Journal of Information Systems Engineering and Management
2025, 10(7s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

An Implementation and Analysis of Modified Approach for Mobile Apps Review Mining using Scrapper Package

Makarand Lotan Mali1 *, Nitin N. Patil2
[1]*R. C. Patel Institute of Technology, Shirpur, Nimzari Naka, Road, Shirpur, 425405, INDIA*
[2]*SVKMs, College of Engineering, Dahiwad, At Post Dahiwad, Shirpur, 425405, INDIA*
**Corresponding Author: Makarand Mali*

ARTICLE INFO

Received: 10 Oct 2024

Revised: 12 Dec 2024

Accepted: 26 Dec 2024

ABSTRACT

This study introduces a new method for mining mobile app reviews using the "scrapper" package and sophisticated ML algorithms. The goal was to improve app marketing and development by gleaning useful information from a massive database of user reviews. This was accomplished by classifying the retrieved reviews as either favorable, negative, or neutral using a thorough sentiment analysis. The outcome of this study laid the groundwork for future research on user preferences and perceptions. After that, we used a number of ML algorithms to refine the sentiment analysis and make the classifications more accurate. We used metrics like accuracy, recall, precision, F1-score, and confusion matrices are used for assessment of the performance of logistic regression, support vector machines (SVMs), neural networks, and Naive Bayes. Results showed that the suggested method worked, outperforming baseline methods in sentiment categorization by a wide margin. Another piece of evidence that the model could distinguish between good and bad feelings came from the ROC curve analysis. Finally, by presenting a strong and effective methodology, this research makes a significant contribution to the area of mobile app review mining. Researchers, app marketers, and developers all stand to benefit from a better understanding of user input and how to optimize app performance as a result of these discoveries.

Keywords: : Mobile applications, User reviews, Sentiment analysis, Machine learning.

INTRODUCTION

Businesses, organizations, and individuals have been producing data at an exponential rate, with an equally exponential growth in the diversity and volume of data. As a result of this expansion, new directions in data mining have emerged, with an emphasis on making data mining methods more precise, efficient, and scalable. There has been encouraging progress in several areas of data mining recently, considering NLP, image and speech recognition, prediction analytics, and machine learning (ML) methods like deep learning and neural networks. To process massive datasets in distributed computing settings, data mining is also becoming more integrated with big data technologies like Spark and Hadoop. The practice of automatically extracting subjective information from reviews or customer comments is called review mining, opinion mining, or sentiment analysis. The reviewers' feelings, thoughts, and attitudes towards the reviewed product, service, or entity are all part of this data. Organizations can use review mining to gain insight into consumer views and make better decisions. This method makes use of NLP techniques to sift through review content, identify themes and features, and categorize the reviews' sentiment (positive, negative, or neutral). Review mining is a natural language processing (NLP) method for analyzing reviews, ratings, comments, and other written texts that convey customer opinions, feedback, and sentiments. Finding out how people feel about a service, product, or brand is the main objective. The procedure entails classifying the text's sentiment as good, negative, or neutral utilizing algorithms and lexicon-based approaches. Review mining is a useful tool for gathering information that can help businesses improve their products and services, which in turn can boost customer happiness and sales [1, 2, 3, 4, 5].

The success or failure of a product or service in today's commercial environment is heavily dependent on consumer feedback. The proliferation of online shopping has resulted in an explosion of consumer reviews. Because of this, there is a pressing need for a streamlined approach to data analysis and interpretation. Since review mining automates sentiment analysis, it effectively solves this problem [6].

Review mining is usually a multi-step procedure. The first step in processing text data is to standardize its format and remove any extraneous information. The next step is feature extraction, which entails making numerical representations of the text data so that machine learning algorithms may exploit them. The next step is to do sentiment analysis utilizing specific ML and NLP approaches, including lexicon-based approaches, rule-based systems, and deep learning models. Sorting reviews into good, negative, or neutral categories according to the tone of the review content is an important part of sentiment analysis [7].

Customer preferences and opinions can be better understood with the help of sentiment analysis results. If a business wants to know how satisfied its customers are, what features work and what could use improvement, it can utilize review mining to do just that. You may also utilize review mining to keep an eye on how people feel about your brand, how they feel about it over time, and even anticipate trends [8].. Review mining is a great way for businesses to get client opinions and comments and analyze it. As the amount of reviews and comments from customers continues to rise, review mining offers a scalable and efficient way to extract valuable insights and make decisions based on data. Businesses can get a competitive edge and boost customer happiness using review mining, which will likely grow more essential as the field of natural language processing (NLP) continues to advance [9, 10, 11].

