Twitter Sentiment Analysis: Case Study on Naïve Bayes algorithm

LaTeX template adapted from:
European Conference on Artificial Intelligence

Zarin Tasnim[1]

Other group members:
Anas Ali[2], **Muhammad Khan**[3], **Ibrahim Mullaj**[4]

Abstract. Over the years, Micro-blogging became a common networking method to attract audiences. Twitter is micro-blogging platform which allows users to share opinions and hence, for sentiment analysis Twitter has become a rich source of data base. This paper aims to analyse unstructured twitter datasets to undertake a stepwise methodology to analyse tweet's sentiment. The algorithm uses a Naïve Bayes sentiment classifier that can evaluate tweets into positive and negative sentiments. The code involves reading tweets from a dataset followed by pre-processing through Natural Language processing (NLP). Attempts have been made to structure the collected twitter dataset to find accurate correlation of negative/positive sentiment. In the end, a ROC curve has been used to find the accuracy of classifying method to interpret probability forecast.

1 Introduction

Today, micro blogging has become a highly discussed platform. Twitter users express their views on various aspects of everyday life. Using multi-label classifier, Twitter supports brief description of ideas through short mining and only accepts tweet that are no more than 140 characters. It allows valuable and well-timed statement of information. The social link on twitter is asymmetric and can be conceptualised towards specific product, brand, news, movies, politics and hence the tweets can be analysed to obtain real time influence around the globe. According to Oxford, Sentiment Analysis is the method to decide whether the sentiment concerning given issue or product is positive or negative polarity.

Sentiment analysis with Naïve Bayes classifier involves tokenisation, and machine level processing. The methods of Naive Bayes are a series of supervised machine learning algorithms based on the application of Bayes' theorem. A distinct characteristic of Bayes theorem is that it makes conditional independence assumption, between each pair of features of given class variable value. There are different classifiers of Naïve Bayes which are as follows:

1. **Bernoulli Naïve Bayes:** This classifier assumes that all characteristics are binary, such that only two values are taken. The definition of 0s may be "word does not occur in the document" and 1s may be "word occurs in the document."
2. **Multinomial Naïve Bayes:** This classifier is used for discrete data type. It represents words in terms of their events (frequency count) and also uses the frequency of occurrence of a given event.
3. **Gaussian Naïve Bayes:** It uses mean and normal distribution to summarise a continuous dataset.

We used the classification of Multinomial Naïve Bayes in this paper to construct a sentiment classifier. It is a supervised machine learning classifier which uses probabilistic algorithms. The probabilistic algorithm depends on Bayes's Theorem to predict the category of a text. This theorem provides a way of calculating highest probability. The classifier algorithm finds the highest likelihood value in the most suitable category to define test results. In this study, twitter dataset are used as test data for Bayes theorem calculation.

$$\Pr(A|B) = \frac{\Pr(B|A) * \Pr(A)}{\Pr(B)} \tag{1}$$

- Pr(A): the probability of A being true. This is called prior probability.
- Pr(B): the probability of B being true. This is called marginalisation probability.
- Pr(A given B): the probability of A given B. This is called posterior probability.
- Pr(B given A): the probability of B given A is true. This is called likelihood probability.

The Naive Bayes classifier claims that the influence of a certain function in a class is separate from other features. It is called Naive Bayes because it simplifies the equations of the probabilities to make their calculations tractable for each class. Sentiment analysis is widely used to help companies to analyse an opinion about the brand and businesses. It helps to conclude to a decision to modify business or product altogether in order to avoid any losses.

