

Smart View – An AI -Powered Visualization, Reviewing and Recommendation App

Sakshi.P. Bachhav¹, Harshad.D. Kumavat², Suraj.M. Javdhav³, Aniket.D. Shirsath⁴, Mrs.V.B. Kale⁵,
Prof. M. P. Bhandakkar⁶

^{1,2,3,4} Student, Department of Information Technology *Matoshri Aasarabai Institute of Technology and Research Centre, Eklahare, Nashik, MH 422105*

⁵Lecturer, Department of Information Technology *Matoshri Aasarabai Institute of Technology and Research Centre, Eklahare, Nashik, MH 422105*

⁶HOD, Department of Information Technology *Matoshri Aasarabai Institute of Technology and Research Centre, Eklahare, Nashik, MH 422105*

Abstract—Smart View is an intelligent app review and sentiment analysis system designed to help users and organizations understand public opinion about mobile applications. The system allows users to enter an application name and automatically fetches real-time reviews from the Google Play Store using web scraping techniques. By converting large volumes of unstructured textual reviews into meaningful insights, SmartView supports better decision-making for developers, marketers, and end users. The proposed system is developed using Python with Flask as the backend framework, while HTML, CSS, and Bootstrap are used for building an interactive and user-friendly interface. Multiple machine learning algorithms such as Logistic Regression, Linear Support Vector Machine, and Naïve Bayes are employed, along with a deep learning-based Bidirectional LSTM model, to accurately classify user reviews into positive, negative, or average sentiments. This hybrid approach improves the accuracy and robustness of sentiment classification. SmartView presents the analysed results through a dashboard that includes average app ratings, sentiment distribution graphs, and sample positive and negative reviews. These visualizations make it easy to interpret user feedback at a glance. Overall, the system provides a scalable and efficient solution for real-time app review analysis, enabling developers to enhance app quality based on genuine user sentiment.

Index Terms—Sentiment Analysis, App Reviews, Machine Learning, Deep Learning, Bidirectional LSTM, Web Scraping, Python, Flask, Dashboard Visualization

I. INTRODUCTION

In today's digital era, mobile applications play a crucial role in daily life, covering domains such as education, healthcare, finance, entertainment, and business. With the rapid growth of smartphone usage, app stores like Google Play Store receive millions of users reviews every day. These reviews contain valuable feedback that reflects user satisfaction, expectations, and problems. However, manually analysing such a massive amount of textual data is impractical and time-consuming. User reviews are often unstructured, informal, and expressed in natural language, making it difficult to directly extract meaningful insights. Despite this challenge, reviews are one of the most reliable sources of information for understanding user experience. Developers rely heavily on this feedback to improve app quality, fix bugs, and introduce new features. Hence, there is a strong need for an automated system that can efficiently analyse app reviews and summarize user sentiment. Sentiment analysis, a subfield of natural language processing (NLP), focuses on identifying and classifying opinions expressed in text as positive, negative, or neutral. By applying sentiment analysis techniques to app reviews, it becomes possible to understand overall user perception and emotional tone at scale. This approach transforms raw textual data into structured information that supports data-driven decision-making. With advancements in machine learning and deep learning, sentiment analysis has achieved higher accuracy and reliability.

Traditional machine learning algorithms such as Logistic Regression, Support Vector Machines, and Naïve Bayes have been widely used for text classification tasks. More recently, deep learning models like Long Short-Term Memory (LSTM) networks have shown superior performance by capturing contextual and sequential information in text data. Another major challenge in app review analysis is the real-time collection of data. Reviews are continuously updated on app stores, and static datasets may fail to reflect current user opinions. Web scraping techniques enable the automatic extraction of live reviews directly from platforms like the Google Play Store, ensuring that the analysis

remains up to date and relevant. The SmartView system is designed to address these challenges by integrating real-time data collection, advanced sentiment analysis models, and an interactive visualization dashboard. Users simply enter an app name, and the system fetches reviews, processes them, and analyses sentiment using both machine learning and deep learning approaches. This end-to-end automation reduces human effort and increases efficiency. To enhance usability, SmartView presents results through clear visual representations such as rating distributions, sentiment graphs, and categorized sample reviews.

