IntelliMed : A Context-Aware Drug Review Analysis Using BERT For Safer Healthcare Decisions

P Vijaya Kumari [1],Thallapelly Amulya[2], Vennapureddy Renuka[3], Rekireddy Ashrutha[4], Rachakatla Bhaskar[5]

[1] Assistant Professor, *Department of CSE (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING) TKR COLLEGE OF ENGINEERING & TECHNOLOGY*

[2,3,4,5] *UG Scholars in Department of CSE (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING) TKR COLLEGE OF ENGINEERING & TECHNOLOGY*

ABSTRACT: IntelliMed is a Django-based medicine recommendation system that analyzes user written reviews using sentiment analysis to classify opinions as positive, negative, or neutral, helping generate trustworthy medicine suggestions supported by real user feedback. The Admin manages all medicine details including symptoms, side effects, disease categories, brands, and images Additionally, the Admin can upload and manage review datasets for sentiment analysis while Users can create accounts, update health profiles, add and review medicines, and view sentiment-based insights shown through pie-chart distributions. The system includes a custom sentiment analysis model trained from scratch and compares algorithms such as BERT, Random Forest, VADER, and TextBlob to identify the best performer. It also offers disease-based medicine suggestions with complete details, and a Gemini-powered chatbot provides quick guidance on medicines, symptoms, and diseases. By converting unstructured medical reviews into clear sentiment categories, IntelliMed helps users make informed decisions and reduces confusion caused by conflicting online medical information.

Keywords: Medicine recommendation system, Sentiment analysis, Django application, BERT, Random Forest, VADER, TextBlob, Chatbot

1. INTRODUCTION

Modern healthcare platforms generate a large amount of user-written content in the form of reviews, feedback, and experience notes. People share how effective a medicine was, what side effects they felt, and whether they would recommend it to others. However, most users struggle to interpret these reviews because the text is unstructured and varies from person to person. Sentiment analysis provides a method to automatically understand these opinions by classifying them into positive, negative, or neutral categories.

Healthcare decisions are sensitive, and incorrect information can lead to poor choices. Many users rely on online reviews before choosing a medicine, but manual reading becomes difficult when reviews are long or contradictory. This project addresses this problem by transforming raw text into meaningful sentiment categories. The idea is to bring clarity and structure to the large amount of informal medical text available online.

IntelliMed includes a complete sentiment analysis pipeline designed to train a model from scratch. The process starts with downloading a dataset of medicine reviews. The dataset is cleaned by handling null values, removing duplicate entries, and preparing the text for machine learning. Tokenization and vectorization techniques convert the text into numerical form, which is required for model training.

Along with the custom-trained model, the system also integrates VADER and TextBlob. These tools offer quick insights and allow comparison between different approaches. The project also compares BERT and Random Forest to identify which algorithm performs better for medical review classification. This

comparison helps in selecting the best model for final deployment in the web application.

The Django web application forms the practical interface through which users interact with the system. The Admin manages the medicine database by adding details such as medicine name, disease, symptoms, side effects, brand, and images. Additionally, the Admin can upload and manage datasets consisting of user medicine reviews, which are 2 used for sentiment analysis. Users create accounts, update their health profile, manage diseases, maintain a personal list of medicines, and also provide reviews and feedback on medicines they have used. When a user enters a disease name, the system displays the matching medicine information along with sentiment results visualized through a pie chart.

1.1 MOTIVATION

The motivation behind developing the IntelliMed system arises from the growing need for intelligent, reliable, and user-friendly digital healthcare solutions. People searching for medicines online often encounter long, unstructured reviews that do not clearly reveal whether the medicine is effective or safe. With the increasing availability of unprocessed medical review data, there is a strong need for sentiment analysis that can automatically classify reviews into positive, negative, and neutral categories. Building such a model helps utilize valuable information about side effects, improvements, and user satisfaction.

Another major motivation is to enhance personalization and system efficiency in healthcare platforms. Many existing websites lack features like disease tracking, tailored medicine suggestions, and visual sentiment insights. IntelliMed aims to fill this gap by offering user profiles, condition-wise medicine recommendations, order tracking, and an admin-controlled backend for maintaining accurate medicine data. Additionally, introducing a chatbot powered by the Gemini API improves user engagement by providing instant support.

1.2 PROBLEM DEFINITION

In today's digital healthcare environment, patients often rely on scattered and unreliable online sources to choose medicines for specific diseases. There is no single platform that combines structured medical information (such as symptoms, side effects, and brands) with real user experiences and sentiment insights. As a result, users face difficulty in selecting the most suitable medicine, understanding its effectiveness, and trusting the available information. Additionally, existing systems do not effectively utilize user-generated reviews to provide meaningful insights through advanced techniques like sentiment analysis. This lack of integration between medical data and user feedback leads to poor decision-making, misinformation, and reduced confidence among users. Therefore, there is a need for an intelligent system that integrates medicine data with user reviews, analyzes sentiments, and presents the results in an understandable format to help users make informed healthcare decisions.

2. PROPOSED SYSTEM

IntelliMed is an artificial intelligence-based web application designed to provide intelligent medicine recommendations and insights by integrating medical data with user-generated reviews. The system is developed using Django as a responsive web platform, where the admin manages medicine details including disease, symptoms, side effects, brand information, images, and also uploads datasets consisting of user medicine reviews. These datasets are used to perform sentiment analysis and enhance the reliability of the system. 4 This The proposed system offers a user-friendly interface where users can create accounts, manage their health profiles, add diseases, maintain a personalized list of medicines, and also provide reviews based on their experiences. When a user searches for a disease, the system retrieves

relevant medicines and displays detailed information along with sentiment analysis results visualized using pie charts (positive, negative, neutral). This helps users quickly understand public opinion and effectiveness of medicines.

Furthermore, the system leverages machine learning models, including transformer-based techniques, to accurately analyze user sentiments and improve recommendations over time. The integration of structured medical data with real-time user feedback ensures better decision-making. The application is scalable, efficient, and designed to provide a reliable and interactive healthcare support system. In addition, the system supports continuous data updates and improvement through admin-managed datasets and user-generated reviews, ensuring that the information remains current and relevant. The use of visualization techniques and interactive features enhances user engagement and simplifies complex data interpretation. This makes IntelliMed a practical and effective solution for assisting users in making informed healthcare decisions.

3. Sequence Diagram

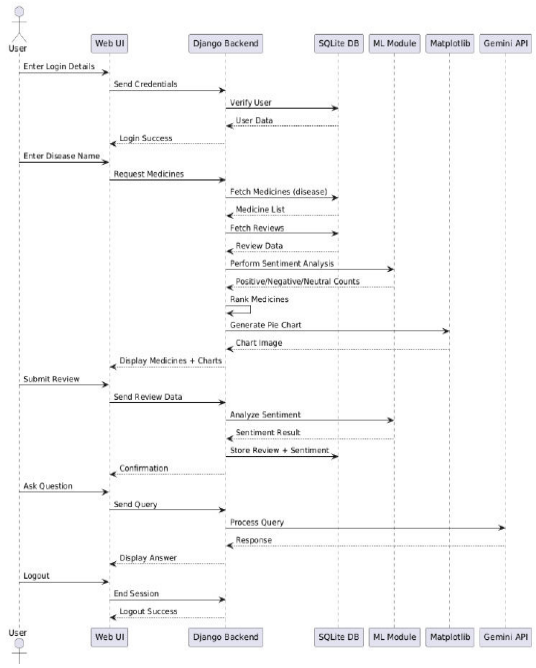


Fig.1 Sequence Diagram

4. ALGORITHMS

4.1.1 Sentiment Analysis

Sentiment analysis is the core component of the IntelliMed system, used to automatically classify user-generated medicine reviews into positive, negative, or neutral categories. The process begins by retrieving review text from the database, followed by preprocessing steps such as removing unwanted characters and cleaning the text. The cleaned text is then passed through a sentiment analysis model, which analyzes the context and determines the polarity of the review. The predicted sentiment is stored in the database and later used for generating insights and recommendations. This approach helps convert unstructured medical reviews into meaningful information for better decision-making.

4.1.2 Dataset Processing and Data Handling

The system allows the admin to upload datasets in CSV format containing medicine reviews. Once uploaded, the dataset is validated to ensure required fields are present. Each row is processed by extracting relevant attributes such as medicine name, disease, review text, rating, and date. The system then applies sentiment analysis to each review before storing the data in the database. This structured storage enables efficient querying, analysis, and retrieval of information for further processing within the system.

4.1.3 Medicine Recommendation Algorithm

The medicine recommendation process is based on disease input and sentimentdriven ranking. When a user enters a disease name, the system retrieves all relevant medicines from the database. For each medicine, associated reviews are analyzed to calculate the number of positive, negative, and neutral sentiments. A sentiment score or percentage is computed, and medicines are ranked based on the highest positive sentiment. This ensures that medicines with better user feedback are displayed first,

helping users make informed healthcare decisions.

4.1.4 Review Processing

What The system enables users to submit reviews for medicines, which are immediately processed and analyzed. Once a review is entered, it is validated and stored in the database. The sentiment analysis module is then applied to classify the review. The result is saved along with the review, ensuring that the overall sentiment data remains updated in real-time. This continuous update mechanism improves the accuracy of recommendations over time

4.1.5 Visualization Algorithm

To make sentiment results easily understandable, the system uses a visualization algorithm to generate pie charts. The system retrieves sentiment counts (positive, negative, neutral) for each medicine and uses a visualization library to create a pie chart. The chart is saved as an image and displayed on the user interface. This graphical representation helps users quickly interpret the overall effectiveness of a medicine.

4.1.6 Chatbot Interaction Algorithm

The chatbot module provides interactive support to users by answering queries related to medicines and diseases. When a user enters a query, the system processes the input and detects the language. The query is then sent to an AI-based chatbot model, which generates an appropriate response. The response is returned to the user through the interface, enhancing user experience and accessibility of information.

5. SYSTEM ARCHITECTURE

The proposed IntelliMed system follows a multi-layered architecture to efficiently process user inputs and perform sentiment analysis on drug reviews. The system begins with the **Client Layer**, where users interact with the application through a web browser by searching medicines, submitting reviews, and viewing results.

The requests are then passed to the **Presentation Layer**, which handles user interface components such as forms, search functionality, and visualization charts.

The **Application Layer (Django)** manages the core business logic of the system, including handling medicine data, processing user reviews, API routing, and request-response operations. This layer connects the frontend interface with backend processing and ensures smooth data flow across the system.

The **ML Inference Layer** is responsible for analyzing user reviews and predicting sentiment using multiple models such as VADER, TextBlob, Random Forest, and BERT. These models classify reviews into positive, negative, and neutral categories, providing accurate insights into user opinions.

To improve performance and handle large data efficiently, the system uses **Async Workers (Celery)** for background processing tasks such as batch sentiment scoring and asynchronous execution. This reduces system load and improves response time.

The processed data is stored in the **Database Layer**, which maintains structured information such as user data, medicine details, and review records. The system supports both **SQLite and PostgreSQL** databases. SQLite is used for lightweight development and testing due to its simplicity, while PostgreSQL is used for scalable and production-level deployment, ensuring efficient data management and performance.

The system also includes a **Storage Layer (Local/S3)** to store datasets, model files, and other resources required for processing. In addition, the system integrates external services through the **External Integration Layer**, which connects with the Gemini API chatbot. This allows users to interact with the system, ask queries about medicines, and

receive intelligent responses, thereby enhancing user experience and functionality.

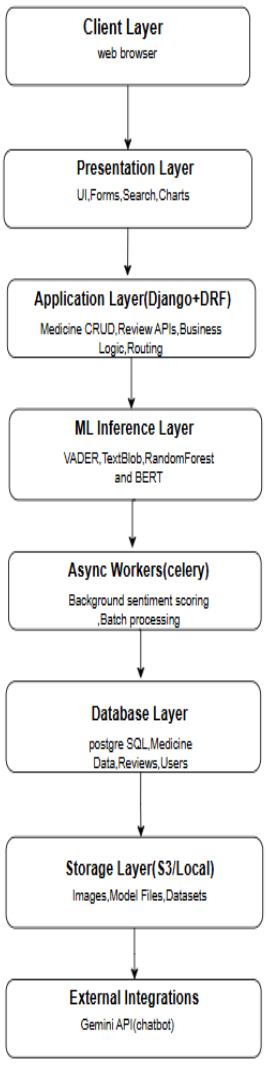


Fig.1 System Architecture

6 IMPLEMENTATION AND RESULTS

6.1 EXPLANATION OF KEY FUNCTIONS

The IntelliMed system is built using multiple integrated functions that work together to provide medicine recommendations based on user reviews and sentiment analysis. Each function plays a vital role in delivering accurate, user -friendly, and datadriven results.

6.1.1 User Authentication Function

The system provides secure user registration and login functionality. During registration, user details such as username, email, and password are validated and stored in the database. During login, the system verifies credentials and grants access to either the user dashboard or admin dashboard based on the role. This ensures secure access and personalized interaction with the system.

6.1.2 Admin Dataset Upload Function

The admin can upload datasets containing medicine reviews in CSV format. The system reads the dataset using Python, extracts fields such as medicine name, disease, and reviews, and stores them in the database. This function helps in populating large amounts of real-world data for analysis and improves recommendation accuracy.

6.1.3 Sentiment Analysis Function

The system entiment analysis function processes user reviews to determine their sentiment as positive, negative, or neutral. The system preprocesses the text by removing unwanted characters and applies classification logic to identify sentiment. The results are stored in the database and used for ranking medicines.

6.1.4 Medicine Recommendation Function

This is the core function of the system.When a user enters a disease name, the system retrieves all related medicines from the database.It then analyzes associated reviews and calculates sentiment counts.Medicines are ranked based on the number of positive reviews and displayed in descending order, helping users choose the most effective medicine.

6.1.5 Review Submission Function

Users can submit reviews for medicines through the system. The input is validated and stored in the database. Immediately after submission, the sentiment analysis function is applied to the review, and the result is updated

dynamically. This ensures that the system continuously learns from user feedback.

6.1.6 Visualization Function

The system generates visual representations of sentiment analysis using pie charts. It calculates the number of positive, negative, and neutral reviews and displays them graphically using Matplotlib. This helps users quickly understand the effectiveness of each medicine.

6.1.7 Chatbot Assistance Function

The chatbot module allows users to ask queries related to medicines and symptoms. The system processes user input and generates appropriate responses, providing quick guidance and improving user experience.

6.1.8 User Personalization Function

The system allows users to save diseases and medicines to their personal dashboard. This enables easy access to previously searched data and improves usability through personalized recommendations.

6.2 METHOD OF IMPLEMENTATION

The system is implemented using a web-based architecture with frontend, backend, and database components.

Frontend (User Interface)

• Developed using HTML, CSS, and JavaScript

• Provides interactive pages for users and admins

• Includes dashboards, forms, and visualization outputs

Backend (Django Framework)

• Handles business logic and request processing

• Manages user authentication and data handling

• Integrates sentiment analysis and recommendation logic

Database

• SQLite database is used for storing users, medicines, diseases, and reviews

• Ensures structured and efficient data retrieval Visualization Module

• Uses Matplotlib to generate pie charts for sentiment analysis

• Displays graphical insights on the result page

Deployment

• The system runs on a local server (127.0.0.1:8000) • Accessible through a web browser

6.3 FORMS

The system includes the following input forms:

1. User Registration Form Collects username, email, and password. Validates input and stores user details securely.

2. User Login Form Allows users to log in using credentials and redirects them to the dashboard.

3. Disease Search Form Enables users to enter disease names and fetch medicine recommendations.

4. Review Submission Form Allows users to submit medicine reviews, which are analyzed for sentiment.

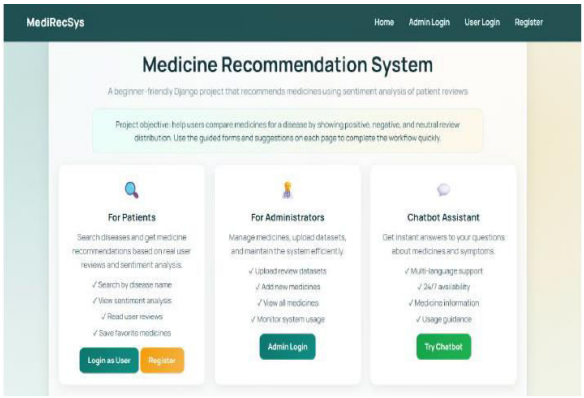
5. Admin Dataset Upload Form Allows admin to upload CSV files containing medicine review datasets

6.4 OUTPUT SCREENS

The system generates the following output screens:

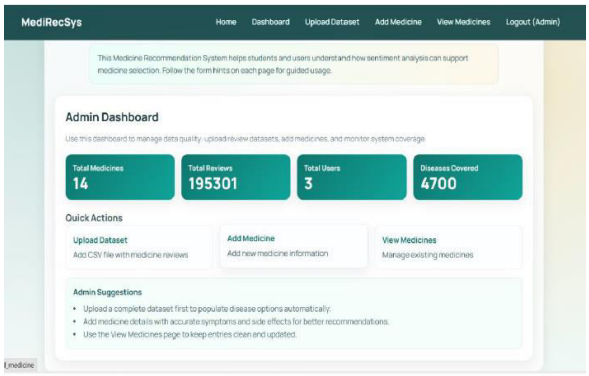
1. Home Page

Displays system overview and navigation options for users and administrators.



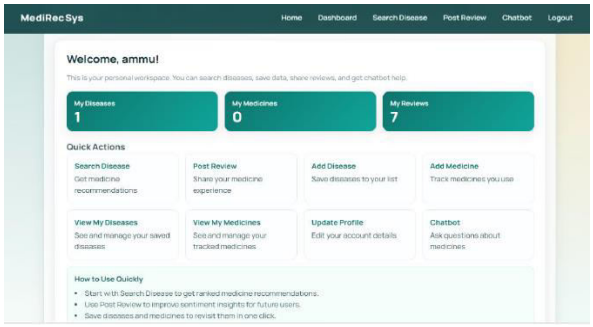
2. Admin Dashboard

Shows system statistics such as total medicines, reviews, users, and dataset management Options



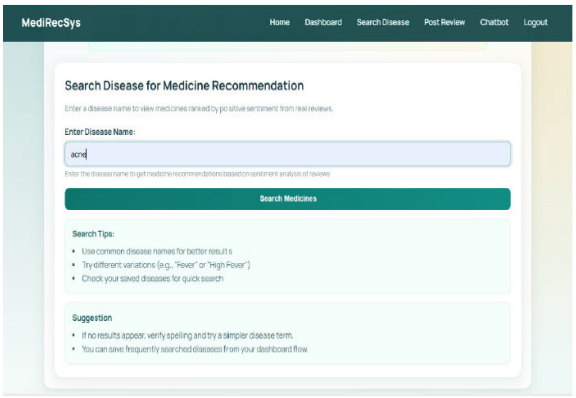
3. User Dashboard

Provides personalized data including saved diseases, medicines, and reviews



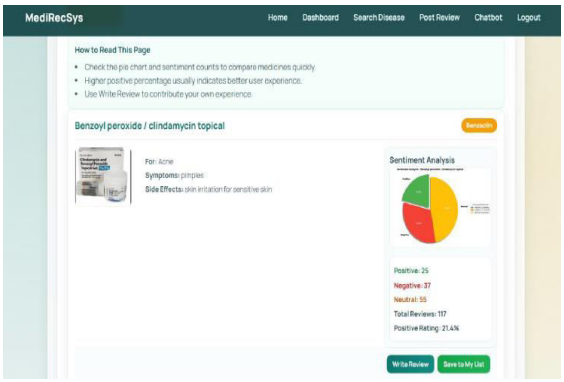
4. Disease Search Page

Allows users to input disease names and initiate search.



5. Recommendation Result Page

Displays medicine details along with sentiment analysis using pie charts and review statistics. Medicines are ranked based on positive sentiment percentage.



6.5 RESULT ANALYSIS

The result analysis evaluates the overall performance and effectiveness of the IntelliMed system in providing accurate medicine recommendations using sentiment analysis of user reviews. The system integrates dataset processing, sentiment classification, and visualization modules to deliver meaningful insights to users.

6.5.1. Medicine Recommendation Accuracy

The system retrieves medicines based on the disease entered by the user and ranks them using sentiment analysis results. Medicines with higher positive review counts are prioritized, ensuring that users receive recommendations based on real-world patient experiences. This approach improves the reliability of suggested medicines compared to simple keyword-based systems.

6.5.2. Sentiment Analysis Results

The sentiment analysis module classifies user reviews into three categories: Positive, Negative, and Neutral. The system processes large volumes of review data efficiently and generates accurate sentiment distributions. The results are stored and used for ranking medicines, providing a data-driven approach to decision-making.

6.5.3. Visualization Effectiveness

The system uses pie charts to visually represent sentiment distribution for each medicine. This graphical representation helps users quickly understand the overall perception of a medicine without reading all reviews. The visualization improves user experience and supports faster decision-making.

6.5.4. Dataset Handling Performance

The system efficiently processes large datasets uploaded by the admin, including thousands of medicine reviews. The dataset is parsed, stored, and analyzed without significant delay, ensuring smooth performance even with large-scale data. This enhances the scalability of the system.

6.5.5. Admin Dashboard Efficiency

The admin dashboard provides an overview of total medicines, reviews, users, and diseases covered. It allows administrators to upload datasets, add medicines, and manage system data effectively. The dashboard ensures easy monitoring and maintenance of the system.

6.5.6. User Interaction and Personalization

The system provides a personalized user dashboard where users can save diseases, medicines, and reviews. Features like review submission and chatbot assistance enhance user engagement and make the system interactive and user-friendly.

6.5.7. System Performance

The system is developed using Django for backend processing and SQLite for data storage, ensuring efficient data handling and quick response times.The integration of sentiment analysis and visualization modules provides real-time insights without affecting system performance.

6.5.8.Training and Validation

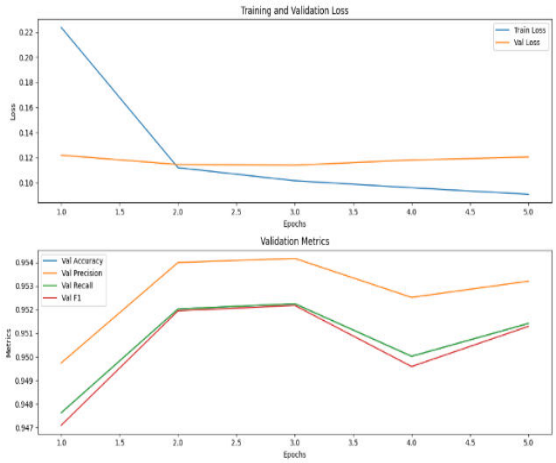


Fig.1 Training and validation

The model learns smoothly as training progresses, with stable validation results showing it isn't overfitting, and key metrics like accuracy, precision, recall, and F1-score. Using BERT helps the system better understand context, leading to more accurate and meaningful predictions.

Metrics :

- Accuracy:

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

Measures overall correctness of predictions.

- Precision:

$$Precision = \frac{TP}{TP+FP}$$

Measures overall correctness of positive predictions.

- Recall:

$$Recall = \frac{TP}{TP+FN}$$

Measures ability to capture all positive cases.

- F1 Score:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Balances precision and recall.

7. CONCLUSION

The IntelliMed system successfully demonstrates the integration of healthcare data analysis with intelligent web-based technology to provide meaningful medicine recommendations. The system utilizes sentiment analysis to process large volumes of patient reviews and classify them into positive, negative, and neutral categories. This approach ensures that medicine recommendations are based on real user experiences, making the results more reliable and practical. The inclusion of visual representations such as pie charts further enhances understanding by allowing users to quickly interpret the effectiveness of different medicines.

The system is designed with a simple yet efficient architecture that ensures smooth interaction between the frontend, backend, and database components. The admin module allows easy dataset management through CSV uploads, enabling continuous system improvement. On the user side, features such as disease search, personalized dashboards, review submission, and chatbot assistance provide a complete and interactive experience. These functionalities ensure that the system is not only technically sound but also userfriendly and accessible.

From a performance perspective, IntelliMed shows stable and efficient operation across all major modules. It is capable of handling large datasets of medicine reviews while maintaining quick response times for search and recommendation tasks. Proper validation, error handling, and structured data storage contribute to the system's reliability and robustness. The use of Django as the backend framework ensures scalability, maintainability, and secure data handling, making the system suitable for real-world deployment.

Overall, IntelliMed highlights the potential of applying artificial intelligence in the healthcare domain to support better decision-making. By transforming raw review data into useful insights, the system bridges the gap between complex medical information and user understanding. With future enhancements such as advanced machine learning models, realtime data integration, and personalized recommendations, the system can be further improved. This project serves as a strong foundation for developing intelligent, scalable, and user-centric healthcare applications.

8. REFERENCES

1. A. Y. Hassan and M. H. Ahmed, "Sentiment analysis techniques for online medicine reviews," International Journal of Data Science and Analytics, vol. 8, no. 3, pp. 210– 220, 2021.

2. B. Liu, Sentiment Analysis and Opinion Mining. Morgan & Claypool Publishers, 2012

3. H. Saif, M. Fernandez, Y. He, and H. Alani, "Evaluation of machine learning models for sentiment classification," in Proc. IEEE Int. Conf. on Advances in Social Networks Analysis and Mining (ASONAM), 2014, pp. 593–600.

4. C. Potts, "Sentiment analysis tutorial," Stanford University, 2011. [Online]. Available: https://nlp.stanford.edu

5. H. Hutto and E. Gilbert, "VADER: A parsimonious rule-based model for sentiment analysis," in Proc. Int. Conf. on Web and Social Media (ICWSM), 2014, pp. 216– 225.

6. S. Bird, E. Klein, and E. Loper, Natural Language Processing with Python. O'Reilly Media, 2009.

7. L. Zhang, L. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 8, no. 4, pp. 1–24, 2018.

8. J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep

bidirectional transformers for language understanding," in Proc. Conf. of the North American Chapter of the Association for Computational Linguistics (NAACL) , 2019, pp. 4171–4186.

9. L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

10. Django Software Foundation, "Django documentation," 2023. [Online]. Available: https://docs.djangoproject.com 57

11. Google AI, "Gemini API documentation," 2024. [Online]. Available: https://ai.google.dev

12. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

13. T. Mikolov et al., "Efficient estimation of word representations in vector space," arXiv preprint arXiv:1301.3781, 2013.

14. J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in Proc. Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.