A Novel Approach for Analyzing Sentiment of Customer Reviews using Gensim and TF-IDF

Neha Bhujbal[1], Gaurav Bavdane[1]

[1]*Undergraduate Research Scholar, Department of Electronics and Telecommunication, St. Francis Institute of Technology, Mumbai, Maharashtra, India*

---***---

Abstract - *Today the world is going digital in every aspect of life and this rise in digital is drastically changing the consumer behavior. Every customer prefers going through reviews and then making decisions. Screening through all the reviews available on the web is tedious at the consumer end as well as hampers the process of decision making for various businesses to build customer-centric products. To get authentic results in a fraction of seconds and understand consumer needs better we have developed a method of sentiment analysis on reviews. While selecting keywords from each review one of the main tasks is to remove stop words without losing the prime sentiment of the review. This has been achieved by selecting specific POS tags (adjective, adverb, verb, noun) instead of explicitly removing all the stop words and performing lemmatization using gensim. Further, TF-IDF vectorizer is used to score each word and highlight the word's relevance in each review. For classification two of the most commonly used classifiers, Naïve Bayes and SVM are used. We also trained a Feed Forward Neural Network and all the results are empirically discussed. We have employed cross validation and GridSearchCV to get the best model that fits the data. This process of sentiment analysis will help gauge the bigger picture rather than scrutinizing each review to drive deeper insights and help organizations formulate effective business strategies.*

***Key Words*: Sentiment Analysis, Gensim, TF-IDF, Naïve Bayes, Support Vector Machine, Artificial Neural Networks**

1.INTRODUCTION

These days making decisions regarding buying a new product, watching a movie or visiting a new place has become easy due to the presence of reviews. Review can be positive or negative or neutral. One cannot conclude or decide just by reading top ten reviews as some can be fake or they could be biased so in order to make the final decision one has to read all the reviews present on the respective website which is practically impossible. Even if one is capable of going through a myriad of reviews posted online it will require a lot of time. According to a survey, 82% of consumers say the content of the review has convinced them to make a purchase and have built their trust for the company. The prime focus of this survey was the fact that consumers' decision is influenced by the content of reviews. Therefore, nowadays all the e-commerce websites ask their customers to review the product they purchased. These reviews are then scrutinized, if they turn out to be negative then business process changes are put in place to ensure that the same mistakes are not repeated again and the products are tailored to fill up all the shortcomings stated in the reviews of the previous product. But reading thousands of reviews and then making out conclusions is hectic and also not efficient. If evaluation of reviews is done manually there is no authenticity of the result concluded and would result in big loss to the company if the result turned out to be wrong. So, there should be a technique which automates this long process of reading thousands of reviews and interpreting authentic conclusions which can be trusted. Natural Language Processing (NLP) can be used with Machine Learning (ML) algorithms to achieve this technique. The technique should involve analyzing the sentiments of the review, removing unnecessary content which does not add any real value and then arriving at a conclusion. In this paper, we aim to automate the process of reading content from thousands of reviews. To analyze the sentiment of the review redundant data which does not aid towards analyzing the sentiment should be eliminated. At the same time, valuable information which may otherwise be labelled as redundant in the form of commonly occurring words or stop words needs to be carefully scrutinized. To achieve this, we have employed part-of-speech-tagging (POS tagging) to capture the real sentiment of the review. Instead of explicitly eliminating all the stop words we propose an approach to select only valuable content. To achieve this, we leveraged the functionality of gensim library to perform text lemmatization to filter the adjectives, adverbs, verbs and nouns from the rest of the Corpus (collection of reviews). Then these words are represented in the form of vectors using TF-IDF vectorizer technique. TF-IDF makes calculation easy, helps extract the most descriptive keywords in a document and measures the uniqueness and relevance of the content. Two commonly used classifiers Naïve Bayes and SVM are used to classify the TF-IDF vectors. Also, a Feed-Forward Neural Network is trained and then the results are compared on the basis of their performance. The aim of this work is to provide a novel approach to classify the reviews as negative or positive and give the best suitable model to capture the sentiment of online customer reviews.

The remainder of this paper is structured as follows: the next section, Section 2 describes the related research done in the field of sentiment analysis. Section 3, provides a description of the details relevant to the two datasets being used. Section 4 provides an in-depth explanation of the proposed system architecture which has been divided into multiple stages. An overview of the experimental setup for sentiment classification by employing three techniques, Naïve Bayes (NB), Support Vector Machine (SVM) and Artificial Neural Network (ANN) is provided in Section 5. In section 6, we present the results of our experiment, their evaluation and discussion. Finally, in Section 7 we discuss the conclusions drawn from our work and the scope for further research.

