Effectiveness of AdaBoost and XGBoost Algorithms in Sentiment Analysis of Movie Reviews

I Gusti Ayu Nandia Lestari [1]*, Ni Made Rai Masita Dewi [2], Komang Gita Meiliana [3]**, I Komang Agus Ady Aryanto [4]***
* Department of Information Technology, Institut Teknologi dan Bisnis STIKOM Bali, Bali, Indonesia
** Department of Information Systems, Institut Teknologi dan Bisnis STIKOM Bali, Bali, Indonesia
nandia@stikom-bali.ac.id [1], raimasita@stikom-bali.ac.id [2], gita@stikom-bali.ac.id [3], agus_ady@stikom-bali.ac.id [4]

Article Info

Article history:

Received 2025-01-11
Revised 2025-01-22
Accepted 2025-01-30

Keyword:

AdaBoost,
Classification,
IMDb Reviews Dataset,
Sentiment Analysis,
XGBoost.

ABSTRACT

Currently there are many entertainment platforms that provide various movies, TV shows, games, and other content. These platforms usually offer a variety of features, one of which is reviews. Review data written by viewers plays an important role in influencing public interest in the film. However, the increasing number of reviews makes it difficult to assess the sentiment of the film quickly and accurately. This highlights the need for a system that can analyze reviews based on sentiment, making it easier for viewers to evaluate the film and supporting the entertainment industry in understanding the needs of the audience. Therefore, this study develops a sentiment analysis model to identify whether a review contains positive or negative sentiment using machine learning algorithms. The data used to build the model is obtained from user reviews of a film on the IMDb platform. This dataset is available on Kaggle with 50,000 movie reviews in text format. The characteristics of the data include two columns: *review_text* and *sentiment*. The methods used to create the classification model are AdaBoost and XGBoost. The data preprocessing process includes several stages such as text cleaning, tokenization, *stopword* removal, lemmatization, and vectorization using TF-IDF to convert the review text into numeric form, as well as converting the positive and negative labels into 1 and 0. Based on the results of model training with cross-validation, the accuracy of the XGBoost model is 85% and AdaBoost is 77%. Feature selection showed an improvement in the XGBoost model's accuracy from 85% to 86%, while the AdaBoost model's performance remained stable at 77%. Thus, it can be concluded that the XGBoost model demonstrates better performance than the AdaBoost model in sentiment classification.

I. Introduction

Film is one of the most popular forms of entertainment and has the ability to influence human life. A film can be a medium for conveying stories, emotions, and messages in an engaging and entertaining way. However, the impact of films is not only as a source of entertainment but also as a means for education, cultural delivery, and even social reflection. Recently, the film industry has grown rapidly with an increasingly diverse and global production, much like K-pop, which is widely enjoyed by people of all ages, as well as many other types of films that offer a broad range of entertainment choices for society [1].

In today's digital era, many film platforms have become places for people to discover and enjoy various movies. One of the largest film platforms in the world, offering a wide range of movies, TV shows, video games, and other content, is IMDb (Internet Movie Database). This platform, founded in 1990, is owned by Amazon [2]. IMDb features various tools that help users find information about films. One of these features allows users to provide reviews and ratings for the movies they watch. These review data serve as an important factor in shaping the reputation and popularity of the film, as viewer reviews reflect their experiences and emotional reactions to the film. The influence of user reviews can even affect the interest of potential viewers and the marketing strategies of the film. The number of reviews published on IMDb daily continues to grow, and manually analyzing these reviews presents a challenge. This has prompted the need for an automated system model capable of analyzing and classifying reviews based on sentiment, which can assist viewers in making decisions and support the entertainment industry in understanding audience preferences. Therefore, this study develops a sentiment analysis model to address this issue by identifying whether a review contains positive or negative sentiment using machine learning algorithms. The main objective of this research is to gain deeper insights into how a film is received by the public and to understand the general patterns of audience reactions.

