International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

ENHANCED USER SENTIMENT PREDICTION FROM MULTILINGUAL MOBILE APP REVIEWS USING ENSEMBLE LEARNING

DR. K. ANURADHA[1], GADDAM DEEPIKA[2], B JAGADEESH[3], DUMPALA SHREYA[4], ADLA RAJU[5], CHINNA SAMA SAMPATH REDDY[6]

PROFESSOR[1], UG SCHOLAR[2,3,4,5&6]

DEPARTMENT OF CSE, NARSIMHA REDDY ENGINEERING COLLEGE (UGC- AUTONOMOUS) MAISAMMAGUDA (V), KOMPALLY, SECUNDERABAD, TELANGANA-500100

ABSTRACT Mobile applications generate a vast amount of multilingual user reviews, which provide valuable insights into user satisfaction and product improvement. Current statistics indicate that over 80% of smartphone users rely on reviews before installation, while approximately 75% of developers use app store feedback to guide updates. However, manual sentiment classification of such multilingual reviews is highly time-consuming and error-prone, while existing single-model approaches often fail to capture contextual nuances across diverse languages. The proposed methodology leverages Natural Language Processing (NLP) on multilingual mobile review datasets with advanced preprocessing and feature extraction techniques to standardize text data across multiple languages. Exploratory Data Analysis (EDA) is performed to identify sentiment distribution and user behavior patterns. For Sentiment Classification Analysis, four classifiers are employed: Logistic Regression (linear classification), Ridge Classifier (regularized classification), and an Ensemble Classifier that combines predictions using a voting mechanism for improved accuracy. Parallelly, Rating Prediction Analysis is conducted using regression models, including Linear Regression (basic model), Ridge Regression (regularized regression), and an Ensemble Regressor that integrates results for robust numerical predictions on a 1–5 scale. Finally, a Universal Real-time Prediction framework is introduced, where a user can input a review in *any language* and instantly obtain both the sentiment polarity (positive, negative, neutral) and predicted star rating (1–5). This dual-layer prediction not only enhances user sentiment tracking but also optimizes developer feedback mechanisms, enabling faster updates and targeted improvements. The proposed system demonstrates significant advancement in multilingual adaptability, scalability, and real-time prediction capabilities.

1.INTRODUCTION

1.1 Overview

The rapid growth of smartphones and affordable internet access has significantly increased mobile application usage across the world. In India, the number of smartphone users has crossed 750 million, making the country one of the largest consumers of mobile applications. Platforms such as the Google Play Store host millions of applications and receive an enormous volume of user reviews every day. These reviews provide valuable insights into application quality, usability, performance, and user satisfaction. However, the diversity of languages used by Indian users creates additional challenges in understanding and analyzing feedback effectively. Automated sentiment analysis and rating prediction systems have become essential tools for handling large-scale, multilingual user-generated content. Sentiment analysis systems automatically identify emotions expressed in user reviews, while rating prediction systems estimate numerical scores based on textual feedback. These technologies support developers in improving application quality and enhance user experience across digital platforms. With the increasing popularity of online platforms and social media in our daily lives, there is an information boom all over the internet. Hence, there is ample data on every topic, ranging from products, businesses, market trends, etc. Correspondingly, the opinions of the users of respective domains are also freely available in plenty. Be it movie reviews, product reviews, financial market sentiments, political opinions, etc., there is easy access to such data for anyone seeking to take an informed decision. In such a context, it becomes necessary as well as useful to have a mechanism that sifts through the mass of data, analyzes them and preferably categorizes, quantifies or scores them, in order to aid decision making. This has given rise to the concept of sentiment analysis, also known as opinion mining. Sentiment analysis is the process of analyzing opinions or views expressed in documents and their overall classification, scoring or quantification. The primary purpose is to get an idea of people's general attitude and feelings towards a certain subject [1, 2]. Sentiment analysis mostly deals with a huge amount of unstructured and unlabelled data. Besides, the available data is generally subjective, vague, and not strictly adherent to language rules. Hence, sentiment analysis becomes a complex task, and involves knowledge of various domains, including natural language processing, data mining, machine learning, data analytics, computational intelligence, etc. At the very basic level, sentiment analysis is a classification problem, which categorizes opinions into broad categories like positive, negative or neutral. But in-depth analysis can lead to exploring finer details and extracting a larger amount of useful information [2, 3]. A lot of businesses need user feedbacks for decision making. They collect it using opinion polls, customer surveys, questionnaires, etc. With

