Journal of Computers, Mechanical and Management
Journal homepage: www.jcmm.co.in
e-ISSN: 3009-075X
AN PUBLISHING

Volume 2 Issue 6

Article Number: 230108

Sentiment Analysis on IMDB Review Dataset

Shubham Kumar Singh*and Neetu Singla

Department of Computer Science and Engineering, School of Engineering, The NorthCap University, Gurugram, Haryana, India 122017

Abstract

A computational method known as sentiment analysis is employed to ascertain the emotional undertone or attitude of a text document, such as a review, tweet, or news story. Using machine learning models, deep neural network models, and natural language processing, the method entails examining the text to determine whether it expresses positive or negative sentiment. In this study, models like Naive Bayes, Logistic Regression, LSTM, LSVM, Decision tree, and BiLSTM are utilized to conduct a sentiment analysis (SA) study on the IMDB dataset. The goal of the investigation is to evaluate how well these models perform in retrospect on movie reviews, categorizing them as positive or negative. The study investigates the effects of data pre-processing methods and hyperparameter tuning on the models' accuracy. The final results demonstrate that the BiLSTM model outperforms the other models in terms of recall, precision, and accuracy, followed by the LSTM, Logistic Regression, LSVM, Decision Tree, and Naive Bayes models. The research emphasizes the potential of deep learning models—in particular, BiLSTM in sentiment analysis tasks, as well as the significance of hyper-parameter tuning and pre-processing methods in achieving high accuracy.

Keywords: Sentiment Analysis; IMDB Review Dataset; Machine Learning Models; Data Preprocessing; Model Performance Evaluation

1 Introduction

Natural language processing (NLP) has a subfield called sentiment analysis [1, 2] that aims to recognise subjectivities, attitudes, and moods in a given textual context. It offers a new way of approaching the traditional method to classify text. Classifying text based on emotion is one of the biggest challenging areas of research in NLP, with ongoing studies in text mining. Recently, sentiment analysis based on deep learning methods such as memory networks, CNN, RNN, and BiLSTM has been extensively explored. These approaches allow for the use of multiple contexts, which can help remove characteristics from training data. However, this method needs a lot of tagged training data, which might not always be accessible. Despite advances in algorithms and methodologies, sentiment analysis still faces some unresolved issues. One limitation is the inability to classify phrases that lack overt emotional keywords, which could lead to the incorrect inference that a sentence is emotionless. Additionally, depending on the context, certain keywords may have multiple meanings, resulting in ambiguity. The system must consider these constraints and provide accurate data classification, particularly when used in sensitive applications. A Python-based programme is used in this study to analyse the IMDb dataset. The training and testing portions of the dataset are separated in a distinct way. The Naive Bayes, Logistic Regression, LSTM, LSVM, Decision Tree, and BiLSTM classifiers are developed using the training phase. The classification precision is then calculated using the testing set.

*Corresponding author: singhshubham0051@gmail.com
Received: 14 November 2023; **Revised:** 02 December 2023; **Accepted**: 04 December 2023; **Published:** 31 December 2023

DOI: 10.57159/gadl.jcmm.2.6.230108.

2 Dataset

The dataset compiled by Andrew Maas was utilized for the study, consisting of 50,000 IMDb film reviews, divided into training and testing sets with 12.5k positive and negative reviews each. The IMDb rating system was applied to categorize the reviews. Various machine learning classifiers, including Logistic Regression, Naive Bayes, LSTM, LSVM, Decision Tree, and BiLSTM, were employed on the original feature sets for text-based sentiment analysis. The balance between positive and negative reviews in the dataset is illustrated in Figure 1, which shows an equal distribution of sentiments, ensuring a fair basis for training and testing the classifiers.

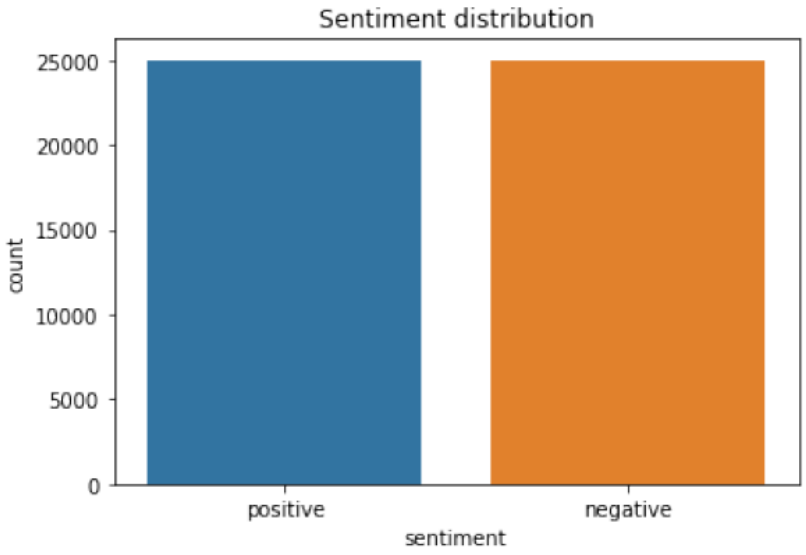

Figure 1: Distribution of Positive and Negative Sentiments in the IMDb Dataset

3 Data Division

It is common practice to split the dataset into training and testing vectors. The training vector encompasses data used for training the classifier. The training data may be divided into validation, testing, or both through various methods. Generally, training constitutes the majority of the data. In machine learning applications, an 80-20 split is often employed, where 20% of the data is used for testing and 80% for training. This division is influenced by the law of the vital few or the Pareto Principle, which is prominent in economic and financial theories. In this study, 10k reviews were selected, ensuring an equal distribution of positive and negative assessments to mitigate bias. The data was divided into 20% for testing and 80% for training. The 10-fold cross-validation method was used to eliminate bias in categorization results. Understanding the distribution of word counts in reviews is essential for preprocessing and feature engineering. Figure 2 presents histograms that illustrate the distribution of word counts in positive and negative reviews within the IMDb dataset. This information can inform decisions on text normalization and vectorization techniques.