II. LITERATURE SURVEY IN TABULAR FORMAT

Sr. No.	Author(s) & Year	Method / Technique Used	Dataset / Source	Key Findings
1	Pinto AP, Reshma KJ; Eser & Sahin (2024)	Transformer-based sentiment analysis models	App review datasets	Transformer models improve sentiment prediction accuracy and rating classification.
2	Islam & Alam (2024)	Skipbangla-Bert, feature metric model	Bangla Bangla dataset	Proposed Bangla-focused model achieved better performance than traditional ML approaches.
3	Rodriguez-Ibanez et al. (2023)	Review of sentiment analysis techniques	Social media platforms	Deep learning and hybrid approaches outperform classical NLP techniques.
4	Joshi, Rana, Gundotra (2023)	Machine Learning algorithms	Amazon reviews	ML models effectively classify customer sentiments for product feedback analysis.
5	Bhowmik et al. (2022)	Lexicon-based + Deep Learning	Bangla text datasets	Hybrid model improved sentiment classification accuracy for regional language data.

- Sharma and Mehta (2024) explored sentiment analysis of mobile application reviews using deep learning techniques, with a primary focus on LSTM networks. Their study demonstrated that LSTM models outperform traditional machine learning classifiers by effectively capturing sequential and contextual information present in user reviews.
- In a comparative study, Kumar, Singh, and Verma (2023) applied traditional machine learning algorithms such as Support Vector Machines, Naïve Bayes, and Logistic Regression to Google Play Store reviews.
- Although these models achieved reasonable classification accuracy, their performance was limited when handling complex sentence structures and sarcasm, emphasizing the need for more advanced NLP-based approaches.
- To overcome these limitations, Li et al. (2022) proposed a sentiment analysis framework using word embeddings combined with an LSTM-based deep learning model. Their approach significantly improved sentiment classification accuracy by preserving semantic meaning and

contextual flow within reviews. The study confirmed that deep learning models are more effective for real-world user feedback analysis.

5. Additionally, Patel and Shah (2021) analysed mobile application reviews using sentiment analysis and visualization techniques to support developer decision-making. Their work demonstrated that combining sentiment classification with interactive dashboards helps identify recurring issues and track user satisfaction trends over time.

III. EXISTING SYSTEM

The existing systems for app review analysis mainly rely on manual feedback reading or basic rating-based evaluation available on platforms like the Google Play Store and Apple App Store. Users typically depend on star ratings and a limited number of visible reviews to understand application quality. However, these approaches do not provide deep insights into user sentiment, emotions, or specific issues faced by users.

Traditional sentiment analysis systems use simple machine learning techniques such as Naïve Bayes, Support Vector Machines, and lexicon-based methods to classify text into positive, negative, or neutral categories. These systems usually work on pre-collected datasets rather than real-time data and often fail to handle sarcasm, context, and complex sentence structures. As a result, the accuracy of sentiment prediction remains limited.

Many existing tools are platform-dependent and do not automatically extract real-time reviews from app stores. They require manual data collection, preprocessing, and analysis, which is time-consuming and inefficient. Additionally, most systems do not provide interactive dashboards or visual summaries, making it difficult for developers and organizations to quickly interpret large volumes of feedback.

Furthermore, earlier approaches focus either on machine learning or deep learning individually, lacking a hybrid architecture. They also struggle with multilingual reviews, noise in textual data, and scalability when handling massive datasets. Therefore, there is a need for an intelligent, automated, and real-time sentiment analysis system that can accurately analyse app reviews and present insights in an easy-to-understand visual format.

IV. PROPOSED METHODOLOGY

The SmartView system follows a structured methodology to collect, process, and analyse app reviews for accurate sentiment prediction. The overall workflow includes data collection, preprocessing, model training, sentiment classification, and visualization of results.

1. Data Collection

The system collects real-time user reviews from the Google Play Store using web scraping techniques. Users enter the application name, and the system automatically extracts review text, ratings, and related metadata. This ensures that the analysis is based on live and up-to-date feedback.

2. Data Preprocessing

Collected reviews are unstructured and noisy. Therefore, preprocessing steps are applied such as:

- Removing stop words, special characters, and emojis
- Tokenization and normalization
- Lowercasing and stemming/lemmatization
- Converting text into numerical form using vectorization techniques

These steps improve data quality and model performance.

3. Feature Extraction

Text reviews are transformed into machine-understandable representations using techniques such as TF-IDF or word embeddings. This helps in identifying important words and contextual relationships within the reviews.

1. MODEL IMPLEMENTATION

The system uses a hybrid approach by integrating both machine learning and deep learning models:

- Logistic Regression
- Linear Support Vector Machine (SVM)
- Naïve Bayes
- Bidirectional LSTM (Deep Learning)

Machine learning models help in fast classification, while the Bilt model captures sequential patterns and contextual meaning in text, improving overall accuracy.

5. System Classification

Each review is classified into three sentiment categories:

- Positive
- Negative
- Average/Neutral

The hybrid model ensures better prediction by combining statistical learning and deep contextual analysis.

6. Result Analysis and Visualization

The classified results are displayed on a dashboard that includes:

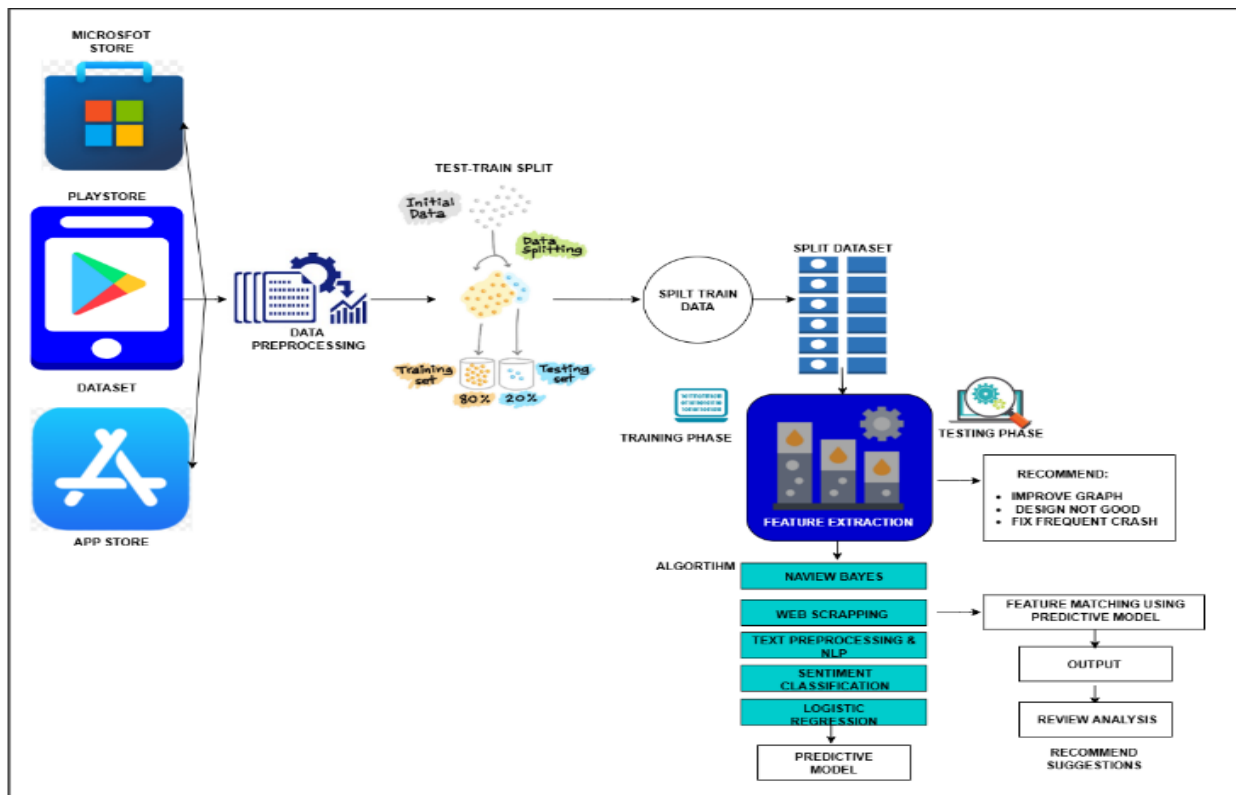
- Sentiment distribution graphs
- Average app ratings
- Sample positive and negative reviews

These visual insights help developers and organizations quickly understand public opinion about the application.

7. System Deployment

The system is developed using Python and Flask for backend processing, and HTML, CSS, and Bootstrap for the frontend interface. It is designed to be scalable, efficient, and user-friendly for real-time sentiment analysis of mobile app reviews.

V. SYSTEM ARCHITECTURE



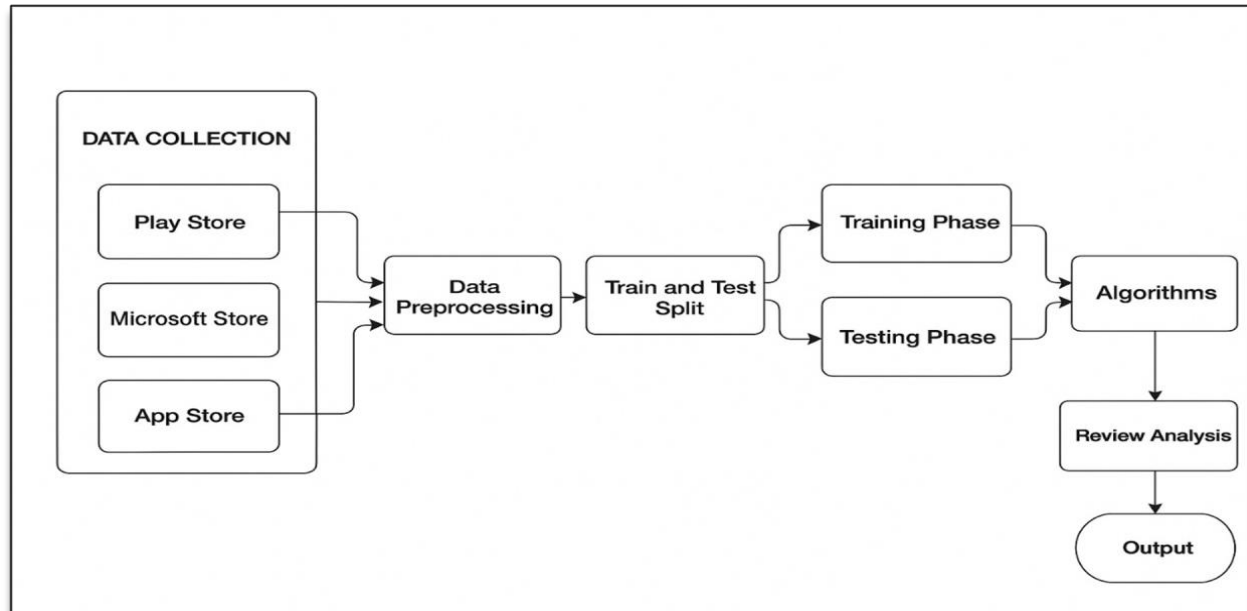
The proposed system, SMARTVIEW, is an AI-powered platform designed to automatically collect, analyze, visualize, and recommend improvements based on multi-platform app reviews. Unlike existing systems that are often limited to single platforms or basic sentiment classification, SMARTVIEW integrates reviews from the Google Play Store, Apple App Store, Microsoft Store, and other sources to create a comprehensive dataset. The system begins with a custom web scraping module that continuously

gathers user reviews, ratings, and feedback. These raw reviews undergo preprocessing using Natural Language Processing (NLP) techniques to remove noise such as emojis, stop words, and irrelevant symbols, followed by tokenization and normalization to prepare them for analysis. The processed data is then passed into the sentiment classification pipeline, where supervised machine learning algorithms like Logistic Regression, Naïve Bayes, and SVM, along with advanced models such as LSTM networks,

classify the feedback into positive, negative, or neutral sentiments. The outcomes are stored in a structured database and visualized through an interactive dashboard using bar charts, pie charts, trend graphs, and word clouds. A unique feature of the proposed system is its recommendation engine, which interprets aggregated sentiments and generates actionable suggestions for developers. For example,

if a majority of users report dissatisfaction with graphics or frequent crashes, the system will automatically recommend targeted improvements. Through this seamless workflow, the proposed system empowers developers to quickly understand user needs, address issues proactively, and deliver enhanced user experiences, ultimately improving app quality, ratings, and competitiveness in marketplace.

BLOCK DIAGRAM



Hardware Requirements for Development of Project

- Processor (CPU) A minimum of Intel Core i5 / AMD Ryen 5 processor or higher is recommended to efficiently handle data preprocessing, model training, and web application execution.
- Memory (RAM) At least 8 GB RAM is required for smooth multitasking, NLP preprocessing, and deep learning model training. 16 GB RAM is preferable for better performance.
- Storage
A minimum of 256 GB HDD/SSD is required to store datasets, trained models, libraries, and project files. An SSD is recommended for faster data access and system responsiveness.
- Graphics Processing Unit (GPU) – Optional A dedicated GPU (such as NVIDIA GPU with CUDA support) is optional but beneficial for faster training of the LSTM deep learning model.
- System Type A 64-bit system architecture is

required to support modern AI, machine learning, and deep learning frameworks.

- Peripheral Devices Standard peripherals such as a keyboard, mouse, and display monitor are required for development and testing of the web-based application.

Software Requirements for Development of Project

- Operating System A 64-bit Windows / Linux / macOS operating system is required to support AI, machine learning, and web development tools.
- Programming Language Python 3.x is used for data collection, NLP preprocessing, sentiment analysis, and deep learning model implementation.
- Development Environment / IDE Visual Studio Code / PyCharm / Jupiter Notebook is used for writing, testing, and debugging the project code.
- Web Scraping Library google_play_scraper is

used to extract real-time user reviews, ratings, and metadata from the Google Play Store.

- Machine Learning & Deep Learning Libraries Libraries such as TensorFlow / Keras are used to build and train the LSTM-based sentiment analysis model.
- Natural Language Processing Libraries NLTK / spacey are used for text preprocessing tasks such as tokenization, stop word removal, and lemmatization.
- Data Handling and Analysis Libraries Pandas and NumPy are used for data manipulation, cleaning, and numerical computations.
- Data Visualization Libraries Matplotlib / Seaborn / Polly is used to generate charts and graphs for sentiment visualization.
- Web Framework Flask / Django is used to develop the web-based application and integrate the backend with the frontend dashboard.
- Database (Optional) MySQL / SQLite can be used to store collected reviews and sentiment analysis results.

IV. CONCLUSION

The SmartView App Review & Sentiment Analysis system successfully demonstrates the effective use of machine learning and deep learning techniques to analyse large volumes of mobile application reviews. By automating the process of collecting, preprocessing, and analysing user feedback, the system transforms unstructured textual data into meaningful insights that help in understanding overall user sentiment. This approach significantly reduces manual effort and improves the efficiency of review analysis. The system provides real-time sentiment classification and presents results through an intuitive and visually appealing dashboard. By categorizing reviews into positive, negative, and average sentiments, along with rating analysis and graphical representations, SmartView enables developers, businesses, and users to quickly assess application performance and user satisfaction. The integration of multiple algorithms enhances accuracy and reliability, making the system robust and practical. In conclusion, SmartView serves as a powerful and scalable solution for app review analysis in the rapidly growing mobile application

ecosystem. Despite certain limitations, the system lays a strong foundation for future enhancements such as multilingual support and advanced NLP models. Overall, the project highlights the potential of sentiment analysis as a valuable tool for improving application quality, customer experience, and informed decision-making.

REFERENCES

- [1] Pinto AP, Reshma KJ. Impact of Project-Based Learning on Entrepreneurial and Social Skills G. Eser and C. Sahin, "Sentiment analysis and rating prediction for app reviews using transformer-based models," 2024.
- [2] M. S. Islam and K. M. Alam, "Sentiment analysis of Bangla language using a new comprehensive dataset bangdas and the novel feature metric skipbangla-bert," *Natural Language Processing Journal*, vol. 7, p. 100069, 2024.
- [3] M. Rodriguez-Ibanez, A. Casanez-Ventura, F. Castejon-Mateos, and P.-M. Cuenca-Jimenez, "A review on sentiment analysis from social media platforms," *Expert Systems with Applications*, vol. 223, p. 119862, 2023.
- [4] Joshi, A., Rana, V., Gandotra, E. (2023). Sentiment Analysis on Amazon Reviews using Machine Learning.
- [5] N. R. Bhowmik, M. Arifuzzaman, and M. R. H. Mondal, "Sentiment analysis on Bangla text using extended lexicon dictionary and deep learning algorithms," *Array*, vol. 13, p. 100123, 2022.