Tweet Sentiment Analysis and Study and Comparison of Various Approaches and Classification Algorithms Used

Aditya Kulkarni[1], Shubham Mhaske[2]

[1]Pimpri Chinchwad College of Engineering and Research, Pune, India
[2]Pimpri Chinchwad College of Engineering and Research, Pune, India

Abstract - *Starting from March 2006, Twitter has been a major face of social media. Twitter provides an efficient way for users to share data in textual, pictorial and video form. Users share multiple aspects of personal and public opinions on various events happening around them. This social interaction has both positive and negative aspects associated with it. Unfortunately, hate speech has been a major issue and as a consequence one of the major drawback of social media platforms. Despite numerous attempts, a perfect detection system is hard to develop due to the vague definition of hate speech and the intent of the writer is not always accurately reflected in the tweet. This study explains, in detail, the processes used to conclude the data (in the form of tweets) and classify them as positive, negative and neutral. This can help in reducing the rate of online harassment and hate speech. This study compares and evaluates various methods of preprocessing of data, feature selection, and predictive algorithms. Various text preprocessing techniques like tokenization, vectorization and supervised classification algorithms like Logistic Regression, Decision Tree, Random Forest Classifier, kNN Classifier, Multinomial Naive Bayes, SVM-C, and Decision Tree are evaluated in this study.*

***Key Words*: Tweet sentiment, Sentiment analysis, machine learning, Opinion Mining, Text Classification, Twitter**

1.INTRODUCTION

Millions of people are using social network sites to express their emotions, opinions and disclose various aspects of their daily lives. By analyzing these tweets various conclusions can be drawn about the current state of person and society regarding the issue that is addressed in the tweet. This research explores the methodology of tweet analysis to get the sentiment from the tweet and apply this knowledge to predict the sentiment from future tweets.

This study explains the logic and process behind building a model that classifies tweets as positive, negative or neutral. The main goal of this project is to define and explore the impact of various methods of preprocessing and various machine learning algorithms that can be used and their respective impact on accuracy and performance. A model sample tweet dataset is used for the same.

1.1 Objective

The goal of this research is to detect hate speech in tweets. If it has a racist or sexist message associated with it, we might suggest a tweet includes hate speech. Therefore, the objective is to separate racist or sexist tweets from other tweets[1]. The major problem on social media is censoring posts that promote Hate Speech without interfering with the user's right to freedom of speech. Hate speech fuels hate crimes that upset social harmony and it's a criminal offense. Early detection of hate speech will avoid more repercussions and will lead to a more stable social media environment. NLP hate speech research is confined due to limitations on defining hate speech due to the demographic conditions of the user and misinterpretation due to language barriers. In this article, various approaches to classify text as hate speech will be evaluated.

1.2 Understanding the dataset

The dataset used in this project is a labeled dataset of 31,962 tweets. The dataset is provided in the form of a CSV file with each line storing a tweet id, its label, and the tweet. In the given dataset, Label ' 1 ' denotes that the tweet promotes hate speech and label ' 0 ' denotes that no hate is directed by the tweet. The tweet can be downloaded from the tweet sentiment analysis competition hosted on AnalyticsVidhya web-portal[1]

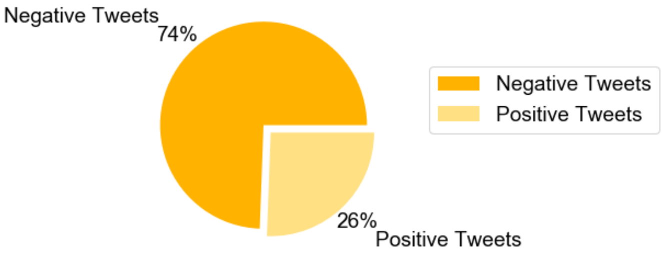


Fig -1: Representation of positive tweets and negative tweets in the training dataset.

2. METHODOLOGY

A majority of this project deals with data in textual form, large use of NLP concepts is done. The project is broadly divided into 4 steps. The steps are listed below:

1. Data Preprocessing and Cleaning: This step involves making data compatible with the input data format of the models.
2. Extracting Features from Cleaned Tweets: After preprocessing data, important features are identified and weighed to create a more efficient model.
3. Model Building: Sentiment Analysis: Based on the nature of data and compatibility, a machine learning algorithm is selected and a model is created.
4. Predicting Sentiments and Evaluation: Model performance is evaluated on various parameters and comparison study is done.