2. RELATED WORK

In recent time, the exploration of ML algorithms for analyzing the sentiment of reviews posted by customers has gained much attention.

In [1], the authors performed sentiment classification based on categorization aspects with positive and negative sentiments. They employed three ML algorithms for classification, i.e., NB, SVM and Maximum Entropy over the n-gram technique. The authors of [2] leveraged the functionality of TextBlob Library in Python for processing review data. They implemented a Feed-Forward Neural Network for the classification task. They further created a neural network graph with three input nodes (positive, negative, neutral) and one hidden layer to study and determine the overall polarity of data. An elaborate discussion and comparison of two supervised ML algorithms i.e., NB and K-NN have been presented in [3]. The model is evaluated using Accuracy, Precision and Recall by varying the size of training dataset of hotel and movie reviews. The experimental results show that NB outperformed K-NN. This paper [4], enlightens the implementation of ML algorithms for processing textual and statistical data. The methodology used highlights the significance of extracting adjectives from customer reviews during the process of training. [5] is mainly focused on comparison of two methods, i.e. ANN and SVM. In ANN two training algorithms are used for pattern recognition, Scaled Conjugate Gradient (SCG) and Resilient Backpropagation (RP). In SVM an open source LIBSVM and radial basis function (RBF) is used to solve the classification problem. [6] presents a comparison of SVM and NB classifiers. In the preprocessing stage inbuilt scikit libraries (tf-idf vectorizer, nltk, stop words) are used. For best results both the models are tuned using Cross-validation and Grid Search.

3. DATASET DESCRIPTION

To conduct this research, one small dataset (1k reviews) and one large dataset (50k reviews) has been considered as described below:

3.1 Amazon Product Reviews

This sentiment labelled dataset is a part of the collection of customer reviews dataset taken from the UCI machine learning repository which was initially created for the research presented in [7]. It consists of 1000 reviews each obtained from three official sources i.e., Amazon, Yelp and IMDB. For the purpose of our research, we randomly selected the Amazon Products Review Dataset which is well balanced with 500 positive and 500 negative reviews. The reviews are label-encoded with positive reviews labelled as 1 and negative reviews as 0. Stored in text file format the reviews are distinguished from respective labels using tab as delimiter.

3.2 IMDB 50k Movie Reviews

This dataset refers to the IMDb movie review sentiment dataset originally introduced by Maas et al. [8] as a benchmark for sentiment analysis. It contains a total of 100,000 movie reviews posted on imdb.com out of which 50,000 unlabeled and the remaining 50,000 are divided into a set of 25,000 highly polar reviews for training and 25,000 reviews for testing. Each of the labeled reviews has a binary sentiment label i.e., "pos" for positive review or "neg" for negative review. In our experiments, we consider only the labelled part of the entire dataset which makes up to 50,000 reviews.

4. PROPOSED METHODOLOGY

This section provides a comprehensive explanation of the flow of the proposed system architecture (Fig -1) right from the data collection phase to the evaluation of the predicted results. The proposed method effectively identifies the polarity of reviews to increase the accuracy of sentiment classification.

