International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 07 | July 2020 www.irjet.net p-ISSN: 2395-0072

Support Vector Machine versus Naive Bayes Classifier: A Juxtaposition of Two Machine Learning Algorithms for Sentiment Analysis

Ananya Arora[1], Prayag Patel[1], Saud Shaikh[1], Prof. Amit Hatekar[2]

[1]*Undergraduate Research Scholar, Department of Electronics and Telecommunication, Thadomal Shahani Engineering College, Mumbai-50, Maharashtra, India*

[2]*Assistant Professor, Department of Electronics and Telecommunication, Thadomal Shahani Engineering College, Mumbai-50, Maharashtra, India*

Abstract - *This paper presents an observation-based comparison between Naive Bayes and Support Vector Machine regarding sentiment classification. We discuss about data preprocessing, resulting models, the context in which the models achieve better results, elucidating the different facets of performance to enhance the understanding of the results achieved and helping businesses understand more than just one or two important metrics like accuracy, by empirically demonstrating the results of this paper. We make use of TF-IDF vectorizer to reflect how important a word is to a document in a collection or corpus. We also make use of cross -validation and grid search to find the best model that fits the data. The discussion of the outputs is more convoluted than what the results convey. We attempt to explain which model is better suited for particular use cases and we also present future scope where we discuss how better results can be obtained.*

Key Words: **Machine learning, Opinion mining, Sentiment analysis, Sentiment classification, Naive Bayes, Support Vector Machine, Comparative study**

1. INTRODUCTION

Sentiment analysis is the field of study that analyses people's sentiments based on their opinions, suggestions and attitudes, via feedbacks for products, services, organizations, individuals, movies, etc. It uses Natural Language Processing and data mining techniques for extracting opinions from texts. Machine learning techniques have been applied to automatically identify the information content present in the text. Sentiment analysis is driven by the increase in the usage of internet in the recent years and the public opinion exchange. Over the years, there has been a lot of research about analysing and classifying textual data, which has led to the development of sentiment analysis and classification systems. [1] (Pang, Lee, & Vaithyanathan, 2002) used movie reviews to train an algorithm that detects sentiment in text. A good source for this kind of work is the movie reviews because authors clearly express an opinion and reviews are accompanied by ratings that makes it easier to train learning algorithms on this data. The rapid growth of the internet has facilitated newer ways for the general public to post their opinions online. This has primarily led to the production of large amounts of content, particularly online, rich in user opinions, sentiments and emotions. Recently, many internet sites have offered reviews of things like books, cars, snow tires, vacation destinations, movies etc. Informers describe the items in some detail and evaluate them as good/bad, liked/disliked and positive/negative clarifying their recommendation towards the same. An example of people expressing their reviews is- "I love the movie because it provides immense knowledge!" In sentiment classification problems, movie review mining is a challenge. The inspiration for this work has come from studies in classification. Similarly, as an online presence has become quintessential for most businesses. It is important that they try and use the data obtained from reviews, not just to improve their products and services, but also to generate business leads by gauging customer sentiments pertaining to their products and services. Businesses have come to realize the importance of sentiment analysis and thus demand amelioration of the pre-existing methods and the discussion of scope between them. At the same time, businesses do not fully understand the distinctions and nuances that come along with sentiment analysis. This demand has driven us to understand and compare in-depth, the already established methods such as Naive Bayes Classifier and Support Vector Machine. In this paper the main contributions of our work are: (i) An elaborate explanation of the two computationally efficient approaches: Naive Bayes (NB) and Support Vector Machine (SVM); (ii) Determining the better model of the two, not just in a general sense i.e. not just the accuracy score, but also in terms of other metrics (like precision, recall and F1 score) in order to make an informed business decision; (iii) a performance evaluation of the two on the Internet Movie Database (IMDB), Amazon customer reviews and Yelp reviews; (iv) Presenting a full account of the terminologies surrounding the machine learning algorithms so that businesses realize what is more important for them; (v) Proposing possible

avenues that might be undertaken to improve the models implemented in this paper.

This paper is organized as follows. In section 2, we discuss research work in the sentiment analysis domain. In order to approach a standard comparative context, in section 3 we discuss the methodology we have used to analyse sentiments and we also discuss the two popular techniques, Naive Bayes and Support Vector Machine. Section 4 presents an overview of the implementation of the two techniques. In section 5, we present the outputs of our experiment. The results are discussed in section 6. In section 7 we discuss our conclusions. Section 8 proposes possible avenues that might be undertaken to improve the models implemented in this paper.

2. RELATED WORK

Various techniques have been used for sentiment classification. [2] in their paper have elaborately discussed the two supervised machine learning algorithms: K-NN and Naive Bayes and compared the overall accuracy, precision as well as the recall values. It was seen that in case of movie reviews, Naive Bayes gave far better results than K-NN but for hotel reviews, these algorithms gave lesser, almost similar accuracies. In the proposed system by [3], Naive Bayes classifier and Neural Network classifier were combined for sentiment classification. As a result, the movie reviews were classified into positive or negative polarities of sentiment. The accuracy of sentiment analysis was increased up to 80.65% by combining the two classifiers for unigram feature on the movie review dataset. [4] presents an empirical comparison between SVM and ANN regarding document-level sentiment analysis. They discuss requirements, resulting models and contexts in which both approaches achieved better levels of classification accuracy. They have adopted a standard evaluation context with popular supervised methods for feature selection and weighting in a traditional bag-of-words (BOW) model. Except for a few unbalanced data contexts, their experiments indicated that ANN produce superior or at least comparable results to SVM's. Specially on the benchmark dataset of Movies reviews, it was observed that ANN outperformed SVM by a statistically significant difference, even on the context of an unbalanced data. Their results have also confirmed some potential limitations of both the models.
[5] worked on a new approach on sentiment analysis by first determining whether an expression is neutral or not by tracking the opinions of both the users and non-users and their ratings on products and services.

3. METHODOLOGY

This section explains the general process of sentiment classification and offers a detailed explanation of the flow of the experiment performed.