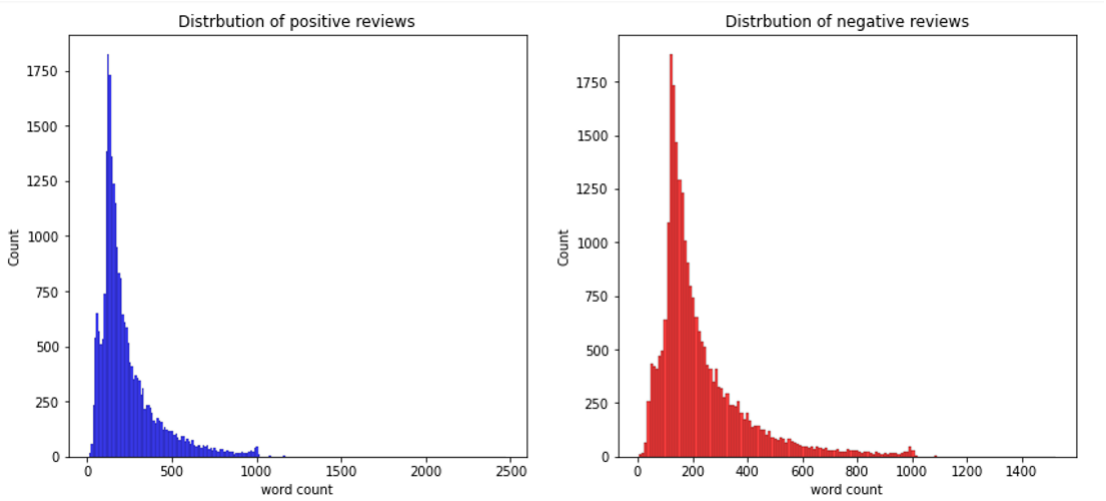

Figure 2: Histograms of Word Count Distribution in Positive and Negative IMDb Reviews

4 Related Work

In one of the previous works [3], researchers explored various methods in NLP and ML to achieve high accuracy. They specifically utilized the LSTM classifier to analyze sentiments in IMDb movie reviews, achieving a maximum classification accuracy of 89.9%. An article [4] demonstrated the use of NLP for Sentiment Analysis (SA) with conventional ML techniques. The study used Naive Bayes, Decision Tree, and Logistic Regression, with notable success in TF-IDF + Logistic Regression, yielding a validation AUC score of about 96%. Another study [5] introduced a ConvLstm neural network architecture, merging CNN and LSTM along with conditioned vector space models. This approach employed LSTM in place of the pooling layer in CNN, focusing on preserving long-term dependencies and minimizing the loss of complex local information in phrase sequences. As reported in one of the article[6], a new capsule network model named caps-BiLSTM was proposed for sentiment analysis. Integrating BiLSTM, this model showed encouraging results on various datasets, including MR, IMDB, and SST, surpassing deep learning models compared to traditional machine learning techniques. In a significant contribution [7], a semi-supervised learning technique was utilized with a limited amount of labeled data from the IMDb and YelpNYC datasets. This model outperformed baseline models like LSTM and SVM, especially in scenarios with minimal labeled data, such as when only 1% of the data is labeled. A noteworthy study [8] combined CNN and LSTM of Recurrent Neural Networks for sentiment analysis, aiming for higher accuracy with minimal loss and reduced computing time. The experiment revealed that CNN alone provided the best accuracy, while combining LSTM and CNN models yielded better efficiency in terms of speed and loss. In an insightful work [9], Sentiment Analysis on the IMDB Dataset was conducted using eight different models for classifying movie reviews. The results highlighted that the RF classifier was the most efficient, outperforming the other models in all evaluation metrics. Notably, KNN also achieved a recall comparable to RF, with a highly competitive f-measure and AUC. Yet another study detailed [10] addressed the limitations of supervised learning algorithms in sentiment analysis, mainly the need for extensive labeled data. The proposed solution was a hybrid model that combined Term Frequency-Inverse Document Frequency weighting, a lexicon for rule-based sentiment analysis, and the LSTM model. The hybrid model utilizes binary classification algorithms, such as Logistic Regression, knn, Random Forest, SVM, and Naive Bayes.

In their study [11], the authors demonstrated the use of hybrid features obtained by combining machine learning features like Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TF-IDF) with lexicon features such as positive-negative word count and connotation. This combination aimed to improve accuracy and complexity in classifiers like Support Vector Machine (SVM), Naïve Bayes, K-Nearest Neighbors (KNN), and Maximum Entropy. In research [12], sentiment analysis of IMDB comments via star ratings was explored, employing SVM classification. Utilizing SVM, known for its efficacy in pattern recognition, and the TF-IDF technique, the study achieved 79% accuracy, 75% precision, and 87% recall. Notably, the research indicated that SVM outperformed logistic regression, highlighting its superiority in sentiment classification. In another study [13], researchers proposed a model combining various sentiment analysis methods to extract valuable insights and determine the optimal classifier for a specific domain, focusing on accuracy. Given the informal nature of movie reviews and their lack of strong grammatical structures, N-Grams and count vectorizer approaches were incorporated. The process involved tokenization, stemming, feature selection, and classification to transform input strings, extract word roots, select essential features, and classify movies as positive or negative. Further, in [14], diverse methods such as Naïve Bayes, SVM, Stochastic Gradient Descent (SGD), and Decision Tree were employed to achieve optimal accuracy in sentiment analysis. This study involved evaluating movie reviews, yielding 94 positive and 65 negative sentiments. The researchers found that the SVM and SGD classifiers stood out, achieving the highest accuracy at 82% and an F1 score of 81%, thus underscoring their effectiveness. Lastly, in [15], a linguistic methodology was introduced to uncover hidden genres within 600 popular science texts. By utilizing computer programs for linguistic analysis and cluster analysis, four text type clusters were identified, revealing shared linguistic traits and aiding in genre identification based on communicative purposes. An evaluation using a test set demonstrated over 70% accuracy, suggesting the method's relevance in discerning popular science genres with potential pedagogical implications.