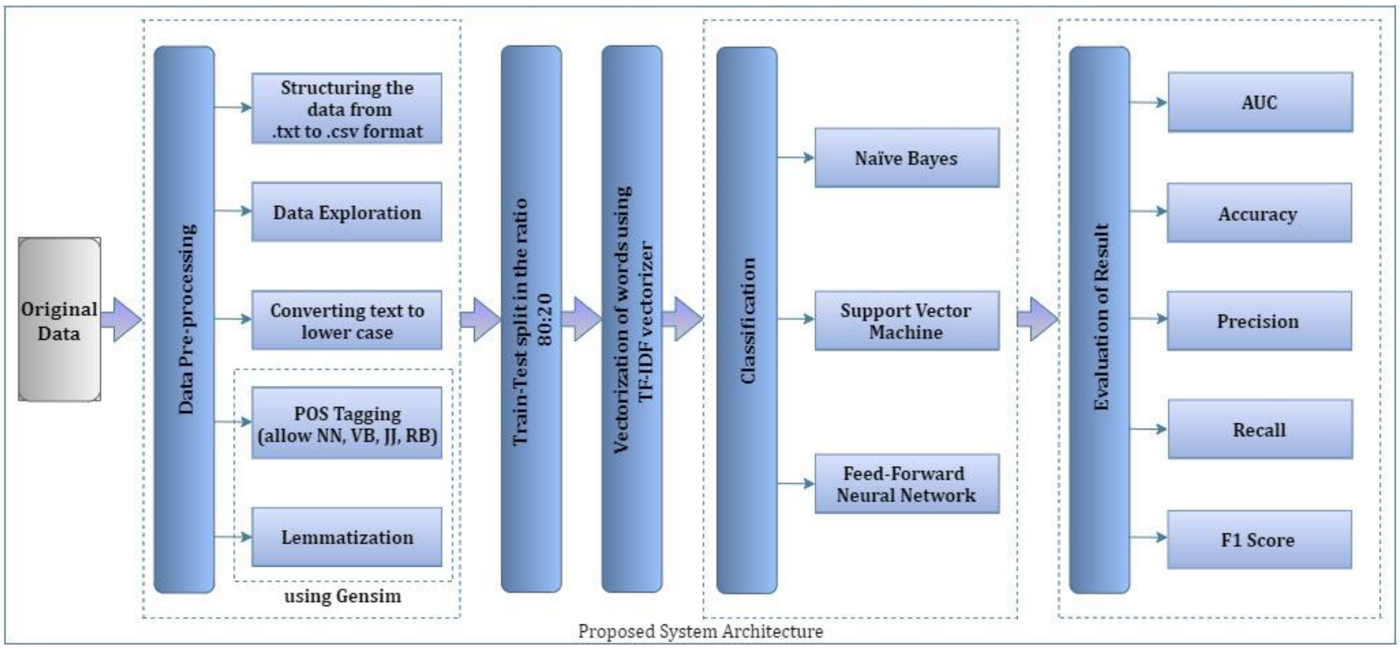

Fig -1: Proposed System Architecture

4.1 Data Pre-processing

Data preprocessing is the most important aspect of ML as it makes data well organized and useful for making the model more efficient. This step gets the data transformed or encoded such that the machine can easily parse it. In our work, it includes the following steps:

Structuring of data: The dataset available was in text format. For better results the data that is fed to the model should be well structured. Thus, to make data handling easy we converted the dataset from text (.txt) format to csv (.csv) using Panda's library. This makes data more readable and handier while accessing.

Data Exploration: Exploration helps us find the null values present in the data as well as gives an idea of how well the reviews are distributed. For this task we used corpus.info() (corpus basically represents the entire data from the csv file) to explore through the entire structured data. No null values were encountered and the reviews were equally distributed between the two classes i.e., positive and negative.

Converting text to lowercase: In NLP it is recommended to convert all text to lowercase due to the fact that all machine learning models interpret words with uppercase letters like 'GOOD', 'Good' and 'good' as three different words. Thus, to avoid misinterpretations and to keep consistency in data we have converted all reviews present in the dataset to lowercase.

POS Tagging: It is important to pass only relevant data to the ML algorithm to make our model more optimal. To achieve this, we used POS Tagging i.e., parts-of-speech Tagging to retain only the relevant information. This task is performed using the gensim library which is commonly used for large text data and is highly efficient. There are a number of POS tags and its subcategories but by default gensim only uses NN (Noun), VB (Verb), JJ (Adjective), RB (Adverb). These are the words which represent the core sentiment of the sentence. Our research focuses on the efficacy of extracting these four POS tags in order to identify the sentiment polarity of customer reviews.

Lemmatization: As the words present in reviews are not always present in their root form, gensim lemmatization helps to get the root word back without changing its meaning. This helps to get the original sentiment of the word and that of the sentence. Gensim plays a vital role as it handles both Lemmatization and POS tagging simultaneously using the lemmatize() method. This method by default retains the words having four specific POS tags i.e NN, VB, JJ, RB in their root form. All the remaining words are scraped.

4.2 Train-Test Split

Once the preprocessing is done, the data is ready to be vectorized. In order to get a good estimate of model performance we split the data into a train and test set where the train set is used for fitting the model and test set is used for evaluating the trained model. In this work we have divided data in 80:20 ratio, with 80% data for training and 20% data for testing.

4.3 Vectorization using TF-IDF

Now the words are vectorized using TF-IDF which stands for Term Frequency-Inverse Document Frequency.

