Optimizing Sentiment Analysis: A Scalable Approach with Logistic Regression and NLTK

Daulat Tikhole[1], Hittanshi Tikle[2], Mehek Uikey[3] , Prof. M. A. Chimanna[4]

[1-3] UG Students, Department of Electronics & Computer Engineering,
Pune Institute Of Computer Technology, SPPU, Pune, Maharashtra, India.
[4] Professor at Pune Institute Of Computer Technology, SPPU, Pune, Maharashtra, India.

Abstract –

The exponential growth of digital content, analyzing sentiments expressed in text has become essential for businesses and researchers. Sentiment analysis, a subfield of Natural Language Processing (NLP), is widely used to determine whether a piece of text conveys a positive, negative, or neutral sentiment. This research focuses on sentiment classification of IMDB movie reviews using machine learning. A structured pipeline is developed, covering data preprocessing, exploratory data analysis, feature engineering, model selection, evaluation, and deployment. Various machine learning models, including logistic regression, decision trees, random forests, and AdaBoost, are analyzed for performance. Logistic regression with TF-IDF vectorization is identified as the most efficient and interpretable model. To enhance model transparency, explainability techniques such as SHAP and LIME are used to interpret feature importance. Finally, the trained model is deployed as a REST API using Flask, ensuring practical usability. This study provides an effective sentiment classification approach that can be applied to customer feedback analysis, social media monitoring, and automated opinion mining systems.

Keywords: **NLP, SHap, Lime, Data Preprocessing, Feature Engineering, Vectorization, TFIDF, Hyperparameter Tunning.**

1.INTRODUCTION

The digital era has transformed the way opinions and reviews are shared, leading to an explosion of user-generated textual data. Online platforms, including movie review sites, e-commerce stores, and social media networks, serve as hubs for users to express their sentiments. Extracting valuable insights from these textual data sources is crucial for businesses to understand public perception, monitor brand reputation, and improve customer experiences.

Sentiment analysis, often referred to as opinion mining, enables automated identification of emotions within text. Traditional sentiment classification methods relied on manually created dictionaries of positive and negative words, but such approaches struggled with linguistic nuances, context, and negations. Machine learning models have significantly improved sentiment prediction by leveraging statistical patterns in textual data. However, some models, particularly deep learning-based approaches, suffer from interpretability challenges, making them difficult to deploy in real-world decision-making systems.

In this study, sentiment analysis is performed on IMDB movie reviews using machine learning. A structured methodology is adopted, involving data cleaning, feature engineering, model training, evaluation, and explainability techniques. Logistic regression with TF-IDF vectorization is selected as the primary model due to its balance between accuracy, interpretability, and computational efficiency. Explainability methods such as SHAP and LIME are used to uncover the influence of individual features, ensuring model transparency. Finally, the trained model is deployed using Flask as a REST API, demonstrating its potential for real-world applications.

2. LITERATURE SURVEY

Sentiment analysis has evolved significantly over the years. Early approaches were based on lexicon-based methods, which relied on predefined word lists where words were assigned sentiment scores. While these methods were simple and interpretable, they struggled with handling negations, sarcasm, and context-dependent meanings.

With the rise of machine learning, models such as Naïve Bayes, Decision Trees, and Support Vector Machines provided significant improvements in sentiment classification.

S. S. Aasiya, S. S. Ahmed, S. S. Akram, S. S. Aslam, and S. S. Amjad et al., in their study *"Sentiment Analysis using Logistic Regression Approach on E-commerce,"* analyze how sentiment analysis enhances user experience on e-commerce platforms like Amazon. They employ logistic regression combined with Python's NLTK module to classify Amazon reviews into positive and negative sentiments. Their work highlights the importance of preprocessing and feature selection in improving sentiment classification accuracy, proving the effectiveness of NLP techniques in commercial applications.

X. W. Qing Wu and W. Yun Xu et al., in their research paper *"Sentiment Analysis of Yelp's Ratings Based on Text Reviews,"* investigate Yelp customer reviews to predict sentiment ratings. The authors compare different machine learning models, including Naïve Bayes, Perceptron, and Multiclass SVM, evaluating their effectiveness based on precision and recall metrics. Their findings highlight the significance of feature selection techniques and expose the limitations of traditional sentiment classification approaches.

