Target Sentiment Analysis Model with Naïve Bayes and Support Vector Machine for Product Review Classification

Target Sentiment Analysis Model with Naïve Bayes and Support Vector Machine for Product Review Classification

Rhoda Viviane A. Ogutu[1], Richard Rimiru[2] and Calvins Otieno[3]

***Abstract*—Sentiment analysis has demonstrated that the automation and computational recognition of sentiments is possible and evolving with time, due to factors such as; emergence of new technological trends and the continued dynamic state of the human language as a form of communication. Sentiment analysis is therefore an Information extraction task that aims at obtaining private sentiments that can either be classified as *positive* or *negative,* toward a specific object or subject. However, social media platforms are marred with informal texts that make extraction and parsing of relevant information a problem for most systems and models. This can pose as a challenge to business enterprises, individuals or organizations seeking to make specific strategic decisions based on the available data. To overcome such inefficiencies, this research first proposes implementation of two classifier models on the basis of feature selection and extraction; and performance evaluation on sentiment classification of product reviews. The research will explore the use of a detailed pre-processing technique with the implementation of Naïve Bayes and SVM classifiers. The effect in terms of performance measure of such computational models, evaluation of how the models can be implemented within Social Listening application fields and Machine Learning approaches to Sentiment analysis; has formed grounds for this research. This paper is however intended to further evaluate the performance of Naïve Bayes and Support Vector Machine (SVM) classifiers with an intension of integrating the two classifiers, and creating an ensemble model.**

***Index Terms*— Feature Extraction, Feature Selection, Machine Learning, Naïve Bayes, Opinion, Sentiment, Sentiment analysis, Sentiment classification, Social listening, Supervised Learning, SVM.**

I. Introduction

The rise in use and popularity of informal language, and the adoption social media platforms, especially Twitter has made Sentiment analysis of tweets an important area of research [1]for enterprises, while at the same time given web users a venue for expressing and sharing their thoughts, opinions or sentiments on all kind of topics and events. As expressed by [2] Twitter has millions of users worldwide that constantly tweet, making it a gold mine for communities, organizations as well as individuals to monitor their reputation, how people feel over time about their brands by extracting and analyzing the sentiment of tweets posted by the public about them, their market or competitors. These monitoring processes can be referred to as *Social Listening*. In addition, popularity of internet shopping has increased, and as such, reviews of almost any product or business exist and are monitored by the respective businesses. One way this is being accomplished is through sentiment analysis [2].

According to [3] these online social interactions can also be used to reveal individuals' or groups of individuals' behavior, or a community's dynamics to better understand public perception, by mining the digital traces left by users while interacting with cyber-physical space, such as microblogs, for profitable reasons. This has led to the emergence of research on Social and Community Intelligence that present opportunities to compile these digital footprints into a comprehensive picture of individuals'

Corresponding Author: [1]Rhoda Viviane A. Ogutu. (e-mail: achiengrh@gmail.com).
[2]Richard Rimiru. (e-mail: rimiru@jkuat.ac.ke).
[3]Calvins Otieno. (e-mail: cotieno@jkuat.ac.ke).

[1]School of Computing and Information Technology (SCIT) Jomo Kenyatta University of Agriculture and Technology – JKUAT (Mombasa Campus), P. O. Box 81310-80100, Mombasa.
[2,3]School of Computing and Information Technology (SCIT), Jomo Kenyatta University of Agriculture and Technology – JKUAT (Main Campus), P. O. Box 62000- 00200, Nairobi, Kenya.

daily life facets, transform the understanding of our needs, organizations and societies; while also enable innovative improvement on products, public safety, resource management and environmental monitoring.

Such information submitted to the online services are a form of data sources that can be used in Sentiment analysis. Models in Sentiment analysis over twitter data and other microblogs faces several new challenges due to the short length and irregular structure of textual data (high dimensions). These may include informal or colloquial content and use of various languages, voluminous data, among other challenges [4]; [5]; [6] and [7]. Consequently, a problem of focusing on the most relevant information from the voluminously complex data, during Sentiment Analysis, may arise. In addition, relevant feature extraction is significant for Sentiment classification as the opinionated texts may have high dimensions, which can affect classifier performance [8].

The goal of this research is to finally develop an ensemble machine learning model for accurate classification purposes so as to achieve and evaluate a better predictive model, as the research problem originates as a combination of a Machine Learning and Information Retrieval challenge. By combining these fields, a problem of extracting relevant features in Sentiment analysis exists.

II. Related Work

As indicated by [9] in the field of Sentiment analysis, an increased attention is now focused on analysis of social media content (Twitter for that matter). This is to facilitate the understanding of social aspects and measure confidence level of products; or the perceived image of a company. These social media contents are often in form of informal messages that are short and textual in nature. Thus, these messages bring in new challenges to Sentiment Analysis. They are limited in length, tend to have many misspellings errors, shortened form of words, over capitalization or an over use of numbers, the use of non-standard expressions such as 'gr8' instead of 'great'. They also have special markers such as hashtags and other characters [10]; [4] and [7]. These challenges can be referred to as high dimensions in social media textual data [11]. According to [12] along with these challenges, a practical sentiment classifier should be able to handle efficiently large workloads.

For this reason, in the field of Artificial Intelligence, Machine learning approaches have been widely applied for the automation of Sentiment analysis so as to provide computers with the ability to handle the large workloads, and also the ability to learn without being explicitly programmed, while at the same time improve efficiency for classifiers. With emphasis on text sentiment analysis, researches are mostly narrowed on feature selection or extraction, and analysis of the classifiers used in models. This has been evidenced in the reports of works done by various researchers.

According to [11] Sentiment analysis is an Information Extraction task that aims at obtaining a writer's feelings expressed in positive or negative comments by analyzing a large number of documents. It is therefore the computational technique for extracting, classifying, understanding and determining opinions expressed in various contents. Sentiment analysis attempts to identify a sentiment held towards an object and helps in the automation of extraction or classification of sentiment from unstructured text. Further, in their research, the researcher describe Sentiment analysis as an aim to determine the state of mind of a speaker/ writer with respect to some topic or the overall tonality of a document.

