Improving Sentiment Analysis of Microblogs through Bagging of Ensemble Classifiers

Charles Brown Tinashe Dhliwayo

Submitted to the Institute of Graduate Studies and Research in partial fulfilment of the requirements for the degree of

Masters of Science
in
Computer Engineering

Eastern Mediterranean University
January 2020
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Ali Hakan Ulusoy
Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Hadi Işik Aybay
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Ekrem Varoğlu
Co-Supervisor

Assoc. Prof. Dr. Nazife Dimililer
Supervisor

Examining Committee

1. Assoc. Prof. Dr. Nazife Dimililer

2. Asst. Prof. Dr. Yiltan Bitirim

3. Asst. Prof. Dr. Fatma Tansu Hocanin

ABSTRACT

The growth of social media and micro-blogs has greatly shifted the dynamics of businesses and the way advertising is carried out. Micro-blogs have transformed the consumer from being mere shoppers to advertiser and reviewers. Micro-blog opinions have become the reflection of society's opinions, attitudes, and preferences at large hence the greater need to not only access data stemming from microblogs, but to be able to analyze the data and make predictions based on it, whether a product is seen in a positive light or negatively. This fierce battle for consumers 'attention has resulted in many corporations investing in data analysis to capture the market; consumers nowadays heavily rely on the opinions and reviews shared across microblogs in order to make a decision on products and services on offer. Thus, the need for organizations to be able to classify these reviews quickly and as proficiently as possible. However, the task of combing through millions of reviews to determine the sentiment of the feedback is humanly tasking henceforth a number of machine learning techniques to detect and perform binary classification – positive and negative- on reviews have already been proposed. However, the nature of the reviews of micro-blogs has resulted in classification increasingly becoming more complex with the usage of emoticons, slang and short phrase which we have dubbed as "social media language". Classifying such complex reviews or blog posts using simplistic single classifiers no longer suffices hence in this paper, we proposed an ensemble classifier-based approach to detect polarity of reviews. The proposed ensemble classifier uses 7 classifiers- Random Forest (RF), Support Vector Machine (SVM), Logistic Regression (LR), Naïve Bayes, K- Nearest Neighbors (KNN), Xgboost and Adaboost classifiers. The proposed technique is assessed on Pang et al.'s polarity dataset v1.0, Bo Pang and

Lillian Lee's 2004 ACL polarity dataset v2.0 and ACL's IMDb dataset. The evaluation results show that the proposed classifier provides better classification accuracy on both datasets than simple classifiers.

Keywords: Ensemble, Bagging, Sentiment Analysis, F1-Score, Accuracy, Classification.

ÖZ

Sosyal medya ve mikro blog kullanımının gittikçe artması, işletmelerin dinamiklerini ve reklamcılık konusundaki yaklaşımlarını büyük ölçüde değiştirmiştir. Tüketici artık sadece müşteri değil, aynı zamanda reklam ve yorum yapan konuma gelmiştir. Mikro-bloglarda ve sosyal medyada paylaşılan düşünceler ve yorumlar, toplumun görüşlerinin, tutumunun, tercihlerinin bir yansıması haline gelmiştir. Bu yüzden bu düşüncelerin ve yorumların doğru şekilde analiz edebilmesi ve Pazar tahminleri için kullanılabilmesi gerekmektedir. Bu durum bir çok şirketin tüketiciye ulaşmak ve Pazar payını artırmak için veri analizine yatırım yapmasına neden olmuştur. Bunlara ek olarak, tüketicilerin, herhangi bir ürünün satın alınmasıyla ilgili bir karar vermek için de mikro-bloglar arasında paylaşılan görüşlere ve incelemelere büyük ölçüde güvendikleri gözlemlenmiştir. Bu nedenle, kuruluşların bu düşünce ve yorumları mümkün olduğunca hızlı ve etkin bir şekilde sınıflandırabilmeleri gerekmektedir. Bu problemin çözümü için mikro-blog ve sosyal medya ortamlarındaki düşünceleri olumlu ve olumsuz olarak iki sınıfa ayırmak üzere bir çok sınıflandırıcı önerilmiştir. Bununla birlikte, sosyal medyada yapılan yorum ve paylaşılan düşüncelerin özellikleri sınıflandırmayı zorlaştırmakta ve tek bir sınıflandırıcı kullanmayı zorlaştırmaktadır. Bu nedenle, bu çalışmada, yorumların olumlu ve olumsuz olarak sınıflandırılması amacıyla sınıflandırıcı topluluğu tabanlı bir yaklaşım önerdik. Önerilen sınıflandırıcı topluluğunda Rastgele Orman (Random Forest), Destek Vektör Makinesi (Support Vector Machine), Lojistik Regresyon (Logistic Regression), Naïve Bayes, K- En Yakın Komşular (K-Nearest Neigbor), Xgboost ve Adaboost sınıflandırıcıları kullanılmıştır. Önerilen yöntem, IMDB etiketli duygu veri seti, Polarite veri seti v1.0 ve 2004 ACL polarite veri seti v2.0 da değerlendirilmiştir. Değerlendirme sonuçları,

önerilen sınıflandırıcının her iki veri setinde de basit sınıflandırma gruplarından daha iyi sınıflandırma doğruluğu (Accuracy) sağladığını göstermektedir.

Anahtar Kelimeler: Grup, Çuvallama, Duygu Analizi, F1-Skoru, Doğruluk, Sınıflandırma.

DEDICATION

This thesis is dedicated to my Lord Jesus Christ who has set me on the course I must follow. I dedicate this work to my family and to the field of Data Mining. I look forward to many more adventures in machine learning!

ACKNOWLEDGEMENT

Firstly, I would like to thank Prof. Dr. Mustafa Ilkan for the opportunity to further my studies as a Research Assistant in Information Technology department. To the Director of the School of Computing and Technology and my Supervisor, Assoc. Prof. Dr. Nazife Dimililer thank you for your tireless effort in helping met to produce this thesis. I'm grateful for the opportunity to study under your supervision. To my co-supervisor, Prof. Dr. Ekrem Varoglu thank you for the words of encouragement and the guidance to see out the thesis. I'm most grateful to you both.

To the jury members, Asst. Prof. Dr. Fatma Tansu Hocanin and Assist. Prof. Dr. Yiltan Bitirim thank you for your guidance and corrections; it helped bring this thesis together and certainly vast improvements were made since our defense because of your input.

To my colleagues, namely Mr. Hossein Ghaderi Zefrehi, Mr. Emmanual Joseph of Industrial Engineering, Ms. Ivy Jamabo of International Relations, and Mr. David Ogbemudia of Mechanical Engineering, I'm most grateful for your encouragement and help in bringing this thesis together. Mr. Housein thank you for all the advice and tutoring in machine learning; To Mr. Ogbemudia thank you for your assistance during the format checks.

Lastly but not the least, I most grateful to my family Mr. Charles Dhliwayo, Mrs. Rachael Dhliwayo, Ms. Prudence Dhliwayo for their support and encouragement in achieving this feat; and of course, to my Eastern Mediterranean University Believer's Love-world Family, I say thank you for your prayers and words of encouragement!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATONS

ACLIMDB	Association for Computational Linguistics Internet Movie Database
AdaBoost	Adaptive Boosting
AUC	Area Under the Curve
BOW	Bag of Words
FP	False Positive
FN	FALSE Negative
IDF	Inverse Document Frequency
IG	Information Gain
KNN	K-Nearest Neighbours
LR	Logistic Regression
LSA	Latent Semantic Analysis
MAE	Mean Absolute Error
ME	Maximum Entropy
MI	Mutual Information
MSE	Mean Squared Error (MSE):
NB	Naïve Bayes
POS	Part-of-Speech Tagging
RF	Random Forest
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TF	Term Frequency
TF-IDF	Term Frequency - Inverse Document Frequency
TN	True Negatives

TP	True Positive
XGBoost	Extreme Gradient Boosting

Chapter 1

INTRODUCTION

1.1 Overview

Sentiment analysis, sometimes referred to as opinion mining is one of the most actively researched areas of natural language processing and has become a desired study area in data mining, text mining, and web mining. Its rapidly increasing importance has coincided with the rise and dominance of social media in our society. Online forums, micro-blogs such as Twitter, Facebook, Reddit, Internet Movie Database (IMDB) are pertinent to people's ability to share their views and judgements on myriads of topics ranging from entertainment to politics. These opinions are central to the activities that can be undertaken by members of a society – from participating in charity campaigns to terrorism. Moreover, information extracted from these blogs and forums can greatly influence how a business is run, as blog recommendations are now the biggest advertising platforms for products and brands hence the rise in the number of studies of sentiment analysis.

Sentiment analysis is the automated process of analysing textual data and classifying opinion polarity into binary classes, negative and positive, or trinary classes which include a neutral polarity class alongside the binary options previously stated. Moreover, sentiment analysis can also be modelled to categorize subjectivity and objectivity of data. The ability to model sentiment analysis of data of various fields, from politics to entertainment, has given rise to the number of stakeholders in this

ever-growing field of text mining. Thus, the interest of more innovative and effective machine learning techniques to derive meaningful information from people's comments on forums and various microblogs.

In general, researchers have employed two main types of machine learning techniques for sentiment analysis, that is lexicon-based approach and machine learning algorithms. Machine learning can be categorized into supervised and unsupervised learning. Gautam and Yadav [1] states that the correct class labels are given with the dataset in supervised learning thus the classifier is trained to obtain outputs based on example input-output pairs from the train data. On the other hand, unsupervised learning entails that the training data is unlabelled hence processing is done through clustering where there are no target classes [2]. According to Feldman [3] sentiment analysis is carried out in three different levels such as document level, sentence level, and aspect level. Each level is briefly defined in the subsequent chapter.

