Journal on Big Data

Tech Science Press

DOI: 10.32604/jbd.2023.041319

ARTICLE

Sentiment Analysis Based on Performance of Linear Support Vector Machine and Multinomial Naïve Bayes Using Movie Reviews with Baseline Techniques

Mian Muhammad Danyal[1], **Sarwar Shah Khan**[2,4], **Muzammil Khan**[2,*], **Muhammad Bilal Ghaffar**[1], **Bilal Khan**[1] and **Muhammad Arshad**[3]

[1]Department of Computer Science, City University of Science & Technology, Peshawar, 25000, Pakistan

[2]Department of Computer Science and Software Technology, University of Swat, Swat, 19200, Pakistan

[3]Department of Computer Software Engineering, University of Engineering & Technology Mardan, Mardan, 23200, Pakistan

[4]Department of Computer Science, IQRA National University, Swat, 19200, Pakistan

*Corresponding Author: Muzammil Khan. Email: muzammilkhan86@gmail.com

Received: 18 April 2023 Accepted: 05 June 2023 Published: 29 September 2023

ABSTRACT

Movies are the better source of entertainment. Every year, a great percentage of movies are released. People comment on movies in the form of reviews after watching them. Since it is difficult to read all of the reviews for a movie, summarizing all of the reviews will help make this decision without wasting time in reading all of the reviews. Opinion mining also known as sentiment analysis is the process of extracting subjective information from textual data. Opinion mining involves identifying and extracting the opinions of individuals, which can be positive, neutral, or negative. The task of opinion mining also called sentiment analysis is performed to understand people's emotions and attitudes in movie reviews. Movie reviews are an important source of opinion data because they provide insight into the general public's opinions about a particular movie. The summary of all reviews can give a general idea about the movie. This study compares baseline techniques, Logistic Regression, Random Forest Classifier, Decision Tree, K-Nearest Neighbor, Gradient Boosting Classifier, and Passive Aggressive Classifier with Linear Support Vector Machines and Multinomial Naïve Bayes on the IMDB Dataset of 50K reviews and Sentiment Polarity Dataset Version 2.0. Before applying these classifiers, in pre-processing both datasets are cleaned, duplicate data is dropped and chat words are treated for better results. On the IMDB Dataset of 50K reviews, Linear Support Vector Machines achieve the highest accuracy of 89.48%, and after hyperparameter tuning, the Passive Aggressive Classifier achieves the highest accuracy of 90.27%, while Multinomial Nave Bayes achieves the highest accuracy of 70.69% and 71.04% after hyperparameter tuning on the Sentiment Polarity Dataset Version 2.0. This study highlights the importance of sentiment analysis as a tool for understanding the emotions and attitudes in movie reviews and predicts the performance of a movie based on the average sentiment of all the reviews.

KEYWORDS

Opinion mining; machine learning; movie reviews; IMDB Dataset of 50K reviews; Sentiment Polarity Dataset Version 2.0

1 Introduction

Every year, large numbers of movies are released. This number has increased in recent years as the movie industries produces more and more films each year. Whether it is a new release or an old classic, there is always something to enjoy. To feel good and escape from reality, people watch movies. Movies make them feel happy, sad, scared, and have lots of other emotions, and that is what makes movies so special and fun to watch over and over again.

Movie reviews are comments expressed by people who have seen the movie. All of these reviews determine if the movie is worth watching or not [1]. In the modern world, watching movies is the most entertaining way to pass the time. People enjoy giving their thoughts on movies [2]. A written movie review provides information about the positive and negative aspects of the movie, and a deeper analysis of a movie review can show if the movie meets the reviewer's expectations. A lot of reviews written by people and posted on the Internet can be helpful to other people. This kind of information is very important for making decisions, which is why a lot of people use the Internet. This information is very helpful for people to save time, internet data, and money. The average human reader finds it challenging to find relevant websites and to extract and summarize the reviews quickly enough to make the best decision because every website has a significant amount of review text. This is why automated opinion mining systems are needed.

Opinion mining is the process of identifying and extracting subjective information from source materials using text analysis, natural-language-processing, and computational linguistics techniques [3]. Opinion Extraction is a way to automatically analyze people's feelings from their opinions. This information can help manufacturers figure out how to analyze their products. In the last fifteen years, research groups, universities, and service industries have worked hard on sentiment analysis which is another term for opinion mining, to find out and analyze the public's feelings and comments [4]. With the fast growth of web technology and easy access to the internet, sentiment analysis is becoming more and more popular and widely used. A large amount of user-generated content available online provides an abundance of data for sentiment analysis algorithms to analyze. In the context of movies, it is possible to use sentiment analysis for analyze movie reviews, instead of asking just a few people, sentiment analysis looks at lots of reviews or posts from lots of people all over the provided source. This helps us see what lots of people think about a product or brand, and we can use this information to make better choices or improve the product.

Sentiment Analysis involves analyzing the opinion, emotions, attitudes, and feelings expressed by individuals towards entities such as events, services, products, organizations, and their features [2]. It is a method to find out if a review of a movie is positive or negative. Analyzing the reviews can give an overall rating for a movie. This can help people decide if a movie is worth watching without having to read all the reviews. By training machine learning algorithms to determine whether a review is positive or negative based on its wording, sentiment analysis can be automated.

2 Literature Review

This section provides an overview of the ongoing research work being done in the field of Opinion Mining or Sentiment Analysis.

Ullah et al. [5] suggested using a seven layers-deep neural network to figure out how people feel about movie reviews. The model includes an embedding layer that converts the dataset into a set of numerical vectors, followed by two successive 1D-CNN layers for feature extraction. A global max-pooling layer is used to reduce the number of dimensions. A dropout layer is included in addition to a dense layer, which is used for classification. The dense layer prevents the network from becoming too good at what it does, and the dropout layer improves the way the network functions in general. The last layer that can tell the difference between two classes is one that is fully connected. Daeli et al. [6] proposed determining and comparing the optimal K for K-Nearest Neighbor to other methods. Informational Gain with KNN performed better than Naive Bayes, Random Forest, and Support Vector Machines, with 96.8 percentage accuracy and an optimal K of 3. Devi et al. [1] compared Decision tree classification and Naïve Bayes classification on the IMDB reviews dataset which has 50k reviews (half positive and half negative). Both classifiers are evaluated by confusion matrix, recall, precision, f-measure, and accuracy in which Naïve Bayes gives the best 0.97 percentage accuracy.

Bodapati et al. [7] predicted sentiment analysis for movie reviews using LSTMs, a variant of RNNs. The task is presented as a binary task of classification, with the review of it as positive or negative. Sentence vectorization techniques are applied to manage the wide range of sentence lengths. They attempt to investigate the effect of hyperparameters such as dropout, layer count, and activation functions. The model's performance with various neural network configurations, as well as the reported performance with respect to every configuration, were examined. The experimental studies make use of the IMDB benchmark dataset.