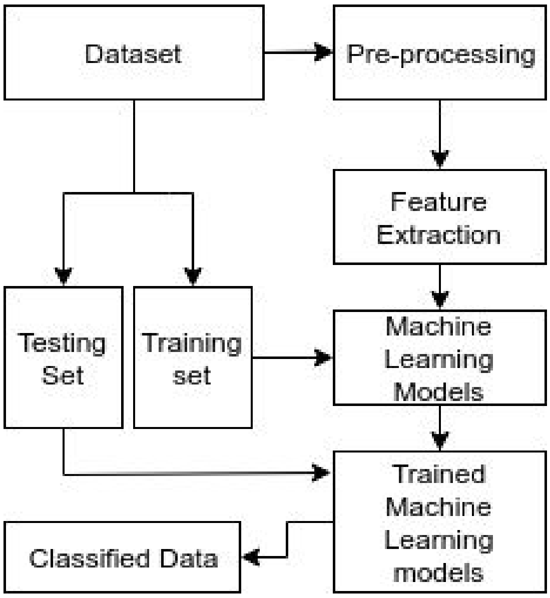


Fig -2: Sentiment Analysis Methodology

3. Preprocessing of data

The data present on twitter is mostly in textual form. The difficulty with analyzing this data is that it contains multiple unwanted features or elements, like twitter handles, emoticons and multiple exceptional characters which may or may not have any effect on the meaning of the tweet. Thus, the preprocessing step has high importance in tweet sentiment analysis. Preprocessing the data requires the use of Regular Expressions, NLP libraries to remove stop-words and stemming or lemmatizing algorithms. The major tasks in preprocessing are discussed in the following sections.

3.1 Removing Twitter user handles(IDs), Punctuations, Numbers, and Special Characters

Twitter handles have unique identifiers defined by user ids that start with the symbol ' @. 'These ids convey no significant information and do not contribute to the task of evaluating sentiments. The tweets also consist of a large number of symbols like punctuation symbols and numbers. Both these text elements provide very less relevance to the definition of the tweet so it is safe and beneficial to remove these elements. These elements are removed with the help of the ReGex library, using regular expressions.

- Regular Expressions:

A regular expression(RegEx) is a pattern describing a certain amount of text, called a search pattern[2]. Regular expressions can be used to check whether a text block has the search pattern or not. Regular Expressions are defined using various symbols that vary depending on the regex engine. In this project, the python library 're' is used to define regular expressions.[2]

3.2 Tokenization

Text is a linear sequence of symbols. In the actual processing of text, the first step is to segment the block into linguistic units such as words, punctuation, numbers, alpha-numerics, etc. The process is known as tokenization. In a sense, tokenization is a part of pre-processing; an identification of the basic units to be processed.

The decision of the basic unit must be taken depending on the need of the application. Most of the applications require tokens in the form of singular words. In tweet analysis, the use of words is an important factor in the determination of the overall sentiment of a tweet. There is no specific function dedicated to tokenization, it must be defined as per the need of the application. String and List manipulation functions and slicing possess an important role in writing a tokenization function[3]

3.3 Removing Short Words and Stop-words

In most of the tokenized text blocks, the words with length less than or equal to 3 pose no importance. For example, the, oh, lol, etc. So removing these words will reduce additional unnecessary computations.

Along with short words, stop words are also removed from the text. Stop words are the words that are the most commonly used in natural languages.\cite{stopwords}

For a given purpose any set of words can be selected as the stop words. In the context of this project, most auxiliary verbs, pronouns and other words with little lexical content like: too, also, just, etc. are considered as stopwords.

To remove stopwords, we use the collection of stopwords present in the nltk.corpus package. This package has a collection of stopwords from all prominent languages.[4]

4. STEMMING AND LEMMATIZING

The process of converting words to their stem form is called stemming. A word stem may not be the same as a morphological root, it is just equal or smaller form of the word. Stemming algorithms are rule-based. It can be viewed as a heuristic process that transforms the words in its smallest possible form i.e. stem. A set of conditional statements are applied to a word and its stem is determined by matching it through multiple conditions[5]}Some of the commonly used stemming algorithms are Porter stemming algorithm and Snowball stemming algorithm.

- Porter stemming algorithm: This stemming algorithm is an older one. It's from the 1980s and its main concern is removing the common endings to words so that they can be resolved to the stem form. It is a very simple algorithm with very little computational complexity. But it should not be used in complex applications as it may provide inaccurate results as it only trims the words.
- Snowball stemming algorithm: This algorithm is also known as the Porter2 stemming algorithm. It is almost universally accepted as better than the Porter stemmer. That being said, it is also more aggressive than the Porter stemmer. A lot of the things added to the Snowball stemmer were because of issues noticed with the Porter stemmer. There is about a 5\% difference in the way that Snowball stems versus Porter.