2 Background

This research paper [4] presents a method using Twitter API to collect corpus of tweets. Emoticons were used in the collected corpus

[1] School of Computing and Mathematical Sciences, University of Greenwich, London SE10 9LS, UK, email: zt2082j@gre.ac.uk
[2] School of Computing and Mathematical Sciences, University of Greenwich, London SE10 9LS, UK, email: aa8664f@gre.ac.uk
[3] School of Computing and Mathematical Sciences, University of Greenwich, London SE10 9LS, UK, email: mk25901@gre.ac.uk
[4] School of Computing and Mathematical Sciences, University of Greenwich, London SE10 9LS, UK, email: im2250r@gre.ac.uk

data i.e. emoticons such as':)":-)' as positive and emoticons such as :(" ":-(" as negative. The used corpus data to train the classifier is based on Naïve Bayes algorithm which used N-gram and POS-tags. They discussed about the statistical linguistic of the collected data and the efficiency of method used to acquire sentiment data. Different experiments were shown with unigrams, bigrams, and trigrams. In terms of future work, they writers aim to work with multilingual corpus of Twitter data and compare with different languages.

The publication [1] examined sentiment analysis on Twitter data. The tweets were classified into positive, negative and neutral sentiment. Two methods were discussed in the paper to do so, a binary classification and a 3 point based classification. The paper introduced POS-specific polarity features. Experiments performed showed Twitter-specific features i.e. emoticons, hashtags, and acronyms, add value to the classifier but to only a limited extent. Furthermore, a tree kernel was designed to represent different tweet features. In future, the paper aims for linguistic analysis.

The purpose of this paper [6] is to include an automated method to predict the tweets' feelings using hadoop. This paper deals with the challenges of real time tweets and provides time based analysis. The writer also focuses on opinion summarisation and sentiment analysing features. There are three types of sentiment analysis: Text, aspect, and sentence Level. The text level analyses works by scanning through whole document and predicting it as negative or positive. Sentence level analysis works by providing sentences with polarity score.

This paper [2] covers various topics like micro-blogging and Natural Language Processing (NLP). The paper attempts to describe the tweets of an ordinary citizen about the stock price movements of a global corporation called Samsung Electronics Ltd. An algorithm to analyse sentiments was implemented with data cleaning and extraction. Different Flowcharts and polarity score has been vastly discussed using tweets related to Samsung products.

The paper proposed [5] two different approaches to sentiment analysis: Supervised and Unsupervised approach. In this paper R software and Twitter API have been discussed. R software can analyse sentiment through data streaming using Twitter API. The methods involved data collection, cleaning followed by a lexicon-based approach. Dictionaries were also used in the proposed algorithm. In future, the writer aims to work with further machine learning approach for sentiment analysis.

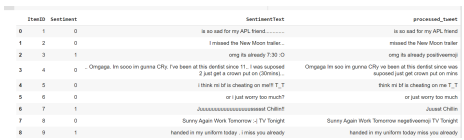
This paper [3] aims to estimate disturbance during public activities, by using sentiment analysis of Twitter content. A sentiment analysis algorithm based on lexicons that assigns a meaning within a message from a range of -100 to 100 has been addressed. A case study revealed a contrast between the sentiment of the English Defence League (EDL) and the degree of disturbance at an incident previous to the EDL demonstration. In addition, by using movie ratings, mood analysis and consistency of long sentences is tested. Automatic and manual approaches to dictionaries have been explored alongside lexicon-based research.

3 Experiments and results

This experiment used streamed twitter data processed in a csv file called "train.csv" taken from Github. The programming part of the experiment required to import all libraries corresponding to the algorithm used. The code reads the csv file line by line in an unstructured form, with the help of panda module, at first the unstructured data has been structured into 3 columns which includes item Id, sentiment of text and the text itself.

For the Naïve Bayes algorithm itself, after reading the tweet dataset, the dataset requires adjusting and fine tuning of each tweet as shown in figure 1. We used the following adjustments for each tweet:

1. Remove Usernames, single characters
2. Remove @mentions, hashtags, hyperlinks, more than one spaces
3. Remove punctuation, numbers
4. Emoticon analysis



