Success Prediction and Recommendation Systems for Google Apps Based on Rating and Sentiment Discrepancies

1st Najla Zuhir
College of Computing and IT
University of Doha for Science and Technology
Doha, Qatar

2nd Amna Mohammad Salim
College of Computing and IT
University of Doha for Science and Technology
Doha, Qatar

3rd Parvathy Premkumar
College of Computing and IT
University of Doha for Science and Technology
Doha, Qatar

***Abstract*—As of recently, there has been a huge increase in the number of mobile applications and this has managed to make app marketplaces like the google play store and the apple play store highly competitive. With countless apps available, most users struggle to select the right one, whereas, developers face challenges in prioritizing success factors and adapting to current market trends. This project aims to identify successful Google applications and proposes an effective solution to the above, two integrated systems: a user-centric app recommendation system and a developer-centric success prediction model. Both systems address rating-sentiment discrepancies to provide precise and actionable insights.**

By leveraging sentiment analysis and predictive modeling, the project enhances app evaluation and delivers tailored recommendations for users. Within our analysis, the decision tree model achieved the highest metric values, classifying the success level of applications, outperforming the rest of the models. Additionally, to make the system more user-friendly, a Streamlit-based web application has been developed that enables interactive visualizations, success predictions, and personalized recommendations, bridging the gap between user sentiment and app performance.

***Index Terms*—Mobile App Evaluation, Rating-Sentiment Discrepancy, Sentiment Analysis, App Success Prediction, Content-Based Filtering, Machine Learning Models, User Engagement, Discrepancy Analysis, Predictive Modeling, Streamlit Web Application**

I. Introduction

The mobile app marketplace is a highly competitive space, with the platforms being a host of millions of apps, like the google play store and the apple app store. Customers and users in search for their perfect application, face difficulties selecting the right application from countless options. Software developers encounter an equally significant challenge of making decisions that can affect the overall status and rating of their app. Understanding the key drivers of app success such as user engagement, user review, and the frequency of updates is crucial to succeed in this constantly evolving market.

Also, accurately assessing an app's performance is challenging as app ratings and review sentiments often mismatch. Many applications do not attract users or generate sufficient revenue due to poor design, usability issues, or neglect of user feedback [1]. Despite containing valuable insights such as user sentiment and actionable suggestions, user feedback often remains underutilized. Developers and business managers often overlook the potential of reviews, missing the opportunity to enhance app performance or inform strategic decisions like recommendation systems.

This project addresses these problems through two integrated strategies. The first is the Success Prediction Model, a developer-focused system that categorizes apps as Successful, Moderately Successful, or Unsuccessful based on key features such as User Engagement, Rating Review Discrepancy, and Update Frequency. The second is the Recommendation System, a user-centric tool that suggests a number of apps tailored to user preferences by considering category selections and acceptable discrepancy thresholds, and these recommendations align with user expectations.

The project carried out uses machine learning techniques to classify apps and generate insights, using features such as user engagement, ratings, and sentiment alignment. As studies have shown, understanding these dynamics can significantly reduce application failure rates by enabling developers to align their efforts with user preferences and also solve the underlying issue of mismatch of ratings versus written review [2]. By addressing these challenges, this project introduces a framework for reliable app success prediction and recommendation systems, offering an efficient solution for actionable insights for users and developers in today's ever-changing app ecosystem.

II. RELATED WORK

It is evident that discrepancies between app ratings and review sentiments impact the overall success of an application. Past studies have effectively explored the relationship between app performance and user feedback, emphasizing the importance of aligning user sentiments with app metrics to ensure accurate evaluations. A notable study, Fault in Your Stars: An Analysis of Android App Reviews highlights that approximately 20 percent of user reviews show inconsistencies between star ratings and the sentiments expressed in textual

Identify applicable funding agency here. If none, delete this.

feedback. These mismatches can distort an app's overall rating, subsequently affecting user perceptions and download decisions. The study highlights the importance of aligning textual feedback with star ratings for a more accurate representation of user satisfaction [3].

Sentiment analysis has emerged as a powerful tool for addressing these discrepancies outlined above. In a Comparative Sentiment Analysis of App Reviews by Ranjan and Mishra (2020), the authors explore how user sentiments influence app success. By using machine learning algorithms for sentiment classification, the study identifies behavioral patterns and sentiment trends that contribute to app performance. This research highlights the potential of sentiment analysis to predict app success and improve app quality, which closely resembles with the overall goal of this project [4].