Rahman et al. [8] collected movie review data and analyzed it using five different types of machine learning classifiers. As a result, the classifiers under consideration are Support Vector Machine, Bernoulli and Multinomial Naive Bayes, Maximum Entropy, and Decision Tree.

Chakraborty et al. [9] proposed using Word2vec and the Doc2vec model to execute a deep learning technique. Aside from these, various classifiers other than the one mentioned above can be used to determine if the best classifier for these purposes can be found. It should be stated that this model can be used tactically with other clustering algorithms such as DBScan, Fuzzy-C-Means, and so on. Mohan Kumar et al. [10] proposed using the Robust Clustering algorithm and the classification and regression technique to do a sentiment analysis on movie reviews from the BookMyShow database that can be found online. This would improve the accuracy of opinion mining. The tone of the review shows how one-sided or two-sided it is. Malini et al. [11] proposed a model to analyze tweets about Bollywood movie reviews. Using classifiers like the Support Vector Machines and Naive Bayes, it divides tweets as positive, neutral, or negative.

Baid et al. [12] introduced a technique to assess an individual's sentiment towards a particular content source via opinion mining. They analyzed various types of online data, including tweets, blogs, and status updates, to gather a vast amount of information. They used different machine learning techniques such as Naive Bayes, K-Nearest Neighbor, and Random Forest to analyze movie reviews. Sahu et al. [13] suggested focusing sentiment analysis on the IMDB database of movie reviews. They utilized sentiment analysis to assign a polarity score to each movie review, ranging from 0 (strongly negative) to 4 (strongly positive). They then conducted feature extraction and ranking to train a multilabel classifier that accurately categorizes each movie review. The accuracy of classification methods increases to 88.95%.

Manek et al. [14] suggested using a Support Vector Machine (SVM) classifier and a Gini Index-based approach to select features for sentiment classification in large movie review datasets. The findings revealed that our Gini Index technique significantly improves classification accuracy and reduces the error rate. Avinash et al. [15] evaluated TF-IDF and Doc2vec sentiment analysis methods on datasets such as Cornell movie reviews and UCI sentiment-labeled data. They improve accuracy by preprocessing, such as stop-word removal and tokenization, and they evaluate features with classifiers.

3 Research Methodology

There are many numbers of algorithms in machine learning. For this research study, Logistic Regression (LR), Multinomial Naïve Bayes (MNB), Linear Support Vector Machine (LSVM), K-Nearest neighbors (KNN), Passive Regressive Classifier (PAC), Decision Tree (DT), Random Forest Classifier (RFC), and Gradient Boosting Classifier (GBC) are chosen. The suggested method's systematic flow is represented in Fig. 1.

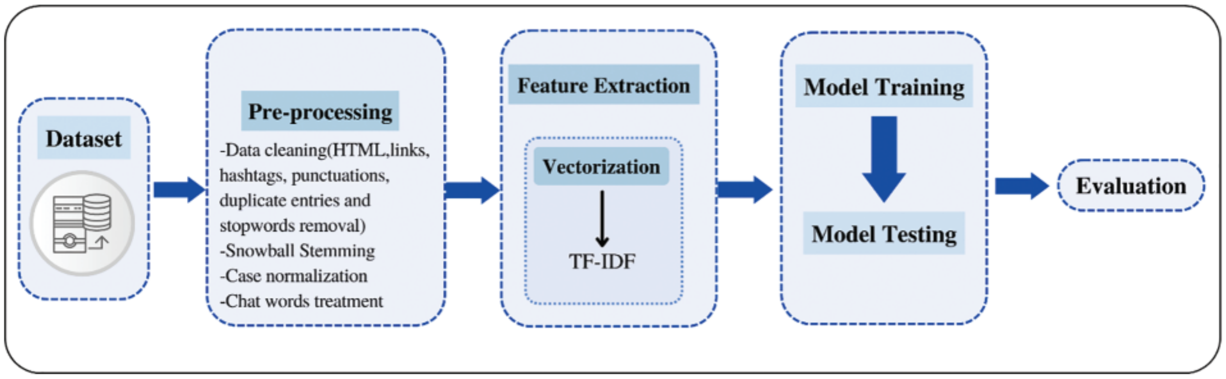


Figure 1: Systematic flow of the suggested method

3.1 Pre-Processing

After data collection, pre-processing is performed. Pre-processing refers to the process of preparing a dataset before the use of any algorithm [16]. Our approach includes the following pre-processing steps:

3.1.1 Data Cleaning

In data cleaning, html tags, links, punctuation, duplicate entries, and stop words are eliminated from both datasets.

3.1.2 Stemming

Stemming is a method for reducing words to their root or base form in natural language processing. The stem of the word "jumping," for example, is "jump," and the stem of the word "jumps" is also "jump." This is done to group related words together and minimize the dimensionality of the data, which can help some machine learning algorithms perform better. Snowball stemming are used in this experiment [17]. For many languages, snowball stemmers are variations on the Porter Stemmer. They provide better accuracy and support for languages other than English, such as French, German, Spanish, and others.

3.1.3 Case Normalization

Case normalization is also a pre-processing step that is used to convert all the text in a dataset to a consistent case. This is often done to ensure that the text is in a standardized format and to prevent issues that can arise from variations in case, so all the data is converted into lowercase [14].

3.1.4 Chat Words Treatment

One common pre-processing step is to treat chat words, or informal words commonly used in online chat or text messaging. The Text Blob library in Python provides several methods for text pre-processing, including a method for dealing with chat words. The process of "chat words treatment" involves the expansion of abbreviated phrases like "W8," "ASAP," and others into their full forms.

3.2 Feature Extraction

After preprocessing, feature extraction is performed to extract features from the dataset. The process of creating new features or variables from existing ones which is used as input for machine learning models is known as feature extraction. The main purpose of feature extraction is to create a set of features that represent the important patterns found in the data and are useful for the task at hand [18]. For this experiment study, TF-IDF is used.

3.2.1 TF-IDF

In text classification projects, TF-IDF is a common measurement. Inverse papers regularity and phrase regularity are the two ratings on the TF-IDF. By dividing the total number of records in which a given term appears by the total number of records, it is possible to determine the Inverse Document Frequency. Simply keeping track of how frequently a specific phrase appears in a document will allow to calculate term frequency. Combining these ideas gives us a ranking that gives terms that frequently appear in a small number of records a higher ranking and terms that frequently appear in all documents a lower ranking, enabling us to find critical conditions in a document [19].

