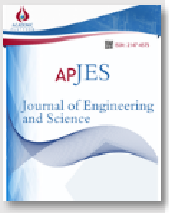
Academic Platform Journal of Engineering and Science

journal homepage: https://dergipark.org.tr/tr/pub/apjes



Sentiment Analysis of Twitter Texts Using Machine Learning Algorithms

*[1]Hawar Sameen Ali Barzenji
[1] Department of Computer Engineering, Faculty of Computer and Information Sciences, Sakarya University, Sakarya, Turkey,
hawar.al-barzenji@ogr.sakarya.edu.tr,

Research Paper | Arrival Date: 19.05.2021 | Accepted Date: 19.07.2021

Abstract

Since the two last decades social media networks have become a part of our daily life. Today, getting information from social media, tracking trends in social media, learning the feelings and emotions of people on social media is very essential. In this study, sentiment analysis was performed on Twitter text to learn about the subjective polarities of the writings. The polarities are positive, negative, and neutral. At the first stage of the sentiment analysis a public data set has been obtained. Secondly, natural language processing techniques have been applied to make the data ready for machine learning training procedures. Lastly sentiment analysis is performed by using three different machine learning algorithms. We reached 89% accuracy with Support Vector Machines, 88% accuracy with Random Forest, and 72% accuracy with Gaussian Naive Bayes classifier.

Keywords: Natural Language Processing, Machine Learning, Random Forest, Support Vector Machines, Gaussian Naive Bayes, Sentiment Analysis

1. INTRODUCTION

With the rise of the modern era, our life faced a new way of communicating, a new way of social interaction [1]; which is the social media platforms. Willy-nilly if we accept it or not, SM became a phenomenon in our daily life; it became an essential part of our recent lifestyle.

Nowadays, people use social media not only for sharing their emotional, desire, and ideas about a particular subject [2]; but also, they use it for marketing [3], political messages, and etc. A huge attention for the latter mentioned category is Twitter's platform. It is clear that most of the politicians around the world are using Twitter as their no. one's favorite platforms, but we should not forget that this platform has its own positive and negative effects on the decision of the people, and what is familiar among the people about all of the social media platforms is that most of the time, its negative side is more than its positive side.

For the all above important reasons of social media, it's the right time to put our focus on this new phenomenon, from philosophy to machine learning (ML), NLP and other evaluating techniques in data analysis, can be used to analyze the activities on this platform.

The Twitter's account of President Donald Trump has about 88 million followers [4] of Twitter users, and many of them are using his account as a way of news and information resource [5]. His frequent use of the social media account and his influence as President of the US, has made his tweets an essential source in a variety of scientific and research studies, like evaluating his tweets [6] [7], applying sentiment analysis, etc.

In this study, sentiment analysis is performed on Twitter text. The tweets which are used in this research are Trump's tweets published freely online [8]. Obtained tweets were firstly prepared for ML algorithms by using natural language processing techniques (NLP). Then three ML classifiers were trained using the data. The ML techniques are Random Forest, Support Vector Machines (SVM), and Gaussian Naive Bayes (GNB).

1.1. NLP

NLP is a subfield of Computer science and AI (particularly the field of machine learning). It deals with the language of the human being and how it understood by the computer. This technique can be obtained with the help of the computational linguistics. To understand the natural language needs a lot of information about lexicon, semantics, syntax, and information about our real world [29, 30, 31].

We can talk about NLP as a synthesis of philosophy of linguistics, computer science, and artificial intelligence. This branch of science deals with the interactions between human language and computers (Robot agent). This field cares about how to code and program computers, in order to process and analyze the data of natural language.

*[1] Corresponding author: Sakarya University, Sakarya, Turkey, hawar.barznji@gmail.com, +90 552 353 2395

Doi: 10.21541/apjes.939338

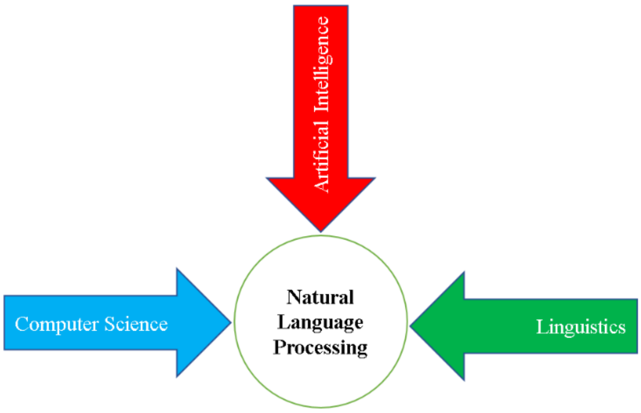


Figure 1. The synthesis of NLP

1.2 ML classifeirs

Machine learning is about taking out knowledge from raw data, and learning from past experience in order to predict the next upcoming data. This research field is an intersection of AI, statistics, and computer science. The usage of machine learning methods in recent years is very useful in nowadays life. Starting from automatic suggestions of which videos to be watched, or what type of fast-food to order or which items to buy, and for customizing the podcasts; most of the modern portals and devices have machine learning algorithms at their kernel; and ML can do all of these based on learning from experiences, the more training the classier, the more accurate the prediction is.

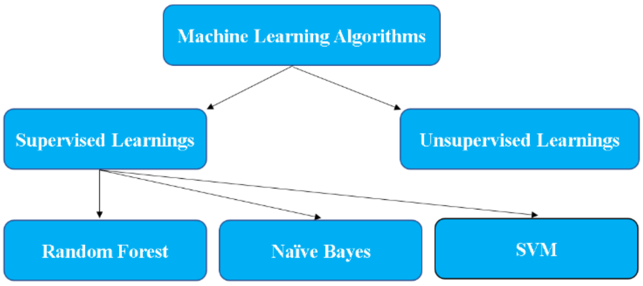


Figure 2. The diagram of ML approach

1.2.1 Random Forest

Originally Random Forest derives from Decision Tree, this means, it shares all the benefits of decision trees, but historically it refers back to an American computer scientist at IBM Watson Health (Tin Kam Ho) in 1995 with the term of (random decision forests) [9]. After a while (Leo Breiman) coined the Random Forest term in 2001 [10].