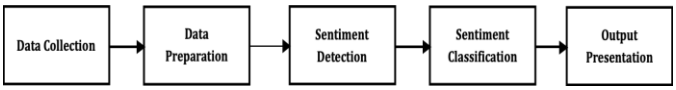

Fig -1: Sentiment analysis process

The first step is data collection. In this paper, a benchmark dataset is being used. The data has to be then prepared for predictive modeling. There are 2 important things to take care of. Firstly, the text from the document has to be broken down/converted to words usually called tokens. The tokens are given numerical values because the machine learning models can only deal with numerical entries. This method of extracting words and assigning them numerical values is called vectorization or feature selection. Secondly, words that do not contribute to the overall meaning of the sentence and usually lend support or structure to the sentence have to be removed so that these words do not skew the results. In this paper, data has been prepared using scikit's inbuilt libraries (tfidf vectorizer, nltk, stop words). Sentiment detection and classification is done with the help of two simple yet powerful machine learning models namely SVM and NB. The processed and prepared data is passed on to these models and the best model is obtained in both cases.

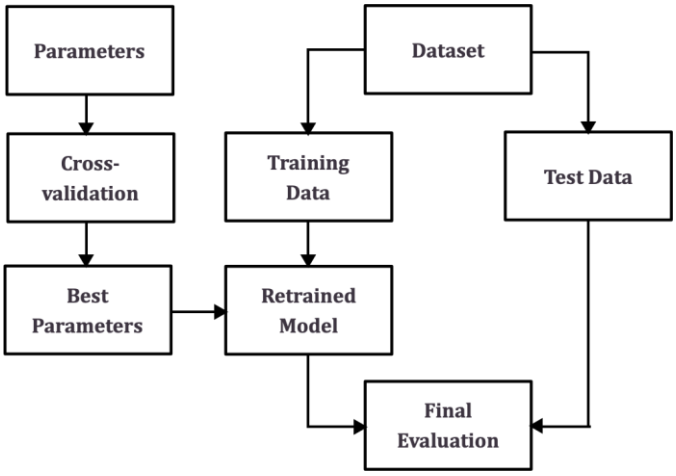

Fig -2: Flowchart for the evaluation process

Cross-validation and Grid Search were conducted to tune both models and determine the optimal hyperparameters that result in the best model. It is used to assess the predictive performance of the models and to evaluate how they perform outside the test data. While trying to fit a model into a training dataset, we use cross-validation techniques. Grid search helps in determining the best parameters or coefficients for a given model. In our case, for SVM it can be kernel, C value, Gamma value. For Naive Bayes it can be the Laplace coefficient. Once this is done, the metrics are evaluated on all the three datasets used. This dataset was chosen mainly because of the versatility.

It has reviews or texts from three completely different use cases – movie reviews, product reviews (Amazon) and service reviews (Yelp) This helps us evaluate which model does well in which of the three scenarios.

3.1 Support Vector Machine

Support Vector Machine (SVM) is a popular supervised machine learning model that is used for classification and prediction of unknown data. It is asserted by several researchers that SVM is a very accurate technique for text classification. It is also widely used in sentiment classification. For instance, if we have a dataset in which data is pre-labeled into two categories: positive and negative reviews, then we can train a model to classify new data into these two categories. This is exactly how SVM works. It is the model that we train on a dataset, so it can analyze and classify unknown data into the categories that were present in the training set. SVM is a linear learning method. It finds an optimal hyper-plane to differentiate two classes. Being a supervised classification model, it tries to maximize the distance between the closest training point and either class so as to achieve better classification performance on test data.

The process for classification functions is as follows:

- It takes the labeled sample of data, and draws a line separating the two classes. This line is called the decision boundary. The solution is based only on those training data points which are really close to the decision boundary. The data points are called Support Vectors. For example, if we are categorizing movie reviews (in our case), one side of the boundary will have positive reviews while the other side has negative reviews.
- Now when new data needs to be classified, it goes either into the left or right side of the decision boundary. Depending on which side the data enters, it is classified under that category.

To classify our data with the best precision, we need to split the two categories such that the decision boundary separates the two classes with maximum space between them.

3.2 Naive Bayes Classifier

Naive Bayes (NB) is an algorithm for binary and multi-class classification problems. The technique is easiest to understand when it is stated using binary or categorical input values. It is called Naive Bayes or Idiot Bayes theorem. It is because the calculation of the probabilities for each hypothesis are simplified to make their calculation tractable. [4] Naive Bayes is a probabilistic learning method that assumes terms occur independently. Given a collection of N documents $\{d\}_{j=1}^{N}$, where each document is represented as a sequence of T terms $d_j = \{t_1, t_2, ..., t_T\}$, the probability of a document d_j occurring in class c_k is given as:

$$P(c_k|d_j) = P(c_k) \prod_{i=1}^{T} P(t_i|c_k) \quad (1)$$

where $P(t_i|c_k)$ is the conditional probability of term ti occurring in a document of class c_k and $P(c_k)$ is the prior probability of a document occurring in class c_k. $P(t_i|c_k)$and $P(c_k)$ are estimated from the training data.

The representation of Naive Bayes is probabilities. A list of probabilities is stored to file for a learned Naive Bayes model which include:

- Class Probabilities: The probabilities of each class in the training dataset.
- Conditional Probabilities: The conditional probabilities of each input value given each class value.

Learning a Naive Bayes model from your training data is fast. Training is fast because only the probability of each class and the probability of each class given different input (x) values need to be calculated. No coefficients need to be fitted by optimization procedures.

3.3 Terminologies

- Kernel Function: Sometimes, the classes cannot be separated linearly and thus we use kernel function to transform the input space to a higher-dimensional space. This is done in order to make the data differentiable linearly. It takes a 1-D input and converts it into a 2-D output.
- SVM Parameter Tuning: C is also known as the penalty parameter. It tells our algorithm, how much we care about misclassified points. A high value for C tells the algorithm that we care about classifying all data accurately. If you increase the C parameter, you are betting that the future data will be further away from the points that you trained the model on. A larger C value creates finer boundaries between classification areas. In the RBF kernel and sigmoid model, a larger C value improves the accuracy of the untuned RBF kernel model. Gamma is another tuning parameter like C. Gamma is a parameter for nonlinear hyperplanes. The greater the value of gamma, the more it tries to fit the training set. It leads to overfitting as the classifier tries to perfectly fit the data. The larger the gamma, the narrower the gaussian bell is. Gamma adjusts the curvature of the decision boundary.
- Text Vectorization: Machine Learning algorithms usually deal with numbers, while language is text. This is one hurdle faced by algorithms in Natural language Processing. We need to perform text vectorization. This is a process that transforms text into numbers. There are several different

algorithms for text vectorization, and all differently affect the output, hence you must choose one that delivers the results you are aiming for.