TF-IDF have been performing better than many feature extraction techniques in various cases. TF-IDF is widely used due to its simplicity, interpretability, efficiency, and as a baseline for fair comparison because TF-IDF weighs the importance of words in a document based on their frequency within that document and their rarity across a larger collection of documents. This helps to filter out common words that are less informative and to emphasize more meaningful and context-specific terms. TF-IDF can be calculated by:

$$TF - IDF = TF * IDF \tag{1}$$

3.3 Machine Learning Techniques

Some of the well-known approaches to machine learning that have been implemented in this research study include the Random Forests Classifier (RFC), the Gradient Boosting Classifier (GBC), Multinomial Naïve Bayes (MNB), K-Nearest Neighbor (KNN), Linear Support Vector Machine (LSVM), the Passive Aggressive Classifier (PAC), Decision Trees (DT), and Logistic Regression (LR). Additional details can be found below.

3.3.1 Random Forest Classifier (RFC)

The random forest classifier (RFC), which was first proposed by Breiman in 2001 [20], has proven to be an extremely effective method of classification and regression for general purposes. It is a tree of different algorithms that can be applied to decision trees, and it can be used for both regression and classification. Most of the time, more trees mean better performance and efficiency in this algorithm. Use the bootstrap method to get a sample set of data points from a given training set. Based on the last step, choose a tree. The first two steps can count trees (in our case 100). Each tree planted votes for the set of data and then the most-voted classification tree [20].

To find the best hyperparameters tuning for a random forest classifier. The parameter grid specifies different values to try for the number of estimators, maximum depth, minimum samples split, and minimum samples leaf. The grid search uses cross-validation with 5 folds to evaluate the performance of each parameter combination using the accuracy metric. After fitting the grid search on the training data, it retrieves the best parameters and the corresponding best score.

3.3.2 Naïve Bayes

The Naïve Bayes algorithm is used to categorize data based on probabilities. This algorithm performs fantastically even with millions of records. It simply classifies data using different probabilities and the Bayes theorem. According to the Naive Bayes model, the class with the highest probability is the predicted class. Maximum a Posterior is another name for Nave Bayes. In numerous fields, Nave Bayes has advantages and disadvantages. The algorithm is quick and incredibly scalable. It is applied to both binary and multiclass classification. It can be applied to small datasets as well, producing good results [21].

Naive Bayes algorithm has several variations, including Multinomial Naive Bayes, Bernoulli Naive Bayes, Gaussian Naive Bayes, Complement Naive Bayes, and Categorical Naive Bayes. In this experiment, Multinomial Nave Bayes is used to analyze sentiments in movie reviews. To find the best hyperparameter tuning value for the smoothing parameters in the Multinomial Naive Bayes classifier, grid search is considered. The parameter grid specifies a range of alpha values are tested. The grid search function with cross-validation of 5 folds, taking all the parameters and uses accuracy as the scoring metric. It fits the grid search model to the training data in parallel, evaluating all combinations of hyperparameters. The resulting grid search object can be used to access the best hyperparameter value and other useful information.

3.3.3 K-Nearest Neighbor

The K-Nearest Neighbor (KNN) algorithm is a machine learning technique that is simple to implement but highly effective. Both classification and regression analysis benefit from its implementation. On the other hand, its most common application is in classification prediction. The K-Nearest Neighbor algorithm groups data into cohesive clusters or subsets and makes predictions for new data based on its similarity to previously trained data. The input is put into the category that best fits it according to which class it shares the nearest neighbors with [22].

A grid search using cross-validation to find the best value for the number of neighbor's parameter (k) in a K-Nearest Neighbor (KNN) classifier is used to tune the model results. The possible values being tested are 3, 5, 7, and 9. The grid search evaluates the performance of the KNN classifier using 5-fold cross-validation and the accuracy scoring metric. The goal is to find the value of k that results in the highest accuracy for the classifier.

3.3.4 Linear Support Vector Machines (LSVMs)

Linear Support Vector Machines (LSVMs) are a type of supervised learning method that can be used for both classification and regression. A hyperplane is what SVMs use to divide the classes. Regression works very well with this algorithm, and SVM's effect grows as the number of dimensions goes up. SVM also works well when the number of dimensions is bigger than the number of samples [23]. SVM applies cross-validation to improve computational efficiency.

Grid search is used to find the best hyperparameters for a Linear Support Vector Machine (SVM) classifier. The regularization parameter 'C' and the term 'penalty' are the hyperparameters being tuned.

The grid search is executed using 5 folds of cross-validation. The 'dual' parameter is set to False, indicating that solving the optimization problem with more samples than features is more efficient, and the best hyperparameters are determined based on the highest accuracy achieved during cross-validation.

3.3.5 Decision Tree (DT)

Decision Tree is a supervised machine learning model which used for regression and classification. The training data in the decision tree is consider as a root node. A decision tree is a structured representation of these mapping relationships. A tree can be either a single leaf node assigned to a specific category or a larger structure consisting of a root node connected to two or more subtrees. The way in which an instance's attributes are set allows a test node to predict the outcome. One of the subtrees is associated with each conceivable outcome. A root node is the first place you look when you are trying to classify something. The instance's result is determined if this node is a test, and the process proceeds with the appropriate subtree if so. An instance's predicted class is displayed on the leaf's label upon discovery.

A decision tree can be created from a collection of cases using the "divide and conquer" approach. When all the nodes in the tree belong to the same class, the node becomes a leaf and the corresponding label is assigned to the leaf. If that isn't the case, then a test is selected that yields different results in at least two of the situations. As a consequence of this finding, the cases are classified differently. The test itself is represented by the first node in the tree. Applying the same procedure to the subset of instances that have that result yields the corresponding subtree [24]. To achieve good results, a grid search for hyperparameter tuning using the Decision Tree Classifier is used. The hyperparameters tuning includes the splitting criterion, maximum depth, minimum samples for splitting, and minimum samples for leaf nodes. By exhaustively searching over the defined parameter grid, the grid search identifies the best classifier. This approach helps optimize the decision tree model's performance by selecting the most suitable hyperparameters for the given dataset.

3.3.6 Logistic Regression (LR)

Logistic regression is a supervised machine learning algorithm that was created to help with classification problems. When the target variable is categorical, the problem is referred to as a classification learning problem. The goal of logistic regression is to predict the probability that a new example belongs to one of the target classes by mapping a function from the dataset's features to the targets [25].

This algorithm is sometimes referred to by its alternate name, Maximum Entropy. The generalized linear models' family of mathematical constructions includes the logistic regression classification algorithm. The modeling method known as logistic regression is used to describe the probabilities of a trial's outcomes [26]. For hyperparameters tuning, a parameter grid containing different values for the hyperparameters 'C', 'penalty', and 'solver' to be used in a logistic regression model. It then creates an instance of the logistic regression model and sets up a grid search using cross-validation of 5 folds to find the best combination of hyperparameters that maximizes accuracy as the scoring metric.