As [13] clearly puts it in their research, there are challenges in sentiment analysis such as subjectivity classification, word sentiment classification, document sentiment classification and opinion extraction. These challenges can be solved through various computational approaches for sentiment analysis such as Linguistic approaches and Machine learning approaches. Linguistic approach relies on disambiguation using background information such as a set of rules and vocabularies. Thus, such a system normally contains lexicons, which consist of words and their polarity values (positive/ negative, bad/good etc.). There are also sets of rules that help produce more accurate results as an integral part of such a system. The machine learning approaches however, are used for automatic sentiment classification and are approved by many researchers as the efficient way to analyze sentiment laden term in a document. The researchers also recommend that improving the quality of the system is an area for future work.

With this regard, Feature selection and extraction has been exhibited as an important area geared towards the improvement of quality and efficiency in Sentiment analysis by many researchers, and therefore this forms in part, the basis of this study. According to [14] employing various feature reduction and extraction techniques decreases the running time of learning while increasing success rate of algorithms.

[15] proposed a novel filter based probabilistic feature selection method that tried to answer the common question that users have when looking for new techniques to select distinctive features so as to result in improvement of classification accuracy and reduce processing time as well. As such, it is clear that to reach an optimal performance level, and improve efficiency of classifiers during analysis, it is advisable to include important features in prediction and extraction of Sentiment information. These important features can be referred to as relevant features.

[16] researched and proposed a model, using n-gram features, stemming and feature selection to overcome some Persian language challenges (such as the use of informal words) in Sentiment classification. The researchers acknowledged, according to their findings, that feature selection in Sentiment analysis can improve classifier performance. The proposed Modified version of Mutual Information (MMI) method considers all possible combinations of co-occurrences of a feature and class label, and is concluded to improve performance. However, it was also proven through experiments that other feature selection approaches such as Document Frequency (DF), Mutual Information (MI) and Term Frequency Variance (TFV) does not measure the co-occurrences of other features and classes.

[17] Acknowledges that Feature engineering is a very important task in the domain of sentiment analysis and generally in text categorization, and converting original documents to feature vectors is critical.

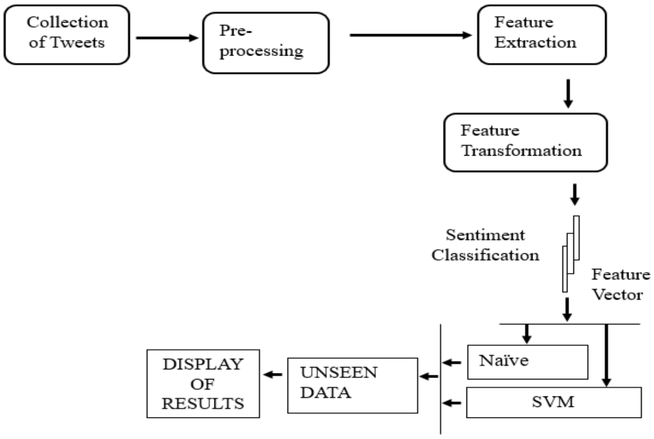


Fig. 1. Sentiment Classification Process

III. Proposed Work

The main purpose of the study is to create a Sentiment Analysis model using Naïve Bayes and Support Vector classifiers, while also evaluate the performance of the proposed classifiers, investigate their universal reliability and recommend how performance improvement can be achieved.

Experiments will also be devised with the aim of producing a Classification model that can categorize tweets (texts) into *positive* and *negative* classes based on supervised learning. In order to achieve a supportive infrastructure of the needs of the research, the following will be carried out. The above Fig. 1 shows the entire proposed system architecture.

A. Data Sources and Collection

The datasets are generated by collecting and mining tweets from Twitter (text documents) using a twitter API (Application Programming Interface). These tweets, are expected to address general topics touching on different product reviews and confidence level measure, while being categorized as *positive* and *negative*.

IV. SENTIMENT ANALYSIS PROCESS AND EXPERIMENT DESIGN.

The proposed system is a system of various phases of development and design.

A. Collection and Mining of Tweets (reviews)

Tweets are mined and collected from twitter by use of a Twitter API.

B. Pre-processing and Data preparation

Once data mining and collection is complete, only the *'text'* is stripped from the entire raw data, and cleaned through a clearly defined pre-processing technique.

According to [18], data pre-processing describes any type of processing methods performed on raw data to prepare it for another processing procedure, and analysis. Commonly used as a preliminary data mining practice, data pre-processing methods transforms the data into a format that can be easily and effectively used for the classification algorithms. These processes are done after the text is 'stripped' so as to enhance performances of the classifiers. As such, the procedure includes;

1) Removing RT retweet texts

This involves removal of retweets (RT) and user names for twitter users who are retweeting from the text.

2) Removing html links

This activity involves removal of *http* references and links from the text.

3) Removing Twitter users' names

This is done to remove user names of Twittersphere users'

4) Removal of punctuation marks

This activity is done to remove all the punctuation marks from the text.

5) Removal of all numbers

This activity is done to remove all the numbers from the text, such as 1, 2, 3, 4, 5, 6, 105 etc.

C. Creating a Bag-of-Word Corpus

The individual words inform of text are taken into account and a Bag-of-word corpus is created. Bag-of-word is a feature vector representation where each dimension of text corresponds to a feature. The assumption is that all features are independent given the class labels. In this model, texts are represented as a bag of its words, disregarding grammar, semantics, context and even word order but keeping multiplicity. The occurrence of each word is used as a feature for classifier training [19]. The Bag-of-word model learns a vocabulary from all of the documents, then models each document by counting the number of times each word appears, by implementing a Document Term Matrix (DTM) process [20]. For further Data preparation procedures, the following activities are carried out;

1) Transform the texts to lower case

This is a procedure that converts the entire text in the corpus to lower case for uniformity. This is because tweets are highly unstructured and reviewers may capture certain texts as capital letters.

2) *Removal of Stop words*

This is where commonly used words such as ‘me’, ‘a’, ‘the’, ‘who’, ‘them’, ‘shall’, ‘has’, ‘have’, among others, that are not meaningful for the analysis are removed. These words are referred to as Stop words.