- Bias and Variance: We must know two prediction errors bias and variance while discussing model prediction. Our model should be able to minimize both these errors, to build accurate models and to avoid underfitting and overfitting. The difference between our model's average prediction and the correct value that we are aiming to predict, is called Bias. The higher the bias, the less attention the model pays to the training data. This leads to high error on the test and train data. Variance is the value that tells us the spread of the data. Models with higher variance pay extra attention to training data and thus aren't able to generalize on the test data. Such models will perform very well on test data but will be erroneous on test data.

- Cross-validation: We might want to use part of our data to test the model. But, reducing the training data will lead to underfitting and loss of patterns and trends. This leads to errors and high bias. To prevent this, we require a method that not only provides enough data for training the model, but also leaves sufficient data for validation. This is where cross-validation comes into the picture. It is used to judge the predictive performance of the models and to assess how they perform outside the training sample to an unknown data (also known as test data).The reason behind using cross-validation techniques is that, when we fit a model, we are fitting it to a training dataset. Without cross-validation we only have information on how our model performs to our in-sample data. Ideally, we would like to see how the model performs, on new data, in terms of accuracy of its predictions.

- Roc curve and score: Roc curve is a plot of false positives versus true positives. Each point on the curve represents a sensitivity/specificity pair corresponding to a particular decision threshold. A test with perfect discrimination has a ROC curve that passes through the upper left corner (100% sensitivity, 100% specificity). Therefore, the closer the ROC curve is to the upper left corner, the higher the overall accuracy of the test. Essentially, what this means is that we can calculate true positive rate and false positive rate for a confusion matrix. But, the entries of the confusion matrix depend on the threshold and the way model classifies the entries which is again subject to the parameters specified for each model. This means that we can get a lot of confusion matrices as the thresholds and parameters change. True Positive Rate (TPR) which can be obtained from the confusion matrix measures the proportion of actual positives that are correctly identified as such. On similar lines, False Positive Rate (FPR) which is the probability that a false alarm will be raised that is a positive result will be given when the true value is negative can also be maintained from the confusion matrix. Depending on what is more important to user, a better rate of classifying true positive entries correctly or minimizing the false positives, a bunch of confusion matrices based on different thresholds can be obtained and the user can pick the one best suited for their needs. This is where the ROC curve comes in. Each point on the graph represents value pairs for false positives and true positives (which is basically like new values for a confusion matrix) and just by looking at the curve a decision can be made easily by the user. It is a plot of false positives versus true positives. The roc_auc score provides an aggregate measure of performance across all possible classification thresholds.

- Precision: It is also called the positive predictive value. It is the fraction of relevant information among the extracted information. The higher the precision, the more relevant instances are being extracted by the model.

- Recall: It is also known as sensitivity. It is the fraction of the total relevant instances that were actually fetched. Both precision and recall are based on a measure of relevance.

- Accuracy: Accuracy is the fraction of predictions that the model gets right.

- Versatility: It is the model's ability to adapt to different test data. A model that is versatile, will give an almost equal performance with every dataset.

3.4 TF-IDF Vectorizer

A word vector represents a text document as a list of numbers. One number for each possible word of the corpus. These vectors represent the text of the document. Once you've transformed words into numbers, in a way the machine learning algorithms can understand, the TF-IDF score can be fed to algorithms such as Naive Bayes and Support Vector Machine, greatly improving the results of more basic methods like word counts. Why does this work? Fundamentally, a word vector represents a document as a list of numbers, with one for each possible word of the corpus. Vectorizing a document simply means taking the text and creating one of these vectors, and the numbers of the vectors represent the content of the text. TF-IDF enables us to give us a way to associate each word in a document with a number that represents how relevant each word is in that document. Then, documents with similar, pertinent words will have similar vectors, which is what we are looking for in a machine learning

algorithm. Term Frequency-Inverse Document Frequency (TF-IDF) is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. This is done by multiplying two metrics: how many times a word appears in a document, and the inverse document frequency of the word across a set of documents. It has many uses, most importantly in automated text analysis, and is very useful for scoring words in machine learning algorithms for Natural Language Processing (NLP) [6].

The classic TF-IDF $_{(t, d)}$ (Manning et al., 2008) assigns to term t a weight in document d as

$TF - IDF_{(t, d)} = TF_{(t, d)} \times IDF_{(t)}$; where: $IDF_{(t)} = \log \frac{N}{DF_{(t)}}$ (2)

$TF_{(t, d)}$ is the number of occurrences of term t in document d, N is the number of documents in the collection and $DF_{(t)}$ is the number of documents in the collection that contain term t.

4. EXPERIMENT

This section includes the details of the datasets and the metrics involved for measurements. It also presents the confusion matrices for both NB and SVM when applied to the three datasets (Movies, Amazon and Yelp).

4.1 Datasets

We conducted our research on Sentiment Labelled Sentences Data Set taken from the UCI machine learning repository. The dataset was initially created for the Paper [7]. Primarily, it is a collection of reviews obtained from three sources. It consists of 1000 reviews each, obtained from the official websites of IMDB for movie reviews, Amazon for product reviews and Yelp for service reviews. Each dataset is equally divided into 500 positive and 500 negative reviews which makes all the three datasets well balanced. These reviews were in turn selected randomly for larger datasets of reviews. The selected sentences have a clear positive or negative connotation. The goal was to have zero neutral sentences to be selected. For each review, positive reviews are labelled 1 and negative reviews are labelled 0. Each text file is a tab delimited file where the review text is followed by a binary value of 0 or 1 indicating the review type.

Table-2 gives a brief description about the three datasets.

Table -1: Description of datasets

Total no. of reviews for each dataset = 1000	
Type of review dataset	**Total no. of words**
IMDB movie reviews	15043
Amazon product reviews	10470
YELP service reviews	11114

4.2 Algorithm

Input: Review dataset
Output: Metric values and Graphs
Steps**:**

1. Importing data
2. Feature extraction
3. Setting up a pipeline
4. Train – Test – Split
5. Grid search
6. Get best parameters
7. Calculating metrics
8. Plotting the graphs

4.3 Performance Measurements

Accuracy = $\frac{TP+TN}{P+N}$; where: P = TP+FN and N = TN+FP

Recall = $\frac{TP}{TP+FN}$; Precision = $\frac{TP}{TP+FP}$;

F-1 Score = $\frac{2TP}{2TP+FP+FN}$;

* P = The number of real positive cases in data
* N = The number of real negative cases in data
* TP = True Positive; TN = True Negative
* FP = False Positive; FN = False Negative

A confusion matrix aids the evaluation of performance of a classification algorithm or a classifier. The calculations of the above-mentioned metrics, i.e. accuracy, recall, precision and f1-score can be easily done using the confusion matrix and the results help in identifying the errors in the classifier. Hence, the confusion matrix is also known as error matrix.