3.3.7 Gradient Boosting Classifier (GBC)

To quickly and effectively learn non-linear classification and regression models, such as decision trees and regression trees, Gradient Boosting Classifiers (GBCs) are increasingly popular. Through the gradual introduction of new learners, we can simulate a collection of weak prediction models, such as regression decision trees. Consisting of a series of nodes and leaves, it uses the results of previous nodes to predict future events. When improved collectively, regression trees outperform their individual counterparts [27]. A grid search using cross-validation of 5 folds finds the best hyperparameters for a Gradient Boosting classifier. The hyperparameters are tuned on the number of estimators, the learning rate, and the maximum depth of trees.

3.3.8 Passive Aggressive Classifier (PAC)

The PAC is an important classifier used in online learning algorithms. The classification function is changed if there is a mistake in newly seen data or if its classification score does not go over a certain limit. The PAC algorithm has been shown to be a very useful and popular way for people to learn online and use what they have learned to solve problems in the real world. PAC is an online learning classifier that is used to keep an eye on data every day, every week, such as with news, social media, etc. The main idea behind this algorithm is that it looks at data, learns from it, and then gets rid of it without having to store it. When a mistake is made, the algorithm reacts quickly by changing the values. When a mistake is not made, it reacts slowly or passively. So, the name is a passive aggressive classifier [28].

A grid search is used to find the optimal hyperparameters for the model. It defines a parameter grid with different values for regularization parameter, fit intercept, and maximum number of iterations. The grid search class is used to perform the search, with settings for scoring metrics, cross-validation of 5 folds, and parallel processing. The goal was to identify the hyperparameter combination that gives the highest accuracy for the model.

3.4 Hyperparameters Tuning

Hyperparameters are parameters that are not learned from the data during training but are set manually before starting the training process. They represent higher-level configuration choices for the machine learning algorithm, and their values are typically based on the characteristics of the data being used and the algorithm's ability to learn from that data. These hyperparameters serve as instructions or constraints for the learning algorithm, controlling its behavior and performance. Examples of common hyperparameters include the learning rate, regularization strength, depth of a decision tree, and the type of kernel used in a Support Vector Machine.

The selection of appropriate hyperparameters is important as they can greatly impact the model's ability to generalize and make accurate predictions. Hyperparameter tuning involves systematically exploring different combinations of hyperparameter values to find the optimal settings that maximize the model's performance on a validation set or through cross-validation [29]. There are several approaches to hyperparameter tuning in machine learning which are grid search, random search and Bayesian optimization.

3.4.1 Grid Search

One of the most popular techniques for exploring the hyper-parameter configuration space is grid search (GS). It evaluates all the hyper-parameter combinations provided to the grid of configurations and can be viewed as an exhaustive search or brute-force technique. The way it operates is by analyzing the Cartesian product of a user-specified finite set of values [30]. For this experiment grid search is used due to its simplicity, easy to implement and effectiveness. Grid search searches through all possible hyperparameter combinations within a predefined grid, making it a comprehensive approach to hyperparameter tuning.

4 Experimentation Setup

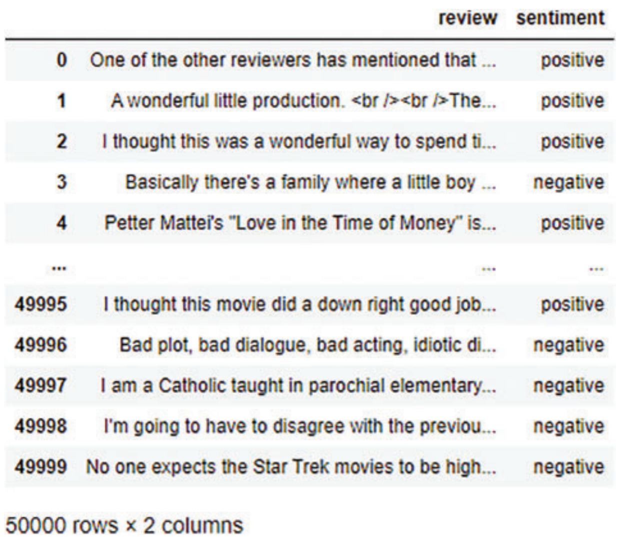
The computer used for this experiment was running Windows 10 Pro. The computer had a Core i5 10th generation processor and 16 GB of RAM. The experiment was performed using the Jupiter Lab development environment, which is an open-source, web-based interactive development environment (IDE) for scientific computing. The Jupiter Lab IDE was accessed through an Anaconda emulator. For use in fields like data science, machine learning, and scientific computing, Anaconda provides a Python and R distribution. It provides a convenient and easy-to-use environment for managing packages and dependencies and running code. By using Anaconda emulator, the user can launch Jupiter lab with all the necessary libraries and packages pre-installed, making the experiment setup and execution process more seamless and efficient.

4.1 Datasets

To compare these machine learning algorithms two movie reviews datasets are taken which are IMDB Dataset of 50K Movie Reviews and Sentiment Polarity Dataset Version 2.0.

4.1.1 IMDB Dataset of 50K Movie Reviews

This is a binary sentiment classification dataset that has a significantly increased amount of data in comparison to earlier benchmark datasets. There are 25,000 extremely polar movie reviews that can be used for training, and there are 25,000 that can be used for testing [31]. The dataset contains two columns: the first one, titled "review," and the second one, titled "sentiment,", respectively. The snapshot of dataset is shown Fig. 2.



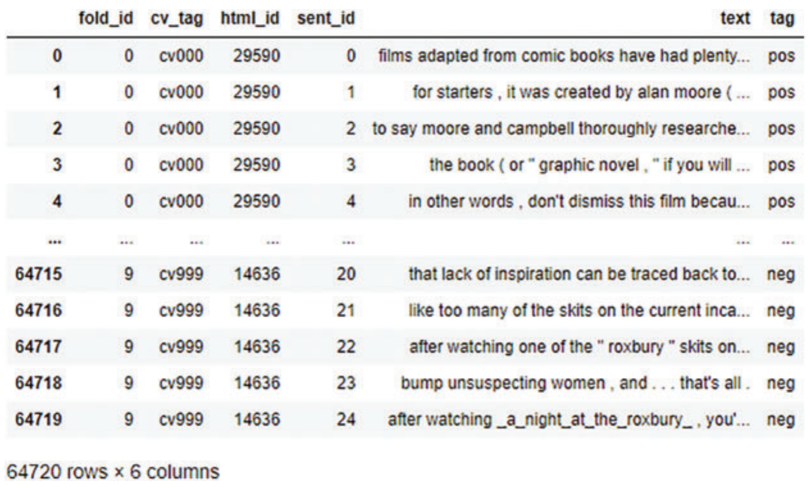
	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. <br /><br />The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive
...	...	...
49995	I thought this movie did a down right good job...	positive
49996	Bad plot, bad dialogue, bad acting, idiotic di...	negative
49997	I am a Catholic taught in parochial elementary...	negative
49998	I'm going to have to disagree with the previou...	negative
49999	No one expects the Star Trek movies to be high...	negative