1.2 Objectives of the Study

This thesis aims to address if in practice a bagged multiple classifier ensemble classification framework produces better performance results than single model classifiers and other ensemble classifier methods currently in use. This paper aims to:

i. To discuss existing machine learning techniques and their effectiveness in comparison to our proposed technique.
ii. To outline the framework of building our proposed bagged multi-classifier ensemble model.
iii. To evaluate the performance measures of bagged multiple classifiers in comparison to single modelled classifiers and existing ensemble classifier methods.

1.4 Significance of Study

The work proposed in this thesis improves the state of the art in sentiment analysis by using a novel idea as bagged ensemble of multiple classifiers.

1.5 Structure of the Thesis

The structure of this thesis is as follows: in chapter two, relevant studies on the subject matter are discussed under the literature review. In chapter three, relevant studies on the subject matter are discussed under the literature review. the research methodology is outlined and chapter four covers the datasets acquired and the results. In chapter five, more information will be provided on the findings and the conclusion of the paper will be presented.

Chapter 2

LITERATURE REVIEW

Several businesses have cited sentiment analysis as an oracle of customer satisfaction as it can be modelled to mine opinions, feelings and thoughts of a community concerning any subject or product. The applicability of data analysis and modelling functionality of sentiment analysis in various fields, from the financial sector to the health sector, has incited many research papers offering insight on the best model building practices in order to obtain optimum performances out of sentiment models in any particular area.

In their influential paper, Pang et al. [4] devised and compared human baselines to machine learning techniques to solve the problem of non-topic-based sentiment analysis. Their aim was to classify the overall sentiment of documents with machine learning techniques in comparison to human generated baselines. Furthermore, Pang et al. [4] conducted this study as a result of their differing opinions with some machine learning experts of their time, as the assumption was that human created baselines should be more accurate than machine learning systems used for text categorization. Furthermore, they were of the opinion that it was sufficient to produce a list of words that expressed strong sentiments by introspection and use them in text classification. In testing this hypothesis, two graduate students in computer science independently chose good indicator words for the positive and negative polarity of sentiments in movie reviews, and their responses were converted into decision procedures,

essentially counting the proposed number of positive and negative words in a document, which were applied to uniformly distributed data for a random choice baseline. The human based classifiers were recorded to produce accuracy measures of 58% and 64% on the polarity dataset version 0.9 [5]. On the other hand, using three standard machine learning algorithms – Naïve Bayes (NB), Maximum Entropy (ME), Support Vector Machines (SVM), they examined whether it sufficed to treat sentiment classification as a special case of topic-based categorization, with the documents consisting of two topics – the positive and negative sentiment, or whether special sentiment-categorization methods needed to be developed to match or surpass the results obtained from human baselines. Simple feature selection procedures such as unigrams and the combinations of unigrams and bigrams applied to NB, ME, and SVM were found to produce better accuracy levels than that of human baselines at 81.0 %, 80.4%, 82.9%. Respectively, when unigrams and bigrams were combined, they obtained accuracy levels of 80.6%, 80.8%, 82.7%. Moreover, Pang et al.[4] noted that ME feature/class functions only reflected the absence or presence of a feature rather than directly incorporating feature frequency, and in investigating the reliance on frequency information and its impact on accuracies of the classifiers, SVM had better performances than, ME when feature presence instead of feature frequency was accounted for both unigram and unigram-bigram hybrid models.

Given the findings of Pang et al. [4] 's work on the effect of unigrams on the accuracy of machine learning classifiers, Tripathy et al. [6] carried out further research on the effect of different n-grams on four machine learning algorithms, particularly NB, ME, SVM and Stochastic Gradient Descent (SGD). In this paper, it is observed that the usage of unigrams yields comparatively better accuracy measures than that of Part-of-

Speech (POS) tags. POS tags for words were found to be dynamic in accordance with the context of their use. For instance, the word “attack” could be rendered as a noun or a verb depending on the positioning of a word positioning within a sentence such as, “After my asthma attack, the doctor decided to attack the problem with new medication.” Thus, Tripathy et al. [6] proposed to implement words as a whole instead of using POS tags as parameters for classification. Moreover, the paper suggested two methods for the vectorization of text after data pre-processing, which were “CountVectorizer” and Term Frequency – Inverse Document Frequency (TF-IDF) to produce the numerical matrices required by the machine learning algorithms. These methods helped bypass the challenge of insufficient random-access memory when working with sparse matrices. Furthermore, those matrices were considered as input for four supervised machining learning algorithms – NB, ME, SGD and SVM. The application of Naïve Bayes method produced accuracy measures of 83.6%, 84.064%, 70.532%. Tripathy et al. [6] noted that Naïve Bayes is probabilistic therefore the features used for training NB classifiers were independent of each other hence unigram features performed better than trigram features. Trigrams were considered to affect the probability of the document because of the repetition of words which coincided with its comparatively low accuracy of 70.532%. In addition, the application of ME n-gram technique yielded accuracy measures of 88.48%, 83.228% and 71.38% for unigrams, bigrams and trigrams on the review dataset [7]. Additionally, SVM’s non-probabilistic nature was used to train models to find hyperplanes in order to separate the dataset, and this led to poorer results when bigrams and trigrams were applied to the Maximum Entropy method instead of unigrams as the plotting of multiple word combinations in a particular hyperplane, confused the classifier and thus provided less accurate results of bigrams and trigrams, that is 83.872% and 70.204% respectively,

in comparison with the value obtained using unigram, which was 86.976%. Lastly, the SGD method produced accuracy measures of 85.116% for unigrams, 95% for bigrams and 58.408% for trigrams. Tripathy et al. [6] reasoned that the randomness of word combination in bigram and trigrams added noise which reduced the accuracy value. Thus, any bigram and trigram model combination with other models affects the overall accuracy of that system.

Having observed the unresolved issue of sparsity from their previous studies [1,3], created when vectorizing text in order to input text data into machine learning, Pang and Lee 's work in [8] showcased the advantage of taking subjectivity as a key component of determining the sentiment of the overall content of a document. The results of their paper showed that the subjectivity extracts of the polarity dataset version 2.0 [9] compactly and accurately represented the sentiment information of the originating documents, achieving significant improvements in accuracy- from 82.8% to 86.4% and in the worst-case scenarios, subjectivity extracts maintained the same level of performance of the polarity classification tasks while retaining only 60% of the reviews' words.

Traditional approaches [4] focused on the selection of indicative lexical features and their frequency within a document when performing binary classification of documents. In contrast, Pang and Lee [8] proposed a method that labels sentences in a document as either objective or subjective, discarding the former and then applying two standard machine-learning classifiers, NB and SVM, to the resultant extract. This prevented polarity classifiers from using irrelevant text – text that's not indicative of the author's opinion for polarity classification. Rather than performing subjectivity detection on single sentences, Pang and Lee [8] proposed a method of classification

suggesting modelling proximity relationships between sentences to enable them to leverage the coherence of the text spans. The assumption was that neighboring text spans were likely to share the same subjectivity status rather than applying the standard classification algorithm on individual sentences to obtain the subjectivity. However, supplying the NB and SVM algorithms with the aforementioned pair-wise information convoluted the features; naturally the classifiers' input consists simply of individual feature vectors. In this paper [8] a more intuitive and efficient graph-based formulation that relies on finding minimum cuts is proposed in order to avoid the upheaval task of defining synthetic features to overcome the obstacle created by the use of pair-wise information in features. Moreover, both NB and SVM were trained on a subjectivity dataset [9] and then used to detect subjectivity. The ten-fold cross-validation performances of the former was slightly better on the subjectivity datasets – 92% versus 90%. Furthermore, it was observed that employing NB as a subjectivity detector in aggregation with a Naïve Bayes document-level polarity classifier produced an accuracy measure of 86.4%. This was a significant improvement over the 82% achieved in the case where the full review was used without performing any subjectivity extract. Additionally, the SVMs indicated a slight performance rise from 86.4% to 87.15% further cementing Pang and Lee [8]'s hypothesis that subjectivity extracts preserve the sentiment of the originating documents.

The previous studies [1, 3, 5], all used single classifiers to approach the problem of sentiment analysis, however Tsutsumi et al. [10] proposed a multiple classifier method to improve the performances in terms of accuracy over that of the usage of single classifiers. The method consisted of three classifiers based on Support Vector Machines, Maximum Entropy and Score Calculation. Alongside the classifiers, two

voting methods and another Support Vector Machine was applied to the integration process to produce a single classifier. The scoring method, based on a score calculation process of word polarity, was an expansion of a previous work by Osajima et al. [11]. The proposed technique [10] identified binary class polarity – positive and negative, of the review documents [9] based on the distances measured from the hyperplane of each classifier. However, the proposed method had an issue of determining the final output, namely positive or negative as the classifier's outputs had to be manually normalized because of differences in the scale of scoring between Support Vector Machines and that of the scoring method. Hence, the need for a third machine learning method, namely Maximum Entropy, which was applied with three different methods for the process of voting. Two voting procedures – Naïve voting and Weighted voting, were adopted and once more Support Vector Machines was used for the integration process of single classifiers. Here the features for the SVM were the outputs of the three-single classifier, namely the distances from the hyperplanes. Tsutusmi et al. [10] compared six methods in their experiment – single classifiers being SVMs, ME, and Scoring and the proposed method based on naïve voting, Weighted voting and SVM integration. Their proposed multiple classifier system outperformed the single classifiers as the integrated SVM method had an accuracy measure of 87.1%, the Weighted voting procedure had an accuracy of 86.4% , the naïve voting procedure had an accuracy of 85.8% all comparatively better than the single classifier performance of SVM, ME, and Scoring which had accuracy measures of 82.2%, 80.5% and 83.4% respectively.

