Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

A Scraper Package Based Approach For Mobile Apps Review Mining And Analysis

Makarand Lotan Mali[1*], Nitin N. Patil[2]

[1]*R. C. Patel Institute Of Technology, Shirpur, Nimzari Road, Shirpur, 425405, India*
[2]*Svkms, College Of Engineering, Dahiwad, At Post Dahiwad, Shirpur, 425405, India*

This paper presents a novel method for mining mobile app evaluations that employs the "scrapper" package and powerful ML algorithms. The idea was to improve app marketing and development by extracting meaningful data from a large database of user reviews. Using a detailed sentiment analysis, the recovered reviews were classified as positive, negative, or neutral. The findings of this study paved the way for subsequent research on user preferences and perceptions. Following that, we applied a number of machine learning techniques to improve the sentiment analysis and categorization accuracy. The accuracy, recall, precision, F1-score, and confusion matrices were used to evaluate the performance of logistic regression, support vector machines (SVMs), neural networks, and Naive Bayes. The results demonstrated that the proposed strategy worked, surpassing baseline sentiment categorization methods by a significant margin. The ROC curve study provided more evidence that the model could distinguish between positive and negative feelings. Finally, by proposing a robust and successful methodology, this study contributes significantly to the field of mobile app review mining. These insights will aid researchers, app marketers, and developers alike by providing a better knowledge of user input and how to maximize app performance.

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

INTRODUCTION

In recent years, there has been significant growth in the volume, variety, and velocity of data generated by businesses, organizations, and individuals. This growth has led to the emergence of new trends in data mining, which are focused on enhancing the accuracy, efficiency, and scalability of data mining techniques. One of the modern trends in data mining is the use of machine learning (ML) algorithms, such as deep learning and neural networks, which have shown promising results in various applications, including image and speech recognition, natural language processing and predictive analytics. Another trend is the integration of data mining with big data technologies, such as Hadoop and Spark, to enable the processing of large-scale datasets in distributed computing environments. Review mining, also known as opinion mining or sentiment analysis, is the process of automatically extracting subjective information from reviews or customer feedback. This information includes opinions, emotions, and attitudes expressed by the reviewers towards a particular product, service, or entity. The goal of review mining is to provide insights into the perceptions of customers and to support decision-making by organizations. The process involves natural language processing techniques to analyze the text data, classify the sentiment expressed (positive, negative, neutral), and extract key features or topics discussed in the reviews. Review mining is an NLP-based approach to analyze customer opinions, feedback and sentiments expressed in written texts such as reviews, ratings, comments, etc. The goal is to gather insights and understand the perception of customers about a product, service, or brand. The process involves using algorithms and lexicon-based methods to categorize sentiments in the text, such as positive, negative, or neutral. The information gathered through review mining can help organizations to improve their offerings and enhance customer satisfaction, ultimately leading to increased sales [1, 2, 3, 4, 5].

In the modern business world, customer opinions play an essential role in determining the success of a product or service. With the rise of e-commerce, the availability of customer feedback has become abundant. This has created a need for an efficient method to analyze and make sense of the vast amount of information available. Review mining provides an effective solution to this problem by automating the process of sentiment analysis [6].

The process of review mining typically involves several steps. Firstly, the text data is pre-processed to remove any irrelevant information and standardize the text format. This is followed by feature extraction, which involves converting the text data into a numerical form that can be used by machine learning algorithms. Next, sentiment analysis is performed using various NLP techniques and machine learning algorithms, such as lexicon-based methods, rule-based systems, and deep learning models. The sentiment analysis process involves categorizing the reviews as positive, negative, or neutral based on the sentiment expressed in the text [7].

The outcomes of the sentiment analysis can be used to gain insights into customer opinions and preferences. For example, a company can use review mining to understand customer satisfaction levels, identify the strengths and weaknesses of its products, and make enhancements accordingly. Additionally, review mining can also be used to monitor brand reputation, track customer sentiment over time, and predict future trends [8].

The review mining is a valuable tool for organizations to gather and analyze customer feedback and opinions. With the growing volume of customer reviews and feedback, review mining provides a scalable and efficient method to extract meaningful insights and make data-driven decisions. As the field of NLP continues to improvement, review mining will likely become an increasingly important tool for businesses to gain a competitive edge and improve customer satisfaction [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 improve customer satisfaction.

METHODOLOGY

Mobile app reviews are user-generated comments, ratings, and feedback on mobile applications available on app stores such as the Apple App Store and Google Play Store. These reviews play a important role in the success of a mobile app as they deliver valuable insights into the user experience and help potential users to make informed decisions about downloading and using the app.