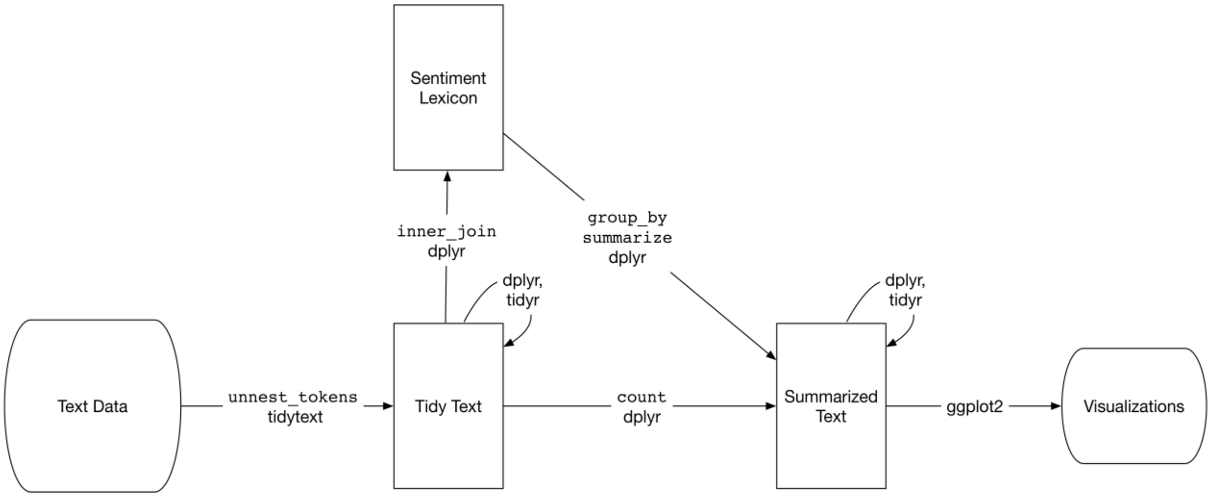


Fig. 2. A flowchart of a typical text analysis that uses tidy data principles [21]

3) *Strip white space*

There may be elements of white space in the corpus of text. This may be difficult for the analysis tool to comprehend, thus, it is important to strip the white spaces so as to once again enhance the classifier performance.

4) *Stemming*

Stemming is the process of reducing a word to its word stem (parent word) that affixes to suffixes and prefixes or to the roots of words known as a lemma. Stemming is part of information extraction (feature extraction), a process of linguistic normalization, in which the variant forms of a word are reduced to a common form.

D. Create a n-gram tokenizer function

According to [22] a n-gram is an ordered sequence of *n* “words” taken from a body of text. In natural language processing, tokenization is the process of breaking human-readable text into machine readable components. The most obvious way to tokenize a text is to split the text into words. The function allows us to specify unigram terms in the Term-Document Matrix. In addition, the rate of Sparsity is also determined. This is done by determining the least number of times a term appears in a document or entry.

E. Word Frequencies and Feature Vectors

Word Frequencies is a filtering and weighting process. The term-frequency is used as a counting function to return how many times the term *‘t’* or word, is present in the document. The most frequent words are then used as feature vectors. A feature vector is just a vector that contains information describing an object's important characteristics [23].

[16] Indicate that Feature Selection methods sort features on the basis of a numerical measure computed from the documents in the dataset collection, and select a subset of the features by thresholding that measure.

For the purpose of this study, we have employed Document Frequency (DF) feature selection method. [24] and [25] identifies Document Frequency (DF) as a filter method and one of the simplest approaches to assess feature relevance in text classification problems. The Document Frequency (DF) of a specific term or word simply corresponds to the number of documents in a class containing that word. Therefore, the Document Frequency (DF) of each term constitutes the relevancy score of the term.

F. Sentiment Analysis for the Tweets.

1) Lexicon Based Sentiment classification process

According to [26] and [8]. One approach to Sentiment analysis is to use a lexicon with information about which words are positive or negative. The lexicon is used to assign each word a sentiment (*positive* or *negative*). This method uses a variety of words annotated by polarity score, to decide the general assessment score of a given content. The lexicon are acquired automatically in RStudio environment (*bing* and *nrc* lexica) from the *tidytext* package in R (as shown in Fig.2).

The Sentiment Score data are then plotted on a data frame using *nrc* lexicon and *syuzhet* package in R so as to summarize the *nrc* values using the *nrc* dictionary, which provides the sentiment scores for each row of text.

2) Segregating Positive and Negative Tweets

We further sum up the emotions from the entire dataset into specific categories of *'positive'* and *'negative'*.

To achieve this, we carry out a text classification process which identifies the 'Most Common Positive' and 'Most Common Negative' sentiments using *bing* lexicon, *tidytext* and *inner_join()* function in R.

The *bing* lexicon from the *syuzhet* Package, as discussed earlier, categorizes words in a binary fashion into positive and negative categories. This is done to find a sentiment score for each word using the lexicon, then count the number of *'positive'* and *'negative'* words in the dataset.

bing was originally created to evaluate the sentiment of social media (twitter data, reviews, forum discussions, and blogs). One way to analyze the sentiment of a text is to consider the text as a combination of its individual words and the sentiment content of the whole text as the sum of the sentiment content of the individual words. This is an often-used approach; an approach that naturally takes advantage of the *tidy* tool in R.

Using *tidy* data principles *(tidytext)* make text mining tasks easier, more effective, and consistent with tools already in R. The infrastructure needed for text mining with tidy data frames already exists in packages like *dplyr, tidyr* and *ggplot2*. To work with the data as a tidy dataset, it is restructured as one-token-per-row format. This function uses the tokenizer's package to separate each line into words.

V. MODEL CREATION

It is evident from most Machine learning researches that every Machine learning algorithm or classification model has its own benefits and drawbacks. There is no solution or one approach that can fit all Machine learning problems. As such, different algorithms have been developed to solve different problem task. Several factors can affect a researcher's choice or selection of an algorithm for a model. For instance, selection of a classification model may mostly be made on the basis of factors such as resources available, accuracy requirement, and training time available among other factors [27]. In addition, [28] also poses a fundamental question in there research; "how should we assess the performance of algorithms on problems so that we may programmatically compare those algorithms?"

According to researches done by [29]; and [28] each classification algorithm has its inherent biases, and no single classification model enjoys superiority if no assumptions are made about the task. The researchers further suggest that, it is essential to first decide upon a metric to measure performance, then, compare at least a handful of different algorithms in order to train and select a best performing model. The commonly used metric is classification accuracy, which is defined as the proportion of correctly classified instances.

A. Naïve Bayes Classifier

For the purpose of implementation of the classifier in this study, our main focus is the Multinomial Naïve Bayes. Multinomial Naïve Bayes classifiers is mostly used in text classification and ultimately makes two simplifying independence assumptions [30] and [31].

- *Bag-of-words assumption:* where it is assumed that the position of words does not matter.
- *Conditional independence assumption:* where it is assumed that the probabilities of features *P(fi|c)* are independent given the class *c*.

