iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** IV **Month of publication:** April 2021

DOI: https://doi.org/10.22214/ijraset.2021.33599

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue IV Apr 2021- Available at www.ijraset.com

Sentiment Analysis of Amazon Reviews using Machine Learning Approach

Somsuvra Dutta[1], Santosh Bothe[2]
[1, 2]*Computer Engineering Department, NMIMS University*

Abstract: ***The society we live in is being more digitalized. In this digitalized environment, e-commerce is gaining momentum by getting goods closer to consumers without forcing them to leave their homes. Since more consumers are depending on online products these days, the worth of a review is growing. Sentiment analysis is a classification method that uses machine learning algorithms to interpret the sentiment of text-driven datasets, such as whether a message is positive or negative about a given subject. We want to see how these sentiment analysis methods can be found on Amazon.com product reviews. In this analysis, various machine learning algorithms are compared, trained, and evaluated on an Amazon product review dataset that was randomly chosen from a Kaggle dataset comprising 4 million reviews.***
Keywords: Sentiment Analysis, Amazon Reviews, Feature Extraction, Machine Learning, Opinion Mining

I. INTRODUCTION

Sentiment analysis, also referred to as opinion mining, is a prominent research topic in demand. It is the most difficult task in natural language processing techniques to determine whether data is positive, negative, or neutral. It was first proposed in the early twentieth century and has gradually evolved into an active research area [1].

Finding out what other people think has always been an important part of the information-gathering process. People's thinking and opinion highly influences nowadays, so Sentiment Analysis can analyse people's opinions towards something and based on that insight one can make better decisions be it related to any field. Due to increased usage of the internet, there are various platforms where people's opinions about things are posted like blogs, social media platforms, videos etc [2]. So, it is a task to scrap those data from various platforms and perform analysis for later work.

Consumers can now buy products and add product ratings for promotion and development purposes to online platforms, which has become one of the most relevant and thrilling aspects. When a buyer wishes to purchase a new product, such as a laptop, cell phone, or clothing, he will review several related items and then read feedback of those products and see the ones are the better from the perspective of former customers, and then determine the one to buy.

Customer input on specific goods is critical to the success of a company. They will increase the consistency and performance of their products. However, sifting through all those ratings is a difficult process, since certain brands can have thousands of reviews.

The primary goal of this paper is to apply various algorithms into the review dataset to extract and find out how the effective algorithm for sentiment classification. Various algorithms are implemented along with NLP techniques to pre-process the data.

This paper is divided into following section as follows: Section I talks about the Introduction, Section II is on Related Works, Section III states the various Methods and Approaches used for the analysis, Section IV discusses about the experimental results we got after applying the algorithms, Section V mentions about Research Gaps and Challenges, and in the last Section VI Conclusion and Future Scope is drawn.

II. RELATED WORK

Several reports on Amazon.com reviews opinion mining have been conducted [9]. Traditional Machine Learning (ML) techniques such as Naive Bayesian (NB), Logistic Regression, Support Vector Machine (SVM), and Decision Trees were used in these experiments, with decent results.

In 2018, a student from KTH [10] performed a sentiment study of Amazon beauty product ratings, and he obtained accuracies of more than 90% using the SVM and NB classifiers. For many results, he discovered that SVM outperformed NB. He also concentrated on summary summaries, which are more descriptive and have better accuracy than full articles.

Xing Fang and Justin Zahn used three different classifiers to evaluate different types of Amazon goods (beauty, print, computer, and home) [11]. They came to the realisation that Random Forest presented them with more reliable outcomes much of the time. They also discovered that SVM outperformed NB for wider data sets.

In 2017, the LSTM algorithm was used to test Amazon book reviews [12]. They compared Gated Recurrent Unit (GRU) and Bidirectional LSTM, two types of recurrent neural networks (RNN). For function extraction, the bag-of-words algorithm was used. They had the highest accuracy for the LSTM algorithm using a data collection of more than 210 000 ratings (86 percent).
Xu Yun et al [13] used existing supervised learning algorithms including the perceptron algorithm, naive bayes, and supporting vector machine to estimate a review's rank on Yelp's ranking dataset. They used hold out cross validation, with 70% of the data being used for training and 30% being used for analysis. The author used several classifiers to assess the accuracy and recall value.

III.METHODOLOGY

The methodology which we used for our research purpose is divided into seven steps as shown in fig. 1 which is a data flow diagram for the steps performed by us.

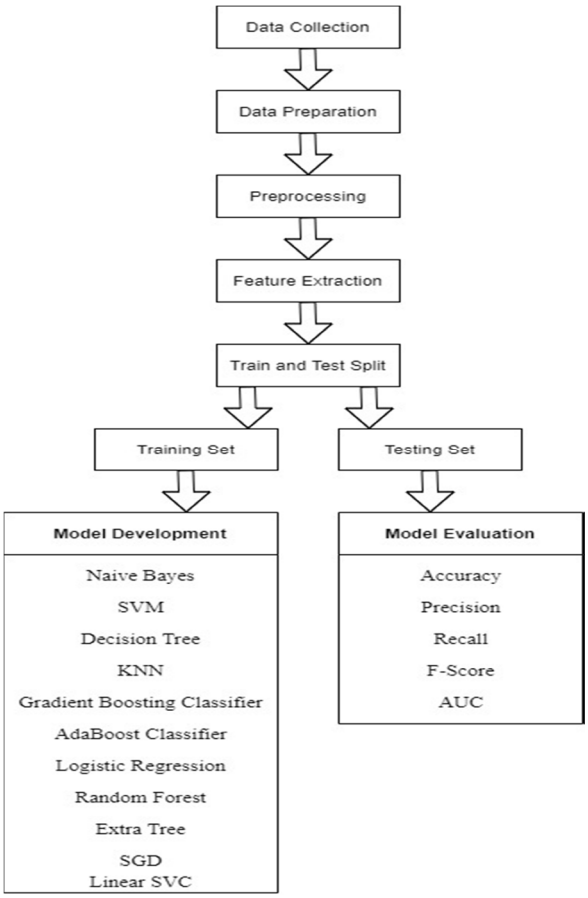


Fig. 1 Research Data Flow

A. Libraries

The algorithms in this study were implemented using Scikit-learn and Natural Language Toolkit, with the help of other scientific computing libraries such as matplotlib, NumPy, and Pandas.