Deep learning techniques have also been implemented to improve the reliability of app evaluations. For instance, Discrepancy Detection between Actual User Reviews and Numeric Ratings of Google App Store Using Deep Learning by Sadiq et al. (2021) investigates mismatches between review sentiments and numeric ratings. This work makes use of deep learning algorithms to detect and address these discrepancies, filling a critical gap in app review analysis and ensuring a more robust evaluation process [5].

Along with the emphasis on improving evaluation results, recommendation systems have gained popularity for their ability to prioritize successful applications based on certain features and categories. A study titled 'Machine Learning Based Recommendation System for Android App' proposes a framework for suggesting apps to users based on metadata and user preferences. By predicting app success through machine learning algorithms, the system aligns recommendations with user interests, demonstrating the importance of combining sentiment analysis and predictive modeling in creating effective recommendation systems [6].

The main goal of all these studies have been to highlight the value of sentiment analysis in understanding user feedback, aligning sentiments with ratings, and enhancing app evaluation systems. By addressing the discrepancies between review sentiments and ratings, app recommendation systems can provide more accurate outcomes, increasing user satisfaction and app success. Building upon these extensive findings, this project proposes a framework that integrates success prediction and recommendation systems, leveraging sentiment-driven insights to address the gaps identified in existing literature.

III. EXPERIMENT METHODOLOGY

The methodology for this project follows a clear and structured approach to predict app success and recommend apps in the Google Play Store. By combining data collection, preprocessing, feature engineering, and model development, the project aims to address the gap between app ratings and user sentiments. This section discusses the key steps taken to achieve meaningful insights and actionable outcomes.

A. Data collection

This study utilized two datasets sourced from Kaggle to implement sentiment-driven strategies for app success prediction and personalized recommendations, focusing on addressing rating-sentiment discrepancies. The first dataset, Google Play Store Apps, contained over 10,000 applications with 13 features, including Category, Rating, Reviews, Installs, and Price. The second dataset, User Reviews, comprised over 60,000 reviews across five attributes such as Translated_Review, Sentiment, and Sentiment_Polarity [7].

To create a unified dataset for analysis, these datasets were merged using the App column as the primary key. This integration enabled app-specific features to connect user feedback, giving a complete picture of app performance. The merged dataset was the starting point for all the next steps, including Exploratory Data Analysis (EDA), feature engineering, and building our models.

These datasets were selected for their comprehensiveness and relevance, providing valuable insights into the discrepancies between app ratings and user sentiments. Their size and diversity ensured statistically meaningful and generalizable findings on app performance and recommendations.

B. Data Pre-processing

The datasets were carefully preprocessed to ensure data quality and consistency. Missing values, redundancies, and irrelevant attributes were addressed during cleaning. Noise, such as special characters in textual data, was removed. Numerical features like Size and Price were normalized to ensure uniform scaling, improving model performance and reliability. After all the preprocessing we had about 9800+ records with 14 columns.

C. Feature Engineering

To enhance analysis and enable effective modeling, key features were engineered, and they are:

1. Sentiment Rating

This is a normalized version of Sentiment Polarity, which originally ranges from -1 to 1. It was rescaled to a 1–5 range to align with the app's star ratings for direct comparison.

Formula Representation:
Sentiment Offset:

$$\text{Sentiment Offset} = \text{Sentiment Polarity} - \min(\text{Sentiment Polarity}) \tag{1}$$

Sentiment Range:

$$\text{Sentiment Range} = \max(\text{Sentiment Polarity}) - \min(\text{Sentiment Polarity}) \tag{2}$$

Normalize Sentiment Rating:

$$\text{Sentiment Rating} = \left(\frac{\text{Sentiment Offset}}{\text{Sentiment Range}}\right) \times 4 + 1 \tag{3}$$

This transformation ensures easy comparison between user sentiment and the app's star ratings, providing insights into rating-sentiment alignment and discrepancies.

2. Discrepancy

This feature represents the calculated difference between the Sentiment Rating and the App Rating, helping to identify mismatches between user sentiments and their assigned star ratings.

Formula Representation:

$$\text{Discrepancy} = |\text{Rating} - \text{Average\_Sentiment\_Rating}|$$

This absolute difference highlights the misalignment between user sentiments and ratings, aiding in understanding user perceptions and improving app evaluation.

3. User Engagement

This feature captures the relationship between reviews and installs, showing a strong positive correlation ($r = 0.53$) based on Pearson analysis. It highlights that apps with higher installs tend to receive more user reviews, reflecting increased user interaction.

Formula Representation:

$$\text{Reviews\_Installs\_Interaction} = \log(1 + \text{Reviews} \times \text{Installs})$$

The logarithmic transformation ensures that the feature remains interpretable and scales appropriately, capturing the combined impact of reviews and installs to reflect user interaction.

4. Success Category

This feature categorizes apps into three groups—*Successful*, *Moderately Successful*, and *Unsuccessful*—based on user engagement, discrepancy metrics, and update recency. The classification criteria use the median as a robust measure to handle outliers and skewed data distributions effectively. Additionally, a tolerance approach allows flexibility, accommodating real-world variability in defining success categories. The table below provides a detailed breakdown of the criteria and indications for each category.

This categorization aligns with research findings that highlight the interconnected relationship between installs, ratings, and reviews for Google Play Store applications. Apps with higher install counts tend to attract more user reviews, reflecting increased user engagement. Similarly, well-rated apps often experience higher installation rates, as users perceive them to be of better quality. Additionally, user feedback frequently drives improvements, leading to higher ratings for apps with more reviews [8]. Frequent app updates also play a critical role in app success, as app stores prioritize freshness when determining rankings. Updates enhance user experience by addressing bugs (63%) introducing new features (35%), and improving existing ones (30%), further boosting an app's popularity and performance [9]. These insights directly informed the creation of the Success Category feature.

TABLE I
SUCCESS CATEGORIES FOR APPS

Category	**Criteria and Indication**
Successful Apps	• Reviews-Installs Interaction: within 15% of median. • Discrepancy: within 20% of median. • Recent update (last 1.5 years). • High engagement, low discrepancies, and active maintenance.
Moderately Successful Apps	• Reviews-Installs Interaction: 50–100% of median (±15%). • Discrepancy: within 1.5 times median (±20%). • Update in the last 3 years. • Moderate engagement and acceptable alignment.
Unsuccessful Apps	• Fails to meet above criteria. • Low engagement, high discrepancies, or outdated updates (older than 3 years).

D. Final Dataset Features

After completing preprocessing and feature engineering, the dataset consisted of 9800+ records with 14 columns, providing a robust foundation for analysis:

TABLE II
FEATURES AND DESCRIPTIONS

No.	**Feature**	**Description**
1	App	Application name
2	Category	App category (e.g., GAME, TOOLS)
3	Rating	User rating (1 to 5)
4	Reviews	Number of user reviews
5	Size	App size (in MB)
6	Installs	Downloads or installs
7	Type	Paid or Free
8	Price	Cost (in USD)
9	Content_Rating	Target age group
10	Last Updated	Last update date
11	Avg. Sentiment Rating	Avg. sentiment score aligned to star ratings
12	Discrepancy	Star rating vs. sentiment rating difference
13	User Engagement	Activity based on reviews and installs
14	Success Category	Success categorization (e.g., Successful)

E. Data Visualization

Visualizations were utilized to explore key trends and provide an overview of the dataset. Additionally, specific visualizations were created to support the project's objective of investigating discrepancies between app ratings and user reviews.

The bar chart and pie chart illustrate the distribution of apps across various categories. Dominant categories include FAMILY, GAME, and TOOLS, indicating areas of higher app concentration. These visualizations provide valuable insights into the diversity and focus of app categories in the

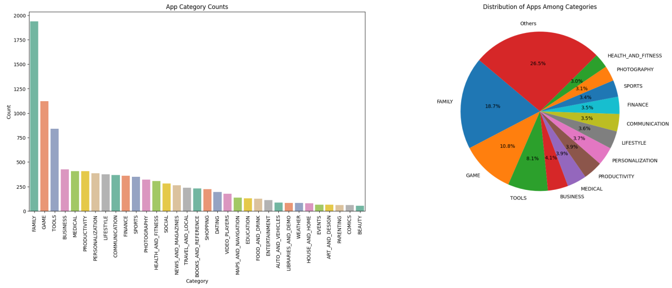

Fig. 1. App Category Distribution.

marketplace.

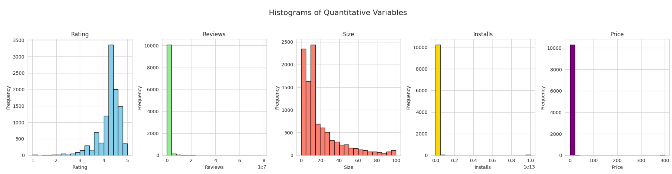

Fig. 2. Histograms of key quantitative variables.