50000 rows × 2 columns

Figure 2: Snapshot of IMDB dataset of 50K reviews

4.1.2 Sentiment Polarity Dataset Version 2.0

The Sentiment Polarity Dataset Version 2.0 is made by Lillian Lee and Bo Pang. With the authors' permission, this dataset is being redistributed with NLTK. This dataset consists of 63K reviews, half are positive and half are negative [32]. The snapshot of dataset is shown in Fig. 3.



	fold_id	cv_tag	html_id	sent_id	text	tag
0	0	cv000	29590	0	films adapted from comic books have had plenty...	pos
1	0	cv000	29590	1	for starters , it was created by alan moore (...	pos
2	0	cv000	29590	2	to say moore and campbell thoroughly researche...	pos
3	0	cv000	29590	3	the book (or " graphic novel , " if you will ...	pos
4	0	cv000	29590	4	in other words , don't dismiss this film becau...	pos
...	...	...	...	...	...	...
64715	9	cv999	14636	20	that lack of inspiration can be traced back to...	neg
64716	9	cv999	14636	21	like too many of the skits on the current inca...	neg
64717	9	cv999	14636	22	after watching one of the " roxbury " skits on...	neg
64718	9	cv999	14636	23	bump unsuspecting women , and . . . that's all .	neg
64719	9	cv999	14636	24	after watching _a_night_at_the_roxbury_ , you'...	neg

64720 rows × 6 columns

Figure 3: Snapshot of Sentiment Polarity Dataset Version 2.0

4.2 Performance Metrices

The model used in this study was evaluated using training time, precision, recall, F1-score and accuracy [33]. Following are the parameters:

Training Time: The time it takes for the model to begin training until it is fully trained and shows results. The training time is measured in seconds.

Training Time = End Time – Start Time

TP: True Positive would refer to a situation where the classifier correctly predicts that a movie review is positive. For example, if a movie review is actually positive, and the classifier predicts it to be positive, then it would be considered a True Positive prediction.

TN: True Negative means the number of negative movie reviews that were correctly classified as negative.

FP: False Positive refers to a situation where a movie review is predicted as positive but it is actually negative.

FN: False Negative refers to a situation where a movie is predicted to be negative but it is actually positive.

By dividing the number of predicted reviews by the total number of reviews, the accuracy is determined.

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

The number of reviews that were correctly predicted as positive is divided by the total number of reviews that were predicted to be positive to determine the Precision.

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

Recall is calculated by dividing the total number of reviews in that class by the number of reviews that were correctly predicted as positive.

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

The weighted average of recall and precision is used to calculate the F1-score.

$$\text{F1-score} = \frac{2 * Precision * recall}{Precision + recall}$$

5 Results and Discussion

In this experiment, the results are compared in two parts. The results of the models on the IMDB Dataset are compared with each other in Section 5.1, and the results are then compared with the hyperparameters of the models on the IMDB Dataset in Section 5.1.1.

Similar to the first part, the second part compares the results of the models on the Sentiment Polarity Dataset v2 in Section 5.2, and then the results of the models when hyperparameters are applied in Section 5.2.1. The different bold results in Tables 1–4 indicate the models that achieved the best performance in terms of the respective evaluation metrics.

Table 1: Performance comparison of models on IMDB movie reviews dataset

Models	Precision	Recall	F1-score	Accuracy	Training time
Linear Support Vector Machine	**88.82%**	90.16%	**89.49%**	**89.48%**	0.81 s
Logistic Regression	87.99%	**90.86%**	89.4%	89.31%	6.18 s
Passive Aggressive ClasSifier	86.63%	87.63%	87.13%	87.14%	0.42 s
Multinomial Naïve Bayes	86.13%	85.99%	86.06%	86.17%	0.09 s
Random Forest Classifier	84.27%	85.53%	84.90%	84.89%	208.67 s
Gradient Boosting Classifier	77.67%	86.90%	82.03%	81.09%	774.43
K-Nearest Neighbor	74.94%	84.4%	79.39%	78.25%	29.00
Decision Tree	71.07%	71.79%	71.43%	71.49%	63.99

Table 2: Hyperparameters tuning performance of models on IMDB movie reviews dataset

Models	Precision	Recall	F1-score	Accuracy	Training time
Linear Support Vector Machine	88.98%	91.06%	90%	89.96%	43.99 s
Logistic Regression	89%	91.03%	90%	89.96%	1140.20 s
Passive Aggressive Classifier	**89.59%**	**90.96%**	**90.27%**	**90.27%**	37.72 s
Multinomial Naïve Bayes	85.83%	87.34%	86.58%	86.56%	0.611 s
Random Forest Classifier	84.26%	87.52%	85.86%	85.68%	8374.85 s
Gradient Boosting Classifier	82.58%	86.89%	82.68%	84.21%	14024.37 s
K-Nearest Neighbor	74.28%	85.64%	79.56%	77.92%	4733.60 s
Decision Tree	75.03%	74.41%	74.22%	74.36%	5540.41 s

Table 3: Performance comparison of models on Sentiment Polarity Dataset Version 2.0

Models	Precision	Recall	F1-score	Accuracy	Training time
Linear Support Vector Machine	**70.16%**	69.35%	69.75%	69.16%	0.34 s
Logistic Regression	69.56%	70.99%	70.27%	69.20%	1.01 s
Passive Aggressive Classifier	66.45%	68.99%	67.70%	66.24%	0.48 s
Multinomial Naïve Bayes	70%	74.99%	**72.41%**	**70.69%**	0.04 s
Random Forest Classifier	65.16%	65.18%	65.17%	64.28%	247.57 s
Gradient Boosting Classifier	55.92%	**86.46%**	67.91%	58.11%	93.80 s
K-Nearest Neighbor	52.62%	45.37%	48.73%	51.04%	13.98 s
Decision Tree	58.77%	58.6%	58.69%	57.7%	25.42 s

Table 4: Hyperparameters performance results on models on Sentiment Polarity Dataset Version 2

Models	Precision	Recall	F1-score	Accuracy	Training time
Linear Support Vector Machine	70.39%	70.07%	70.23%	69.54%	43.49 s
Logistic Regression	70.07%	70.27%	70.17%	69.37%	24.05 s
Passive Aggressive Classifier	70.36%	70.88%	70.62%	69.76%	35.2 s
Multinomial Naïve Bayes	**72.45%**	70.22%	**71.32%**	**71.04%**	0.57 s
Random Forest Classifier	63.49%	71.32%	67.18%	64.27%	7128.54 s
Gradient Boosting Classifier	58.03%	81.36%	67.74%	60.76%	2058.31 s
K-Nearest Neighbor	51.51%	**97.97%**	67.52%	52.64%	1474.46 s
Decision Tree	58.15%	58.12%	58.14%	58.13%	2351.60 s

5.1 Results on IMDB Dataset

For the IMDB Dataset of 50K reviews, train test split was performed in which the test size was 25% and training size was 75%. After applying these models, Linear Support Vector Machines performed the highest accuracy among the models, with a maximum accuracy of 89.48% without using hyperparameters tuning. The results are shown in Table 1 and Fig. 4.