B. Dataset

The training dataset was a large dataset (4 million reviews) accessible on Kaggle [3]. This Kaggle dataset contains Amazon consumer feedback (input text) as well as binary output labels (positive and negative) depending on the review's star score. A score of 1 or 2 stars is considered negative, while 4 or 5 stars is considered positive, and 3 stars (representing the neutral class) is omitted. Both the positive and negative classes have an equal number of members.
The dataset contains approximately 3,600,000 customer reviews from Amazon. Most of the reviews are written in English, although there are a handful that are written in other languages, such as Spanish.

C. Data Pre-processing and Cleaning

The raw review data is cleaned for various elements that could degrade the classifier's accuracy. Any review of more than 100 words was imported and tokenized. After that, all punctuation, labels, stop words, and tokens (emoji, special characters, and foreign languages) that were not English words were removed. Before being appended to a clean list, the remaining tokens were lemmatized using WordNetLemmatizer to reduce words to their root form, and then lemmatized using WordNetLemmatizer to obtain primal expressions.

Lowercasing was performed as the model can judge the same words equally, for example “GREAT” and “great” both are the same but in uppercase the model might be predicting it as some other word. The words were lowercased using word_tokenize(). Since the tokenizers rely on capitalization to determine when to break, deleting them until naming the functions will be inefficient.

Tokenization is done two levels i.e. (1) Sentence and (2) Word. In sentence level tokenization it splits the strings into “sentences” and in word level tokenization it splits the “sentences” into “words”.

The process which followed in the experiment was: (1) Lowercasing; (2) Tokenization; (3) Stemming and Lemmatization; (4) Removing Stop words; (6) Removing Punctuations; (7) Removing Digits; (8) Removing URL’s.

D. Feature Extraction

The vector space model, also known as the word vector model, is an algebraic representation of text documents as vectors of identifiers, such as index terms. It is used in content filtering, retrieval, indexing, and rating relevancy.

Count Vectorizer is used in our model where it transforms a text document into a token count matrix or an integer matrix. To begin with, it tokenizes the review. The number of occurrences of each token is then used to construct a sparse matrix. The scikit-learn Python library is used to build a vector of word counts for each analysis in our implementation. To train the model, structured data is generated with features.

E. Training Phase

A randomly chosen subset of the training dataset, consisting of 11000 ratings, was extracted to keep the computational expenses down. The classifiers were trained using 10,000 of the 11000 reviews, while the remaining 1000 were used to test their efficiency.

F. Machine Learning (ML) Algorithms

The machine learning (ML) algorithms used in the study are listed in this section. The algorithms mentioned below are used here because they are widely used by researchers.

The Naive Bayes (NB) classifier is a Bayes theorem-based probabilistic classifier. Rather than making predictions, the Nave Bayes classifier produces probability estimates. For each class value, they calculate the probability that a given instance belongs to that class. The Naive Bayes classifier has the advantage of only requiring a limited amount of training data to approximate the classification parameters. The effect of an attribute value on a given class is believed to be independent of the values of the other attributes. Class conditional independence is the term for this.

The Support Vector Machine (SVM) algorithm's aim is to find the best line or decision boundary for categorising n-dimensional space into classes so that new data points can be conveniently placed in the correct category in the future. A hyperplane is the term for the best decision boundary.

The K-Nearest Neighbors (KNN) classification divides instances into groups based on how close they are. It is one of the most widely used pattern recognition algorithms. It is a form of lazy learning in which the function is only approximated locally, and all computation is postponed until after classification. Most of its neighbours classify an object. The number K is always a positive number. The neighbours are chosen from a group of entities for which the proper classification has been determined. The IBK classifier in WEKA is the name of this classifier.

Random forest is a supervised machine learning algorithm that can be used for classification and regression. The "forest" refers to a group of uncorrelated decision trees that are then combined to minimise uncertainty and make more accurate data predictions.

Linear SVM is a method for generating a classifier (a vector) that can distinguish between labelled datasets. Given two types of points, circles in a space, it tries to optimise the minimum distance between one of the points and the other. To put it another way, it maximises the margin.

Stochastic Gradient Descent Classifiers uses stochastic gradient descent (SGD) learning to apply regularised linear models. The loss of gradient is measured one sample at a time, and the model is revised with a diminishing strength schedule along the way (i.e., learning rate). The partial fit approach in SGD allows for minibatch (online/out-of-core) learning.

Logistic Regression is a statistical analysis algorithm that is based on the probability principle. A Logistic Regression model is like a Linear Regression model, except that the Logistic Regression uses a more dynamic cost function, which is known as the "Sigmoid function" or "logistic function" instead of a linear function. The logistic regression theorem implies that the cost function be limited to a value between 0 and 1. As a result, linear functions struggle to represent it so it may have a value greater than 1 or less than 0, which is impossible according to the logistic regression theorem.

Decision Trees (DTs) are a non-parametric supervised learning process. The aim is to learn basic decision rules from data features to construct a model that forecasts the value of a target variable. A tree approximates a piecewise constant.

Extra Trees Classifier uses a meta estimator to fit a variety of randomised decision trees (a.k.a. extra-trees) on different sub-samples of the dataset and use averaging to increase statistical precision and control over-fitting.

Gradient Boosting Classifier creates an additive model in a stage-by-stage way, allowing for the optimization of every differentiable loss function. The negative gradient of the binomial or multinomial deviance loss function is used to match n classes regression trees in each point. A special case of binary classification is where only one regression tree is caused.

AdaBoost classifier is a meta-estimator that starts by fitting a classifier on the initial dataset, then fits additional copies of the classifier on the same dataset but adjusts the weights of wrongly categorised instances such that subsequent classifiers concentrate more on difficult situations.

G. Performance Evaluation Criteria

A confusion matrix is a graphic representation of the accuracy of classifiers in classification. It is used to demonstrate how events and expected groups are linked. The number of correct and incorrect classifications in each potential value of the attribute being classified in the confusion matrix is used to measure the classification model's effectiveness.

Fig 2 describes the confusion matrix, where TP means number of instances which are positive and predicted positive, FP means where instances which are negative are predicted as positive, FN means the instances which are positive are predicted as negative.

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

Fig. 2 Confusion Matrix

1) *Accuracy:* Accuracy is a good predictor of the model's degree of correctness in training and how it will behave in general. It can be defined as the proportion of correct predictions to incorrect predictions. As a result, the equation given can be used to calculate the accuracy value.