Expanding on the previous study [10] of classifier combination, Li et al. [12] proposed classifier combination using multiple feature sets. In this method, different classifiers

were generated through training the review data [9] with different features – unigrams and some POS tag features before a classifier selection method was used to select a part of the classifiers for the next-stop combination. The selected classifiers were combined using five combination rules – sum, product, max, min and voting rule. The experimental results showed that all the combination approaches with different combination rules outperformed individual classifiers, with the sum rule achieving the best performances. In their experiment, six different types of features were used for six classifiers using the SVM method; these features were unigrams, adjectives, adjectives and adverbs, nouns, verb and adjectives, and verbs with adverbs. A ten-fold cross-validation procedure was performed on the dataset [9], with each fold they used 90% of the 700 positive reviews, 90% of the 700 negative reviews for training, and 10% for both positive and negative reviews for testing. As a result of the application of the aforementioned feature list, unigrams were noted to produce the best precision results at 80.44%, with adjectives, adjectives-adverbs combination, nouns, verb-adjective combination, verb-adverb combination respectively producing accuracy measures of 76.0%, 76.14%, 65.35%, 75.78%, and 73.29%. After obtaining the six different classifiers, they selected the best committee of the classifiers for further combination by applying *N*-best classifier selection method, and it was observed that the combination classifier performed best when *N* was three therefore three individual classifiers for the combination were selected, alongside the unigram, adjective-adverb, and adjectives feature sets. Finally, the previously stated combination rules were applied to the combination classifier producing precision measures of 83.00%, 82.71%,82.36%,81.43% for sum rule, product rule, max rule, min rule and voting rule respectively. Comparisons between the precision measures of the best individual

classifiers and that of the combination classifier clearly signified the combination classifier as being better.

Chapter 3

BACKGROUND

In this chapter, a background of the components of the proposed system architecture of our study such as sentiment analysis levels, machine learning algorithms, ensemble techniques, feature selection techniques and performance measures, are briefly discussed in the proceeding sections.

3.1 Sentimental Analysis

Sentiment analysis is a natural language process that identifies and extracts meaning from unstructured data. The application of sentiment analysis can be at differing granularities such as document level, sentence level and entity level. Each level is briefly defined in the subsequent section.

i. Document level sentiment analysis: In this level, the polarity for an entire document is classified; document level sentiment analysis considers a single review about a single topic at a time. However, documents extracted from blogs and forums tend to be made up of comparative topics and reviews thus document level sentiment classification is not always useful.

ii. Sentence level sentiment analysis: At this level, sentiment analysis occurs at a lower granularity than that of document level sentiment analysis as the querying of sentiment is done on single statements that make up a document rather than the entire document at once. Furthermore, sentence level sentiment

classification determines the polarity expressed in a sentence whether positive, negative, or neutral.

iii. Entity or Aspect level sentiment analysis: Sentiment analysis at entity or aspect level aims to classify the sentiment of a particular entity, such as a computer, or an aspect of an entity, such as the screen resolution of a computer. Consider a statement such as, “My HP computer has a wonderful screen resolution but a tacky hardware.” In the aforementioned statement, the entity would be “HP computer” and the aspects would be “screen resolution”, and “hardware” respectively. Furthermore, in entity level sentiment analysis, such a statement is processed deeply in search for the finer-grained analytical meaning in relation to the “HP computer”. On the other hand, aspect level sentiment analysis aims to identify the sentiment of one of the mentioned aspects, either the “screen resolution” or the “hardware”. The polarity of the latter aspect would be classified as being negative because of the adjective, “tacky”, that is used to describes it, whilst the former would be classified as positive because of the “wonderful” description tag.

3.2 Machine Learning Algorithms

Machine learning often referred to as predictive modelling, uses programming algorithms that receive and analyze input data to predict output values. In the process of analysis these algorithms learn and optimize their operations to improve performances. This study utilized machine learning algorithms for supervised classification. The supervised learning techniques applied in this study are briefly defined in the following section.

i. Naïve Bayes (NB) method: A probabilistic classifier based on Bayes' theorem with the independence assumptions between predictors. It is widely used for both training and classification. Its fundamental theory outlines that categories and joint probabilities of features are used in calculating the probability score of categories of a given document.

ii. K-Nearest Neighbour (KNN) method: As the name suggests, the variable *"K"* is used to identify unknown class samples; the algorithm inspects the *K* - closest instances in a training dataset and applies the soft computing of 7 predictions and computational intelligence. The usefulness of KNN algorithm hinges on the assumption that data points in close proximity are similar.

iii. Support Vector Machines (SVM) method: Introduced for binary classification, SVMs cater for both non-linear and linear classifications. The advantage of SVMs is that they are used to capture the best accessible surfaces for separating positive and negative training samples as datasets tend to be nonlinearly inseparable [13].

iv. Logistic Regression (LR) method: A linear classifier of probabilistic nature, that is best implemented when the dependent variable is binary. It has the ability to discriminate data and to elucidate the relationships between dependent binary variables and one or more of interval, ordinal, nominal or ratio-level independent variables.

v. Random Forest (RF) method: In this method, classifiers consist of a large number of distinct decision trees that operate as an ensemble. Furthermore, each individual tree that make up the "forest" spits out a class prediction and the class with the most votes becomes the model's prediction. This reduces the individual errors of single trees.

vi. Extreme Gradient Boosting (XGBoost) method: An augmented distributed gradient boosting algorithm from the XGBoost library. Friedman et al. [14] states that XGBoost used for boosting of parallel trees as it provides an efficient and scalable implementation of the gradient boosting framework.

vii. Adaptive Boosting (AdaBoost) method: Proposed by Freund and Schapire [15], AdaBoost is a boosting algorithm that combines multiple weak classifiers in order to form a strong classifier. These weak classifiers are made up of decision trees with single splits.

3.3 Feature Selection

Feature selection refers to the process of reducing the number of input variables used in a predictive model. This procedure is done by assessing the relationship between each input variable and the target variable using statistics; selections are based on those input variables that have the strongest relationship with the target variable. Feature selection techniques can either be categorized as filter based or wrapper methods. The latter, creates several models with different subsets of input features, selecting those features that result in the best performing models. However, filter-based feature selection methods assess the relationship between each input variable and the target variable, and the scores are used as the basis filter the input variables that will be used in the model. Some popular feature selection techniques are described in the subsequent section.

i. Chi- Square: A filter-based feature selection technique that is used to test the independence of two variables / features in order to determine the relationship between the independent categorical feature, the predictor, and the dependent categorical feature, the response. A higher chi-square value

suggests the feature is more dependent on the response and can be selected for training the model.

ii. Mutual information (MI): A filter-based feature selection method that determines the measure of mutual dependence between two random variables. The value of MI is always larger than or equal to zero; whereby the largeness of the value indicates the closeness of the relationship between the two variables. However, If the result is zero, then the variables are independent.

iii. Information Gain (IG): Information Gain measures the reduction in entropy by splitting the dataset according to a given random variable. The larger the IG the lower the entropy group. Entropy quantifies the amount of information there is in a random variable.

iv. Boruta Feature package: A wrapper feature selection method that finds the optimal combination of features through a repeated cycle of adding and/or removing predictors that will be used to build the model. First, it duplicates the dataset, and shuffles the values in each feature set / column. These values are called shadow features. Furthermore, a classifier such as Random Forest is trained, on the datasets [5,7,9]. By doing this the importance of each feature is determined - via the Mean Decrease Accuracy or Mean Decrease Impurity- for each of the features of the datasets. The higher the score, the better or more important a feature is. Then, the algorithm checks whether the feature has a higher Z-score than the maximum Z-score of its shadow features, these are the best of the shadow features. If they do, Boruta records this in a vector. At every iteration, the

algorithm tries to validate the importance of a feature by comparing it with random shuffled copies.

3.4 Feature Extraction / Engineering

i. Term Frequency-Inverse Document Frequency (TF-IDF): It is a statistical measure that assesses how relevant a word is to a document within a collection of documents by multiplying two metrics: Term Frequency (TF) and the Inverse Document Frequency (IDF). Term frequency refers to the number of times a word or a term appears in a document. On the other hand, IDF is a measure that is used to penalize the terms with the highest frequency count in a corpus of documents. The general idea being that a word that is frequent across corpus of documents is most likely not to be influential in prediction modelling. The formulae are shown in the subsequent section.

$$TF(t,d) = \frac{freq(t,d)}{\sum_i^n freq(t,d)} \quad (1)$$

Equation (1) states that term frequency is the proportion of the frequency of *t* terms in in document *d*.

$$IDF(t) = \log\frac{N}{count(t)} \quad (2)$$

Equation (2) shows that the *log* of the ratio of the document in proportion to the frequency of terns in that particular document gives the inverse document frequency. Having calculated the TF and IDF, the results are multiplied in order to obtain the TF-IDF as observed in equation (3).

$$TF - IDF\ (t,d) = \quad TF(t,d) * \quad IDF(t) \quad (3)$$

In general, equation (3) applies a Weighted scheme to the frequency count so as to normalize the frequency count. This is done in order to ensure the learning algorithm receives terms that contain the most relevant information for predicting the target variables.

ii. Bag of Words (BOW): Bag-of-words is a Natural Processing Language (NPL) approach used to represent text, such as in Table 3.1, as single multi-set of words known as unigrams that appear in the text.