	ItemID	SentIment	SentimentText	processed_tweet
0	1	0	is so sad for my APL friend..........	is so sad for my APL friend
1	2	0	I missed the New Moon trailer...	missed the New Moon trailer
2	3	1	omg its already 7:30 :O	omg its already positiveemoji
3	4	0	. Omgaga. Im sooo im gunna CRy. I've been at this dentist since 11.. I was supposed 2 just get a crown put on (30mins)...	Omgaga Im soo im gunna CRy. ve been at this dentist since was supposed just get crown put on mins
4	5	0	i think mi bf is cheating on me!!! T_T	think mi bf is cheating on me T_T
5	6	0	or i just worry too much?	or just worry too much
6	7	1	Juuuuuuuuuuuuuuuuussst Chillin!!	Juuust Chillin
7	8	0	Sunny Again Work Tomorrow :-| TV Tonight	Sunny Again Work Tomorrow negativeemoji TV Tonight
8	9	1	handed in my uniform today . i miss you already	handed in my uniform today miss you already

Figure 1. Comparison between Unprocessed v. Processed Tweets

Pre-processing dataset involves **Tokenization**; it breaks down text into individual words.

The next step is to remove **Stopwords** like "but," "or," etc. as machines don't understand the sentiment behind these words.

The next optional step used is **Normalizing** that is done by condensing all forms of a word into a single representation of that word. We used Lemmatization to convert word to its actual form i.e. 'dealt', 'dealing' to deal.

The last step is **Feature extraction or Vectorization**. Vectorization maps each word to a numerical representation for machines to understand. In the code, Count Vectorization was implemented to convert dataset to a matrix of the counts of occurrences of each word in the dataset and then assigns numerical representation of 0 and 1.

Finally multinomial classifier is used to test the algorithm. The dataset is partitioned into two sets; training and test data. 80 percent of dataset is used to train the algorithm model. Remaining, 20 percent of dataset has been used as test data that helps to evaluate the model's performance in terms of prediction accuracy.

After training the algorithm, we got accuracy with Naïve Bayes of approximately 0.7479. A Confusion matrix of 2 x 2 matrixes has been used to evaluate efficiency of the Multinomial Naïve Byes algorithm. The target variable contains a Positive or Negative value

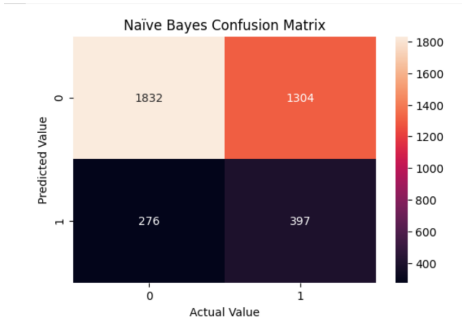


Figure 2. Confusion Matrix

represented in column and row. The column represents actual target value and row represents predicted target value.

The uncertainties of the Confusion matrix in figure 2 are as follow:

- True Positive Value is 1749; this means the model properly identified 1749 positive class data.
- True Negative Value is 1131; this means the model properly identified 1131 negative class data.

- False Positive Value is 570; this means the model wrongly identified 570 negative class data as positive class.
- False Negative Value = 359; this means the model wrongly identified 359 positive class data as negative class.

For data visualisation, we used Wordcloud module to print out the most positive words. For graphical representation, we used pie chart to show percentage of negative and positive tweets as shown in figure 3.

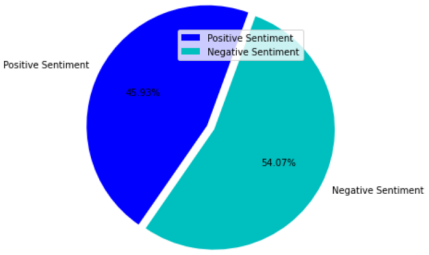


Figure 3. Percentage of Positive and Negative Tweets from dataset

A ROC curve is plotted as shown in figure 4 to compare the likelihood threshold between the true positive and false positive value. Probability threshold is the decision for converting a predicted probability after training of dataset. For normalized predicted probabilities or score is in the range between 0 and 1, the default value for the threshold is 0.5. Prediction ranging below 0.5 is assigned as class 0 indicating lower false positive and greater than 0.5 is called class 1 meaning higher true positive.The area under the curve (AUC) is used as a summary of the model skill; the graph below shows 0.83 true positive value skill.