This study uses machine learning methods to train models for determining the sentiment value of a review. The methods used to create the classification model are AdaBoost and XGBoost, both of which are boosting algorithms. The choice of these two methods is due to their effective approach in handling user reviews. AdaBoost is one of the first boosting algorithms, working by combining several simple models to form a stronger model [3]. On the other hand, XGBoost is a more modern development or version, and it can be used for classification tasks with its ability to handle complex data [4]. Based on this, the focus of this research is on sentiment analysis of IMDb user reviews using AdaBoost and XGBoost methods to compare the performance of the two algorithms in classifying positive or negative reviews, to support better decision-making in the entertainment industry.

In this study, the data used is obtained from user reviews of a film on the IMDb platform. The data consists of text written by users expressing their opinions about the film [5][6]. Once this data is collected, it undergoes data cleaning to ensure that the data used contains relevant information. The next step is to ensure that each review has a label based on the sentiment contained in the data. The data is then split into two parts for training and testing, with the training process using AdaBoost and XGBoost algorithms. After training the models, the testing data will be used to evaluate the accuracy and effectiveness of both algorithms in classifying sentiment in the reviews. The results of this evaluation will be used to compare the performance of the two methods in the context of sentiment analysis.

It is expected that this study can provide insights into the entertainment industry on how audience opinions in reviews can be a factor in a film's success. In addition, the results obtained can contribute to the development of better sentiment analysis techniques and provide a foundation for future research.

TABLE I
PREVIOUS STUDIES IN SENTIMENT ANALYSIS

Item	Reference	Method
1	Chengying et al. (2022) % [7]	LightGBM
2	Adam et al. (2023) [8]	BERT, LSTM, GRU
3	Selen et al. (2023) [9]	Support Vector Machine (SVM)
4	Haifa et al. (2023) [10]	Logistic Regression (LR), Multinomial Naive Bayes (MNB), Stochastic Gradient Descent (SGD)
5	Our works	AdaBoost dan XGBoost

II. METHOD

A. System Overview

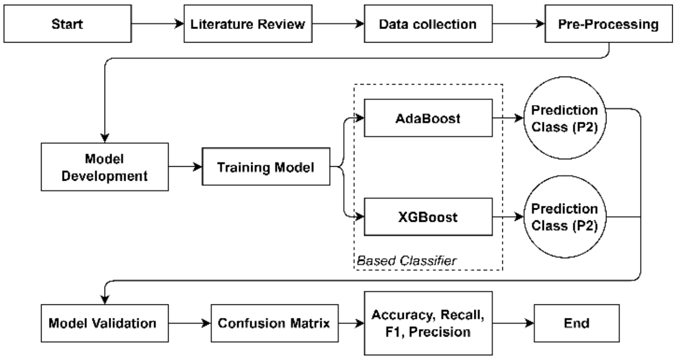


Figure 1. System overview

Figure 1 illustrates the general flow of the research in the sentiment analysis system for IMDb reviews using the AdaBoost and XGBoost algorithms for classification. The research process consists of several stages. The first stage involves conducting a literature review to understand the concepts, methods, and approaches suitable for sentiment analysis using machine learning algorithms. The next stage involves collecting data on movie reviews written by viewers on the IMDb platform. After that, a pre-processing step is performed to prepare the data so that it contains relevant information for use in the model training or development process.

The next stage involves model development or the training process using two algorithms, AdaBoost and XGBoost [11][12]. Both machine learning algorithms, AdaBoost and XGBoost, are used as classifiers to determine whether the review belongs to the positive or negative class. The results obtained from the training process are then validated to measure their performance using an evaluation matrix, the confusion matrix. This confusion matrix shows the number of correct and incorrect predictions made by the model. The model performance is measured using several metrics, namely Accuracy, Recall, F1, and Precision [13].

B. Dataset Description

This study uses audience review data from the IMDb platform. The dataset is available on Kaggle and contains

50,000 movie reviews in text format. Of these, 25,000 reviews are positive and 25,000 reviews are negative, as shown in Figure 2, which visualizes the data distribution. The data consists of two columns: review text and sentiment, as shown in Table 2. The review text column contains movie reviews written by viewers with various opinions, criticisms, or praises for the film. The reviews in the review text column are the primary data used for sentiment analysis. Meanwhile, the sentiment column contains review labels, either positive or negative, which are used as targets in sentiment classification to train the model [14].