Table -2: Structure of confusion matrix

		Predicted Class	
		P	**N**
Actual Class	**P**	TP	FN
	N	FP	TN

Following are the confusion matrices obtained by applying the two sentiment analysis methods of SVM and NB on the three review datasets.

Table -3(a): Confusion matrix for SVM (Movie)

		Predicted Class	
		P	N
Actual Class	P	53	18
	N	9	70

Table -3(b): Confusion matrix for NB (Movie)

		Predicted Class	
		P	N
Actual Class	P	57	14
	N	14	65

Table -4(a): Confusion matrix for SVM (Amazon)

		Predicted Class	
		P	N
Actual Class	P	80	28
	N	16	76

Table -4(b): Confusion matrix for NB (Amazon)

		Predicted Class	
		P	N
Actual Class	P	78	30
	N	13	79

Table -5(a): Confusion matrix for SVM (Yelp)

		Predicted Class	
		P	N
Actual Class	P	87	21
	N	23	69

Table -5(b): Confusion matrix for NB (Yelp)

		Predicted Class	
		P	N
Actual Class	P	83	25
	N	21	71

5. EXPERIMENTAL RESULTS

This section includes the tables and graphs obtained on performing the sentiment classification on the three datasets using Naive Bayes classifier and Support Vector Machine.

Table -6: Calculated metrics for individual datasets

Metrics	Movie Reviews Dataset		Amazon Reviews Dataset		Yelp Reviews Dataset	
	SVM	NB	SVM	NB	SVM	NB
Roc_auc Score	0.896	0.886	0.867	0.897	0.836	0.866
Precision	0.796	0.823	0.731	0.732	0.767	0.739
Recall	0.886	0.823	0.826	0.891	0.75	0.771
Accuracy	0.82	0.813	0.78	0.8	0.78	0.77
F1-Score	0.838	0.823	0.776	0.803	0.758	0.755

