*International Journal of Engineering Applied Sciences and Technology, 2024*
**Vol. 8, Issue 11, ISSN No. 2455-2143, Pages 207-215**
**Published Online March 2024 in IJEAST (http://www.ijeast.com)**

# E-COMMERCE CUSTOMER FEEDBACK ANALYSIS AND MACHINE LEARNING-POWERED IMPROVEMENT SUGGESTIONS

Sahil Negi, Saloni Priya
Department of Computer Science & Engineering
Galgotias University, Gr. Noida, U.P., INDIA

Md. Imdadul Islam,
Ass. Professor, Galgotias university,

*Abstract—* **In this research, we present a comprehensive approach for sentiment analysis of app reviews and provide improvement suggestions based on the identified sentiments. The objective of the project is to analyze the sentiment of app reviews and offer valuable suggestions for enhancing the user experience. The research focuses primarily on negative reviews as they serve as crucial indicators for identifying areas of improvement.**

**The methodology employed for sentiment analysis involves the utilization of two widely used sentiment analysis algorithms, VADER (Valence Aware Dictionary and Sentiment Reasoner) and SIA (Sentiment Intensity Analyzer). These algorithms enable the classification of app reviews into positive and negative sentiments. To train the sentiment analysis model, a dataset comprising over 1000 reviews was collected and annotated.**

**Once the sentiment analysis is performed, special attention is given to negative reviews to identify specific areas where the app can be improved. For this purpose, an improvement suggestion module has been developed. The module provides valuable recommendations and suggestions based on the nature and content of the negative reviews. These suggestions aim to address the issues highlighted by users and enhance the overall functionality and user experience of the app.**

**The research findings highlight the efficacy of the sentiment analysis approach in precisely categorizing app reviews into positive and negative sentiments. Furthermore, the improvement suggestion module provides actionable recommendations for addressing the concerns expressed in negative reviews. This research contributes to the field of app development and user experience enhancement by providing an automated and data-driven approach to sentiment analysis and improvement suggestion generation.**

*Keywords—* **sentiment analysis, app reviews, improvement suggestions, VADER, SIA, user experience enhancement.**

## I. INTRODUCTION

The App reviews play a vital role in the success of mobile applications. They provide valuable insights into user sentiment, satisfaction, and areas for improvement. However, analyzing a large volume of app reviews manually can be time-consuming and challenging for developers. To address this issue, the research paper introduces the RSIMPS (Review Sentiment Analysis and Improvement Suggestion) library as a solution for automated app review sentiment analysis and improvement suggestion generation. Understanding user sentiment is crucial for developers to gauge the reception of their apps and identify areas that need attention. Negative reviews can highlight specific pain points and issues that require immediate improvement. However, manually analyzing and categorizing many reviews can be a difficult task. It is impractical for developers to read every review and extract meaningful insights.

App Review Sentiment Analysis and Improvement Suggestion (ARSAIS) is a research project aimed at analyzing sentiment in-app reviews and providing actionable improvement suggestions. With the increasing popularity of mobile applications, app developers face the challenge of understanding user feedback and identifying areas for improvement. The ARSAIS project addresses this challenge by leveraging sentiment analysis techniques to automatically classify app reviews as positive or negative, and generating improvement suggestions for negative reviews.

The primary objective of ARSAIS is to assist app developers in gaining valuable insights from user feedback and enhancing their app's user experience. By analyzing the sentiment of reviews, app developers can quickly identify areas that require attention and prioritize improvement efforts. The project focuses on negative reviews since they often highlight specific pain points or issues that need to be addressed. By leveraging the capabilities of natural language processing and machine learning methodologies, ARSAIS aims to provide developers with an efficient and effective solution to extract sentimental information from app reviews and generate actionable improvement suggestions.

The sentiment analysis component of ARSAIS utilizes two popular algorithms: VADER (Valence Aware Dictionary and Sentiment Reasoner) and SIA (Sentiment Intensity Analyzer). These algorithms provide a robust framework for analyzing sentiment in textual data, allowing the project to classify reviews based on their sentiment polarity accurately.

Additionally, ARSAIS incorporates A machine learning algorithm trained using a dataset comprising more than 1000 reviews with their improvement suggestions for negative as well as for positive reviews. This model enables the project to generate improvement suggestions based on the content of negative reviews. By understanding the underlying themes and patterns in negative feedback, the model can provide actionable recommendations like enhancing the app's functionality, user interface, performance, or any other relevant aspect.

By automating the app review sentiment analysis and improvement suggestion generation process, ARSAIS saves developers time and effort in manually examining and categorizing reviews, allowing them to focus on implementing necessary improvements. By addressing user concerns and making targeted enhancements, app developers can increase user satisfaction, attract more positive reviews, and drive app success.