Term Frequency (TF): It measures the frequency of a word in a document (corpus). The weight of a term/word which occurs in a document is simply proportional to the term frequency. Every document has its own term frequency.

$$tf(t,d) = \frac{count\ of\ t\ in\ d}{number\ of\ words\ in\ d} \qquad (1)$$

t = term/word in the corpus
d = document

Document Frequency (DF): is the measure of the importance of the document in which the term is present in the whole set of corpus.

$$df(t) = occurencesof\ t\ in\ documents \qquad (2)$$

Inverse Document Frequency (IDF): Most common words (is, are, of, that, the) will have very high frequency values due to high occurrence, giving them prime importance. As a result, we might end up with poor results. These commonly occurring words or stop words need to be eliminated. IDF diminishes the weight of these stop words (terms) that occur frequently in the document set N and increases the weight of terms that occur rarely. The IDF determines the weight of rare words across all reviews in the corpus.

$$idf(t) = log(\frac{N}{(df+1)}) \qquad (3)$$

TF-IDF: It is simply the multiplication of TF and IDF.

$$tf - idf(t,d) = tf(t,d) \times idf(t) \qquad (4)$$

4.4 Classification

The features extracted using the TF-IDF vectorizer are passed on to NB, SVM and ANN to be classified as having either a positive or negative polarity. Hyperparameter tuning is performed using GridSearchCV to find the optimal model parameters. The working of these classifiers is elucidated below.

Naïve Bayes (NB): It is a probabilistic algorithm that is typically used for classification problems. In accordance with the name, NB applies Bayes' theorem with the "naïve" assumption of conditional independence between each pair of extracted features. In other words, the probability of each feature vector, x = {x1, x2...xn}, is not influenced by other features. The probability of each feature can be estimated as:

$$P(C/x) = P(C).\prod P(xi/C) \qquad (5)$$

P(C) = the probability of each class C
P(xi/C) = conditional probability of feature xi occurring in text of class C

In our work we have utilized one of the two classic NB variants commonly used for text classification known as MultinomialNB. Though being simple and intuitive it performs surprisingly well for many classification tasks as we will observe in Section 6.

Support Vector Machine (SVM): SVM is widely used for sentiment classification and is claimed to be a very accurate technique for text classification. Functionally, an SVM takes the data in relatively lower dimension as input. Then it uses the famous "kernel trick" to compute the relationship between different data points. This involves incorporating kernel functions (e.g., linear, rbf, polynomial, etc.) to calculate the relation between different pairs of data points as if they were in a higher dimensional space and find the best hyperplane to separate different classes. The kernel trick reduces the amount of computation required for SVM by avoiding the math for transformation from lower to higher dimension thus saving time. In our case (review classification) one side of the hyperplane will have positive reviews and the other side will have negative reviews. The performance of SVM highly depends on the selection of parameters (C and gamma) and kernel. We have employed GridSearchCV to find the optimal classification parameters.

Feed-Forward Neural Network (FFN): A FFN is one of the first and simplest ANN in which the information moves only in one direction, from input to the output nodes. Neural nets consist of an artificial network of functions, called parameters, which allow the network to learn and fine-tune itself by analyzing new data. The TF-IDF vectors are fed as input to the network. We have implemented a FFN with 2 hidden layers containing 64 and 32 nodes respectively followed by a sigmoid function as output layer to perform classification.

4.5 Model Evaluation

The designed model is then evaluated in terms of various metrics like accuracy, AUC, precision, recall and f-1 score in order to find the most suitable classification technique.

Accuracy: It is simply defined as the fraction of correct predictions for the test data.

$$Accuracy = \frac{Correct\ Predictions}{Total\ Prediction} \qquad (6)$$

Initially the performance of each model is compared w.r.t. accuracy. Furthermore, various other metrics are taken into consideration.

AUC (Area Under the Curve): It refers to the entire 2-D area under the ROC (Receiver Operating Characteristics) curve. An ROC curve plots the True Positive Rate (TPR) vs. False Positive Rate (FPR) at different classification thresholds. In other words, AUC is a performance measurement metric to check the ability of the model to distinguish between the underlying classes across all possible classification thresholds.

Precision (P): It helps to understand how precise/accurate the designed model is. It tells us out of all the predicted positive values, how many are actually positive.

$$P = \frac{True\ Positive}{True\ Positive + False\ Positive} \qquad (7)$$

It is a good measure to determine when the cost of False Positive alarms is high. In our case, if a review that conveys a negative sentiment (true negative) is identified as positive (predicted positive) then the business might lose on important information provided by the customers.

Recall (R): Also known as sensitivity, it tells us out of all the true positive values, how many are actually labelled correctly.

$$R = \frac{True\ Positive}{True\ Postive + False\ Negative} \qquad (8)$$

This metric is used when there is a high cost associated with False Negatives. For instance, if a positive review (true positive) is identified as negative (predicted negative) then the consequences can be dire for the business.

f1 Score: It is a measure used to strike out a balance between P and R. It is calculated as the harmonic mean of these two metrics.

$$f-1\ score = \frac{2 \times P \times R}{P + R} \qquad (9)$$

This metric takes values between 0 and 1 with 1 indicating perfect P and R.

5. EXPERIMENTS

This section shows the functionality of gensim library for the task of lemmatization and the process of fine-tuning the parameters of the classification stage.

5.1 Should all Stop Words be eliminated?

Commonly occurring words such as articles as well as some verbs are usually categorized as stop words as they do not aid in finding the context or the true meaning of a sentence. At the same time, problems like sentiment classification are much more sensitive to the removal of stop words. Let's consider the following review from Amazon Dataset:

original text = 'I was not impressed by this product.'

Originally, the review has a negative connotation. But, after removing stop words the output is:

resultant text = 'I impressed product'

In this scenario, the overall sentiment of the *resultant text* is converted to positive which is in stark contrast with the *original text.* Therefore, instead of focussing on eliminating the stop words, we propose to filter the *original text* on the basis of their POS-tags and keep only adjectives, adverbs, verbs and nouns using the lemmatization technique of gensim library.

5.2 Text Lemmatization using Gensim

The proposed method is successful to recognize the polarity of natural language text by utilizing specific POS categories which are identified in Table -1.

Table -1 Selected POS categories

POS-Tags	Definition
JJ	Adjective
RB	Adverb
VB	Verb
NN	Noun

Extending the discussion presented in Section 5.1 we can claim that lemmatization technique using gensim with the help of pattern library helps to identify the keywords having high semantic importance from a given review. This technique eliminates all words apart from adjectives, adverbs, verbs and nouns. For the purpose of clarity, the lemmatized output of the *original text* is displayed below:

lemmatized text = [b'be/VB', b'not/RB', b'impress/VB', b'product/NN']

The semantic polarity of the *original text* is retained in the *lemmatized text* i.e., both have negative polarity. Here, the word "not" isn't eliminated as it is an adverb (RB). Instead of eliminating all the stop words we propose filtering on the basis of POS tags. Thus, the efficacy of this technique is proved for sentiment classification.

5.3 Hyperparameter tuning

The hyperparameters of each classifier need to be adjusted in order to find the best combination which provides better results. These parameters are fine-tuned using GridSearchCV. It is a function used to loop through predefined hyperparameters and fit the estimator (model) on the training set. In this work, we used GridSearchCV which tries all the combinations of the values passed in the form of a dictionary of parameters and evaluates the model for each combination using the 10-fold Cross-Validation method. The accuracy and loss for every combination of hyperparameters is displayed and we can choose the pair of hyperparameters which gives the best result.
The performance of NB depends on the selection of the parameter alpha. It is also called the smoothing parameter. In this work we have used five values of alpha i.e., 0.01, 0.1, 0.5, 1.0 and 10.0.
The optimal values selected for each dataset are:

alpha = 1.0 for Amazon Data
alpha = 0.5 for IMDB Data

The performance of SVM highly depends on the selection of kernel and the C and gamma parameters. A total of 50 combinations of these parameters with 10-fold Cross Validation, totalling to 500 fits was carried out to achieve better results. The optimal parameters are:

kernel = 'rbf', C = 100 and *gamma = 0.01*

These values hold true for both the datasets under consideration. Finally, for ANN fine-tuning of the network is done based on the optimization algorithm, loss function, learning rate and regularization using dropout. The optimal parameters selected are:

optimizer = adam, learning rate = 0.001,
loss = binary cross entropy, and no dropout

The model is further tested based on the learnt parameters and a discussion is presented in the next section.

6. RESULTS AND DISCUSSION

This section provides a comprehensive discussion of the results obtained using the proposed methodology for sentiment classification using NB, SVM and ANN. Initially, accuracy score was calculated to evaluate the performance of the proposed model which is given as the ratio of correctly predicted reviews to the total number of reviews present in the Dataset. The results have been summarized in Table-2.