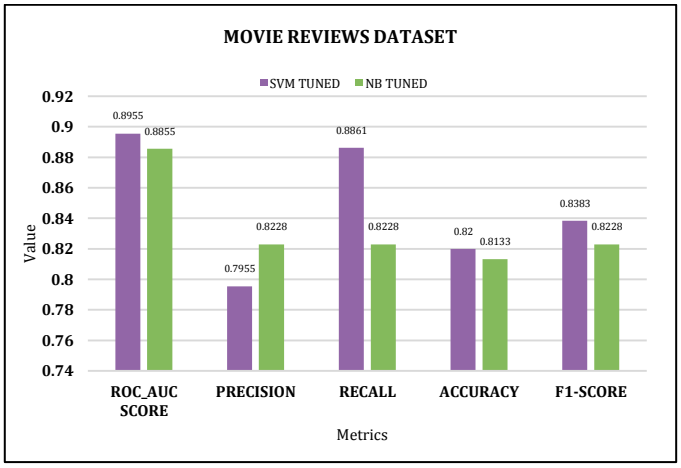

Chart -1: Comparison of SVM and NB for Movie reviews

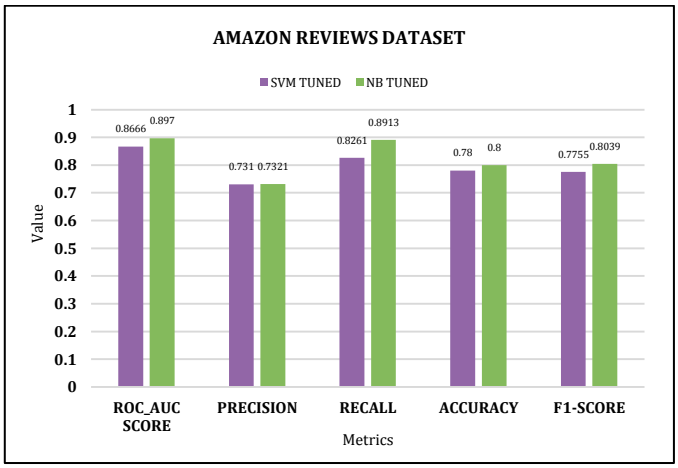

Chart -2: Comparison of SVM and NB for Amazon reviews

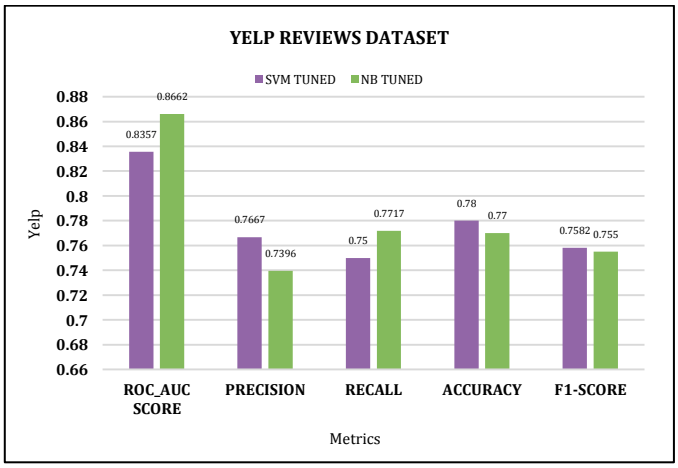

Chart -3: Comparison of SVM and NB for Yelp reviews

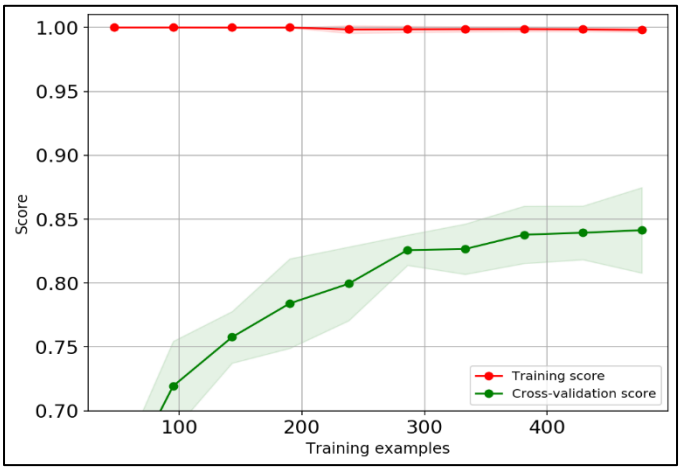

Chart -4: Cross-validation Score of SVM for Movie reviews

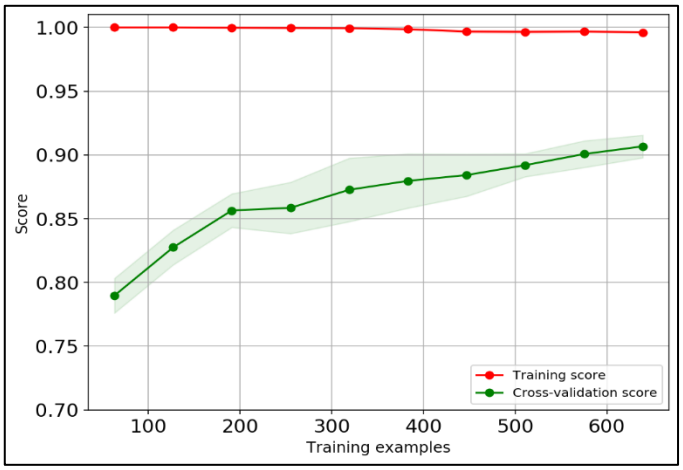

Chart -7: Cross-validation Score of NB for Amazon reviews

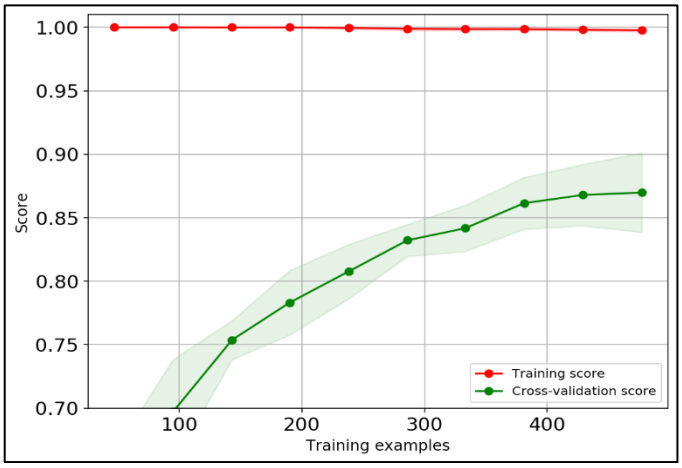

Chart -5: Cross-validation Score of NB for Movie reviews

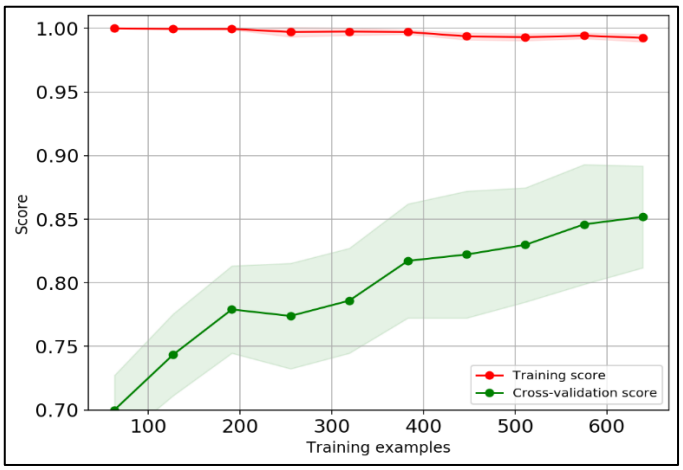

Chart -8: Cross-validation Score of SVM for Yelp reviews

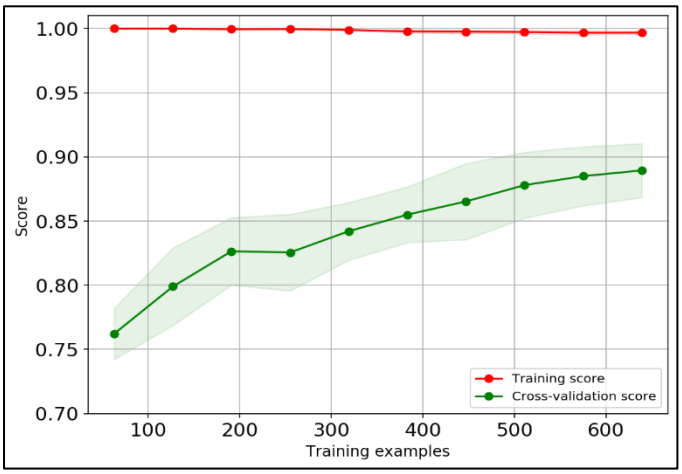