Figure 3 shows the text length distribution in the dataset, with an average text length of 231.16 words per review. Meanwhile, Figure 4 illustrates the word frequency in the dataset. This graph highlights the most frequently occurring words in the reviews discussed by users.

TABLE 2
DATASET ATTRIBUTES

Attribute	Description
Review Text	Movie review text written by users
Sentiment	Sentiment of the user review

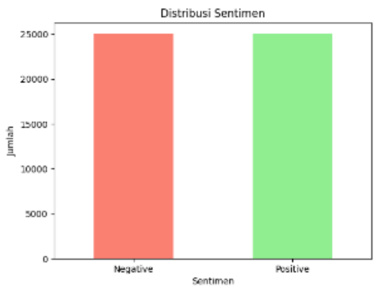


Figure 2. Distribution of Data Count

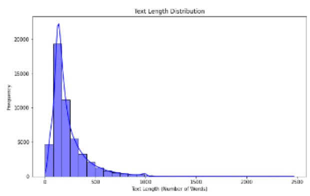


Figure 3. Text Length Distribution

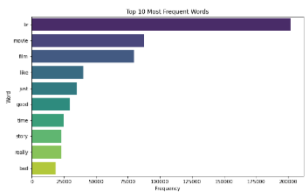


Figure 4. Word Frequency

C. *Preprocessing Data*

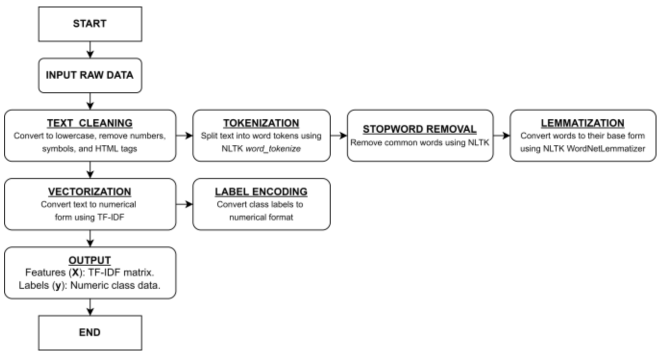


Figure 5. Preprocessing flow

Figure 5 shows the flow for performing the preprocessing process. This stage describes the preprocessing steps carried out in sentiment analysis of IMDb reviews. The preprocessing steps include several processes such as text cleaning, tokenization, *stopword* removal, lemmatization, vectorization, and label encoding [15][16]. The first step is text cleaning, where the raw review data is cleaned of irrelevant elements such as punctuation, numbers, or special characters. Additionally, the text is converted to lowercase to avoid unnecessary variations in the text. After the cleaning process, the review text becomes a simpler format, as shown in Table 3. Table 3 contains two columns: the *review* column, which holds the raw data, and the *cleaned_review* column, which contains the result of the cleaning process.

Next, tokenization is performed to break the reviews into individual words or tokens [15]. The result of this process is shown in Table 4, which contains two columns: the *cleaned*_review column and the *tokenized_review* column, which are the results of the tokenization process. The next step is *stopword* removal, which involves eliminating words that do not provide meaningful information in sentiment analysis. This process is done by filtering out words that are in the *stopwords* list, leaving only the relevant words. After the *stopword* removal process, the word frequency is calculated before and after *stopword* removal. The results are shown in Figure 6.a for the top 10 most frequent words before *stopword* removal, and Figure 6.b for the top 10 most frequent words after *stopword* removal. This word frequency analysis highlights the most influential words in the film reviews after elements that do not contribute to the analysis are removed.

In the lemmatization process, words that have been filtered for *stopwords* are transformed into their base forms to avoid meaning duplication. The lemmatization results in consistent word forms, making them more focused on their true meaning. Figure 7 displays the word distribution visualization after lemmatization, showing the top 10 most frequent words. Afterward, to prepare the text for numerical representation, the TF-IDF (Term Frequency-Inverse Document Frequency) method is used, which generates vectors by considering the frequency of each word in the text. In this analysis, the top features are identified as the most significant words, highlighting their importance in the film reviews, as shown in Figure 8. Figure 9 also displays a Word Cloud visualization of the words based on all TF-IDF features, combining the features and their weights for the visualization. Next, the generated TF-IDF matrix (5000 x 5000) shows the weight of each word (feature) in each review, which is used to train the sentiment analysis model, as shown in Table 5. Finally, the sentiment labels, "positive" and "negative," are converted into numerical values of 1 and 0 so they can be processed by machine learning algorithms.