Table -2: Accuracy score for different classifiers

Accuracy			
Dataset	**Classifier**		
	Naïve Bayes	**SVM**	**ANN**
Amazon	0.865	0.84	0.83
IMDB	0.861	0.897	0.894

Though simple and intuitive, accuracy score has its own limitations. While providing the overall effectiveness of a classifier, we cannot derive deeper insights to answer questions like whether a positively predicted value was actually positive or not. Thus, a cleaner and unambiguous way of presenting the results of a classifier is by using the confusion matrix given in Table -3.

Table -3: Confusion Matrix

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

6.1 Evaluated metrics

With the help of the confusion matrix, we calculated precision, recall, f-1 score and AUC (Area Under the ROC Curve) to evaluate the performance of the classification algorithms in addition to the accuracy score. The formulae for these metrics are given as follows:

$$\boldsymbol{Precision(P)} = \frac{TP}{TP + FP}; \ \boldsymbol{Recall} = \frac{TP}{TP + FN}$$

$$\boldsymbol{F - 1\ Score} = \frac{2 \times P \times R}{P + R}$$

Precision, recall and f-1 score facilitated in identifying any underlying errors and aid in making informed business decisions. Furthermore, we quantified the ROC curve by calculating the AUC score, a metric which takes values between 0 and 1, with a higher number indicating better classification performance.

The confusion matrices and a concise summary of the evaluated metrics for NB, SVM and ANN for Amazon dataset are displayed in Fig 5.1 and Table -4 respectively.

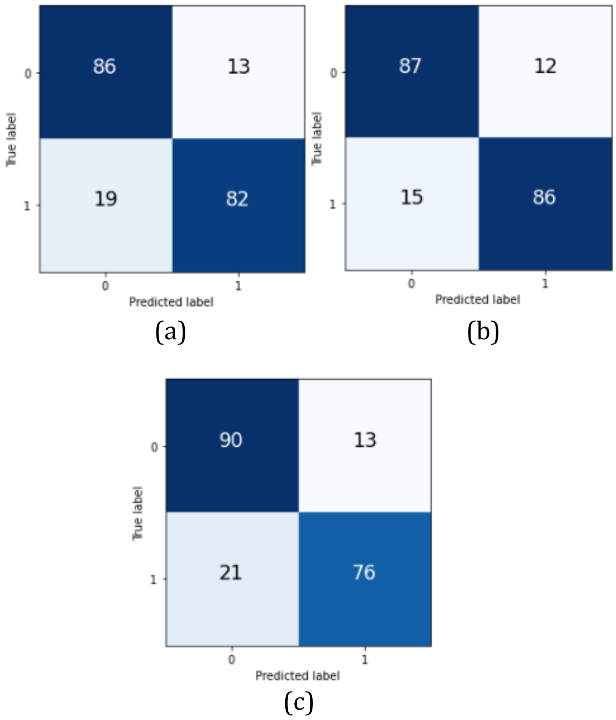

Fig 5.1: Confusion Matrix for (a) Naïve Bayes, (b) SVM and (c) ANN for Amazon Dataset

Table -4: Evaluated metrics for Amazon dataset

Amazon Product Review Dataset			
Evaluation Metrics	**Classifier**		
	Naïve Bayes	**SVM**	**ANN**
AUC	0.95	0.93	0.912
Precision	0.851	0.812	0.854
Recall	0.878	0.863	0.783
F1 Score	0.864	0.837	0.817

Combining the results obtained from Table-2 and Table-4 we conclude that NB outperformed SVM and ANN for the Amazon product reviews dataset.

6.2 Scalability

A scalable model is one that performs well on different datasets. The ability to dynamically adjust to different sizes of training data and deliver consistent results is crucial to prove the overall efficacy of the model. Therefore, the proposed methodology was tested on the large IMDB dataset with 50k movie reviews.

The confusion matrices and a concise summary of the evaluated metrics for NB, SVM and ANN for IMDB dataset are displayed in Fig 5.2 and Table -5 respectively.