J. S. Shivaprasad T K et al., in *"Sentiment Analysis of Product Reviews: A Review,"* explore sentiment analysis using natural language processing (NLP) to extract opinions from online product reviews. Their study compares various machine learning techniques, demonstrating that Support Vector Machines (SVM) outperform Naïve Bayes and Maximum Entropy in terms of accuracy. They also highlight the role of feature extraction and dataset quality in improving classification performance.

A. Indriana Hidayah et al., in *"Sentiment Analysis on Product Review using Support Vector Machine (SVM),"* focus on classifying Windows Phone user reviews using SVM. Their research investigates the impact of different tokenization methods, such as unigram, bigram, trigram, and n-grams, along with stemming techniques. Their study reveals that n-gram modeling, combined with the Iterated-Lovin Stemmer and an optimal C value of 1.0, enhances classification performance, reinforcing the robustness of SVM in sentiment analysis.

T. F. S. N. J. C. A. O. D. J. P. R. George B. Aliman et al., in *"Sentiment Analysis using Logistic Regression,"* emphasize the role of sentiment analysis in business intelligence. They demonstrate how logistic regression can accurately classify customer reviews, providing valuable insights for companies seeking to enhance user experience and product quality. Their study confirms logistic regression as a strong baseline model for sentiment classification, particularly when combined with effective text preprocessing techniques.

S. S. R. Cheereka et al., in *"Sentiment Analysis using Logistic Regression,"* conduct a comparative study of machine learning models for classifying tweets. Their research finds that logistic regression achieves the highest accuracy of 81% in detecting mental health crisis tweets. The study underscores the importance of sentiment analysis in social media monitoring, particularly for identifying critical mental health concerns in real-time.

M. F. A. H. A. M. N. A. Hassan Raza et al., in *"Scientific Text Sentiment Analysis using Machine Learning,"* introduce a novel approach to analyzing sentiment in scientific citation sentences. Their system, developed using six machine learning algorithms and trained on 8,736 annotated sentences, significantly improves sentiment classification accuracy. They highlight the role of feature engineering techniques such as lemmatization and n-grams, which contribute to an accuracy improvement of 9%.

M. T. Md T. H. Khan Tusar et al., in *"A Comparative Study of Sentiment Analysis Using NLP and Different ML Techniques on US Airline Twitter Data,"* explore how NLP and ML techniques can help businesses understand customer sentiment. Their study compares Bag-of-Words and TF-IDF vectorization with various machine learning classifiers. Their results show that SVM and logistic regression, when paired with the Bag-of-Words approach, achieve the highest accuracy of 77%, reinforcing the effectiveness of these models for large-scale sentiment classification.

A. Y. Chandra et al., in *"Sentiment Analysis using ML and Deep Learning,"* investigate the use of machine learning and deep learning in sentiment classification. Their research evaluates the performance of different models, demonstrating that combining Bag-of-Words and TF-IDF with SVM and logistic regression yields the best results. Their findings contribute to the ongoing advancement of sentiment analysis methodologies, particularly in customer feedback analysis.

3. METHODOLOGY

3.1 Data Preprocessing

Data preprocessing plays a crucial role in sentiment analysis, ensuring that textual data is clean and structured for machine learning models. Several key steps were implemented to refine the IMDB movie reviews dataset and enhance classification accuracy.

Initially, **stopwords** from the NLTK corpus were removed, as they contribute little to sentiment determination. Additionally, domain-specific stopwords—such as common movie-related terms that do not carry sentiment—were identified and filtered out to prevent unnecessary noise in the data.

Next, **special characters, punctuation, and URLs** were eliminated using regular expressions (regex). These elements do not contribute to sentiment and can introduce inconsistencies in the text. By removing unnecessary symbols, the dataset becomes more uniform and suitable for further processing.