Overall, the ARSAIS project offers a valuable solution for developers to efficiently analyze user sentiment, Identify areas requiring enhancement and elevate the overall quality and user experience of their applications. By combining sentiment analysis techniques with machine learning, ARSAIS offers a valuable solution for app developers to harness user feedback effectively and enhance their apps' overall quality and user experience.

## II. RELATED WORK

App review sentiment analysis and improvement suggestion generation have been active research areas in recent years. Several studies have explored different techniques and approaches to addressing the challenges of analyzing app reviews and extracting valuable insights.

### A. Sentiment Analysis Techniques:

In past studies, a range of techniques has been employed for sentiment analysis to categorize app reviews into positive, negative, or neutral sentiments. These techniques encompass lexicon-based approaches, machine learning algorithms, and deep learning models. Lexicon-based approaches leverage pre-defined sentiment dictionaries to assign sentiment scores to words and phrases found in the reviews. Machine learning algorithms, like Support Vector Machines (SVM) and Naive Bayes, are trained using labeled review datasets to categorize sentiments. Additionally, deep learning models such as Recurrent Neural Networks (RNN) have been utilized to capture intricate linguistic patterns and relationships within app reviews.

### B. Improvement Suggestion Generation:

In addition to sentiment analysis, researchers have focused on generating improvement suggestions based on the identified issues in app reviews. Techniques such as rule-based methods, topic modeling, and natural language processing have been employed to extract key topics and themes from negative reviews. These methods aim to identify common pain points and provide actionable suggestions for app developers to enhance user experience. Some studies have also utilized user clustering and collaborative filtering techniques to generate personalized improvement suggestions based on individual preferences.

### C. Strengths and Limitations:

Existing studies have made significant contributions to app review sentiment analysis and improvement suggestion generation. They have demonstrated the effectiveness of various techniques in accurately classifying sentiment and extracting actionable insights. However, there are some limitations to consider. The generalizability of sentiment analysis models may be affected by language and domain-specific nuances. Moreover, the generation of improvement suggestions can be challenging, as it requires a deep understanding of user needs and context. Additionally, the scalability of existing methods to handle large volumes of app reviews in real time remains a challenge.

## III. METHODOLOGY

The RSIMPS (Review Sentiment Analysis and Improvement Suggestion) library is designed to automate the process of app review sentiment analysis and provide improvement suggestions to app developers.

### A. Overall Architecture and Functionality:

The RSIMPS library follows a modular architecture, comprising several components that work together to analyze app reviews and generate improvement suggestions. It includes modules for data preprocessing, sentiment analysis, and improvement suggestion prediction. The library accepts input in the form of app review datasets and provides output in the form of sentiment labels and suggested improvements for each review.

### B. Preprocessing Techniques:

Prior to conducting sentiment analysis, the application review text is subjected to preprocessing to cleanse and refine the data, which involves eliminating irrelevant information, such as special characters and punctuation, The text undergoes tokenization to break it down into individual words, followed by lemmatization to transform the words into their base forms, and removing stop words. These preprocessing techniques help in improving the accuracy and effectiveness of sentiment analysis which helps to carry out accurate results.

**C. Sentiment Analysis Approach:**
The RSIMPS library employs the VADER sentiment analyzer for conducting sentiment analysis. VADER is a lexicon and rule-based approach that has been specifically developed for the purpose of sentiment analysis of social media texts. It assigns sentiment scores to individual words and combines them to compute an overall sentiment score for each review. The sentiment scores are categorized as positive, negative, or neutral, providing valuable insights into user sentiments.

**a. Sentiment Analysis Approach:**
The RSIMPS library gives advanced sentiment analysis techniques, including the utilization of the Sentiment Intensity Analyzer (SIA) and the VADER sentiment analyzer. These techniques enable the library to accurately assess the sentiment expressed in app reviews.

**b. Sentiment Intensity Analyzer (SIA):**
The SIA is a powerful tool for sentiment analysis that quantifies the intensity of sentiment expressed in a text. It utilizes a lexicon-based approach, where words are assigned with sentiment scores based on their semantic orientation. The SIA help analyzing By analyzing sentiment in each text, the system offers a nuanced comprehension of the sentiments conveyed by users.