APPROACH

- Determine the review sources:
  - Decide which sources you want to mine, such as social media, review websites, or customer feedback forms.
- Collect the reviews:
  - Use web scraping tools to gather reviews from the selected sources.
- Pre-process the reviews:
  - Remove irrelevant information like usernames, timestamps, and other metadata.
- Segment the reviews:
  - Group reviews by product or service, sentiment, and other relevant factors.
- Analyse the reviews:
  - Use NLP techniques like sentiment analysis, topic modelling, and keyword extraction to identify patterns and insights.
- Identify common themes:
  - Look for frequently mentioned keywords and topics to identify common issues or areas of improvement.
- Summarize the findings:
  - Create a report that summarizes the key insights and recommendations for improving products or services.
- Take action:
  - Use the insights gained from review mining to make data-driven decisions and advance customer satisfaction.

METHODOLOGY

User reviews of mobile apps can be found in app stores preferably Google Play Store by Google Inc. and the App Store by the Apple Inc., and can consist of ratings, comments, and other user-generated content. Reviews like these are crucial to a mobile app's success because they provide prospective users a sense of what it's like to use the app and what they can expect from it.

Businesses and app developers frequently utilize user evaluations to learn how satisfied users are with their apps, find places to improve, and ultimately make the apps better for everyone. Reviews have a powerful impact on an app's exposure and trustworthiness. Positive reviews can boost downloads and user engagement, while negative reviews can damage the app's reputation and cause downloads to drop. NLP and ML methods are often used in the analysis of mobile app evaluations in order to classify the reviews' sentiment as positive, negative, or neutral. Finding recurring problems and complaints, seeing patterns over time, and keeping tabs on the brand's reputation are all possible with this data.

Because they reveal so much about the app's usability and performance, mobile app reviews are crucial to an app's success. Organizations may enhance their offers and boost user satisfaction by analyzing these reviews, which is a crucial component of app development and maintenance.

We want to provide a modified approach for mining user reviews of mobile apps in our present study. On the Google Play Store, we do app review mining.

APP REVIEW MINING ON GOOGLE PLAY STORE:

Google Play is full of apps, reviews, and ratings. The Google-play-scraper program can provide us with information about apps as well as reviews of them. There is a large number of applications available for research purposes. However, each app category serves a diverse user base and has distinct characteristics. To simplify matters, let's begin with some fundamental considerations [12, 13, 14].

We favor applications that have been available for an extended period, as this enables users to provide their comments organically. Minimizing the impact on advertising efforts is crucial, as applications are frequently updated, and the timing of evaluations might influence their content. One should ideally collect numerous reviews to facilitate informed decision-making. However, in practice, data may be constrained due to factors such as its volume or lack of accessibility. Notwithstanding this, we shall endeavor to attain our objectives. Consequently, we shall choose applications from the diverse categories that align with our specifications. The subsequent stages must be adhered to for the suggested hybrid approach.

First Thing to Do: Pick an App to Mine Reviews
Obtaining app data and storing it to a csv file is the second step.

```
# First Thing to Do: Choose a Mining App
The name of the application is "My App."
#Proceeding to Step 2: Collecting App Data and CSV File Saving.
file import
# Presuming you have a method called get_app_info() that retrieves the app's details, set app_info =
get_app_info(app_name).`
# Opening the "app_info.csv" file using the "mode='w'" option, then saving the data to a CSV file:
csv.writer(file) is the new writer.
The following string is being written: ['App Name', 'App Developer', 'Number of Downloads', 'Rating']
app_info['name'], app_info['developer']], app_info['downloads'], app_info['rating']] is called by writer.
```

The app whose ratings we wish to mine is initialized in the code by setting the app_name variable to that name. After that, we bring in the csv module and pretend to have a method named get_app_info() that, given the app's name, obtains all of its associated data.