TABLE 3
THE RESULT OF THE TEXT CLEANING PROCESS

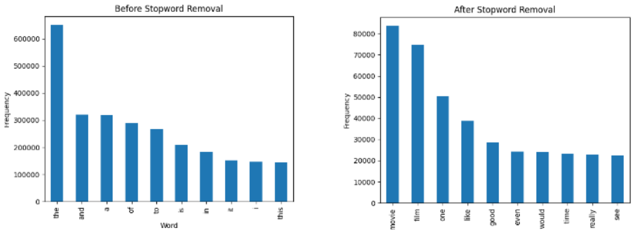
Item	Review	Cleaned_Review
0	One of the other reviewers has mentioned that ...	one of the other reviewers has mentioned that ...
1	A wonderful little production. The...	a wonderful little production the filming tech...
2	I thought this was a wonderful way to spend ti...	i thought this was a wonderful way to spend ti...
3	Basically there's a family where a little boy ...	basically theres a family where a little boy j...
4	Petter Mattei's "Love in the Time of Money" is...	petter matteis love in the time of money is a ...
...	...	...
49999	No one expects the Star Trek movies to be high...	no one expects the star trek movies to be high...

TABLE 4
THE RESULT OF THE TOKENIZATION PROCESS

Item	Cleaned_Review	Tokenized_Review
0	one of the other reviewers has mentioned that ...	[one, of, the, other, reviewers, has, mentione...
1	a wonderful little production the filming tech...	[a, wonderful, little, production, the, filmin...
2	i thought this was a wonderful way to spend ti...	[i, thought, this, was, a, wonderful, way, to,...
3	basically theres a family where a little boy j...	[basically, theres, a, family, where, a, littl...
4	petter matteis love in the time of money is a ...	[petter, matteis, love, in, the, time, of, mon...
...	...	...
49999	no one expects the star trek movies to be high...	[no, one, expects, the, star, trek, movies, to...

TABLE 5
WORD WEIGHTS FROM TF-IDF CALCULATION

Item	aaron	...	movie	film	one	like	good	character	see	...	zombie
1	0	...	0.000	0.000	0.022	0.000	0.000	0.000	0.000	...	0.000
2	0	...	0.000	0.000	0.039	0.000	0.000	0.000	0.052	...	0.000
3	0	...	0.000	0.000	0.039	0.000	0.000	0.052	0.052	...	0.000
4	0	...	0.094	0.067	0.000	0.039	0.000	0.000	0.046	...	0.114
5	0	...	0.030	0.065	0.198	0.000	0.083	0.089	0.044	...	0.000
6	0	...	0.095	0.000	0.052	0.058	0.000	0.000	0.000	...	0.000
7	0	...	0.000	0.000	0.000	0.104	0.000	0.000	0.062	...	0.000
8	0	...	0.000	0.000	0.039	0.000	0.000	0.000	0.053	...	0.000
9	0	...	0.000	0.312	0.105	0.000	0.000	0.000	0.071	...	0.000
10	0	...	0.193	0.000	0.000	0.236	0.000	0.000	0.000	...	0.000
11	0	...	0.055	0.178	0.060	0.000	0.000	0.081	0.000	...	0.000
5000	...	...	...	...	...	...	...	...	...	...	...



(a) (b)

Figure 6. Result of Stopword Removal: (a) word frequency before, (b) word frequency after