The intuition of Multinomial Naïve Bayes algorithm is that text documents are represented as if they were a bag-of-words. This means that unordered set of words with their positions are ignored, while keeping only their frequency in the document.

Given a Hypothesis (H) and evidence (E), Bayes' Theorem states that the relationship between the probability of the hypothesis before getting the evidence, P(H), and the probability of the hypothesis after getting the evidence, P(H|E), is :

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}$$

Bayes Rule

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)} \tag{1}$$

To apply the Naïve Bayes classifier to text, we consider word positions, by simply introducing an index through every word position in the document as follows;

Positions ← all word positions in the test document

$$C_{NB} = argmaxP(c) \cdot \prod_{i\in positions} P(w_i|c) \tag{2}$$

Naïve Bayes calculations are done in log space. This is to avoid underflow and increased speed, thus (2) is generally instead expressed as the final equation;

$$C_{NB} = argmax\ logP(c) + \ \Sigma_{i\in positions}\ logP(w_i|c) \tag{3}$$

1) Training the Multinomial Naïve Bayes Classifier

To learn the probabilities of *P(c)* and P *P(f_i|c)* , that is, the prior probability of a given class *'c'* and the probability likelihood of a given feature fi given a class *'c'*; we consider the maximum likelihood estimate by using frequencies in the data. For *P(c)*, we identify what percentage of documents in the training set are in each class *'c'*. Let N_c be the number of documents in our training data with class *'c'*, and N_{doc} be the total number of documents. Thus;

$$P(c) = \frac{N_c}{N_{doc}} \tag{4}$$

To learn the probability *P(f$_i$|c)*, we assume a feature is the existence of a word in the document's bag-of-words, and so we have *P(w$_i$|c)*, which is computed as a fraction times the word w_i appears among all words in all the documents of class *'c'*. We first sum up all documents with category c into one big *"category c"* text. We then use the frequency of w_i in this summed up document to establish (by counting) a maximum likelihood estimate of the probability.

$$P(w_i|c) = \frac{count(w_i|c)}{\Sigma_{w \in V} count(w, c)} \tag{5}$$

A vocabulary 'V' of the document which consist of all the words in all classes, and not just the words in one class c, is created.

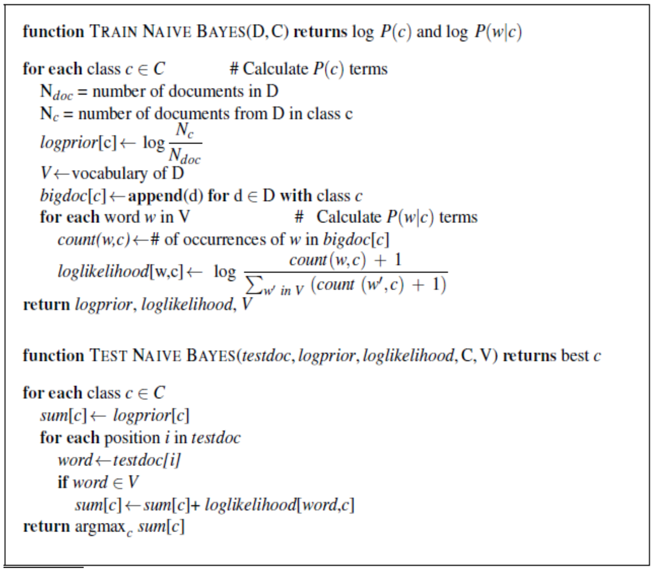
2) Laplace Smoothing (add-1)

According to [30] and [32] Naïve Bayes Classifier has a problem with maximum likelihood training. For instance, the problem of *'unknown word'* particularly where if a feature (or word) does not occur in any document in the training set, all documents in the test set that contain this same feature will be zero for all classes *'c'*, causing Multinomial Naïve Bayes to lose all discriminative power. In addition, rarely occurring features may also be problematic if smoothing is not performed. For example, a rare feature that may occur in some classes in the training set but does not occur in the test set will dominate probability estimates since it will force *P(w$_i$|c)* to be zero, regardless of the values of the remaining word features.

For instance, when trying to estimate the likelihood of the word "great" given class a positive, but there may be no training documents that contain the word "great" and are classified as positive. The word "great" may have occurred sarcastically in the class negative. In such a scenario the probability for this word feature will be zero as shown in (6) below.

$$P(\text{great}|positive) = \frac{count(\text{great}|positive)}{\Sigma_{w \in V} count(w, positive)} = 0 \tag{6}$$

Since Naïve Bayes naively multiplies all the feature likelihoods together, zero probability in the likelihood word for any class will cause the probability of the class to be zero, despite the evidence.



```
function TRAIN NAIVE BAYES(D, C) returns log $P(c)$ and log $P(w|c)$

for each class $c \in C$          # Calculate $P(c)$ terms
   $N_{doc}$ = number of documents in D
   $N_c$ = number of documents from D in class c
   logprior[c] ← log $\frac{N_c}{N_{doc}}$
   V ← vocabulary of D
   bigdoc[c] ← append(d) for d ∈ D with class c
   for each word w in V                # Calculate $P(w|c)$ terms
      count(w,c) ← # of occurrences of w in bigdoc[c]
      loglikelihood[w,c] ← log $\frac{count(w,c) + 1}{\sum_{w' \text{ in } V} (count(w',c) + 1)}$
return logprior, loglikelihood, V

function TEST NAIVE BAYES(testdoc, logprior, loglikelihood, C, V) returns best c

for each class $c \in C$
   sum[c] ← logprior[c]
   for each position i in testdoc
      word ← testdoc[i]
      if word ∈ V
         sum[c] ← sum[c]+ loglikelihood[word,c]
return $\text{argmax}_c$ sum[c]
```

Fig. 3. The Naïve Bayes algorithm, using Laplace Smoothing [30]

The solution to these limitations is parameter smoothing, and for the purpose of this study, we apply Laplace smoothing (add-1 smoothing) to prevent cases where missing, unknown or rarely occurring features inappropriately dominate the probability estimates in Multinomial Naïve Bayes. This is commonly used in Naïve Bayes text categorization:

$$P(w_i|c) = \frac{count(w_i, c) + 1}{\Sigma_{w \in V}(count(w, c) + 1} = \frac{count(w_i, c) + 1}{(\Sigma_{w \in V} count(w, c)) + |V|} \tag{7}$$