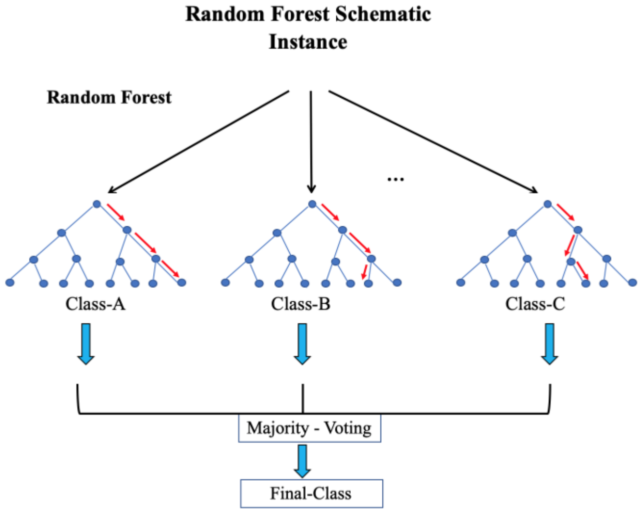


Figure 3. Random Forest schematic

1.2.2. Gaussian Naïve Bayes

Naive Bayes is a kind of probabilistic or statistical supervised ML algorithm. It builds a probability model on the category description for all feature vectors in the training set. It works based on Bayes theorem [11], which calculates conditional probability. Gaussian distribution, is one of the most usual and main technique in calculating statistics and probability field, stating the "naive" supposition of conditional independence between every pair of attributes given the value of the class variable [12].

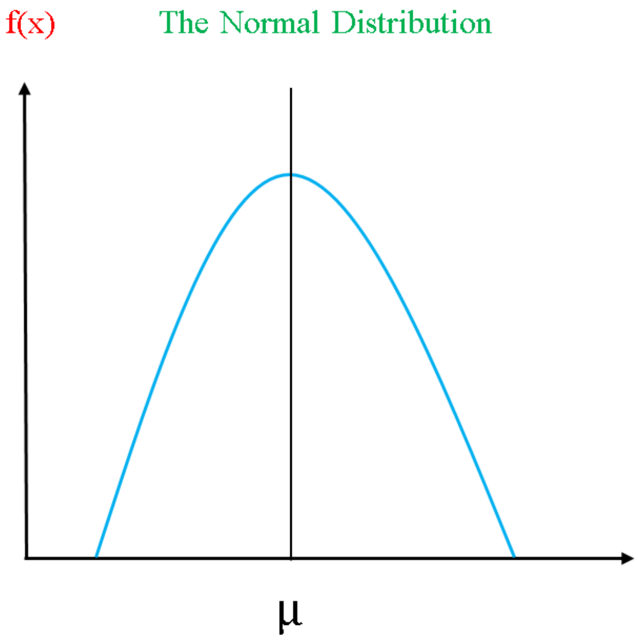


Figure 4. The Gaussian distribution

1.2.3. Support Vector Machine

Support Vector Machine (SVM) or originally Support Vector Networks (SVN), is a type of supervised ML algorithm that was coined by both C. Cortes and V. Vapnik in 1995 [13]. It can be used in both classification and regression tasks. This prediction tool uses ML theory to maximize required accuracy and automatically avoids overfitting of the data. This supervised learning ML uses in two group classification problems. It can solve linear and non-linear. This algorithm is efficient when dealing with high dimensional data such textual data. The idea of SVM is simple, its objective is to find a hyperplane that has the largest edge (side), i.e. the decision boundary that separates the support vectors to the farthest.

Table 1. The number of input features with the required number of hyperplanes

Input features	Output hyperplane
2-features	1-line hyperplane
3-features	2-dimentional plane

In SVM, we are trying to find those points which are the closest to the line from both the classes, the points are called *support vectors*. Then, the distance between the line and the support vectors will be calculated, this distance is called the margin [32]. The goal of SVM is to maximize the margin, as it shown in below figure.

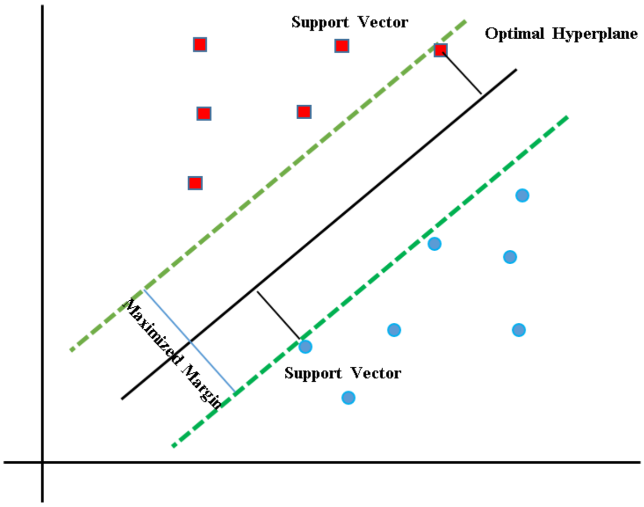


Figure 5. Optimal Margin

Finally, since every approach is measured by its outcomes, Random Forest classifier has been applied to get the following targets; *Accuracy*, *Precision, F1 Score*, and *Recall* metrics [14] [15]. Also, the data size of cleaned text with its original source have been compared and tested in macOS operating system for information purpose; the results have been showed in tables and related figures.

The suggested research paper can have a good impact in many scientific areas and can have a good contribution to science. The proposed system can be used in pre- and post-scraping processes. It can be used during web page scraping before the crawled data is going to be saved in the form of csv file; or after saving the crawled page.

Also, it can be used in various fields; It can be used in Digital Image Processing (DIP) and Pattern Recognition (PR); since there are images that contain textual data, it can be extracted by one of the mentioned proceedings, and the proposed system can be applied on it, of course, the more gathered data, the more accurate it will.

Also, it can be used in AI and robotics; the agents can use the suggested method in converting speech-to-texts, which is famous as Speech Recognition (SR) for cleaning the noisy speech in order to understand and perform the required given orders, with the help of time, the agent can get more experience and gradually becomes smarter.

Since philosophy is one of the closest fields to AI and NLP, it can be used in social sciences, like the philosophy of linguistics, we shall not forget that philosophy as the mother of all sciences, raised the first and most early questions about language and a thinker machine.

Also, the mathematical results in confusion matrix, classification report table can be used in the field of data analysis and statistical purposes.