Then, we acquire the app details by calling the get_app_info() function and adding them to the app_info dictionary. Lastly, we use the csv.writer() function to create a CSV file named app_info.csv and write the app information into it. Assuming that the app's information consists of the following fields: name, developer, download count, and rating, we set these as the first row of the CSV file and then write the values in the second row.

```
Step 3: Compile review data and save it as a csv file for future use in analysis
file import
# Here we're assuming you have a function named get_reviews() that retrieves the reviews for a certain app: ratings
= get_reviews(app_name)
# Opening the reviews in a CSV file with the 'open' function and the'mode='w' option:
csv.writer(file) is the new writer.
the code for the review in the reviews collection: writer.writerow([review['text'], review['rating']])
```

It is assumed that a function called get_reviews() exists to retrieve the reviews for a certain app, and the csv module is imported in this code. We obtain the app name reviews by calling the get_reviews() function and then store them in a reviews dictionary array. The next step is to use the csv.writer() function to create a CSV file named reviews.csv and then to write the reviews into it. The CSV file is structured with the fields for each review—text and rating—on the first row, and the values for each review follow in subsequent rows. This is based on the assumption that each review includes two fields.

```
# Step 4: Separating good reviews from bad ones using the existing dataset
input filename = csv
```

```
import good reviews
unfavorable_reviews
'reviews.csv' as file: open('reviews.csv', mode='r')
# Assuming the reviews are stored in a CSV file:
using csv.DictReader(file) as reader and looping through each row:
The rating is the integer of the row that contains the rating.
add the review to the positive reviews row if the rating is more than or equal to 4.
include negative reviews in the row where the review is located if the rating is less than or equal
to 2.
# Step 5: Findings from the experiment
print(f"Ratings: {len(ratings)} for positive reviews and {len(ratings)} for bad reviews)\}
```

We start by importing the CSV module and making two empty lists, positive_reviews and negative_reviews, in this program. Assuming a CSV file named reviews.csv contains the reviews.

After that, we use the CSV to read the file. Using the DictReader() method, a dictionary is generated for every row in the CSV file. We add the Review field to the positive reviews list after checking the Rating field of each review. If the Rating field is equal to or greater than 4, we consider the review positive. We add the Review field to the negative reviews list if the rating is 2 or lower, indicating a bad review.

Our discussion of the experimental results concludes with printing, using the len() function, the total number of good and negative evaluations. You can tweak the code to conduct sentiment analysis or topic modeling on the reviews, but for now it merely prints the amount of positive and bad reviews.

RESULTS AND DISCUSSION

Here we discuss about the results that are obtained from our experimentation.

ROC CURVE:

A basic method for evaluating a binary classifier's performance through different refinement thresholds is the ROC (Receiver Operating Characteristic) curve. An understanding of binary classifiers and the confusion matrix's role in quantifying their performance is necessary before delving into ROC curve analysis [15, 16, 17].

The properties of an item are used by a binary classifier to place it into one of two groups. The objective of our research is to use a labeled training dataset to train a classifier, and then use that model to predict the labels of unseen data using the attributes that were provided. The four results listed below provide a description of the classifier's outcome and performance:

- The classifier correctly identifies a positive occurrence as positive; this is known as a true positive (TP).
- For a classifier to be considered true negative (TN), it must correctly identify negative instances as negative.
- When a classifier makes a Type I error the incorrect prediction that an instance is positive and we call it as a false positive (FP).
- False Negative (FN): A classifier makes a Type II error when it incorrectly labels a positive case as negative.

Important performance metrics like F1-Score, Precision, and Recall are computed using confusion matrices that are in turn based on these outcomes. At different threshold values, the ROC curve graphically shows the relationship between the True Positive Rate (TPR) and the False Positive Rate (FPR). Insights regarding the classifier's ability to differentiate between the two classes across various decision limits are provided.

Notice the distribution of TPs, TNs, FPs, and FNs in Figure 1, which shows the link between expected and actual values. To evaluate the classifier's discriminative power and predictive accuracy, this picture is useful. The AUC is a popular summary statistic that measures the classifier's overall performance. If the AUC is 1, then the classifier is doing an excellent job of classifying the data; if it's 0.5, then it's just as well as guesswork. The accuracy of the classifier is proportional to how closely the curve tracks the upper left corner.

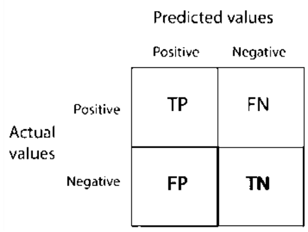

Figure 1: Confusion Matrix

LOGISTIC REGRESSION:

Logistic regression is a numerical technique that is applied for the purpose of analyzing situations involving binary categorization. This technique is often used to identify the relationship that exists between a dependent variable as well as one or more independent variables. In order to determine the probability of an event taking place, such as a client purchasing a product, it takes into account a number of predictors or characteristics [18, 19, 20].

