Comparative Evaluation of Supervised Learning Algorithms for Sentiment Analysis of Movie Reviews

Raj K. Palkar
Department of Computer Engineering
K.J. Somaiya College of Engineering, Mumbai, Maharashtra, India.

Kewal D. Gala
Department of Computer Engineering
K.J. Somaiya College of Engineering, Mumbai, Maharashtra, India

Meet M. Shah
Department of Computer Engineering
K.J. Somaiya College of Engineering, Mumbai, Maharashtra, India

Jay N. Shah
Department of Computer Engineering
K.J. Somaiya College of Engineering, Mumbai, Maharashtra, India

ABSTRACT
Online forums and social networking websites provide users with a platform for expressing their opinions. Manually evaluating these reviews for crucial analytical information is cumbersome. Sentiment analysis deals with analyzing such massively available textual data and determining its polarity. This research paper provides a comparative study of multiple well-known supervised machine learning algorithms on three standard datasets confined to the domain of movie reviews. The study is supported by illustrative plots and experimental results. The research work can be used as a base for further exploration in predicting the sentiment value of textual data in alternate domains using advanced machine learning algorithms.

General Terms
Data Mining, Machine learning

Keywords
Sentiment Analysis, Machine Learning, Text classification, Naïve Bayes, Support Vector Machine, Maximum Entropy, Classification and Regression Trees, Random Forest, movie reviews.

1. INTRODUCTION
One of the important factors that affect the decision-making process is what kind of opinions and views people have regarding the subject being considered [1]. Since the advent of online sources of public expression like blogs, social networking websites and web forums, consumers are able to look for reviews regarding a particular commodity and this makes analysis of online reviews a fundamental variable in purchase decisions [7]. Even business organizations consider these sources for evaluating the overall feedback regarding their commodities.

The large amount of textual data available online can prove to be of great significance, when analyzed with necessary expertise and tools. One such field of text mining caters to support decision making by extracting and analyzing opinion oriented text. This field is referred to as Sentiment Analysis or Opinion Mining which aims at identifying the orientation of text; whether the writer has provided a positive opinion or a negative one. Sentiment analysis involves application of natural language processing [10], computational linguistics [10], and text analytics to identify and extract subjective information [10] in source materials. It benefits both the users as well as the manufacturers in obtaining statistical information regarding how a commodity performs in the market from the consumer as well as business point of view.

Among the different approaches to deal with Sentiment Analysis, the focus is on Machine Learning strategies considering their extensive usage in today's world. The machine learning algorithms included in the comparative study comprise of Naive Bayes, Support Vector Machine, Maximum Entropy, Support Vector Machine, Classification and Regression Trees, and Random Forest. Modifications are avoided in the widely accepted algorithms and use the default strategies to set a base for further research in these strategies.

The domain for sentiment analysis task is restricted to movie reviews. Data is obtained from three standard datasets provided by prestigious educational institutions. Large movie reviews dataset v1.0 [12], which consists of 25,000 labeled and processed reviews for training and another 25,000 labeled and pre-processed reviews for testing, is provided by Maas et. al. (2001) [12] from Stanford University. Another dataset for evaluating the algorithms is Cornell polarity dataset v2.0 [13], produced by Pang and Lee (2004) [13], which consists of 1000 positive and 1000 negative processed reviews. Sentiment labeled sentences IMDb dataset[14] retrieved from University of California, Irvine which is originally gathered from movie reviews dataset created by Maas et al is also considered. It comprises of 500 positive and 500 negative reviews.

2. LITERATURE SURVEY
The problem of sentiment analysis is well adhered in the branch of computer science and different approaches have been proposed to analyze textual reviews in distinct domains. Two of the most widely used approaches include Lexicon-based approach and machine learning.

2.1 Lexicon-based approach
Lexicon-based approach [15] basically comprises of lookup methods and application of linguistic rules. In this approach, a dictionary is taken into account which helps to map words with their sentimental polarity [15] and semantic value.

2.2 Machine learning
Machine learning is a branch of Artificial Intelligence that deals with interpreting and extracting useful information from the given data, without human intervention. Machine learning techniques are classified into three categories: Supervised learning, unsupervised learning and reinforcement learning. For sentiment analysis task, supervised learning approach is preferred. In the category of supervised learning, the study focuses on Naïve Bayes, Support Vector Machine, Maximum Entropy, Classification and Regression Trees, and Random Forest methodologies.

2.2.1 Naïve Bayes

Sentiment Analysis can be visualized as a binomial classification problem. Naive Bayes algorithm is used to classify the text in two categories, namely positive and negative. The problem then left is to consider each and every term or only those terms which expresses the opinion. The formula for calculating probability of likelihood of a class given a document is provided below

$$P(c|d)\ \alpha\ P(c)\prod_{1\leq k\leq nd} P(t_k|c)[2]$$

Where, P (t_k/c) is the class C conditional probability in document d and P(c) is the prior probability of class C of a document.

The primary goal in classification of text is to find the best possible class for a particular document. The features for classification may vary with respect to application such as term frequency greater than threshold can be a feature. The simplified equation for calculating class membership with respect to a text document is represented as:

$$c_{map} = \arg max_{c\in C}[\log \hat{P}(c) + \sum_{1\leq k\leq n_d} \log \hat{P}(t_k|c)][2]$$

Where, P is an estimated value obtained from the training set. The term indicates the prior class c relative frequency and every parameter (conditional) is a measurement to provide knowledge of goodness of indicator.

2.2.2 Support Vector Machine

Support Vector machine (SVM) is the methodology of a vector oriented model which is related to the transformation of text document to feature vector and then they are processed for classification. [2]

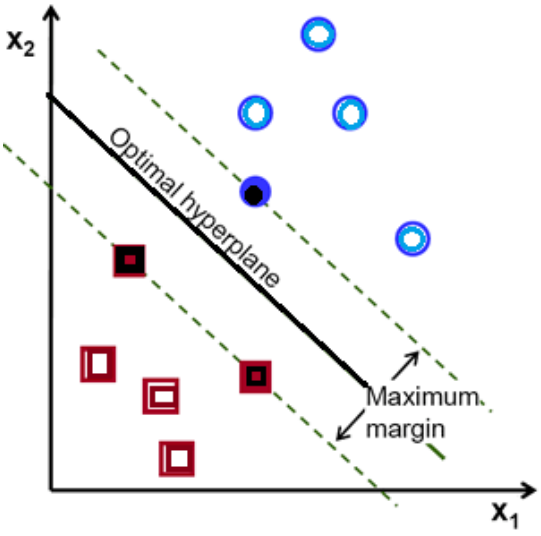

Figure 1: Classification using SVM

The figure above shows the margin classifier which separates the classes with the separating hyper plane and the hyper plane is surrounded by parallel planes on both sides of it known as plus plane and minus plane. The primary goal is in finding the separating hyper plane with the help of support vectors. Currently SVM is considered to be most accurate supervised classification approach. The performance of SVM is independent of the size of training data. The distance between two parallel planes (plus plane and minus plane) is called the marginal distance and the points lying on those planes are called support vectors. The hyper plane is perpendicular to the normal vector and hence, all the points lying on the hyper plane should satisfy the equation:

$$w^T x = -b[2]$$

The support vectors are defined by the sigmoidal function:

$$f(x) = sign(w^T x + b)[2]$$

Where, the classes are represented in a bipolar manner i.e. +1 and -1. Here one class can be represented by a value of -1, and other class can be represented by a value of +1. The next step is to define the geometric mean and functional mean with the objective function to minimize $\frac{1}{2}w^T w$, subject to $y_i(w^T x_i + b) \geq 1$ for all$\{(x_i|y_i)\}$. [2] Quadratic programming is used for solving this problem.

2.2.3 Maximum Entropy

The maximum entropy uses technique of estimating probability distribution. It is widely used in natural language processing tasks. The primary principle of maximum entropy is that the distribution of data should be kept uniform in cases where not much knowledge about the data is available. The derivation of constraints is obtained from the labeled training data and they are represented as features expected value. As the constraints keep on increasing, the model creation becomes more complex. Following steps are considered while using maximum entropy:

- Identification of features to be included in the model
- Calculation of the expected value of feature to be used as a constraint for the model

Thus, maximum entropy imposes a restriction on the distribution to have the same value as expected value of a feature in the distribution of model. The formula for calculation of entropy is as given below:

$$P(c|d) = \frac{1}{z(d)}\exp(\sum_i \lambda_i f_i(c,d))[16]$$

maxent is a package used in R with tools for data classification using multinomial logistic regression, also known as maximum entropy.[4] The maxent package provides maximum entropy classifier which is fast and occupies low memory for execution in order to avail variety of classification tasks which includes natural language processing and text classification.[4]

2.2.4 Classification and Regression Trees

A tree is generated by binary partitioning in recursive manner by using the response obtained from the specified formula and choosing the splits from the terms of the RHS of formula of the syntax. Numeric variables are divided into two modules which are X < a and X > a;[3] i.e. the levels of pair which are unordered are divided into non empty sets. The split or set which minimizes the sum of impurities or maximizes the reduction in impurity is needed to be chosen, and the process is repeated. The continuation of splitting occurs until the leaf nodes are small enough to split further. The limitation of growth of tree is restricted to the depth of 31 by labelling nodes with the help of integers other than factor predictor variables which have limitation of 32 levels. The reason for imposing limit is to provide ease in labelling, but since their use in a classification tree with three or more levels in a response involves a search over $2^{(k-1)} - 1$ groupings for k levels, the practical limit is much less. [3]

General Algorithm for tree construction by exhaustive search in CART:

- Start from the root node.
- For each value of X, find the set S that should minimize the sum of the impurities of the node in the two child nodes and then choose the split {X∗∈ S∗} that gives the minimum overall values of X and S.
- Exit once the stopping criterion is achieved. Otherwise, go to step 2 for each child node in turn.

2.2.5 Random Forest

Breiman (2001) proposed the algorithm for random forests. It also adds another layer for randomness approach in bagging of words. Along with construction of tree for every sample of data there is a huge change in the way trees get created in regression trees. The standard methods like regression trees involves splitting of nodes using best split within all variables while in random forest each node is split by considering the best randomly chosen predictor's subset.[6] This strategy is robust when viewed as property of over fitting. Above that, it is user friendly since it consists of only two variables involved in the algorithm. The random forests algorithm for classification as well as regression is as follows:

- Draw ntree[6] bootstrap samples from the original data.
- For each of the bootstrap samples, grow anon-pruned classification or regression tree, with the modification viz. Instead of choosing the best split among all predictors at each node, sample mtry[6] of the predictors in a random way and choose the best split from variables.
- The next step is predicting new data by aggregation of the predictions of the ntree[6] i.e., majority number of votes are for classification and average number are for regression.[6]

Based on the training data, estimate can be obtained on error rate by the following steps:

- Predict the data not in the bootstrap sample using the tree grown with the bootstrap sample at each iteration.
- Aggregate the Out Of Bag (OOB) predictions. Calculate the error rate, and call it the OOB estimate of error rate.[6]

3. METHODOLOGY

The proposed methodology for experimentation is shown in Figure 2. The steps involved in sentiment analysis using the approach are described below:

3.1 Stage 1: Data retrieval

In order to provide an exhaustive comparative study of machine learning algorithms, the experiment is based on analyzing the sentiment value of standard datasets obtained from three different sources namely from Cornell University, Stanford University and University of California, Irvine. The datasets under consideration are specific to the chosen domain of movie reviews.

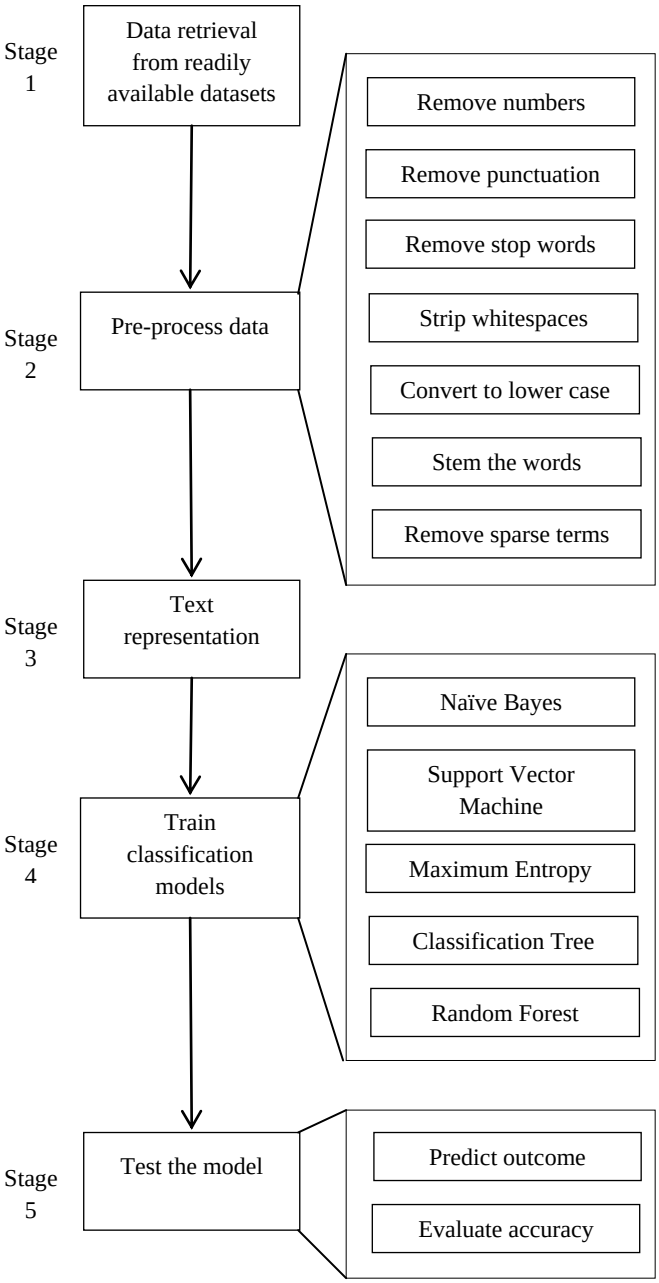

Figure 2: Operational model diagram

3.2 Stage 2: Data pre-processing

Pre-processing stage involves preliminary operations which help in transforming the data for ease of use before the actual sentiment analysis task can be carried out. In order to demonstrate the effect of pre-processing on the classification models, the experiment involves recording results by considering the complete pre-processing step in one approach and eliminating this step in the second approach. Thus two cases are considered: one for non-pre-processed data and the other for pre-processed training data. Pre-processing stage involved following operations:

3.2.1 Remove numbers

Numbers do not play any vital role in determining the orientation of text and hence, they are removed from the data.

3.2.2 Remove punctuation
In the experiment, the punctuation marks are eliminated from text under consideration. Although emoticons created using punctuations can help in predicting sentiment, they can even be used in sarcastic manner. The use of emoticons is not considered and base the experiment only on analysis of words obtained from the sentences.

3.2.3 Remove stop words
Words like pronouns, prepositions, conjunctions, etc. which occur frequently in the data but do not convey any meaningful content [8] or important information regarding the sentiment value of a sentence are called stop words. Stop word removal can help in reducing the memory requirement while classifying the reviews.

3.2.4 Strip whitespaces
Whitespaces do not have any meaningful purpose in the task and thus are stripped from the original text.

3.2.5 Convert to lower case
In order to maintain consistency and map words irrespective of their case, the sentences are converted to lower case.

3.2.6 Stem the words
Many of the words originate from a root word and stemming involves elimination of prefixes and suffixes of words leaving the stem [8] of the considered words. Stemming can significantly reduce the memory load during training and classification.

3.2.7 Remove sparse terms
The terms which have a low occurrence frequency in the dataset are called sparse terms. Based on the data, terms should be considered only when they occur in at least a specific percentage of the documents. The percentage needs to be estimated depending upon how sparse the terms are. This can gravely affect the size of document term-matrix formed in the next step.

3.3 Stage 3: Text representation
The classification algorithms cannot directly take the text under consideration as the input. It is essential to represent the sentences in a format that the algorithms can operate on. For the current study, document term-matrix has been used to represent the text with the weighting scheme of Term Frequency [8]. Both the training and testing data are represented in this format before being used in the latter stages.

3.4 Stage 4: Train Classification models
The experiment focuses on a comparative evaluation of five machine learning algorithms widely used for the classification task of sentiment analysis. The five classification models generated are: Naïve Bayes, Support Vector Machine, Maximum Entropy, Tree and Random Forest. This phase emphasizes on creation of the model based on the training data obtained from the former step in the form of a document term-matrix.

3.5 Stage 5: Test the model
Once the model has been trained in the previous step, the next phase involves predicting the output of the model on testing dataset. The outcomes are the class of the review whether positive or negative. The results include the following attributes:

- True Positives [9]: Positive reviews in the testing data, which are correctly classified by the model as Positive.
- False Positives [9]: Negative reviews in the testing data, which are incorrectly classified by the model as Positive.
- True Negative [9]: Negative reviews in the testing data, which are correctly classified by the model as Negative.
- False Negatives [9]: Positive reviews in the testing data, which are incorrectly classified by the model as Negative.

Based on these recordings, accuracy for the model is calculated as:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$$ [9]

Where,

- TP – True Positives [9]
- FP – False Positives [9]
- TN – True Negatives [9]
- FN – False Negatives [9]

4. EXPERIMENTAL SETUP
The comparative study is conducted in the R data mining language based on the models trained using the RTextTools [11] package and its dependencies like e1071 [5]. Each of the three datasets is considered individually while evaluating the performance of the classification algorithms. The Large Movie Reviews Dataset v1.0 [12] by Stanford University has separate training and testing datasets each containing 25000 reviews. However, the Polarity Dataset v2.0 [13] by Cornell University and IMDb dataset [14] by University of California, Irvine need to be split into training and testing datasets. For this purpose, both the datasets are split in the ratio 3:2 (60% training data and 40% testing data) while maintaining the ratio of positive to negative reviews in the split datasets.

The seed for pseudorandom operations is set to 123 and results can be reproduced with the usage of this seed during experimentation. The experiment involves the use of n-grams of length 1 i.e. unigrams are considered as the features. Further, the threshold used is 0.995 for removing sparse terms which means that only those terms will be retained which occur in 0.5% or more of the reviews in the considered dataset.

For each of the three datasets, the performance of machine learning algorithms is evaluated in two cases based on whether pre-processing was carried out or not. The same can be viewed from the operational model diagram shown in Figure 2 as the operation proceeding directly from stage 1 to stage 3 while entirely skipping stage 2.

5. EXPERIMENTAL RESULTS
Classification models are trained and tested on each dataset independently and the results of the experiments are explained in the following subsections:

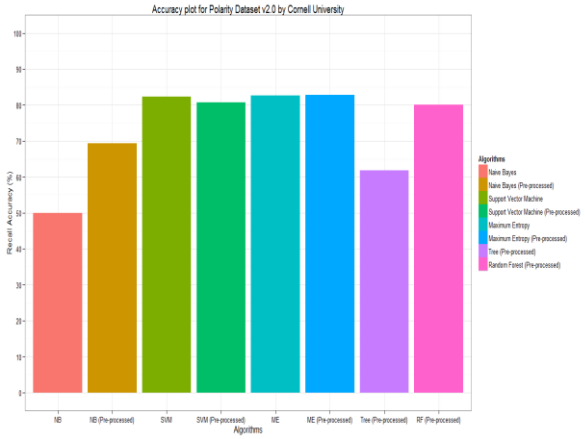

Figure 3: Accuracy plot for Polarity Dataset v2.0

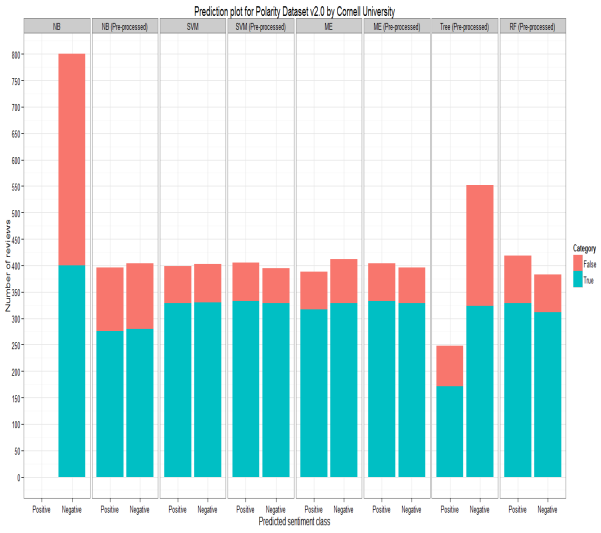

Figure 4: Prediction plot for Polarity Dataset v2.0

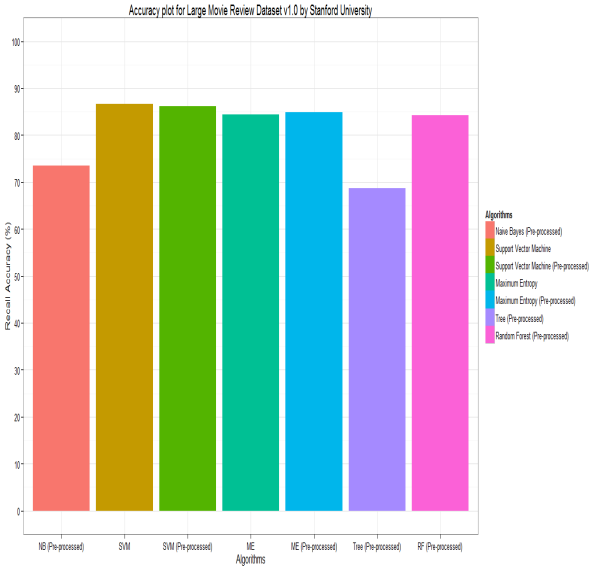

Figure 5: Accuracy plot for Large Movie Review Dataset v1.0

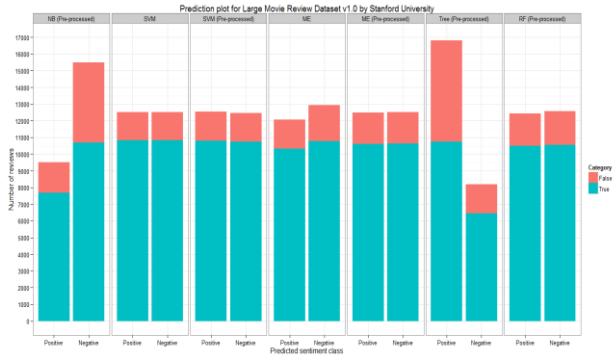

Figure 6: Prediction plot for Large Movie Review Dataset v1.0

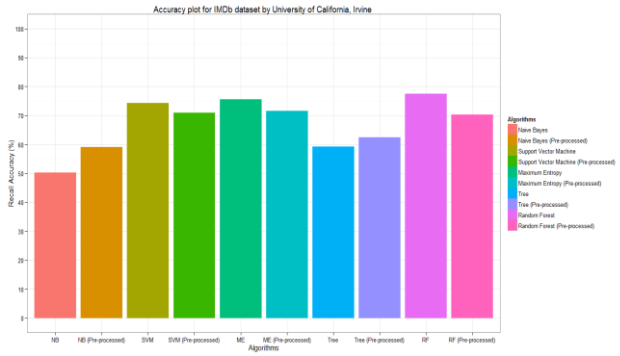

Figure 7: Accuracy plot for Sentiment labeled IMDb Dataset

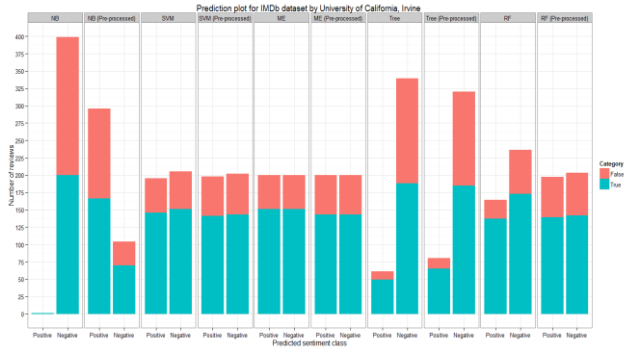

Figure 8: Prediction plot for Sentiment labeled IMDb Dataset

5.1 Polarity dataset v2.0 by Cornell University

From the prediction and accuracy plots shown in Figures 3 and 4, it is observed that the Naïve Bayes classifier produces very poor results without pre-processing of the textual movie reviews. SVM and Maximum Entropy outperform the other algorithms even when the input data is not pre-processed. Both these algorithms benefit from abundance of input data. Tree and Random Forest algorithms produce average results as compared to the other classification models in comparison. The data is too large for Tree and Random Forest to operate without pre-processing the input and hence the results for that task couldn't be obtained.

5.2 Large movie reviews dataset v1.0 by Stanford University

This dataset contains maximum movie reviews from all the datasets under consideration. The number as well as size of the reviews is very large and that is the reason why Naïve Bayes, Tree and Random Forest couldn't be trained on such huge data without pre-processing. But the SVM and Maximum Entropy models could be trained and tested in both the cases. The results obtained for the evaluation on this dataset clearly depict how well the SVM and Maximum Entropy classification models performed on this dataset too. The performances of the algorithms are illustrated in the plots presented in Figures 5 and 6.

5.3 Sentiment labeled sentences (IMDb) by University of California, Irvine

UCI's IMDb dataset [14] is the smallest of the three datasets. Due to small number and size of the reviews, the results of all algorithms could be recorded on pre-processed data as well as on the data which is not pre-processed. As observed in the plots of Polarity Dataset v2.0 [13], Naïve Bayes algorithm performed quite poorly on input which lacked pre-processing. Similar to the other two cases, SVM and Maximum Entropy outperformed the other algorithms while Tree and Random Forest produced average results in this case too. The same can be visualized using Figures 7 and 8.

Table 1.Summarized experimental results table

	Polarity Dataset v2.0 by Cornell University [Testing data size: 800 Positives: 400 Negatives: 400]					Large Movie Review Dataset v1.0 by Stanford University [Testing data size: 25000 Positives: 12500 Negatives: 12500]					IMDB dataset by University of California, Irvine [Testing data size: 400 Positives: 200 Negatives: 200]				
	Acc.	TP	FP	TN	FN	Acc.	TP	FP	TN	FN	Acc.	TP	FP	TN	FN
Naïve Bayes	0.5	0	0	400	400	-	-	-	-	-	0.50	1	0	200	199
Naïve Bayes (PP)	0.69	275	121	279	125	0.73	7679	1826	10674	4821	0.59	166	130	70	34
Support Vector Machine	0.82	328	70	330	72	0.87	10837	1663	10837	1663	0.74	146	49	151	54
Support Vector Machine (PP)	0.81	317	71	329	83	0.86	10791	1748	10752	1709	0.71	141	57	143	59
Maximum Entropy	0.83	333	72	328	67	0.84	10329	1736	10764	2171	0.76	151	49	151	49
Maximum Entropy (PP)	0.83	333	71	329	67	0.85	10605	1876	10624	1895	0.72	143	57	143	57
Tree	-	-	-	-	-	-	-	-	-	-	0.59	49	12	188	151
Tree (PP)	0.62	171	77	323	229	0.69	10747	6059	6441	1753	0.63	65	15	185	135
Random Forest	-	-	-	-	-	-	-	-	-	-	0.78	137	27	173	63
Random Forest(PP)	0.80	329	89	311	71	0.84	10491	1945	10555	2009	0.70	139	58	142	61

The above table (Table 1.) provides a summary of recordings obtained from the experiment. The tabulated observations list the readings as well as accuracies obtained for a specific machine learning algorithm on a particular dataset. The abbreviations used in the table are explained as follows:

- Acc. – Accuracy
- TP – True Positives [9]
- TN – True Negatives [9]
- PP – After Pre-processing data
- FP – False Positives [9]
- FN – False Negatives [9]

6. CONCLUSION

The research paper focuses on sentiment analysis of movie reviews obtained from three standard datasets. Machine learning has been widely used for sentiment analysis and so five popular classification algorithms of machine learning have been considered namely Naïve Bayes, Support Vector Machine, Maximum Entropy, Classification and Regression Tree, and Random Forest. As its main contribution, the paper provides a comparative evaluation of these five classification models on datasets obtained from three different sources, thereby presenting a thorough study of performance of these algorithms. Results produced also demonstrate the effects of pre-processing of input data on the performance of the classification algorithms. The research does not involve any handcrafted features while training and results have been tabulated to form a basis for further analysis and customization of algorithms as per the task and domain.