The framework of this research paper will be as follows: In section 1; a general introduction about the whole study (i.e. the problem definitions) is given, like NLP, all three types of ML classifiers; also, other related topics like scopes, and the suggested solutions have been given too. In section 2; the closest studies in the same area (i.e. related work) have been given. In section 3; which is the material and methods; a full detailed explanation about and text cleaning procedure (i.e. pre-processing) with all of its included steps have been given. In section 4; is the results in confusion matrices, classification report, and the conclusion with our recommendations for the future upcoming studies.

2. RELATED WORK

Nowadays Twitter sentiment analysis gained most of the researcher's attention [16]. These concise texts are used as a raw material for data analysis. By using text polarities (positive, neutral, and negative), emotions (angry, sad, loved, etc.) are judging on each text’s subjectivities.

Before going deeper into our own study. We will give a brief overview about the previous articles (i.e. Literature Review) that have been done in the same area which is the combination of NLP and ML.

In [17] they proposed a study for detecting fake news spread through images from SM like Facebook, Twitter, etc. They proposed K-means clustering (based on issuing day) to get a general outline of how the images were used throughout the time.

In [18] they introduce a hybrid method which is a combination of NLP and ML techniques to guess and recognize hate speech from social network websites. After gathering the hate speech, steaming, tokenizing, unwanted character removal was applied on it. Finally, they classified the texts into neutral, offensive, and hate (in our study, we classified the tweets into positive, neutral, negative) language. The performance of the system is then evaluated using overall accuracy, f1 score, and precision and recall metrics. The system achieved an accuracy of 98.71%.

In [19] they applied NLP techniques to analyze tweets with regard to mental health. They used Deep Learning (DL) models to classify each tweet regarding of the following emotions: angry, anticipation, disgust, frighten, delight, sadness, surprise, and confidence.

In [20] a group of researchers made a comparison study of the Naïve Bayes algorithm and NLP on the dataset of Twitter. Their comparison is in two categories: *accuracy* and *speed.* Their experimental results showed that the Naïve Bayes algorithm got 63.5% accuracy, which is lower than that achieved by the NLP method. But in the processing speed analysis, the ML method performance is 5.4 times higher than that of the NLP method.

In [21] they used sentiment analysis to extract human feeling and evaluate whether it's negative, positive or neutral. Through unconstructed text by using NLP. They also Machine learning in order to train and test the dataset. They compared the results using different ML classifier, like Naïve Bayes, Random Forest, Support Vector Machine, and etc.

In [22] USSAMA YAQUB applied sentiment analysis on trump's tweets during the early appearance of the coronavirus pandemic (i.e. COVID-19) in the United States. Statistically, he discovered a negative correlation between the sentiments of his tweets and the no. of cases in the United States. One thing which is very important in his study research is that he noticed a gradual shifting in his tweets from positive to negative sentiments polarities while he is mentioning China and COVID-19 together. What USSAMA did is amazing, but his study is not a hybrid method, which means he didn't apply ML classifier after his sentiment analysis, this makes his research stay in the domain of data analysis and NLP techniques.

In [23] In this paper, they analyzed the relationship between the tweets written by POTUS (stands for the President of the United States) and his approval rating using sentiment analytics and data visualization tools. They applied all the NLP requirements on the tweets of POTUS; they mined, cleaned, and gave a quantitative measure based on the content, which they named the "sentiment score". By comparing tweets before the election, during the election and inauguration, and after the inauguration, they found that the "sentiment score" of Trump's tweets feed has been increased with an average in time by a factor of 60%. By using cross-correlation analysis, they find a preliminary causative relationship between POTUS Twitter activity and approval rating. Still, their study is one-sided research, it seems something is missing. What we do with sentiment analysis and NLP techniques, somehow leaves the problem unsolved. By using ML methods, we can train our data in a way that can recognize the next upcoming data which gives to the system, so the robot can predict it.

In [24] this paper, they used social media content to forecast real-world result. In particular, they used the chatter from Twitter platform to predict box office incomes for movies. They revealed that the tweets which are generated about specific movies can perform better in market-based predictors. They applied sentiment analysis on the extracted Twitter data, but they didn't mention which method they did the forecasting.

From all above research studies and articles, we can notice that most of them have a combination method, which means a duality of NLP and ML algorithms. It seems that without combining those two fields, the suggested work would be incomplete. In our research, after applying NLP techniques on the Twitter texts, Random Forest classifier, GNB, and SVM have been to train and test the cleaned texts.

3. MATERIALS AND METHODS

In this section, which covers the most important part of our study, talks about the most required methods and algorithms that need to be applied to our dataset in order to get the target results.

As it already mentioned above, the aim of this study is to apply sentiment analysis on Twitter's textual data and performing text polarities on it. At the final step, the Random Forest classifier, GNB, and SVM have been used to train and test the data. The accuracy and time of the used classifier have been compared. The results showed the proposed method is working well. The details of the results will be given in a confusion matrix in the result section.