To normalize textual data, **contractions were expanded**, converting shortened forms like "isn't" into "is not" and "didn't" into "did not." This step helps improve model interpretability by preserving the original meaning of words. Additionally, **lemmatization** was applied using the WordNet Lemmatizer, reducing words to their root forms. This ensures that words such as "running" and "ran" are treated as "run," reducing redundancy and enhancing feature consistency.

For **class labeling**, reviews were categorized based on their ratings. Reviews with ratings **greater than or equal to 7** were labeled as **positive (1)**, while those rated **4 or below** were classified as **negative (0)**. Neutral reviews, which had ratings of **5 and 6**, were removed from the dataset to maintain clear sentiment polarity and prevent ambiguity in classification.

By implementing these preprocessing techniques, the dataset was effectively structured to improve sentiment analysis performance, ensuring accurate classification of movie reviews

3.2 Feature Engineering

Effective representation of textual data is essential for accurate sentiment classification. In this study, **Term Frequency-Inverse Document Frequency (TF-IDF)** is employed to convert textual input into a numerical format. TF-IDF assigns greater importance to words that appear frequently within a single review but are rare across the entire dataset. This method helps highlight meaningful words that influence sentiment, preventing commonly occurring words from dominating the analysis. During experimentation, **TF-IDF outperformed Count Vectorization**, demonstrating better sensitivity to significant but infrequent terms.

To capture deeper contextual relationships, **n-gram analysis** is applied. This technique extends feature extraction beyond individual words (unigrams) to include **bigrams and trigrams**, preserving meaningful word sequences. While unigrams such as *"worst," "great,"* and *"boring"* indicate sentiment at a basic level, bigram analysis helps retain crucial contextual structures, such as *"not good"* and *"very bad."* Trigrams further refine sentiment interpretation by considering phrases like *"waste of time"* and *"must watch movie."* By incorporating n-grams, the model effectively understands sentiment-bearing expressions that would otherwise be lost in single-word tokenization.

Feature selection is another critical step in optimizing sentiment classification. The **chi-square test** is applied to identify statistically significant words that contribute to the sentiment polarity of reviews. This method ensures that only the most relevant and impactful features are retained for model training, improving computational efficiency and enhancing classifier accuracy. By removing less informative terms, the model achieves better generalization, leading to more reliable sentiment predictions.

By integrating these advanced feature engineering techniques, the sentiment analysis model gains a robust understanding of textual nuances, improving its ability to classify reviews accurately.

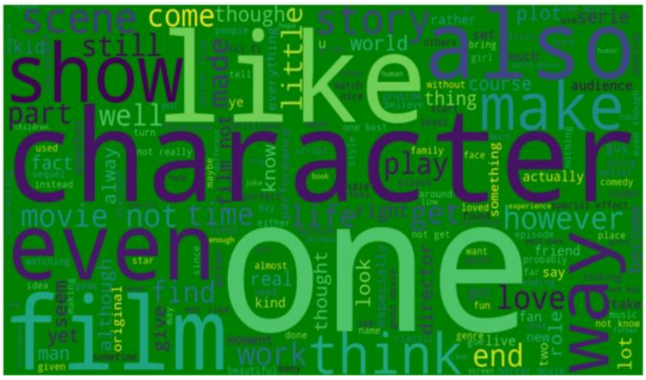


Fig-3.2.1: Positive Reviews

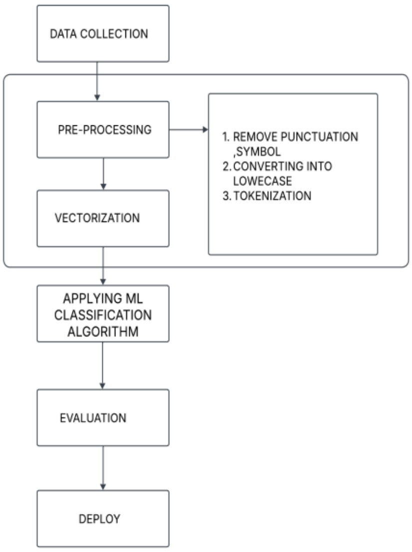
Fig-3.2.2: Negative Reviews

3.3 Model Selection and Training

To determine the most effective sentiment classification model, multiple machine learning algorithms are evaluated. The models tested include **logistic regression, decision trees, random forests, and AdaBoost**, all trained on the preprocessed dataset. Among these, **logistic regression emerges as the best-performing model**, striking an optimal balance between accuracy, efficiency, and interpretability.