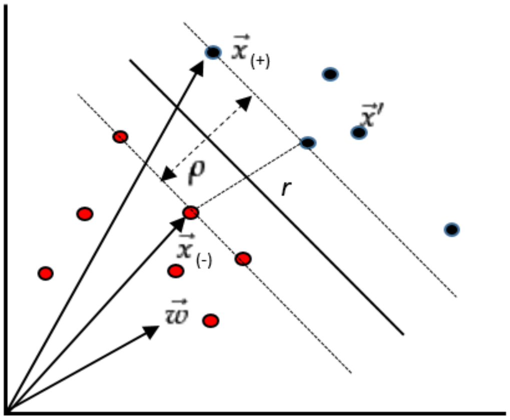


Fig. 4. Geometric margin of a point (r) and a decision boundary width ρ

B. Support Vector Machine (SVM) Classifier

There is considerable belief according to many researchers such as [33] that support vector machines provide one of the best models for predicting textual information. For instance, SVM's provide strong responses to high-dimensional input spaces, which is the case with text analysis. Also, SVM's deals well with the fact that document vectors are sparse. The goal of the Support

Vector Machine (SVM) is to find a hyperplane that best separates our two possible independent categorical variables (*positive* class and *negative* class), the classification problem. Intuitively, the model tries to find a decision boundary that can 'best' (by maximizing the geometric margin) split the data values based on the potential target value of our 'positive' and 'negative' classes, as shown in Fig. 4 above.

According to [34] Maximizing the margin is good for SVM because the support vectors (data points) near the decision boundary represents uncertain classification decisions, meaning there is almost a 50% chance of the classifier deciding either way. An SVM classifier with a large margin makes certain or good classification decisions, this is because a slight error in measurement or a slight document variation will not cause a misclassification. This can also be referred to as, a classification safety margin.

1) The Classification concept with Support Vector Machine (SVM)

A decision hyperplane as shown in Fig. 4 above, can be defined by an intercept term *b* and a decision hyperplane normal vector $\vec{w}$ which is perpendicular to the hyperplane. To choose among all the possible hyperplanes that are perpendicular to the normal vector, we specify the intercept term *b*. We then have a set of training data points $D = \{ (\vec{x_i}, y_i) \}$, where each member is a pair of a point $\vec{x_i}$ and a class label y_i corresponding to it.

We then use a training dataset of *n* points, such that; $(\vec{x_1}, y_1), \ldots, (\vec{x_n}, y_n)$, a normal vector $\vec{w}$ to the plane and some unknown data point on the plane point $\vec{x}$. Our main interest is knowing whether the unknown data point is on the *positive* class or the *negative* class category. At the same time we need to also employ a constraint (cost) to control and regulate the degree of miss classification. When $C = -b;$ we therefore find;

$$\vec{w} \cdot \vec{x} = C \tag{8}$$

$$\vec{w} \cdot \vec{x} + b \geq 0 \tag{9}$$

Without loss of generality if (9) is true, then we can make a decision rule that the unknown is a *positive* sample. In addition, with known data points categorized as *positive* or *negative*, we further conclude that;

$$\vec{w} \cdot \overrightarrow{x_{(+)}} + b \geq 1 \tag{10}$$

Eq. 10. Any data point on or above this boundary is of *positive* class

$$\vec{w} \cdot \overrightarrow{x_{(-)}} + b \leq -1 \tag{11}$$

Eq. 11. Any data point on or above this boundary is of the *negative* class

This means that there is a separation of distance of +1 or -1 for all the data point samples. We then introduce a variable y_i, where we consider our linear SVM that separates two classes y_i= +1 for *positive* samples, y_i = -1 for *negative* samples.

Therefore, we find that;

$$y_i(\vec{w} \cdot \vec{x_i} + b) - 1 \geq 0 \tag{12}$$

We add a constraint to the equation and set that it will be equal to zero for samples that end up in the margin;

$$y_i(\vec{w} \cdot \vec{x_i} + b) - 1 = 0 \tag{13}$$

To maximize the geometric margin, we want to find $\vec{w}$ and constant *b* such that: $p = 2 / ||\vec{w}||$ is maximized.

To maximize $2\,/\,||\vec{w}||$, we are tasked with minimizing $||\vec{w}||$. Which in turn results to;

$$\frac{1}{2}||\text{w}||^2 \tag{14}$$

Given the constraint (13) (the equation that describe the boundary/ hyperplane) we are again tasked with finding an optimized function, which is subject to the given constraint. Optimization means finding the minimum and maximum value of our function [24].

According to [24] we can use the Lagrange Multiplier to find the extreme number of the function (maximum or minimum), subject to our constraint (13). This will result to a new expression which we can optimize. Achieving the maximum possible margin is the underlying goal of the SVM classifier. Maximization of the margin requires the minimization of the training error;

$$f(x) = \frac{1}{2}||\text{w}||^2 + C\sum_{i=1}^{N} e_i \tag{15}$$

In the (15) above, *C* is the user defined constant while *e* is the margin error. A margin error occurs if data belonging to a particular class are found on the wrong side of the hyperplane. Minimizing the cost is therefore a trade-off issue between a large margin and a small number of training margin errors. The solution of this optimization problem is obtained as:

$$f(x) = \sum_{i=1}^{N} \lambda\, y_i x_i \tag{16}$$

The (16) above, is the weighted average of the training features; where λ is the Lagrange Multiplier of the optimizing task, as y_i remains the class label. The values of λ are nonzero (+1 or -1) for all of the points lying outside the margin and which appear on either the *positive* or *negative* side of the classifier, and exactly zero for those that appear inside or on the margin (support vectors), resulting to an SVM classifier.

VI. Evaluation

The evaluation results of this study were evaluated based on various performance metrics as shown in Table II below. These performance criteria were chosen because they are commonly used in evaluation metrics of text classification researches [35]. However, with regard to the study, emphasis was made on accuracy, precision and robustness. Robustness is the ability of a model to cope with errors [36].

A. The Confusion Matrix

TABLE I
Confusion Matrix Table

	Actual Negative Class	**Actual Positive Class**
Predicted Negative Class	True Negative (TN)	False Negative (FN)
Predicted Positive Class	False Positive (FP)	True Positive (TP)

Our study is a binary classification problem in which sentiments are classifies as either *positive* or *negative*. According to [37] the discrimination evaluation of the optimal solution during a classification training can be defined based on confusion matrix as in the Table I above. The confusion matrix is used for determining the correctness and accuracy of the model. The rows of the confusion matrix table represents the predicted class, while the column represents the actual class.