5 Proposed Methodology

To effectively analyze the sentiment of reviews and evaluate the accuracy and processing times of various approaches, the study utilizes four advanced machine learning classification models on the IMDB dataset. Figure 3 illustrates the proposed machine learning-based model for sentiment analysis of text, which comprises six primary building blocks and several minor components functioning together as a cohesive unit. The methodology involves a procedure for conducting sentiment analysis on text data. Initially, data is collected and stored in a CSV or Excel file before being imported into the application. Preprocessing is then performed, which includes converting the entire dataset to lowercase, removing HTML tags and URLs, eliminating punctuation, and replacing chat terms and emoticons with their accurate meanings. Subsequently, tokenization is employed to cover sensitive data with recognizable identifying symbols without compromising data security. The data is then divided into training and testing sets. The dataset is subsequently used to evaluate the recall, accuracy, and precision of four machine learning models: Naive Bayes, Logistic Regression, Long Short-Term Memory (LSTM), and Bidirectional Long Short-Term Memory (BiLSTM). The performance of each model is assessed in the final step to determine which model performs the best on the given dataset.

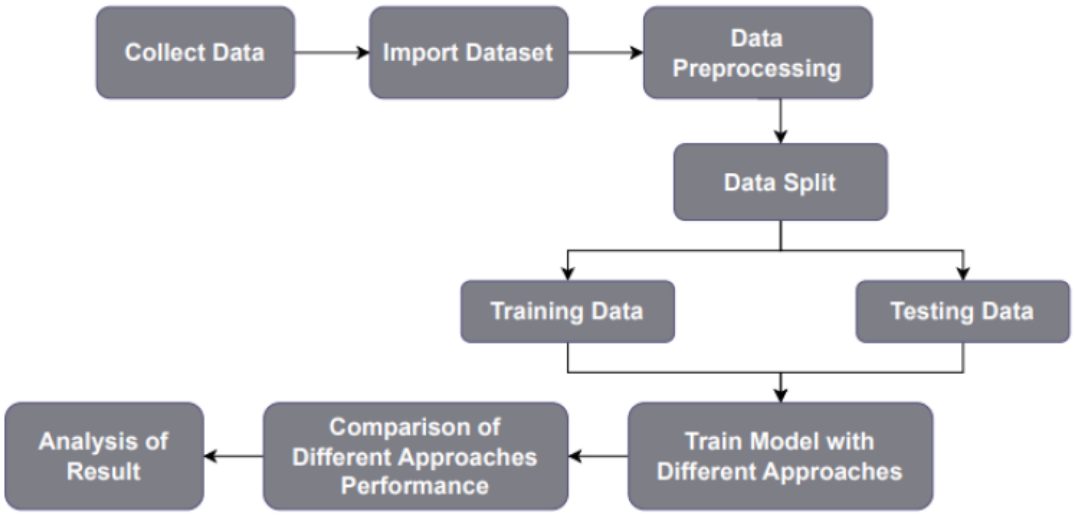

Figure 3: The proposed system flowchart for text sentiment analysis

5.1 Data Collection

The data collection procedure is essential in determining the quantity and quality of data available for sentiment analysis on the IMDb dataset. IMDb, recognized for its extensive user-generated reviews and ratings, serves as a significant online database for movies, TV series, and celebrity content. The review dataset from IMDb, which is accessible across various devices, was procured from Kaggle for this study. Kaggle is a well-known platform for data science and machine learning projects, where data scientists and machine learning experts can collaborate, exchange knowledge, and participate in diverse data science challenges. It provides users with a plethora of tools including datasets, kernels (code notebooks), competitions, and discussion forums, which are instrumental in creating, testing, and refining machine learning models.

5.2 Data Preprocessing

Data preprocessing is a vital step in sentiment analysis as it cleans and prepares the raw data for subsequent analysis. The following steps were undertaken to ready the IMDb dataset for sentiment analysis:

- **Removing HTML tags:** IMDb reviews often contain HTML tags that are extraneous to sentiment analysis. These are removed using regular expressions or the BeautifulSoup library in Python.
- **Removing special characters and punctuation:** The dataset might include special characters and punctuation that do not contribute to sentiment analysis, hence are eliminated using regular expressions or Python's string library.
- **Converting text to lowercase:** Normalizing the text to lowercase ensures that the model does not treat the same word with different cases as distinct entities.
- **Removing stop words:** Common words like "the", "and", and "is", known as stop words, are filtered out to reduce dataset dimensionality and boost model performance. Tools for this include NLTK and spaCy in Python.
- **Tokenization:** This involves segmenting text into tokens, typically single words, which is a crucial step for feature engineering, enabling the creation of a bag of words or n-grams.
- **Stemming and lemmatization:** These techniques reduce inflectional forms and sometimes derivationally related forms of a word to a common base form, thereby enhancing model performance and reducing dataset dimensionality. Lemmatization returns a word to its base or dictionary form, while stemming simply removes the suffixes of words.
- **Spell checking and correction:** To enhance the accuracy of the sentiment analysis model, it is imperative to correct spelling mistakes that might be present in the dataset. Python offers several libraries such as PySpellChecker and TextBlob that provide functionalities for spell-checking and automatic corrections.
- **Handling negation:** In sentiment analysis, correctly interpreting negation is essential since words like "not" or "never" can completely change the sentiment of a sentence. Techniques to address negation include using a separate feature to indicate negation or appending a negation prefix to the affected words.

- **Removing rare and frequent words:** Words that are too rare or too frequent may not contribute significantly to sentiment analysis. Eliminating these words can reduce the dimensionality of the dataset and enhance model performance. Methods for this include setting frequency thresholds, removing stopwords, applying lemmatization or stemming to consolidate variations of the same word, and maintaining curated lists of terms for removal, especially noisy or domain-specific ones. These strategies can significantly refine the dataset, thus improving the data quality for sentiment analysis projects.