Table 3.1: Training Set of Sample Text Documents

Text no.	**Text**
1	The food was terrible, I hated it.
2	The restaurant was very far away, I hated it.
3	The pasta was delicious, will come back again.

In order to convert Table 3.1's texts into BOWs, a vector of all words that appear in the entire set of text in the training set such as in Figure 3.1 is developed.

The | food | was | terrible | I | hated | it | restaurant | very | far | away | pasta | delicious | will | come | back | again

Figure 3.1: Unigrams Generated from Table 3.1

Furthermore, iterate each text in Table 3.1, marking a *"1"* and *"0"*, the former is indicated in the row vector corresponding to the word it contains.

Table 3.2: BOW visualization

BOW	**Text no.**	**Text Vector**
The | food | was | terrible | I | hated | it | restaurant | very | far | away | pasta | delicious | will | come | back | again	1	1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
	2	1 0 1 0 1 1 1 1 1 1 1 0 0 0 0 0 0
	3	1 0 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1

Table 3.2 shows a simplified. feature vector representation of the text, which can be used in tasks such as document classification whereby detecting the topic of a document based on the frequent words in it is paramount. Moreover, it can be used to find the similarity between sentences by comparing the two sentences' vector representations. In BOW model, the order of words is ignored.

iii. *N*-grams: This is an expansion on BOW; *N*-grams are a contiguous sequence of n words that take word-ordering in to account. N-grams tend to be more useful than BOW when selecting features as n-grams provide the context of a particular word, which helps produce optimum performances for predictive modelling.

Table 3.3: N-gram of Sample text

***N*- gram**	**Name**	**Text**	**Result**
1	Unigram	This is a sentence	This | is | a | sentence
2	Bigram		This is | is a | a sentence
3	Trigram		This is a | is a sentence

3.5 Vector Space Models (VSM)

Vector Space models (VSM) are algebraic models that represents text documents as vectors. VSMs can be used for information filtering, retrieval and indexing and can be applied to relevancy rankings [16]. VSM essentially represents the Bag of Words model; it can be visually conceptualized as a data frame by placing data into an array-like structure, with the rows and columns being the document and the latter being the 'terms' in question. Horizontally parsing across the data frame presents a collection

of numbers which mathematically could be represented as vectors as observed in Table 3.4

Table 3.4: Document Feature Matrix

Document	Frequency of Word Feature in Document	
	Lost	Flat
1	6	10
2	10	3
3	8	7

Table 3.4 shows the frequency count of two terms in three documents. The core intuition is that if documents can be represented as vectors of numbers then the hypothetical document term frequency matrix can be viewed geometrically in the vector space as in Figure 3.2. Given that the above document corpus contains only two terms, the document is visualized in a two-dimensional plane with each feature representing a plane in the vector space.

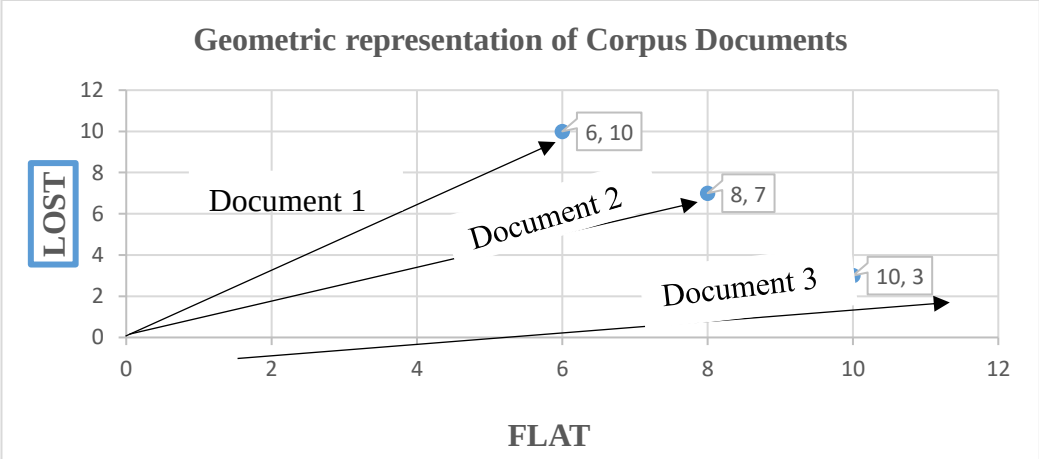

Figure 3.2: Geometric Representation of Sample Corpus Documents

As observed in the diagram above, the assumption is that all document vectors originate from the origin (0,0) and each document is plotted in the vector space. By intuition the geometric representation suggests that the closeness of lines to each other

illustrates the similarity between documents. Intuitively looking at the diagram above, document 1 is more similar to document 2 more than it is to document 3. This is proved mathematically through the usage of trigonometric functions to analyze the angles between the documents in order to make an interpretation about the documents by obtaining the higher-level contextual meaning of grouped similar terms. Furthermore, there are several mathematical methodologies that can be used to solve the underlying correlation as explained in the subsequent sections.

3.5.1 Cosine Similarity

As mentioned in [17], cosine similarity is a metric used to determine how similar documents are irrespective of their size. Mathematically, it measures the cosine of the angle between two vectors that have been projected in a multi-dimensional space. In the context of VSMs, the two vectors are array-like structures that contain the frequency count of words of two documents. Research [17] shows that when vectors of words are plotted on a multi-dimensional space, unlike Euclidean distance method, the cosine similarity method does not use the magnitude between documents but captures the orientation of the documents. This is advantageous as two similar documents can be far apart by the Euclidean distance because of the size difference in the frequency of a term in one document as compared to another, for instance the word "movie" appears 40 times in document 1 and only 5 times in document 2 resulting in a large Euclidean distance however the angle might be small. Equation (4) shows that the closer the angle between documents the greater the cosine similarity.

$$Cos\theta = \frac{\vec{a}}{\|\vec{a}\|} \cdot \frac{\vec{b}}{\|\vec{b}\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2}\sqrt{\sum_1^n b_i^2}} \quad (4)$$

As the number of words from the document increase the harder it becomes to visualize in higher dimensional spaces hence the necessity of equation (4) whereby $\overrightarrow{a}$ and $\overrightarrow{b}$ represent two document vectors.

3.5.2 Dot Product

The dot product of two document vectors is taken as a general indication of the similarity of two vectors, with the condition that vectors ought to have the same length. Table 4 indicates that this condition is satisfied as every row has the same number of columns or features. The general formula of the dot product given document A and document B is as follows:

$$A \quad . \quad B \quad = \quad \sum_i^n A_i B_i \tag{5}$$

Furthermore, leveraging matrix multiplication allowed us to calculate all document vectors at once. This is achieved by multiplying the document-term frequency matrix with its transpose as indicated in the formula below:

$$Dot\ Product\ of\ all\ documents = XX^T \tag{6}$$

Alternatively, it may be more useful to find the term correlation instead of document correlation hence to obtain the term correlation perspective of a document corpus such as in Table 4, the dot product formula is adjusted to:

$$Dot\ Product\ of\ all\ documents = X^T X \tag{7}$$

3.5.3 Latent Semantic Analysis (LSA)

LSA a natural language processing technique used for topic modeling. The core idea is to take a matrix of documents and terms and decompose it into a separate document-topic matrix and a topic-term matrix in order to obtain higher level constructs of terms of a document feature matrix for instance terms such as “fun”, “excitement” and “joy” can produce a higher level construct such as “happiness” or “enjoyment”. The aim of

LSA is to reduce the dimensionality of a feature set of a document feature matrix, which tends to become sparser as the aforementioned feature extraction methods like TF-IDF are applied. Research [18] suggests that LSA is best implemented by Singular value decomposition (SVD) method. Furthermore, the experimental results of [18] show that a reduction in the dimension of the item neighborhood leads to an increase in the accuracy of systems employing it. SVD factorizes the term document matrix to extract higher level constructs of terms through equation (8).

$$SVD\ of\ X = U \quad \sum V^T \tag{8}$$

Equation (8) shows the SVD of document corpus *X*, whereby *U* and *V* represent the eigenvectors of the term correlations of XX^{T} and the term correlations of X^TX respectively. LSA operations are performed on term document matrices rather than document term matrices hence the need for transposition indicated in equation (8). However, the implementation of the SVD process is not only computationally intensive but results in reduced factorized matrices that are approximations. Hence, not only are the selected features unknown to the user but there is no autonomy in selecting the features as they are selected by the most favorable mathematical computation. Moreover, SVD will require any new data to be transformed into the same vector space before predictions can be made as indicated by the subsequent equation.

$$D \quad = \quad \sum^{-1} U^T X \tag{9}$$

Equation (9) shows the new data, document D, that's to be projected after data pre-processing steps and feature engineering and extractions steps are applied. This is achieved by multiply the inverted sigma values with the transposed matrix U previously used in equation (8). The usage of equations (8) and (9) helps improve representation of data as well as reduces the dimensionality of feature sets so as to

allow the application of more robust algorithms such as RF rather than single decision trees as the richness of feature sets or signal of columns increases.

3.6 Performance Measures

An essential part of evaluating machine learning algorithms are the performance metrics [19]. Most performance metrics can be classified as confusion metrics as most model performances are described using a confusion matrix. In the subsequent section the confusion matrix and its parameters, and other evaluation measures are briefly described.

3.6.1 Confusion Matrix

The performance of a machine learning algorithm can be visually represented as a table. It is used to describe the performance classification model on test data for which the true value is known.