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

the widespread availability of the internet, such feedback collection methods are able to reach a broader range of consumers, who are able to provide their honest and unbiased opinions. The entire process is easier and takes minimal time. Hence, it helps businesses, service providers, e-commerce organizations, governments, etc. collect varied opinions and use them to aid decision-making processes [4]. Owing to the huge amount of text available online that expresses the views of users on common forums, manual processing and analysis of the text is cumbersome and time consuming. Automated Sentiment Analysis techniques help save manpower, get faster output, sift through massive unnecessary data to find relevant material, and present the results in necessary formats. Sentiment Analysis tools are greatly useful for extracting data from various sources, viz. review sites, feedback forums, social networking sites, blogs, and so on, and performing detailed analytical operations on it [4, 5]. Even though research on sentiment analysis has seen a lot of milestones, most of the work is performed on English data. In comparison, very less research has been performed on languages other than English. It is more critical to analyze non-English data and there are multiple challenges to this task. Mechanisms that work on other languages either rely on their own limited resources, or prefer translation to English and using the abundantly available English resources. This task, known as multi-lingual sentiment analysis, is still an open area of research with a considerable scope for improvement

1.2 Problem Definition

Before the adoption of machine learning techniques, app review analysis depended on manual inspection and basic keyword matching methods. These approaches were inefficient due to the massive volume of reviews generated daily. Manual analysis suffered from inconsistency, bias, and high time consumption. Traditional systems were unable to handle multilingual reviews effectively. They also failed to understand contextual meaning and could not accurately predict user ratings from text.

1.3 Research Motivation

The increasing scale of user-generated content motivates the need for intelligent systems that can analyze reviews accurately and efficiently. Developers require automated tools to interpret customer sentiment and predict satisfaction levels. The presence of multiple Indian languages in app reviews increases the complexity of analysis. Machine learning and AI techniques provide a reliable solution to these challenges. This research focuses on transforming raw user feedback into meaningful insights for better decision-making.

1.4 Objective of the System

The objective of the system is to design an intelligent framework capable of analyzing multilingual mobile app reviews. The system aims to classify user sentiment into positive, negative, and neutral categories. Another objective is to predict numerical ratings based on review text. The system also integrates AI-based real-time prediction with machine learning models. Reliability and user-friendly interaction are central goals of the system.

1.5 Applications

The system is applicable in mobile application development for understanding customer feedback and improving product quality. App store platforms can use it to analyze user sentiment trends automatically. Business analysts benefit from insights into customer satisfaction levels. Marketing teams can evaluate product reception using predicted ratings. Customer support teams can identify frequent issues quickly. Educational institutions can use the system for research in natural language processing. Public service and government applications can monitor user feedback effectively. Enterprises can improve decision-making through data-driven insights.

1.6 Significance

The significance of this system lies in its ability to automate the analysis of large-scale multilingual app reviews. It reduces manual effort and improves accuracy in sentiment classification and rating prediction. The system enhances understanding of user behavior and expectations. It supports faster decision-making for application improvement. The integration of machine learning and AI increases reliability and scalability. This approach contributes to improved user satisfaction and better digital services.