Figure 4 displays the most frequent words in both positive and negative reviews after the data preprocessing steps. This analysis helps in understanding the distribution of words in the dataset. Figure 5 illustrates the most commonly used words in the preprocessed dataset. This analysis provides insights into the prevalent terms within the dataset, which can be essential for understanding sentiment trends.

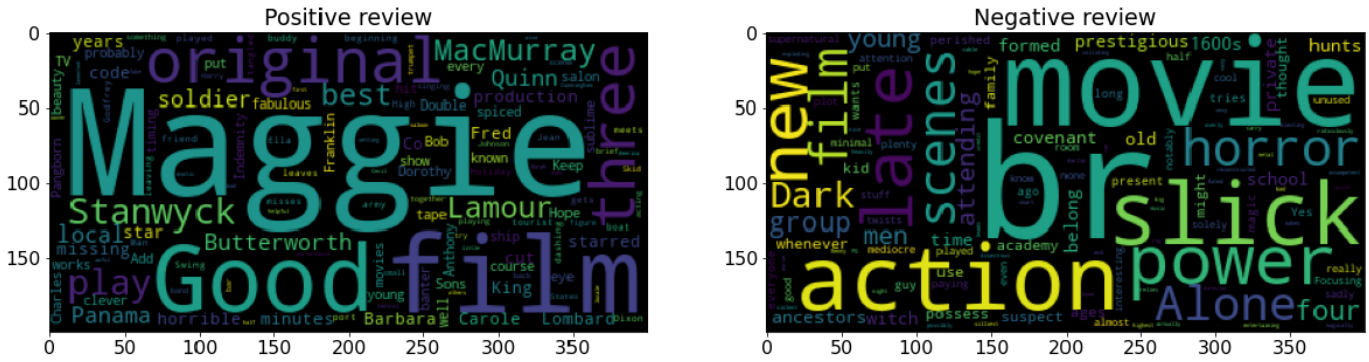

Figure 4: Most Frequent Words in Positive and Negative Reviews

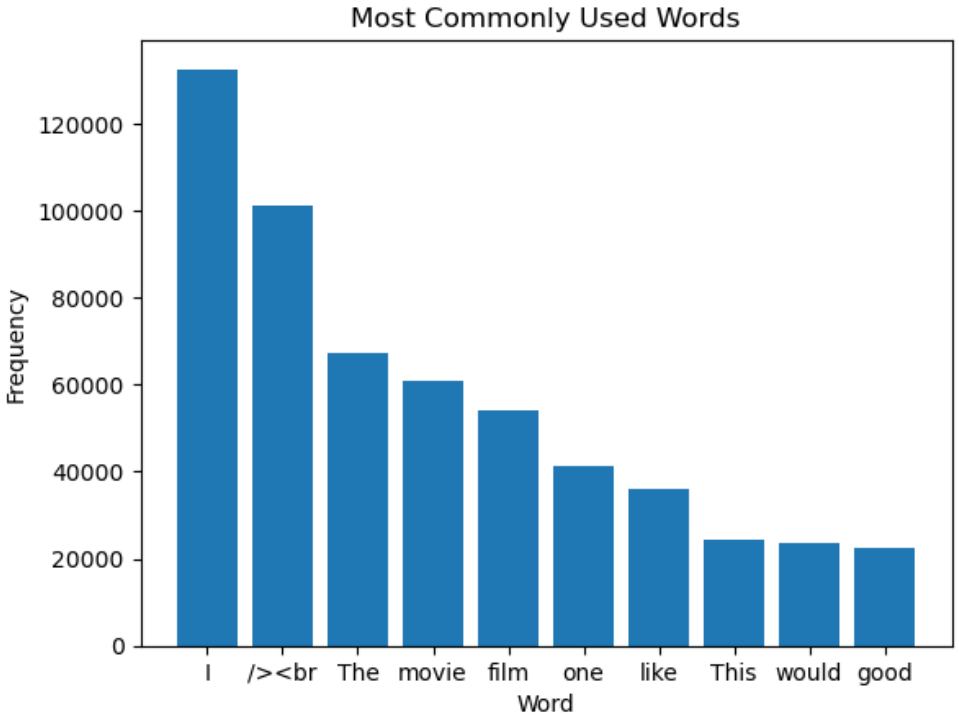

Figure 5: Most Commonly Used Words

6 Feature Engineering

Feature engineering is essential for converting unstructured textual data into a format that machine learning models can understand and utilize efficiently in the fields of NLP, which stands for natural language processing, and sentiment analysis. The *Bag of Words* (also referred to as BoW) and *Term Frequency-Inverse Document Frequency* (also termed as TF-IDF) are basic methods used for feature engineering in my project, and they are covered in this section.

- **Bag of Words (BoW)**: The BoW model treats documents as collections of individual words, focusing on the presence and frequency of terms while ignoring the order. In this study, BoW involves tokenizing text into words to create a vocabulary of unique terms and then representing each document as a vector, with values indicating the frequency of each word. Although this results in a high-dimensional matrix, dimensionality reduction techniques like Principal Component Analysis (PCA) can be applied to make the computational complexity manageable. By converting text to numerical data, BoW facilitates the understanding and analysis of text by machine learning models.
- **TF-IDF**: Term Frequency-Inverse Document Frequency (TF-IDF) is a sophisticated method that evaluates the importance of a word in a document relative to a collection of documents, the corpus. Term Frequency (TF) emphasizes words that occur more frequently in a document, whereas Inverse Document Frequency (IDF) gives more weight to words that are rare across the corpus. In this study, TF-IDF is utilized to extract significant text features, thus enhancing the predictive capabilities of the model. It involves calculating the TF for each word, determining the IDF to weigh word scarcity, and then generating vectors for documents based on these TF-IDF scores. This method is invaluable for identifying and stressing the unique and relevant terms in each document, proving to be an essential tool for sentiment analysis.

6.1 Model Architecture

The architecture of a model encompasses several integral stages such as model selection, training, and evaluation, each of which plays a crucial role in the overall machine learning process.