The performance visualization of IMDB Dataset is visualized in Fig. 4.

5.1.1 Hyperparameters Tuning Results

The hyperparameters tuning results of models used in this study are shown below in Table 2.

The Passive Aggressive Classifier achieved higher accuracy than Logistic Regression and Linear SVM because of more flexible and adaptable learning after applying grid search and hyperparameter tuning that better captured complex patterns and relationships in the IMDB dataset.

On the other hand, K-Nearest Neighbor and Decision Tree achieved the lowest accuracy than other models used in this study. The lower accuracy of K-Nearest Neighbor (KNN) can be attributed to its sensitivity to irrelevant or noisy features in the dataset, as well as the computational complexity that increases with larger datasets.

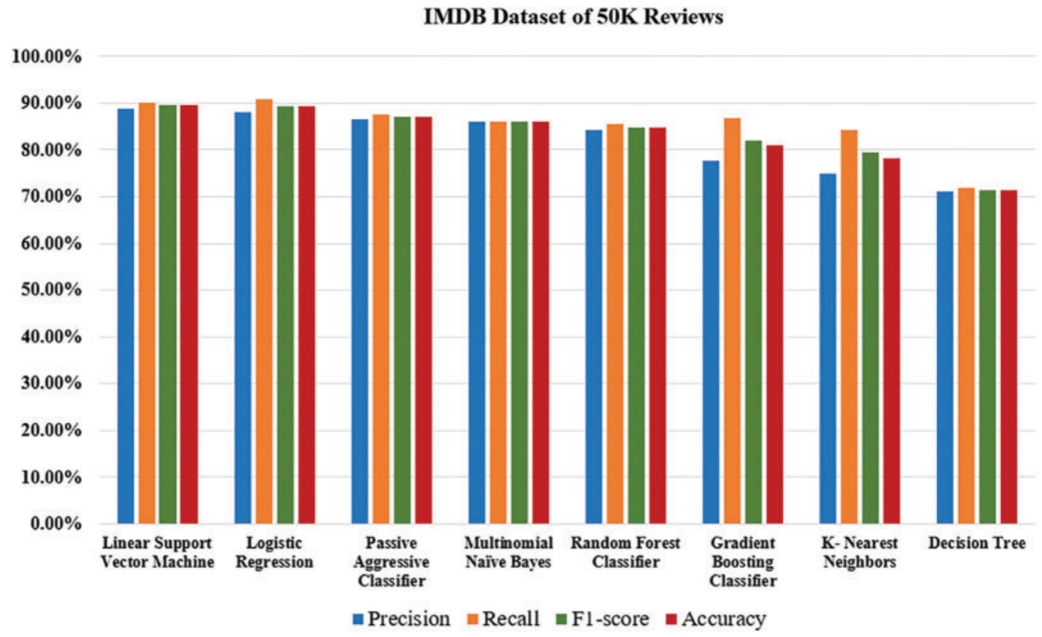


Figure 4: Performance evaluation of models on IMDB movie reviews dataset

The lower accuracy of the Decision Tree model is due to its tendency to overfit the training data with complex decision boundaries, resulting in reduced generalization, as well as its high variance and sensitivity to small changes, which leads to instability and lower accuracy on the IMDB dataset. The comparison of model accuracies used in this study with and without hyperparameter tuning are shown in Fig. 5.

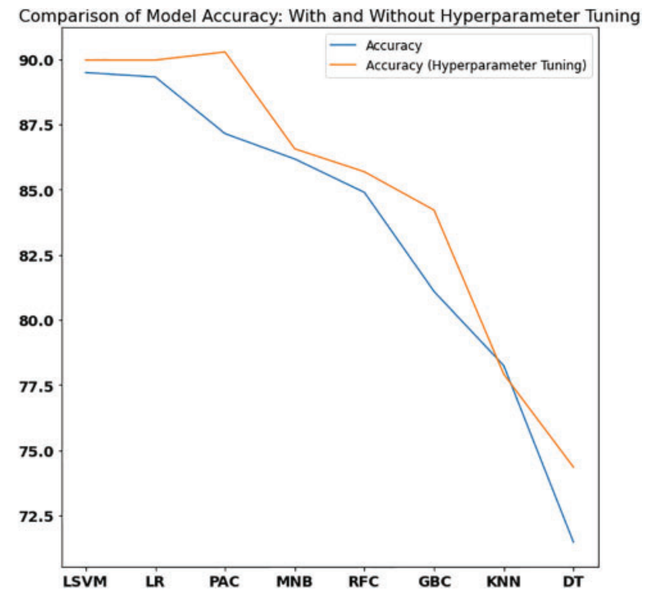


Figure 5: Comparison of model accuracy on IMDB dataset with and without hyperparameter tuning

5.2 *Sentiment Polarity Dataset 2.0*

For the Sentiment Polarity Dataset, train test split was performed in which the test size was 25% and training size was 75%. After applying, some results were achieved. From the following machine learning techniques, Multinomial Naïve Bayes gives the maximum accuracy, from which 70.69% of accuracy was achieved. Other results are given below in Table 3 and Fig. 6.

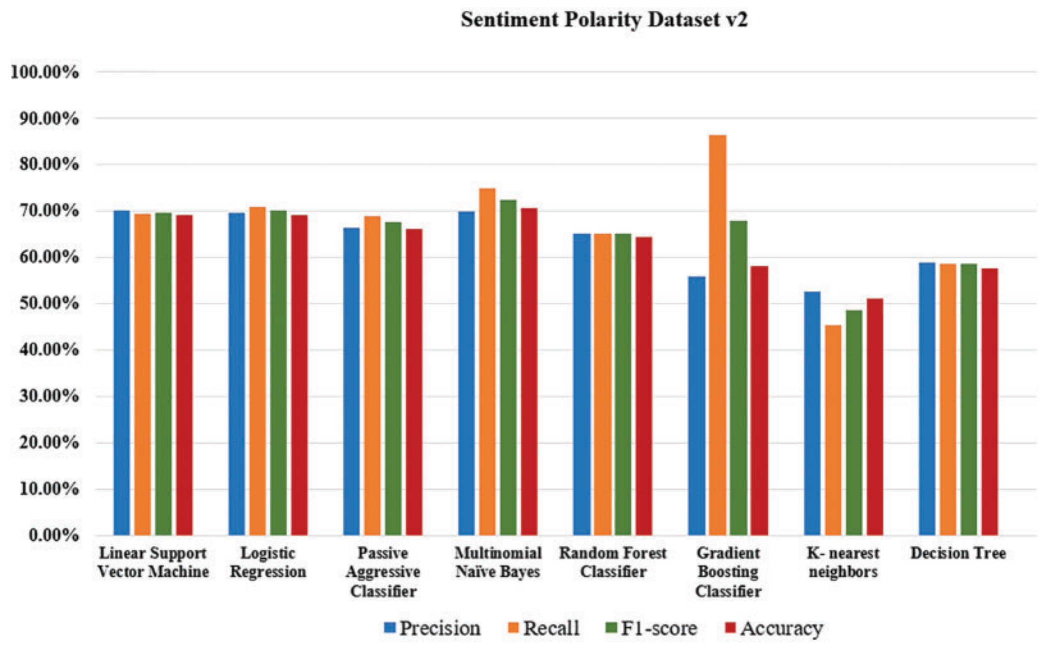


Figure 6: Graphical performance evaluation of models on Sentiment Polarity Dataset Version 2

The performance measurement of Sentiment Polarity Dataset is visualized in Fig. 6.

Among the models compared on the Sentiment Polarity Dataset Version 2.0, Multinomial Naïve Bayes (MNB) performed the best accuracy score of 70.69% then other models. Multinomial Naïve Bayes is known for its ability to handle larger datasets efficiently. The assumption of conditional independence between features (words) given the class in MNB allows it to scale well with the number of features, making it suitable for text classification tasks with a high number of words or features [34]. This scalability contributes to its strong performance on larger datasets. MNB's ability to model the probabilities of words appearing in each sentiment class helps it capture sentiment patterns more effectively.

On the other hand, Decision Tree (DT) and K-Nearest Neighbor (KNN) performed the worst, potentially due to DT's tendency to create complex decision boundaries and overfit the training data, as well as KNN's sensitivity to irrelevant features and challenges in handling high-dimensional text data.

5.2.1 Hyperparameters Tuning Results

The hyperparameters tuning results of models on Sentiment Polarity Dataset Version 2 are shown below in Table 4.

The comparison of model accuracies used in this study with and without hyperparameter tuning are shown in Fig. 7.

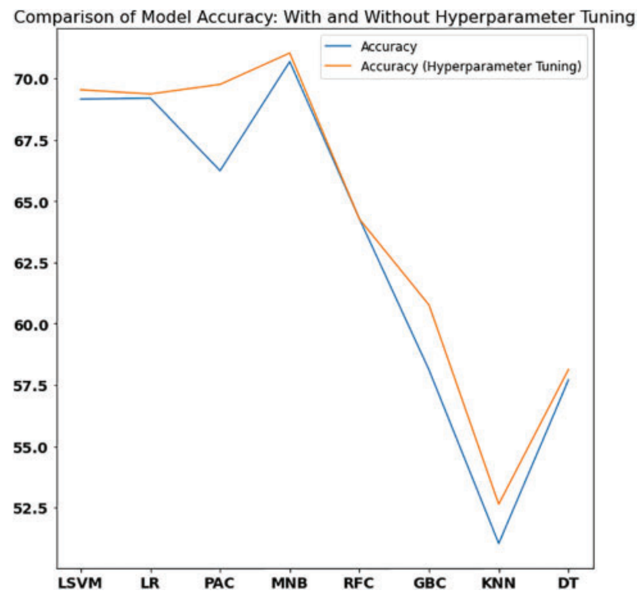


Figure 7: Comparison of model accuracy on Sentiment Polarity Dataset Version 2.0 with and without hyperparameter tuning

According to the results of hyperparameter tuning on the Sentiment Polarity Dataset Version 2, Multinomial Naive Bayes dominates with a maximum accuracy of 71.04%, but the passive aggressive classifier outperformed Linear SVM and Logistic Regression once more. In terms of recall, K-Nearest Neighbor and Decision tree outperformed other models due to their ability to identify a greater proportion of positive or negative sentiment instances.

6 Conclusion

Opinion mining also called sentiment analysis is the extraction of subjective data from textual data. The paper focuses on a comparison of Linear Support Vector Machines and Multinomial Naive Bayes with baseline machine learning algorithms for extracting sentiments from movie reviews. The findings of the study discovered that the Linear Support Vector Machines achieved a maximum accuracy of 89.48% on the IMDB movie reviews dataset containing 50,000 reviews. After applying hyperparameter tuning, both Linear Support Vector Machines and Logistic Regression improved their accuracy to 89.96%. However, the Passive Aggressive Classifier surpassed them all with a maximum accuracy of 90.27% on the IMDB dataset. While on Sentiment Polarity Dataset Version 2.0, Multinomial Naive Bayes appeared as the leading algorithm with a maximum accuracy of 71.04%. In this study, it was discovered that Linear Support Vector Machines and Logistic Regression performed better on high-dimensional textual data and Multinomial Naïve Bayes performs better on larger datasets than other classifiers used in this study. Naïve Bayes, Passive Aggressive Classifiers, Logistic Regression, and LSVM, which achieved the highest accuracy, can serve as benchmark models for future research in sentiment analysis, providing valuable information to enhance prediction performance. Our future work will focus on two main directions: firstly, conducting a comprehensive comparative analysis of the latest techniques in sentiment analysis for movie reviews, and secondly, exploring deep learning models to further enhance sentiment analysis in this domain.

Acknowledgement: We would like to express our sincere gratitude to all the individuals and organizations who have contributed to the successful completion of this research paper. We also acknowledge the contribution of our participants, who provided valuable feedback and suggestions that helped to refine this research methodology and enhance the quality of the findings. We would like to thank the participants who generously gave their time and participated in our study, without whom this research would not have been possible. Thank you all for your contributions to this research paper.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: SSK and BK; data collection: MMD, MK and MBG; analysis and interpretation of results: MMD, SSK, MA and BK; draft manuscript preparation: MMD and SSK. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The datasets analyzed during the current study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] B. L. Devi, V. V. Bai, S. Ramasubbareddy and K. Govinda, "Sentiment analysis on movie reviews," in *Emerging Research in Data Engineering Systems and Computer Communications*. Singapore: Springer, pp. 321–328, 2020.

[2] D. S. Sisodia, S. Bhandari, N. K. Reddy and A. Pujahari, "A comparative performance study of machine learning algorithms for sentiment analysis of movie viewers using open reviews," in *Performance Management of Integrated Systems and its Applications in Software Engineering*. Singapore: Springer, pp. 107–117, 2020.

[3] S. Poria, E. Cambria and A. Gelbukh, "Aspect extraction for opinion mining with a deep convolutional neural network," *Knowledge-Based Systems*, vol. 108, no. 2, pp. 42–49, 2016.

[4] S. S. Khan, M. Khan, Q. Ran and R. Naseem, "Challenges in opinion mining, comprehensive review," *A Science and Technology Journal*, vol. 33, no. 11, pp. 123–135, 2018.