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

2) *Recall:* In general, recall, also known as sensitivity, can be defined as the ratio of correctly determined positive instances to all observations. Recall can be thought of as a metric for how well a system predicts positives and calculates costs.

$$Recall = \frac{TP}{TP + FN}$$

3) *Precision:* Precision may be understood as the degree of accuracy in predicting positive outcomes. It's actually the proportion of true positives to the total number of positives. This shows the system's ability to handle positive values, so it doesn't show how it handles negative values.

$$Precision = \frac{TP}{TP + FP}$$

4) *F- Score:* Precision and Recall are combined to form a weighted average. As a result, this metric takes into account all types of false values. When the F1 score is 1, it is considered ideal, and when it is 0, it is considered a complete failure.

$$F\ Score = \frac{2 * (Precision * Recall)}{Precision + Recall}$$

5) *AUC:* The False Positive Rate (FPR) and True Positive Rate (TPR) are merged into a single parameter called the Area Under Curve (AUC). For the classification algorithm, the FPR and TPR are first computed with a variety of thresholds. The Receiver Operating Characteristic (ROC) curve is generated by parametrically plotting these FPRs and TPRs in a single line. Finally, we consider the Area of this Curve, also known as AUROC or AUC.

IV. EXPERIMENTAL RESULTS

A comparison study was conducted using multiple algorithms that were implemented in a computer with an Intel Core i5 processor and 8GB of RAM. We used open-source machine learning libraries in Python called NumPy, Pandas, and Scikit-learn. The programme was ran using Kaggle, an open-source web application.

The frequency of words used in the reviews are shown in Fig. 3 where the distribution shows that most of the reviews are written in about 50-100 words.

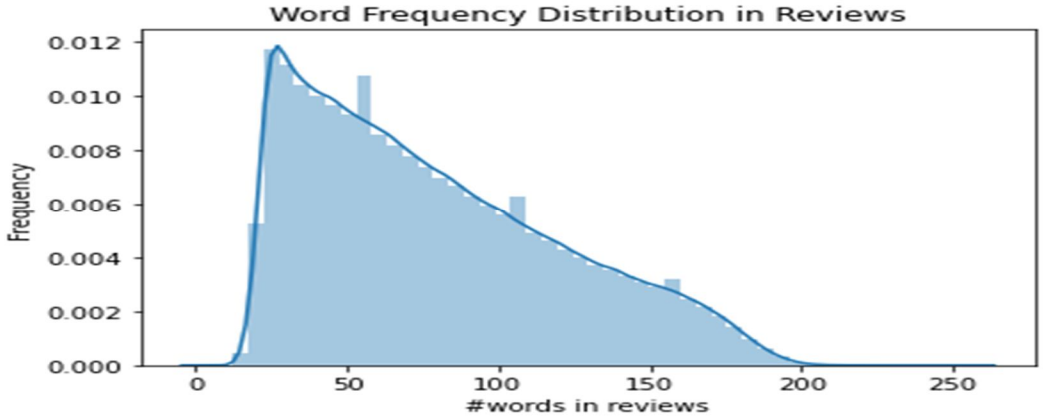


Fig. 3 Word Frequency Distribution

The above proves was followed by finding out the most frequent words used as shown in Fig. 4 from which feature important positive and negative features were extracted as shown in Fig. 5 and Fig. 6.

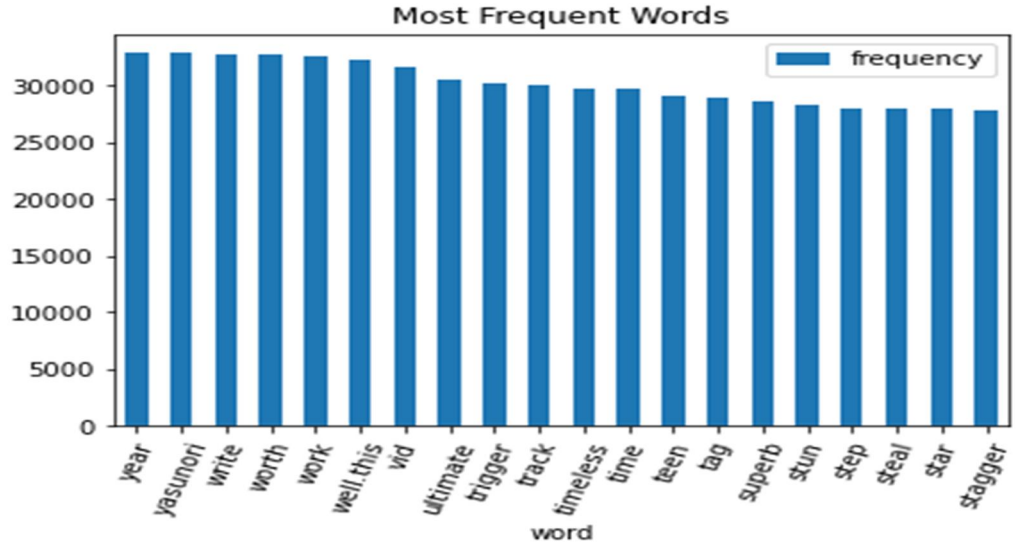


Fig. 4 Frequently used Words.

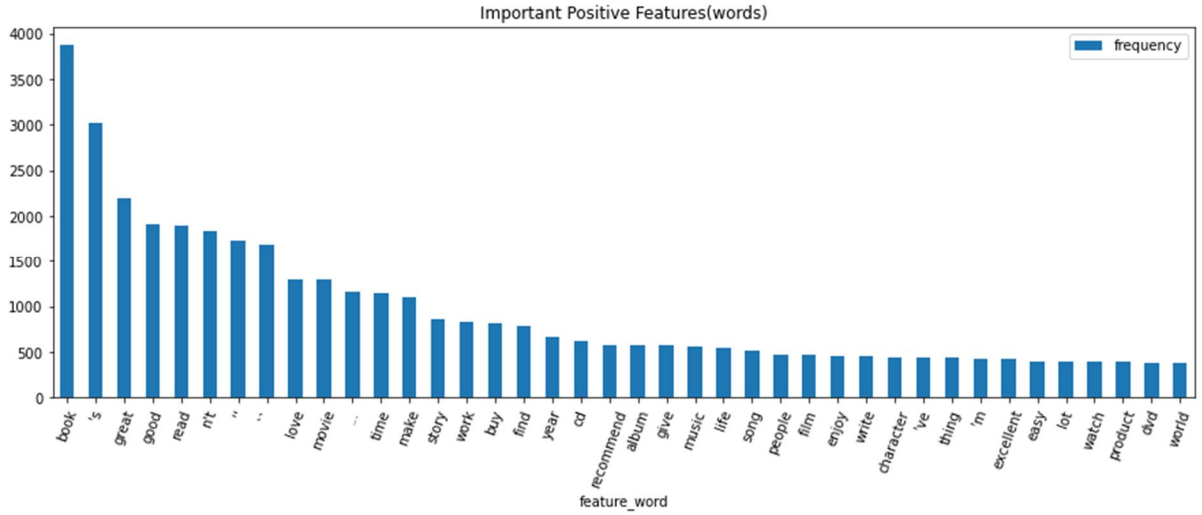


Fig. 5 Frequently used Positive Words.