Mobile app reviews are often used by app developers and businesses to understand user satisfaction levels, identify areas for improvement, and make changes to the app to enhance the user experience. Positive reviews can increase an app's visibility and credibility, leading to more downloads and user engagement, while negative reviews can harm the app's reputation and result in decreased downloads.

The process of analyzing mobile app reviews typically involves NLP and machine learning techniques to categorize the sentiment expressed in the reviews as positive, negative or neutral. This information can then be used to identify common complaints and issues, track trends over time, and monitor brand reputation.

The mobile app reviews play an important role in the success of a mobile app and provide valuable insights into the user experience. Analyzing these reviews is an important part of app development and maintenance and can help organizations to improve their offerings and increase user satisfaction.

In current research work we aim to propose hybrid technique for mining the reviews of mobile app users. We perform app review mining on Google’s Play Store.

APP REVIEW MINING ON GOOGLE PLAY STORE:
Google Play has plenty of apps, reviews, and scores. With the Google-play-scraper package, we can get information about apps and reviews of them. There is a vast selection of applications available for studying purposes. However, each app category caters to different user groups and possesses unique characteristics. To simplify things, let's start with some basic considerations [12, 13, 14].

We prefer apps that have been in circulation for some time, as it allows users to offer their feedback naturally. It's important to minimize the impact on advertising strategies, as apps are regularly updated, and the timing of reviews can affect their content. Ideally, one should gather as many reviews as possible to make informed decisions. Nevertheless, in reality, data can be limited due to various reasons such as its size or inaccessibility. Despite this, we will strive to achieve our goals. Therefore, we will select apps from the variety of category that match our requirements.

Following steps are to be followed for proposed hybrid technique.

Step 1: Selection of app for mining reviews

Step 2: Getting App information and saving information in csv file.

```
# Step 1: Selection of app for mining reviews

app_name = “My App”

# Step 2: Getting App information and saving information in csv file.

import csv

# Assuming you have a function to get the app information called get_app_info()

app_info = get_app_info(app_name)

# Saving app information in a CSV file

with open('app_info.csv', mode='w') as file:

    writer = csv.writer(file)

    writer.writerow(['App Name', 'App Developer', 'Number of Downloads', 'Rating'])

    writer.writerow([app_info['name'], app_info['developer'], app_info['downloads'],
app_info['rating']])
```

In this code, we first set the app_name variable to the name of the app we want to mine reviews for. Then, we import the csv module and assume that we have a function called get_app_info() that retrieves the information about the app based on its name.

Next, we call get_app_info() function to get the app information and save it into a dictionary app_info. Finally, we create a CSV file called app_info.csv and write the app information into it using the csv.writer() function. We assume that the app information includes the app name, developer, number of downloads, and rating, and we write these fields as the first row of the CSV file, followed by the values in the second row.

```
# Step 3: Creation of dataset of reviews and store information in csv file for further analysis
import csv
# Assuming you have a function to get the reviews for a given app called get_reviews()
reviews = get_reviews(app_name)
# Saving the reviews in a CSV file
with open('reviews.csv', mode='w') as file:
    writer = csv.writer(file)
    writer.writerow(['Review', 'Rating'])
    for review in reviews:
        writer.writerow([review['text'], review['rating']])
```

In this code, we import the csv module and assume that we have a function called get_reviews() that retrieves the reviews for a given app. We call get_reviews() function to get the reviews for the app_name and store them in a list of dictionaries called reviews. Next, we create a CSV file called reviews.csv and write the reviews into it using the csv.writer() function. We assume that each review has a text and a rating field, and we write these fields as the first row of the CSV file, followed by the values for each review in succeeding rows.

```
# Step 4: Tracing out positive and negative reviews from the available dataset
import csv
positive_reviews = []
negative_reviews = []
# Assuming the reviews are stored in a CSV file called 'reviews.csv'
with open('reviews.csv', mode='r') as file:
    reader = csv.DictReader(file)
    for row in reader:
        rating = int(row['Rating'])
        if rating >= 4:
            positive_reviews.append(row['Review'])
        elif rating <= 2:
            negative_reviews.append(row['Review'])
# Step 5: Experimental results
print(f"Positive reviews:  {len(positive_reviews)}")
print(f"Negative reviews:  {len(negative_reviews)}")
```

In this program, we initially import the CSV module and create two empty lists called positive_reviews and negative_reviews. We assume that the reviews are stored in a CSV file called reviews.csv.