Figure 2 represents the ROC curve for the logistic regression model. Where the X-axis representing the false positive rate and the Y-axis representing true positive rate.

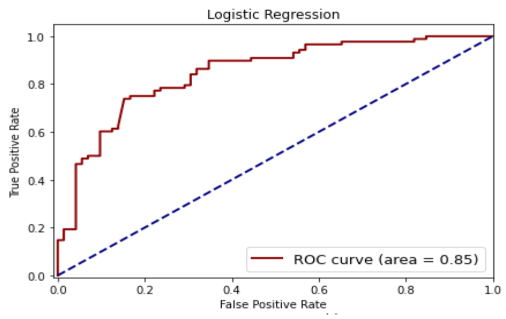

Figure 2: ROC Curve (Logistic Regression)

NAÏVE BAYES:

The following ROC curve of a Naive Bayes classifier offers a graphical representation of the classifier's performance and which is utilized to conduct a comparison with different classifiers. The AUC, of the ROC curve is a measure of the overall performance of the classifier. A larger AUC value indicates that the classifier exhibits greater classification ability. Naive Bayes classifiers that have higher AUC values have superior discrimination power and are more operational when it comes to distinguishing between positive and negative occurrences. As a result, the ROC curve is an extremely useful instrument for evaluating and improving the performance of Naive Bayes classifiers [21, 22, 23]. A representation of the ROC curve for the naïve Bayes algorithm can be found in Figure 3. In which the X axis reflects the number of false positives and the Y axis represents the rate of genuine positives to the test.

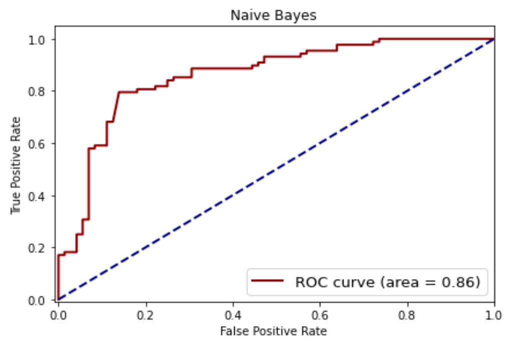

Figure 3: ROC Curve (Naïve Bayes)

SUPPORT VECTOR MACHINE:

The ROC curve of the Support Vector Machine (SVM) classifier offers a visual depiction of its performance, while the AUC of the ROC curve offers a measurement of the classifier' comprehensive performance. An increased AUC value shows improved classification performance. This suggests that the support vector machine SVM classifier possesses enhanced discrimination capacity when it comes to distinguishing between positive and negative examples. SVM classifiers can be evaluated and optimized with the help of the ROC curve, which is a valuable tool for evaluating and comparing the performance of SVM classifiers to that of other classifiers [24, 25, 26]. A representation of the ROC curve for the SVM model can be found in figure 4. It is important to note that the X-axis represents the false positive rate, while the Y-axis represents the genuine positive rate.

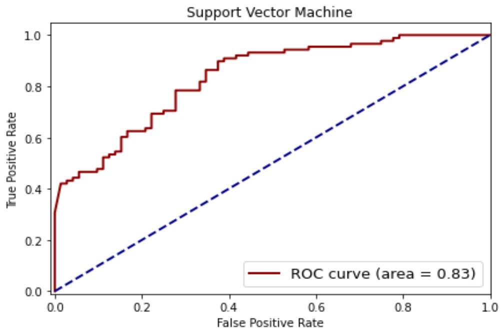

Figure 4: ROC Curve (Support Vector Machine)

NEURAL NETWORKS:

Artificial neural networks (ANN), which are invented after the consideration of structure and working of human brain and its neurons, it helps to give computers the ability to learn from given data and make relative predictions for a variety of tasks [27, 28, 29, 30]. The figure 5 representing ROC for NN.