Chart -6: Cross-validation Score of SVM for Amazon reviews

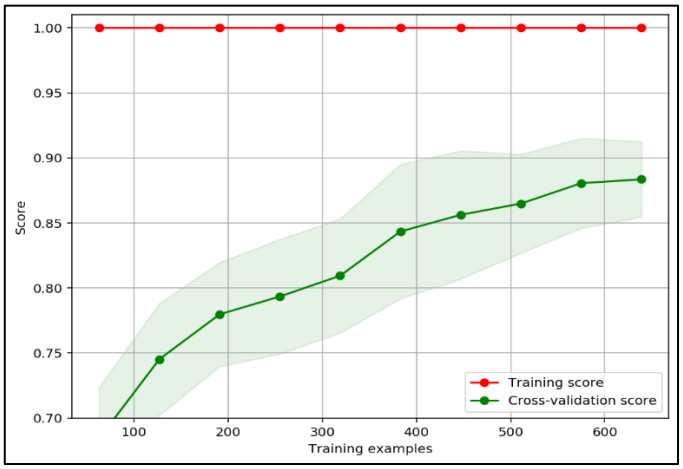

Chart -9: Cross-validation Score of NB for Yelp reviews

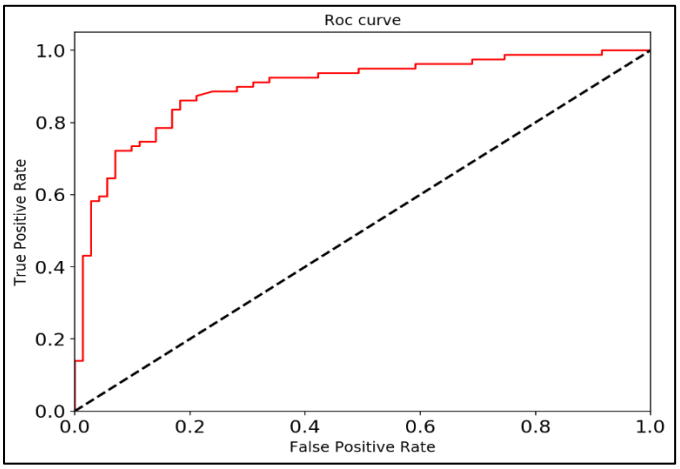

Chart -10: Roc curve of SVM for Movie reviews

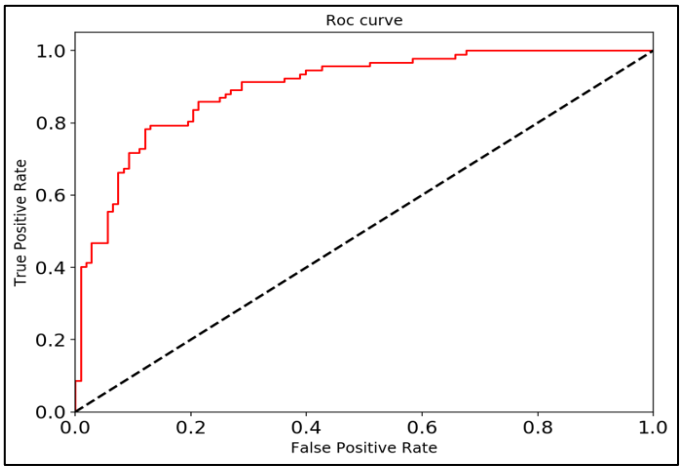

Chart -13: Roc curve of NB for Amazon reviews

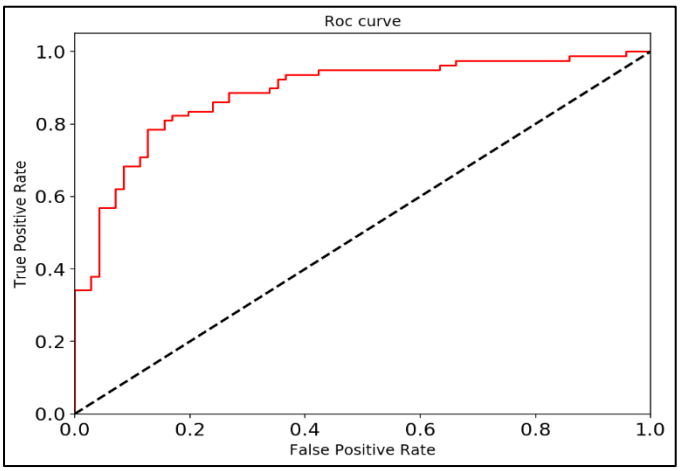

Chart -11: Roc curve of NB for Movie reviews

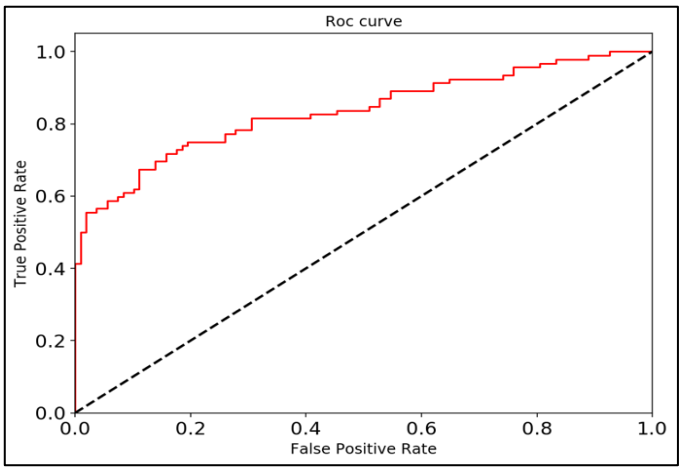

Chart -14: Roc curve of SVM for Yelp reviews

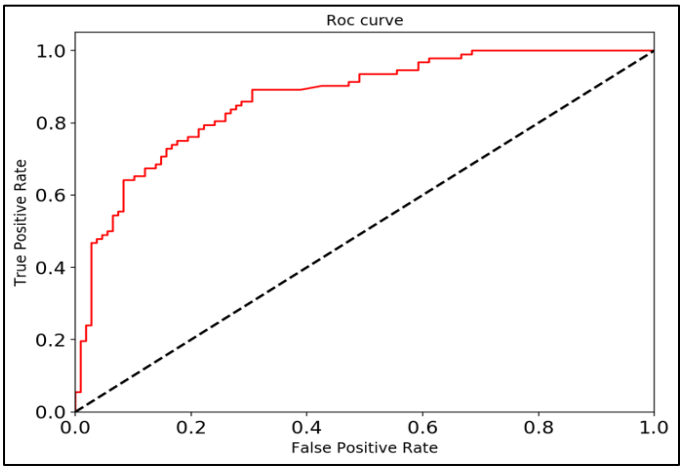

Chart -12: Roc curve of SVM for Amazon reviews

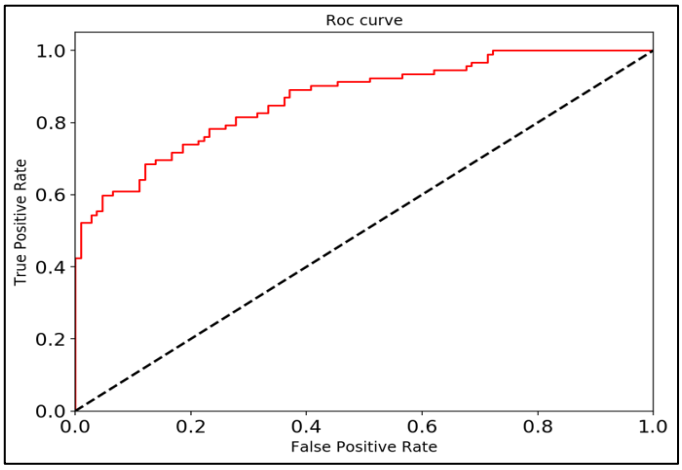

Chart -15: Roc curve of NB for Yelp reviews

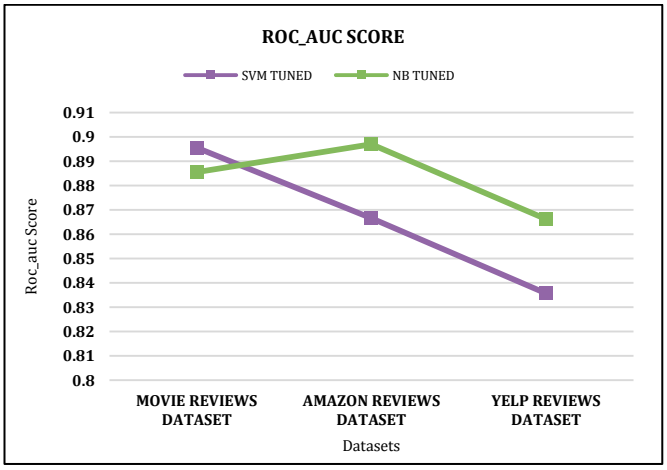