Next, we read the CSV file using the CSV. DictReader() function, which creates a dictionary for each row in the CSV file. We check the Rating field of each review and if it is greater than or equal to 4, we consider the review as positive and append the Review field to the positive_reviews list. If the rating is less than or equal to 2, we consider the review as negative and append the Review field to the negative_reviews list.

Finally, we print the number of positive and negative reviews using the len() function and discuss the experimental results. Note that the code only prints the number of positive and negative reviews, but you could modify it to perform further analysis on the reviews, such as sentiment analysis or topic modeling.

RESULTS AND DISCUSSION

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

ROC CURVE:

An ROC (Receiver Operating Characteristic) curve is a valuable visual aid for assessing how well a binary classifier performs at different discrimination thresholds. Before diving into ROC curves, it's important to understand what a binary classifier is and how the confusion matrix is used to evaluate its performance. In binary classification, you're presented with a set of objects and asked to divide them into two groups based on their characteristics [15, 16, 17].

Our goal is to train a classifier using the training dataset and then use it to predict the labels of new examples based solely on their features. While the classifier may not always predict the correct labels, it can produce one of four promising outcomes:

- True Positive (TP): The classifier correctly predicts a positive item as positive.
- True Negative (TN): The classifier correctly predicts a negative item as negative.
- False Positive (FP): The classifier incorrectly predicts a negative item as positive, which is known as a type I classification error.
- False Negative (FN): The classifier mistakenly predicts a positive item as negative, which is known as a type II classification error.

The figure 1 represents the relationship among the predicted values and actual values.

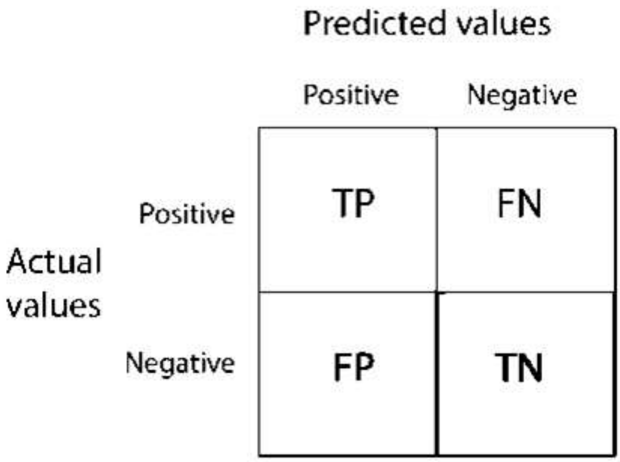

Figure 1: Confusion Matrix

LOGISTIC REGRESSION:
Logistic regression is a numerical technique utilized to analyze binary classification problems. This method models the connection between a reliant variable and one or several independent variables. It calculates the likelihood of an event occurring, such as a customer buying a product, based on a set of predictors or features [18, 19, 20].

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

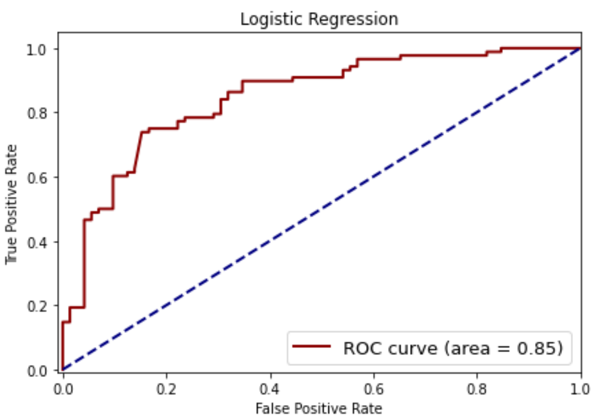

Figure 2: ROC Curve for Logistic Regression

NAÏVE BAYES:
The ROC curve for a Naive Bayes classifier provides a visual representation of its performance and can be used to compare it to other classifiers. The AUC (Area Under the Curve) of the

ROC curve provides a measure of the total performance of the classifier, with a higher AUC value indicating better classification performance. Naive Bayes classifiers with higher AUC values have better discrimination power and are more operational at distinguishing between positive and negative instances. Therefore, the ROC curve is a valuable tool for assessing and optimizing the performance of Naive Bayes classifiers [21, 22, 23]. Figure 3 represents the ROC curve for the naïve Bayes algorithm. Where X-axis represents the false positive rate and Y-axis represents the true positive rate.