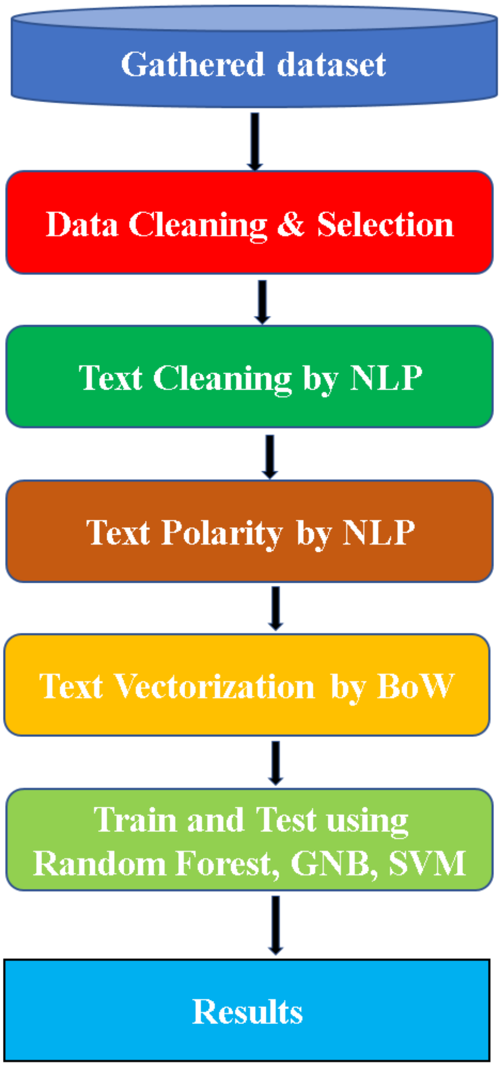
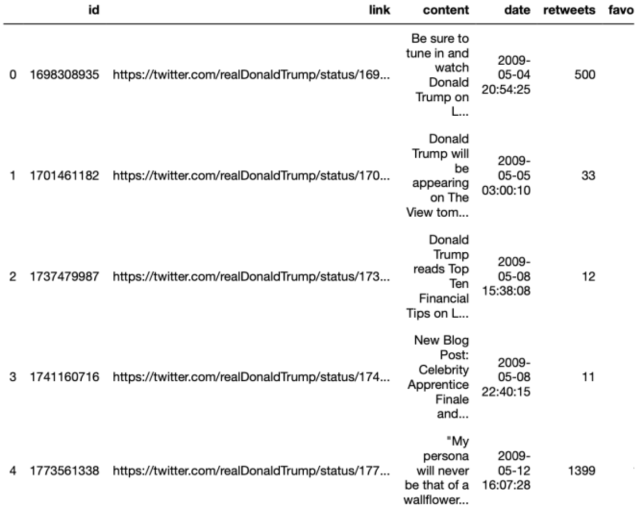


Figure 6. A quick overview of the proposed study

The above diagram displays the most essential steps in the proposed research. It starts with the collected tweets and applies the most required processes which are the data cleaning and selection level. In text cleaning which represents all the NLP techniques that have been used in order to prepare the text for conversion. At text polarity level, each cleaned tweet has been judged regarding their subjectivity. At text to number level, BoW has been applied for converting the categorical data (i.e. textual data) into numerical data. Finally, in the ML level, Random Forest classifier, GNB, and SVM have been used to train and test the data for getting the required results.

3.1. Gathered dataset

In this study, an already prepared dataset has been used from kaggle online community for data scientists and ML practitioners, the dataset is Trump's tweets. It can be found and downloaded from the cited link [25].



	id	link	content	date	retweets	favo
0	1698308935	https://twitter.com/realDonaldTrump/status/169...	Be sure to tune in and watch Donald Trump on L...	2009-05-04 20:54:25	500	
1	1701461182	https://twitter.com/realDonaldTrump/status/170...	Donald Trump will be appearing on The View tom...	2009-05-05 03:00:10	33	
2	1737479987	https://twitter.com/realDonaldTrump/status/173...	Donald Trump reads Top Ten Financial Tips on L...	2009-05-08 15:38:08	12	
3	1741160716	https://twitter.com/realDonaldTrump/status/174...	New Blog Post: Celebrity Apprentice Finale and...	2009-05-08 22:40:15	11	
4	1773561338	https://twitter.com/realDonaldTrump/status/177...	"My persona will never be that of a wallflower...	2009-05-12 16:07:28	1399	

Figure 7. The first five records of our host dataset

3.2. Data cleaning and selection

Whenever there is a dataset, there should be data analysis too, for the reason that any dataset needs some special commands to manipulate them. Data analysis performs most of the actions that need to be done on any dataset, including importing the dataset, performing most of the actions on its columns and rows, appending and deleting the records, and etc. without data analysis, applying NLP and ML algorithms would be impossible. Deciding which features should be used and which one should be eliminated, will be occurred in this step in any studying research in the same area.

As it clear, data visualization (like charts, infographics, etc.) is giving a good way to represent the important information based on the dataset, but what if your raw data is textual-based document? The solution is using Wordcloud which is available in Python programing language. Wordcloud refers to a cloud filled with lots of words in different shapes and sizes. The size of each of the word represents the frequency or the importance of each word; bigger size, means more repeated word. From the below figure, you will see the Wordcloud of our dataset for the feature of "content".

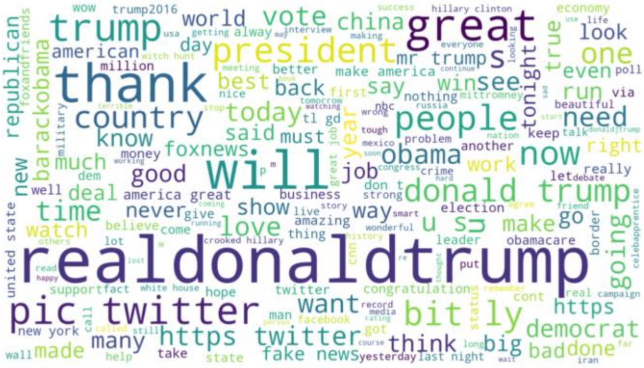


Figure 8. Wordcloud for Trump's tweet

From the Wordcloud above, the words with bigger size represent the most repeated words in the tweet dataset.

3.3. Text Cleaning

The data (in our case Twitter texts) needs to be fully cleaned and prepared before applying any classifier algorithms. In every text, there are many (mentions, hashtags, emoticons, unconventional punctuation, spaces, symbols), that do not have any value on classifying, have to remove (filter out). One of the biggest advantages of this step is that it makes our data smaller which saves our storage capacity. Decreasing the size of the hosted dataset can have a good effect on the performance of the work and the data size can be used for information purpose. (Details have been given in the result section).

In the experimental part, the following text cleaning which includes the following steps have been applied in the dataset: first, stopwords have been removed, then word lemmatizing has been applied in order to change the words into their roots, finally, regular expressions have been applied for removing the links and emails, etc.

3.4. Detecting text polarity

This part is one of the main goals of the study. What we do from the beginning until the final step, is to prepare the text for subjective sentiment polarities (or in some resources, sentiment score). Text polarity is a method to detect each tweet's subjectivity. Since the tweets have been written by human as a subject, and he is tweeting his own ideas about a specific event or anything else, so the tweets are not objective. It needs to be detected in order to be classified into three levels, which are positives, negatives, and neutrals.