2.LITERATURE SURVEY

Noteworthy early research in multilingual and cross-lingual sentiment analysis includes the NTCIR6 pilot project conducted in Japan in 2007, which developed an annotated corpus in Chinese and Japanese for research purposes [14]. This corpus was later utilized in NTCIR7 in 2008, where more detailed subtasks were introduced, such as opinion holder extraction, polarity identification, and sentence- and clause-level annotation. In addition, Simplified Chinese was introduced as a new target language in this phase [15]. During the same period, Boiy and Moens [16] explored supervised machine learning techniques to minimize the number of annotated samples required for accurate sentiment prediction in French and Dutch. This study is regarded as one of the earliest applications of machine learning for sentiment analysis in non-English languages. Concurrently, Wan [17] conducted experiments in which Chinese product reviews

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

were translated into English to leverage the extensive sentiment analysis resources available in English. The results demonstrated that this translation-based approach outperformed direct sentiment analysis on Chinese reviews, which suffered from limited data availability. Machine translation was employed, and a hybrid approach was also evaluated, yielding highly effective results. Similarly, Denecke [18] analyzed German product reviews using the widely adopted SentiWordNet resource on translated texts, demonstrating the feasibility of this method. This research laid the foundation for subsequent studies in multilingual and cross-lingual sentiment analysis. In 2009, Wan [19] extended this work on Chinese text by proposing a co-training approach that combined labeled English data translated into Chinese with unlabeled Chinese data translated into English. This dual-training strategy achieved better performance compared to conventional classifiers. Brooke et al. [20], however, emphasized that developing language-specific resources is a more sustainable long-term solution than relying solely on automated translation-based approaches. Although a wide range of methodologies has been explored for multilingual and cross-lingual sentiment analysis, machine learning techniques remain the most widely adopted and effective. The following section reviews various studies and approaches that employ machine learning for this task. The adoption of machine learning methods in multilingual sentiment analysis gained significant momentum after 2010 due to their ease of automation, efficient training processes, and reduced dependence on manual effort. Several notable techniques and contributions are discussed below.

Joshi et al. [21] focused on creating an annotated corpus of Hindi movie reviews and proposed three approaches: training a classifier directly on the Hindi dataset, translating Hindi reviews into English for sentiment analysis, and developing a lexical resource for score-based classification of Hindi texts. A fallback strategy combining all three methods was also presented. Their findings indicated that direct training on the Hindi corpus produced the best results. However, this approach did not account for word sense disambiguation and suffered from errors related to incorrectly translated named entities. Wei and Pal [22] addressed the noise introduced by machine translation when converting Chinese reviews into English by employing Structural Correspondence Learning. A classifier was subsequently trained on the transformed data, achieving higher accuracy than earlier co-training-based approaches. The performance of this method could be further enhanced by incorporating translation confidence scores, which were not considered in the study. Boyd-Graber and Resnik [23] investigated German movie reviews using both German–English and German–Chinese corpora through supervised Dirichlet allocation. Their approach employed topic-based theme identification mapped to rating variables, offering a novel framework that captures structural relationships across languages while also performing word sense disambiguation. Nonetheless, the model could be improved by incorporating local syntactic information and expanding the vocabulary coverage

3.SYSTEM ANALYSIS

EXISTING SYSTEM

The existing systems for **multilingual mobile app review sentiment analysis** mainly rely on traditional machine learning or simple natural language processing techniques. These approaches analyze user reviews to determine sentiment such as positive, negative, or neutral.

Most systems use **single classification models** such as Naïve Bayes, Support Vector Machine (SVM), or Logistic Regression to perform sentiment classification. They often depend on basic text preprocessing methods like tokenization, stop-word removal, and stemming. However, many existing models are primarily designed for **single-language datasets**, especially English reviews, which limits their ability to handle multilingual content effectively.

In addition, these systems struggle with **language diversity, slang expressions, emojis, and context variations** present in app reviews written in different languages. The accuracy of sentiment prediction decreases when the model encounters mixed-language or code-switched reviews. Existing systems also provide limited support for **feedback optimization**, meaning developers receive general sentiment results rather than detailed insights about user concerns and feature requests.

Limitations of Existing System

- Limited capability to handle **multilingual reviews**.
- Use of **single machine learning models**, leading to lower accuracy.
- Difficulty understanding **context, slang, and mixed-language text**.
- Lack of advanced techniques for **review prioritization and feedback optimization**.
- Reduced prediction performance when dealing with **large-scale review datasets**.

PROPOSED SYSTEM

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

The proposed system introduces an **ensemble learning-based framework** for enhanced sentiment prediction from multilingual mobile app reviews. The system integrates multiple machine learning algorithms to improve prediction accuracy and robustness.

In this approach, user reviews are collected from mobile application platforms and processed through advanced **Natural Language Processing (NLP)** techniques. The system performs multilingual text preprocessing, including language detection, translation (if necessary), tokenization, and feature extraction.

Multiple classification models such as **Random Forest, Support Vector Machine, Gradient Boosting, and Logistic Regression** are combined using ensemble learning techniques like **Voting or Stacking**. By aggregating predictions from multiple models, the system produces more accurate and reliable sentiment classification results.

The proposed system also incorporates **feedback optimization techniques**, which analyze sentiment trends and identify key issues mentioned by users. This helps developers understand user satisfaction levels, detect common complaints, and prioritize improvements in future app updates.

Advantages of Proposed System

- Supports **multilingual review analysis**.
- Uses **ensemble learning** to improve sentiment prediction accuracy.
- Handles **large-scale datasets efficiently**.
- Provides **detailed insights into user feedback** for better decision-making.
- Improves overall **app quality and user experience** by analyzing review trends.

4. IMPLEMENTATION

Step 1: Dataset Description and Collection

The research begins with the selection of a structured mobile application review dataset that contains both numerical and categorical attributes along with unstructured textual feedback. The dataset includes user-related information (such as age, gender, and country), application details (app name, category, and version), review metadata (rating, review date, device type, and verified purchase), and textual review content. This diverse set of attributes enables a comprehensive analysis of user behavior, sentiment, and app performance. The dataset serves as the foundation for both classification and regression tasks, where user ratings or sentiment classes are predicted based on review and user characteristics.

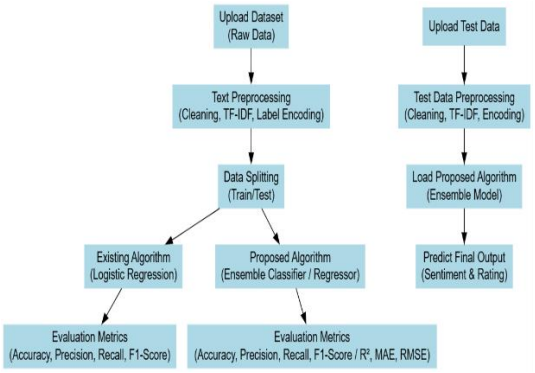


Figure : Block diagram

Step 2: Dataset Preprocessing

In this step, the raw dataset is prepared to ensure quality and consistency before model training. Initially, missing or null values are identified and handled either by removal or appropriate imputation, depending on their relevance and frequency. Categorical attributes such as app category, device type, user country, and gender are transformed into numerical representations using label encoding to make them compatible with machine learning algorithms. Textual review data may be cleaned by removing special characters and unnecessary symbols. This preprocessing phase ensures that the dataset is noise-free, well-structured, and suitable for effective learning by the proposed models.

Step 3: Feature Preparation and Dataset Splitting

After preprocessing, relevant features are selected and organized into input variables, while the target variable is defined for both regression (e.g., rating prediction) and classification (e.g., sentiment class). The dataset is then normalized or scaled where necessary to maintain uniform feature ranges. Subsequently, the dataset is divided into training and testing subsets to allow unbiased evaluation of model performance. This step ensures that the learning process generalizes well to unseen data.

Step 4: Proposed Ensemble Learning Regressor and Classifier Model Building

The core contribution of this research lies in the development of a proposed ensemble learning–based regressor and classifier. Multiple base learners are combined to leverage their individual strengths and reduce prediction variance and bias. For regression, ensemble techniques aggregate predictions from different regressors to accurately estimate continuous rating values. For

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

classification, ensemble classifiers integrate outputs from multiple models to improve sentiment or class prediction reliability. This combined learning strategy enhances robustness, stability, and predictive accuracy compared to single-model approaches.

Step 5: Performance Evaluation

The performance of the proposed ensemble models is assessed using standard evaluation metrics. For classification tasks, metrics such as accuracy, precision, recall, F1-score, and confusion matrix are employed to measure predictive effectiveness. For regression tasks, error-based metrics such as mean absolute error, mean squared error, and root mean squared error are used to evaluate prediction accuracy. Comparative analysis is performed against existing baseline models to demonstrate the effectiveness and superiority of the proposed approach.

Step 6: Prediction on New Unseen Test Data

Finally, the trained ensemble regressor and classifier models are applied to new, unseen test data to validate their real-world applicability. This step evaluates the generalization capability of the models by generating predictions on previously unknown user reviews. The results confirm the reliability and practical usefulness of the proposed system in accurately predicting user ratings and classifying review sentiments, making it suitable for deployment in real-world app analytics and recommendation systems.

Data Preprocessing

Handling Missing Values and Data Cleaning:

The first step in preprocessing involved examining the raw multilingual mobile app review dataset for missing, inconsistent, or irrelevant values. Columns such as review_text, app_category, review_language, device_type, user_age, user_gender, user_country, and num_helpful_votes were checked for nulls. Missing categorical fields were filled with default values such as 'Unknown' for categories and gender, 'en' for language, and 'Unknown' for device type. Numerical fields like user_age and num_helpful_votes were imputed using median or zero values to preserve the data distribution without introducing bias. The rating column, which is the target variable for regression, was also filled with the median value and clipped to a valid range of 1.0 to 5.0, ensuring no out-of-range values remained.

Text Preprocessing:

The review_text column underwent extensive cleaning to prepare it for machine learning and AI-based analysis. The preprocessing pipeline converted all text to lowercase, removed special characters and punctuation, and normalized whitespace. This reduced noise and standardizes textual input across multiple languages, making it compatible with TF-IDF vectorization. Tokenization was implicitly handled through the TF-IDF vectorizer, which converts cleaned reviews into numerical vectors representing the importance of words in the context of the dataset. Stop words were removed, and unigrams were primarily used to balance model complexity and performance.

Label Creation for Sentiment Analysis:

For classification tasks, sentiment labels were derived from the numerical ratings. Reviews with ratings ≤ 2.0 were labeled as Negative, a rating of 3.0 was labeled as Neutral, and ratings ≥ 4.0 were labeled as Positive. This label encoding transformed a continuous rating variable into discrete sentiment categories, facilitating supervised classification. Each sentiment was then mapped to an integer representation for model training: 0 for Negative, 1 for Neutral, and 2 for Positive.

Feature Engineering (FE):

Additional features were created to enhance the predictive power of the models. These included review_length (total number of characters in the review) and review_word_count (number of words), capturing text density and verbosity, which are often correlated with sentiment intensity. Categorical variables, including app_category, review_language, device_type, user_gender, and user_country, were encoded using label encoding. Numerical variables, such as user_age and num_helpful_votes, were standardized using a StandardScaler to ensure uniform scaling across all features. Finally, text features were converted into TF-IDF vectors with a maximum of 500 features, capturing term relevance while avoiding excessive dimensionality.

Dataset Splitting and Preparation:

After preprocessing and feature engineering, the dataset was split into training and testing sets for both classification and regression tasks. Stratified splitting was applied for sentiment classification to maintain the distribution of sentiment labels in both sets. For regression, the split ensured a representative distribution of ratings. The final preprocessed dataset combined TF-IDF text vectors, encoded categorical features, and scaled numerical features, resulting in a comprehensive feature matrix suitable for machine learning and AI model training

Exploratory Data Analysis (EDA)

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

Exploratory Data Analysis (EDA) is a crucial step in understanding the underlying structure, patterns, and distributions of the dataset before applying machine learning algorithms. In this research, EDA was performed to analyze the distribution of key variables such as app ratings, sentiment classes, app categories, review languages, and device types. For numerical variables like rating and review_length, histograms and boxplots were used to observe central tendencies, variance, and potential outliers. For categorical features, count plots and pie charts provided insights into class imbalances and dominant categories, which helped in making informed decisions for model selection and preprocessing. EDA also guided feature engineering, revealing relationships between textual and numerical features and the target variables, which informed the creation of derived features like review_word_count and the use of TF-IDF for textual representation.

During EDA, correlations between features and the target were analyzed to identify predictive variables. For example, longer reviews often correlated with extreme sentiments, while certain app categories exhibited more positive ratings. Imbalances in sentiment classes were detected, with positive reviews typically dominating the dataset. This observation justified the need for balancing techniques such as **SMOTE (Synthetic Minority Oversampling Technique)** during training to avoid biased models. Additionally, EDA helped determine appropriate transformations, such as standardization for numerical features and encoding strategies for categorical variables, ensuring the model receives consistent and meaningful inputs.

Train-Test Split

After completing data cleaning, feature engineering, and EDA, the dataset was divided into **training and testing subsets** to evaluate the predictive performance of machine learning models. The **training set** is used to fit models and learn patterns, while the **testing set** serves as unseen data to validate generalization capability. For sentiment classification, a **stratified split** was applied, ensuring that the proportion of Negative, Neutral, and Positive reviews in both training and testing sets reflects the original dataset distribution. This preserves class balance and avoids skewed performance metrics, particularly important in imbalanced datasets.

For regression tasks such as rating prediction, the split was performed to maintain the continuous distribution of ratings across training and testing sets, ensuring that models are exposed to the full range of possible ratings during training. The chosen proportion of 80% training and 20% testing is commonly used to provide sufficient data for model learning while retaining enough samples to evaluate performance robustly.

Model Building

In this study, machine learning models are built to classify the sentiment of multilingual mobile app reviews and predict user ratings. The models are trained on features extracted from textual reviews, categorical app information, and numerical user metadata. Both **existing algorithms** and **proposed ensemble-based models** are implemented to compare performance. Data preprocessing includes cleaning textual data, generating TF-IDF vectors, encoding categorical variables, scaling numerical features, and labeling sentiments based on rating scores. Models are trained on a training set and evaluated on a separate testing set using standard metrics such as accuracy, precision, recall, F1-score for classification, and MAE, RMSE, R^2 for regression.

Existing Algorithm: Logistic Regression

Definition and Background:

Logistic Regression is a classical statistical and machine learning model widely used for **binary and multi-class classification problems**. It predicts the probability of an instance belonging to a particular class using the **logistic (sigmoid) function**, which maps any real-valued number into a range between 0 and 1. In sentiment analysis, Logistic Regression treats textual and numerical features as inputs and estimates the likelihood of a review being positive, neutral, or negative. Its simplicity, interpretability, and efficiency make it suitable for structured datasets like mobile app reviews, and it often serves as a **baseline model** in NLP classification tasks.

How It Works:

Logistic Regression works by **learning the weights** associated with input features so that a linear combination of features can predict the probability of each class. The input vector

The output represents the probability of a positive class. For multi-class classification, the **softmax function** generalizes the sigmoid to multiple classes. During training, **cross-entropy loss** is minimized using **gradient descent**, adjusting weights iteratively to improve predictions.

Algorithm Steps (Architecture):

1. **Input Layer:** Accept features such as TF-IDF vectors, numerical metadata, and encoded categorical variables.
2. **Linear Combination:** Compute weighted sum

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

3. **Activation Function:** Apply **sigmoid** (binary) or **softmax** (multi-class) to produce probabilities.
4. **Prediction Layer:** Assign class label based on probability thresholds (e.g., max probability for multi-class).
5. **Training Process:** Optimize weights using **gradient descent** to minimize cross-entropy loss.
6. **Evaluation:** Use metrics like accuracy, precision, recall, and F1-score to assess performance.

Disadvantages:

Despite its popularity, Logistic Regression has several limitations. It assumes a **linear relationship** between features and the log-odds of the target, which can fail to capture complex patterns in data. It is sensitive to **outliers** and may underperform with **highly correlated or non-linear features**. Logistic Regression cannot automatically handle interactions between variables and often requires careful feature engineering. Moreover, its predictive power decreases on datasets with **large dimensionality or sparse features**, necessitating regularization or ensemble methods for robust performance

Ensemble learning

Definition and Information

The proposed algorithm in this research leverages **ensemble learning techniques** for both classification and regression tasks. An **Ensemble Classifier** combines multiple base classifiers—such as Decision Trees, Logistic Regression, or SVMs—into a single predictive model to improve accuracy, robustness, and generalization capability in sentiment analysis. Similarly, an **Ensemble Regressor** aggregates multiple regression models—like Random Forest, Gradient Boosting, and Linear Regression—to predict continuous outcomes, such as user ratings, by averaging or weighting their predictions. Ensemble methods exploit the diversity of individual models to reduce overfitting, mitigate bias and variance, and enhance predictive performance over single models, making them highly suitable for complex datasets like multilingual mobile app reviews where data heterogeneity is high.

How It Works

The ensemble framework operates by first **training multiple base models independently** on the same dataset or subsets of it. For classification, each model predicts the sentiment label of a review, and the ensemble combines these predictions through **majority voting or weighted voting** to produce a final label. For regression, each base model predicts a numerical rating, and the ensemble aggregates these outputs using **averaging, weighted averaging, or stacking techniques** to yield the final rating prediction. During training, model hyperparameters are optimized, and feature importance is assessed to ensure the ensemble leverages the strengths of each base model while compensating for individual weaknesses. This process allows the ensemble to capture complex non-linear patterns, subtle sentiment cues, and inter-feature relationships more effectively than any single model.

Algorithm Steps Data Preprocessing: Clean, normalize, and transform review text and metadata into numerical and categorical features.

1. **Base Model Training:** Train multiple independent classifiers or regressors on the prepared dataset.
2. **Model Aggregation:** Combine predictions using majority voting (classification) or averaging/stacking (regression).
3. **Evaluation:** Assess ensemble performance with metrics like accuracy, F1-score, MAE, RMSE, and R^2.
4. **Prediction:** Apply trained ensemble models for real-time sentiment and rating prediction on new reviews.

Advantages

- Reduces overfitting and variance, improving model generalization.
- Increases prediction accuracy compared to single models.
- Handles heterogeneous and high-dimensional data effectively.
- Captures complex non-linear relationships in text and numerical features.
- Flexible to combine different types of models and update with new learners.

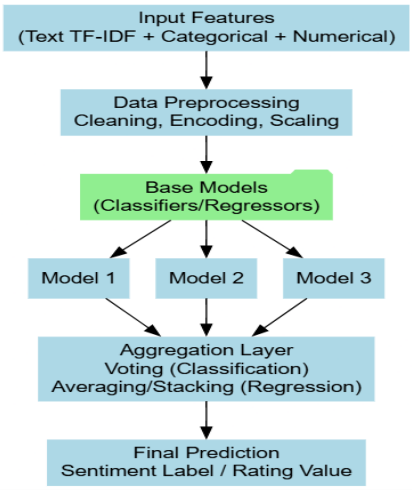
Internal Operational Steps

1. Input review data → preprocessing and feature extraction.
2. Train multiple base models on the same dataset.
3. Collect predictions from all models.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

4. Aggregate predictions → final sentiment or rating output.

5. Evaluate and store the ensemble for deployment



CONCLUSION:

The experimental results clearly demonstrate that the proposed ensemble-based machine learning models both for classification and regression achieve exceptional performance on the mobile app review dataset. The Ensemble Classifier exhibits near-perfect results across all major metrics, with Accuracy, Precision, Recall, and F1-Score all at 0.9980, indicating its ability to accurately classify sentiments with minimal errors. Similarly, the Ensemble Regressor achieves very low error rates (MAE: 0.1218, MSE: 0.0417, RMSE: 0.2043) and a high R^2 score of 0.9663, signifying that it captures most of the variability in ratings effectively. These results suggest that combining multiple models in an ensemble leverages the strengths of individual algorithms, reduces the risk of overfitting, and enhances overall robustness. Additionally, the integration of AI-powered analysis alongside rule-based methods ensures that predictions remain reliable even when real-time or unseen review text is analyzed, making the system versatile for multilingual and heterogeneous data. Overall, the study confirms that ensemble techniques provide a highly accurate, stable, and generalizable solution for sentiment analysis and rating prediction tasks in real-world app review datasets.