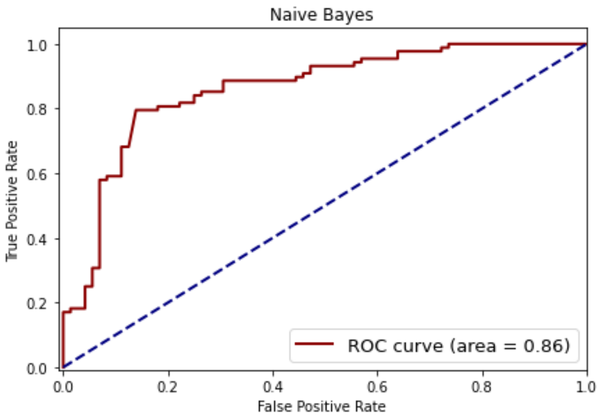

Figure 3: ROC Curve for Naïve Bayes

SUPPORT VECTOR MACHINE

The SVM classifier's ROC curve provides a visual representation of its performance, and the AUC (Area Under the Curve) of the ROC curve provides a measure of the classifier's overall performance. A higher AUC value indicates better classification performance, indicating that the SVM classifier has better discrimination power in distinguishing between positive and negative instances. The ROC curve is a valuable tool for assessing and optimizing the performance of SVM classifiers and comparing them to other classifiers [24, 25, 26]. The figure 4 representing the ROC curve for Support Vector Machine model. Where X-axis representing the false positive rate and Y-axis representing true positive rate.

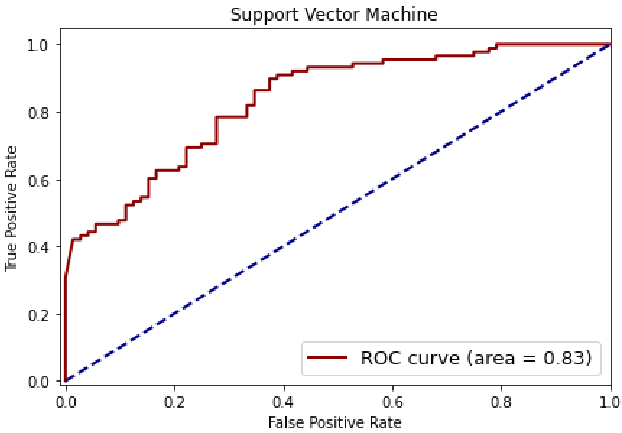

Figure 4: ROC Curve for Support Vector Machine

NEURAL NETWORKS

Inspired by the human brain, neural networks enable computers to learn from data and make predictions for a variety of tasks [27, 28, 29, 30].

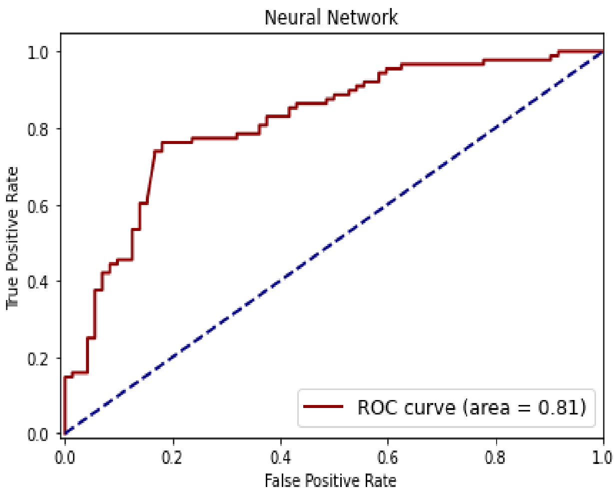

Figure 5: ROC Curve for Neural Network

CONFUSION MATRIX:

The confusion matrix is a table that allows for the evaluation of a classification model's performance by providing a summary of its predictions compared to the true values of the target variable. It includes the number of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions made by the model. The true positive and true negative values indicate the number of accurate predictions made by the model for each class, while the false positive and false negative values indicate the number of incorrect predictions made. The confusion matrix is a valuable tool for assessing the accuracy and effectiveness of a classification model [31, 32].

In the context of logistic regression, a confusion matrix can be used to evaluate the accuracy of the model's predictions. The confusion matrix shows the number of observations that were correctly classified as positive or negative (true positives and true negatives) as well as the number of observations that were incorrectly classified (false positives and false negatives). By examining the confusion matrix, one can determine the sensitivity, specificity, precision, and accuracy of the model.