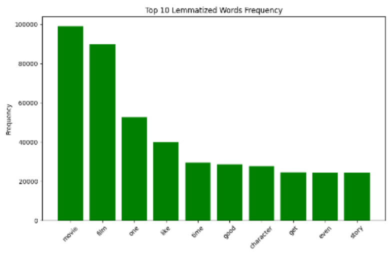


Figure 7. Result of Lemmatization based on word frequency

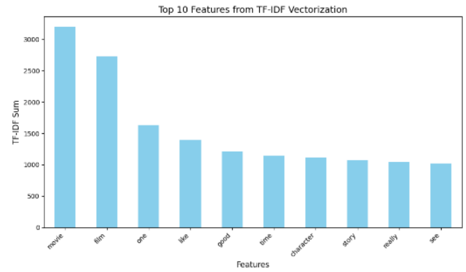


Figure 8. Top features from TF-IDF Vectorization

Figure 9. Word Cloud visualization of all TF-IDF features

D. AdaBoost

In this study, the AdaBoost algorithm which is an ensemble learning method is used to handle data classification problems. The model training process with this algorithm is carried out by assigning weights at each iteration so that the model performance improves over the course of training. After all models are trained, AdaBoost combines their results by assigning weights based on the accuracy of each model, where the model with higher accuracy will contribute more to the final decision. The equation used in this algorithm can be seen in Equation 1, where $H(x)$ is the final prediction value for input x. The variable t represents the number of iterations used, and α_t is the weight given to the t iteration. Then, $h_t(x)$ is the prediction value at the t iteration [17].

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t \, \mathrm{h}_t(x)\right) \quad (1)$$

E. XGBoost

The XGBoost algorithm is also used in this study, it is an implementation of the gradient boosting algorithm that focuses on optimizing speed and memory efficiency. There are four main features of this algorithm: L1 (Lasso) and L2 (Ridge) regularization, which are used to reduce overfitting. Handling Missing Values to address missing data. Parallel Processing to support parallel processing thus improving training speed, and Tree Pruning to reduce the size of the model, thereby enhancing model performance [18][19]. The equation for the XGBoost algorithm can be seen in Equation 2.

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \quad (2)$$

F. Model Evaluation

TABLE 6
EVALUATION METRICS

Metrics	Equation
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$
F-Measure	$F_\beta = \frac{1}{\beta \times \frac{1}{precision} + (1-\beta) \times \frac{1}{recall}}$

The model evaluation process utilizes the confusion matrix technique, where the values in the confusion matrix are used to calculate accuracy, precision, recall, and F1-measure [20]. The calculations for these metrics are shown in Table 6. The confusion matrix contains four distinct combinations of predicted and actual values: True Positive (TP) refers to correctly predicted positive data, True Negative (TN) refers to correctly predicted negative data, False Positive (FP) refers to negative data incorrectly predicted as positive, and False Negative (FN) refers to positive data incorrectly predicted as negative [21][22].

III. RESULT AND DISCUSSION

This section discusses the results obtained from the analysis and preprocessing of data conducted to build the sentiment classification model. The model was developed using two methods, AdaBoost and XGBoost, with the performance of each method evaluated using various metrics. To ensure robust evaluation, 5-fold stratified cross-validation was employed. This approach divides the data into five subsets while maintaining the same class distribution in each fold, enabling a balanced evaluation and minimizing bias in performance metrics.

Tables 7 and 8 present the confusion matrices for the AdaBoost and XGBoost methods. Using these confusion matrices, evaluation metric parameters such as accuracy, precision, recall, and F1-score were calculated for each method [23]. Based on the results, the accuracy of the AdaBoost method reached 77%, while the XGBoost method achieved a higher accuracy of 85%, as shown in Table 9. Subsequently, a statistical test using McNemar was conducted to determine whether the performance difference between the two models was statistically significant [24]. Based on the test results, a p-value of 3.54×10^{-15} was obtained, which is close to zero ($\alpha \leq 0.05$). This indicates that the XGBoost model is significantly better at sentiment classification compared to the AdaBoost model.

Next, the precision calculation results for the AdaBoost method show a value of 75% for the positive class and 80% for the negative class. Meanwhile, the XGBoost method has higher precision, with 84% for the positive class and 87% for the negative class, as shown in Table 10. In terms of the recall metric, the AdaBoost method yields 82% for the positive class and 72% for the negative class. The XGBoost method demonstrates better performance with a recall of 87% for the positive class and 83% for the negative class, as displayed in Table 11. Finally, for the F1-score metric, the AdaBoost method results in 78% for the positive class and 76% for the negative class. In the XGBoost method produces an F1-score of 86% for the positive class and 85% for the negative class, as seen in Table 12. Therefore, the XGBoost method proves to be more effective for sentiment classification on this dataset.

