P(c)$$

Above,

- $P(c|x)$ is the posterior probability of *class* (c , *target*) given *predictor* (x , *attributes*).
- $P(c)$ is the prior probability of *class*.
- $P(x|c)$ is the likelihood which is the probability of a predictor given *class*.
- $P(x)$ is the prior probability of the predictor.

Algorithm inside our code:

```
NBclassifier = NaiveBayesClassifier.train(train_set)
```

Maximum Entropy

The Maximum Entropy (MaxEnt) classifier gauges the conditional distribution of a class denoted a given record by using a sort of exponential family with one load for each limitation. The model with greatest entropy is the one in the parametric family that amplifies the probability. Mathematical techniques, for example, iterative scaling and quasi Newton optimisation are normally utilised to tackle the advancement issue.

Algorithm of Maximum Entropy

Begin

.t = 0

Initialize population P(t) /* popsize = | P | */

For i = 1 to popsize

 Compute fitness P(t)

t=t+1

If termination criterion achieved go to step 10

Select (P)

crossover(P)

Mutate (P)

Go to step 3.

Output best chromosome and stop

end

Algorithm inside our code:

```
MaxEntClassifier = MaxentClassifier.train(
```

```
train_set, 'GIS', trace=0, encoding=None, labels=None, gaussian_prior_sigma=0, max_iter=1)
```

Support Vector Machine

SVM explores data, describes decision cutoff points and uses the parts for the count, which are acted in the information space. The vital information is presented in two arrangements of vectors, each of size m . At this point, each datum (expressed as a vector) is ordered into a class. Next, the machine identifies the boundary between the two classes that is far from any place in the training samples. The separate characterizes the classification edge, expanding the edge lessens ambivalent choices.

Algorithm of SVM

Candidate SV = {closest pair from opposite classes}

While there are violating points do

 Find a violator

 Candidate SV = \cup Candidate SV

 S

 Violator

 If any $\alpha_p < 0$ due to addition of c to s then

 CandidateSV = CandidateSV / P

 Repeat till all such points are pruned

 End if

End while

Algorithm inside our code:

```
SVCclassifier = SklearnClassifier(LinearSVC(), sparse=False)
SVCclassifier.train(train_set)
```

Result and Analysis

Along with the bag of words analysis we are also considering the positive or negative impact of some specific words/adjectives, we can determine the magnitude of the sentiment reflected. We also found out that emoticons plays an important role in determining the sentiment of a tweet and result in higher accuracy of the model if included in the bag of words.

Initially the difference in accuracy among different classifiers is negligible :

- Naive Bayes: 73.05%
- Maximum Entropy: 73.50%
- SVM : 71.05%

However, after taking emoticons into account, the overall accuracy of all three classifiers seem to improve drastically;

- Naive Bayes: 99.05%
- Maximum Entropy: 95.85%
- SVM : 99.35%

References

[1][N. Azzouza](#), H. Boumedien, [K. Akli-Astouati](#) and H. Boumedien, "A real-time Twitter sentiment analysis using an unsupervised method", [WIMS '17: Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics](#), page 1-10, 2017

[2] A. Barhan and A. Shakhomirov, "Methods for Sentiment Analysis of twitter messages," in the 12th Conference of FRUCT Association, 2012.

[3] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," CS224N Project Report, Stanford, vol. 1, no. 2009, p. 12, 2009.

[4] Achmad Bayhaqy, Nusa Mandiri, Sfenrianto Sfenrianto, Kaman Nainggolan, "Sentiment Analysis about E-Commerce from Tweets Using Decision Tree, K-Nearest Neighbor, and Naïve Bayes" ., Conference: 2018 International Conference on Orange Technologies (ICOT)2018

[5] Dalibor Bužić, "Sentiment Analysis of tweets" ,Central European Conference on Information and Intelligent Systems,page no. 215, 2019

[6] Abdullah Alsaeedi, Mohammad Zubair Khan, "A Study on Sentiment Analysis Techniques of Twitter Data " , International Journal of Advanced Computer Science and Applications, Vol. 10, No. 2, 2019