Figure 6: Confusion Matrix for Logistic Regression

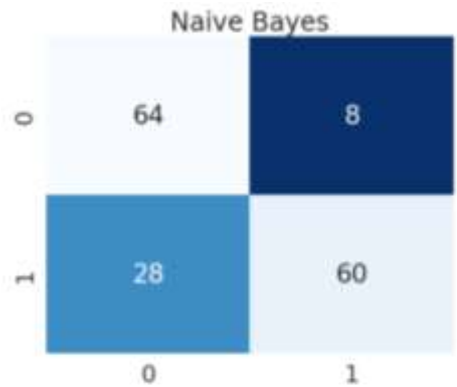

Figure 7: Confusion Matrix for Naïve Bayes Algorithm

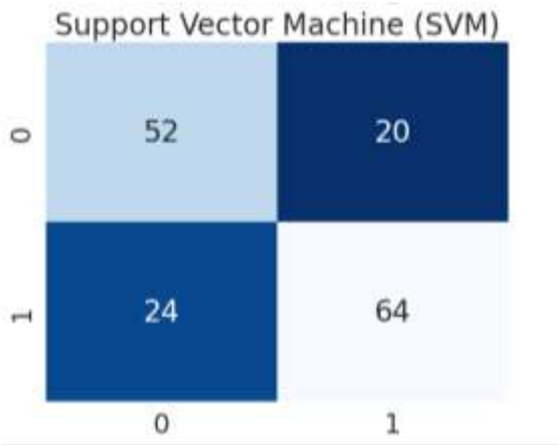

Figure 8: Confusion Matrix for Support Vector Machine

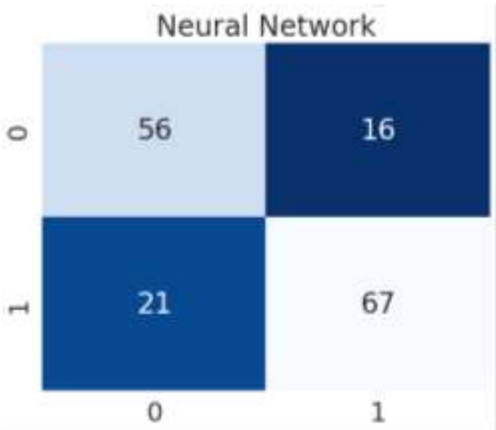

Figure 9: Confusion Matrix for Neural Network Algorithm

ACCURACY

Accuracy is a metric that assesses the frequency with which a classification model accurately predicts the class of an observation. To calculate accuracy, you divide the number of correct predictions by the total number of predictions [33].

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

Precision: the first part of the F1 score

Precision is the metric of reproducibility or repeatability of a measurement. It is the metric to which repeated measurements of the same quantity give the same results [33].

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

RECALL: the second part of the F1 score
Recall is a machine learning metric that measures the proportion of positive instances that are correctly identified as positive [33]. It is one of the two components of the F1 score, which is a measure of the overall accuracy of a machine-learning model. The formula for recall is as follows:

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

THE F1 SCORE: combining Precision and Recall:
The F1 score is designed to merge the precision and recall metrics into a single, balanced evaluation metric. Moreover, it is particularly effective when dealing with imbalanced data. The F1 score is defined as the harmonic mean of precision and recall. This harmonious blend helps capture a more comprehensive view of a model's performance, especially when you need to strike a balance between precision (the ability to avoid false positives) and recall (the ability to capture true positives while minimizing false negatives) [33]. The F1 score is calculated by taking the harmonic mean of precision and recall. The harmonic mean is used because it gives more weight to low values, which is important for precision and recall. The formula for the F1 score is as follows:

$$\text{F1 Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Table 1. Precision, Recall, F1 score of Respective Algorithms

Algorithm	**Parameter**	**Precision**	**Recall**	**F1-score**	**Support**
Logistic Regression	FALSE	0.74	0.71	0.72	72
	TRUE	0.77	0.80	0.78	88
	Accuracy			0.76	160
	Macro Avg	0.75	0.75	0.75	160
	Weighted Avg	0.76	0.76	0.76	160

Naive Bayes	FALSE	0.70	0.89	0.78	72
	TRUE	0.88	0.68	0.77	88
	Accuracy			0.78	160
	Macro Avg	0.79	0.79	0.77	160
	Weighted Avg	0.80	0.78	0.77	160
Support Vector Machine	FALSE	0.68	0.72	0.70	72
	TRUE	0.76	0.73	0.74	88
	Accuracy			0.73	160
	Macro Avg	0.72	0.72	0.72	160
	Weighted Avg	0.73	0.72	0.73	160
Neural Network	FALSE	0.73	0.78	0.75	72
	TRUE	0.81	0.76	0.78	88
	Accuracy			0.77	160
	Macro Avg	0.77	0.77	0.77	160
	Weighted Avg	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. In summary, 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] 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.

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

[29] 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.

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

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

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

[33] 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.