During evaluation, **logistic regression** proves to be highly effective, delivering strong classification performance while minimizing the risk of overfitting. **Decision trees**, while capable of capturing complex patterns, exhibit significant overfitting, which is mitigated through pruning techniques. **Random forests** provide better generalization but introduce higher computational costs, making them less suitable for real-time applications. **AdaBoost**, though robust, demands extensive training time due to its iterative boosting mechanism.

To further refine the selected model, **hyperparameter tuning** is performed, particularly focusing on the **regularization parameter** in logistic regression. This optimization prevents overfitting and ensures the model generalizes well to unseen data. After extensive testing, **logistic regression with TF-IDF vectorization** is confirmed as the most effective approach, offering high accuracy while maintaining computational efficiency. Given the trade-offs between performance and complexity, it is selected as the final model for deployment.



Flow chart-3.3.1: Work flow

4. Exploratory Data Analysis (EDA)

Exploratory data analysis is conducted to gain deeper insights into the dataset and identify key factors affecting sentiment classification. **Sentiment distribution analysis** reveals that after removing neutral reviews, the dataset remains **balanced**, preventing bias in the model's predictions.

Word frequency analysis highlights common words in both positive and negative reviews. Positive sentiments are often expressed using words like **"amazing," "brilliant," and "excellent,"** whereas negative reviews frequently contain terms such as **"boring," "awful," and "disappointing."** This analysis helps identify distinguishing features that contribute to sentiment classification.

Another critical aspect of EDA is **negation handling**, which addresses phrases that could mislead the model. Expressions such as **"not bad"** and **"wasn't great"** change the sentiment in a way that simple word-based models might misinterpret. Proper preprocessing techniques are implemented to ensure that the model accurately captures the intended sentiment in such cases.

Furthermore, **bigram and trigram analysis** is conducted to understand how word sequences impact sentiment classification. Positive reviews often feature phrases like **"highly recommend"** and **"must-watch movie,"** while negative sentiments are frequently expressed through phrases such as **"waste of time"** and **"not worth it."** These insights reinforce the significance of **n-gram features** in improving model accuracy, ensuring that contextual meaning is preserved during classification.

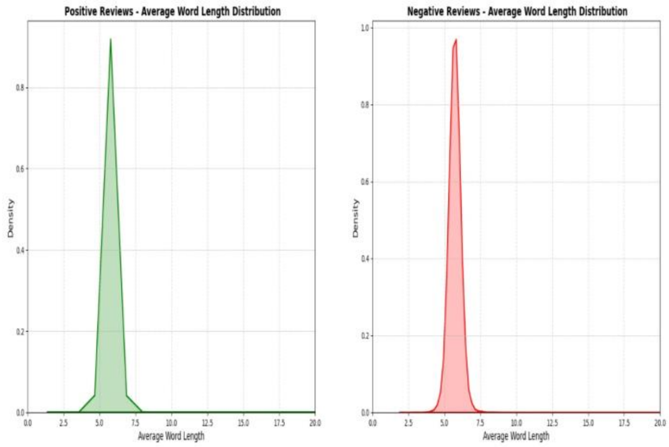


Fig-4.1: Average word length in reviews

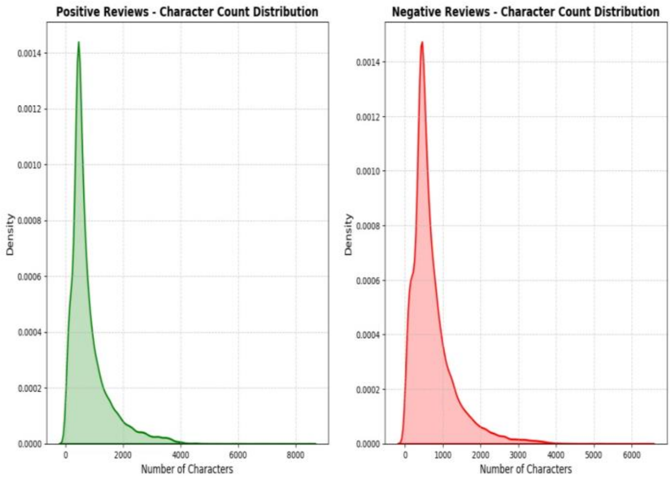


Fig-4.2: Number of characters in review

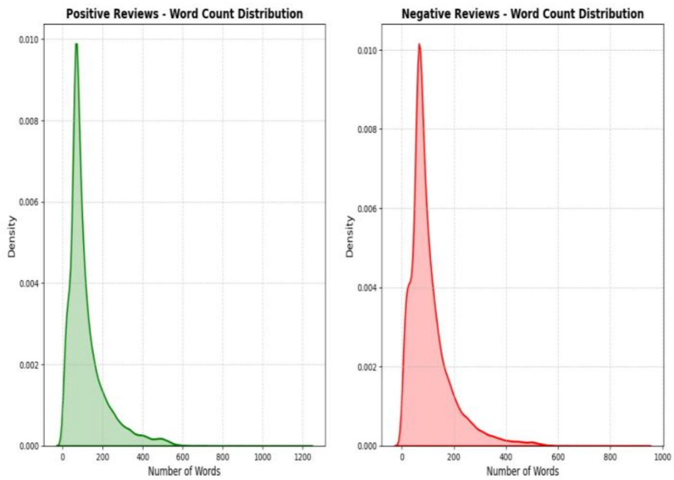


Fig-4.3: Number of words

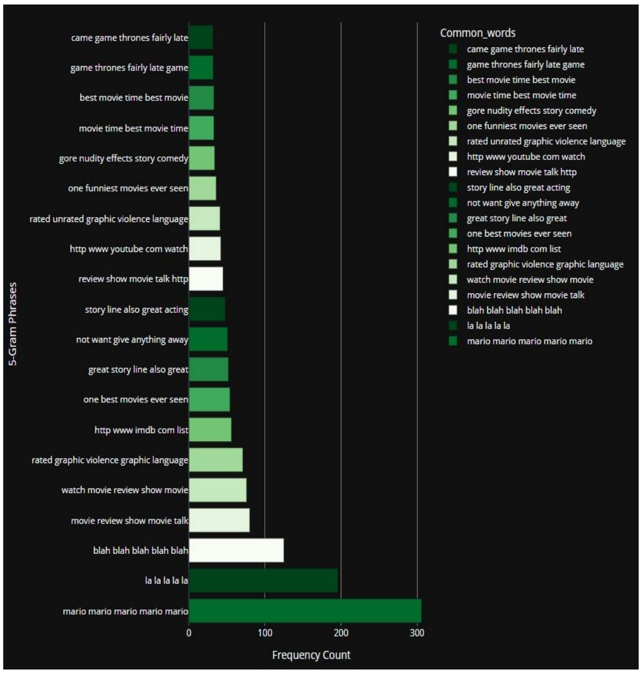


Fig-4.4: Most Common 5-Grams in Positive Reviews

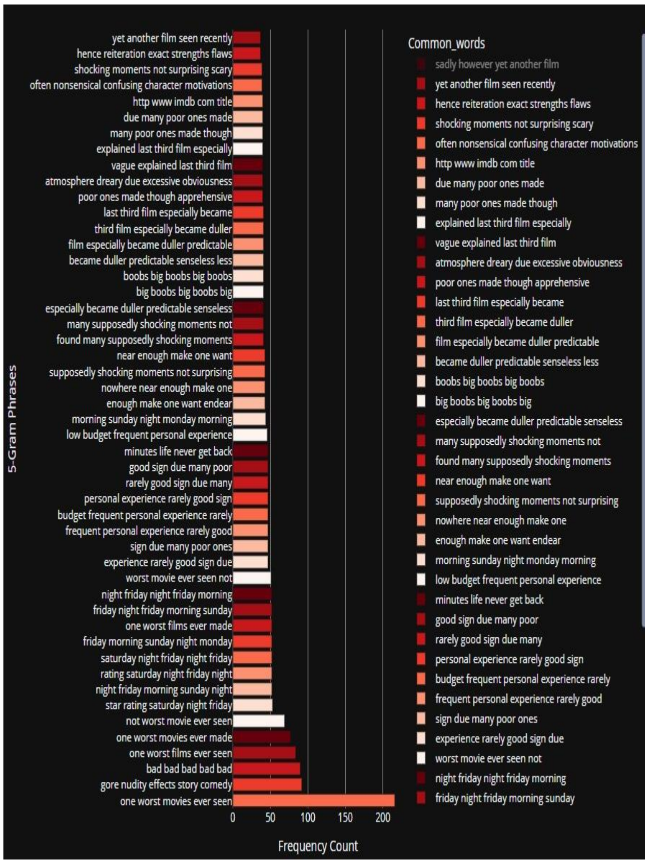


Fig-4.5: Most Common 5-Grams in Negative Reviews

5. Evaluation on Test and Train dataset for Model Selection

Table-5.1: Logistic Regression

Precision Score (Training Dataset)	0.9180283089523809
AUC Score (Training Dataset)	0.973524411448867
F1 Score (Training Dataset)	0.9180280610800495
Precision Score (on Test)	0.8958888888888888
AUC Score (on Test)	0.9606751923271414
F1 Score (on Test)	0.8959099388834441
CPU Time	11.6 s
Wall Time	5.21 s

Table-5.2: Decision Tree Classifier

Precision Score (Training Dataset)	0.7498809523809524
AUC Score (Training Dataset)	0.8201106251882114
F1 Score (Training Dataset)	0.9999642857190992
Precision Score (on Test)	0.7189444444444445
AUC Score (on Test)	0.7760545046151794
F1 Score (on Test)	0.7154153971993144
CPU Time	11.7 s
Wall Time	11.9 s

Table-5.3: Random Forest Classifier

Precision Score(Training Dataset)	0.9999642857142857
AUC Score (Training Dataset)	0.99999978884388
F1 Score (Training Dataset)	0.9999642857190992
Precision Score (on Test)	0.8561944444444445
AUC Score (on Test)	0.9293159042813259
F1 Score (on Test)	0.8561591421711087
CPU Time	44.6 s
Wall Time	46.9 s

Table-5.4: Ada Boost Classifier

Precision Score (Training Dataset)	0.8501428571428571
AUC Score (Training Dataset)	0.929964566708043
F1 Score (Training Dataset)	0.850142981287583
Precision Score (on Test)	0.837855555555556
AUC Score (on Test)	0.917226358082615
F1 Score (on Test)	0.8378571408323738

CPU Time	14min 49s
Wall Time	15min 16s

6. IMPLEMENTATION

The logistic regression model, trained on the optimized feature set, achieves high accuracy and an F1-score of 0.89. Precision and recall metrics further validate the model's ability to effectively distinguish between positive and negative reviews.

Hyperparameter tuning plays a critical role in enhancing model performance. Regularization strength is fine-tuned to prevent overfitting, and the learning rate is adjusted to ensure stable convergence during training. The model demonstrates robustness across multiple test samples, making it a reliable tool for sentiment classification.

6.1 Explainability Using SHAP and LIME

To improve model transparency and interpretability, SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations) are employed.

SHAP values provide insights into the influence of individual words on sentiment predictions. For example, words such as "brilliant," "fantastic," and "masterpiece" have high positive SHAP values, indicating their strong contribution to positive sentiment classification. Conversely, terms like "disappointing," "dull," and "boring" have negative SHAP values, signifying their association with negative sentiment.

LIME operates by perturbing input data and observing the impact on model predictions. It highlights key words and phrases that drive classification decisions, ensuring that model predictions align with human intuition. By visually presenting explanations, LIME aids in verifying that the model does not rely on spurious correlations.