- **Model Selection**: The initial step involves selecting appropriate machine learning models for sentiment analysis. A variety of models are considered, including neural networks, Support Vector Machines (SVMs), logistic regression, and Naive Bayes. Each of these models is adept at extracting meaningful patterns from data, managing high-dimensional spaces, and recognizing subtle sentiment indicators within text. The deliberate choice of the most fitting model is essential for accurate sentiment classification and is a key component of the methodology.
- **Model Training**: After selecting a model, it is trained using the preprocessed IMDb review dataset. Training involves presenting the model with the data and fine-tuning its parameters to optimize performance for the task at hand.
- **Model Evaluation**: After training, the model undergoes an evaluation phase to determine its effectiveness in sentiment analysis on the IMDb dataset. The dataset is divided into training and testing sets, with the former used for developing the model and the latter for gauging its performance. Evaluation metrics such as accuracy, precision, and recall are used to assess the model's capabilities.

The results from the evaluation guide further refinements of the model. If necessary, additional rounds of training and evaluation are conducted to achieve the targeted performance metrics.

7 Model

7.1 Naive Bayes

The Naive Bayes classifier is based on applying probabilities to text sentiment analysis, utilizing the principles of Bayes' theorem. The theorem posits that the probability of a hypothesis, in this context, a sentiment label, given observable evidence, which is the textual data, is proportional to the probability of the evidence given the hypothesis, multiplied by the prior probability of the hypothesis.

$$P(M|N) = \frac{P(N|M)P(M)}{P(N)} \tag{1}$$

In sentiment analysis, Naive Bayes assesses the probability of a text belonging to certain sentiment categories (e.g., positive or negative) based on the presence of particular words or text features. The approach is termed "Naive" because it assumes that all features (or words) in the text are independent of each other. This simplification expedites the computation process and increases the efficiency of the algorithm, albeit at the expense of possibly ignoring actual interdependencies between features. To prepare Naive Bayes for sentiment analysis, it is trained on a dataset where texts are pre-labeled with sentiment classes. Once trained, the algorithm is capable of predicting sentiments for new, unlabeled texts. Despite its simplicity, Naive Bayes has demonstrated effectiveness in sentiment analysis, particularly when working with large datasets and straightforward feature models. The performance of the Naive Bayes model is visually represented in the confusion matrix shown in Figure 6. This matrix provides insights into the model's precision and recall for sentiment classification.

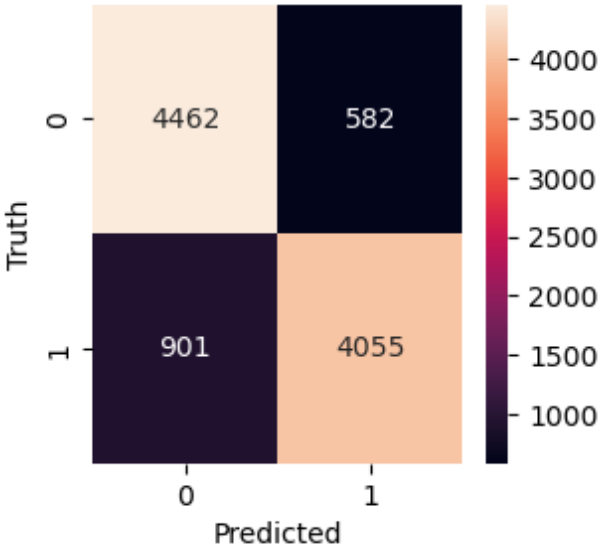

Figure 6: Confusion Matrix for Naive Bayes Model

7.2 Logistic Regression

Logistic regression is a statistical method frequently employed in sentiment analysis of textual data. It uses the logistic function to estimate the probability that a given piece of text belongs to a certain sentiment class, such as positive or negative, based on its constituent words or features. In the context of sentiment analysis, logistic regression applies a logistic function to model the probability of text belonging to a specific sentiment class. This is based on the linear combination of the text's features, with optimal feature weights determined through the process of maximum likelihood estimation on labeled data.

After the training phase, the algorithm can then be used to predict sentiment classes for new, unlabeled texts. Its effectiveness in sentiment analysis is well-established, with applications ranging from binary to multiclass classification tasks. The logistic regression model's performance is further elucidated by the confusion matrix depicted in Figure 7, which outlines the number of true positive, false positive, true negative, and false negative predictions made by the model. This visual representation aids in comprehending the precision and recall of the model. The logistic regression model is described mathematically as:

$$P(X) = \frac{1}{1 + e^{-(\gamma_0 + \gamma_1 x_1 + \gamma_2 x_2 + \ldots + \gamma_n x_n)}} \tag{2}$$

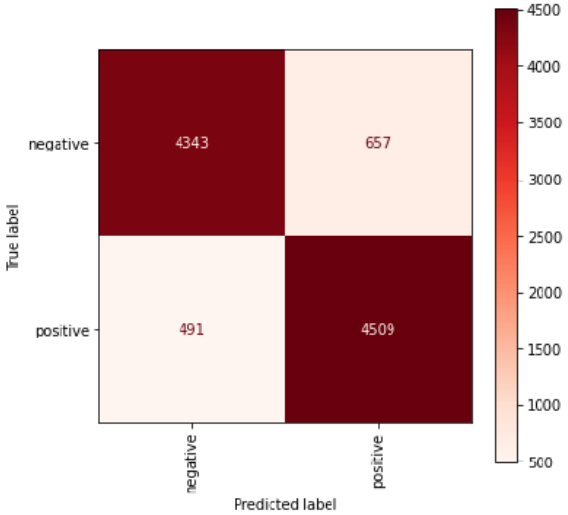

Figure 7: Logistic Regression Confusion Matrix

where $P(X = 1|z)$ is the probability of the binary outcome variable being 1 (for instance, a positive sentiment) given the predictor variables z. The $\gamma_0, \gamma_1, \gamma_2, \ldots, \gamma_n$ are the coefficients or weights corresponding to the predictor variables $x_1, x_2, \ldots, x_n$.