With reference to the future work, the intention is to consider different feature selection methods and N-gram lengths in order to demonstrate their effect on the performance of considered classification algorithms. In addition, the field of feature learning and deep learning provide an alternative approach for solving the problem of sentiment analysis. Extending the scope of the research to deep learning is also being considered.

7. ACKNOWLEDGMENTS

The authors are grateful to Project Mentor and Asst. Professor, Mrs. Pallavi Kulkarni for continued support and guidance during the research.

8. REFERENCES

[1] Pang, B., Lee, L.: "Opinion Mining and Sentiment Analysis", in "Foundations and Trends in Information Retrieval", Volume 2, Issue 1-2, January 2008, pp. 1-135.

[2] P.Walia, Marisha, V.K.Singh, and M.K,Singh, "Evaluating Machine Learning and Unsupervised Semantic Orientation Approaches for Sentiment Analysis of Textual Reviews", 2012 IEEE International Conference on Computational Intelligence and Computing Research.

[3] B.Ripley. tree: Classification and Regression Trees, 2012. URL http://CRAN.R-project.org/package=tree. R package version 1.0-31. [p7]

[4] T.P.Jurka. maxent: An R package for low-memory multinomial logistic regression with support for semi-automated text classification. The R Journal, 4(1): 56-59, June 2012. [p7]

[5] D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, and F. Leisch. e1071: Misc Functions of the Department of Statistics (e1071), TU Wien, 2012. URL http://CRAN.R-project.org/package=e1071. R package version 1.6-1. [p7]

[6] A. Liaw and M. Weiner, Classification and regression by randomForest. R News, 2(3): 18-22, 2002. [p7].

[7] F. Neri, C. Aliprandi, F. Capeci, M. Cuadros and T. By, "Sentiment Analysis on Social Media," Advances in Social Networks Analysis and Mining (ASONAM), 2012 IEEE/ACM International Conference, Istanbul, 2012, pp. 919-926.

[8] J.Akaichi, Z. Dhouioui, M. J. Lopez-Huertas Perez, "Text mining facebook status updates for sentiment classification", 2013 17th IEEE International Conference on System Theory, Control and Computing (ICSTCC), pp. 640-645.

[9] Nagamma P, Pruthvi H. R., Nisha K. K., Shwetha N. H., "An Improved Sentiment Analysis Of Online Move Reviews Based On Clustering For Box-Office Prediction", 2015 IEEE International Conference on Computing, Communication and Automation (ICCCA2015), pp. 933-937.

[10] Eduard H. Hovy, "What are Sentiment, Affect, and Emotion? Applying the Methodology of Michael Zock to Sentiment Analysis",Book Part 1, 2015, pp. 13-24.

[11] T. P. Jurka et al., "RTextTools: A Supervised Learning Package for Text Classification", The R Journal Vol. 5/1, June 2013, pp. 6-12.

[12] Maas et al., "Learning Word Vectors for Sentiment Analysis", in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, June 2011, pp. 142-150.

[13] Bo Pang and Lillian Lee, "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts", in Proceedings of the ACL, 2004.

[14] Kotzias et al., 'From Group to Individual Labels using Deep Features', KDD 2015.

[15] Anna Jurek, Yaxin Bi, Maurice Mulvenna, "Twitter Sentiment Analysis for Security-Related Information Gathering", 2014 IEEE Joint Intelligence and Security Informatics Conference.

[16] Sagar Bhuta, Avit Doshi, Uchit Doshi and Meera Narvekar,"A Review of Techniques for Sentiment Analysis Of Twitter Data", 2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT).