International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

Explainable Hate Speech Detection and Classification

Er. Syed Kaiser Mehraj[1], Areeba Nisar[2], Aazeen Sultan[3], Shabra Sultan[4]

[1]Assistant Professor Department of CSE SSM College of Engineering
[2,3,4]Students SSM College of Engineering Pattan, Baramulla

***Abstract*-- The widespread use of social media and online platforms has made communication faster and more accessible, allowing people to share information and opinions with a global audience. However, this has also increased the spread of harmful content, including hate speech. Hate speech involves offensive or abusive language targeting individuals or groups based on characteristics such as race, religion, gender, or ethnicity.**

Automatic hate speech detection is a challenging task in Natural Language Processing (NLP) because the meaning of text depends heavily on context. Although traditional machine learning and deep learning models can classify text as hate or non-hate speech, they often lack transparency by providing little explanation for their predictions.

To improve interpretability, this project presents an Explainable Hate Speech Detection and Classification System using NLP models such as BERT combined with explainable AI techniques, including LIME, SHAP, and Integrated Gradients. The proposed system not only identifies hate speech but also explains the factors influencing its predictions, enhancing transparency and accountability.

***Keywords*-- Hate Speech Detection, Natural Language Processing (NLP), BERT, Explainable Artificial Intelligence (XAI), Text Classification**

I. Introduction

1.1 Background And Context

Deep learning models such as BERT, BiLSTM, CNN, and RNN-based hybrid architectures have achieved high accuracy in text classification tasks, particularly in detecting offensive and hateful content. However, these models often operate as black boxes, making it difficult to understand how they generate their predictions.

To improve model transparency, Explainable Artificial Intelligence (XAI) techniques have gained significant attention. The primary objective of XAI is to provide clear explanations of the decision-making process of machine learning models.

- *LIME (Local Interpretable Model-Agnostic Explanations):* Explains individual predictions by identifying the contribution of each word to the model's decision.
- *SHAP (SHapley Additive exPlanations):* Offers both local and global explanations by measuring the impact of each feature on the model's predictions.
- *Integrated Gradients:* Highlights the words or phrases that have the greatest influence on the model's output.

1.2 Problem Statement

With the rapid growth of social media platforms such as Twitter, Facebook, Instagram, and Reddit, online communication has become more accessible and widespread. However, these platforms have also facilitated the spread of hate speech, which includes abusive or discriminatory content directed at individuals or groups based on attributes such as religion, gender, race, caste, or nationality.

The increasing presence of hate speech poses significant social challenges, including harassment, discrimination, and the promotion of violence against vulnerable communities. Consequently, there is a growing need for automated systems capable of identifying and limiting the spread of such harmful content.

Natural Language Processing (NLP) and Machine Learning (ML) techniques have been widely applied to classify text as hateful or non-hateful. Despite their effectiveness, these models generally operate as black boxes, producing predictions without explaining the reasoning behind them.

This lack of interpretability creates several challenges:

- Users and content moderators cannot determine why a post has been classified as hate speech.
- Hidden biases in the model or training data may remain undetected.
- Limited transparency reduces trust and accountability in automated moderation systems.

1.3 Objectives Of The Project

Objective 1: To collect and preprocess hate speech datasets

Objective 2: To develop baseline and advanced classification models

Objective 3:

- Build a simple user interface or visualization module showing:ϖ
- The text inputϖ
- Whatever comes out - hate or not hate - that is the callϖ
- Highlighted words contributing to the prediction.ϖ

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

Objective 4: To analyze and document findings

- Identify patterns of bias, misclassification, or ambiguity. Perform an in-depth analysis of model performance and explainability results.
- Provide insights and recommendations for developing transparent AI moderation ϖ systems.

1.4 Scope Of The Project

The scope of this project includes the development and evaluation of an explainable hate speech detection system using machine learning and deep learning models such as SVM, BiLSTM, and BERT. The study integrates Explainable Artificial Intelligence (XAI) techniques, including LIME, SHAP, and Integrated Gradients, to provide clear explanations for model predictions.

The project also involves comparing different classification models and explanation methods to determine their effectiveness. Visual explanation techniques will be used to identify the words or features that influence classification decisions. The performance of the proposed system will be assessed using quantitative measures such as accuracy and F1-score, along with qualitative evaluation of interpretability and explanation quality.

The implementation will be carried out using open-source datasets, including Jigsaw, HASOC, and Twitter datasets, with Python, TensorFlow/PyTorch, and XAI libraries.

1.5 Significance Of The Study

The rapid growth of social media platforms and online communication has increased the spread of hate speech and other harmful content. Although automated moderation systems are widely used, most operate as black-box models, making their decisions difficult to interpret. This lack of transparency can reduce user confidence and raise concerns about fairness and accountability.

This study proposes an Explainable Artificial Intelligence (XAI)-based hate speech detection system that combines Natural Language Processing (NLP) with explainability techniques to improve both detection accuracy and model transparency. The system not only identifies hateful content but also explains the reasons behind its predictions, supporting more reliable and responsible content moderation.

The significance of this study is outlined below:

- *Academic Significance:* The study contributes to research in Natural Language Processing, Machine Learning, and Explainable Artificial Intelligence by demonstrating the application of XAI techniques in hate speech detection.
- *Technical Significance*: It presents a framework that integrates classification models with explainability methods, which can be extended to multilingual datasets and other forms of harmful online content.
- *Practical Significance:* The proposed system supports social media platforms and content moderators by providing transparent explanations, enabling more informed and accurate moderation decisions.
- *Ethical Significance:* The integration of explainable AI enhances fairness, transparency, and accountability by helping users and moderators understand the reasoning behind automated classifications.
- *Social Significance:* By detecting and explaining harmful content, the system contributes to reducing online toxicity, promoting respectful communication, and creating safer digital environments.

II. Literature Review

The field of hate speech detection has advanced significantly with the adoption of deep learning and transformer-based language models. Recent research has focused not only on improving classification accuracy but also on enhancing the interpretability of model predictions through Explainable Artificial Intelligence (XAI).

Liu et al. (2021) proposed the use of RoBERTa, an optimized transformer-based language model that improves upon BERT through enhanced pre-training strategies. Their study demonstrated superior performance in complex and multilingual text classification tasks, making RoBERTa highly effective for hate speech detection. However, despite its strong predictive capabilities, the model provides limited insight into the reasoning behind its decisions, reducing transparency and user trust.

Mathew et al. (2021) introduced HateXplain, a benchmark dataset containing human-annotated explanations alongside hate speech labels. The dataset enabled researchers to develop and evaluate explainable hate speech detection systems by incorporating human rationales into model training. Although HateXplain significantly advanced explainable NLP research, training and evaluating models on the dataset requires considerable computational resources.

Zampieri et al. (2020) investigated the application of transformer-based models for offensive language and hate speech detection. Their work demonstrated that transformer architectures outperform traditional machine learning methods in terms of classification accuracy and contextual understanding. However, these models still function primarily as black-box systems, offering limited interpretability of their predictions.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

The current proposed work (2026) builds upon these existing studies by integrating the BERT language model with Explainable Artificial Intelligence (XAI) techniques such as LIME and SHAP. Unlike conventional hate speech detection systems that focus only on prediction accuracy, the proposed approach emphasizes both accurate classification and transparent decision-making. By highlighting the words and phrases that influence model predictions, the system enables users and content moderators to better understand and verify the classification process. Nevertheless, further improvements are needed to enhance computational efficiency and scalability for large-scale real-world deployment.

Sanh et al. (2019) proposed DistilBERT, a lightweight version of BERT designed to reduce model size while maintaining competitive performance. The study demonstrated that DistilBERT achieves faster inference and lower computational requirements, making it suitable for resource-constrained applications. However, this efficiency comes with a slight reduction in classification accuracy compared to the original BERT model.

Mozafari et al. (2019) developed a BERT-based model for hate speech detection and demonstrated its effectiveness in accurately identifying offensive and hateful content. By leveraging contextual word representations, the model outperformed several traditional machine learning approaches. Despite its high performance, the model lacks transparency, as it does not provide explanations for its predictions.

Devlin et al. (2018) introduced BERT (Bidirectional Encoder Representations from Transformers), a transformer-based language model that captures bidirectional contextual information from text. BERT significantly improved the performance of various Natural Language Processing tasks, including text classification and hate speech detection. Nevertheless, the model operates as a black box, making it difficult to interpret the reasoning behind its decisions.

Agrawal and Awekar (2018) applied a Bidirectional Long Short-Term Memory (BiLSTM) network for hate speech detection. Their approach effectively captured contextual information by processing text in both forward and backward directions, leading to improved sequence understanding and classification accuracy. However, the model requires substantial computational resources and longer training times, limiting its efficiency for large-scale applications.

Gambäck and Sikdar (2017) proposed a character-level Convolutional Neural Network (CNN) for detecting hate speech in social media text.

Their model effectively handled noisy and informal language, including slang, abbreviations, and spelling variations commonly found in online platforms. Despite its robustness in processing social media content, the model offers limited interpretability, making it difficult to understand the factors influencing its predictions.

III. Methodology

Software Engineering Approach

3.1 Development Methodology

The Explainable Hate Speech Detection System was developed using the Agile software development methodology, which emphasizes iterative development, continuous testing, and regular feedback. This approach is well suited for machine learning projects, where models require repeated training, evaluation, and refinement.

The project was completed in sequential iterations, including data collection, data preprocessing, model development, interface design, testing, and deployment. Each phase was evaluated before moving to the next to improve both prediction accuracy and model explainability.

Core Principles of Agile

Customer Collaboration:

Regular feedback from supervisors and stakeholders was incorporated throughout the development process to ensure the system met user requirements and improved usability.

Iterative Development:

The system was built incrementally through multiple stages, including data preparation, model training, explainability integration, interface development, and testing.

Flexibility and Adaptability:

Datasets, model parameters, and visualization methods were updated whenever necessary to improve system performance and interpretability.

Self-Organizing Team:

Development tasks such as data preprocessing, model implementation, user interface design, and system integration were completed collaboratively to ensure efficient project execution.

Continuous Feedback and Improvement:

The model was continuously evaluated using performance metrics and user feedback, allowing regular improvements in both classification accuracy and explanation quality.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

3.2 How The Agile Model Operates

1. Iterative Development

Sprint 1: Requirement Analysis and Data Collection

Project requirements were identified, and hate speech datasets were collected from publicly available sources for training and testing.

Sprint 2: Data Preprocessing

The collected text was cleaned by removing unnecessary symbols, links, and special characters. Tokenization and text normalization were then performed to prepare the data for model training.

Sprint 3: Feature Extraction and Model Training

Text features were extracted using TF-IDF, and machine learning and deep learning models were trained to classify text as hateful or non-hateful. The models were refined through multiple iterations to improve accuracy.

Sprint 4: Explainability Integration

Explainable AI techniques were incorporated to provide clear explanations for the model's predictions, improving transparency and user trust.

Sprint 5: Web Application Development

A web-based interface was developed by integrating the frontend with the backend, allowing users to submit text and view both predictions and explanations.

Sprint 6: Testing and Deployment

The complete system was tested for accuracy, performance, usability, and reliability before deployment.

2. Regular Meetings

Daily Discussions:

The team reviewed completed tasks, discussed challenges, and planned upcoming activities.

Sprint Planning:

Objectives were defined for each sprint, and responsibilities were assigned accordingly.

Sprint Review and Retrospective:

Each completed module was evaluated, and feedback was used to improve the next development cycle.

3. Collaboration and Communication

Cross-Functional Collaboration:

The project combined Natural Language Processing, machine learning, backend development, and frontend design to build an integrated system.

Development Tools:

Python, Jupyter Notebook, Visual Studio Code, and GitHub were used for implementation, testing, version control, and collaboration.

4. User-Centric Approach

Stakeholder Involvement:

Regular feedback from supervisors and reviewers ensured that the project met its objectives.

User Testing:

Test users evaluated the web application for usability, prediction accuracy, and explanation clarity.

5. Adaptability and Flexibility

Change Management:

The Agile approach allowed updates to datasets, preprocessing methods, and model parameters whenever improvements were required.

Continuous Improvement:

User feedback and evaluation results were used to refine the system throughout development.

6. Quality Assurance

Continuous Testing:

Unit testing and integration testing were conducted regularly to ensure system reliability.

Model Evaluation:

The models were evaluated using Accuracy, Precision, Recall, and F1-Score to measure classification performance.

Performance Monitoring:

Prediction speed, resource utilization, and explanation quality were monitored to maintain overall system efficiency.

Agile Benefits in the Proposed System

The Agile methodology enabled rapid development through iterative cycles, continuous testing, and regular feedback. It improved collaboration, supported changing project requirements, enhanced model performance through continuous refinement, and increased user confidence by providing transparent and explainable predictions.

IV. Analysis Section

4.1 Software Requirements Specification (SRS)

The Software Requirements Specification (SRS) defines the functional and non-functional requirements of the Explainable Hate Speech Detection System.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

It provides a clear description of the system's expected behavior, features, and performance constraints. The system is designed not only to detect hate speech but also to explain its predictions using Explainable Artificial Intelligence (XAI) techniques, ensuring transparency and reliability.

4.1.1 Functional Requirements

• *Text Input Module*

The system allows users to enter text manually or upload text files for hate speech analysis. Once the input is received, it is processed automatically.

• *Text Preprocessing*

The input text is cleaned by removing special characters, URLs, punctuation, and unnecessary words. The remaining text is normalized and prepared for classification.

• *Hate Speech Detection*

The system classifies the processed text as **Hate Speech** or **Normal Speech** based on the trained machine learning model.

• *Explainable AI Module*

The system highlights the important words or phrases that contributed to the prediction using XAI techniques, enabling users to understand the reasoning behind the classification.

• *Prediction Results*

The output displays the predicted class along with the confidence score and a visual explanation of the decision.

• *Dataset Management*

The system supports dataset loading, preprocessing, training, and evaluation for effective model development.

• *Model Training and Evaluation*

Machine learning and deep learning models are trained and evaluated using standard performance metrics to ensure accurate classification.

• *User Interface*

A simple web-based interface enables users to input text, perform analysis, and view prediction results with corresponding explanations.

4.1.2 Non-Functional Requirements

• *Performance*

The system should generate predictions within a few seconds while maintaining high classification accuracy.

• *Scalability*

The application should efficiently handle multiple users and large volumes of text without significant performance degradation.

• *Security*

User inputs and system data must be protected from unauthorized access, ensuring data privacy and secure system operation.

• *Reliability*

The system should provide consistent and dependable predictions with minimal errors or downtime.

• *Usability*

The interface should be user-friendly, allowing users to easily submit text and understand the prediction results and explanations.

• *Maintainability*

The system should support easy updates to datasets, models, and explainability techniques for future improvements.

4.1.3 USE CASES

• *Analyze Text*

Actor: User The user enters or uploads text for hate speech analysis.

• *Generate Prediction Actor:* System The system preprocesses the input and classifies it as Hate Speech or Normal Speech.

• *Display Explanation Actors:* User, System The system displays the prediction along with highlighted words that influenced the decision.

• *Train Model Actor:* Administrator/Developer The system trains the model using labeled datasets and evaluates its performance.

• *View Results Actor: User*

The user views the prediction, confidence score, and explanation.

4.1.4 Assumptions And Constraints

- The system requires an internet connection and a web browser.
- Model performance depends on the quality of the training data.
- Only English text is supported.
- Images, audio, and video inputs are not processed.
- Model training requires adequate computational resources.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

4.2 Data Flow Diagrams (DFD)

The Data Flow Diagram (DFD) illustrates how data moves through the Explainable Hate Speech Detection System, from user input to prediction and explanation.

DFD Elements

• *Process (Rounded Rectangle):* Performs preprocessing, classification, and explanation generation.

• *Data Flow (Arrow):* Shows the movement of data between system components.

• *Data Store (Open Rectangle):* Stores datasets, trained models, and prediction records.

• *External Entity (Rectangle):* Represents the user who provides input and receives the prediction and explanation.

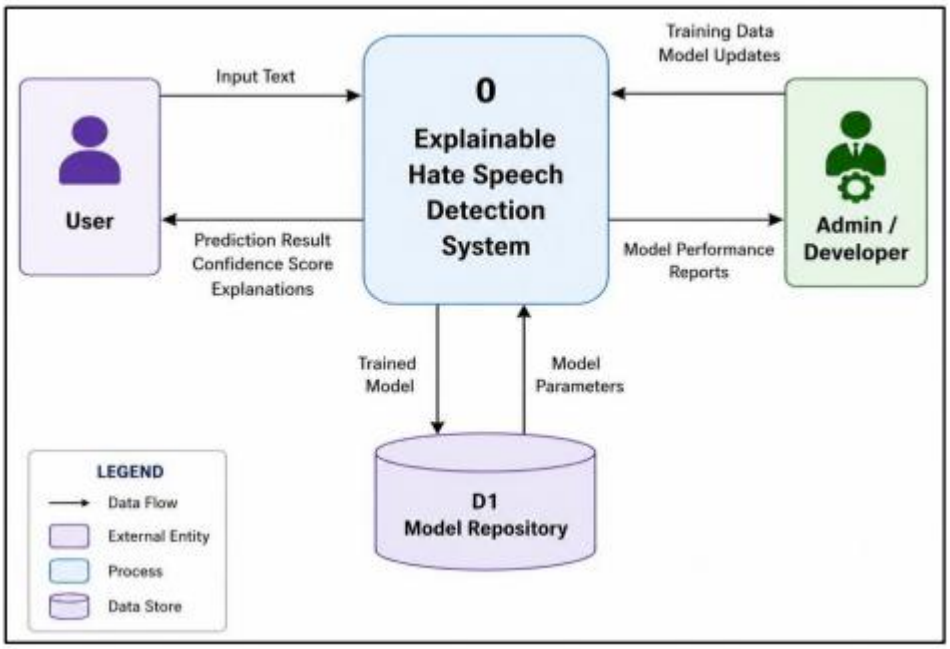

Figure 4.1 Data Flow Diagram

V. Results And Discussion

5.1 Introduction

This chapter presents the results of the Explainable Hate Speech Detection and Classification System. The system was evaluated for hate speech detection accuracy, explainability, and web application performance. By integrating the **BERT** model with **LIME** and **SHAP**, the system not only classifies text accurately but also provides clear explanations for its predictions, improving transparency and trust.

5.2 Text Preprocessing Results

The preprocessing module successfully cleaned and normalized user-generated text by removing URLs, special characters, and unwanted content. The processed text was prepared for BERT tokenization, resulting in more reliable and consistent classification.

Key Outcomes:

- Cleaned and normalized text
- Removed unwanted characters and URLs
- Validated user input
- Prepared text for classification

5.3 Hate Speech Classification Results

The BERT-based classification model accurately identified hate speech and non-hateful content by understanding contextual information rather than relying on keywords. The system generated confidence scores and supported real-time prediction.

Key Outcomes:

- Hate speech detection
- Normal content classification
- Context-aware predictions
- Confidence score generation

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

5.4 Lime Explanation Results

The LIME module successfully identified and highlighted the words that influenced the model's predictions. This improved the interpretability of the system by helping users understand why a particular text was classified as hateful or non-hateful.

Key Outcomes:

- Local explanation generation
- Important word highlighting
- Improved prediction transparency
- Better model interpretability

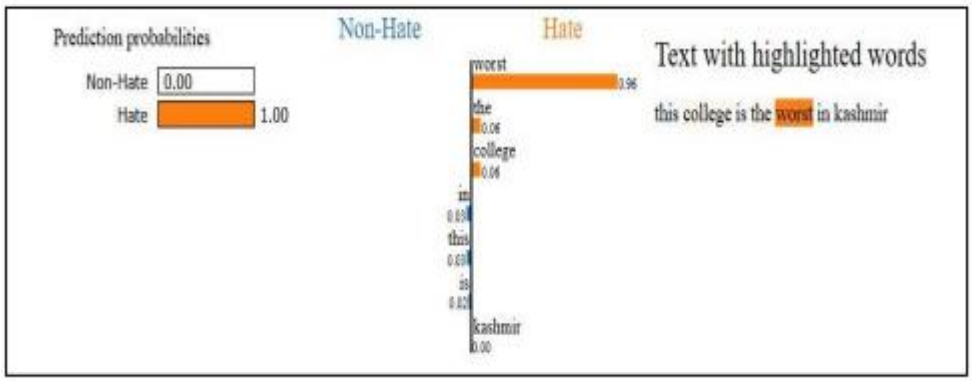

Figure 5.1 LIME Explanation Visualization

5.5 SHAP Explanation Results

The SHAP module was integrated to provide detailed explanations of the model's predictions by measuring the contribution of each word. It identified both positive and negative impacts on the classification outcome and generated clear explanation visualizations.

Key Outcomes:

- Feature contribution analysis
- Positive and negative word impact identification
- Detailed prediction explanations
- Improved model transparency
- Explainability visualization

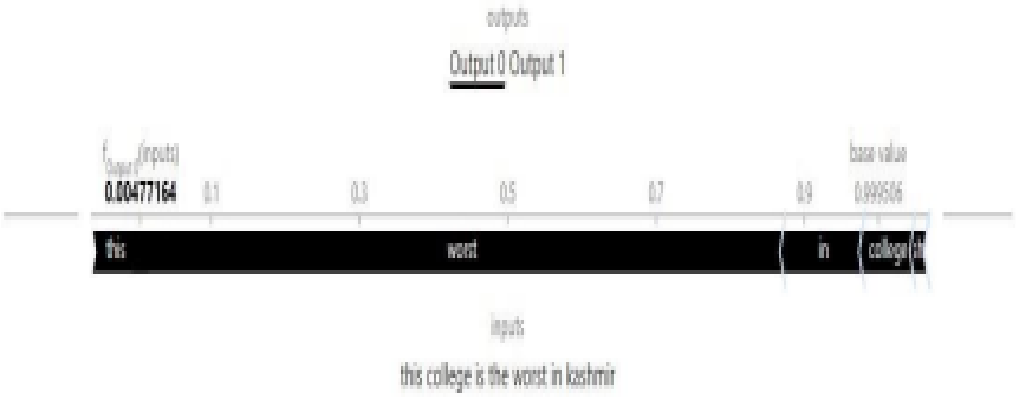

Figure 5.2 SHAP Explanation

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

5.6 Web Application Results

The web application was successfully integrated with the classification and explainability modules, allowing users to analyze text through a simple interface. Using **FastAPI**, the system provided real-time predictions along with **LIME** and **SHAP** explanations.

Key Outcomes:

- User text input
- Real-time prediction
- Classification results with confidence scores
- LIME and SHAP visualizations
- Error handling and input validation
- Responsive and user-friendly interface