1) **Rating:** Most apps have ratings clustered around 4 to 5 stars, indicating a general trend of positive user feedback.
2) **Reviews:** A large proportion of apps have relatively low review counts, while a few apps have significantly higher review counts, suggesting variability in user engagement.
3) **Size:** App sizes vary widely, with a higher concentration of apps in the lower size range, reflecting optimization for mobile devices.
4) **Installs:** The distribution is heavily skewed, with most apps having lower install counts and only a few achieving very high download numbers.
5) **Price:** Most apps are free or low-priced, with a few high-priced outliers, consistent with the dominance of free-to-use apps in the marketplace.

These visualizations provide insights into the characteristics of apps and serve as a foundation for understanding trends and relationships in the dataset.

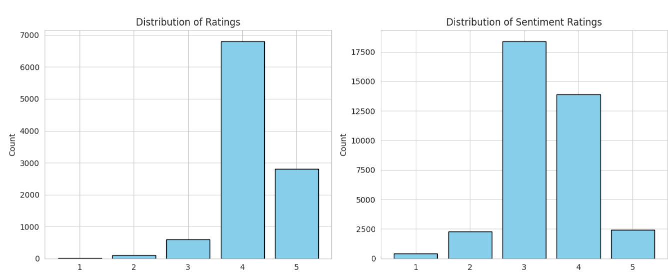

Fig. 3. Discrepancies between Ratings and Sentiment Ratings.

Star ratings are heavily skewed towards higher values (4 and 5), indicating a positive bias in user ratings, while sentiment ratings are centered around 3 and 4, offering a more balanced distribution. This discrepancy suggests that numeric ratings may overestimate user satisfaction, as they don't fully capture the nuanced feedback expressed in written reviews. Sentiment ratings, on the other hand, provide a more grounded perspective, highlighting the importance of combining both metrics for an accurate assessment of user satisfaction.

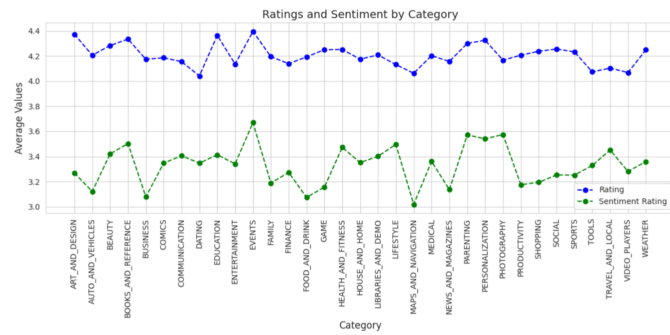

Fig. 4. Average Ratings and Sentiment by Category. The chart highlights that while Ratings consistently remain high, Sentiment Ratings vary significantly, uncovering discrepancies in user feedback quality among categories.

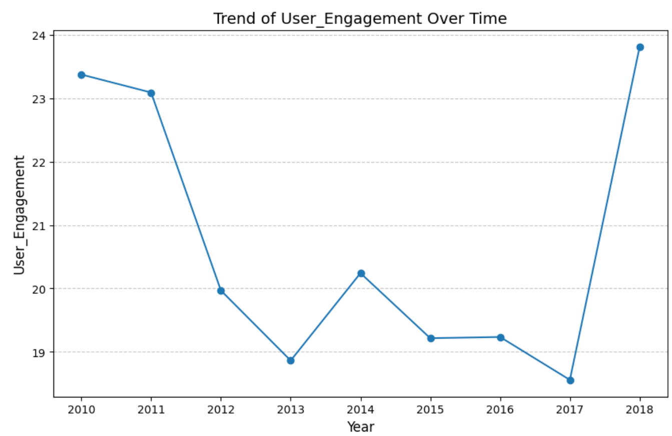

Fig. 5. User Engagement Over Time in Relation to App Updates. The chart shows a significant increase in interactions for apps updated more recently, with a sharp peak in 2018, indicating the strong impact of timely updates on user engagement. The trend line effectively captures fluctuations in user activity over time, showcasing steady declines in earlier years followed by a dramatic rise in 2018, underscoring the importance of frequent updates in driving app success and maintaining user interest.

F. App Success Classification Models and Evaluation Metric

To predict app success categories (*Successful, Moderately Successful, and Unsuccessful*), a range of models from simple to complex were implemented:

1) **Logistic Regression:** Served as a baseline for performance comparison.
2) **Decision Tree and Random Forest:** Medium-complexity models designed to capture non-linear patterns in the data.