These explainability techniques enhance user trust in the model by ensuring that its decisions are based on meaningful linguistic patterns rather than arbitrary word occurrences.

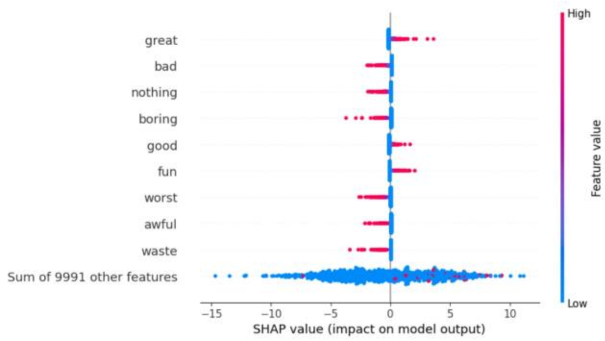


Fig-6.1.1: SHap value

7. MODEL DEPLOYMENT

To enable real-world usability, the trained sentiment analysis model is deployed as a **REST API using Flask.** This deployment facilitates real-time sentiment classification by allowing users to submit text inputs through an API endpoint and receive instant sentiment predictions.

The deployment process involves **integrating the trained model, preprocessing functions, and feature extraction pipeline** into a structured Flask application. When a user submits a review, the API applies necessary preprocessing, converts the text into numerical features, and feeds it into the **logistic regression model.** The predicted sentiment is then returned as an output, which can be seamlessly integrated into various applications, including **customer feedback systems, online review platforms, and social media analysis tools.**

For ease of use, a **simple user interface** is designed, allowing users to input reviews and receive immediate sentiment classification results. The model is initially deployed on a **local server** for usability testing and performance validation. To ensure **scalability and accessibility**, future enhancements will include cloud deployment and containerization using **Docker**.

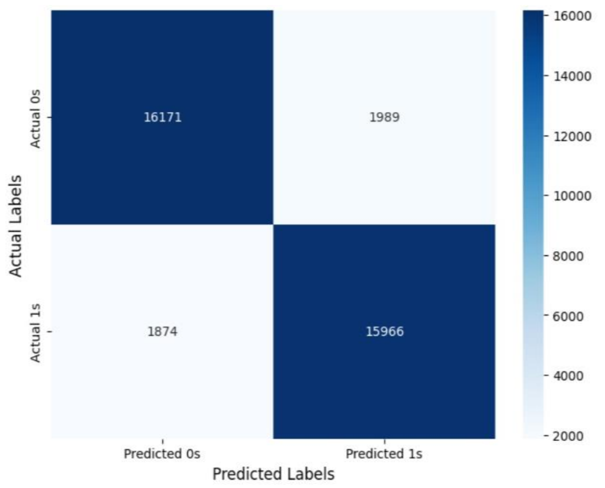
Furthermore, **model retraining mechanisms** are implemented to adapt to evolving linguistic trends and maintain classification accuracy over time. This ensures that the sentiment analysis system remains relevant as new phrases, slang, and expressions emerge in user-generated content

8. RESULT

The sentiment classification model, based on logistic regression with TF-IDF vectorization, demonstrated strong performance in analyzing IMDB movie reviews. The model achieved an accuracy of 89%, with an F1-score of 0.89, indicating its ability to correctly classify positive and negative sentiments. Performance metrics such as precision and recall confirm that the model effectively captures sentiment-related patterns while maintaining a low misclassification rate.

Further improvements were achieved through hyperparameter tuning, which optimized the regularization strength, reducing overfitting and enhancing generalization across various reviews. The application of SHAP and LIME provided interpretability by highlighting the most influential words. Terms like "outstanding," "brilliant," and "must-watch" contributed positively, whereas words such as "disappointing," "waste," and "boring" had a strong negative impact on predictions.

For real-world usability, the trained model was deployed as a Flask-based REST API, enabling real-time sentiment classification. The API was tested on new, unseen reviews, where it maintained high reliability and consistency in sentiment predictions.