The logistic or sigmoid function is employed to convert the linear combination of predictor variables and their coefficients into a probability score that ranges between 0 and 1. This relationship is also expressed in the form of log-odds or the logit function:

$$\log\left(\frac{P(X=1|z)}{1-P(X=1|z)}\right) = \gamma_0 + \gamma_1 x_1 + \gamma_2 x_2 + \ldots + \gamma_n x_n \tag{3}$$

This equation is utilized to calculate the weights or coefficients of the predictor variables using maximum likelihood estimation from a labeled dataset. The left side of the equation, the log odds, indicates the likelihood that the binary outcome variable is equal to 1.

7.3 Long Short-Term Memory

An RNN model architecture that works well for evaluating sentiment in textual data is LSTM. Its goal is to deal with the problem of vanishing gradients, which may happen in conventional RNNs and make it impossible to maintain long-term dependencies in the input sequence. Specialized memory cells that may selectively preserve or delete data at various time steps are incorporated into LSTM in order to address this issue. The flow of information is controlled by various "gates", including an input gate, a forget gate, and an output gate. Figure 8 illustrates these components of an LSTM unit. Three components make up an LSTM: the input gate, forget gate, and output gate, which control the flow of data

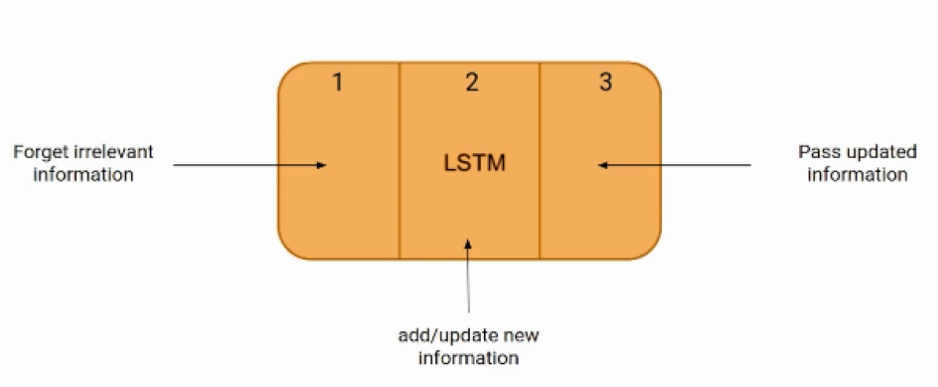

Figure 8: Illustration of LSTM Unit with Input, Forget, and Output Gates

into and out of the memory cell. These components are responsible for managing the flow of data within the LSTM. The extent to which fresh data is permitted to enter the cell is governed by the input gate, while the forget gate is responsible for determining the level at which previous information is eliminated from the cell. The amount of the cell's current state that is used to create predictions is controlled by the output gate. The LSTM's ability to forecast sentiment on new, unlabeled text data post-training can be assessed by its confusion matrix, as depicted in Figure 9. LSTMs can be trained

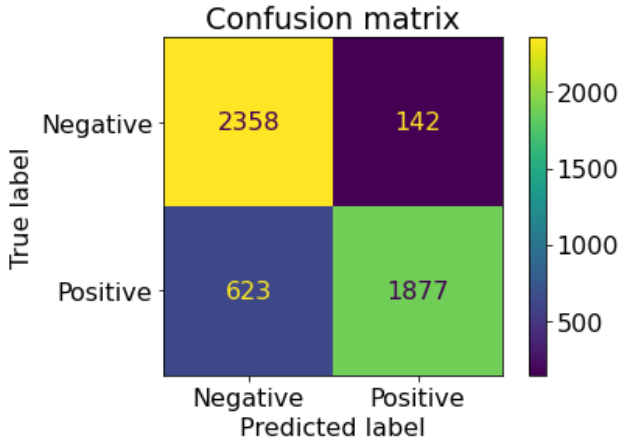

Figure 9: Confusion Matrix for LSTM Model Sentiment Classification

on labelled datasets of text data and are frequently used in sentiment analysis tasks. The network can be used to forecast the sentiment of fresh, unlabeled texts after training. For sentiment analysis tasks involving longer input sequences and more complicated features, LSTMs are particularly effective.

7.4 Bidirectional Long Short-Term Memory

A neural network architecture known as a BiLSTM is useful for analysing the sentiment of textual data. It is an expansion of the LSTM architecture that considers the forward and backward information flow of the input sequence. A BiLSTM is made up of two independent LSTM layers, one for the forward direction of the sequence of inputs and another for the backward direction. The output of each LSTM layer is then combined to generate a single output containing data from both the forward and backward directions. The ability of the BiLSTM architecture to collect contextual data from both the forward and backward directions of the input sequence can increase the accuracy of sentiment analysis. To train BiLSTMs, labelled datasets of textual data that pair each text with a corresponding sentiment class label are used. The network can be used to forecast the sentiment of fresh, unlabeled texts after training. For sentiment analysis tasks requiring longer input sequences and more complicated feature sets, BiLSTMs perform particularly well. The BiLSTM architecture, which accounts for both forward and backward information flow in the input sequence, is represented in Figure 10.

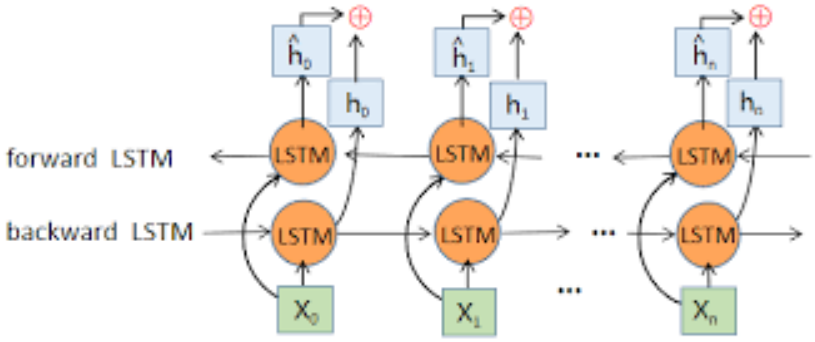

Figure 10: Bidirectional LSTM (BiLSTM) Network Architecture

7.5 Linear Support Vector Machine

For binary classification tasks, LSVM is intended. Finding the hyperplane in the feature space that best separates the two classes is its main objective. When used for sentiment analysis on textual data, LSVM can tell whether a text carries a positive or negative sentiment.