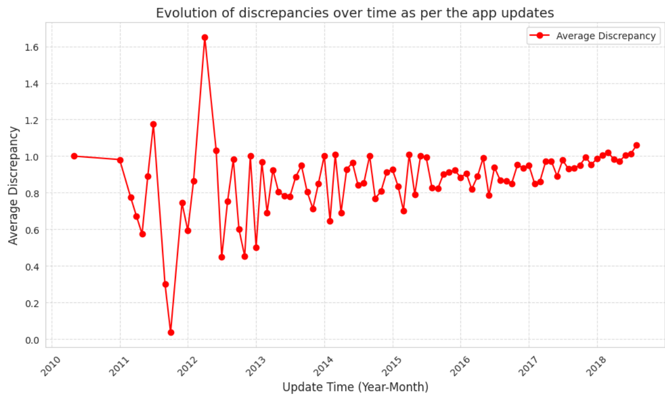

Fig. 6. Evolution of Average Discrepancies Over Time in Relation to App Updates. The chart shows significant variability in average discrepancies between app ratings and user sentiments from 2010 to 2018, indicating inconsistent alignment.

3) **Feed Forward Neural Network:** A deep learning model optimized using Keras Tuner for hyperparameter selection.
   - Hyperparameters included the number of layers, units per layer, activation functions, dropout usage, optimizer, and learning rate.
   - The model was trained for up to 20 epochs, with hyperparameter tuning conducted over 10 trials to ensure robust performance evaluation.

Given the highly imbalanced dataset, a hybrid sampling technique (*SMOTEENN*) was applied to the simpler and medium-complexity models. This method generated synthetic samples and removed noisy data, effectively balancing the dataset.

Evaluation Metrics: Model performance was assessed using Accuracy, Recall, Precision, F1 Score, and Confusion Matrix to ensure a comprehensive evaluation.

G. App Recommendation System

The recommendation system employs a content-based filtering approach, utilizing normalized and weighted features such as *Rating, Sentiment Polarity, Discrepancy, and Reviews-Installs Interaction*. It also integrates the target feature, *Success Category*, to prioritize successful apps.

1) This app-specific approach eliminates the need for user-specific data, delivering precise and relevant recommendations tailored to user-defined preferences such as category, number of apps, and discrepancy threshold.
2) The algorithm enhances accuracy by assigning feature weights based on their significance, ensuring recommendations effectively align with user expectations.

H. Web Application Design Using Streamlit

An interactive web application was developed using Streamlit, a Python library designed for creating data-driven applications. The application features a user-friendly interface with three main pages:

1) **Graphs:** Enables users to visualize relationships between features. Users can choose from various plot types, including bar plots, scatter plots, line plots, and count plots, and specify the x-axis and y-axis features for dynamic visualizations.
2) **App Success Prediction:** Offers an interface to predict the success category of an app. Users can adjust sliders for features like Rating, Reviews, Installs, Price, and Sentiment Polarity, select an app category, and choose between models such as Random Forest or Decision Tree. The predicted success category is displayed dynamically based on the input values.
3) **App Recommendation:** Suggests top apps based on user-defined criteria, such as the selected category, the desired number of recommendations, and the discrepancy threshold. Results are presented in a clear, tabular format, ensuring ease of interpretation.

IV. Results and Discussions

This section presents and analyzes the results of the success-level classification models.

The performance of various models in predicting the success category of apps was evaluated using metrics such as accuracy, F1 score, precision, recall, and confusion matrix. To avoid data leakage, input data excluded the following:

- **App:** As it represents app names and does not contribute to the prediction.
- **Success Category:** Being the target variable.
- **Features like Reviews-Installs Interaction, Last Updated, and Discrepancy:** These were used to define the thresholds for app success and determine the success categories.

The models were assessed on both training and test datasets to ensure generalizability and robustness. The table below provides a summary of the key results and insights.

TABLE III
Model Performance Summary

Model	Train Acc.	Test Acc.	Train F1	Test F1
Logistic Regression	84.80	62.81	83.85	68.04
Decision Tree	100.00	96.73	100.00	96.81
Random Forest	100.00	95.95	100.00	96.06
Feed Forward Neural Network	83.00	80.00	83.00	75.51

Discussion: Random Forest and Decision Tree outperformed the others, showing good generalizability and strong test results. Logistic Regression and Feed Forward Neural Networks faced issues with overfitting and class imbalance.