Table 3.5: Confusion Matrix

N=165	**Predicted: NO**	**Predicted: YES**
Actual: NO	50 (TN)	10 (FP)
Actual: YES	5 (FN)	100 (TP)

As observed in Table 3.5, the total number of samples of this model is 165. The total number of samples of negative class in N samples is 60, whilst the total number of samples of positive class in N samples is 105. The descriptions of other aspects of the matrix are elaborated upon below.

i. True Negatives (TN): This is the value of correctly predicted negative values of the N samples. In this case, the value of the actual class is no (negative) and value of predicted class is also no (negative).

ii. True Positives (TP): This is the value of correctly predicted positive values of the N samples. In this case, the value of the actual class is yes (positive) and the value of predicted class is also yes (positive).

iii. False Positives (FP): This is the value of incorrectly predicted negative values of the N samples. In this case, the value of the actual class is no (negative) and the predicted class is yes (positive).

iv. False Negatives (FN): This is the value of incorrectly predicted positive values of the N samples. In this case, the value of the actual value is yes (positive) and the predicted class is no (negative).

3.6.2 Confusion Metrics

The aforementioned of the parameters of the confusion matrix can be used to calculate the confusion metrics, which are also known as performance measures.

i. Accuracy: The ratio of correctly predicted observations to the total number of observations.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (10)$$

ii. Precision: The ratio of correctly predicted observations to the total predicted positive observations.

$$Precision = \frac{TP}{TP + FP} \quad (11)$$

iii. Recall: Also known as Sensitivity, it is the ratio of the correctly predicted positive observations to the observations in the actual class – yes.

$$Recall = \frac{TP}{TP + FN} \quad (12)$$

iv. Specificity: This is the ratio of incorrectly predicted negative observations with respect to all negative observations.

$$Specificity = \frac{FP}{FP + TN} \quad (13)$$

v. F1 Score: F1 Score refers to the Weighted average of Recall and Precision. It tends to be more useful than accuracy measure, as it takes both false positives and false negatives into account.

$$F1\ Score = \frac{2 * (Recall * Precision)}{Recall + Precision} \quad (14)$$

3.6.3 Other Performance Matrix

In this section, other machine learning performance measures besides confusion metrics are briefly explained.

i. Area Under the Curve (AUC): Often used for binary classification, AUC is the area under the curve of plot False Positive rate vs True Positive Rate at differing points in range [0,1]. The true positive rate is equivalent to sensitivity measure, henceforth it is described with equation (12) and the false positive rate is also known as specificity, hence it is also described with equation (13).

ii. Mean Absolute Error (MAE): This is the average of the difference between the original values and predicted values. In other words, it is the measure of how far apart the predictions are from the actual values. It is mathematically presented in equation (14).

$$MAE = \frac{1}{N}\sum_{j=1}^{N}|y_j - \hat{y}_j| \quad (15)$$

iii. Mean Squared Error (MSE): This measure takes the average of the square of the difference between the original values and the predicted values. The effect of errors becomes more pronounced than in MAE as MSE takes the square of the error found in MAE.

$$MSE = \frac{1}{N}\sum_{j=1}^{N}(y_j - \hat{y}_j)^2 \quad (16)$$

iv. Logarithmic Loss: Known as Log Loss; it is a measure that penalizes the false classifications. In order for log loss to be applied to a classifier model, classifiers must assign probability to each class for all the samples. For instance, if there are N samples that belong to M classes then the log loss is calculated as follows:

$$Logarithmic\ Loss = \frac{-1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} * \log(p_{ij}) \quad (17)$$

Where, y_{ij}, indicates whether sample *"i"* belongs to class *"j"* and p_{ij} indicates that the probability of sample *"i"* belonging to class *"j"*. Furthermore, a log loss nearer to 0 indicates higher accuracy.

Chapter 4

PROPOSED SYSTEM

In this section, the proposed system architecture and its components are outlined and briefly explained. The system architecture is shown in the subsequent section.

4.1 System Architecture

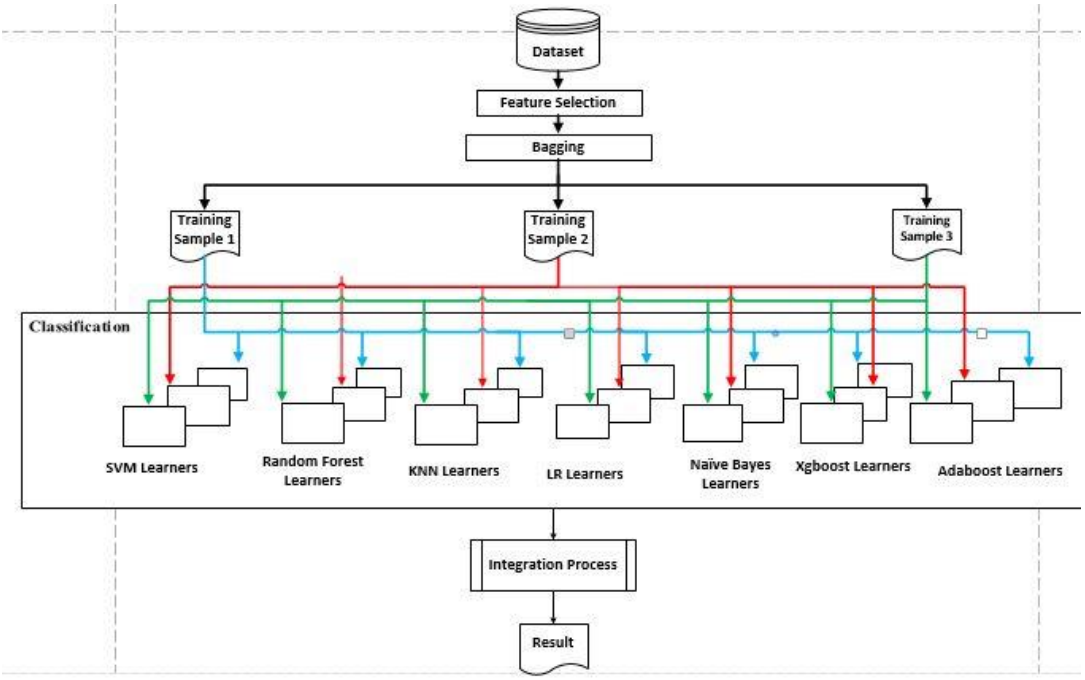

Figure 4.1: System Architecture

4.2 Implementation Framework of Proposed System on Datasets

To maintain authenticity of the experiment and to allow for classifier comparison, the following framework was adopted for this study, whereby experimental conditions are maintained throughout each classifier experiment and our proposed system's experiment. The framework is described in the subsequent section:

1. Dataset [5,6,7] is split into train and test data at 80:20 ratio, with 80% of the train data being used for training and 20% for testing.
2. Perform Feature Selection and Feature engineering - the same feature selection techniques are maintained for all datasets and experiments.
3. Apply sparsity reduction at 0.99% such that for every token feature that appears less than 1% in the entire dataset is removed as a feature.
4. Bag train data, with replacement, into three clusters of 70% of the *N* samples of dataset.
5. Maintain the indexes of the clustered samples, to be applied to all other classifiers to ensure the validity of the experiment results.
6. Perform predictive modelling using SVM, RF, LR, k-nearest neighbors, Naïve Bayes, XGBoost, and AdaBoost classifiers.
7. Step 6 is repeated for the aforementioned three clusters produced in step 4.
8. Apply all seven trained machine learning algorithms to the test data.
9. Perform voting on the seven classifier predictions by Majority Voting and Weighted Majority Voting.

4.3 System Components

In this section the components of the system architecture are explained.

4.3.1 Datasets

In this study three standard datasets [5,7,9] are used during the experimental phase. Firstly, the Polarity dataset version 0.9 [5], introduced as part of Pang and Lee's inaugural work in sentiment analysis, the dataset consists of 1400 reviews, 700 positive and 700 negative processed reviews. Dataset [5] was created from a subset of the Internet Movie Database (IMDb) archives. The original data consisted of star ratings alongside the review information, which were removed to ensure that feature selection

techniques and classification algorithms would make use of text analytics rather than the star rating system for predictive modelling. Thus, the processed reviews ensured classification was made solely based on review data.

Another dataset [7] used was the Association for Computational Linguistics Internet Movie Database (aclImdb) version 1. The dataset consisted of a set of 25,000 highly polar movie reviews for training, and 25,000 for testing with an equal distribution of class polarity, that is 12,500 reviews were classified as positive and 12,500 as negative. There is additional unlabeled data for use as well. [9] was the Polarity dataset version 2.0, which consisted of 1000 positive and 1000 negative processed reviews. The class distribution of datasets is summarized in the subsequent table.

Table 4.1: Summary of Dataset Information

Dataset Name	**Dataset used in**	**Class distribution in dataset**	
		Positive Class	Negative Class
Polarity Dataset version 0.9	Training	560	560
	Testing	140	140
Polarity Dataset version 2.0	Training	816	784
	Testing	204	196
ACL's Internet Movie Database (IMDb)	Training	12500	12500
	Testing	12500	12500

At this stage of the proposed system, the aforementioned datasets underwent through the following procedures.

4.3.1.1 Data Pre-Processing

Pre-processing refers to the cleaning and structuring of data. This stage generally involves Tokenization and normalization. In our experiment the following sub-stages of Pre-processing were performed on [5].

i. Tokenization: The segmentation of documents into a list of tokens either of words, phrases or sentences in order to optimize the document for further processing. For instance, the sample sentence from Table 1 is transformed into Figure 4.2.