Prior to using LSVM for sentiment analysis, text data must be transformed into numerical features. A bag-of-words model, where each text document is represented as a vector of word frequencies or binary indicators, is one way to accomplish this. After that, we can use a labelled dataset of text documents to train the LSVM. In this dataset, each text document has a binary label that indicates whether it carries positive or negative sentiment. The maximum distance of the margin from the nearest data points in each class determines the decision boundary that the LSVM creates in the training phase, which improves the separation between the two classes. The sentiment of new and unlabeled text data can be predicted using the decision boundary once it has been established. Due to its ease of implementation and high accuracy on many text classification tasks, LSVM has gained popularity as an algorithm for sentiment analysis.

$$y = s^{\top} p + m \tag{4}$$

where y is the predicted sentiment label for a given text document, p is the feature vector representing the text data, s is the weight vector learned during training, and m is the bias term. The LSVM gains knowledge of the ideal s and m values during training that maximise the margin between the two classes. The hyperplane formed by the bias term m and weight vector m then establishes the decision boundary. The work undertaken merely apply the learned decision boundary to the feature vector x to produce the predicted label y in order to predict the sentiment of fresh and unlabeled text data. The text is categorised as having a positive sentiment if y is positive, and as having a negative sentiment if y is negative. The LSVM model works by finding the hyperplane that best separates two classes within the feature space, as visualized in Figure 11.

7.6 Decision Tree

A decision tree is an algorithm used to classify data, especially suited for classification tasks. It achieves this by recursively dividing the feature space into subsets based on the input feature values and assigning a class label to each leaf node in the resulting tree structure. Decision Trees are renowned for their versatility and interpretability in the realm of machine learning, excelling at deciphering complex decision-making processes by breaking them down into a sequence of clear, rule-based choices. The methodology involves the meticulous selection of relevant features from the dataset, which primarily comprises movie reviews.

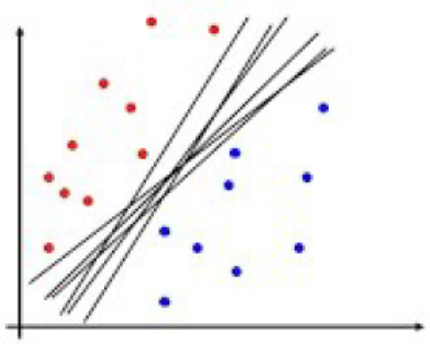

Figure 11: Illustration of LSVM Decision Boundaries

These features include words, phrases, and various linguistic attributes, forming the basis of the analysis. The Decision Tree algorithm orchestrates the construction of a hierarchical tree structure, where each internal node represents a critical decision point based on a specific feature, and each leaf node encapsulates a sentiment label, distinguishing between positive and negative sentiments. When a new movie review is analyzed, it passes through the Decision Tree, starting from the root and moving toward a leaf node. The trajectory is determined by the presence or absence of certain features within the review. At each internal node, a decision is made as to which branch to follow, guided by the content of the review. Ultimately, as the review arrives at a leaf node, the model assigns a sentiment label. If a leaf node is labeled "positive," the review is categorized as having a positive sentiment; conversely, if the path leads to a leaf labeled "negative," the review is deemed to express a negative sentiment. Within the broader context of the study, the Decision Tree Model plays a crucial role in achieving the objective of sentiment analysis. Its transparency and ability to process text data enable it to elucidate the reasoning behind particular sentiment classifications. Moreover, it complements other models used in the research, enhancing the understanding of the effectiveness of decision tree-based approaches in sentiment classification.

8 Results and Discussion

The sentiment analysis project on the IMDb dataset utilized Python for implementation. The dataset was divided into two sets, with 20% reserved for testing and the remaining 80% used for training the models. A diverse array of methodologies was employed to classify the sentiment of movie reviews effectively. These included the application of directional LSTM (Long Short-Term Memory) and Bidirectional LSTM models, which are known for their ability to capture intricate text patterns. Moreover, the predictive capabilities of Linear Support Vector Machine (SVM), Logistic Regression, Naive Bayes, and Decision Tree algorithms were harnessed. This comprehensive approach enabled a rigorous evaluation of the performance of these models in identifying positive and negative sentiments within movie reviews, contributing valuable insights to the field of sentiment analysis.

Table 1: Performance of Classification Model

Name	Type	Performance		
		Accuracy	Precision	Recall
Model 1	Naive Bayes	0.86	0.86	0.87
Model 2	Logistic Regression	0.89	0.90	0.88
Model 3	LSTM	0.91	0.89	0.92
Model 4	BiLSTM	0.91	0.89	0.94
Model 5	Linear SVM	0.89	0.90	0.88
Model 6	Decision Tree	0.70	0.70	0.71

From the data presented in the table, it is concluded that the BiLSTM model surpasses the others in terms of accuracy, recall, and precision. The training process for Model A is visualized in Figure 12, showcasing the trends in accuracy and loss across epochs for both training and validation datasets. Similarly, the training and validation progress for Model B can be observed in Figure 13, which indicates the model's learning curve and generalization capability over successive epochs.

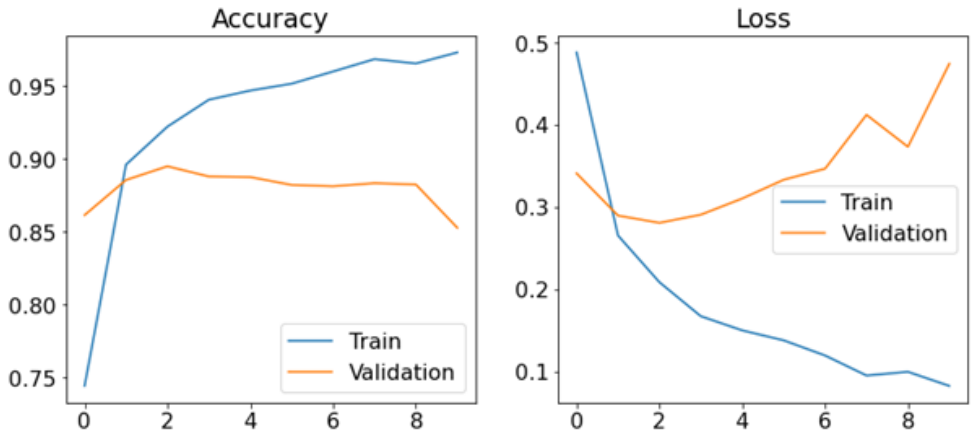

Figure 12: Training and Validation Accuracy and Loss Over Epochs for Model A

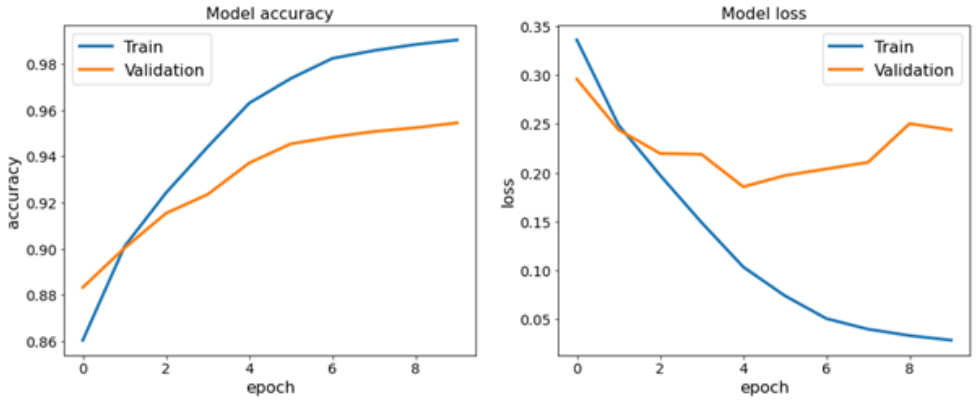

Figure 13: Training and Validation Accuracy and Loss Over Epochs for Model B

9 Conclusion

The study on sentiment analysis of the IMDB review dataset has successfully demonstrated the capabilities of various machine learning models, highlighting the superior performance of the BiLSTM model. The research revealed that deep learning models, particularly BiLSTM, offer significant advantages in processing and analyzing natural language data. These models not only excel in accuracy but also provide insights into the nuances of sentiment analysis in large datasets. Despite the promising results, the study acknowledges limitations in dataset diversity and model generalizability. It suggests that future research could focus on expanding the dataset variety and exploring more complex model architectures to enhance the accuracy and applicability of sentiment analysis techniques. This research contributes to the understanding of machine learning applications in natural language processing, particularly in sentiment analysis. It lays a foundation for future studies to build upon, potentially leading to more robust and versatile sentiment analysis tools for diverse datasets.

Declaration of Competing Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding Declaration

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Author Contribution

Neetu Singla: Methodology, Software, Validation, Investigation, Data curation, Original draft writing, visualization; **Shubham Kumar Singh**: Conceptualization, Formal analysis, Resource management, Review and editing, Supervision.

References

[1] I. Chaturvedi *et al.*, "Distinguishing between facts and opinions for sentiment analysis: Survey and challenges," *Information Fusion*, vol. 44, pp. 65–77, 2018.

[2] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004.

[3] S. M. Qaisar, "Sentiment analysis of imdb movie reviews using long short-term memory," in *2020 2nd International Conference on Computer and Information Sciences (ICCIS)*, IEEE, 2020.

[4] S. Tripathi *et al.*, "Analyzing sentiment using imdb dataset," in *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)*, IEEE, 2020.

[5] A. Hassan and A. Mahmood, "Deep learning approach for sentiment analysis of short texts," in *2017 3rd International Conference on Control, Automation and Robotics (ICCAR)*, IEEE, 2017.

[6] Y. Dong, Y. Fu, L. Wang, Y. Chen, Y. Dong, and J. Li, "A sentiment analysis method of capsule network based on bilstm," *IEEE Access*, vol. 8, pp. 37014–37020, 2020.

[7] Y. Pan, Z. Chen, Y. Suzuki, F. Fukumoto, and H. Nishizaki, "Sentiment analysis using semi-supervised learning with few labeled data," in *2020 International Conference on Cyberworlds (CW)*, IEEE, September 2020.

[8] S. Mathapati *et al.*, "Collaborative deep learning techniques for sentiment analysis on imdb dataset," in *2018 Tenth International Conference on Advanced Computing (ICoAC)*, IEEE, 2018.

[9] M. Yasen and S. Tedmori, "Movies reviews sentiment analysis and classification," in *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, IEEE, 2019.

[10] M. Chiny, M. Chihab, O. Bencharef, and Y. Chihab, "Lstm, vader and tf-idf based hybrid sentiment analysis model," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 7, 2021.

[11] B. Harish, K. Kumar, and H. Darshan, "Sentiment analysis on imdb movie reviews using hybrid feature extraction method," 2019.

[12] N. G. Ramadhan and T. I. Ramadhan, "Analysis sentiment based on imdb aspects from movie reviews using svm," *Sinkron: jurnal dan penelitian teknik informatika*, vol. 7, no. 1, pp. 39–45, 2022.

[13] A. Singh, C. Kulkarni, and N. A. Ayan, "Sentiment analysis of imdb movie reviews," 2022.

[14] P. Gunawan, T. Alhafidh, and B. Wahyudi, "The sentiment analysis of spider-man: No way home film based on imdb reviews," *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, vol. 6, no. 1, pp. 177–182, 2022.

[15] A. Lieungnapar, R. Todd, and W. Trakulkasemsuk, "Genre induction from a linguistic approach," *Indonesian Journal of Applied Linguistics*, vol. 6, no. 2, pp. 319–329, 2017.