V. Business Implications and Future Work

A. Business Implications

- **Strategic Decision-Making:** Predictive modeling enables developers to identify areas for improvement, prioritize updates, and enhance user engagement while minimizing rating-review discrepancies.

- **Revenue Growth:** Accurate recommendations and success predictions attract a larger user base, increasing downloads and driving revenue growth.
- **Market Competitiveness:** Sentiment-driven analytics and timely updates empower apps to meet user expectations effectively, providing a competitive advantage in dynamic marketplaces.

For instance, marketplaces like the Google Play Store and Apple App Store could adopt systems to detect mismatched reviews and notify users or exclude such reviews from app ratings. This ensures more accurate evaluations and better user trust. Additionally, developers can focus on high-discrepancy, small-app categories to address unmet needs, capturing niche markets with less competition and significant growth potential.

B. *Future Work*

Future efforts will aim to enhance model performance through advanced techniques like ensemble learning and real-time feedback integration. Expanding the dataset to include additional marketplaces, such as the Apple App Store, will help generalize findings across platforms. Scaling the system to handle larger datasets and real-world applications will make it industry-ready, offering robust tools for developers. Furthermore, the project seeks to explore broader app categories and extend methodologies to other digital products, driving innovation in app evaluation and recommendation systems.

VI. Conclusion

This project demonstrates the effectiveness of integrating predictive modeling and sentiment analysis to address discrepancies in app ratings and user reviews. By introducing a developer-centric success prediction model and a user-centric recommendation system, it provides actionable insights for both app developers and marketplaces. The Decision Tree emerged as the most effective model, achieving high accuracy and F1 scores, while the Streamlit-based application ensures accessibility and usability. Future expansions include scaling the dataset, incorporating real-time feedback, and exploring applications beyond app marketplaces, emphasizing the broader potential of sentiment-driven evaluations in dynamic digital ecosystems.//

References

[1] K. Tabianan, D. Arputharaj, M. N. B. A. Rani, and S. Nagalingham, "Data analysis and rating prediction on google play store using data mining techniques," *JOURNAL OF DATA SCIENCE*, vol. 2022:01, 2022. [Online]. Available: http://eprints.intimal.edu.my/1575/1/jods2022$_0$1*.pdf*

[2] V. N. Inukollu, D. D. Keshamoni, T. Kang, and M. Inukollu, "Factors influencing quality of mobile apps: Role of mobile app development life cycle," *International Journal of Software Engineering Applications (IJSEA)*, vol. 5, no. 5, pp. 15–23, 2014. [Online]. Available: https://airccse.org/journal/ijsea/papers/5514ijsea02.pdf

[3] R. Aralikatte, G. Sridhara, N. Gantayat, and S. Mani, "Fault in your stars: An analysis of android app reviews," *arXiv preprint*, 2017. [Online]. Available: https://arxiv.org/abs/1708.04968

[4] S. Ranjan and S. Mishra, "Comparative sentiment analysis of app reviews," *arXiv preprint*, 2020. [Online]. Available: https://arxiv.org/abs/2006.09739v1

[5] A. Almansour, R. Alotaibi, and H. Alharbi, "Text-rating review discrepancy (trrd): an integrative review and implications for research," *Future Business Journal*, vol. 8, 2022. [Online]. Available: https://fbj.springeropen.com/articles/10.1186/s43093-022-00114-y

[6] M. U. Khan, R. M. U. Khalid, S. A. Burhan, and M. Nauman, "Machine learning based recommendation system for android apps," in *13th International Conference on Machine Learning and Data Mining*. IEEE, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9515087

[7] Lavanya, "Google play store apps dataset," 2020, web scraped data of 10k Play Store apps for analyzing the Android market. [Online]. Available: https://www.kaggle.com/datasets/lava18/google-play-store-apps

[8] A. Mahmood, "Identifying the influence of various factors of apps on google play apps ratings," *Springer Journal of Mobile Computing*, vol. 2, pp. 15–23, 2020. [Online]. Available: https://link.springer.com/article/10.1007/s42488-019-00015-w

[9] V. A, S. R, L. P, and A. R. S, "Google playstore reviews prediction using ml and nlp," *Journal of Science, Computing and Engineering Research (JSCER)*, vol. 6, pp. 1–13, 2023. [Online]. Available: https://doi.org/10.46379/jscer.2023.0604013