Lemma is a term used to represent all the other possible forms of the word. The lemma "build," for example, reflects "builds", "building", "built", etc.Lemmatization is the process of converting words into their lemma. Lemmatization uses word vocabulary and morphological analysis to reduce the different forms of the word to its dictionary form. To solve a word to his lemma, a lemmatizer needs to know much more about a language's structure and therefore requires extra computational linguistic power. Algorithms for lemmatization can be either simple dictionary-based or rule-based. Rules can either be hand-crafted or learned automatically from an annotated corpus for the rule-based lemmatizers.[5]

Comparison of Stemming and Lemmatizing:

Lemmatization uses a language dictionary to perform an accurate reduction to root words. Stemming uses simple pattern matching to simply strip suffixes of tokens (e.g. remove “s”, remove “ing”, etc.).Lemmatization is strongly preferred to stemming if available.

5. FEATURE EXTRACTION

To feed it to machine learning algorithms, the textual data must be translated into numerical data. This conversion is called feature extraction. Bag of Word model is used for vectorization of textual data. This model generates a bag or a collection of distinct words, called as vocabulary, from textual data. From this vocabulary, the vectorization uses either the count of word or its frequency in the text. Two basic approaches are described below based on the vectorization technique:

1. Count Vectorizer: Count Vectorizer uses the simplest form in which features are assigned importance. It uses the count ie. The number of occurrences of a word in the document to determine whether it is relevant. But this approach is not very successful because the meaning of the word in the text is not taken into account. It only works on term frequency and is therefore very rarely used.
2. TF-IDF: The frequency of a distinct word in each document provides the term frequency (TF), and the number of documents in which a given word appears defines the Inverse document frequency(IDF). The document is, in our case, a single tweet.TF and IDF together provide a measure as TF-IDF that signifies the importance of a given word in data[6]

$$w_{i,j} = tf_{i,j}.log(\frac{N}{tf_{i,j}})$$

6. MODEL BUILDING: SENTIMENT ANALYSIS

The scope of this study is to implement and evaluate the basic machine learning models, without dwelling into the complex and advanced models used in NLP. This section explores the algorithms that are implemented

6.1 Logistic Regression

Logistic regression is a statistical model used to model a binary dependent variable using a logistic function and it is widely used for classification problems. A linear equation with independent predictors is used by the logistic regression algorithm to predict a value that is a real number, which is squashed to class value (0 or 1) using the Sigmoid function.[7]

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

Here z is the value we want to transform to the class value. Sigmoid function for the Logistic regression is given below in which z is a linear function in a univariate regression model and y is the output variable.

$$y = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1)}}$$

6.2 Decision Tree

Decision Tree Classifier is a supervised method used for classification tasks. It learns a set of rules from the data and uses them for decision-making. The deeper the tree the more complicated the decision-making rules and the more fitter the model. The decision tree is constructed by repetitively partitioning the data set into smaller sets on the basis of a set of tests identified at each node. A decision tree consists of a root node that is the top node and represents the entire training data, decision nodes are created when sub-node splits into additional sub-nodes and terminal nodes that do not further split up. When data sets have a large set of features, a large number of splits result, which in turn gives a tremendous tree. These trees are complex and can lead to overfitting, which can be prevented by pruning. Pruning is the process of removing branches that make use of features of low importance, transforming certain branch nodes into leaf nodes, and removing leaf nodes under the original branch. For splitting the nodes ID3 is a widely used algorithm. ID3 uses Entropy and Information Gain to construct a decision tree. In a collection of examples, entropy is the measure of impurity, disorder or uncertainty. Information gain measures how much “information” a feature gives us about the class.

6.3 Random Forest

Random forest is an ensemble learning method used for classification as well as regression, which operates by constructing a group of decision trees from training data. Random Forests are trained using bagging which reduces the variance of the trees. Bagging is a random sampling of training data subsets, fitting a model to those smaller data sets, and aggregating predictions. Randomness is also introduced by considering subsets of the features when splitting the nodes. The output of the random forest is the class which is the class mode in the task of classification and the mean prediction of the individual trees in the task of regression.