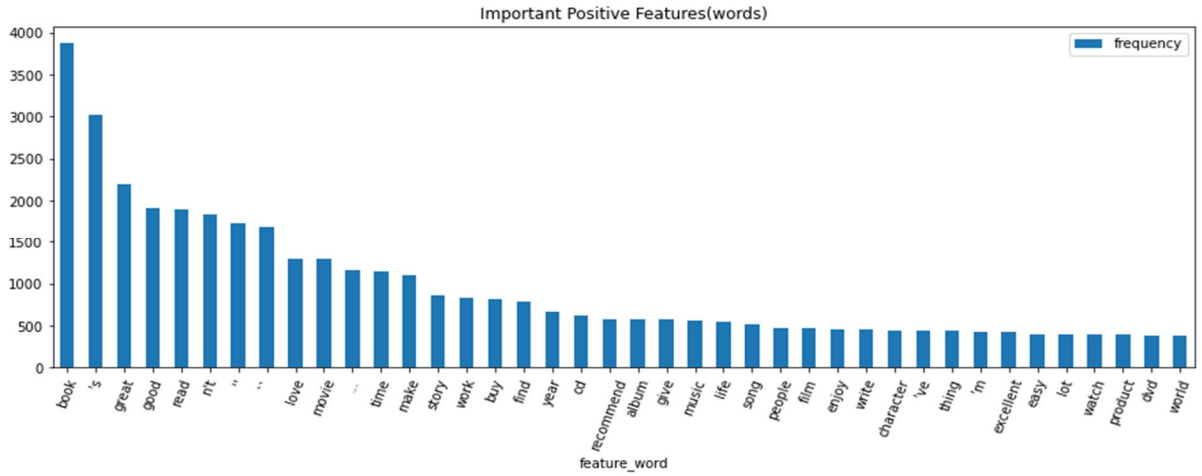


Fig. 6 Frequently used Negative Words.

The confusion matrices of the algorithms stated in the previous section are show in Fig. 8 from which various calculations were done like finding the accuracy, recall, precision etc.

The accuracies of the classifiers are compared in Fig 7. The best classifier is Extra Forest (86.2%), and the accuracy achieved by logistic regression is greater than that obtained by other classifiers.