[5] K. Ullah, A. Rashad, M. Khan, Y. Ghadi, H. Aljuaid *et al.,* "A deep neural network-based approach for sentiment analysis of movie reviews," *Complexity*, 2022. https://doi.org/10.1155/2022/5217491

[6] N. O. F. Daeli and A. Adiwijaya, "Sentiment analysis on movie reviews using information gain and K-nearest neighbor," *Journal of Data Science and its Applications*, vol. 3, no. 1, pp. 1–7, 2020.

[7] J. D. Bodapati, N. Veeranjaneyulu and S. Shaik, "Sentiment analysis from movie reviews using LSTMs," *Ingenierie des Systemes d'Information*, vol. 24, no. 1, pp. 125–129, 2019.

[8] A. Rahman and M. S. Hossen, "Sentiment analysis on movie review data using machine learning approach," in *2019 Int. Conf. on Bangla Speech and Language Processing (ICBSLP)*, Sylhet, Bangladesh, pp. 1–4, 2019.

[9] K. Chakraborty, S. Bhattacharyya, R. Bag and A. E. Hassanien, "Comparative sentiment analysis on a set of movie reviews using deep learning approach," in *Int. Conf. on Advanced Machine Learning Technologies and Applications*, Cairo, Egypt, pp. 311–318, 2018.

[10] A. V. Mohan Kumar and A. N. Nanda Kumar, "Sentiment analysis using robust hierarchical clustering algorithm for opinion mining on movie reviews-based applications," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 8, pp. 452–457, 2019.

[11] R. Malini and M. R. Sunitha, “Opinion mining on movie reviews,” in *2019 1st Int. Conf. on Advances in Information Technology (ICAIT)*, Chikmagalur, India, pp. 282–286, 2019.
[12] P. Baid, A. Gupta and N. Chaplot, “Sentiment analysis of movie reviews using machine learning techniques,” *International Journal of Computer Applications*, vol. 179, no. 7, pp. 45–49, 2017.
[13] T. P. Sahu and S. Ahuja, “Sentiment analysis of movie reviews: A study on feature selection & classification algorithms,” in *2016 Int. Conf. on Microelectronics, Computing and Communications (MicroCom)*, Durgapur, India, pp. 1–6, 2016.
[14] A. S. Manek, P. D. Shenoy and M. C. Mohan, “Aspect term extraction for sentiment analysis in large movie reviews using Gini index feature selection method and SVM classifier,” *World Wide Web*, vol. 20, no. 2, pp. 135–154, 2017.
[15] M. Avinash and E. Sivasankar, “A study of feature extraction techniques for sentiment analysis,” in *Advances in Intelligent Systems and Computing*, Singapore: Springer, vol. 814, pp. 475–486, 2019. https://doi.org/10.1007/978-981-13-1501-5_41
[16] Å. Rinnan, L. Nørgaard, F. van den Berg, J. Thygesen, R. Bro *et al.,* “Data pre-processing,” *Infrared Spectroscopy for Food Quality Analysis and Control*, pp. 29–50, 2009. https://doi.org/10.1016/B978-0-12-374136-3.00002-X
[17] D. Khyani, B. S. Siddhartha, N. M. Niveditha and B. M. Divya, “An interpretation of lemmatization and stemming in natural language processing,” *Journal of University of Shanghai for Science and Technology*, vol. 22, pp. 350–357, 2021.
[18] M. Z. Asghar, A. Khan, S. Ahmad and F. M. Kundi, “A review of feature extraction in sentiment analysis,” *Journal of Basic and Applied Scientific Research*, vol. 4, no. 3, pp. 181–186, 2014.
[19] T. O’Keefe and I. Koprinska, “Feature selection and weighting methods in sentiment analysis,” in *Proc. of the 14th Australasian Document Computing Symp.*, Sydney, Australia, pp. 67–74, 2009.
[20] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
[21] M. Rathi, A. Malik, D. Varshney, R. Sharma and S. Mendiratta, “Sentiment analysis of tweets using machine learning approach,” in *2018 Eleventh Int. Conf. on Contemporary Computing (IC3)*, Noida, India, pp. 1–3, 2018.
[22] K. Taunk, S. De, S. Verma and A. Swetapadma, “A brief review of nearest neighbor algorithm for learning and classification,” in *2019 Int. Conf. on Intelligent Computing and Control Systems (ICCS)*, Madurai, India, pp. 1255–1260, 2019.
[23] G. Angulakshmi and R. M. Chezian, “Three level feature extraction for sentiment classification,” *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, no. 8, pp. 5501–5507, 2014.
[24] J. R. Quinlan, “Learning decision tree classifiers,” *ACM Computing Surveys*, vol. 28, no. 1, pp. 71–72, 1996.
[25] E. Bisong and E. Bisong, “Logistic regression,” in *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*. Berkeley, CA, USA: Apress, pp. 243–250, 2019.
[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion *et al.,* “Scikit-learn: Machine learning in Python,” *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[27] N. Chakrabarty, T. Kundu, S. Dandapat, A. Sarkar and D. K. Kole, “Flight arrival delay prediction using gradient boosting classifier,” *Emerging Technologies in Data Mining and Information Security*, vol. 2, pp. 651–659, 2019. https://doi.org/10.1007/978-981-13-1498-8_57
[28] S. Gupta and P. Meel, “Fake news detection using passive-aggressive classifier,” In: G. Ranganathan, J. Chen, Á. Rocha (Eds.), *Inventive Communication and Computational Technologies*, vol. 145. Singapore: Springer, 2021. https://doi.org/10.1007/978-981-15-7345-3_13
[29] T. Agrawal, “Introduction to hyperparameters,” in *Hyperparameter Optimization in Machine Learning: Make Your Machine Learning and Deep Learning Models More Efficient*. New York, NY, USA: Apress, pp. 1–8, 2021.
[30] L. Yang and A. Shami, “On hyperparameter optimization of machine learning algorithms: Theory and practice,” *Neurocomputing*, vol. 415, no. 1, pp. 295–316, 2020.

[31] A. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng *et al.,* "Learning word vectors for sentiment analysis," in *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA, pp. 142–150, 2011.

[32] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proc. of the 42nd Annual Meeting on Association for Computational Linguistics (ACL-04)*, Barcelona, Spain, pp. 271–278, 2004.

[33] B. Khan, M. Arshad and S. S. Khan, "Comparative analysis of machine learning models for PDF malware detection: Evaluating different training and testing criteria," *Journal of Cybersecurity*, vol. 5, pp. 1–11, 2023.

[34] M. Khan, M. S. Khan and Y. Alharbi, "Text mining challenges and applications—A comprehensive review," *International Journal of Computer Science and Network Security*, vol. 20, no. 12, pp. 138, 2020.