6.4 k-Nearest Neighbours Classifier

The k-Nearest-Neighbours (kNN) is essentially a non-parametric method of classification. For the classification of a data record, its nearest ' k ' neighbors are collected, and this establishes a neighborhood for the data record.[8]The data item class is determined based on voting among the neighborhood records. The classification can be accomplished both with and without distance-based weights[8].K-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computations are delayed until classification. This implies the model does not actually "learn" anything but calculates all the metrics required for each instance. In order to apply kNN, we need to choose an appropriate value for k, and classification success depends very much on this value.

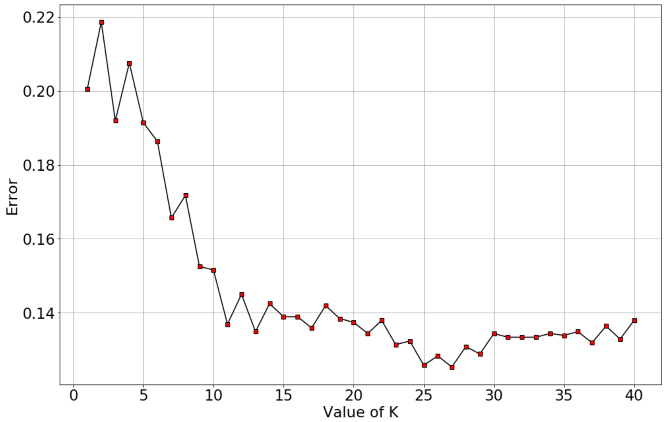


Fig -3: The variation in the value of error with different values for 'k'. This is a result of the application of the algorithm on a small block of data from the training set with different k values.

6.5 Random Forest

Naive Bayes classifier is a supervised approach based on the theorem of Bayes. To predict the sample's group, it makes a naive assumption that each feature is independent of the others. Consider the classes $C_1, C_2, ..., C_k$ to which an object may belong. Naive Bayes algorithm calculates the conditional probability that an object having feature vector $x_1, x_2, x_3, ..., x_n$ belongs to some class C_i [9]

$$P\ (C_i|x_1, x_2 ... x_n)\ \propto\ \left(\prod_{j=1}^{j=n} P\ (|C_i)\right).P(C_i)\ for\ 1 \leq i \leq k$$

Multinomial Naive Bayes classifier is a variant of a Naive Bayes classifier which assumes a multinomial distribution of the features. For modeling feature vectors, a multinomial distribution is useful where each value represents the number of occurrences of a word or its relative frequency.

6.6 Support Vector Machine Classifier

A Support Vector Machine (SVM) is a classifier that discriminates the data in numerous planes. For a given supervised learning instance the SVM gives an optimal hyperplane as output which is used to categorize new data records. A hyperplane is a line dividing a plane into two sections in two-dimensional space, in which each class lay on either side.[10]

The SVM algorithms are implemented using a kernel. The kernel is responsible for the input data getting converted into the appropriate form. The SVM kernel takes input data of small dimensions and transforms it into data space of higher dimensions, thus improving data separability. This technique is called a kernel trick. The most common kernels are listed below:

- Linear Kernel: Linear SVM kernel is used mostly to handle the textual data as it contains multiple features and is mostly linearly separable. The linear kernel can be defined by the following equation:

$$K(x, x_i) = \sum(x, x_i)$$

- Polynomial Kernel: Polynomial kernel is a more generalized form of the linear kernel. The polynomial kernel is used in the case of curved or nonlinear input space. The homogenous polynomial kernel can be defined by the following equation:

$$K(x, x_i) = \sum(x, x_i)d$$

  Here, *d* is called the degree of the kernel . The polynomial kernel is used in natural language processing (NLP). The most common degree is 2 (quadratic) since larger degrees tend to overfit on NLP problems.[10]

- Radial Basis Function(RBF) Kernel: The Radial basis function kernel is a popular kernel function used in support vector machine classification. RBF can handle input in infinite-dimensional space.[10]

$$k(x, x_i) = e^{-\gamma.(x-x_1{}^2)}$$

  Here gamma is a parameter, which ranges from 0 to 1. The higher the value of gamma, the higher the fitting of the model to the training dataset, which causes over-fitting.

7. RESULTS

Fig. 4 and Fig. 5 shows the performance comparison of the models using the AUR-ROC curve. Higher the AUC, the better the model is at distinguishing between positive tweets and negative tweets.

ROC Curve: The Receiver Operating Characteristic Curve (or ROC curve) is the sensitivity vs 1-specificity plot for the different threshold values. When sensitivity increases, the specificity decreases and therefore the sensitivity and the 1-specificity increase or decrease together. The ROC curve is used to compare the performance of different Machine Learning models by taking into account the shape of the curve. The good performance of the model is indicated by a curve that is close to the x-axis and away from the y-axis.

AUC: AUC stands for ' Area under the ROC Curve,'meaning that AUC measures the two-dimensional area below the ROC curve. This indicates how much a model can differentiate between classes. The AUC consolidates the performance information provided by the ROC curve into a single quantitative measure.

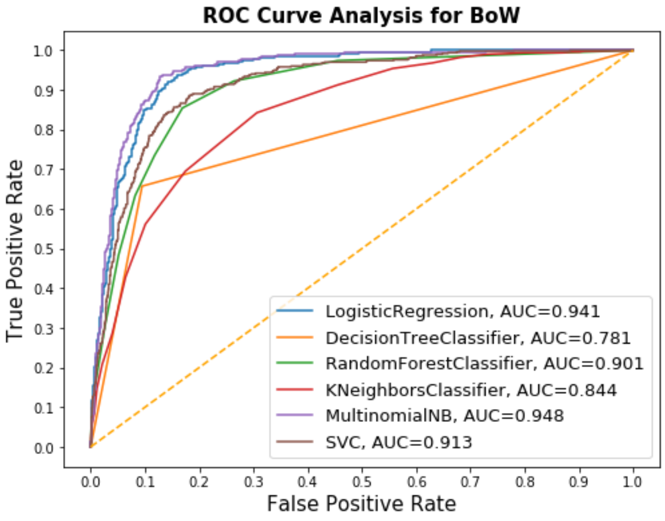


Fig -4: ROC Curve for the bag of words vectorizer

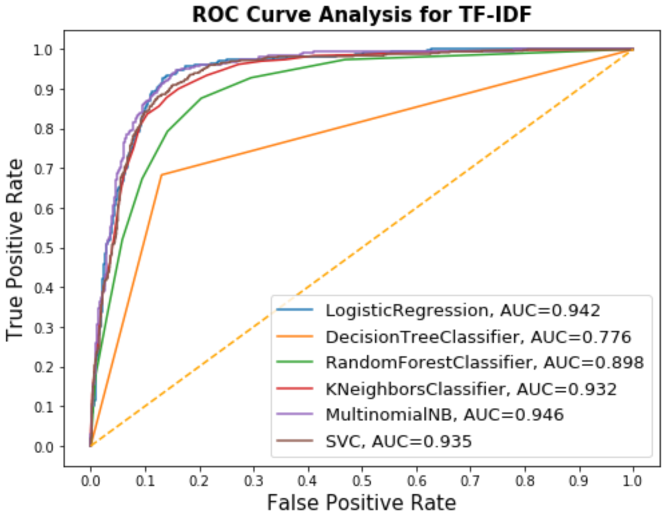


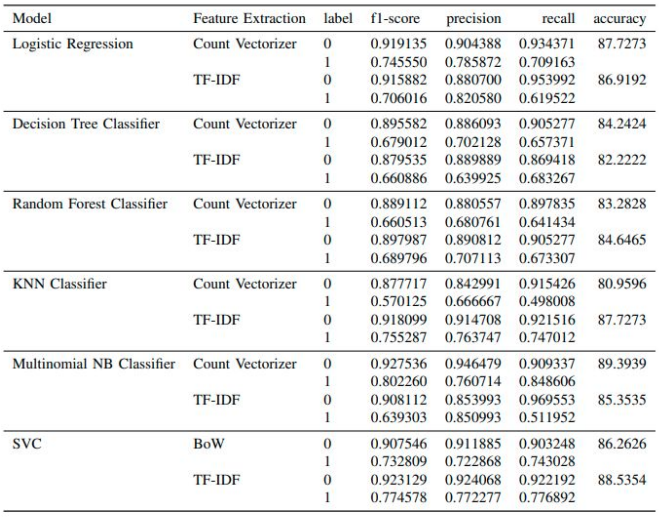
Fig -5: ROC Curve for TF/IDF vectorizer

The most common measures of performance of the machine learning model are accuracy, recall, and precision.

Based on f1-score, precision, recall, and accuracy, we compared the performance of the models in Table 1.

Also, we used two different vectorizers, thus there will difference in the performance of the algorithms.