Chart -16: Roc_auc Score of SVM and NB

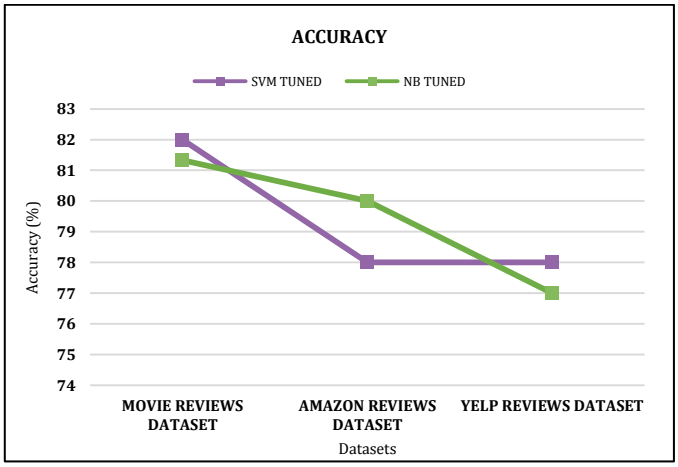

Chart -19: Accuracy of SVM and NB

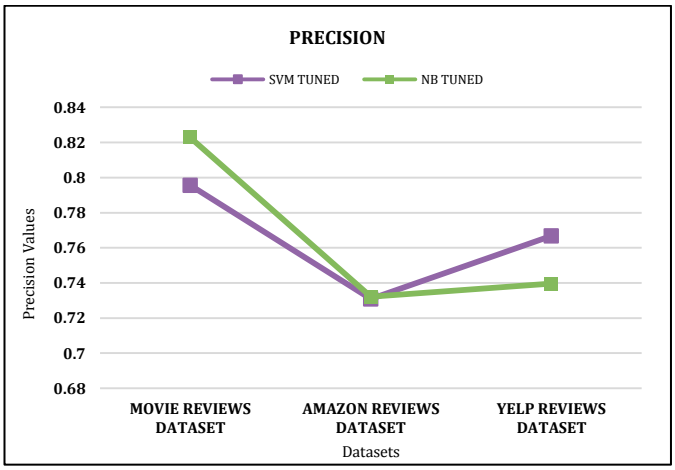

Chart -17: Precision of SVM and NB

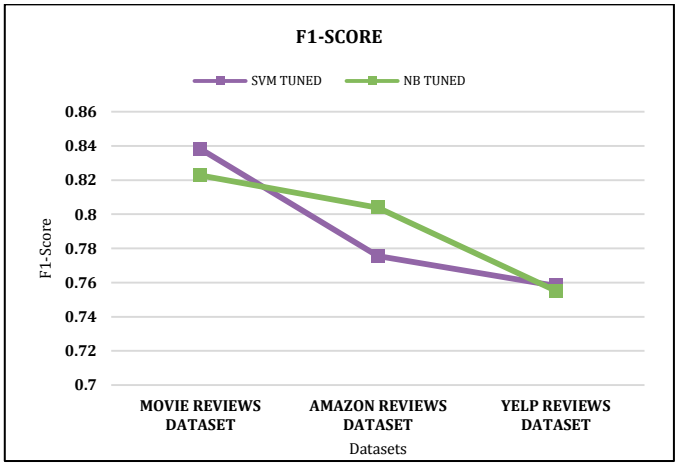

Chart -20: F1-Score of SVM and NB

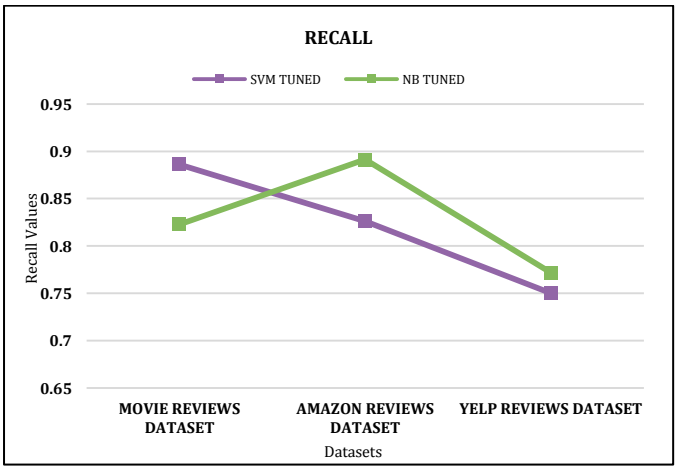

Chart -18: Recall of SVM and NB

Table -7: Average values of metrics for all three datasets

Metrics	All 3 Databases	
	SVM	NB
Roc_auc Score	0.8659	0.8829
Precision	0.7643	0.7648
Recall	0.8207	0.8286
Accuracy	0.7933	0.7944
F1-Score	0.7907	0.794

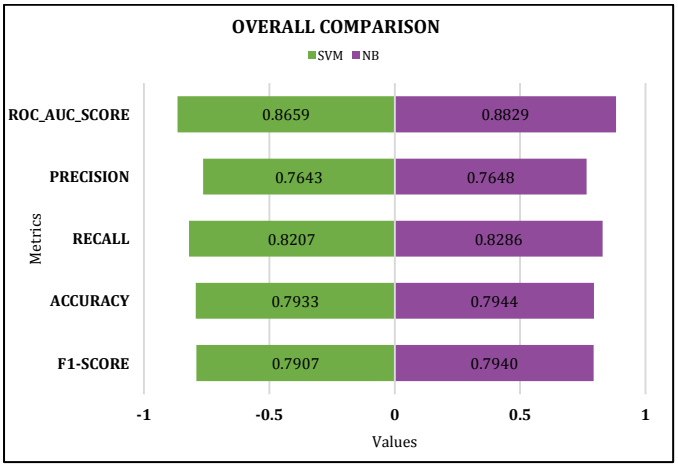

Chart -21: Versatility of SVM and NB

6. DISCUSSION OF RESULTS

This section discusses the charts and tables displayed in the previous section.