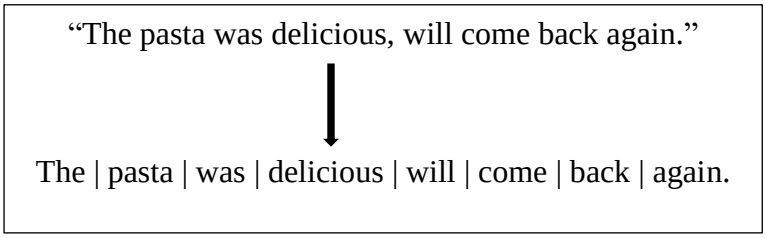

Figure 4.2: Tokenization of Sample Sentence

ii. Normalization: The process of converting all the word tokens in a document into one constant case - lower case or upper case, to avoid case-sensitive issues when using tokens for predictions.

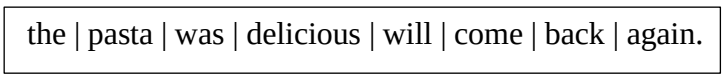

Figure 4.3: Uniform Case of Tokens of Figure 4.2

iii. Removal of stop words: The removal of very common and high-frequency words that do not affect the semantic meanings of sentences. This process was carried out by removing spaces and tabs, and frequently used stop words (irrelevant words, prepositions, ASCII codes and the list of inbuilt stop words from the Quanteda library [21] which amounted to 400 terms.

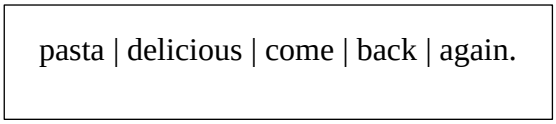

Figure 4.4: Stop Words Removed from Figure 4.3

iv. Stemming: It is the process of transforming tokens into their stem or root form. It lends to a seamless feature extraction process.

4.3.2 Feature Selection

This component of the system architecture refers to the process of reducing the number of input variables used in a predictive model as explained in the Background chapter. In this study, the following features techniques were used on [5].

i. Bag of Words: The BOW model or unigram model was applied to the corpus of documents [5] and tokens of sentences were used as features of the vector space model representation. However, the unigram model on its own proved to be ineffective, as it did not provide clear details to the model about the subject matter. Hence, the Bag of Words model was extended to the Bi-gram model.

ii. N-grams: Unigrams and Bigrams of tokenized terms were used as features of the data frame matrix (dfm). Expanding the BOW model to account for word-ordering largely improved accuracy measures.

iii. Text length: In this study text length was used as a feature based on the findings in human psychology research [20] with regards to negative and positive news. In [20] negative reviews on average are observed to be lengthier than positive reviews as the psychophysiological experiment done showed that negative news elicits stronger and more sustained reactions than positive news. This formed the hypothesis of text length being possibly related to class distribution. In this study, text length refers to the word count of each review in the datasets [5,7,9].

The distribution of text length in relation with the class labels confirmed our hypothesis as shown in Figure 4.5.

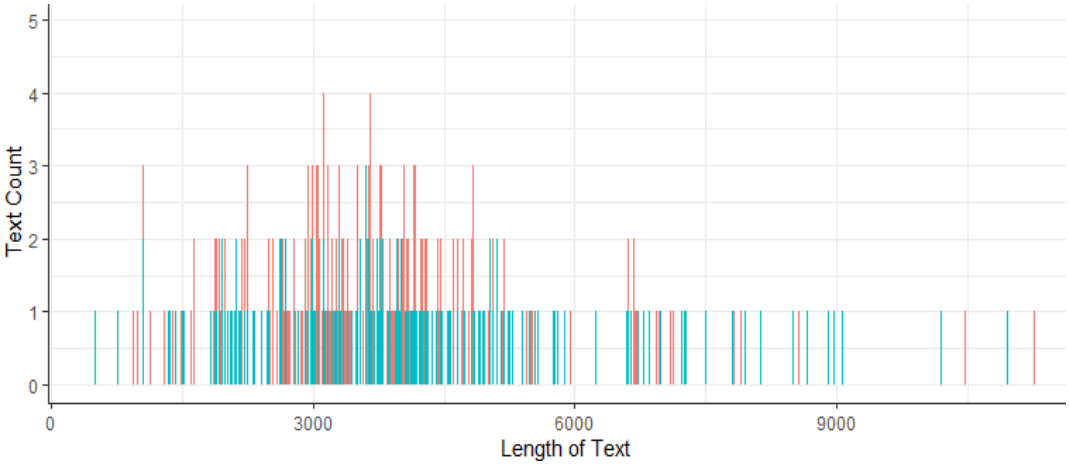

Figure 4.5: Text Length of Negative and Positive Class Labelled Reviews in Polarity dataset version 0.9

As observed in Figure 4.5, the text length of each review in one of our training datasets [5] is of value in determining polarity as the text length graphing is indicative of the fact that negative reviews shown in red, are frequently much longer than positive reviews in blue. This is also the case for other datasets [7,9] used in this study. In this study we appropriated text length as a feature alongside cosine similarity in order to determine the closeness of reviews in during training phase.

iv. Cosine Similarity: Projecting document term frequency relations in a vector space model enabled each word in the train set of our datasets [5,7,9] to correspond to one of the dimensions in the multi-dimensional space. The size of angles of review documents in the train sets of our datasets were compared to determine the similarity between two documents. The similar documents had smaller angles between them. Moreover, text length was also used along

document term frequency counts when representing the documents in order to compute their cosine similarity. In our proposed system's graphical user interface, matching documents were grouped together according their class labels, with red indicating negatively labelled documents and blue for positive cases.

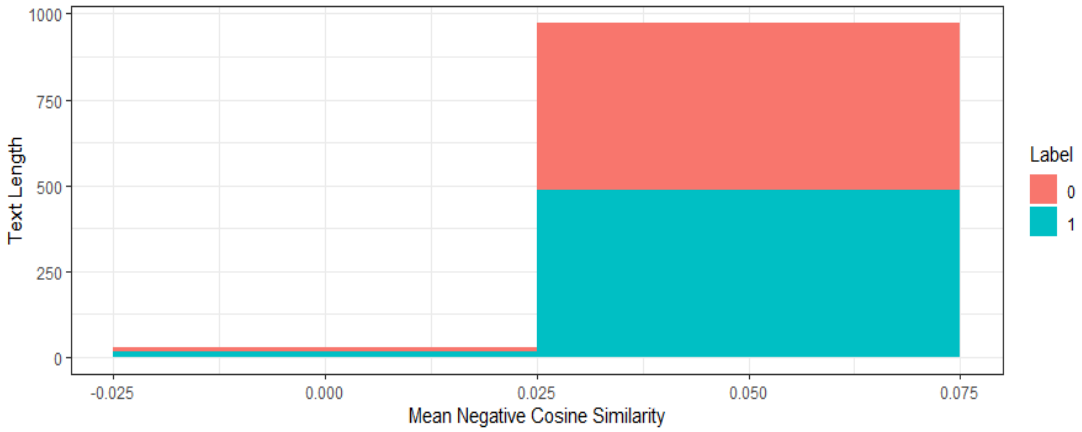

Figure 4.6: Cosine Similarity of Positive and Negative Reviews of Polarity Dataset Version 0.9

Figure 4.6 shows that the notion of using cosine similarity alongside text length in order to categorize polarity of the reviews is quite useful. In the figure above all train reviews in dataset [5] determined to be most similar to negatively labelled document are shown. As expected, most positively labelled documents which were not found to be similar to a negatively labelled review document in train data of dataset [5] formed a red assemblage. Those documents determined to be most similar to the negative review document of the train data formed a red assemblage.

v. Boruta Feature: After applying the aforementioned features, a wrapper method was applied over the feature set of n-gram tokens, negative cosine similarity value, text length value of each reviews in the train data, in order to determine the most important features in dataset [5] with respect to the outcome variable,

that is negative and positive polarity. The set number of iterations to determine whether to accept or reject a feature in dataset [5] was set to 1000 maximum runs. After 1000 runs, 66 features of the 4461 features were determined to be the most important in predicting polarity. Furthermore, of the 66 features automatically selected, 2 were text length and cosine similarity, the rest were n-gram tokens. The lengthy number of runs ensured there were no tentative features. Dataset [5] produced the list of features below.

Table 4.2: Boruta Selected Features in Polarity Dataset version 0.9

Sequence of Most Important Feature						
hilari	movi	seri	hair	minut	half	terribl
mayb	job	worst	view	ten	make_sens	dull
fun	there	inept	credit	observ	whatsoev	fail
director	perform	supp	stupid	laughabl	unfunni	attempt
time_minut	wasn.t	Complex	idiot	wast	aw	TextLength
bad	bother	sequel	memor	pathet	ridicul	negClassSimilarity
guy	enjoy	water	reel	uninterest	flat	
joke	fun	Outstand	solid	insult	lifeless	

As can be seen the most relevant unigrams and bigrams were chosen along our proposed feature selections.

4.3.3 Bagging

At this stage, 3 samples of 70% of size "A" of the datasets are generated from an initial dataset [5,7,9] of size N by a random draw with replacement "A" observation. The same 3 samples of the dataset are used for training for every classifier used in the ensemble to maintain consistency in the experiment. Sampling of the dataset produces the 3 bootstrap samples previously mentioned, these samples behaved as independent datasets drawn from true distribution, in order to fit weak learners for each of the

samples and finally aggregate their results and so we obtained an ensemble model of lower variance than its components.