True Negatives (TN): The cases in which we predicted Negative and the actual output was also negative.

True Positive (TP): The cases in which we predicted Positive and the actual class of the data point was also positive.

False Negative (FN): False Negatives are the cases in which the actual class of the data point was positive and the predicted is Negative. False is because the model has predicted incorrectly and negative because the class predicted was a negative.

False Positive (FP): False Positives are the cases in which the actual class of the data point was Negative and the predicted is Positive. False is because the model has predicted incorrectly and positive because the class predicted was positive.

TABLE II
PERFORMANCE METRICS FOR MODEL EVALUATION

Metrics	Formula	Evaluation
Accuracy *(acc)*	$\frac{tp + tn}{tp + fp + tn + fn}$	The accuracy metric measures the ratio of correct predictions over the total number of instances evaluated
Error rate *(err)*	$\frac{fp + fn}{tp + fp + tn + fn}$	The metric is a misclassification error measure that measures the ratio of incorrect predictions over the total number of instances evaluated
Sensitivity *(sn)*	$\frac{tp}{tp + fn}$	The metric measures the fraction of positive patterns that are correctly classified
Specificity *(sp)*	$\frac{tn}{tn + fp}$	The metric is used to measure the fraction of negative patterns that are correctly classified
Precision *(p)*	$\frac{tp}{tp + fp}$	Precision is used to measure the positive patterns that are correctly predicted from the total predicted patterns in a positive class
Recall *(r)*	$\frac{tp}{tp + tn}$	Recall is used to measure the fraction of positive patterns that are correctly classified
F-Measure (FM)	$\frac{2 * p * r}{p + r}$	This metric represents the harmonic mean between recall and precision values

VII. RESULTS AND DISCUSSION

For the purposes of carrying out our experiments in the study, we mined and collected a data set of tweet reviews of "OnePlus 7 Pro" mobile phone. We then aimed at observing the impact of using a detailed pre-processing and data preparation technique, while at the same time observing the impact of using Document Frequency (DF) feature selection method as a filter technique to assess feature relevance in our tweet sentiment classification problem. The data was categorized as *positive* and *negative*.

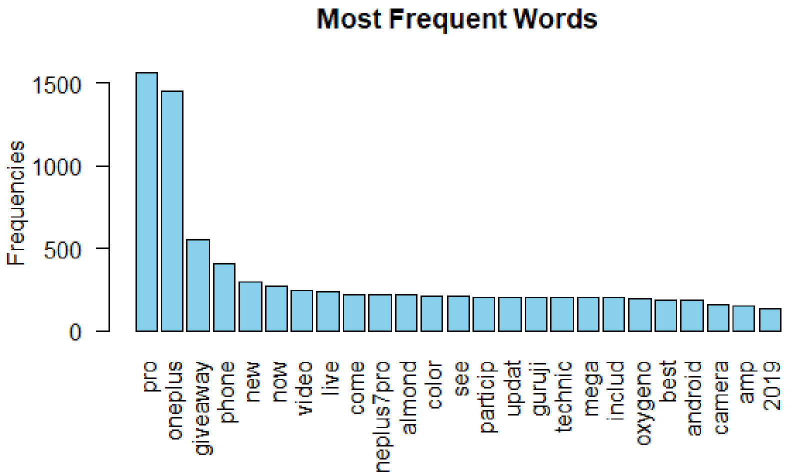


Fig. 5. A Document Frequency indicating the most frequent words from 2000 reviews

Fig. 5 above indicates a subset of high frequency features that were used for model training with Naïve Bayes and Support Vector Machine classifiers.

A. Naïve Bayes and SVM Classifiers Comparative Results

TABLE III
COMPARATIVE RESULTS FOR SENTIMENT CLASSIFICATION AND CLASSIFIER PERFORMANCE

Sr. No.	No. of Reviews	Accuracy		Sr. No.	No. of Reviews	Precision	
		Naïve Bayes	SVM			Naïve Bayes	SVM
1	500	89.10%	90.80%	1	500	87.50%	88.70%
2	700	94%	95.70%	2	700	92.90%	94.60%
3	1000	95.80%	93.50%	3	1000	94.60%	92.50%
4	1500	97.90%	95.80%	4	1500	97.40%	95.60%
5	2000	98.40%	97.50%	5	2000	97.90%	97.70%
6	3000	93.8%	97.6%	6	3000	93.1%	97.4%