In our experimental work, in each sentence has been judged after being cleaned by NLP techniques. Each textual data (In our case is Twitter tweet) is labeled with three possible values: negative, positive or neutral. In this work, we first determined the sentiment polarity of each tweet by adapting the following measurement [26] [27],

$$Sentiment\ Score\ (C) = \frac{Positive-Negative}{Positive+Negative+2} \qquad (1)$$

Where,
Positive represents total number of the positive words; and negative counts the negative words in the tweet. We represent it by a separate two valued with variable C, which represents the sentiment class:
$C \in \{-1, 1\}$.

Where,
C can hold three values, since of having different thresholds,

$$C = \begin{Bmatrix} 1\ (Positive) & if\ Sentiment\ score \geq 0.1 \\ -1\ (Negative) & if\ Sentiment\ score\ \ < 0.1 \\ 0\ (Neutral) & ifSentiment\ score\ \ \ = 0 \end{Bmatrix} \qquad (2)$$

In the Python programming language, there is a library for text polarity, with the name of TextBlob (Also VADER can be used).

From the experiment result, the total of (41122) records, the distribution of sentiment polarities will be as follows:

Positive tweets: $\frac{22274}{41122}$ = 54.16%

Negative tweets: $\frac{7148}{41122}$ = 17.38%

Neutral tweets: $\frac{11700}{41122}$ = 28.45%

If we put all the results in one table, we will get,

Table 2. Sentiment polarities of our dataset

Sentiment polarities	No. of occurrences	Percentages
Positive	22274	54.16%
Negative	7148	17.38%
Neutral	11700	28.45%
Total	41122	100%

As it can be seen from the result table above, the majority of the tweets are positive, and we got the least number of negative tweets, also about 12000 neutral tweets. The graphical distribution of the tweets will be as follows,

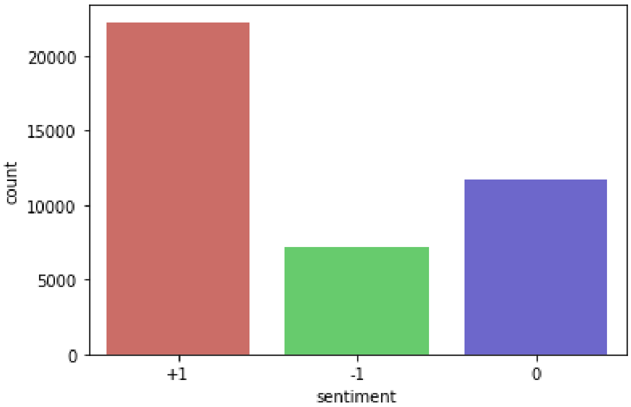


Figure 9. Sentiment polarity distribution of the tweets

3.5. Text Vectorization

Since all of the ML classifiers are dealing with numbers only, the cleaned text has to be changes into a matrix of numbers, and the field will be ready for the training and test process. Text vectorization is a technique of changing texts into quantitative data.

There are some popular types of text vectorization, which they all do the same task but in different ways. Some of them are:

1. Bag of Words (BoW)

2. TF-IDF

3. Word2Vec

3.5.1. Bag of Words

Simply, BoW is a method for representing text in the form of numbers. This model is used for simplifying representation which is used in NLP and information retrieval (IR). In this method, a list of all the text will be considered as a bag of its words, with ignoring the grammar and even the order of the words, but protecting variety.

Simply, BoW is a link between NLP and the ML classifier. It connects NLP techniques to ML.

Figure 10. BoW as a link between NLP and ML

In order to clarify BoW concept, in the example below, let's take three sentences:

Sentence 1: "*The wolf sat*"

Sentence 2: "*The wolf sat on the hill*"

Sentence 3: "*The wolf with the hill*"

We will construct a vector form, from all the unique words in the above three sentences. This vector contains six words which are: 'The', wolf', sat, 'on', 'hill', 'with'. Finally, we will make a table for the results,

Table 3. The BoW table

	the	wolf	sat	on	hill	with
Sentence 1	1	1	1	0	0	0
Sentence 2	2	1	1	1	1	0
Sentence 3	2	1	0	0	1	1

So, the numerical ouotputs for each sentence will be a vector as follows:

Vector 1: [1, 1, 1, 0, 0, 0]

Vector 2: [2, 1, 1, 1, 1, 0]

Vector 3: [2, 1, 0, 0, 1, 1]

3.6. Splitting into train and test

Every used machine learning algorithm needs to a technique which is a kind of division. In this process, the whole dataset will be divided into two parts: Training and Testing. It's up to the researcher divides each part into how many percentages. It can 80%, 20% for both training and testing respectively. Also 70%, and 30%. This method is used to evaluating the performance of the used machine learning algorithm. As we said earlier, this process requires taking the dataset and dividing it into two subsets:

Training set: Used to fit and train the machine learning model.
Testing set: Used to evaluate the fit machine learning model.

This technique is an important step in any supervised learnings. While the agent does not have any default information about the environment, this procedure gives that ability to the agent to learn from the experiences by training more than half of the data. In most of the cases, 70% of the dataset is given to the agent in order to learn from the training; and the remaining part which is 30% is put for the test to see the accuracy of the used classifier, in order to check whether it works good or not. In case if the suggested ML algorithm is not doing well, another classifier has to be applied on the hosted data. The figure below will explain the procedure of splitting in ML.

For example, in our case, the total records of the dataset equal 41122 records. The mathematical calculating of the splitting method (30% and 70%) for the three proposed classifiers will be as follows:

Training set: $70\% \times 41122 = \frac{70}{100} \times 41122 = 0.7 \times 41122 =$ 28785 records

Testing set: $30\% \times 41122 = \frac{30}{100} \times 41122 = 0.3 \times 41122 =$ 12337 records

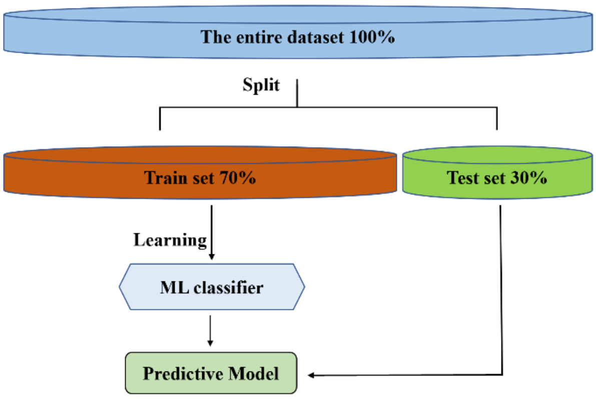


Figure 11. The procedure of splitting dataset in ML

4. RESULT AND DISCUSSION

This section has been divided into two part. In the first part, a formula of each rule has been given, and examples in order to clarify each used formula. In the second part, the experimental results have been discussed and the output of the confusion matrix have been evaluated; in additional, with checking the size of the dataset before and after cleaning.