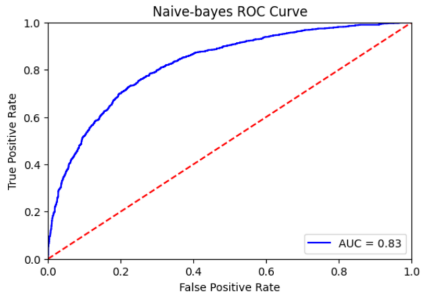


Figure 4. Receiver Operating Characteristic

4 Discussion

Naïve Bayes classifier assumes strong independent relation between the features. In this experiment the assumptions are word frequencies collected from tweets. The training of Naive Bayes classifier requires linear time compared to other expensive approaches [4]. A disadvantage of Naïve Bayes is that it uses independent assumption to make predictions. Another drawback of the algorithm is; it cannot handle sarcasm in a tweet. The presumption of independent predictors is a weakness of Naive Bayes, which results in inaccuracy of probability estimation. If a group attribute in the evaluation data set has a category that is unknown to trained data, the model will result in probability of 0, known as zero frequency, and will be unable to make a forecast.

5 Conclusion and future work

Micro-blogging has been one of the key forms of networking. The vast volume of data found in websites for micro-blogging makes them an attractive source for various types of sentiment analysis. The Naïve Bayes algorithm was used to train a sentiment classifier in this article. The classifier is capable of evaluating both positive and negative emotions.

Advanced technologies of NLP and Machine Learning such as Normalization, Vectorization and stopwords have been used in our algorithm. We used Lemmatization method to normalize each tweet because it converts word to its actual form. We didn't use stemming because it may lead to incorrect word, spelling as it aims to remove the suffix for example; stemming won't be able to normalize between 'dealt', 'dealing' to deal. We used count vectorization instead because Tf-Idf vectorization it functions by assigning the most common words weights from the dataset and gives lower weights to other words. Whereas, count-vectorization converts dataset to a matrix of the counts of each word's occurrences in the text and then assigns numerical representation.

Limitation of the code includes use of Nltk in built stop words that increased classifier accuracy. One limitation of stop words was that we have not used custom stop words in contrast to the data set. Another limitation of our algorithm is that we have not pre-processed contraction words such as 'don't; to 'do not'. This can affect the accuracy of Naïve Bayes classifier.

As the future work, the credibility of the tweets being analysed can be checked by taking reviews from different websites. The sarcastic element could be further assessed in the future by better processing.

ACKNOWLEDGEMENTS

To perform sentiment analysis using the extracted features, we use Multinomial Naïve-Bayes classifier. The source code and dataset has been taken from GitHub (Gunjan993) link provided in code; it has been modified accordingly to my use. This source code is licensed under the MIT license found in the LICENSE file. I would also like to acknowledge websites used such as ''TowardsDataScience" and "GeeksforGeeks".

REFERENCES

[1] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca J Passonneau, 'Sentiment analysis of twitter data', in *Proceedings of the workshop on language in social media (LSM 2011)*, pp. 30–38, (2011).

[2] R. K. Bakshi, N. Kaur, R. Kaur, and G. Kaur, 'Opinion mining and sentiment analysis', in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 452–455, (2016).

[3] Anna Jurek, Maurice D Mulvenna, and Yaxin Bi, 'Improved lexicon-based sentiment analysis for social media analytics', *Security Informatics*, **4**(1), 1–13, (2015).

[4] Alexander Pak and Patrick Paroubek, 'Twitter as a corpus for sentiment analysis and opinion mining.', in *LREc*, number 2010, pp. 1320–1326, (2010).

[5] P. Ray and A. Chakrabarti, 'Twitter sentiment analysis for product review using lexicon method', in *2017 International Conference on Data Management, Analytics and Innovation (ICDMAI)*, pp. 211–216, (2017).

[6] M. Trupthi, S. Pabboju, and G. Narasimha, 'Sentiment analysis on twitter using streaming api', in *2017 IEEE 7th International Advance Computing Conference (IACC)*, pp. 915–919, (2017).