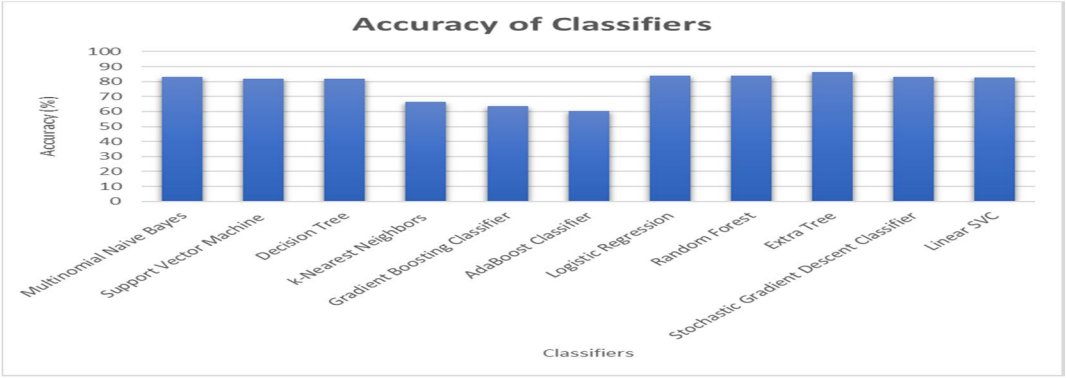


Fig. 7 Accuracy Obtained by Algorithms

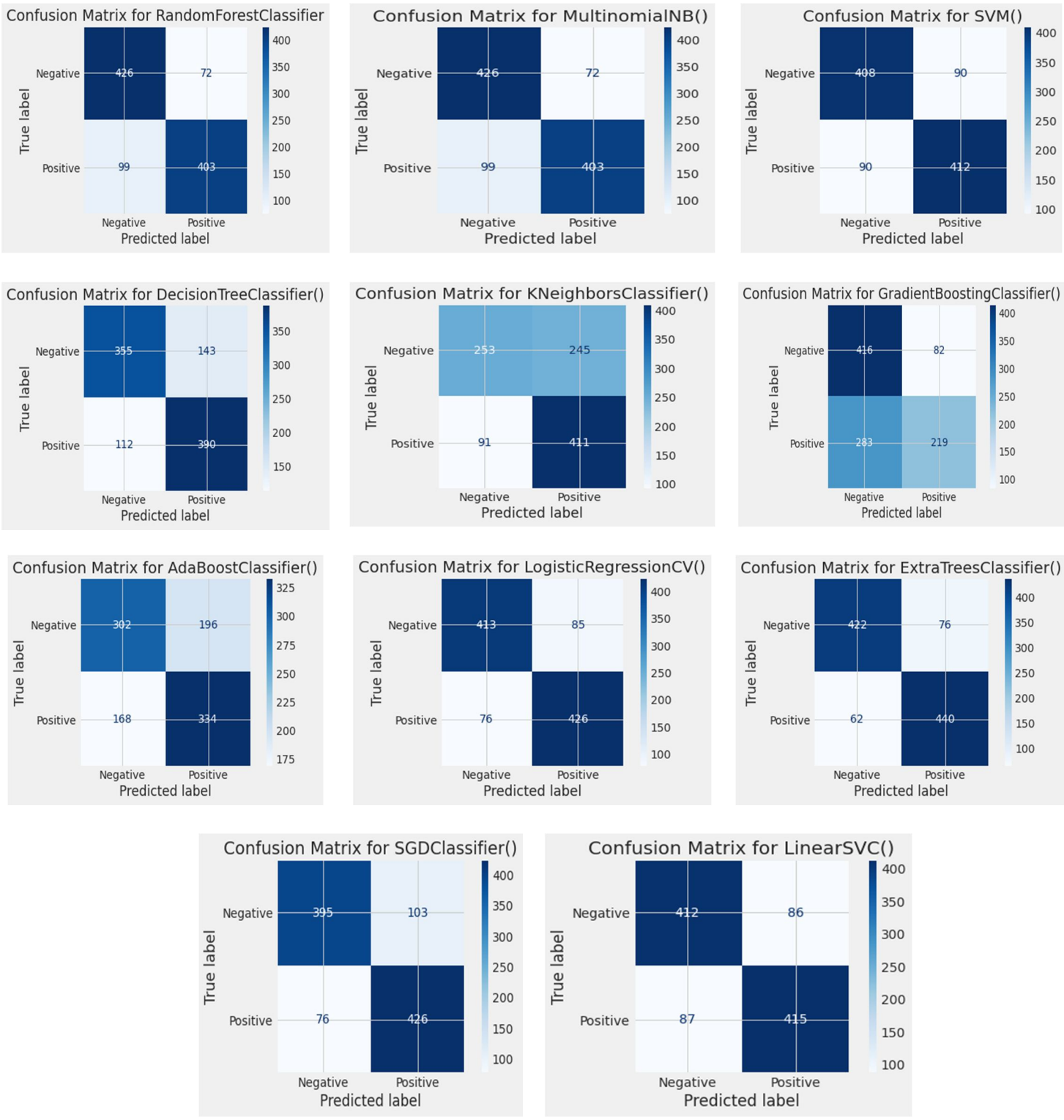


Fig. 8 Confusion Metrices of Classifiers

The precisions of the classifiers are compared in Figure 9. The best classifier is Extra Forest Classifier (85.27%), and the precision achieved by multinomial naïve bayes classifier is greater than that obtained by other classifiers.

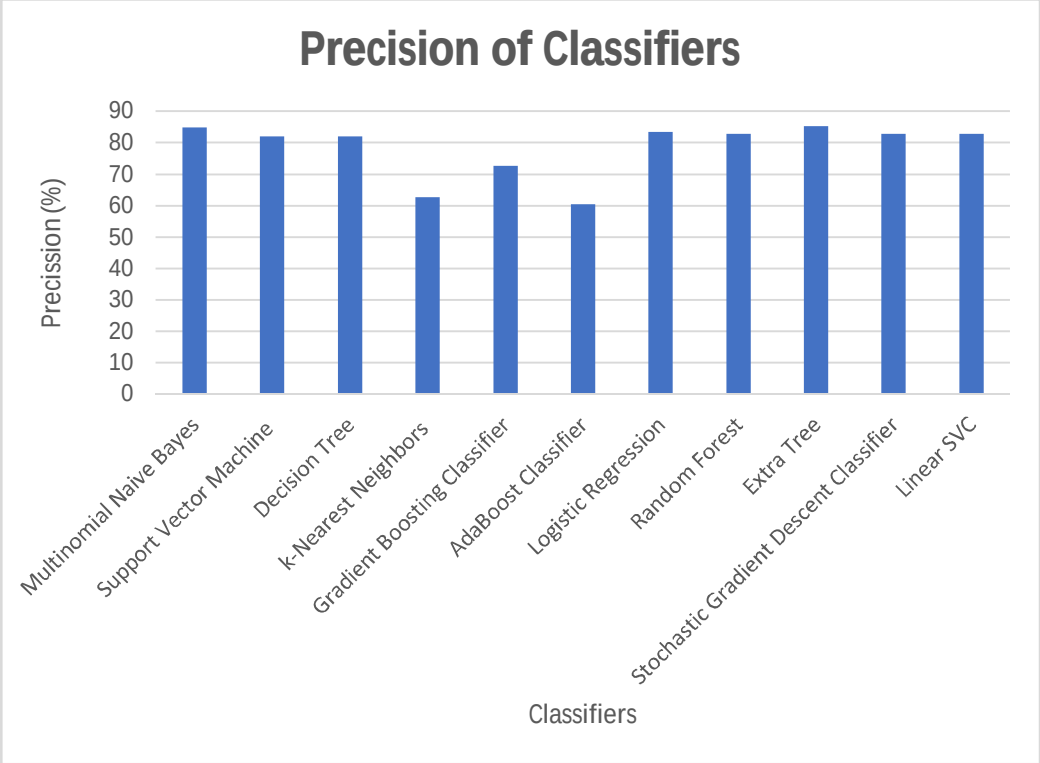


Fig. 9 Precision Obtained by Algorithms

The recall performance measures of the classifiers are compared in Fig. 10. The best classifier is Extra Tree Classifier (87.65%), and the recall achieved by random forest classifier is greater than that obtained by other classifiers.