We used macOS operating system; with the following technical specifications (which is shown in the below table) for training and testing each classifier:

Table 4. The technical specifications of the used host computer

Computer manufacture	**Type of OS**	**Processor**	**Amount of RAM**
MacBook Pro	macOS Big Sur 2020	Intel Core i5, ~2.6 GHz	8GB

4.1. The evaluation metrics

This metrics measurements concludes the following results:

1. Accuracy,
2. Precision,
3. Recall metrics,
4. F1 score.
5. Required time for training and testing
6. Data size

Accuracy is calculated as the total number of correct predictions, over the total number of the dataset (i.e. all correct / all). The rule of accuracy is,

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{3}$$

The rule of precision is,

$$Precision = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Positive\ (FP)} \tag{4}$$

The rule of recall is,

$$Recall = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Negative\ (FN)} \tag{5}$$

Recall works on the horizontal lines (i.e. the rows) of our table.

The solution for the misleading performance of accuracy on imbalanced data, is F1 score. We use F1 score when our data is imbalanced. F1 score is the average of precision and recall. The rule of F1 score is,

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{6}$$

In order to understand the above rules, we will take an example.

Consider a classification system that has been trained to classify (or recognize) the pictures of three types of animals: phoenix, owl, and wolf. The system gave the results in a confusion matrix. Assume that the number of animals are given to the system which are 30 animals; there where 10 phoenixes, 8 owls, 12 wolves.

Table 5. Confusion matrix for three samples of animals

		Predicted class			Total no. of each
		phoenix	owl	wolf	
Actual class	phoenix	6	3	1	10
	owl	1	5	2	8
	wolf	0	1	11	12

In this confusion matrix 3 by 3 table, out of 10 actual phoenixes, the system predicted that 3 were owls, and 1 was a wolf; and of the 8 owls, it predicted 1 was a phoenix, and 2 were wolves; and out of 12 wolves, predicted 1 was owls. The green colors are the true actual values for each class.

Considering the confusion matrix above, the corresponding table of confusion which is (Table 5), for the phoenix, owl, and wolf classes, would be as follows:

Table 6. Confusion matrix for phoenix class

6 true positive (actual phoenixes that were correctly classified as phoenixes)	1 false positive (owls that were incorrectly labeled as phoenixes)
3, 1 false negative (phoenixes that were incorrectly marked as owls, and wolf respectively)	19 true negative (all the remaining animals, correctly classified as non-phoenixes)

Table 7. Confusion matrix for owl class

5 true positive (actual owls that were correctly classified as an owl)	3, 1 false positive (phoenixes and wolf that were incorrectly labeled as owl respectively)

1, 2 false negative (owls that were incorrectly marked as phoenix and wolf respectively)	18 true negative (all the remaining animals, correctly classified as non-owls)

Table 8. Confusion matrix for wolf class

11 true positive (actual wolves that were correctly classified as wolves)	1, 2 false positive (phoenix and owls that were incorrectly labeled as wolves)
1 false negative (wolf that was incorrectly marked as owls)	15 true negative (all the remaining animals, correctly classified as non-phoenixes)

The results of (Table 5) will be as follows,

$$Overall\ accuracy = \frac{(5+6+11)}{30} = \frac{22}{30}$$
$$= 0.73 \times 100\% = 73\%$$

$$Precision\ for\ phoenix\ class = \frac{6}{(6+1+0)} = \frac{6}{7}$$
$$= 0.85 \times 100\% = 85\%$$

$$Precision\ for\ owl\ class = \frac{5}{(5+3+1)} = \frac{5}{9}$$
$$= 0.55 \times 100\% = 55\%$$

$$Precision\ for\ wolf\ class = \frac{11}{(11+2+1)} = \frac{11}{14}$$
$$= 0.78 \times 100\% = 78\%$$

$$Overall\ average\ precision = \frac{(0.85 + 0.55 + \ 0.78)}{3}$$
$$= \frac{2.18}{3} = 0.72 \times 100\% = 72\%$$

$$Recall\ for\ phoenix\ class = \frac{6}{(6+3+1)} = \frac{6}{10}$$
$$= 0.6 \times 100\% = 60\%$$

$$Recall\ for\ owl\ class = \frac{5}{(5+2+1)} = \frac{5}{8}$$
$$= 0.62 \times 100\% = 62\%$$

$$Recall\ for\ wolf\ class = \frac{11}{(11+1+0)} = \frac{11}{12}$$
$$= 0.91 \times 100\% = 91\%$$

$$Overall\ average\ recall = \frac{(0.60 + 0.62 + \ 0.91)}{3} = \frac{2.13}{3}$$
$$= 0.71 \times 100\% = 71\%$$

$$F1\ score = 2 \times \frac{(0.72 \ \times \ 0.71)}{(0.72 \ + \ 0.71)} = 2 \times \frac{0.5112}{1.43}$$
$$= 2 \times 0.357 = 0.71 \times 100\% = 71\%$$

4.2. Evaluating the experimental results

From the three tables below, all the results from the classification report for the three algorithms will be as below,

Table 9. Classification report for Gaussian Naïve Bayes classifier

Types of polarity	**Precision**	**Recall**	**F1-score**
Positive	0.95	0.60	0.74
Negative	0.54	0.73	0.62
Neutral	0.65	0.96	0.78
Accuracy	72%		

Table 10. Classification report for SVM classifier

Types of polarity	**Precision**	**Recall**	**F1-score**
Positive	0.96	0.91	0.93
Negative	0.90	0.73	0.80
Neutral	0.81	0.98	0.88
Accuracy	89%		

Table 11. Classification report for Random Forest classifier

Types of polarity	**Precision**	**Recall**	**F1-score**
Positive	0.93	0.91	0.92
Negative	0.88	0.67	0.76
Neutral	0.81	0.95	0.87
Accuracy	88%		

From the results above, it can be noticed that the applied algorithms work on both positive and neutral polarities better than negative tweets, this due to the total no. of negative tweets which lesser that other polarities. As it's given in (Table 1) the total no. of the whole records are 41122 tweets; from this number, 33974 tweets (which is about 83% of the dataset) are both positive and neutral, only 7148 tweets are negative; which is equal to,

$$\frac{7148}{41122} \times 100\% = \ 0.1738\ \times 100\% = 17.38\%$$

17% of the whole dataset; that's why we notice from the CM table, the proposed algorithms are not working well on negative polarities. Since the agent does not have any previous knowledge about the dataset, it has to be trained a lot, the more data for training, the more accuracy results will get.