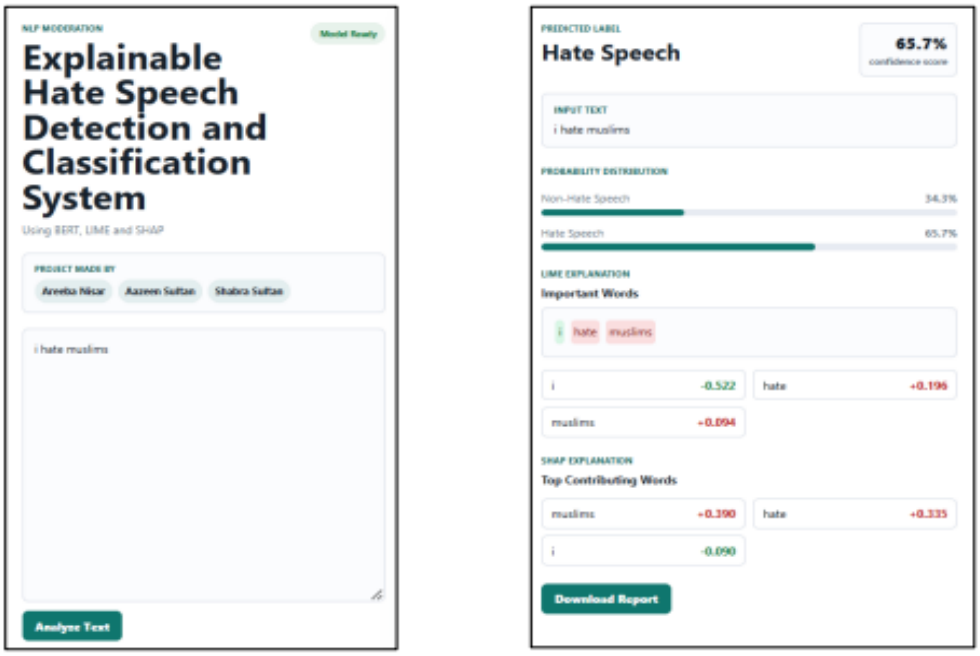

Figure 5.3. Web Application Interface Prediction Dashboard

5.7 Overall System Performance And Discussion

The proposed Explainable Hate Speech Detection and Classification System successfully integrated **BERT**, **LIME**, **SHAP**, and **FastAPI** into a unified platform. All modules, including preprocessing, classification, explainability, and the web interface, functioned effectively and worked together seamlessly.

A key strength of the system is its ability to provide transparent predictions through explainable AI, enabling users to understand the reasons behind each classification. The user-friendly interface further improves accessibility and ease of use.

The modular architecture supports easy maintenance and future enhancements. However, the system's performance depends on the quality of the training data and may face challenges with sarcasm, implicit hate speech, and highly context-dependent content. Additionally, LIME and SHAP increase computational overhead.

Overall, the results demonstrate that the proposed system achieves accurate hate speech detection while improving transparency, interpretability, and user trust, making it suitable for responsible content moderation.

VI. Conclusion, Limitations And Future Scope

6.1 Conclusion

This project developed an Explainable Hate Speech Detection and Classification System using BERT with LIME and SHAP to accurately detect hate speech while providing transparent explanations for its predictions. The system includes text preprocessing, classification, explanation generation, and a web-based interface for user interaction. By combining NLP with Explainable AI, the proposed system improves prediction transparency, user trust, and supports more effective and responsible content moderation.

6.2 Limitations

- Developed as a prototype and not deployed at production scale.
- Performance depends on the quality and diversity of training data.
- Limited ability to detect sarcasm, implicit, or context-dependent hate speech.
- Supports only English text.
- Explainability methods increase computational cost.
- Does not support image, audio, or video content.
- Large-scale real-time deployment requires further optimization.

6.3 Future Scope

Future enhancements may include:

- Support for multilingual hate speech detection.
- Real-time integration with social media platforms.
- Improved detection of sarcasm and implicit hate speech.
- Multi-class toxicity classification.
- Enhanced explainability techniques.
- Mobile application development.
- Image and multimedia hate speech detection.
- Cloud-based deployment for scalability.
- Continuous model learning and personalized moderation policies.

REFERENCES

[1] **FastAPI Documentation.** (2025). *FastAPI: Modern, Fast Web Framework for Building APIs.* Available: https://fastapi.tiangolo.com

[2] **GitHub Documentation.** (2025). *Version Control and Collaborative Software Development.* Available: https://github.com

[3] **Google Colaboratory Documentation.** (2025). *Cloud-Based Machine Learning Development Environment.* Available: https://colab.research.google.com

[4] **Hugging Face Documentation.** (2025). *Transformers Library Documentation.* Available: https://huggingface.co/docs

[5] Brownlee, J. (2022). *Machine Learning Mastery.* Machine Learning Mastery.

[6] Hao, J., & Ho, T. K. (2019). Machine learning made easy: A review of the scikit-learn package in Python programming language. *Journal of Educational and Behavioral Statistics, 44*(3), 348–361.

[7] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 4171–4186).

[8] Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems, 30.*

[9] Ashish, V. (2017). *Attention is All You Need. Advances in Neural Information Processing Systems, 30.*

[10] Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media, 11*(1), 512–515.

[11] Goldberg, Y. (2016). A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research, 57*, 345–420.

[12] Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep Learning* (Vol. 1). Cambridge, MA: MIT Press.

[13] Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit.* O'Reilly Media.