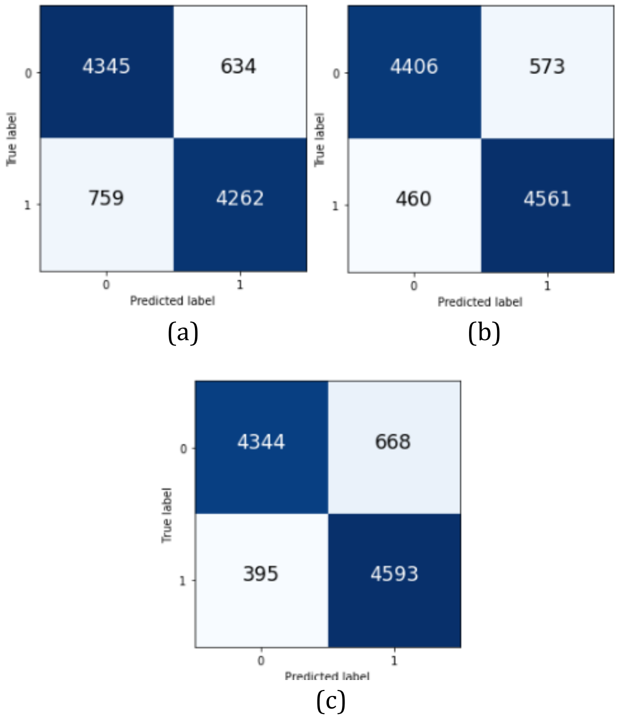

Fig 5.2: Confusion Matrix for (a) Naïve Bayes, (b) SVM and (c) ANN for IMDB 50k Dataset

Table -5: Evaluated metrics for IMDB dataset

IMDB 50k Movie Review Dataset			
Evaluation Metrics	**Classifier**		
	Naïve Bayes	**SVM**	**ANN**
AUC	0.93	0.96	0.96
Precision	0.849	0.908	0.873
Recall	0.871	0.888	0.921
F1 Score	0.860	0.898	0.896

We observed an increase in performance of SVM and ANN. Upon comparison of the results, we can say that the proposed method is capable of scaling well to large datasets. Data in itself is not a panacea but data when combined with analytical techniques usually tends to provide enhanced results as is observed in the case of the large IMDB dataset.

6.3 Optimal model Selection

Model selection refers to the process of selecting the best fit model for a given classification problem. To accomplish this, we compared the relative performance of the three classification techniques, NB, SVM and ANN. This comparison is based on the visualization of the five-classification metrics, accuracy, AUC, precision, recall and f-1 score presented in Chart-1 and Chart-2. These metrics are fundamental for selecting the optimal model.