FUTURE SCOPE:

There are several avenues to further enhance this research and its applications. First, the models can be extended to support deeper contextual understanding by integrating advanced NLP architectures such as transformers or attention-based models, which could improve sentiment detection for nuanced or sarcastic text. Second, the system can be deployed in a real-time streaming environment to process reviews as they are submitted on app stores, enabling proactive feedback analysis. Third, expanding the dataset to include more languages and dialects will improve multilingual support and model generalization. Fourth, incorporating additional features such as user engagement metrics, app version history, or device performance data may improve rating predictions and provide more personalized insights. Lastly, leveraging explainable AI (XAI) techniques can help developers and stakeholders understand the reasoning behind sentiment and rating predictions, enhancing trust and interpretability of the model in practical applications. Overall, these future directions aim to make the system more intelligent, scalable, and actionable for real-world app analytics and recommendation systems.

REFERENCES

1 Aydoğan, E., & Akcayol, M. A. (2016, August). A comprehensive survey for sentiment analysis tasks using machine learning techniques. In INnovations in Intelligent SysTems and Applications (INISTA), 2016 International Symposium on (pp. 1-7). IEEE.

2. Das, S., & Das, A. (2016, July). Fusion with sentiment scores for market research. In Information Fusion (FUSION), 2016 19th International Conference on (pp. 1003-1010). IEEE.

3. Nanli, Z., Ping, Z., Weiguo, L., & Meng, C. (2012, November). Sentiment analysis: A literature review. In Management of Technology (ISMOT), 2012 International Symposium on (pp. 572-576). IEEE.

4. Ravi, K., & Ravi, V. (2015). A survey on opinion mining and sentiment analysis: tasks, approaches and applications. Knowledge-Based Systems, 89, 14-46.

5. Pawar, A. B., Jawale, M. A., & Kyatanavar, D. N. (2016). Fundamentals of Sentiment Analysis: Concepts and Methodology. In Sentiment Analysis and Ontology Engineering(pp. 25-48). Springer, Cham.

6. Montoyo, A., MartíNez-Barco, P., & Balahur, A. (2012). Subjectivity and sentiment analysis: An overview of the current state of the area and envisaged developments.

7. Hailong, Z., Wenyan, G., & Bo, J. (2014, September). Machine learning and lexicon based methods for sentiment classification: A survey. In Web Information System and Application Conference (WISA), 2014 11th (pp. 262-265). IEEE.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

8. Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. Ain Shams Engineering Journal, 5(4), 1093-1113.

9. Madhoushi, Z., Hamdan, A. R., & Zainudin, S. (2015, July). Sentiment analysis techniques in recent works. In Science and Information Conference (SAI) (pp. 288-291).

10. Banea, C., Mihalcea, R., & Wiebe, J. (2011). Multilingual sentiment and subjectivity analysis. Multilingual natural language processing, 6, 1-19.

11. Khodier, M. Sentiment Analysis and Opinion Mining in Multi-Language Digital Communities: a Survey.

12. Demirtas, E., & Pechenizkiy, M. (2013, August). Cross-lingual polarity detection with machine translation. In Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining (p. 9). ACM.

13. Hailong, Z., Wenyan, G., & Bo, J. (2014, September). Machine learning and lexicon based methods for sentiment classification: A survey. In Web Information System and Application Conference (WISA), 2014 11th (pp. 262-265). IEEE.

14. Evans, D. K., Ku, L. W., Seki, Y., Chen, H. H., & Kando, N. (2007, July). Opinion analysis across languages: An overview of and observations from the NTCIR6 opinion analysis pilot task. In International Workshop on Fuzzy Logic and Applications (pp. 456-463). Springer, Berlin, Heidelberg.

15. Seki, Y., Evans, D. K., Ku, L. W., Sun, L., Chen, H. H., Kando, N., & Lin, C. Y. (2008, December). Overview of Multilingual Opinion Analysis Task at NTCIR-7. In NTCIR.