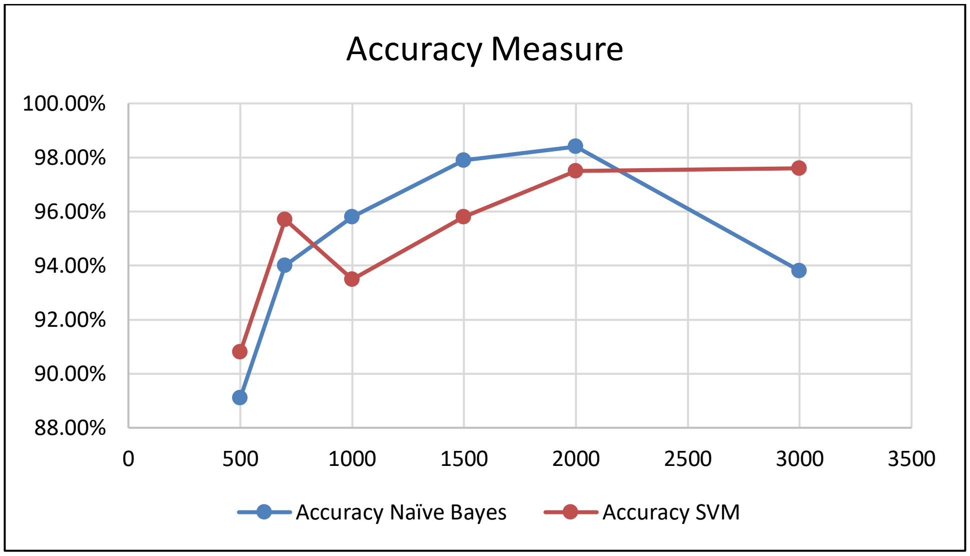


Fig. 6. Diagrammatic presentation of Accuracies in the experiments

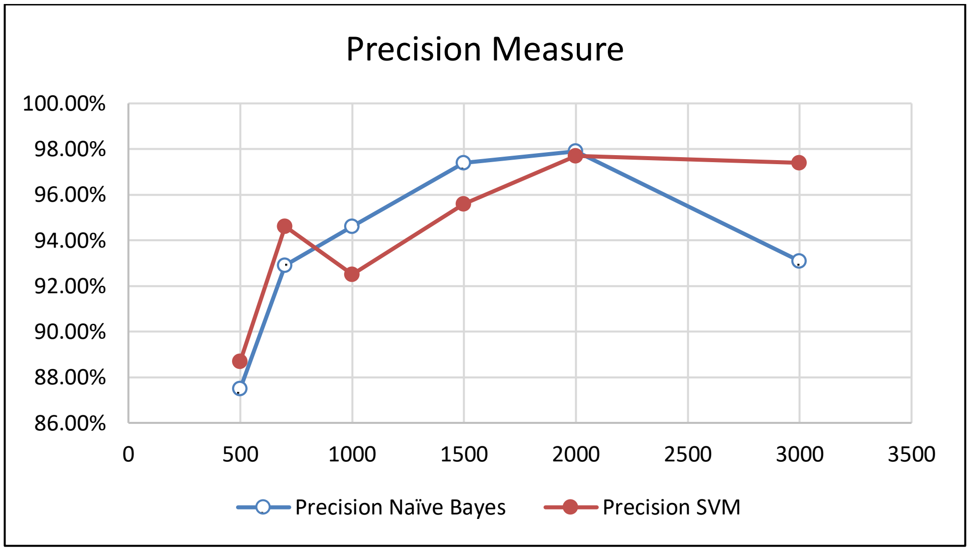
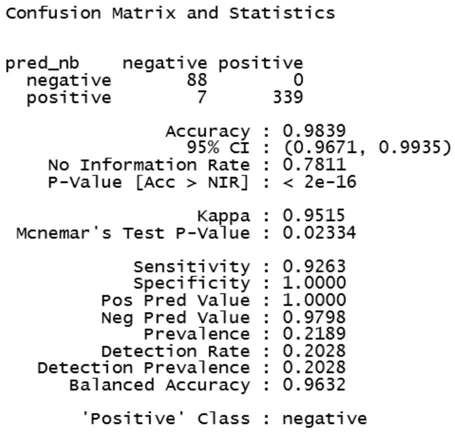


Fig. 7. Diagrammatic presentation of precision in the experiments

B. *Confusion Matrix Statistics*

TABLE V
CONFUSION MATRIX TABLE FOR SVM (2000 REVIEWS)

	Actual Negative Class	**Actual Positive Class**
Predicted Negative Class	95	3
Predicted Positive Class	8	338



```
Confusion Matrix and Statistics

pred_nb    negative positive
  negative       88        0
  positive        7      339

               Accuracy : 0.9839
                 95% CI : (0.9671, 0.9935)
    No Information Rate : 0.7811
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.9515
 Mcnemar's Test P-Value : 0.02334

            Sensitivity : 0.9263
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.9798
             Prevalence : 0.2189
         Detection Rate : 0.2028
   Detection Prevalence : 0.2028
      Balanced Accuracy : 0.9632

       'Positive' Class : negative
```

Fig. 8. A Confusion Matrix Statistics for Naïve Bayes for 2000 reviews

While comparing the two classification approaches, with respect to our experiments, Naïve Bayes gave a progressive improvement in performance with larger datasets for accuracies and precision measures. Generally speaking, Naïve Bayes approach and SVM approach achieved good performances, with a minimum performance record of 87.5%.

With the ‘Cost’ parameter, we were able to develop a robust model with SVM classifier, as we expected our model to be robust against the outliers (controlled by regulating the parameter ‘C’ during model training). The ‘Cost’ parameter can essentially be referred to as the penalty associated with making an error, or more arguably, the tolerance for errors is more or less accentuated with the Cost parameter. The higher the value of ‘C’ during classifier training, the less likely it is that a misclassification will occur for SVM classifier.

A 10-fold cross validation was also carried out during SVM classifier training with an intension of tuning and developing the ‘best’ model; a model with minimum amount of error rate.

VIII. Conclusion.

The goal of the study was to evaluate the performance for Sentiment classification in terms of accuracy, precision and robustness. We therefore compared two supervised Machine Learning algorithms, Naïve Bayes and Support Vector Machine (SVM) for sentiment classification of Twitter product reviews. The experimental results have indicated Naïve Bayes to have a better performance when compared to SVM with up to 2000 reviews. However, with 3000 reviews, SVM makes a surprising improvement! With this outcome, we propose to further carryout studies and develop an Ensemble model of the two classifiers, which will be discussed in our next paper.

IX. References

[1] S. Vosoughi, H. Zhou and D. Roy, "Enhanced Twitter Sentiment Classification Using Contextual Information," *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis,* pp. 1-10, 2015.

[2] H. Saif, Y. He and H. Alani, "Semantic Sentiment Analysis of Twitter," in *International Semantic Web Conference*, 2012.

[3] D. Yang, D. Zhang, Z. Yu, Z. Yu and D. Zeghlache, SESAME: Mining User Digital Footprints for Fine-Grained Preference-Aware Social Media Search, ACM Transactions on Internet Technology, 2014.

[4] S. Kiritchenko, X. Zhu and S. M. Mohammad, "Sentiment Analysis of Short Informal Texts," *Journal of Artificial Intelligence Research 50*, pp. 723-762, 2014.

[5] Chandni, N. Chandra, S. Gupta and R. Pahade, "Sentiment Analysis and its Challenges," *International Journal of Engineering Research & Technology (IJERT),* pp. 968-970, 2015.

[6] A. G. Shirbhate and S. N. Deshmukh, "Feature Extraction for Sentiment Classification on Twitter Data," *International Journal of Science and Research (IJSR),* pp. 2183-2189, 2016.

[7] P. B. Awachate and V. P. Kshirsagar, "Improved Twitter Sentiment Analysis Using N Gram Feature Selection and Combinations," *International Journal of Advanced Research in Computer and Communication Engineering,* pp. 154-157, 2016.

[8] A. Sharma and S. Dey, "A comparative study of feature selection and machine learning techniques for sentiment analysis," Proceedings of the 2012 ACM-SIGAPP Research in Applied Computation Symposium, San Antonio, Texas, 1-7., 2012.