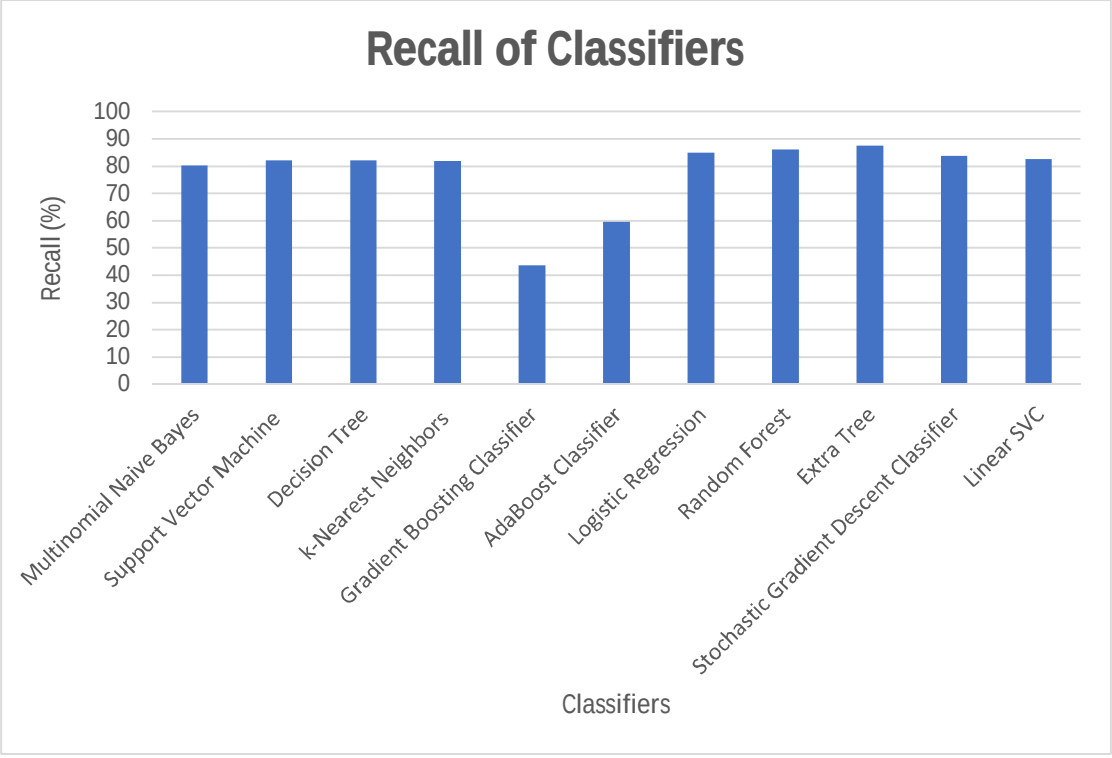


Fig. 10 Recall Obtained by Algorithms

The F-Scores of the classifiers are compared in Fig. 11. The best classifier is Extra Tree Classifier (86.44%), and the F-Score achieved by random forest classifier is greater than that obtained by other classifiers.

The AUC of the classifiers are compared in Fig. 12. The best classifier is Extra Tree Classifier (86.19%), and the F-Score achieved by random forest classifier is greater than that obtained by other classifiers.

Fig. 13 shows a graphical representation of the performance metrics for the different algorithms. Table I reveals that Extra Tree Classifier has the best performance measures, followed by Random Forest Classifier having the second highest accuracy, precision, recall and F1 Score when compared to other algorithms. It can be said that the model having the worst performance measures is given by Gradient Boosting Classifier.