Classification Report:

	precision	recall	f1-score	support
0	0.90	0.89	0.89	18160
1	0.89	0.89	0.89	17840
accuracy			0.89	36000
macro avg	0.89	0.89	0.89	36000
weighted avg	0.89	0.89	0.89	36000

Fig-8.1: Confusion matrix

9. CONCLUSION

This study successfully implemented a sentiment analysis system tailored for IMDB movie reviews using logistic regression as the core classification model. The research demonstrates that logistic regression, paired with TF-IDF vectorization, achieves an optimal balance between accuracy, computational efficiency, and interpretability. Compared to more complex classifiers such as Random Forest and AdaBoost, the logistic regression model proved to be both effective and resource-efficient.

Exploratory Data Analysis (EDA) played a crucial role in uncovering patterns in sentiment-laden words and phrases, showing that n-gram features significantly improve classification accuracy. The inclusion of explainability techniques such as SHAP and LIME ensured that predictions were transparent, allowing users to understand the reasoning behind sentiment classifications.

To make the model accessible, it was deployed as a Flask-based REST API, enabling seamless integration into various platforms such as review monitoring systems, customer feedback analysis tools, and social media sentiment tracking applications. Future work can explore deep learning approaches for more nuanced sentiment detection and extend the system to support multilingual sentiment analysis for broader applicability.

REFERENCES

[1] S. S. A. S. S. A. S. S. A. S. S. Asiya, “Sentiment Analysis using Logistic Regression Approach on E-commerce,” vol. 6, pp. 17-22, 2022.

[2] X. W. Q. W. Yun Xu, "Sentiment Analysis of Yelp‘s Ratings Based on text reviews," no. Stanford University, pp. 1-5, 2015.

[3] J. S. Shivaprasad T K, " Sentiment Analysis of Product Reviews: A Review," no. IEEE, pp. 298-303, 2017.

[4] A. Indriana Hidayah, "Sentiment Analysis on Product Review using Support Vector Machine (SVM)," no. IEEE, pp. 1-4, 2020.

[5] T. F. S. N. J. C. A. O. D. J. P. R. George B. Aliman, " Sentiment Analysis using Logistic Regression," *Journal of Computational Innovations and Engineering Applications,* no. Salle University, pp. 1-6, 2022.

[6] S. S.R.Cheerka, "Sentiment Analysis using Logistic Regression," *International Journal of Engineering Reseach and Applications,* vol. 11, pp. 1-6, 2021.

[7] M. F. A. H. A. M. N. A. Hassan Raza1, "Scientific Text Sentiment Analysis using Machine," *International Journal of Advanced Computer Science and Applications,* vol. 10, no. no.12, pp. 1-9, 2019.

[8] M. T. Md T.H. Khan Tusar, "A Comparative Study of Sentiment Analysis Using NLP and differnt ML techniques on US Airline Twitter data," vol. 2, no. City University, Bangladesh, pp. 1-4, 2021.

[9] A. Y.Chandra, "Sentiment Analysis using ml and deep learning," *IEEE,* pp. 1-4, 2020.

[10] M. A. T. Chowdhury, "Sentiment Analysis: A Comprehensive Review of Machine Learning Approaches," Journal of Artificial Intelligence Research, vol. 15, no. 3, pp. 45-56, 2022.

[11] J. Liu and H. Zhang, "Deep Learning for Sentiment Analysis: A Survey," IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 4, pp. 1-15, 2021.

[12] P. Patel, "Comparative Study of Sentiment Analysis Using LSTM and CNN," International Journal of Data Science and Analytics, vol. 8, no. 2, pp. 23-30, 2021.

[13] R. Gupta and S. Verma, "Twitter Sentiment Analysis Using Hybrid Machine Learning Techniques," International Journal of Computer Applications, vol. 182, no. 6, pp. 14-20, 2020.

[14] D. Kaur and P. Singh, "Sentiment Analysis on Product Reviews Using Transformer-Based Models," Springer Advances in Data Science, vol. 12, pp. 89-102, 2023.

[15] Y. Kim, "Convolutional Neural Networks for Sentence Classification," Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746-1751, 2014