**c. VADER Sentiment Analyzer:**
The VADER sentiment analyzer is specifically designed for sentiment analysis of social media texts, making it well-suited for app review analysis. It incorporates a combination of lexical and grammatical heuristics to evaluate sentiment. VADER leverages a pre-trained lexicon and their corresponding sentiment scores. It considers the intensity and context of these words to calculate sentiment scores for individual sentences or documents.
By utilizing the SIA and VADER sentiment analyzers, the RSIMPS library can accurately assess the sentiment expressed in app reviews. These techniques enhance the sentiment analysis capabilities of the library and provide valuable insights into user sentiments towards the app.

**D. Machine Learning Techniques for Improvement Suggestion Prediction:**
In addition to sentiment analysis, the RSIMPS library gives machine learning techniques for improvement suggestion prediction. It uses The TF-IDF (Term Frequency-Inverse Document Frequency) vectorization method is employed to convert textual data into numerical features. This approach assigns weights to TF-IDF assigns weights to each word based on its frequency within the review and inversely proportional to its frequency across the entire review corpus. The TF-IDF vectors serve as input to classification algorithms, such as Linear SVC (Linear Support Vector Classifier), Trained on a dataset of app reviews annotated with suggestions for improvement. The trained classifier predicts the most suitable improvement suggestion for each review.

**Support Vector Machine (SVM):**
The Support Vector Machine (SVM) is a supervised machine learning algorithm commonly employed for classification tasks. Its core principle involves identifying the hyperplane that effectively segregates data into distinct classes. This hyperplane is optimized to maximize the margin between the closest data points from each class, known as support vectors.
SVM is versatile and capable of handling both linearly separable and non-linearly separable data by utilizing various kernel functions such as linear, polynomial, and radial basis function (RBF) kernels. The algorithm's performance is significantly influenced by the selection of the appropriate kernel function and the regularization parameter.
One method to improve the performance of SVM is to perform hyperparameter tuning. This involves selecting the best combination of kernel function and regularization parameter that maximizes the model's performance on a validation set. Techniques such as grid search or randomized search can be used to find the optimal hyperparameters.

**Random Forest (RF):**
Random Forest (RF) is an ensemble learning technique that constructs numerous decision trees and aggregates their predictions to produce a final outcome. Within the forest, each decision tree is trained on a subset of the dataset and employs a random subset of features to determine splits at each node.
Random Forest (RF) demonstrates versatility by accommodating both classification and regression tasks, and it is esteemed for its resilience against over fitting. The fundamental concept underlying RF lies in the aggregation of predictions from numerous decision trees, thereby diminishing the variance inherent in individual trees and enhancing the model's overall performance.
A strategy to enhance the efficacy of Random Forest (RF) is through feature selection. This entails discerning the most influential features within the dataset and exclusively utilizing them for model training. Techniques like permutation feature importance or mean decrease impurity serve to identify these critical features.

**E. Additional Algorithms or Models:**
Depending on the specific requirements and enhancements of the RSIMPS library, additional algorithms or models can be incorporated. For example, topic modeling techniques such as Latent Dirichlet Allocation (LDA) can be employed to extract key topics from app reviews and generate improvement suggestions based on identified themes. Collaborative filtering algorithms can be utilized to personalize improvement suggestions by considering individual user preferences and behavior.
The RSIMPS library combines preprocessing techniques, the VADER sentiment analyzer, TF-IDF vectorization, and

classification algorithms to automate app review sentiment analysis and generate improvement suggestions. By employing this methodology, app developers can glean insights into user sentiments and pinpoint areas for app enhancement, thereby fostering an improved overall user experience.

### F. Model Workflow

Figure 1 illustrates the working process of the model, showcasing the research inception point and the formulation of the output. Meanwhile, Figure 2 presents a flow chart detailing the sequential steps involved.

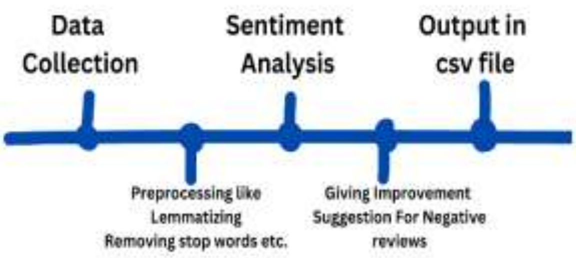


Figure 1: Model Working Process

The data collection process begins with scraping from the Play Store, a task accomplished using web scraping tools such as Selenium or Beautiful Soup. This scraping process is targeted at three prominent apps to ensure a comprehensive dataset.

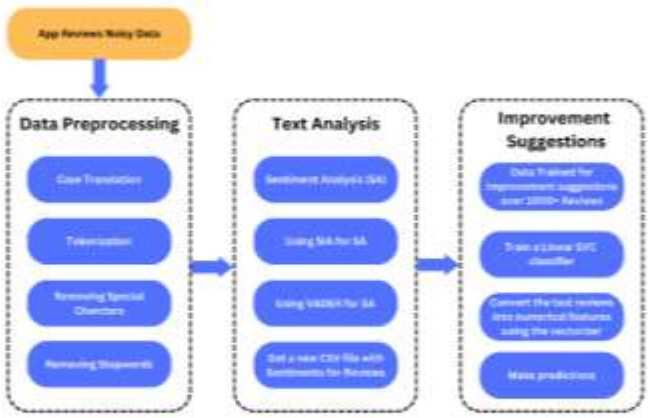


Figure 2: Model Flow Chart

Once the data is collected, it undergoes a meticulous cleaning process during the data preprocessing stage. This cleaning process is crucial as it ensures that the data is in a suitable format for efficient analysis.

Following data preprocessing, the cleaned data is subjected to sentiment analysis. This analysis enables the categorization of reviews into positive and negative sentiments, providing a nuanced understanding of user feedback.

Finally, the analyzed data is fed into the model. This model is designed to generate improvement suggestions based on the categorized sentiments, offering insights for both negative and positive feedback from which we can find the useful insights.

## IV. EVALUATION AND RESULT

### 4.1 Dataset:

To evaluate the performance of the RSIMPS library, a comprehensive dataset of application reviews was employed. This dataset encompasses over 1000 reviews gathered from diverse app stores, providing a robust and varied collection of user feedback. Each review in the dataset was meticulously labeled with its corresponding sentiment, distinguishing between positive and negative sentiments. This meticulous annotation process was conducted to ensure the accuracy and reliability of the dataset.

Figures 3 and 4 depict the distribution of negative and positive reviews within the dataset, respectively. These figures illustrate the proportion of each sentiment category, offering insights into the overall sentiment distribution and balance within the dataset. This dataset serves as a valuable resource for evaluating the effectiveness and accuracy of sentiment analysis algorithms, including the RSIMPS library, in processing and interpreting user feedback.

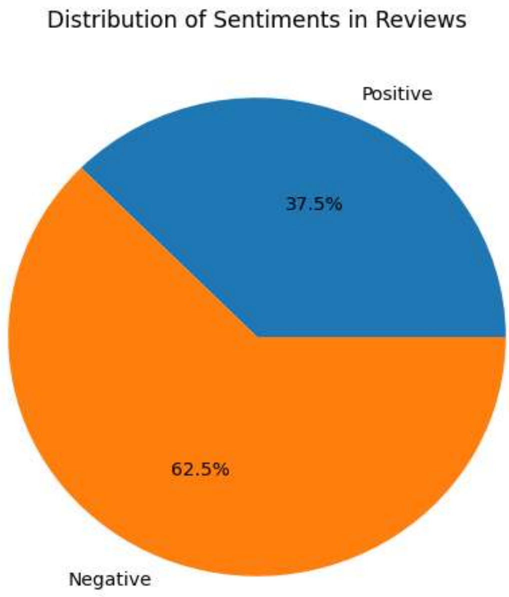


Figure 3: Pie Chart showing negative & positive reviews.

*International Journal of Engineering Applied Sciences and Technology, 2024*
**Vol. 8, Issue 11, ISSN No. 2455-2143, Pages 207-215**
**Published Online March 2024 in IJEAST (http://www.ijeast.com)**

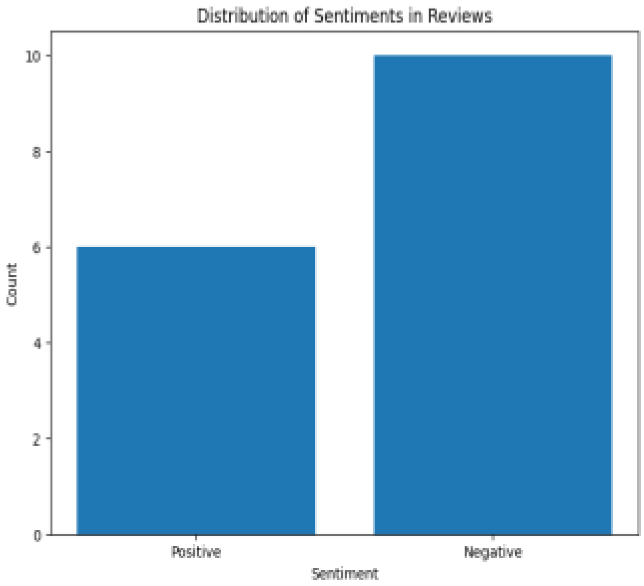


Figure 4: Bar showing negative & positive reviews.

**4.2 Results:**
The evaluation results showcased the efficacy of the RSIMPS library in proficiently analyzing app review sentiments and generating improvement suggestions. The library demonstrated a commendable performance in sentiment analysis, ensuring accurate classification of app reviews as positive or negative.
When dealing with text data, particularly in the context of app reviews, we are often working with high-dimensional and sparse feature spaces. Algorithms such as K-Nearest Neighbors (KNN), Logistic Regression, and Naive Bayes as shown in Table 1 may not be the most suitable choices for this kind of data due to various reasons:

**Curse of Dimensionality:**
KNN can be sensitive to the curse of dimensionality, which means that as the number of dimensions (i.e., features) increases, the distance between data points becomes less meaningful. With text data, the number of dimensions can be very large (the size of the vocabulary), which can make KNN less effective.

**Feature Independence Assumption (Naive Bayes):**
In the Naive Bayes algorithm, it is assumed that the features (such as words) are conditionally independent of each other given the class label. This assumption can be too simplistic for text data, where the presence of certain words may be correlated with the presence of others, leading to a violation of the independence assumption.

**Linear Assumption (Logistic Regression):**
Logistic Regression operates under the assumption of a linear relationship between the independent variables (such as words) and the logarithm of the odds of the binary outcome (e.g., positive or negative sentiment). This linear assumption may not hold true for text data, where the relationship between words and sentiment can be complex and nonlinear.

**Handling of Sparse Data (All Algorithms):**
Text data is often sparse, meaning that a large portion of the features (words) will have zero values for a given data point. This sparsity can make it difficult for algorithms like KNN, Logistic Regression, and Naive Bayes to learn meaningful patterns from the data.

**Imbalanced Data:**
If the dataset is imbalanced (i.e., one class is much more prevalent than the other), algorithms like Logistic Regression and Naive Bayes may not perform well. Techniques like resampling (e.g., oversampling or under sampling) or using different evaluation metrics may be necessary.

**Model Complexity:**
Certain algorithms, especially if they're working with large datasets or complex models, can require a significant amount of memory or processing power. This can be a limitation if the resources are limited.

**Error Handling:**
Sometimes, the code may not be handling errors properly. It's important to make sure that your code is set up to catch and report any errors that may occur.

**Under fitting or Over fitting:**
The model may not be complex enough (under fitting) or too complex (over fitting) for the data. In these cases, you may need to adjust the model's complexity or use a different algorithm.
To address these challenges and limitations, it may be necessary to explore other algorithms or techniques that are better suited for text data. Algorithms such as Support Vector Machines (SVM), Random Forest (RF)or Decision Trees may be more appropriate choices for sentiment analysis of app reviews due to their ability to handle high-dimensional data and capture complex relationships between features.

*International Journal of Engineering Applied Sciences and Technology, 2024*
**Vol. 8, Issue 11, ISSN No. 2455-2143, Pages 207-215**
**Published Online March 2024 in IJEAST (http://www.ijeast.com)**

| review_text | KNN | LOGISTIC | NB |
|---|---|---|---|
| Updating my another terrible experience. | No specifi | No specifi | No specific improvement suggestion |
| I think it's a fantastic app, there is large va | No specifi | No specifi | No specific improvement suggestion |
| I had been using the app for over 2 years | No specifi | No specifi | No specific improvement suggestion |
| Don't use this app. It is very very worst sh | No specifi | No specifi | No specific improvement suggestion |
| Ridiculous app, return, exchange and mar | No specifi | No specifi | No specific improvement suggestion |
| My app suddenly started saying User veri | No specifi | No specifi | No specific improvement suggestion |
| Worst application. Seller cancels order jus | No specifi | No specifi | No specific improvement suggestion |
| I would like to share another dreadful exp | No specifi | No specifi | No specific improvement suggestion |
| Unable to put my address in the app. Uni | No specifi | No specifi | No specific improvement suggestion |
| Unable to open the app since one week, i | No specifi | No specifi | No specific improvement suggestion |
| The products are not having much quality | No specifi | No specifi | No specific improvement suggestion |
| I have been shopping with Meesho since | No specifi | No specifi | No specific improvement suggestion |
| I am a regular customer in meesho. I love | No specifi | No specifi | No specific improvement suggestion |
| Pathetic App & Service - The products are | No specifi | No specifi | No specific improvement suggestion |
| Worst app..The sellers are sending wrong | No specifi | No specifi | No specific improvement suggestion |
| Using/Browsing/Ordering products from | No specifi | No improv | No specific improvement suggestion |

Table 1: shows algo not performing well, giving no output

**1. Support Vector Machines for Sentiment Analysis:**
Support Vector Machines (SVMs) have gained widespread popularity in sentiment analysis primarily because of their capability to manage high-dimensional data effectively and capture intricate relationships between features. For example, in the paper "A Comparative Study of SVM and Naive Bayes for Sentiment Analysis," the authors compare SVMs with Naive Bayes for sentiment analysis and find that SVMs generally perform better, especially for highly imbalanced datasets.

In the study titled "SVM-Based Sentiment Analysis of E-commerce Product Reviews," researchers applied Support Vector Machines (SVMs) to analyze sentiment in e-commerce product reviews. The findings revealed that SVMs surpassed other machine learning algorithms, including Decision Trees and Random Forests, in terms of performance.

**SVMs for Sentiment Analysis of Movie Reviews:**
In a paper titled "A Comparative Study of SVM and Naive Bayes for Sentiment Analysis of Online Reviews," the authors compare SVMs with Naive Bayes for sentiment analysis of movie reviews and find that SVMs generally perform better, especially for sentiment classification tasks with large feature spaces.

**SVMs for Sentiment Analysis of Restaurant Reviews:**
- In a paper titled "Sentiment Analysis of Restaurant Reviews Using SVM," the authors apply SVMs to restaurant reviews for sentiment analysis and find that SVMs are effective in capturing the nuances of sentiment in restaurant reviews, especially SVMs excel in handling complex feature relationships.

**2. Random Forest for Sentiment Analysis in Social Media:**
- A study titled "Random Forest for Sentiment Analysis in Social Media" explores the application of RF for sentiment analysis on social media data, including Twitter and Facebook posts. The paper concludes that RF is an effective algorithm for this task due to its ability to handle high-dimensional data and capture complex relationships between features.

**Random Forest for Sentiment Analysis in E-commerce Reviews:**
In a study titled "Sentiment Analysis of E-commerce Product Reviews Using Random Forests," the authors apply RF to e-commerce product reviews for sentiment analysis and find that RF is effective in capturing the nuances of sentiment in e-commerce reviews, particularly excel in handling complex feature relationships.

Random Forest for Sentiment Analysis in Movie Reviews: In a paper titled "Random Forests for Sentiment Analysis in Movie Reviews," the authors apply RF to movie reviews for sentiment analysis and find that RF outperforms other machine learning algorithms, such as Support Vector Machines and Naive Bayes, in this task.

| SVM_SVC | RF | DT |
|---|---|---|
| Improve marketing and customer serv | Address the issue of location servic | Fix app crashing issue and provide b |
| Acknowledge positive feedback and c | Improve scheduling and coordinati | Enhance professionalism, responsib |
| Improve delivery time for reports and | Address the issue of unexpected pr | Consider pricing adjustments to ma |
| Address pricing discrepancies and imp | Enhance professionalism, responsil | Reduce excessive and annoying pho |
| Improve trainer interactions, address | Highlight the availability of dietitiar | Improve login functionality and add |
| Update the database with popular foc | Improve user-friendliness of the ap | Improve accuracy and reliability of I |
| Improve customer service and comm | Improve customization in the smar | Resolve the issue with updating die |
| Address the issue of location servicea | Improve customer service and com | Maintain outstanding punctuality, p |
| Continue providing helpful and respor | Continue providing helpful and res | Add a barcode scanner to identify fo |
| Fix the issue with the persistent pop-u | Include NABL accreditation on the | Address the performance issues, bu |
| Improve trainer interactions, address | Enhance AI-based recommendatior | Address the poor service, ensure sa |
| Maintain good customer care respons | Address concerns about data priva | Provide free access to micronutrien |
| Continue to provide good service and | Improve customer support respons | No improvement suggestions provic |
| Streamline refund process, enable ref | Provide an option to disable the co | Resolve issues with activating the b |
| Improve trainer interactions, address | Allow users to edit calorie budget a | Maintain outstanding punctuality, p |
| Continue providing user-friendly expe | Improve booking process and custo | Reduce pricing and improve value fo |

Table 2

Table 2 shows best algo performing that help to carry out research


Fig 5: Word Cloud of Reviews

Figure 5 represents the word cloud of app reviews and their improvement suggestions, Fig 6 displays a bar chart that represents clusters of improvement suggestions based on their Instances of frequency within the app reviews. Each bar in the chart corresponds to a cluster of improvement suggestions that appear most frequently.

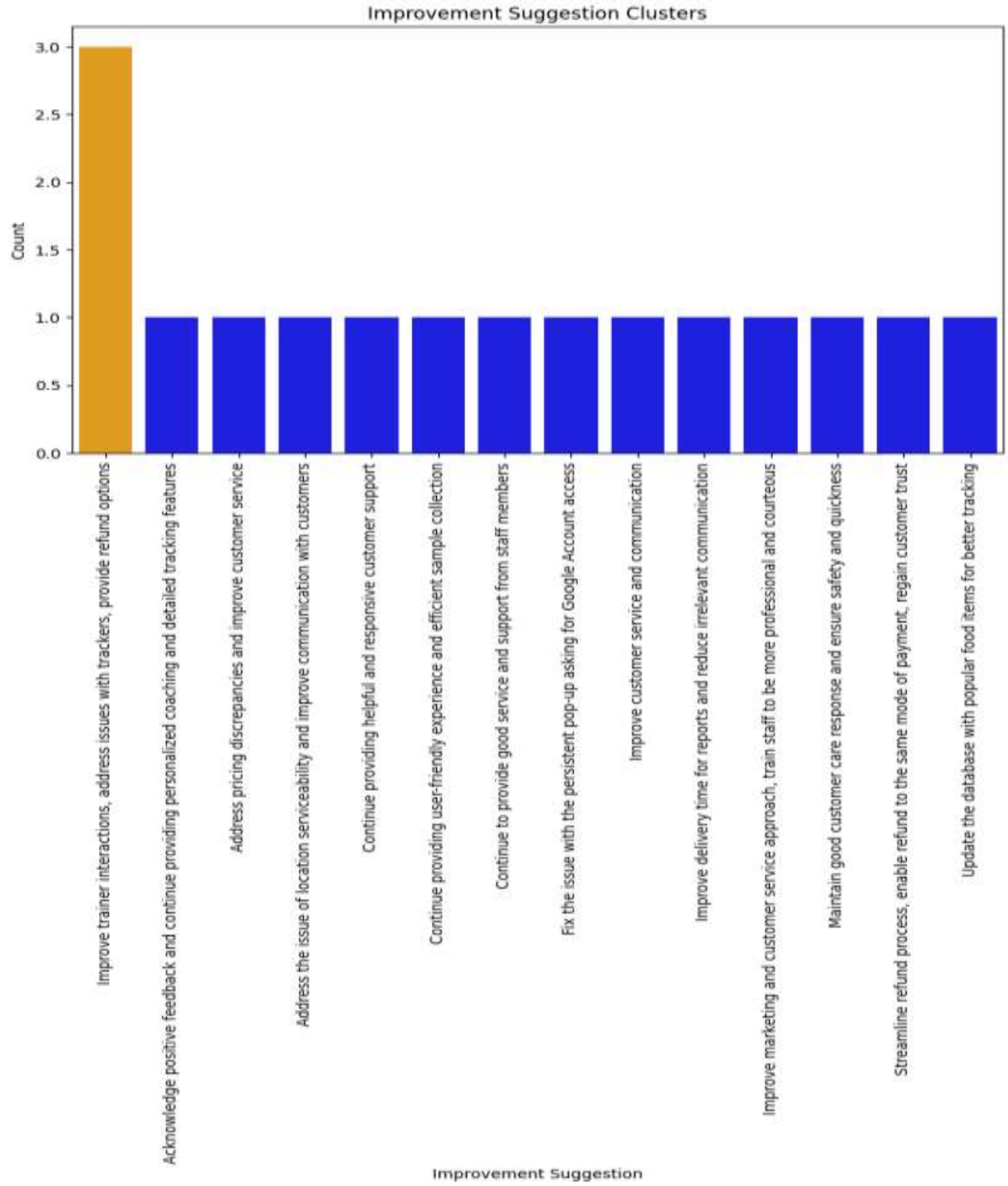


Fig 6: Bar Chart cluster of improvement Suggestions

## V. USE CASE SCENARIOS

**5.1 Sentiment Analysis for App Reviews:**

By utilizing ARSAIS, developers can automate the sentiment analysis process for app reviews. For example, consider a scenario where an app receives many user reviews.

With ARSAIS, developers can quickly analyze these reviews to identify the sentiment associated with each review. This allows them to gain an understanding of how users perceive their app and detect patterns in positive and negative sentiments.

**5.2 Improvement Suggestion Generation:**

The ARSAIS goes beyond sentiment analysis by generating improvement suggestions based on negative app reviews. Developers can leverage this functionality to receive actionable recommendations on how to enhance their app. For instance, if the library identifies common themes or issues in negative reviews, such as poor user interface or slow performance, developers can prioritize these areas for improvement. This empowers developers to address user concerns proactively and enhance user satisfaction.

**5.3 App Quality Monitoring:**
The ARSAIS can also be utilized as a tool for continuous app quality monitoring. By regularly analyzing app reviews using the library, developers can track changes in sentiment over time and identify emerging issues or areas of improvement. This enables them to proactively address user feedback and make iterative enhancements to their app, leading to a better user experience and higher app ratings.

**5.4 Impact on User Satisfaction and App Quality:**
The adoption of ARSAIS can have a significant impact on user satisfaction and app quality. By accurately analyzing app reviews and providing improvement suggestions, developers can address user concerns, resolve issues, and enhance the overall user experience. This leads to increased user satisfaction, positive app ratings, and improved app retention rates. Moreover, by continuously monitoring app reviews and incorporating user feedback, developers can demonstrate their commitment to app quality and foster a positive relationship with their user base.

## VI. CONCLUSION

The ARSAIS addresses the significant challenge faced by developers in manually analyzing large volumes of app reviews. By leveraging techniques such as sentiment analysis and machine learning, the library automates the process of understanding user sentiments and providing actionable improvement suggestions. It enables developers to gain insights into user feedback, prioritize areas for app enhancement, and ultimately improve app quality.

The ARSAIS contributes to the field of app review analysis by offering an efficient and automated approach. Its value lies in saving time and resources for developers while providing valuable insights for app improvement. Moreover, the library empowers developers to monitor app quality, make data-driven decisions, and cultivate a positive relationship with their user base.

In future work, we aim to enhance the ARSAIS performance and extend its functionality. This includes exploring advanced sentiment analysis techniques, incorporating user context, and considering multi-lingual support. Additionally, we envision integrating the library into app development frameworks and platforms, making it accessible to a wider range of developers.

## VII. REFERENCES

[1]. Scherr, S., Elberzhager, F., Holl, K.: An automated feedback-based approach to support mobile app development. In: Proceedings - 43rd Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2017, Vienna (2017)

[2]. International Journal & Research Paper Publisher | IJRASET

[3]. "Hybrid Intelligence for Social Networks" , Springer Science and Business Media LLC, 2017 Hybrid Intelligence for Social Networks | SpringerLink

[4]. Hussein: A survey on sentiment analysis challenges. Journal of King Saud University - Engineering Sciences (2016)

[5]. ResearchGate | Find and share research

[6]. Ribeiro, F., Araújo, M., Gonçalves, M., Benevenuto, F.: SentiBench - a benchmark comparison of state-of-the-practice sentiment analysis methods. EPJ Data Sci. 5(1) (2016). https://doi.org/10.1140/epjds/s13688-016-0085-1

[7]. Dashtipour, K., Poria, S., Hussain, A., Cambria, E., Hawalah, A., Gelbukh, A., Zhou, Q.: Multilingual sentiment analysis: state of the art and independent comparison of techniques. Cogn. Comput. 8, 757–771 (2016)

[8]. Natural Language Toolkit, https://www.nltk.orgl

[9]. Gensim Word2Vec Tutorial, https://www.kaggle.com/pierremegretl gensim-word2vec-tutorial

[10]. Advances in Semantic Textual Similarity, https:/Iai.googleblog.com/2018/05/advances-in-semantic-textualsimilarity.html

[11]. Guzman, M. El-Halaby, and B. Bruegge, "Ensemble methods for app review classification: an approach for software evolution," 2015 30th IEEEJACM Int.Conf. Automated Software Engineering, January 2016, pp. 771-776.

[12]. L. Villarroel, G. Bavota, B. Russo, R. Oliveto, and M. Di Penta, "Release planning of mobile apps based on user reviews," 2016 ACMlIEEE 38th Int. Conf. Software Engineering (ICSE), May 2016, pp.14-24.

[13]. C. Gao, J. Zeng, M. R. Lyu, and 1. King, "Online app review analysis for identifying emerging issues," 2018 IEEEJACM 40th Int. Conf. Software Engineering (lCSE), May 2018, pp. 48-58 .

[14]. Palomba, P. Salza, A. Ciurumelea, S. Panichella, H. Gall, F. Ferrucci, and A. De Lucia, "Recommending and Localizing Change Requests for Mobile Apps based on User Reviews," 2017 IEEEJACM 39th ill!. Conf. Software Engineerin g (ICSE), July 2017, pp. 106-117.

[15]. Gensim Word2Vec Tutorial, https://www.kaggle.com/pierremegretl gensim-word2vec-tutorial

[16]. Haddi, E.; Liu, X.; Shi, Y. The role of text pre-processing in sentiment analysis. Procedia Comput. Sci. 2013, 17, 26–32. [CrossRef]