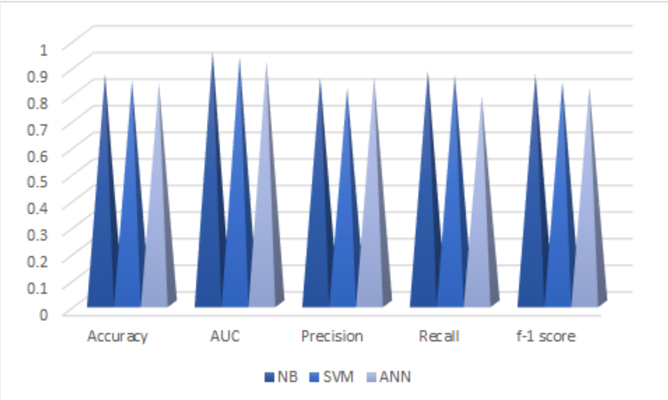

Chart -1: Visual representation of evaluated metrics for Amazon dataset

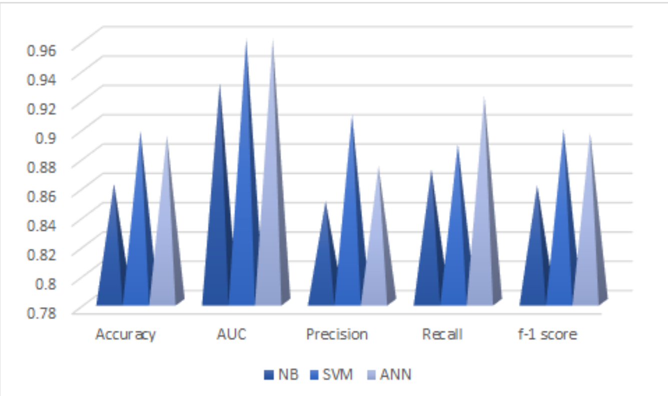

Chart -2: Visual representation of evaluated metrics for IMDB dataset

Training a NB model on any dataset is way faster as compared to SVM and ANN. This is due to the fact that only the probabilities P(C) and conditional probabilities P(xi/C) of each class need to be calculated. In accordance with this, Chart -1 indicates that for the Amazon dataset, NB classifier performs better w.r.t. all the metrics under consideration. But when we extend the discussion to a larger set of data, IMDB dataset, clearly from Chart -2 it can be observed that even with increase in the size of the training set the performance of NB has saturated as it shows no improvement. Due to its sheer simplicity and the "naïve" assumption of conditional independence, NB models are often beaten by models properly trained and fine-tuned like SVM and ANN. Thus, our findings suggest that SVM and ANN perform better with increase in the number of training samples. Finally, in our quest to find the optimal classification technique among SVM and ANN we measure the average performance of both the datasets which is presented in Chart -3.

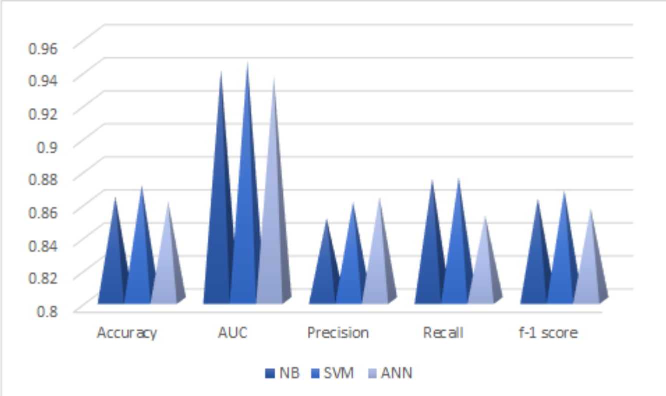

Chart -3: Overall comparison of classification performance

In order to investigate Chart-3, if we compare the overall classifiers performance, SVM outperforms the other methods. It is observed that the precision in case of ANN is better as compared to NB and SVM but if we evaluate the f-1 score which measures the balance between precision and recall, SVM clearly outperforms. SVM also holds an upper hand in terms of accuracy and AUC score. In other words, we can say that the SVM classifier applied on top of the proposed model is able to generalize better on different sets of test data. Thus, after analyzing all the results we select SVM (kernel='rbf', C=100, gamma=0.01) as the optimal classifier for sentiment analysis in this work.

7. CONCLUSION AND FUTURE WORK

Today, every industry is making strides to synergize the bond between information technology and business to tunnel new opportunities in order to develop better associations. This is just the starting point; the field is endless and the goals we may be able to achieve are ineffable. The method proposed in this work will be able to act as the basis for further innovation in Sentiment Analysis. It will help in understanding how customer sentiments can be leveraged to bring about the best experience for them and make savvy business decisions. From the experimental results it is clear that our technique is capable of scaling well to a bigger and more nuanced dataset. The limitation of the proposed method is that the vector size for the larger dataset is massive, resulting in a lot of time and computational complexity. The future scope will be to employ other text vectorization techniques in the place of TF-IDF. Word embeddings can also be considered for better results. This work has been a great learning experience and we sincerely hope that the proposed methodology will aid in further studies.

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] Sneh Paliwal, Sunil Kumar Khatri, Mayank Sharma "Sentiment Analysis and Prediction using Neural Networks", IEEE Xplore, Jan 2019, doi: 10.1109/ICIRCA.2018.8597358

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

[4] Hemalatha S, Ramathmika "Sentiment Analysis of Yelp Reviews by Machine Learning", IEEE Xplore, April 2020, doi: 10.1109/ICCS45141.2019.9065812

[5] Halil Bisgin, Tanmay Bera, Hongjian Ding, Howard G. Semey, Leihong Wu, Zhichao Liu, Amy E. Barnes, Darryl A. Langley, Monica Pava-Ripoll, Himansu J. Vyas, Weida Tong & Joshua Xu "Comparing SVM and ANN based Machine Learning Methods for Species Identification of Food Contaminating Beetles", nature/scientific reports, April 2018, doi: 10.1038/s41598-018-24926-7

[6] Ananya Arora, Prayag Patel, Saud Shaikh, Prof. Amit Hatekar, "Support Vector Machine versus Naïve Bayes Classifier: A Juxtaposition of Two Machine Learning Algorithms for Sentiment Analysis", International Research Journal of Engineering and Technology (IRJET), July 2020, e-ISSN: 2395-0056, p-ISSN: 2395-0072.

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

[8] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis", June 2011, in proceedings of 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 142–150.