- Table-6 presents a tabular representation of the outputs that we have obtained. It shows the metrics for both the methods when applied over the three datasets individually.
- Chart 1 shows all metric values for the Movie reviews dataset. This bar graph presents a broader comparison of the two models on this particular dataset. We can observe that SVM outshines NB in every parameter except precision.
- Chart 2 shows all metric values for the Amazon reviews dataset. This bar graph presents a broader comparison of the two models on this particular dataset. We can observe that NB outperforms SVM in each and every metric.
- Chart 3 shows all metric values for the Yelp reviews dataset. This bar graph presents a broader comparison of the two models on this particular dataset. We can observe that the roc_auc score and recall of NB is better whereas accuracy, precision and f-1 score of SVM are better. The ratio of performance of NB to SVM in the Yelp dataset is 2:3.
- From Chart 4, 5, 6, 7, 8 and 9, we can observe that for all three datasets, the cross-validation score of NB increases more with time than SVM.
- From Chart 10, 11, 12, 13, 14 and 15, we can observe that for each dataset, NB's curve is closer to the upper left corner than SVM's curve.
- From Chart 16, we can observe that NB has a higher roc_auc score for Amazon and Yelp reviews datasets while SVM has a higher score for Movie reviews dataset.
- From Chart 17, we can observe that the precision of NB is better than SVM for Movie reviews dataset while the opposite is true for the Yelp reviews dataset. For Amazon reviews dataset, both methods have almost the same precision.
- From Chart 18, we can observe that NB has a higher recall for Amazon and Yelp reviews datasets while SVM has a better recall for Movie reviews dataset.
- From the accuracy graph (Chart 19) we can observe that SVM has a higher accuracy (gets more predictions right) for the Movie and Yelp reviews dataset while NB has a higher accuracy for the Amazon reviews dataset.
- The harmonic mean of precision and recall is the f1-score. It is a measure of the test's accuracy. From Chart 20, we can observe that SVM dominates in terms of f1-score on the Movie reviews dataset while NB has a better f1-score for the Amazon dataset and they both have an almost equal f1-score for the Yelp dataset.
- To compare the versatility of SVM and NB in sentiment analysis, we have found out the average of all parameters of both models over the three datasets. By seeing the results in Chart 21 and Table-7, we can observe that, when an average of performance on three datasets is considered, NB gives a higher value for each metric as compared to SVM.

7. CONCLUSIONS

This section discusses the inference of the charts and tables discussed in the previous section.

The aim of the experiment was to present a detailed study by analyzing and classifying sentiments of reviews (Movies, Amazon, Yelp) using Naive Bayes classifier and Support Vector Machine. The study involved different metrics like accuracy, precision, recall, roc score and f1-score which aided in the yield of a lucid comparison between the two methods. Pertaining to sentiment analysis, in accordance with SVM and NB, following are our findings and contributions:

- The charts indicate that for the Movie reviews dataset, SVM performs better. On the other hand, for the Amazon reviews dataset, NB outshines SVM. For Yelp reviews dataset, NB and SVM were in close competition with SVM performing slightly better than NB.
- NB performs better with increase in the number of training examples as compared to SVM. This can be inferred from the cross-validation charts. With the increase in time and training sets, NB achieved a better cross-validation score than SVM.
- Since the roc curve of NB is closer to the upper left corner for all three datasets, we can conclude that NB has better overall accuracy and a better performance than SVM.
- NB has a higher roc_auc score than SVM for two out of the three datasets, leading us to conclude

that NB provides a higher roc score which in turn indicates better performance.

- Precision is higher, if we use NB on the Movie reviews dataset. It is higher using SVM on the Yelp reviews dataset. For Amazon reviews dataset, both NB and SVM are at par. Thus, we can claim that NB and SVM have similar precisions.

- NB has a higher recall for two out of the three datasets, leading us to conclude that NB has a greater sensitivity to relevant instances in the dataset.
- SVM has a higher accuracy for two out of three datasets. This shows that SVM gets more predictions right as compared to NB.
- F1-score is higher for NB on the Amazon reviews dataset, while it is higher for SVM on the Movie reviews dataset. SVM and NB have a similar f1-score for the Yelp dataset. This leads us to conclude that both methods have a similar f1-score.
- But, while speaking in terms of the average of each parameter over all three datasets, NB leads with a higher recall, accuracy, f-1 score and roc_auc score. Precision for the two are observed to be similar. This proves that NB is more versatile than SVM.

From the points above, we can understand that NB and SVM both have their own individual strengths. The choice of the method should depend on the application and purpose. If your application demands more accuracy, then SVM is preferable. On the other hand, if your application requires a higher recall or versatility then NB is the right choice.

8. FUTURE SCOPE

In this paper, two machine learning techniques were compared, in order to come to a conclusion as to which method performs better. Naive Bayes was marginally better overall, and Support Vector Machine also had a slight edge in some cases. For the future, an ensemble of Naive Bayes and Support Vector Machine can be developed, which combines the strengths of both methods thus leading to increase in performance. From the results of this experiment, it is clear that the two methods are similar in terms of performance and complement each other very well. Additionally, these could be tested on bigger and more nuanced datasets that will further help come to a better conclusion about which method is better for a particular use case and which method does well generally.

ACKNOWLEDGEMENT

We would like to express our gratitude to our professor Amit Hatekar and our parents for their immense support and guidance.

REFERENCES

[1] Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment Classification using Machine Learning. Empirical Methods in Natural Language Processing (EMNLP) (pp. 79-86). Philadelphia: Association for Computational Linguistics.

[2] Dey, L., Chakraborty, S., Biswas, A., Bose, B., & Tiwari, S. (2016). Sentiment analysis of review datasets using naive bayes and k-nn classifier. arXiv preprint arXiv:1610.09982, 54-62.

[3] Dhande, L. L., & Patnaik, G. K. (2014). Analyzing Sentiment of Movie Review Data. International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), 3(4), 313-320.

[4] Moraes, R., Valiati , J. F., & Neto , W. P. (2013). Document-level sentiment classification: An empirical comparison between. Expert Systems with Applications, 621-633.

[5] Wilson, T., Wiebe, J., & Hoffmann, P. (2005). Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. Human Language Technology and Empirical Methods in Natural Language Processing (pp. 347-354). Vancouver: Association for Computational Linguistics (ACL).

[6] Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. New York, USA: Cambridge University Press. doi:10.1017/CBO9780511809071

[7] Kotzias, D., Denil, M., De Freitas, N., & Smyth, P. (2015). From group to individual labels using deep features. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 597-606).