TABLE 7
CONFUSION MATRIX FOR ADABOOST

	Positif (1)	Negatif (0)
Positif (1)	20394	4606
Negatif (0)	6934	18066

TABLE 8
CONFUSION MATRIX FOR XGBOOST

	Positif (1)	Negatif (0)
Positif (1)	21873	3127
Negatif (0)	20775	4425

TABLE 9
MODEL ACCURACY PERFORMANCE

Item	Method	Accuracy (%)
1	AdaBoost	77
2	XGBoost	85

TABLE 10
MODEL PRECISION PERFORMANCE

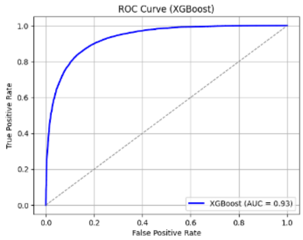
Item	Method	Precision Positive (%)	Precision Negatif (%)
1	AdaBoost	75	80
2	XGBoost	84	87

TABLE 11
MODEL RECALL PERFORMANCE

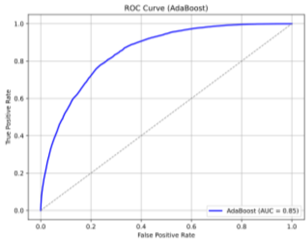
Item	Method	Recall Positive (%)	Recall Negatif (%)
1	AdaBoost	82	72
2	XGBoost	87	83

TABLE 12
MODEL F1-SCORE PERFORMANCE

Item	Method	F1-Score Positive (%)	F1-Score Negatif (%)
1	AdaBoost	78	76
2	XGBoost	86	85



(a)



(b)

Figure 10. ROC and AUC: (a) XGBoost Model, (b) AdaBoost Model.

In the model evaluation using the AUC-ROC metric, XGBoost achieved a score of 0.93, indicating the model's strong ability to distinguish between positive and negative sentiments. In comparison, the AdaBoost model obtained an AUC-ROC score of 0.85. Figure 10 illustrates the ROC curves for the XGBoost and AdaBoost models, with respective AUC scores of 0.93 (Figure 10.a) and 0.85 (Figure 10.b).

To improve the model's accuracy, a feature selection process was performed on the dataset. The feature selection process was done using the XGBoost model, which calculates the Feature Importance value for each feature based on its contribution to the accuracy results. Features with a Feature Importance value greater than the threshold of 0.00025 were retained. From the initial 5,000 features generated by TF-IDF, this process left 811 features that were considered relevant, thus reducing the data's dimensionality. The results of the feature selection are shown in Figure 11, which displays the top 20 features with the highest Feature Importance values. After feature selection, the accuracy of the XGBoost model increased from 85% to 86%. However, the AdaBoost method did not show any accuracy improvement, maintaining a steady value of 77%. The comparison of accuracy between the models with and without feature selection is shown in Table 13 and Figure 12.

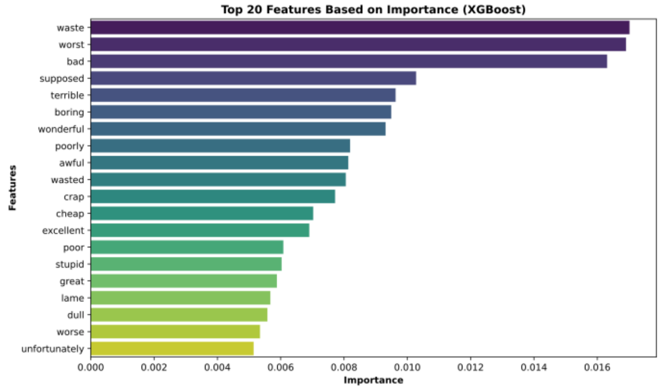


Figure 11. Feature Importance Values Based on Feature Selection with XGBoost

TABLE 13
ACCURACY COMPARISON

Item	Method	Accuracy Without Feature Selection (%)	Accuracy With Feature Selection (%)
1	AdaBoost	77	77
2	XGBoost	85	86