4.3.4 Classifiers

Having performed bagging at this stage of the system architecture, seven machine learning methods are applied for predictive modelling using the three samples of bagged data. These seven classifiers are SVM, RF, LR, KNN, NB, AdaBoost, XGBoost. In this study seven classifiers were proposed in order to eliminate classifier bias in predictive modelling by picking machine learning methods of the same family, moreover the aim was to introduce classifier diversity, a mixture of strong and weak classifier methods. The three samples of bagged data are fed to each of the seven machine learning methods hence each machine learning algorithm will have a learner classifier for each sample of data plugged into it. In other words, 21 models will be produced, three for each machine learning method. The combination method of these model results is explained in the subsequent section.

4.3.5 Integration Process

In this study we propose a two method for the integration of classifiers namely, Majority Voting and Weighted Majority Voting.

i. Majority Voting: Sometimes referred to as Naïve voting. In this voting scheme, as the final output, the majority vote from the seven classifiers was used.
ii. Weighted Majority Vote: In this voting scheme, the final outputs of the seven classifiers used the average F1 Score from the training phase as given weight of each classifier, the ensuing products were summed and normalized to obtain predictions in the range of [0,1].

Chapter 5

EXPERIMENT

In this chapter, the results of the implementation framework of the proposed system on datasets [5,7,9] are discussed, the classifier results are outlined and the proposed system results are also shown.

As outlined by the implementation framework in the previous chapter, all experiments on each dataset are held under the same conditions for all classifiers. These conditions include the data splitting ratio of 80:20 for training and testing respectively; the same feature selection and feature engineering techniques as discussed in the background chapter are used across all experiments. Lastly, experiments done on each dataset use the same bagged data indexes for training all 7 classifiers and the proposed classifier in order to maintain the integrity of the experiment and allow for classifier comparison 5.1.1 Hypothesis 1.

5.1 Results on Datasets

In this section the performance measures, such as F1 Score and Accuracy, of our study's proposed classification framework on datasets [5,7,9] are discussed and shown in the subsequent tables.

5.1.1 Results on Polarity Dataset Version 0.9 Experimentation

Table 5.1 shows the results of the implementation of this study's proposed ensemble classifier and the results of 7 classifiers, that is SVM, RF, LR, KNNs, Naïve Bayes, XGBoost, and AdaBoost when applied to the polarity dataset version 0.9 [5].

Table 5.1: Polarity Dataset Version 0.9 Results

Method	**Evaluation Parameters**				
	F1-Score	Accuracy	Recall (Sensitivity)	Precision	Specificity
SVM	0.8195	0.8141	0.8400	0.8000	0.7879
Random Forest	0.8041	0.809	0.7800	0.8298	0.8384
Logistic Regression	0.8159	0.8141	0.8200	0.8119	0.8200
KNN	0.4199	0.4742	0.3800	0.4691	0.3800
Naïve Bayes	0.7798	0.7588	0.8500	0.7203	0.6667
XGBoost	0.78849	0.799	0.7300	0.8488	0.8687
AdaBoost	0.8063	0.8141	0.7700	0.8462	0.8586
Proposed Ensemble (Majority Voting)	0.8125	0.8191	0.7800	0.8478	0.8586
Proposed Ensemble (Weighted Majority Voting)	0.8144	0.8191	0.7900	0.8404	0.8485

In this study, our core measure was the F1 Score. As F1 Score is best suited to deal with class imbalances, and F1 Score also gives a balance between the recall and precision. In this study, the proposed system's ability will be valued highly on the F1 Score as many real-life classification applications such as cancer reoccurrence detection rely on the classifier's ability to screen false positive while accounting for false negative cases (reoccurrences) which are measured in the F1 Score.

As observed in Table 5.1, our proposed system has the second highest F1 Score at 0.8411, just 0.0014 lower than the highest, however our system carries the highest accuracy measure at 0.8191 followed up by Logistic Regression and SVM at 0.8141. Our proposed system had an F1 Score of 0.8144 under Weighted Majority Voting and 0.8125 under Majority Voting. These F1 Scores ranked second after SVM and Logistic regression measures. Furthermore, our proposed system under Majority Voting and Weighted majority respectively produced precision measures of 0.8404 and 0.8487, and specificity measures of 0.8586 and 0.8485, which was greater than any other classifier.

Overall, our proposed system produced greater performances on dataset [5] than 7 other standard machine learning classifiers with Logistic Regression and SVM comparatively close in their performances.

5.1.2 Results on Polarity Dataset Version 2.0 Experimentation

Table 5.2 shows the results of the implementation of this study's proposed ensemble classifier and the results of 7 classifiers, that is SVM, RF, LR, KNNs, Naïve Bayes, XGBoost, and AdaBoost when applied to the polarity dataset version 2.0 [9].

Table 5.2: Polarity Datasets Version 2.0

Method	Evaluation Parameters				
	F1-Score	Accuracy	Recall (Sensitivity)	Precision	Specificity
SVM	0.8444	0.8425	0.8550	0.8341	0.8300
Random Forest	0.8394	0.8450	0.8100	0.8710	0.8800
Logistic Regression	0.8053	0.8150	0.7650	0.8500	0.8650
KNN	0.3376	0.4800	0.2650	0.4649	0.6950
Naïve Bayes	0.7677	0.7700	0.7600	0.7755	0.7800
XGBoost	0.7684	0.7800	0.7300	0.8111	0.8300
AdaBoost	0.7839	0.7850	0.7800	0.7879	0.7900
Proposed Ensemble (Majority Voting)	0.8300	0.8400	0.8000	0.8602	0.8700
Proposed Ensemble (Weighted Majority Voting)	0.8300	0.8400	0.8000	0.8602	0.8700

Table 5.2 shows that under the same experimental conditions for the polarity dataset version 2.0 our proposed ensemble is in the top 1% percentile as it ranks 3rd in the F1 Score and Accuracy measure categories by 1.444 and 0.0050 respectively. SVM led all classifiers in F1 Score measure at 0.8444 and Random Forest led all classifiers in the accuracy measure at 0.8450.

In both experiments the top two performing classifiers, under the training conditions described in the framework implementation section in chapter 4, are SVM and our proposed ensemble classifier which produced relatively the same results across the evaluation metrics with an average difference of 0.0051% in the Accuracy and F1 Score metric respectively.

The closeness of performances of our proposed classifier with SVM over several samples of data and under the same experimental setup suggests that this study's proposed classifier, like SVM, has excellent scaling capabilities when dealing with high dimensionality data such as were our datasets [5,7]. Moreover, by observation of these experiments we suggest that the proposed classifier works well with unstructured even data and semi-structured data such as text and tress.

5.1.3 Results on ACL's Internet Movie Database experimentation

Table 5.3 shows the results of the implementation of this study's proposed ensemble classifier and the results of 7 classifiers, that is SVM, RF, LR, KNNs, Naïve Bayes, XGBoost, and AdaBoost when applied to ACL's Internet Movie Database [7].

Table 5.3: ACL's Internet Movie Database Results

Method	**Evaluation Parameters**				
	F1- Score	Accuracy	Recall (Sensitivity)	Precision	Specificity
SVM	0.8093	0.794	0.8447	0.7768	0.7396
Random Forest	0.7379	0.7286	0.7379	0.7379	0.7188
Logistic Regression	0.8128	0.7940	0.8641	0.7672	0.7188
KNN	0.5490	0.5377	0.5437	0.5545	0.5312
Naïve Bayes	0.7639	0.7236	0.8641	0.6846	0.5729
XGBoost	0.7580	0.7337	0.8058	0.7155	0.6562
AdaBoost	0.7814	0.7638	0.8155	0.7500	0.7803
Proposed Ensemble (Majority Voting)	0.8145	0.8000	0.8738	0.7627	0.7083
Proposed Ensemble (Weighted Majority Voting)	0.7941	0.7889	0.7864	0.8020	0.7917

As observed in Table 5.3, our proposed system performs better than other machine learning classifiers on the ACL's IMDb dataset. We obtained an accuracy of approximately 0.79 and 0.80 which matched the highest accuracy measures from SVM and LR, the latter slightly surpassing those measures. However, our proposed system when using majority voting, obtained a greater F1 Score ,0.8145, than every other classifier.

In this study, we observed that the proposed ensemble classifier across 3 standard dataset experiments almost always achieves better performances than single classifiers in terms of accuracy, precision and F1 Score or at the least match the best performing single classifier. In the subsequent chapter we compare this proposed system with other existing literatures.

Chapter 6

DISCUSSION

In this chapter, the comparative analysis based on results obtained using the proposed approach to that of other literatures using the datasets [5,7,9] is shown in the subsequent table.

Pang et. al he implemented an n-gram feature-based classification experimentation using the polarity dataset version 0.9 [5] on 3 machine learning algorithms – Naïve Bayes, Maximum Entropy, and Support Vector Machine [4]. They obtained the accuracy measures of 81.0%, 80.8% and 77.1% for Naïve Bayes, Maximum Entropy and Support Vector Machine respectively, which were all less than our proposed classifier's accuracy measure of 81.91 as shown in Table 9. Moreover, Li et.al proposed a system to carry out sentiment classification through classifier combinations with multiple feature sets on the polarity dataset [5]. They obtained precision measures of 83.00%, 82.71%, 82.36%,82.36%,81.43% for sum, product, max, min and vote combination rules. Which were all less than the precision of our proposed system at 0.84.78%. hence our proposed ensemble has a greater positive predictive value than that of the precision obtained by [22]. Tsutsumi et.al was another comparative study for our results, using multiple classifiers and scoring calculations such as Naïve Voting, Weighted Voting, and SVM which obtained accuracy measures of 85.8%,86.4%, 87.1% all considerably higher than our accuracy of 81.91% [10]. However, without the scoring calculations, the accuracy measures were a lot more

comparative as Tsutsumi et.al noted that they obtained accuracy measures of 82.2%, 80.5% for SVM and ME respectively.

Moreover, ACL's Internet Movie Database was also used for experimentation in this study. Salvetti et. al appropriated the dataset in their study of the impact of lexical filtering of movie reviews on the accuracy of two classifiers namely, Naive Bayes and Markov Model [23]. With respect to the Overall Opinion Polarity (OvOP) identification, [23] achieved an accuracy measure of 80%, which matched the accuracy obtained by our proposed classifier as shown in Table 6.1.

Pang and Lee's proposed subjectivity summarization for sentiment analysis using minimum cuts produced slightly better performances than our proposed system as it had an accuracy of 86.40% compared to our system's 84.00% accuracy measure [8]. The intricacies of data partitioning used in their system might have been the cause of the better accuracy performance they achieved. Lastly, we also compared our proposed system to Baid et.al's work using an automated open source software called WEKA [24]. We obtained better accuracy measures than that of [24]. As our accuracy was 84.00% compared to their 81.40%.

In the subsequent section, we summarize the comparative performance of our proposed ensemble classifier against other classifiers from existing literatures.

Table 6.1: Comparative Performance Results of Proposed System with other Literature

Dataset	Literature	Method	Result	
			Literature's System Result Accuracy (%)	Our Proposed system Accuracy (%)
Polarity Version 0.9	Pang et. al	N-gram feature-based classification	81.00	81.91
	Tsutsumi et. al	Multiple classifiers and score calculation	87.10	81.91
	Lee et. al	Classifier combination through POS tags, unigrams etc.	83.00 Precision)	84.78 (Precision)
ACL's Internet Movie Database	Salvetii et.al	Lexical filtering	80	80
	Beineke et. al	Human baselines and machine learning algorithms	65.9	80
	Tripathy et. al	N-gram machine learning approach	83.65	80
Polarity Dataset Version 2.0	Baid et. al	Machine learning with WEKA open source software	81.40	84.00
	Pang and Lee	Subjectivity summarization based on minimum cuts	86.40	84.00

As observed in the table above, our proposed system largely had the better or equal performance to existing literature works only coming in second to Pang and Lee's subjectivity summarization framework [8] and Tripathy et. al's ngram approach [6]. This is noteworthy as we didn't perform any classifier selection experiments in order to determine our ensemble. Selections were done considering classifier types, as we desired to eliminate classifier biases as much as possible for the predictive modelling. The classifier types used include the following linear classifiers, decision trees, boosted trees and support vector machines. In future studies, we would expand on

classifier selection and introduce more nuanced feature selection tools such as part-of-speech tags and minimized subjective cuts. Moreover, the study would expand to other types of social micro-blogs such as Twitter and Facebook.

Chapter 7

CONCLUSION

In conclusion, the growth of social media has resulted in advertisements shifting to local blogs and forums. Businesses now thrive on the opinions and sentiments expressed about their company, product lines and reviews of rival companies found on micro-blogs. Hence the need for more robust and quality sentiment analysis classifiers for quicker and more effective predictions of data modelling on micro-blogs. Henceforth we proposed a classifier model that performed bagging of models for 7 classifiers that were combined by majority voting and weighted majority voting. We observed that our proposed classifier produced better results than any singular classifier in all of our standard datasets. Furthermore, we observed that our proposed method ranked highest amongst the best classification frameworks when compared to other existing literatures. Conclusively, our hypothesis that bagging ensembled classifiers constantly produces better performance than any singular or ensemble learner was mostly met as our method was usually better or at most equal to existing sentiment analysis frameworks.

REFERENCES

[1] G. Gautam, D. Yadav," Sentiment analysis of twitter data using machine learning approaches and semantic analysis," in *Proc. Seventh Int. Conf. Contemporary Comput. (IC3)*, Aug 2014, pp. 437-442.

[2] T. Hastie, R. Tibshirani, J. Friedman, "Unsupervised learning" in *The Elements of Statistical Learning*, Springer Series in Statistics. New York, NY, USA: Springer, 2009, ch.14, pp. 485-585.

[3] R. Feldman, "Techniques and applications for sentiment analysis." Commun. ACM, vol. 56, pp.82-89, Apr. 2013

[4] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques," in *Proc. EMNLP*., 2002, pp. 79-86

[5] B. Pang, L. Lee, and S. Vaithyanathan, "Introduced Polarity dataset v0.9," in *Proc. EMNLP*., 2002, pp. 79-86. http://www.cs.cornell.edu/people/pabo/movie-reviewdata/?fbclid=IwAR3sS3y6Tg_A_yy9pNrt2xWwR1bqQP6SfiNOulFARfOv388KP8ADjMFzE

[6] A. Tripathy, A. Agrawal, S.K. Rath, "Classification of sentiment reviews using n-gram machine learning approach," Expert Systems with Applications, vol. 57, pp.117-126, Mar. 2016

[7] A.L. Maas, R.E. Daly, P.T. Pham, D. Huang, Y. Andrew, C. Potts, “Learning Word Vectors for Sentiment Analysis, Large Movie Review dataset v1.0” in *Proc. The 49th Annual Meeting of ACL.,* Jun. 2011, pp. 142-150. https://ai.stanford.edu/~amaas/data/sentiment/aclImdb_v1.tar.gz

[8] B. Pang and L. Lee, “A Sentimental Education: Sentiment analysis using subjectivity summarization based on minimum cuts,” in *Proc. ACL*., Jul. 2004, pp. 271-278.

[9] B. Pang and L. Lee, “Introduced polarity dataset v2.0,” in *Proc. ACL*., Jul. 2004, pp.271-278.http://www.cs.cornell.edu/people/pabo/movie-review-data/review_polarity.tar.gz

[10] K. Tsutsumi, K. Shimada, T. Endo, “Movie Review Classification Based on a Multiple Classifier”. In *Proc. 21st Pacific Asia Conf. on Lang., Inf. and Computation*, 481–488, doi: http://hdl.handle.net/2065/29106

[11] I. Osajima, K. Shimada, and T. Endo. “Classification of evaluative sentences using sequential patterns”. In *Proc. 11nd Annual Meeting of The Association for Natural Lang. Process.*, Japan., 2005.

[12] S. Li, C. Zong and X. Wang, "Sentiment Classification through Combining Classifiers with Multiple Feature Sets," In *Proc. 2007 Int. Conf. on Natural Lang. Process. and Knowl. Eng.*, Beijing, 2007, pp. 135-140, doi: 10.1109/NLPKE.2007.4368024

[13] C. W. Hsu, C.C. Chang, C.J. Lin, “A practical guide to support classification,” Nov. 2003. http://www.csie.ntu.edu.tw/~cjlin/papers.html

[14] J. H. Friedman, “Greedy function approximation: a gradient boosting machine.” Ann. Statist., vol 5, pp. 1189–1232. Apr. 2001, doi:10.1214/aos/1013203451

[15] Y. Freund and R. Schapire, “Experiments with a new boosting algorithm,” In *Proc. Thirteenth Int. Conf. Mach. Learn.,* Italy, 1996, pp. 148-156

[16] S.K.M Wong and V.V. Raghavan, “Vector Space Model of Information Retrieval – A Reevaluation,” In *Proc. 7th Annual Int. ACM SIGIR Conf. Research and Development in Info. Retrieval.*, England, 1984, pp. 369-381

[17] S. Prabhakaran, *Machine Learning Plus*, Oct. 22, 2018. [Online]. Available: https://www.machinelearningplus.com/nlp/cosine-similarity/

[18] M. G. Vozalis and K. G. Margaritis, "Applying SVD on item-based filtering," 5th International Conference on Intelligent Systems Design and Applications (ISDA'05), Warsaw, 2005, pp. 464-469, doi: 10.1109/ISDA.2005.25

[19] M. Sokolova, N. Japkowicz, and S. Szpakowicz, “Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation”. In *Australas. Joint Conf. Arti. Intell., in Sattar A., Kang B. (eds) AI 2006: Advances in Arti. Intell.*, vol. 4304, pp. 1015 – 102, doi: https://doi.org/10.1007/11941439_114

[20] S. Soroka and S. McAdams, “News, Politics, and Negativity, Political Communication,” *Political Comm.*, vol. 32, no. 1, pp. 1-22, Feb. 2015. Accessed on: Feb 27, 2020. [Online]. Available doi: 10.1080/10584609.2014.881942

[21] B. Kenneth, K. Watanabe, H. Wang, P. Nulty, A. Obeng, S. Müller, and A. Matsuo. “Quanteda: An R package for the quantitative analysis of textual data”, *J. Open Source Software*., 3(30), pp. 774. (2018). [Online]. Available: https://doi.org/10.21105/joss.00774.

[22] S. Li, C. Zong and X. Wang, "Sentiment Classification through Combining Classifiers with Multiple Feature Sets," In *Proc 2007 Int. Conf. on Natural Lang. Process. Knowl. Eng.*, Beijing, 2007, pp. 135-140. doi: 10.1109/NLPKE.2007.4368024

[23] F. Salvetti, S. Lewis, & C. Reichenbach. Automatic Opinion Polarity Classification of Movie Reviews. (2004)

[24] B. Palak, A. Gupta, and N. Chaplot, “Sentiment Analysis of Movie Reviews using Machine Learning Techniques,” In *Proc. Int. J. Comp. App.*, Dec. 2017, vol. 179, no. 7, pp. 45-49. doi: 10.5120/ijca2017916005