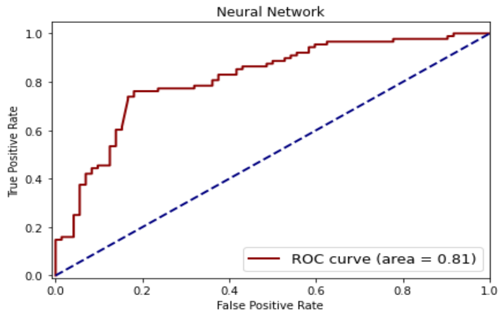

Figure 5: ROC Curve (Neural Network)

CONFUSION MATRIX:

It is likely to estimate the performance of a classification model by considering a table called the confusion matrix. This table provides a summary of the predictions made by the model in comparison to the actual values of the variable that is being evaluated. It comprises the amount of predictions that the model made that were TP, TN, FP, and FN. When it comes to determining how accurate and efficient a classification model is, the confusion matrix is a very helpful tool [31, 32].

An evaluation of the accuracy of the model's predictions can be carried out with the use of a confusion matrix when it comes to the context of logistic regression.

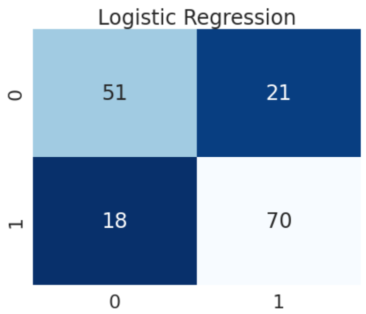

Figure 6: Confusion Matrix (Logistic Regression)

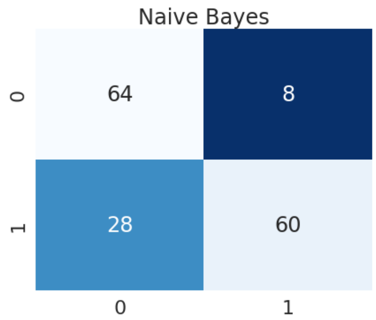

Figure 7: Confusion Matrix (Naïve Bayes)

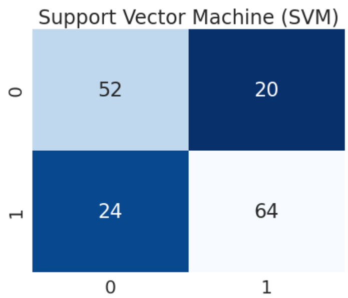

Figure 8: Confusion Matrix (Support Vector Machine)

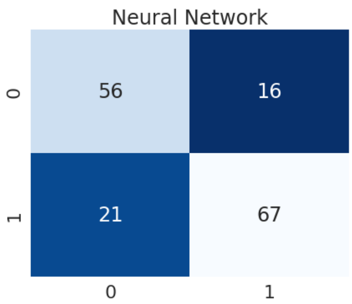

Figure 9: Confusion Matrix (Neural Network Algorithm)

ACCURACY

A classification model's accuracy is a statistic that evaluates the frequency with which it successfully predicts the class of an observation. Accuracy may also be referred to as precision [33]. The following equation is used for accuracy calculations.

$$Accuracy = \frac{No\ of\ correct\ Predictions}{No\ of\ total\ Predictions}$$

Precision: the first part of the F1 score

The reproducibility or repeatability of a measurement is the statistic that is used to determine precision. It refers to the metric that the results of repeated measurements of the same amount are consistent with one another [33]. The following equation is used obtaining precisions results.

$$Precision = \frac{No\ of\ True\ Positives}{No\ of\ True\ Positives + No\ of\ False\ Positives}$$

RECALL: the second part of the F1 score

Recall is a metric that is used in ML that quantifies the proportion of positive cases that are correctly classified as positive [33]. Two components make up the F1 score, which is a measurement of the overall accuracy of a machine-learning model. This component is one of the two components. Recall can be calculated using the following formula:

$$Recall = \frac{No\ of\ True\ Positives}{No\ of\ True\ Positives + No\ True\ False\ Negatives}$$

THE F1 SCORE: combining Precision and Recall:

The purpose of the F1 score is to facilitate the combination of the precision as well as recall metrics into a single assessment statistic that is more balanced. In addition to this, it is especially useful when dealing with data that is not evenly distributed. In order to determine the F1 score we have to take in consideration of the harmonic mean with precision and recall scores. When you need to strike a balance between accuracy and recall, this harmonic blend helps get a more comprehensive perspective of a model's performance [33]. This is especially helpful when you need to avoid false positives. As a result of the fact that it lends more weight to low values, the harmonic mean is utilized. This is extremely significant for both precision and memory. This is the formula that is used to calculate the F1 score is given below where p is precision and r is recall.

$$F1\ Score = 2 * \frac{\mathrm{p\ x\ r}}{\mathrm{p + r}}$$

Table 1. Precision, Recall, F1 score of Respective Algorithms (Parameters False: F, True: T, Accuracy: AC, Average: AV)

Algorithm	Para.	Precision	Recall	F1-score	Support
Logistic Regression	F	0.74	0.71	0.72	72
	T	0.77	0.80	0.78	88
	AC			0.76	160

	AV	0.75	0.75	0.75	160
Naive Bayes	F	0.70	0.89	0.78	72
	T	0.88	0.68	0.77	88
	AC			0.78	160
	AV	0.79	0.79	0.77	160
Support Vector Machine	F	0.68	0.72	0.70	72
	T	0.76	0.73	0.74	88
	AC			0.73	160
	AV	0.72	0.72	0.72	160
Neural Network	F	0.73	0.78	0.75	72
	T	0.81	0.76	0.78	88
	AC			0.77	160
	AV	0.77	0.77	0.77	160

CONCLUSION

The purpose of this research was to investigate a modified method of mobile application review mining. To do this, we utilized the Scraper package for the purpose of data extraction and implemented a binary classification model for the purpose of review analysis. Our major objective was to improve the efficiency and accuracy of review mining, with a particular emphasis on locating ideas, sentiments, and helpful insights that have the potential to greatly contribute to the enhancement of the quality of mobile applications. We have showed that mining user evaluations from app stores can generate valuable feedback for developers to tweak and improve their programs. This was accomplished through the utilization of advanced approaches and algorithms. It is common for the conventional approaches of review mining to be inadequate when it comes to managing the volume, complexity, and noise that are inherent in user-generated content. In order to streamline the process of identifying actionable insights, we utilized the Scraper package for efficient review gathering and combined it with algorithms for sentiment analysis and recommendation mining. This allowed us to significantly improve the efficiency of the process. Tokenization, stopword elimination, and stemming are examples of pre-processing techniques that were utilized in the execution of this strategy. These techniques helped to increase the accuracy of classification by minimizing the amount of noise in the data that was not relevant to the classification tasks. One of the most important aspects of our strategy was the utilization of a binary classifier, which allowed us to differentiate between evaluations that included ideas and those that did not include suggestions. Multiple metrics, such as accuracy, recall, F1-score, and the ROC curve, were utilized in order to assess the effectiveness of the classifier. When compared to more conventional approaches, the findings demonstrated a discernible and encouraging improvement in the accuracy of recognizing suggestion-based reviews. A detailed visual depiction of the classifier's performance over a variety of thresholds was provided by the ROC curve. This representation highlighted the classifier's capacity to strike a balance between the rates of true positives and false positives. Additionally, the value of the area under the curve (AUC) provided additional evidence that the classifier was effective. This value demonstrated that the model is able to differentiate between reviews that suggest something and reviews that do not suggest anything with a high degree of precision. In addition to this, the research highlights the significance of using sentiment analysis with review mining. It was possible for us to acquire a more in-depth comprehension of user feedback by classifying user reviews into three distinct categories: positive, negative, and neutral attitudes. The categorization, which was based on sentiment, offered insights into the level of customer contentment, pain points, and places that could use improvement. Our technique provides developers with a comprehensive tool that not only identifies user problems but also prioritizes them based on the intensity of the sentiment. This is accomplished by marrying this sentiment analysis with suggestion mining. The mobile application market is going to be significantly impacted by the implementation of this updated technique to mobile app review mining with substantial ramifications. The first benefit is that it provides developers with a solution that is both automatic and scalable, allowing them to continuously monitor user feedback and identify areas that may be improved in real time.

Through the usage of this technique, it is possible to personalize app updates to fit the demands of users, which ultimately results in increased levels of user happiness and retention rates for the app. Furthermore, the capability to filter suggestions from the enormous pool of reviews enables developers to concentrate on constructive comments, which can lead to significant changes in the functionality of the app as well as the user experience. The findings of our study have highlighted the need of employing a modified method to mobile app review mining. This technique involves utilizing the Scraper package for the purpose of efficient data gathering and binary classification for the purpose of recommendation mining. The incorporation of sentiment analysis allowed us to provide a comprehensive perspective of user feedback, which can be utilized to guide decisions on the development of mobile applications. Our technique appears to be helpful in enhancing the quality and usability of mobile applications, providing developers with a powerful tool that can help them maintain their competitive edge in a market that is always shifting. The findings of this study imply that our strategy is effective. In the future, research might investigate the possibility of incorporating deep learning techniques and broadening the scope of analysis to include multi-class classification and cross-platform review mining. This would further improve the accuracy and usefulness of the strategy that has been proposed.

REFERENCES

[1] Ikegwu, A.C., Nweke, H.F., Anikwe and C.V., "Recent trends in computational intelligence for educational big data analysis.," Iran Journal of Computer Science, vol. 7, no. 1, pp. 103-129, 2024.

[2] Mienye, Domor, Ibomoiye, Swart, T. G., Obaido and George, "Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications," Information, vol. 15, no. 9, p. 517, 2024.

[3] Arif, Zeravan and S. R. Zeebaree, "Distributed Systems for Data-Intensive Computing in Cloud Environments: A Review of Big Data Analytics and Data Management.," Indonesian Journal of Computer Science,, vol. 13, no. 2, 2024.

[4] Rajendran, S., Srinivas, S. Pagel and E., "Mining voice of customers and employees in insurance companies from online reviews: A text analytics approach.," Benchmarking: An International Journal, vol. 30, no. 1, pp. 1-22, 2024.

[5] Jim, J.R., Talukder, M.A.R., Malakar, P., Kabir, N. M.M., K. and M. Mridha, "Recent advancements and challenges of nlp-based sentiment analysis: A state-of-the-art review.," Natural Language Processing Journal, p. 100059., 2024.

[6] Rane, N. Liladhar, A. Achari, Saurabh and P. Choudhary., "Enhancing customer loyalty through quality of service: Effective strategies to improve customer satisfaction, experience, relationship, and engagement," International Research Journal of Modernization in Engineering Technology and Science, vol. 5, no. 5, pp. 427-452, 2023.

[7] Bharadwaj and Lakshay, "Sentiment analysis in online product reviews: mining customer opinions for sentiment classification.," Int J Multidiscip Res , vol. 5, no. 5, 2023.

[8] Hartmann, Jochen, M. Heitmann, C. Siebert and C. Schamp, "More than a feeling: Accuracy and application of sentiment analysis.," International Journal of Research in Marketing, vol. 40, no. 1, pp. 75-87, 2023.

[9] Asadabadi, M. Rajabi, M. Saberi, N. S. Sadghiani, O. Zwikael and E. Chang, "Enhancing the analysis of online product reviews to support product improvement: integrating text mining with quality function deployment," Journal of Enterprise Information Management, vol. 36, no. 1, pp. 275-302, 2023.

[10] Park and Jaehun, "Combined Text-Mining/DEA method for measuring level of customer satisfaction from online reviews.," Expert Systems with Applications, vol. 232, p. 120767, 2023.

[11] Wang, Juite, Jung-Yu, Lai, Lin and Yi-Hsuan, "Social media analytics for mining customer complaints to explore product opportunities.," Computers & Industrial Engineering, vol. 178, p. 109104, 2023.

[12] Elistiana, K. Melina, B. A. Kusuma, P. Subarkah and H. A. A. Rozaq, "Improvement of Naive Bayes Algorithm in Sentiment Analysis of Shopee Application Reviews on Google Play Store," Jurnal Teknik Informatika, vol. 4, no. 6, pp. 1431-1436, 2023.

[13] Islam, M. Jahidul, R. Datta and A. Iqbal, "Actual rating calculation of the zoom cloud meetings app using user reviews on google play store with sentiment annotation of BERT and hybridization of RNN and LSTM," Expert Systems with Applications, vol. 223, p. 119919, 2023.

[14] Mahmood and Ahsan, "Identifying the influence of various factor of apps on google play apps ratings," Journal of Data, Information and Management,, vol. 2, pp. 15-23, 2020.

[15] Nahm, Francis and Sahngun, "Receiver operating characteristic curve: overview and practical use for clinicians.," Korean journal of anesthesiology, vol. 75, no. 1, pp. 25-36, 2022.

[16] Fan, Jerome, S. Upadhye and A. Worster., "Understanding receiver operating characteristic (ROC) curves.," Canadian Journal of Emergency Medicine, vol. 8, no. 1, pp. 19-20, 2006.

[17] Pepe, Margaret and Sullivan, "Receiver operating characteristic methodology.," Journal of the American statistical association, vol. 95, no. 449, pp. 308-311, 2000.

[18] D. W. Hosmer Jr, S. Lemeshow and R. X. Sturdivant., Applied logistic regression., John Wiley & Sons, 2013.

[19] Menard and Scott., Applied logistic regression analysis, Sage, 106.

[20] Speelman and Dirk, "Logistic regression," Corpus methods for semantics: Quantitative studies in polysemy and synonymy, vol. 43, pp. 487-533, 2014.

[21] Webb, G. I, E. Keogh and R. Miikkulainen., "Naïve Bayes.," Encyclopedia of machine learning, vol. 15, no. 1, pp. 713-714, 2010.

[22] Yang and Feng-Jen, "An implementation of naive bayes classifier.," in In 2018 International conference on computational science and computational intelligence (CSCI), 2018.

[23] Murphy and K. P, "Naive bayes classifiers," University of British Columbia, vol. 18, no. 60, pp. 1-8, 2018.

[24] Stitson, J. A. E. M. O., Weston, A. Gammerman, V. Vovk and V. Vapnik, "Theory of support vector machines.," University of London, vol. 117, no. 827, pp. 188-191, 1996.

[25] Suthaharan, Shan, Shan and Suthaharan, "Support vector machine.," Machine learning models and algorithms for big data classification: thinking with examples for effective learning, pp. 207-235, 2016.

[26] Pisner, Derek, A., David and M. Schnyer., "Support vector machine.," In Machine learning Academic Press, pp. 101-121, 2020.

[27] Abdi, Hervé, Dominique, Valentin, Betty and Edelman., Neural networks, Sage, 1999.

[28] Gurney and Kevin, An introduction to neural networks, CRC press, 2018.

[29] Bishop, Chris and M, "Neural networks and their applications.," Review of scientific instruments, vol. 65, no. 6, pp. 1803-1832, 1994.

[30] Miikkulainen, Risto, Jason, Liang, Elliot, Meyerson, Aditya, Rawal, Dan, Fink, Olivier, Francon, Bala and Raju, "Evolving deep neural networks.," in In Artificial intelligence in the age of neural networks and brain computing, Academic Press, 269-287, 2024.

[31] Visa, Sofia, Brian, Ramsay, Anca, L, Ralescu, Esther, Van and D. Knaap, "Confusion matrix-based feature selection," Maics, vol. 710, no. 1, pp. 120-127, 2011.

[32] Susmaga and Robert, "Confusion matrix visualization.," in ntelligent Information Processing and Web Mining: Proceedings of the International IIS: IIPWM '04, Zakopane, Poland,, 2004.

[33] Bhor HN, Kalla M. TRUST-based features for detecting the intruders in the Internet of Things network using deep learning. Computational Intelligence. 2022; 38(2): 438–462.

[34] Pinjarkar, V. U. ., Pinjarkar, U. S. ., Bhor, H. N. ., Mahajan, Y. V. ., Patil, V. R. ., Rajput, S. D. ., Kothari, P. ., Ghori, D. ., & Bhabad, H. P. . (2023). Student Engagement Monitoring in Online Learning Environment. International Journal of Intelligent Systems and Applications in Engineering, 12(1), 292–298.

[35] Bhole, V. ., Bhor, H. N. ., Terdale, J. V. ., Pinjarkar, V. ., Malvankar, R. ., & Zade, N. . (2023). Machine Learning Approach for Intelligent and Sustainable Smart Healthcare in Cloud-Centric IoT. International Journal of Intelligent Systems and Applications in Engineering, 11(10s), 36–48.

[36] Terdale, J. V. ., Bhole, V. ., Bhor, H. N. ., Parati, N. ., Zade, N. ., & Pande, S. P. . (2023). Machine Learning Algorithm for Early Detection and Analysis of Brain Tumors Using MRI Images. International Journal on Recent and Innovation Trends in Computing and Communication, 11(5s), 403–415.

[37] H. N. Bhor and M. Kalla, "An Intrusion Detection in Internet of Things: A Systematic Study," 2020 International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2020, pp. 939-944, doi: 10.1109/ICOSEC49089.2020.9215365.

[38] Yacouby, Reda, Dustin and Axman., "Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models.," in the first workshop on evaluation and comparison of NLP systems, 2020.