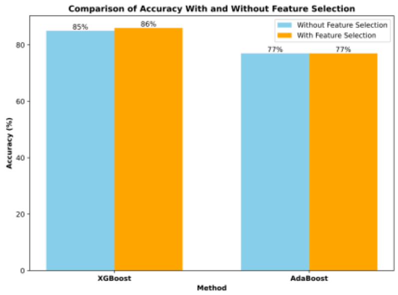


Figure 12. Accuracy Comparison of Models

IV. CONCLUSION

This study aims to develop a sentiment analysis model for movie reviews. The data used in this research was obtained from Kaggle and consists of 50,000 movie reviews from the IMDb platform. The dataset is divided into two classes: positive and negative, with 25,000 reviews in each class. The research process begins with data preprocessing, which includes text cleaning, tokenization, stopword removal, lemmatization, and vectorization using TF-IDF to convert the text into numerical format. Afterward, the positive and negative labels are converted into values 1 and 0.

Based on the model training results with cross-validation, the XGBoost model demonstrated better performance compared to the AdaBoost model. The XGBoost model achieved an accuracy of 85%, while AdaBoost achieved an accuracy of 77%. Additionally, other evaluation metrics such as precision, recall, and F1-score also showed that XGBoost produced better results than AdaBoost. To further improve the model's accuracy, a feature selection process was applied to the dataset, reducing the number of features from 5,000 to 811. This feature selection successfully increased the accuracy of the XGBoost model from 85% to 86%, while the performance of the AdaBoost model remained stable at 77%.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to the Institute of Technology and Business STIKOM Bali for their support in this research, as well as to all our colleagues who have greatly assisted throughout this process.

REFERENCES

[1] K. Lu and J. Wu, "Sentiment analysis of film review texts based on sentiment dictionary and SVM," *ACM Int. Conf. Proceeding Ser.*, vol. Part F1481, pp. 73–77, 2019, doi: 10.1145/3319921.3319966.

[2] K. K. Singh, J. Makhania, and M. Mahapatra, "Impact of ratings of content on OTT platforms and prediction of its success rate," *Multimed. Tools Appl.*, vol. 83, no. 2, pp. 4791–4808, 2024, doi: 10.1007/s11042-023-15887-9.

[3] S. Wu and H. Nagahashi, "Parameterized AdaBoost: Introducing a Parameter to Speed Up the Training of Real AdaBoost," *IEEE Signal Process. Lett.*, vol. 21, no. 6, pp. 687–691, 2014, doi: 10.1109/LSP.2014.2313570.

[4] M. Chen, H. Xu, Y. Wu, and J. Wu, "Sentiment Analysis of Hotel Reviews based on BERT and XGBoost," in *2024 3rd International Conference on Computer Technologies (ICCTech)*, 2024, pp. 11–15. doi: 10.1109/ICCTech61708.2024.00011.

[5] A. Ghosh, "Sentiment Analysis of IMDb Movie Reviews : A comparative study on Performance of Hyperparameter-tuned Classification Algorithms," in *2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2022, vol. 1, pp. 289–294. doi: 10.1109/ICACCS54159.2022.9784961.

[6] S. Tripathi, R. Mehrotra, V. Bansal, and S. Upadhyay, "Analyzing Sentiment using IMDb Dataset," in *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)*, 2020, pp. 30–33. doi: 10.1109/CICN49253.2020.9242570.

[7] C. Zhu, J. Yao, G. Zhao, S. Wang, S. Liu, and Z. Liu, "Negative review detection model based on LightGBM," in *2022 4th International Conference on Intelligent Information Processing (IIP)*, 2022, pp. 171–174. doi: 10.1109/IIP57348.2022.00042.

[8] Á. Kovács and T. Tajti, "Enhancing Sentiment Analysis Accuracy on IMDB Reviews Through Ensemble Machine Learning Techniques," in *2023 IEEE 21st Jubilee International Symposium on Intelligent Systems and Informatics (SISY)*, 2023, pp. 289–294. doi: 10.1109/SISY60376.2023.10417873.

[9] S. N. Başa and M. S. Basarslan, "Sentiment Analysis Using Machine Learning Techniques on IMDB Dataset," in *2023 7th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 2023, pp. 1–5. doi: 10.1109/ISMSIT58785.2023.10304923.

[10] H. B. Habib, M. K. Chowdhury, M. T. Islam, M. S. Mahmud, and A. Sattar, "Sentiment Classification for IMDB Movie Reviews in Benchmark Dataset Using LR, MNB and SGD," in *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2023, pp. 1–6. doi: 10.1109/ICCCNT56998.2023.10307321.

[11] S. Li, G. Yin, and T. Yang, "Research on product iterative requirement analysis method based on internet review data and XGBoost," in *2020 IEEE International Conference on Information Technology,Big Data and Artificial Intelligence (ICIBA)*, 2020, vol. 1, pp. 179–184. doi: 10.1109/ICIBA50161.2020.9277005.

[12] X. Feng, "Research of Sentiment Analysis Based on Adaboost Algorithm," in *2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*, 2019, pp. 279–282. doi: 10.1109/MLBDBI48998.2019.00062.

[13] M. M. Ahsan, M. A. P. Mahmud, P. K. Saha, K. D. Gupta, and Z. Siddique, "Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance," *Technologies*, vol. 9, no. 3, 2021, doi: 10.3390/technologies9030052.

[14] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning Word Vectors for Sentiment Analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Jun. 2011, pp. 142–150. [Online]. Available: http://www.aclweb.org/anthology/P11-1015

[15] M. Sergii V. and N. Oleksandr V., "Data preprocessing and tokenization techniquesfortechnical Ukrainian texts," *Appl. Asp. Inf. Technol.*, vol. 6, no. 3, pp. 318–326, 2023, doi: 10.15276/aait.06.2023.22.

[16] C. P. Chai, "Comparison of text preprocessing methods," *Nat. Lang. Eng.*, vol. 29, no. 3, pp. 509–553, 2023, doi: 10.1017/S1351324922000213.

[17] F. Mekhalfa and N. Nacereddine, "Gentle Adaboost algorithm for weld defect classification," in *2017 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, 2017, pp. 301–306. doi: 10.23919/SPA.2017.8166883.

[18] L. Sun, "Application and Improvement of Xgboost Algorithm Based on Multiple Parameter Optimization Strategy," in *2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, 2020, pp. 1822–1825. doi: 10.1109/ICMCCE51767.2020.00400.

[19] T. Kurniawan, L. Hermawanti, and A. N. Safriandono, "Interpretable Machine Learning with SHAP and XGBoost for Lung Cancer Prediction Insights," vol. 8, no. 2, pp. 296–303, 2024, doi: https://doi.org/10.30871/jaic.v8i2.8395.

[20] I Gusti Ayu Nandia Lestari and I Komang Agus Ady Aryanto, "Peningkatan Akurasi Klasifikasi Kualitas Udara melalui Oversampling dengan Metode Support Vector Machine dan Random Forest," *J. Sist. dan Inform.*, vol. 18, no. 1, pp. 1–9, 2023, doi: 10.30864/jsi.v18i1.596.

[21] I. G. Ayu Nandia Lestari, D. G. Hendra Divayana, and K. Y. Ernada Aryanto, "A Concentration Selection In Study Programs Using SMOTE Techniques With Ensemble Learning Algorithms," in *2023 5th International Conference on Cybernetics and Intelligent System (ICORIS)*, 2023, pp. 1–6. doi: 10.1109/ICORIS60118.2023.10352192.

[22] T. S. Nabila *et al.*, "Classification of Brain Tumors by Using a Hybrid CNN-SVM Model," vol. 8, no. 2, pp. 241–247, 2024, doi: https://doi.org/10.30871/jaic.v8i2.8277.

[23] A. Anggrawan, H. Hairani, and C. Satria, "Improving SVM Classification Performance on Unbalanced Student Graduation Time Data Using SMOTE," *Int. J. Inf. Educ. Technol.*, vol. 13, no. 2, pp. 289–295, 2023, doi: 10.18178/ijiet.2023.13.2.1806.

[24] B. Şener, K. Acici, and E. Sümer, "Categorization of Alzheimer's disease stages using deep learning approaches with McNemar's test," *PeerJ Comput. Sci.*, vol. 10, p. e1877, Feb. 2024, doi: 10.7717/peerj-cs.1877.