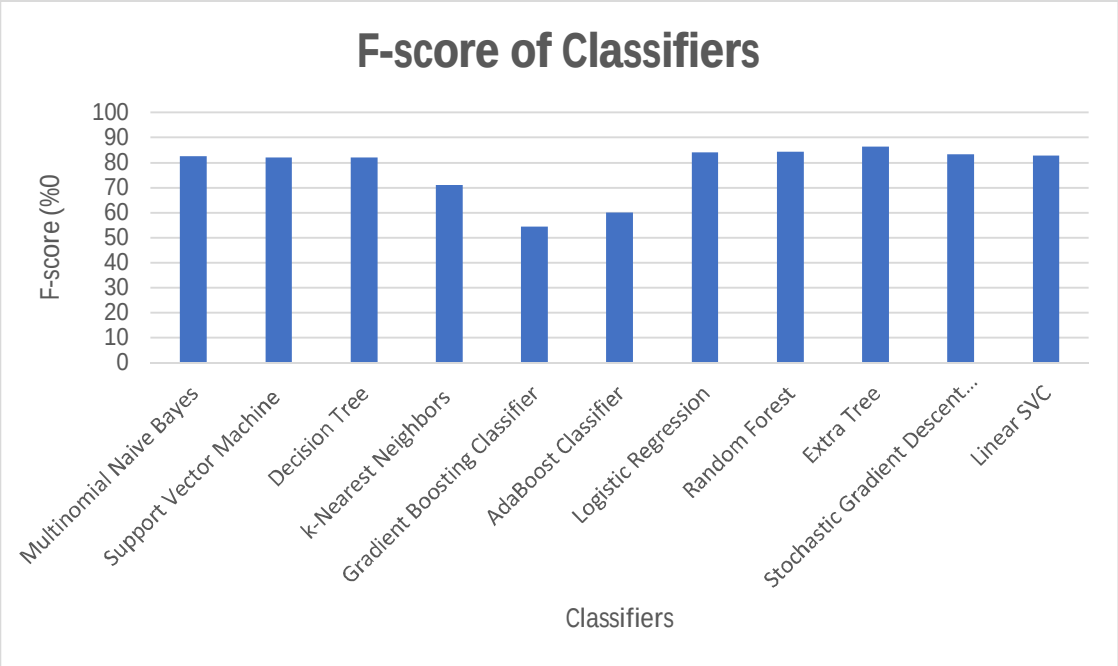


Fig. 11 F-Score Obtained by Algorithms

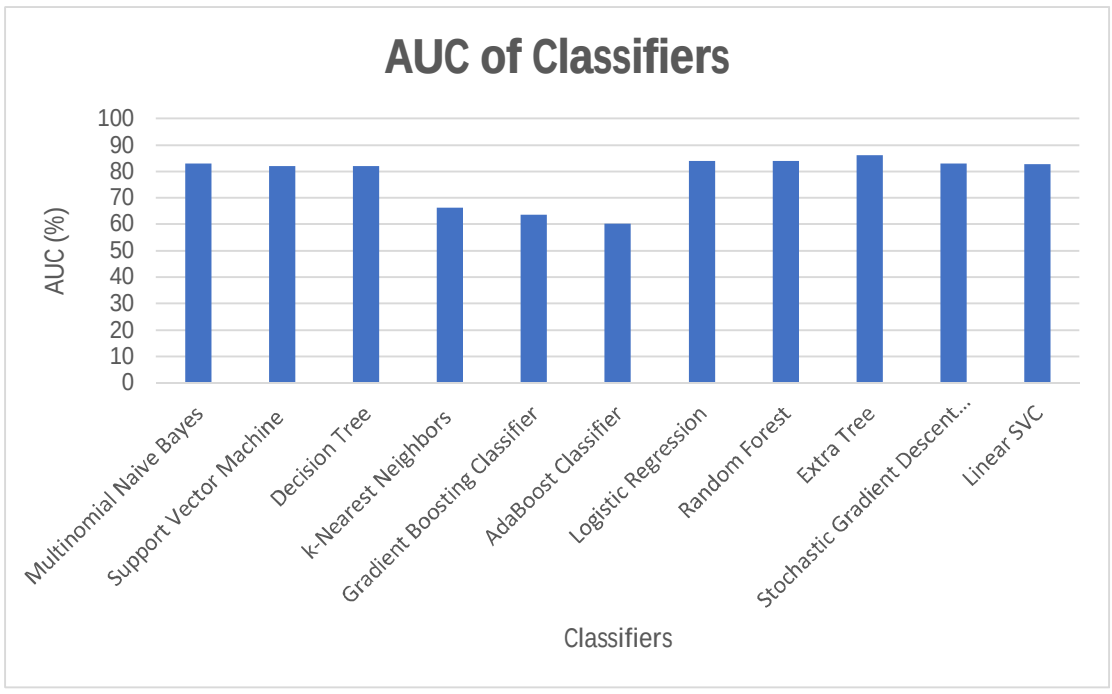


Fig. 12 AUC Obtained by Algorithms

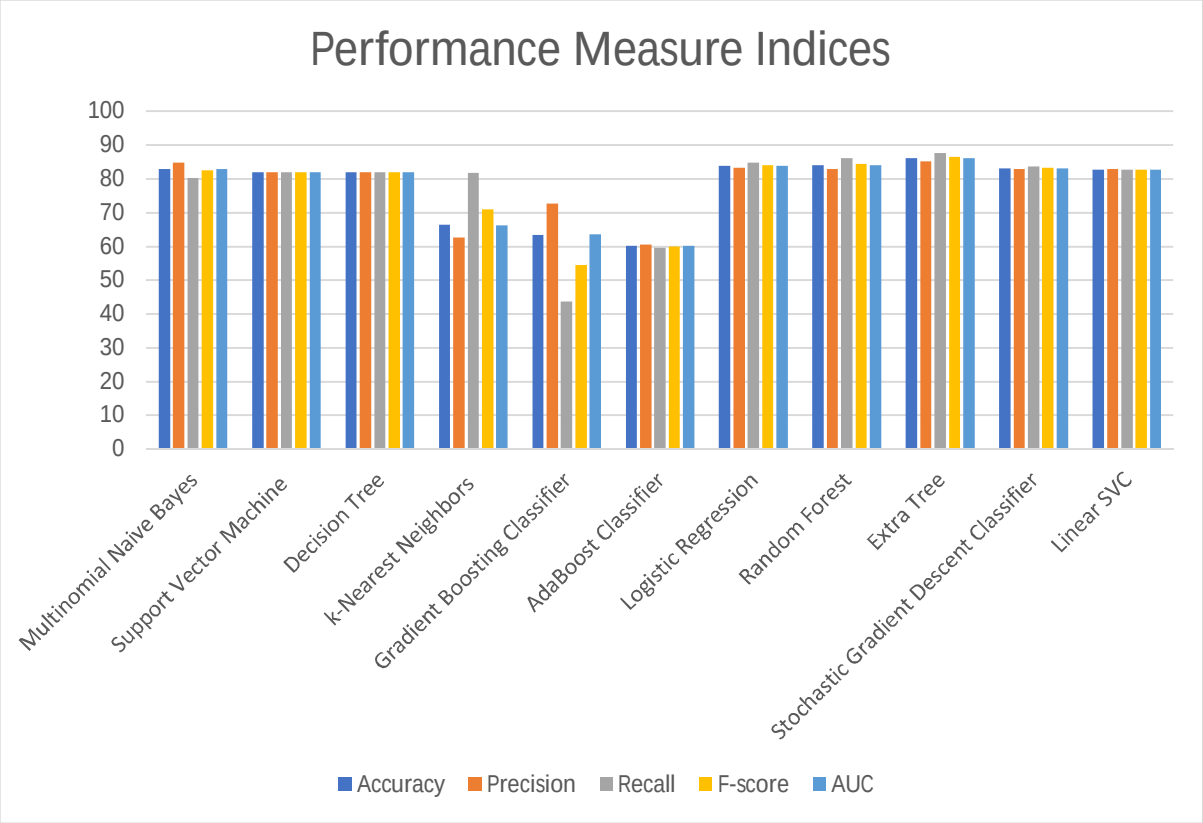


Fig. 13 Performance Measure Indices

TABLE I
Comparative Study of Performance Measures

Model Performance (Testing Phase)						
Sr No	Technique Used	Accuracy	Precision	Recall	F-score	AUC
1	Multinomial Naive Bayes	82.9	84.8421	80.2789	82.4974	82.9105
2	Support Vector Machine	82	82.0717	82.0717	82.0717	81.9997
3	Decision Tree	82	82.0717	82.0717	82.0717	81.9997
4	k-Nearest Neighbors	66.4	62.6524	81.8725	70.9845	66.3379
5	Gradient Boosting Classifier	63.5	72.7575	43.6255	54.5455	63.5798
6	AdaBoost Classifier	60.2	60.5263	59.5618	60.0402	60.2026
7	Logistic Regression	83.9	83.3659	84.8606	84.1066	83.8961
8	Random Forest	84.1	82.9175	86.0558	84.4575	84.0921
9	Extra Tree	86.2	85.2713	87.6494	86.444	86.1942
10	Stochastic Gradient Descent Classifier	83.1	82.8402	83.6653	83.2507	83.0977
11	Linear SVC	82.7	82.8343	82.6693	82.7517	82.7001

V. LIMITATIONS

In this paper the model was trained on Amazon Product Review data and cannot promise that it will predict the correct label (positive or negative) for other e-commerce sites. Since we do not know anything about product categories, we cannot estimate the performance of prediction of product category segments. Predictions are often restricted to the effectiveness of: Algorithms for Data Cleaning, Text Embedding and Prediction algorithms.

VI. CONCLUSION AND FUTURE WORK

This paper explored sentiment analysis model that could be used to determine if something is positive or negative. With the aim of finding the most effective classifier, it used eleven popular machine learning classifiers: Multinomial Naive Bayes, Support Vector Machine, Decision Tree, K-Nearest Neighbors, and Gradient Boosting Classifier AdaBoost Classifier, Logistic Regression, Random Forest, Extra Tree, Stochastic Gradient Descent Classifier and Linear SVC. We also included detailed descriptions of every aspect of the model, as well as evaluation metrics. The dataset used is made up of about 3,600,000 Amazon customer reviews obtained from the Kaggle website. To assess each algorithm's results, we used recall, precision, F-score, AUC, and accuracy. Extra Tree and Random Forest outperformed the other classifiers, according to the analysis. The accuracy of the Gradient Boosting Classifier was the lowest of all the experiments.

This work will undoubtedly be improved in a variety of ways. Rather than deleting emotions during the pre-processing phase, we may transform them to text that contains a meaning for the customer's opinion, which can have a good impact on classification accuracy. We also recommend using additional data to experiment for this model to improve performance.