In some cases, we may face imbalanced data, which means the data in the host dataset are not coherent. Due to this reason, measuring the accuracy alone is not enough. It has to be compared with the result of F1-score. If their results are near to each other, it means it performs well. In the case of Random Forest, for positive and neutral tweets, the results of F1 score are 92% and 87% respectively; with regarding to overall accuracy result which is 88% it means they are near in each other.

Also, in order to test the performance of the suggested classifier, the overall accuracy of Random Forest classifier with GNB and SVM have been compared as it shows in the table below,

Table 12. The accuracy comparison between the classifiers with their required time

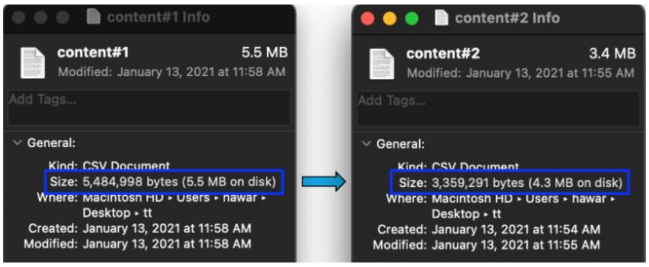
ML Classifier	**Accuracy**	**Required time**	
		Training	**Prediction**
Random Forest	88%	7min 5s	2.12 s
GNB	72%	1.64 s	367 ms
SVM	89%	15min 37s	1min 42s

The accuracy result shows 88%, 73%, and 89% respectively. From the comparison, it seems that the Ransom forest classier works better than GNB. So, the author is suggesting Random Forest over GNB, but in the case of Random Forest with SVM, we can notice one-degree difference in their accuracies, but SVM has the problem of time requiring. The table above which contains the required time for each classifier, in the used macOS system (the hardware specification is given in Table 2). The results show that GNB needs the least amount of time, while SVM needs the most amount of time, with is a huge difference from the two other classifiers.

Another way for testing the proposed system is to check the data size of the dataset before and after cleaning processes for information purpose. Reducing the size of the data, means the used cleaning method has worked well on the dataset; also, it causes and essential impact on the performance of the work, the less and more cleaned data, means the faster system is.

Thus, cleaning-out the noisy, or wrong samples in the original training dataset; is a very important step for the training dataset methods in enhancing the classification accuracy [28].

The figure below, shows the text before and after cleaning, in the macOS system, the differences can be noticed between both CSV files,



content#1 5.5 MB
Modified: January 13, 2021 at 11:58 AM

General:
Kind: CSV Document
Size: 5,484,998 bytes (5.5 MB on disk)
Where: Macintosh HD ▸ Users ▸ nawar ▸ Desktop ▸ tt
Created: January 13, 2021 at 11:58 AM
Modified: January 13, 2021 at 11:58 AM

content#2 3.4 MB
Modified: January 13, 2021 at 11:55 AM

General:
Kind: CSV Document
Size: 3,359,291 bytes (4.3 MB on disk)
Where: Macintosh HD ▸ Users ▸ nawar ▸ Desktop ▸ tt
Created: January 13, 2021 at 11:54 AM
Modified: January 13, 2021 at 11:55 AM

Figure 12. Both saved "content" with "cleaned content" features in macOS hard drive

One of the benefits of cleaning textual data is reducing capacity. We succeeded to decrease the size of our dataset about 1.2 MB (from 5.5 MB to 4.3 MB).

This decreasing of size leads to apply the ML algorithms faster. For example, in the case of Random Forest, the training and testing technique on the macOS took 7min 5s, and 2.12 s respectively.

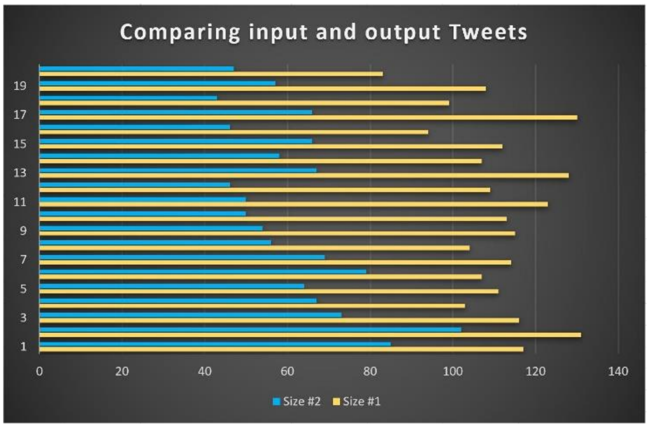


Figure 13. Comparing the sizes of In/Out texts

This decreasing of size leads to apply the ML algorithms faster. For example, in the case of Random Forest, the training and testing technique on the macOS took about 7min. This is useful in those situations that have a dataset with a huge amount of information and a small with a determined amount of capacity.

CONCLUSION

In the proposed research paper, sentiment analysis as the use of use NLP and ML classifiers have been applied on Trump's tweet dataset. After data preparation, the most important sentiment analysis procedures have been applied on the host dataset, like cleaning the dataset in order to be ready for text vectorization. Cleaning the dataset, which includes removing stopwords, word lemmatization, regular expression and tokenization. We succeeded in reducing the size of content feature with the target of taking fewer capacity. With the rapid growth of social media networks, it became a challenged task to know the subjective polarities of tweets. Therefore, we judged each sentence regarding their polarities whether they are *positive*, *negative* or *neutral*. At the end, the accuracies of Random Forest classifier with both GNB and SVM have been compared. Other related results to the confusion matrix and classification report tables have been given in the result and dissection section.

The author is suggesting Random Forest over other two classifiers, which are GNB, and SVM, since Random Forest has a good percent of accuracy and need less time compared to SVM.

For the future studies, detecting text polarities can be classified into 7 levels (strong, moderate, weak) each with 0.25 degrees of threshold. Also, we recommend the same system for not only on texts, but for speech recognition and cleaning noisy data in practical AI and Robotics.
The proposed method can be used in AI industries, and applied linguistics.

REFERENCES

[1] Duncombe, Constance. "The politics of Twitter: emotions and the power of social media." International Political Sociology 13.4 (2019): 409-429.

[2] Akram, Waseem, and Rekesh Kumar. "A study on positive and negative effects of social media on society." International Journal of Computer Sciences and Engineering 5.10 (2017): 347-354.

[3] Ajjoub, Carl, Thomas Walker, and Yunfei Zhao. "Social media posts and stock returns: The Trump factor." International Journal of Managerial Finance (2020).

[4] Social Blade Organization, "Twitter Stats Summary," *User Statistics for RealDonalTrump*. https://socialblade.com/twitter/user/realdonaldtrump (accessed Dec. 7, 2020).

[5] Wells, Chris, et al. "Trump, Twitter, and news media responsiveness: A media systems approach." New Media & Society 22.4 (2020): 659-682.

[6] Clarke, Isobelle, and Jack Grieve. "Stylistic variation on the Donald Trump Twitter account: A linguistic analysis of tweets posted between 2009 and 2018." PloS one 14.9 (2019): e0222062.

[7] Yaqub, Ussama, et al. "Analysis of political discourse on twitter in the context of the 2016 US presidential elections." Government Information Quarterly 34.4 (2017): 613-626.

[8] Kaggle Data science Company, "Datasets," *Datasets*. https://www.kaggle.com/austinreese/trump-tweets (accessed Nov.7, 2020).

[9] Kam, Ho Tin. "Random decision forest." Proceedings of the 3rd International Conference on Document Analysis and Recognition. Vol. 1416. Montreal, Canada, August, 1995.

[10] Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32. Cutler, Adele, D. Richard Cutler, and John R. Stevens. "Random forests." Ensemble machine learning. Springer, Boston, MA, 2012. 157-175.

[11] Scikit-learn Software. https://scikit-learn.org/stable/modules/naive_bayes.html (accessed May 2, 2021)

[12] Syafie, Lukman, et al. "Comparison of Artificial Neural Network and Gaussian Naïve Bayes in Recognition of Hand-Writing Number." 2018 2nd East Indonesia Conference on Computer and Information Technology (EIConCIT). IEEE, 2018.

[13] Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." Machine learning 20.3 (1995): 273-297.

[14] Tharwat, Alaa. "Classification assessment methods." *Applied Computing and Informatics* (2020).

[15] Kulkarni, Ajay, Deri Chong, and Feras A. Batarseh. "Foundations of data imbalance and solutions for a data

democracy." Data Democracy. Academic Press, 2020. 83-106.

[16] Elbagir, Shihab, and Jing Yang. "Twitter sentiment analysis using natural language toolkit and VADER sentiment." Proceedings of the International MultiConference of Engineers and Computer Scientists. Vol. 122. 2019.

[17] Li, Irene, et al. "What Are We Depressed About When We Talk About COVID-19: Mental Health Analysis on Tweets Using Natural Language Processing." International Conference on Innovative Techniques and Applications of Artificial Intelligence. Springer, Cham, 2020.

[18] Al-Makhadmeh, Zafer, and Amr Tolba. "Automatic hate speech detection using killer natural language processing optimizing ensemble deep learning approach." Computing 102.2 (2020): 501-522.

[19] Vishwakarma, Dinesh Kumar, Deepika Varshney, and Ashima Yadav. "Detection and veracity analysis of fake news via scrapping and authenticating the web search." *Cognitive Systems Research* 58 (2019): 217-229.

[20] Back, Bong-Hyun, and Il-Kyu Ha. "Comparison of sentiment analysis from large Twitter datasets by Naïve Bayes and natural language processing methods." *Journal of information and communication convergence engineering* 17.4 (2019): 239-245.

[21] Jindal, Kanika, and Rajni Aron. "A systematic study of sentiment analysis for social media data." *Materials Today: Proceedings* (2021).

[22] Yaqub, Ussama. "Tweeting During the Covid-19 Pandemic: Sentiment Analysis of Twitter Messages by President Trump." *Digital Government: Research and Practice* 2.1 (2020): 1-7.

[23] Sahu, Kalyan, Yu Bai, and Yoonsuk Choi. "Supervised Sentiment Analysis of Twitter Handle of President Trump with Data Visualization Technique." *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2020.

[24] Asur, Sitaram, and Bernardo A. Huberman. "Predicting the future with social media." *2010 IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology*. Vol. 1. IEEE, 2010.

[25] Kaggle Data science Company, “Datasets,” *Datasets*. https://www.kaggle.com/austinreese/trump-tweets (accessed Nov.7, 2020).

[26] Ruz, Gonzalo A., Pablo A. Henríquez, and Aldo Mascareño. "Sentiment analysis of Twitter data during critical events through Bayesian networks classifiers." *Future Generation Computer Systems* 106 (2020): 92-104.

[27] Patro, V. M., and M. R. Patra. “A Novel Approach to Compute Confusion Matrix for Classification of N-Class Attributes with Feature Selection”. *Transactions on Machine Learning and Artificial Intelligence*, Vol. 3, no. 2, May 2015, p. 52, doi:10.14738/tmlai.32.1108.

[28] Wang, Yidi, Zhibin Pan, and Yiwei Pan. "A Training Data Set Cleaning Method by Classification Ability Ranking for the $ k $-Nearest Neighbor Classifier." IEEE transactions on neural networks and learning systems 31.5 (2019): 1544-1556.

[29] Deshmukh, Kamalakshi V., and Sankirti S. Shiravale. "Ambiguity Resolution in English Language for Sentiment Analysis." *2018 IEEE Punecon*. IEEE.

[30] Verma, M. Tech Scholar Rajat. "Natural Language Processing (Nlp): A Comprehensive Study." (2018).

[31] Vasiliev, Yuli. *Natural Language Processing with Python and SpaCy: A Practical Introduction*. No Starch Press, 2020.

[32] Jakkula, Vikramaditya. "Tutorial on support vector machine (svm)." *School of EECS, Washington State University* 37 (2006).