Table -1: Comparison of multiple performance metrics of different algorithms



Model	Feature Extraction	label	f1-score	precision	recall	accuracy
Logistic Regression	Count Vectorizer	0	0.919135	0.904388	0.934371	87.7273
		1	0.745550	0.785872	0.709163	
	TF-IDF	0	0.915882	0.880700	0.953992	86.9192
		1	0.706016	0.820580	0.619522	
Decision Tree Classifier	Count Vectorizer	0	0.895582	0.886093	0.905277	84.2424
		1	0.679012	0.702128	0.657371	
	TF-IDF	0	0.879535	0.889889	0.869418	82.2222
		1	0.660886	0.639925	0.683267	
Random Forest Classifier	Count Vectorizer	0	0.889112	0.880557	0.897835	83.2828
		1	0.660513	0.680761	0.641434	
	TF-IDF	0	0.897987	0.890812	0.905277	84.6465
		1	0.689796	0.707113	0.673307	
KNN Classifier	Count Vectorizer	0	0.877717	0.842991	0.915426	80.9596
		1	0.570125	0.666667	0.498008	
	TF-IDF	0	0.918099	0.914708	0.921516	87.7273
		1	0.755287	0.763747	0.747012	
Multinomial NB Classifier	Count Vectorizer	0	0.927536	0.946479	0.909337	89.3939
		1	0.802260	0.760714	0.848606	
	TF-IDF	0	0.908112	0.853993	0.969553	85.3535
		1	0.639303	0.850993	0.511952	
SVC	BoW	0	0.907546	0.911885	0.903248	86.2626
		1	0.732809	0.722868	0.743028	
	TF-IDF	0	0.923129	0.924068	0.922192	88.5354
		1	0.774578	0.772277	0.776892	

8. CONCLUSIONS AND FUTURE SCOPE

In this paper, We first described the pre-processing steps for cleaning unstructured Twitter. We discussed various techniques for extracting features from this textual data and discussed the different machine learning models as well. We have trained these machine learning models to perform sentiment analysis on tweets and to classify them as either positive or negative. Also, we compared and evaluated the results of various algorithms on the data.

Future opportunities in this field are developing techniques for aspect-based sentiment analysis, which will take into account the different aspects mentioned in the text. Social media data can be in different languages, making it a barrier to sentiment analysis. Developing a multilingual sentiment analysis approach is, therefore, a key challenge.

ACKNOWLEDGEMENT

We would like to extend our special thanks to Prof. Dr. Archana Chaugule, Head, Department of Computer Engineering, and Mr. Mahendra Salunke, Asst. Professor, Department of Computer Engineering, Pimpri Chinchwad College of Engineering and Research for their encouragement and useful critiques for this research work.

REFERENCES

[1] “Twitter-sentiment-analysis-supervised-learning,”2018[Online]

[2] J.Goyvaerts, Regular expressions: the complete tutorial.LightningSource,2006[Online].

[3] C. Trim, “The art of tokenization,” Jan 2013. [Online]

[4] “Accessing text corpora and lexical resources” [Online].

[5] H.Heidenreich,“Stemming?lemmatization?what?”Dec2018. [Online]

[6] J. Ramoset, “Using TF-IDF to determine word relevance in documentaries,” in Proceedings of the first instructional conference on machine learning, vol. 242. Piscataway, NJ, 2003

[7] R. Gandhi, “Introduction to machine learning algorithms: Logistic regression.” [Online]

[8] G. Guo, H. Wang, D. Bell, and Y. Bi, “Knn model-based approach in classification,” 2004.

[9] R.Jain,“Introduction to naive Bayes classification algorithm in python and r,” May 2019. [Online]

[10] T. Evgeniou and M. Pontil, “Support vector machines: Theory and applications,” vol. 2049, 01 2001

[11] A. Sarlan, C. Nadam, and S. Basri, “Twitter sentiment analysis,” 2014

[12] T. W. Gruen, T. Osmonbekov, and A. J. Czaplewski, “eWOM: The impact of customer-to-customer online know-how exchange on customer value and loyalty,” Journal of Business Research, vol. 59, no. 4

[13] A. Krouska, C. Troussas, and M. Virvou, “The effect of preprocessing techniques on twitter sentiment analysis,”2016 7th InternationalConference on Information, Intelligence, Systems Applications (IISA),2016

[14] P. Wang, G. R. Bai, and K. T. Stolee, “Exploring regular expression evolution,”2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER), 2019

[15] V. S and J. R, “Text mining: open source tokenization tools – an analysis,” Advanced Computational Intelligence: An International Journal(ACIIJ), vol. 3, no. 1, 2016