REFERENCES

[1] M. A. M. Salem and A. Y. A. Maghari, "Sentiment Analysis of Mobile Phone Products Reviews Using Classification Algorithms," 2020 International Conference on Promising Electronic Technologies (ICPET), Jerusalem, Palestine, 2020, pp. 84-88

[2] Shaukat, Zeeshan & Zulfiqar, Abdul Ahad & Xiao, Chuangbai & Azeem, Muhammad & Mahmood, Tariq. (2020). Sentiment analysis on IMDB using lexicon and neural networks. SN Applied Sciences, Springer.

[3] "Amazon Reviews for Sentiment Analysis." [Online]. Available: https://www.kaggle.com/bittlingmayer/amazonreviews

[4] Mohan Kamal Hassan et al 2017," Sentimental analysis of Amazon reviews using naïve bayes on laptop products with MongoDB and R" IOP Conf. Ser.: Mater. Sci. Eng. 263 042090

[5] Coyne, Emilie & Smit, Jim & Güner, Levent. (2019). Sentiment analysis for Amazon.com reviews. 10.13140/RG.2.2.13939.37920.

[6] T. P. Sahu and S. Ahuja, "Sentiment analysis of movie reviews: A study on feature selection & classification algorithms," 2016 International Conference on Microelectronics, Computing and Communications (MicroCom), Durgapur, India, 2016, pp. 1-6, doi: 10.1109/MicroCom.2016.7522583.

[7] R. I. Permatasari, M. A. Fauzi, P. P. Adikara and E. D. L. Sari, "Twitter Sentiment Analysis of Movie Reviews using Ensemble Features Based Naïve Bayes," 2018 International Conference on Sustainable Information Engineering and Technology (SIET), Malang, Indonesia, 2018, pp. 92-95, doi: 10.1109/SIET.2018.8693195

[8] T. M. Untawale and G. Choudhari, "Implementation of Sentiment Classification of Movie Reviews by Supervised Machine Learning Approaches," 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2019, pp. 1197-1200, doi: 10.1109/ICCMC.2019.8819800.

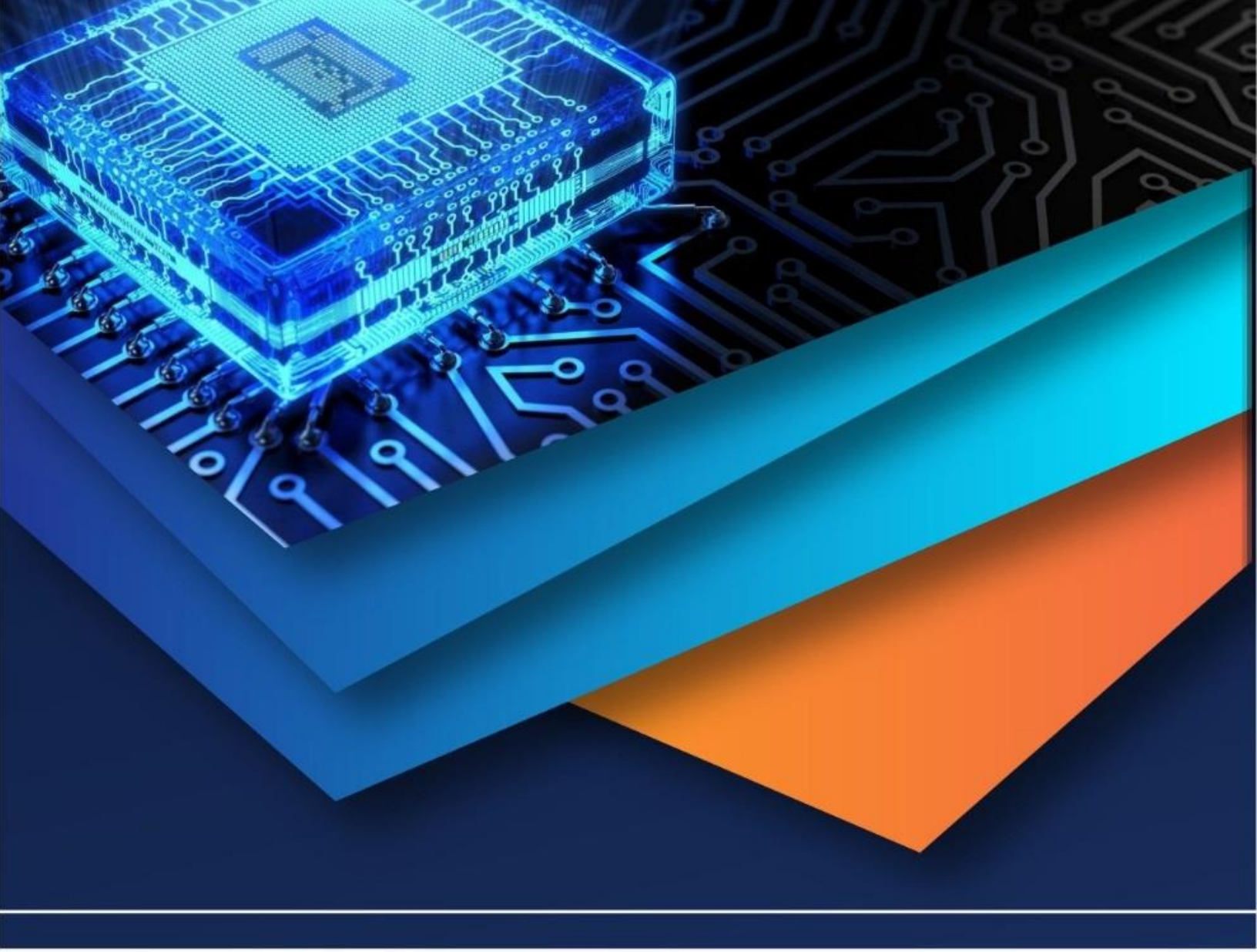
[9] T. U. Haque, N. N. Saber, and F. M. Shah, "Sentiment analysis on large scale Amazon product reviews," 2018 IEEE International Conference on Innovative Research and Development, ICIRD 2018, no. May, pp. 1–6, 2018

[10] S. Paknejad, "Sentiment classification on Amazon reviews using machine learning approaches," 2018.

[11] X. Fang and J. Zhan, "Sentiment analysis using product review data," Journal of Big Data, vol. 2, no. 1, 2015. [Online]. Available: http://dx.doi.org/10.1186/s40537- 015-0015-2

[12] J. Nowak, A. Taspinar, and R. Scherer, "LSTM recurrent neural networks for short text and sentiment classification," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10246 LNAI, pp. 553–562, 2017.

[13] Y. Xu, X. Wu, and Q. Wang. Sentiment analysis of yelps ratings based on text reviews, 2015.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY