Sentiment Analysis of IMDb Reviews using Machine Learning Based Classification

*

1st Junaid Zia
University of Engineering and Technology
Lahore, Pakistan

2nd Abdullah Fasih
University of Engineering and Technology
Lahore, Pakistan

3rd Kumail Haider
University of Engineering and Technology
Lahore, Pakistan

4th Amna Zafar
University of Engineering and Technology
Lahore, Pakistan

***Abstract*—In this study, we basically classify the sentimental representations of Internet Movie Database(IMDb) reviews using machine learning-based classification. Our methodology is to obtain a dataset and then clean and preprocess it and then apply different machine learning algorithms to the dataset. Machine learning allows data analytics to efficiently study large data. Sentiment analysis is one of the most important areas in Natural Language Processing (NLP). Different machine-learning algorithms are used in the training of the model and also evaluated to determine which algorithm performs better. In addition, DNN, logistic regression, and SVM are among the ones that provide greater prediction accuracy, contributing to the novelty of work in the journal field. These results are important for sentiment analysis and developing more accurate text-analytics tools.**

I. Introduction

In this world of large data, machine learning is a trending and emerging field. Data analysis of large data in an efficient manner has become easy due to machine learning. This technique is very helpful in classifying and predicting the content of language [1]. NLP's most prominent area is basically sentiment analysis. Sentiment analysis is usually applied on three levels in machine learning, sentence level, document level, and aspect level [2]. The Sentence level is to analyze the sentiment of each sentence. Document level is to classify the entire document as binary class or multi-class. While the aspect level is more complicated level. It is used to identify the corpus first and then classify each document with respect to the observed aspects of each document.

This model will first remove the stop words and then normalize the words in IMDb reviews to enhance the performance of classification. In the next step, reviews will be transformed into the word matrix, which represents the features for the classification. At last, several machine learning algorithms (logistic regression, SVM, Naive Bayes, random forest, boosting, and deep neural network) are utilized to train and test the word matrix to evaluate the algorithms and decide which performs better on classification. This report is organized as follows. Section two shows the related work on sentiment analysis via machine learning. Segment three explains the basic methodology of this model. Section four addresses the accuracy of the model. At last, chapter five concludes this and shows the future possible research direction.

Identify applicable funding agency here. If none, delete this.

II. Related Work

Tripathy et al. [1] presented a text classification by using Naïve Bayes (NB) and support vector machine (SVM). The results showed that these two algorithms can classify the dataset with high accuracy compared to the other existing research.

Sharma et al. [3] classified the sentiment of the short sentences via convolutional neural network (CNN) with Word2Vec vectorization. The authors cleaned the data with Word2Vec, and implemented CNN to solve the issues of inconsistent noise in language. The results showed that CNN was able to extract better features for short sentences categorization.

Vijayaragavan et al. [4] discussed an optimal SVM-based classification for the sentimental analysis of online product reviews. The paper firstly applied SVM and K-means to cluster the reviews into two groups. Then, the authors employed fuzzy-based soft set theory to determine the possibility of a customer to purchase the product. However, the above research is limited in the exploration of different algorithms to better the classification. The report will, therefore, extend the previous research's effort to more choices of algorithms for a better prediction accuracy.

III. Methodology

The report utilizes a methodology to conduct the analysis of the sentiment analysis of IMDb reviews, as shown in Fig. 1. First, the report illustrates and feeds the data into the data cleaning and preprocessing. Next, the report removes the stop words and some irrelevant words from the original data; then,

the vectorization techniques are applied to transform text into a feature matrix. Last, the report applies six different algorithms to train and test the feature matrix.

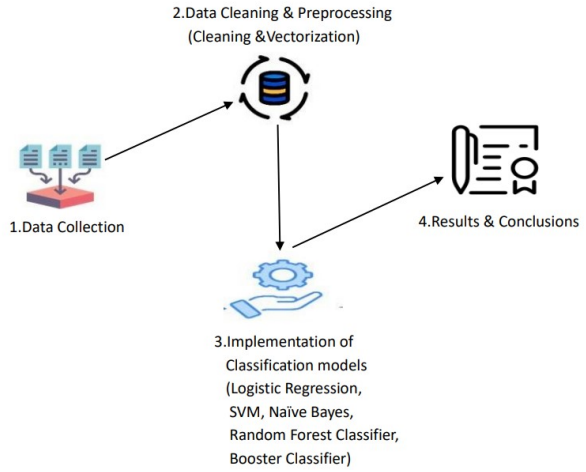

Fig. 1. System Architecture for Sentiment Analysis

A. Dataset

The dataset is retrieved using the method described in [5, 6]. This dataset consists of 50,000 movie reviews taken from IMDb. Half of the data is used for training while the other half is used for testing. Moreover, both the training and testing dataset have 50 percent of positive reviews and 50 percent of negative reviews.

In each of the reviews, users are allowed to rate the movie from 1 to 10. In order to transform this rating scale to a binary label, we define a review as negative if its rating is less than 4 and positive if its rating is more than 7, reviews with ratings between this range are omitted.

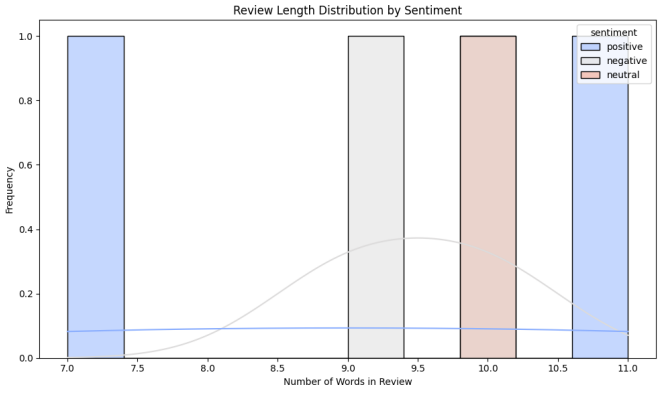

Fig. 2. Distribution of Review Lengths

The figure above illustrates the distribution of review lengths in the IMDb sentiment analysis dataset, where the x-axis represents the number of words in each review and the y-axis indicates the frequency of reviews falling within each length range. The plot differentiates reviews based on sentiment, with distinct colors representing positive, negative, and neutral sentiments. The inclusion of a Kernel Density Estimate (KDE) curve helps visualize the smooth distribution of review lengths, providing insights into how the number of words in reviews varies across different sentiment categories. From the histogram, we can observe the frequency of reviews with shorter and longer word counts, along with the overall trend in review length distribution.

This visualization offers valuable insights into the characteristics of reviews in the dataset. A higher frequency of reviews with shorter lengths may indicate that many users prefer to leave concise feedback, while longer reviews could suggest a more detailed expression of sentiment, particularly for positive or negative opinions. The histogram also helps to identify any patterns related to sentiment and review length, highlighting whether there are notable differences in review length between positive, negative, or neutral sentiments. These findings can be leveraged to further refine models for sentiment analysis by taking into account the varying review lengths within each sentiment category.

B. Data Cleaning and Preprocessing

1) Data Cleaning: In order to facilitate the data interpretation in the later work, raw texts obtained from the previous section are preprocessed. First, elements such as punctuations, line breaks, numbers, and stop words like 'a', 'the', and 'of' are removed since they provide little information about the user's impression towards a movie. Then, all the words are converted to lower cases and normalized to its true root (e.g. played to play) in order to reduce the noise of the vocabularies.

2) Vectorization Techniques: Vectorization is the process of transforming the text data into numeric representations so that the data can be understandable by machine learning algorithms. In this project, we use 4 different methods of vectorization.

a) Binary Vectorization: : One of the simplest vectorization methods is to represent the data as a binary-valued $n \times m$ matrix, where the element $i_{n,m} \in \{0, 1\}$ denotes the existence of the n^{th} vocabulary of the corpus in the m^{th} movie review.

b) Word-count vectorization: : We can also replace the binary values in the matrix with word counts, in which the element of the matrix $i_{n,m} \in \mathbb{R}$ now becomes the frequency that the n^{th} vocabulary of the corpus appears in the m^{th} movie review. This method increases the weight of the more frequently-appeared words in the predictions.

c) n-grams vectorization: In the vectorization method mentioned above, each column of the matrix represents a unique word in our corpus, which means that we are using the appearance of words as our features to predict the rating of a movie review. However, we can also expand the features to a group of consecutive words, called the n-grams of the texts. For example, if we are using n-grams of size 3 to vectorize our data, the columns of the matrix become a sequence of 3 consecutive words appeared in our corpus. This method is useful in cases when phrases provides more information for the prediction than individual word.

d) tf-idf vectorization): The term frequency-inverse document frequency (tf-idf) is a measure of how a given word is concentrated into relatively few documents [11]. This method is based on the idea that the terms which appear more frequently and concentratedly in fewer documents are more representative of the content in the documents.designations.

C. Classification Models

The report implements six classification models to analyze the sentiment of the context, including logistic regression, support vector machine, Naïve Bayes classifier, random forest classifier, boosting classifier, and deep neural networks.

1) Logistic Regression: Logistic regression performs the binary classification by using a sigmoid function as the hypothesis, which is given by:

$$P(y=1|x;\theta) = h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$

The logistic regression model is trained by fitting the parameter θ via maximum likelihood, where the log-likelihood function can be represented.

Then, θ can be updated using stochastic gradient ascent rule

$$\begin{aligned}\theta_j &= \theta_j + \alpha \frac{\partial}{\partial \theta_j}\ell(\theta) \\ &= \theta_j + \alpha\left(y^{(i)} - h_\theta(x^{(i)})\right)x_j^{(i)}\end{aligned}$$

The setting of logistic regression in this report:

1) Inverse of regularization value: [0.01, 0.05, 0.25, 0.5, 1], choose the best performance
2) Penalty = l2
3) Tolerance = 1e-4

2) Support Vector Machine: Support vector machine (SVM) is considered as one of the best algorithms for supervised learning. The main idea of this algorithm is to map the data from a relatively low dimensional space to a relatively high dimensional space so that the higher dimensional data can be separated into two classes by a hyper plane. The hyperplane that separates the data with maximum margin is called the support vector classifier, which can be determined using Kernel Functions in order to avoid expensive computation to transform the data explicitly [10].

1) Inverse of regularization value: [0.01, 0.05, 0.25, 0.5, 1], choose the best performance
2) Penalty = l2
3) Tolerance = 1e-4

3) Naïve Bayes classifier: The Multinomial Naïve Bayes algorithm is useful in the case when the features x_j are discrete-valued due to its simplicity and ease of implementation. This algorithm is based on a strong assumption that x_j's is conditionally independent given y, which is also known as the Naïve Bayes (NB) assumption [10]. The model is parameterized by $\phi_{j|y=1}$, $\phi_{j|y=0}$, and ϕ_y; these parameters can be calculated as:

$$\begin{aligned}\phi_{j|y=1} &= p(x_j=1|y=1) \\ &= \frac{\sum_{i=1}^n 1\{x_j^{(i)}=1 \wedge y^{(i)}=1\}}{\sum_{i=1}^n 1\{y^{(i)}=1\}} \\ \phi_{j|y=0} &= p(x_j=1|y=0) \\ &= \frac{\sum_{i=1}^n 1\{x_j^{(i)}=1 \wedge y^{(i)}=0\}}{\sum_{i=1}^n 1\{y^{(i)}=0\}} \\ \phi_y &= p(y=1) = \frac{\sum_{i=1}^n 1\{y^{(i)}=1\}}{n}\end{aligned}$$

The setting of Naïve Bayes classifier in this report:

1) Laplace smoothing: 1

4) Random forest classifier: Tree classification is very powerful to classify the nonlinear dataset, like NLP. The classification includes bagged tree, random forest, and boosting [8]. Random forest provides an improvement over the bagged trees. Bagged trees consider all the predictors (p predictors) in every split of the tree, whereas random forest limits the selection of the predictors to m predictors. The number of predictors considered in the split in random forest is equal to the square root of the total number of predictors, $m = \sqrt{p}$. In other words, random forest decorrelates the trees through considering less predictors. Unlike highly correlated bagged trees, the variance in random forest is significantly decreased [8].

The setting of random forest in this report:

1) The number of trees: 100
2) Quality criterion: Gini index.
   K is the class number. M is the sample size. The value will take on a small value if the node is pure.

$$G = \sum_{k=1}^{K} P_{mk}\log P_{mk}$$

3) The maximum depth of the tree: None
4) The minimum number of samples required to split an internal node: 2

5) Deep neural networks (DNN): Neural network is recognized as a useful tool for nonlinear statistical modeling [9]. This model is able to incorporate combinations of different neurons (functions) into one giant network [10]. Neural network has evolved to encompass a large class of models and learning algorithms, such as deep neural networks, convolutional neural networks, recurrent neural networks. This report utilizes a five-layer deep neural networks to classify the sentiment of the language. The setting of DNN in this report:

1) The hidden layer: (30, 30, 20, 10, 10)
2) Activation function for the hidden layer: Logistic function
3) L2 penalty (regularization term): 0.0001
4) Early stopping: True
5) The solver for weight optimization: Adam

IV. Results and Discussion

The report evaluates the algorithms' performance by the confusion matrix. A confusion matrix shows the relation between the correct and wrong predictions, as shown in Table. 1.

TABLE I
CONFUSION MATRIX

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

The matrix provides several evaluation parameters, including:

1) Positive precision: The accuracy of the positive prediction.

$$\text{Positive precision} = \frac{TP}{TP + FP}$$

2) Negative precision: The accuracy of the negative prediction.

$$\text{Negative precision} = \frac{TN}{TN + FN}$$

3) Accuracy: The ratio of the correct predictions, which is the average of negative and positive precision.

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

These metrics provide a comprehensive understanding of a model's performance by evaluating its ability to correctly identify both positive and negative predictions. Positive precision highlights the model's success in predicting true positives while minimizing false positives, whereas negative precision reflects its capacity to avoid false negatives when predicting true negatives. Accuracy, on the other hand, offers an overall measure of the model's correctness by considering all prediction outcomes.

Together, these metrics allow for a detailed analysis of the model's strengths and weaknesses, helping to identify areas where the model performs well and aspects that may require further improvement. Such insights are crucial for fine-tuning machine learning models and enhancing their predictive capabilities. Additionally, understanding the trade-offs between precision, recall, and accuracy enables practitioners to optimize models for specific use cases, ensuring the right balance between true positive identification and the avoidance of false alarms or misses. By carefully analyzing these metrics, data scientists can also iteratively refine their models, leading to more robust and reliable predictions and more accuracy in real-world applications.

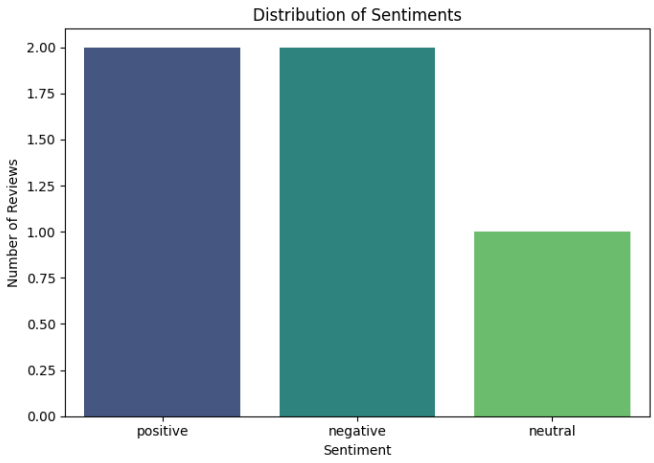

Fig. 3. Visualization of Sentiment Polarity

The figure above visualizes the sentiment distribution of a set of reviews, categorizing them into positive, negative, and neutral sentiments. The x-axis represents the sentiment classes, while the y-axis shows the frequency of reviews falling within each sentiment category. The chart provides a clear picture of the sentiment polarity in the dataset, helping to identify whether the dataset contains a balanced distribution of sentiments or if there is a predominance of one sentiment type. Understanding this distribution is essential for training sentiment analysis models, as an imbalanced dataset may lead to biased predictions and reduced model performance.

The formulas provided above are fundamental to evaluating the performance of a sentiment analysis model. Positive precision focuses on the model's ability to correctly identify positive sentiment while minimizing false positives. A high positive precision score indicates that when the model predicts positive sentiment, it is likely to be correct. Similarly, negative precision reflects the model's ability to correctly predict negative sentiment, avoiding false negatives. Both of these precision metrics are critical when dealing with imbalanced datasets, where one sentiment class may be overrepresented.

Accuracy provides an overall measure of a model's performance by calculating the ratio of correct predictions (both positive and negative) to the total number of predictions made. This metric is a comprehensive evaluation tool, but it may not always be sufficient in cases of class imbalance, where one class (e.g., negative sentiment) dominates the dataset. In such cases, relying solely on accuracy can be misleading, and it is crucial to consider both positive and negative precision alongside accuracy to better understand the model's effectiveness in different sentiment categories. By analyzing these metrics, we can optimize the model for more balanced and reliable predictions, ensuring its applicability in real-world scenarios where sentiment polarity is often nuanced.

TABLE II
COMPARISON OF DIFFERENT ALGORITHMS

Algorithm	Vectorization	Regularization	Positive precision	Negative precision	Accuracy
Logistic	binary, 3 grams	1	0.908	0.893	0.900
	word count, 3 grams	1	0.899	0.894	0.897
	tf-idf, 3 grams	1	0.881	0.872	0.877
SVM	binary, 3 grams	100	0.908	0.894	0.901
	word count, 3 grams	20	0.900	0.895	0.898
	tf-idf, 3 grams	1	0.904	0.896	0.900
Naïve Bayes classifier	binary, 3 grams	-	0.839	0.923	0.881
	word count, 3 grams	-	0.836	0.912	0.874
	tf-idf, 3 grams	-	0.819	0.868	0.879
Random Forest classifier	binary, 3 grams	-	0.859	0.845	0.852
	word count, 3 grams	-	0.860	0.839	0.849
	tf-idf, 3 grams	-	0.864	0.786	0.844
Boosting classifier	binary, 3 grams	-	0.863	0.798	0.831
	word count, 3 grams	-	0.869	0.800	0.834
	tf-idf, 3 grams	-	0.865	0.789	0.825
Deep neural network	binary, 3 grams	0.0001	0.911	0.901	0.906
	word count, 3 grams	0.0001	0.896	0.900	0.898
	tf-idf, 3 grams	0.0001	0.881	0.921	0.901

Vectorization

As the table shows, the binary and 3 grams vectorizations perform best among all three vectorizations for all the algorithms. The reason for this may be binary vectorization can reduce the noise of the parameters. The word-count and tf-idf vectorization count the number of the word in the matrix. In this case, some irrelevant words are counted multiple times, increasing the variance of the model.

Regularization

As for regularization, SVM has many noise parameters since it has a low inverse of the regularization value, whereas logistic regression and DNN fits a model with a relatively low regularization strength. One possible way to further improve SVM is trying to increase the regularization value, minimizing the noise parameters as much as possible. Or the other method may be clean the data more comprehensively, for example, removing subject term should be helpful for the prediction, since the adjective term contributes more to the accurate predictions.

Positive and negative precision

These two estimates provide us a tool to evaluate the accuracy of positive and negative predictions. Logistic regression, SVM, Random Forest, and boosting contribute to a better positive prediction accuracy, whereas Naïve Bayes classifier and DNN contribute to a better negative prediction accuracy, about 92 percent (highest among all the models). In other words, if the scenarios need more accurate negative predictions, the users can implement Naïve Bayes classifier and DNN to predict, whereas if the positive predictions matter more, then the users can implement Logistic regression, SVM, random forest, and boosting.

Accuracy

In term of accuracy, DNN in binary and 3 grams vectorization performs best, 90.6 percent accuracy. Five hidden layers of DNN are able to account for more non-linear relationship of the dataset. It won't be surprised if DNN with more layers can provide better results. In addition, logistic regression and SVM in binary and 3 grams vectorization also perform well (90 percent) in shorter amount of time. Especially surprising is that Naïve Bayes in binary and 3 grams vectorization has 88 percent accuracy. It is the easiest model among all six models, meaning further improving the performance of predictions is possible.

V. CONCLUSION AND FUTURE WORK

The report proposes a methodology to conduct the sentiment analysis of IMDb reviews. The methodology has three major steps, as shown in Fig. 1. As the results show, the binary and 3 grams vectorization performs best among all three vectorizations for all the algorithms. In term of negative accuracy, Naïve Bayes classifier and DNN contribute to a better prediction, whereas the other four models perform better on the positive prediction. In addition, DNN, logistic regression, and SVM provide 90 percent prediction accuracy, which is very promising for the sentiment analysis

Last, future work should implement other vectorizations to better the word matrix. For instance, the researchers can try to remove the subjects in the sentences. In addition, future work can try more complicated models for the analysis. For example, recurrent neural network may be able to provide better performance since it is able to further account for the relationship of the sentences.

REFERENCES

[1] A. Tripathy, A. Agrawal, and S. K. Rath, "Classification of Sentimental Reviews Using Machine Learning Techniques", 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015), Procedia Computer Science, vol. 57, 2015, pp. 821 – 829.

[2] R. Feldman, "Techniques and applications for sentiment analysis," Communications of the ACM, vol. 56, 2013, pp. 82–89.

[3] A. K. Sharma, S. Chaurasia, and D. K. Srivastava, "Sentimental Short Sentences Classification by Using CNN Deep Learning Model with Fine Tuned Word2Vec", International Conference on Computational Intelligence and Data Science (ICCIDS 2019), Procedia Computer Science vol. 167, 2020, 1139–1147.

[4] P. Vijayaragavan, R. Ponnusamy, and M. Aramudhan, “An optimal support vector machine based classification model for sentimental analysis of online product reviews”, Future Generation Computer Systems, vol. 111, 2020, 234–240.

[5] A. Kub, “Sentiment Analysis with Python (Part1)”, https://towardsdatascience.com/sentiment-analysis-with-python-part-1-5ce197074184, accessed on Jun. 5, 2020.

[6] A. Kub, “Sentiment Analysis with Python (Part2)”, https://towardsdatascience.com/sentiment-analysis-with-python-part-2-4f71e7bde59a, accessed on Jun. 5, 2020.

[7] S. Bansal, “A Comprehensive Guide to Understand and Implement Text Classification in Python”, https://www.analyticsvidhya.com/blog/2018/04/a-comprehensive-guide-to-understand-and-implement-text-classification-in-python/, accessed on Jun. 5, 2020.

[8] G. James, D. Witten, T. Hastie, and R. Tibshirani, “An Introduction to Statistical Learning: with Applications in R”, Springer Publishing Company, Incorporated, 2014.

[9] T. Hastie, R. Tibshirani, J. H. Friedman, “The elements of statistical learning: data mining, inference, and prediction. 2nd ed”, New York: Springer, 2009.

[10] M. Tengyu, A. Avati, K. Katanforoosh, and A. Ng, “CS 229 machine learning”, class handout, Stanford University, 2020.

[11] J. Leskovec, A. Rajaraman, and J. D. Ullman. “Mining of Massive Datasets (2nd. ed.)”, Cambridge University Press, USA, Chapter 1, pp. 8-19, 2014.

[12] Statista, “Internet Users Worldwide 2023 - Statista, 2022,” https://www.statista.com/statistics/190263/internet-users-worldwide/, accessed on June 5, 2023.

[13] J. Prentice, “The Influence of Movie Reviews on Consumers,” Honors Theses and Capstones, https://scholars.unh.edu/honors/433.

[14] G. Preeti, V. Uma, and A. Kumar, “Temporal Sentiment Analysis and Causal Rules Extraction from Tweets for Event Prediction,” Procedia Computer Science, vol. 48, pp. 84-91, 2015. DOI Link.

[15] B. Pang, and L. Lee, “Opinion Mining and Sentiment Analysis,” *Foundations and Trends in Information Retrieval*, vol. 2, no. 1–2, pp. 1–135, 2008. DOI Link.

[16] R. Socher et al., “Recursive Deep Models for Semantic Compositionality,” *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, 2013. ACL Link.

[17] Z. Zhou, H. Zhao, and Y. LeCun, “Character-Level Convolutional Networks for Text Classification,” *NIPS’15*, vol. 1, pp. 649-657, 2015. PDF Link.

[18] Y. Mejova, and B. Srinivasan, “Cross-Cultural Sentiment Analysis,” *Proceedings of the 1st Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, pp. 23-30, 2010.

[19] L. Devi et al., “Sentiment Analysis on Movie Reviews,” *Emerging Research in Data Engineering Systems and Communications*, vol. 1054, pp. 321–328, 2020. DOI Link.

[20] K. Kumar, “Deep Neural Networks for Sentiment Analysis,” *IEEE International Conference on Big Data Analytics*, 2020, pp. 622-628. DOI Link.

[21] D. Sarkar, *Text Analytics with Python*, Apress, 2019. Publisher Link.