[9] S. L. Lo, E. Cambria, R. Chiong and D. Cornforth, "Multilingual Sentiment Analysis: From Formal to Informal and Scarce Resource Languages," *School of Design, Communication and Information Technology, The University of Newcastle,Callaghan, NSW 2308, Australia,* 2016.

[10] A. Agarwal, B. Xie, I. Vovsha, O. Rambow and R. Passonneau, "Sentiment Analysis of Twitter Data," Proceedings of the Workshop on Language in Social Media (LSM 2011), 2011.

[11] S. V. Gaikwad, A. Chaugule and P. Patil, "Text Mining Methods and Techniques," *International Journal of Computer Applications (0975 – 8887) Volume 85 – No 17,* 2014.

[12] E. S. Tellez, S. Miranda-Jimenez, M. Graff, D. Moctezuma, O. S. Siordia and E. A. Villasenor, A case study of Spanish text transformations for twitter sentiment analysis., 2017.

[13] P. Routray, C. K. Swain and S. P. Mishra, "A Survey on Sentiment Analysis," *International Journal of Computer Applications, Volume 76 – No.10,* p. 0975 – 8887, 2013.

[14] F. Akba, A. Ucan, E. A. Sezer and H. Sever, "Assessment of Feature Selection Metrics for Sentiment Analyses, Turkish Movie Reviews," in *Proceedings from European Conference Data Mining, International Conferences Intelligent Systems and Agents and Theory and Practice in Modern Computing*, 2014.

[15] K. A. Uysal and S. Gunal, "A novel probabilistic feature selection method for text classification.," *Knowledge-Based Systems 36:226-235 DOI: 10.1016/j.knosys.2012.06.005,* 2012.

[16] M. Saraee and A. Bagheri, "Feature Selection Methods in Persian Sentiment Analysis," in *International Conference on Application of Natural Language to Information Systems*, 2013.

[17] B. Sabri and S. Saad, "Arabic Sentiment Analysis with Optimal Combination of Features Selection and Machine Learning Approaches," *Research Journal of Applied Sciences, Engineering and Technology 13(5),* pp. 386-393, 2016.

[18] T. Iliou, C. Anagnostopoulos, M. Nerantzaki and G. Anastassopoulos, "A Novel Machine Learning Data Preprocessing Method for Enhancing Classification Algorithms Performance," in *Proceedings of the 16th International Conference on Engineering Applications of Neural Networks (INNS), ACM, September 2015.* , RHODES, GREECE, 2015.

[19] R. B. Basaveswar, B. V. R. Krishna, R. K. Gangadhara and K. Chandan, "An Enumerative Framework For Extraction Of Bag-Of-Words From Legal Documents," *Asian Journal of Computer Science And Information Technology 5:11,* p. 62 – 66, 2015.

[20] S. P. Narke, "A Study of Different Pre-ProcessingApproaches of Text Categorization," 2017.

[21] J. Silge and D. Robinson, "Text Mining with R : A Tidy Approach," 16 May 2019. [Online]. Available: https://www.tidytextmining.com/sentiment.html.

[22] D. Schmidt and C. Heckendorf, "Guide to the ngram Package - The R Project for Statistical Computing," 17 November 2017. [Online]. Available: https://cran.r-project.org/web/packages/ngram/vignettes/ngram-guide.pdf.

[23] J. M. Girard, "Research Gate," 1 January 2015. [Online]. Available: https://www.researchgate.net/post/in_simple_words_what_do_you_mean_by_feature_vector_in_image_processing.

[24] S. Gunal, "Hybrid feature selection for text classification," *Turk J Elec Eng & Comp Sci, Vol.20, No.Sup.2,* pp. 1296-1311, 2012.

[25] J. Yang, Y. Liu, Z. X, Z. Liu and X. Zhang, "A new feature selection based on comprehensive measurement both in inter-category and intra-category for text categorization.," *Information Processing and Management, Vol 48 Issue 4. Science Direct,* pp. 741-754, 2012.

[26] S. Schrauwen, "Machine Learning Approaches to Sentiment Analysis Using the Dutch Netlog Corpus," Computational Linguistics and Psycholinguistics Technical Report Series, CTRS-001, JULY 2010, 2010.

[27] A. Gupte, S. Joshi, P. Gadgul and A. Kadam, "Comparative Study of Classification Algorithms used in Sentiment Analysis," *(IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (5) , 2014, ,* pp. 6261-6264, 2014.

[28] D. Wolpert and W. Macready, Macready, W.G.: No Free Lunch Theorems for Optimization., Evolutionary Computation, IEEE Transactions on, 1997.

[29] D. H. Wolpert, The Supervised Learning No-Free-Lunch Theorems : Soft Computing and Industry, London: Springer, 1996.

[30] D. Jurafsky and J. H. Martin, Speech and Language Processing 3rd Edition., 2017.

[31] M. T. Khan, M. Durrani, A. Ali, I. Inayat, S. Khalid and K. H. Khan, "Sentiment analysis and the complex natural language," *Complex Adaptive Systems Moedling. SpringerOpen Journal,* pp. 1-19, 2016.

[32] A. Y. Liu and C. E. Martin, "Smoothing Multinomial Naïve Bayes in the Presence of Imbalance. Machine learning and data mining in pattern recognition.," in *7th international conference, MLDM 2011, New York, NY, USA, August 30 – September 3, 2011 Proceedings*, 2011.

[33] V. Elango and G. Narayanan, "Sentiment Analysis for Hotel Reviews.," in *Proceedings of the Tenth International Conference on Soft Computing and Pattern Recognition (SoCPaR 2018)*, 2014.

[34] C. D. Manning, P. Raghavan and H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008.

[35] P. Kalaivani and K. L. Shunmuganathan, "Sentiment Classification Of Movie Reviews By Supervised Machine Learning Approaches," *Indian Journal of Computer Science and Engineering (IJCSE) ISSN : 0976-5166 Vol. 4 No.4 Aug-Sep 2013,* pp. 285-292, 2013.

[36] J. Rodriguez, "The Three Pillars of Robust Machine Learning: Specification Testing, Robust Training and Formal Verification.," 29 March 2019. [Online]. Available: https://towardsdatascience.com/the-three-pillars-of-robust-machine-learning-specification-testing-robust-training-and-formal-51c1c6192f8.

[37] M. Hossin and M. N. Sulaiman, "A Review on Evaluation Metrics for Data Classification Evaluations.," *International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.5, No.2, March 2015*, 2015.