Library Progress International *Print version ISSN 0970 1052*
Vol.44 No. 3, Jul-Dec 2024: P.1907-1918 *Online version ISSN 2320 317X*

Original Article *Available online at www.bpasjournals.com*

Predicting Android App Success Before Google Play Store Launch Using Machine Learning

Muhammad Aleem [1], Noorhuzaimi Mohd Noor[2*]

[1,2*]Faculty of Computing, College of Computing and Applied Sciences, Universiti Malaysia Pahang, Pekan 26600, Pahang, Malaysia,
aleemmian380@gmail.com1, nhuzaimi@umpsa.edu.my2

How to cite this article: Muhammad Aleem, Noorhuzaimi Mohd Noor (2024) Predicting Android App Success Before Google Play Store Launch Using Machine Learning. *Library Progress International,* 44(3), 1907-1918

ABSTRACT
The Google Play Store serves as a pivotal platform for Android app developers, drawing millions of users globally. With an overwhelming influx of new apps introduced daily, developers are faced with the critical challenge of sustaining market share amidst fierce competition. Predicting an app's success before its official launch can provide a strategic advantage, potentially transforming how developers approach app releases. In this study, we aim to forecast the success of new Android apps by predicting key indicators such as user ratings and installation numbers prior to their debut on the Google Play Store. Leveraging machine learning techniques, we analyze historical app data and various app features to build predictive models that offer valuable insights into app performance, empowering developers to make data-driven decisions before launch.

Keywords: Mobile Application, Forecasting, Android, App Success, Google Play Store, Launch, Predictive Analytics

INTRODUCTION

Google Play Store is a prime destination for Android app developers, attracting a vast user base. With a large number of new apps being introduced daily, developers face the challenge of maintaining their market share amidst intense competition. The ability to predict an app's performance before its release can provide a significant competitive advantage. Traditionally, an app's success is measured by user ratings and installation numbers. This study aims to forecast the popularity of new apps by predicting user ratings and installation numbers prior to their launch on the Google Play Store. By leveraging advanced machine learning techniques and focusing on internal features, we develop a comprehensive prediction model that offers valuable insights for developers, helping them to optimize their apps for better market performance.

Mobile applications have become integral to daily life due to the widespread use of smartphones. This surge in smartphone usage has led to a significant increase in mobile apps, making the market both extensive and highly competitive. Nearly every digital need has resulted in the creation of multiple apps, offering users a wide range of choices for downloading and usage. In such a competitive environment, numerous factors determine an app's popularity and success. Consequently, developers must carefully consider various aspects during the development and deployment phases. Despite the app store hosting millions of apps, many receive few downloads or go unused. Therefore, understanding how apps become essential to people's lives is crucial. The rapid expansion of the mobile app market has spurred advanced innovation, and as the market continues to grow, so does the number of developers. This growth contributes significantly to the global mobile application sector's revenue. Mobile apps have become the primary utility for smartphones, offering diverse features and services such as social networking, entertainment, shopping, information access, and navigation (Venkata et al., 2014). An app store, such as Google Play, serves as a platform where users can download mobile applications for various services and software (Abdul et al., 2018). Users can browse and download apps from these stores, while developers can monitor their apps through reviews and star ratings (Tuckerman, 2014). Reviews often include user experiences, problem reports, requests for additional features, or numerical ratings of the app. Various app features, both internal and external, significantly influence an app's success. This research utilizes a dataset from the Google Play Store to investigate several external characteristics that may determine an app's success. By developing and analyzing several classification models, the study aims to better understand how to classify an app's success (Karnik, 2019).

Padhye et al., (2019) examines app ratings and highlights that once a store's rating reaches a certain threshold, additional user ratings do not significantly impact the overall rating. Instead, the updated app rating depends on the app's version, leading to the concept of grading different versions. Version ratings can be calculated using the available store rating. This approach aims to help developers improve their apps even after they have reached a notable value. The idea of revealing the app's current version ratings is supported to provide more transparency. Among mobile app distribution platforms, the Google Play Store is becoming one of the most appealing and user-friendly options for mobile apps (Liu, 2012).

Motivation

After reviewing multiple research papers on Android apps, we are motivated to conduct a study on predicting the success of Android apps, addressing significant gaps in current research. Most existing studies focus on post-launch data, which, while useful, does not offer proactive guidance to developers. Our study aims to fill this gap by accurately predicting key success metrics such as the number of installations and user ratings before an app is released on the Google Play Store. This predictive capability is crucial as it provides developers with actionable insights, allowing them to make necessary adjustments to their apps before launch. By optimizing features, design, and functionality based on these predictions, developers can enhance user engagement and satisfaction, thereby improving the app's overall success prospects in a highly competitive market. This proactive approach not only increases the likelihood of a successful launch but also helps in efficient resource allocation and strategic planning, ultimately contributing to the app's long-term viability and profitability.

Literature Review

The authors in paper (Herodotou et al., 2022) developed different systems to detect inconsistencies between reviews and ratings. The results of the study were not surprising - both Android app developers and users agreed that an app's rating should align with its reviews. They also suggested implementing an automated system to identify any inconsistencies between ratings and reviews. This study suggested an effective method to address the review-rating mismatch, which inspired us to develop a feature based on this concept. However, due to time limitations, only 2000 reviews out of 199763 could be manually annotated by three team members, resulting in a wide variance that hindered accurate results, leading us to abandon the idea.

In a similar vein, authors in paper (Strzelecki, 2020) noted that once store-ratings reach a certain point, user ratings have no impact on the overall store rating. They also observed that app ratings vary by version after updates, but this does not influence the store-rating. So, they devised the concept of version rating by utilizing the same calculation method used for store rating in the App Store. This suggestion was made to encourage app developers to continue creating apps, even after the store rating surpasses a certain point, and they suggested that all app store owners should show the rating of the current version. The method they use to calculate version rating in order to incentivize developers to work on updates is highly effective. In another study focusing on this aspect (Badawood & AlBadri, 2021) the author suggests that there is a significant discrepancy between the numerical ratings, such as star ratings, and the actual reviews provided by users. To address this issue, the author has proposed a new rating system to eliminate the confusion caused by inconsistent ratings and reviews from the same user. Users heavily rely on others' opinions when making decisions, as suggested by the author, users decide to install an app based on its rating. The author believes that two sub problems are ambiguity and bias in user ratings. Adding more details, he mentioned that previously individuals would only determine the rating by analysing the feedback rather than relying on the star rating provided. He suggested a system that will first analyse user reviews sentimentally to address the issues. Next, it will produce a numerical rating based on the polarity. Therefore, combining the sentiment analysis rating with the star ratings from users will result in the ultimate rating. This suggestion aims to clarify for users and provide a comprehensive rating from both reviews and star ratings. This paper highlights the strong correlation between user ratings and reviews, leading us to incorporate app reviews into our research. A detailed analysis of the reviews is presented in subsequent sections of this paper.

Similarly, in reference (Dietterich, 2000) Luiz and colleagues introduced a model for mobile app developers that enables them to make adjustments to features that have been deemed negative by end users during their app evaluation. Their Framework was developed to showcase to developers the significance of sentiment rating over star rating in accurately evaluating user feedback for an application, as well as the importance of acknowledging the bias of features impacting an application's overall rating. This paper led us to determine the sentiment average by demonstrating its superior accuracy compared to star ratings. Instead of utilizing their approach, we opted to employ a Python library to calculate the sentiment mean.

Given the significance of polarity values, the study by Ongsulee aided in understanding the significance of eliminating inconsistent reviews to improve sentiment analysis accuracy by reducing dataset noise and enhancing polarity value precision. Fu, Lin, and Li presented WisCom, a system that can examine a substantial quantity of user input in mobile app stores at three tiers, requiring at least ten million comments.

The writer in reference (Soumik et al., 2019) performed a preliminary investigation on collected App Store information, recognizing connections with specific attributes. They aimed to predict the app's success on the Google Play Store by analysing the extracted features and recent user reviews before its launch. Every single document we referenced provided valuable insights that we integrated into our work, contributing to its organization and facilitating a more thorough analysis of the dataset's features.

Federica (2020) from the Google Play Store, focusing on existing characteristics and employing three different models to derive conclusions. Key characteristics, installation figures, and average user feedback were utilized to make more accurate predictions. A linear model was used to forecast the mean rating for all provided data. Principal Component Analysis (PCA) revealed significant relationships among the features, leveraging inputs from the Generalized Linear Model (GLM) and Linear Regression.

In examining app descriptions provided by creators, the author found that around thirty-five percent of successful apps included the term "photo," while approximately thirty-one percent included the term "share." These findings highlight the types of applications that attract the majority of users. The author also recommended using revenue as a measure of success and for predicting future work.

Islam (2014) examines app evaluations, highlighting that once store ratings reach a certain point, additional user ratings no longer significantly impact the overall rating. However, this does not mean that store ratings completely lose their relevance. It also points out that an app's rating is influenced by its version, leading to the concept of evaluating different editions. Version ratings can be determined by assessing the store's existing ratings, helping developers continue improving their apps as they gain popularity. The idea of displaying current version ratings is endorsed and recommended.

Similarly, Luiz et al. (2018) focus on feature-review discrepancies, noting the ineffective combination of reviews and ratings. To address this, the author developed various systems to identify discrepancies between mentioned features. Techniques such as Naïve Bayes Classifier, Decision Tree, Decision Stump, Decision Table, along with other Machine Learning Algorithms and Deep Learning Approaches, were utilized. Multiple surveys were conducted to gather feedback from users and developers about these features. In conclusion, the importance of aligning reviews with app ratings was emphasized and agreed upon by the author, end users, and developers. (Grover, 2015) discusses the distinction between star ratings on app stores and user reviews, based on sentiment analysis. The author proposes a new rating system to eliminate the uncertainty and disparity caused by users between reviews and ratings. According to the author, users are keen on downloading apps based on their ratings. The issue is condensed into two components: uncertainty and biases. The suggested system will analyze the sentiment of user reviews and generate a numerical rating, with the final rating being the average of these numerical ratings. This proposed system is claimed to reduce user confusion, providing a final rating informed by both reviews and star ratings. The paper establishes a close connection between reviews and ratings, aiding users in finding the most suitable app based on their preferences.

Prasad (2018) addresses why users reject apps and explains their lack of success. Not all reviews can be categorized and analyzed for research purposes, as some generate interference and contribute to overall error count. The authors worked on eliminating such reviews, reducing data noise and improving sentiment analysis performance, leading to a more precise polarity value. Li suggested WisCom, The system proposed is capable of analyzing a minimum of ten million user ratings and comments in app markets across three levels. First, it detects inconsistencies in reviews, then investigates why users dislike a specific app, and finally examines how reviews evolve based on user preferences over time. This comprehensive system, along with the highly regarded research, delves into some of the most intriguing issues surrounding app reviews. It develops methods for summarizing and extracting information from reviews, helping users select the top app without needing to read through detailed user comments. The paper also aids users in identifying issues related to app demand or utility and suggests updating features and areas to increase user reach and improve app rankings.

N. Picoto et al. utilized multivariate logistic regression to examine app performance based on five parameters: user ratings, category attractiveness, diversity, capacity factor, and theatrical release. Additionally, they applied fuzzy set qualitative comparative analysis (fsQCA) to uncover new causal explanations for mobile app performance. According to Fernandez (2021), multivariate analysis revealed that the attractiveness of the subcategory, variety, capacity factor, and app release date are all factors that increase the likelihood of an app being ranked among the top 50.

Lukas et al. (2019) worked with an Android app dataset, employing exploratory data analysis and multiple machine learning models to identify the most impactful aspects of an app's success. They used Decision Tree, Random Forest, SVM, XGBoost, and KNN models, achieving accuracy rates of 70.49 percent, 80.34 percent, 75.59 percent, 79.99 percent, and 77.26 percent, respectively. Their analysis determined that an app's rating and content rating are significant factors in its performance in the competitive online marketplace.

Sarro et al. (2018) analyzed a dataset from the Play Store to predict an app's "success," defined by its rating and number of installations. They also considered actual user reviews, using a list of commonly used terms from these reviews for sentiment analysis. The sentiment score from this analysis was included as a feature in their predictions. Despite some assumptions in feature selection, they achieved an accuracy of 85.09% using classifiers such as XGBoost, K-Nearest Neighbor, Random Forest, and SVM. Their handling of the binary classification issue likely contributed to this high accuracy rate.

Kumar et al. (2018) examined a dataset of 100 successful and 100 unsuccessful Android apps from the Google Play Store, each with 34 features. They evaluated the accuracy of various neural network and classification techniques, both with and without Principal Component Analysis (PCA). Without PCA, the NPR algorithm achieved a prediction accuracy of 95.5%, while the Multi-Layer Perceptron (MLP) method achieved a prediction accuracy of 99.995%. This suggests that with more features, the NPR algorithm is more accurate.

Suleman et al. (2019) used a dataset of 10,839 Google Play Store app entries to predict app success based on user reception. Despite the dataset's size potentially being insufficient for the problem at hand, they trained various machine learning models to determine which was most effective for predicting app ratings.

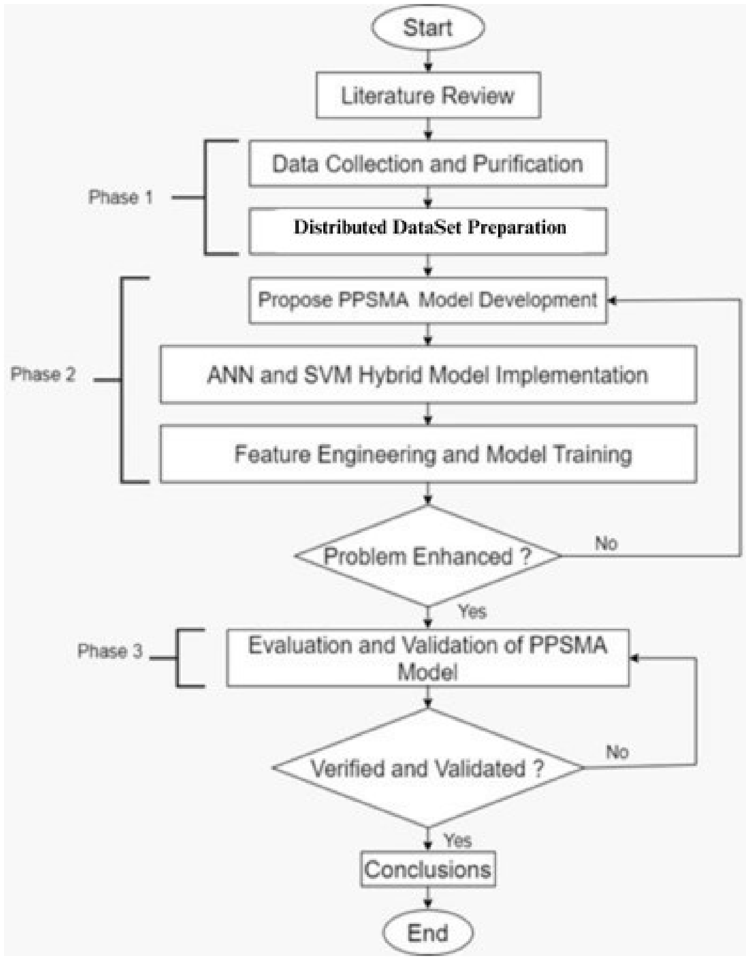
Bashir et al. (2019) evaluated a Google Play dataset to assess app success before launch, using the number of installs and ratings as metrics for success. They encountered binary classification challenges in their predictions and found that the SVM model provided the most accurate results.

Umer et al. (2021) noted the increasing importance of mobile apps as computational capabilities of devices improve. With 7.1 million apps available (Kayalvily et al., 2022), standing out becomes a significant challenge. Acquiring visibility is time-consuming, and understanding app demand and sales is difficult due to limited data. Given that mobile users spend more time on apps than on websites, businesses must grasp the factors influencing app performance to develop effective mobile strategies and marketing.

Aralikatte et al. (2018) addressed the discrepancy between app reviews and ratings by developing algorithms to automatically detect these mismatches. They employed two approaches: the first used classifiers such as Naive Bayes, Decision Tree, Decision Stump, and Decision Table, while the second utilized deep learning techniques. Surveys conducted with end users and developers revealed that both groups agreed that an app's rating should align with its review, and they supported having an automatic system to identify discrepancies between ratings and reviews.

Ranjan et al. (2020) introduced a novel technique for developing an app recommendation system that considers app versions. They highlighted that app ratings can significantly vary based on the application's version. Their approach included a Version-Aware Matrix Factorization (VAMF) framework, which incorporated both version-level and overall app-level review text to generate recommendations. Additionally, researchers employed a greedy method to extract mobile app elements from official pages for business and technical analysis and used a hidden model to classify spam reviews, distinguishing between malicious and non-malicious categories.

System Architecture



Our approach involves collecting data from the Google Play Store dataset, selecting relevant information, and removing irrelevant values. By applying appropriate algorithms to the dataset, we can forecast the user rating and installation numbers for new apps.

Data Preprocessing

Dealing with undesirable values is essential to safeguard the performance of predictive models. Various methods are used to prepare the data.

A. Dataset

We evaluated a dataset from the Google Play Store containing columns such as App Name, Category, Rating, Reviews, Installs, Size, Price, and Content Rating.

App Name	Category	Rating	Reviews	Installs	Size	Price	Latest Version
Door Dash - Food Delivery	FOOD_AND_DRINK	4.548561573	305034	5,000,000+	22M	0	Varies with device
First Mobile	FINANCE	4.31726265	18937	1,000,000+	11M	0	1.9.6.0

Baby Basics	EDUCATION	4.208595288	67	10,000+	50M	0	1.0.6
Harkins Theatres	ENTERTAINMENT	4.218806744	1659	100,000+	12M	0	2.2.5
Pencil Sketch	PHOTOGRAPHY	3.983387871	287582	50,000,000+	23M	0	6.9.1
Sudoku	GAME_PUZZLE	4.457985279	557883	10,000,000+	32M	0	Varies with device
Google Primer	BUSINESS_PRODUCTIVITY	4.357943535	72802	10,000,000+	13M	0	3.801.0
Evernote	PRODUCTIVITY	4.54367824	1503556	100,000,000+	25M	0	Varies with device
Zara	LIFESTYLE	4.293999536	111841	10,000,000+	30M	0	Varies with device
Local deals	TOOLS	5	22	1,000+	2.4M	0	1.3

Table I: Sample Dataset

B. Data Cleaning

The data arrives in different formats with errors that need to be standardized for optimal use in creating a machine learning model.

C. Transforming Data into Suitable Formats

- Dimensions: Convert app dimensions into numerical values. For instance, remove the 'M' from '22M' to get '22' and convert '273k' to its equivalent in megabytes.
- Installation values: Remove commas and the '+' symbol from installation numbers.
- Category and Content Rating: Transform categorical values into numerical values for regression analysis.
- Cost: Convert price details from string format by eliminating the dollar sign.

Analyzing the data reveals that app ratings are crucial in determining an app's performance relative to other apps in the market. It also demonstrates how effectively the company implements feedback from end users, who are crucial to modern software businesses.

Experimental Evaluation

A. Category vs App

Our dataset contains 34 distinct categories. We have displayed the number of different types of apps within these categories. Figure 3 presents a bar chart illustrating the distribution of apps across these categories.

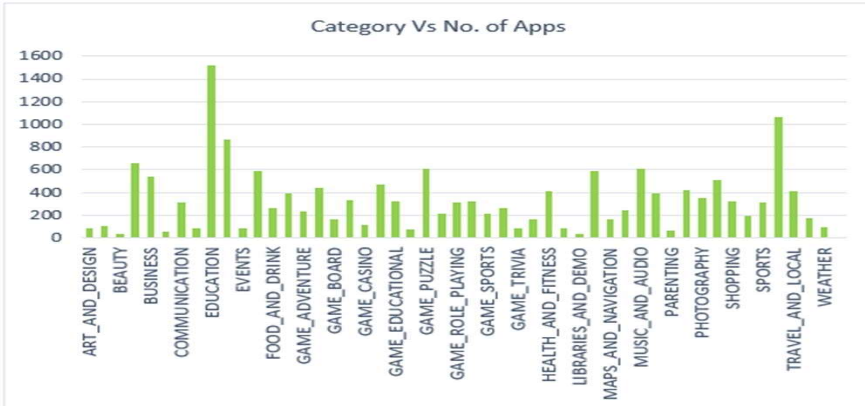


Fig. 2: Bar Chart of all categories against the number of installs

B. Category against the number of Install

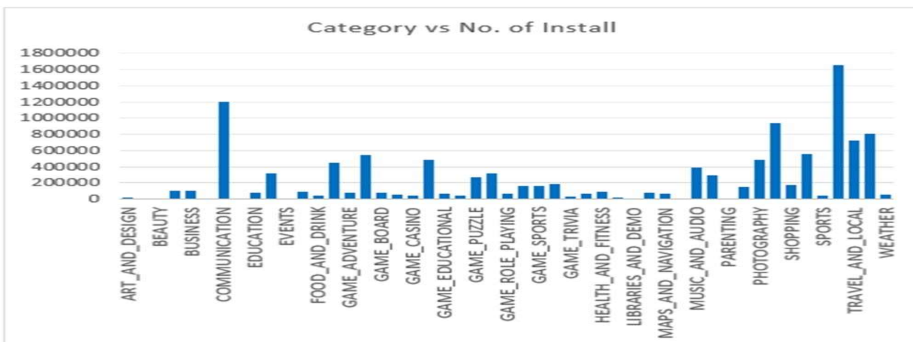


Fig. 3: Bar Chart of all categories against the number of installs

C. Category vs Average Rating

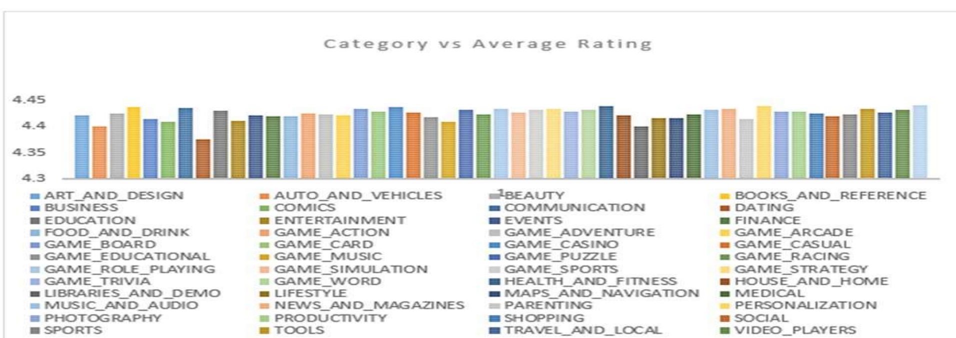


Fig. 4: Bar Chart of all Category vs Average Rating

Place	App	Frequency
1	Game	35,508
2	Education	33,394
3	Tools	21,592
4	Books and Reference	21,377
5	Entertainment	20,604

6	Music and Audio	17,876
7	Personalization	15,044
8	Lifestyle	10,534
9	Finance	10,342
10	Business	10,230

Table 2: Top 10 Most Dominant Categories by Number of Apps

Result Analysis

A. Result Through Used Algorithms

1. Random Forest

Random Forest creates numerous decision trees and merges their outputs to improve predictive power. The Random Forest algorithm's pseudo code involves selecting random subsets of features, identifying the best splitting points, and repeating the process to create multiple trees.

App Name	Original Rating	Predicted Rating
Angry Birds	4.5	4.41
Facebook	4.1	4.22
Messenger	4.36	4.33
UC Browser	4.5	4.31
Clean Master Lite	4.58	4.36

Table 3: Predicted User Rating by Random Forest Algorithm

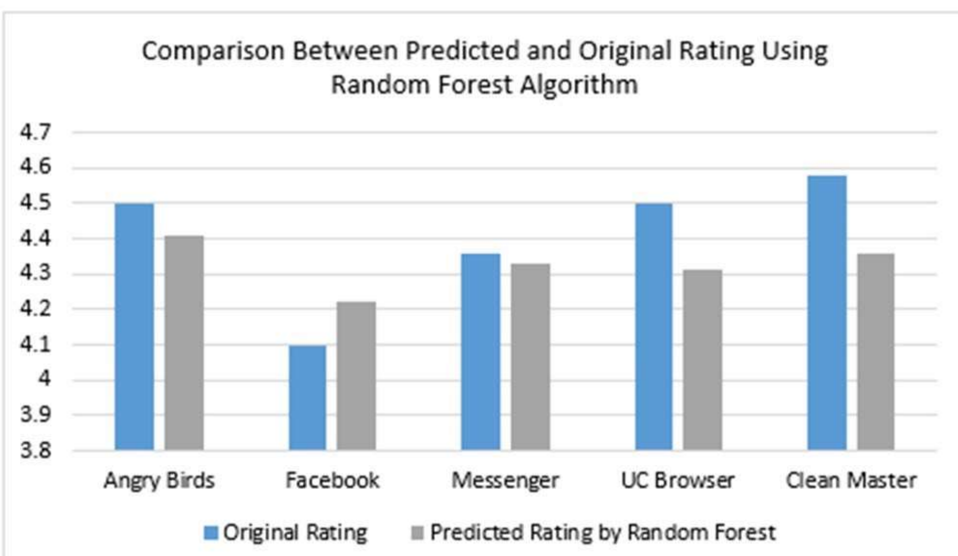


Fig. 5: Bar chart of comparison of original rating and predicted rating using

App Name	Original Rating	Predicted Rating
Angry Birds	4.5	4.61
Facebook	4.1	4.05
Messenger	4.36	4.43
UC Browser	4.5	4.37
Clean Master Lite	4.58	4.68

Table 4: Predicted Installation Number by Random Forest Algorithm

KNN classifies an unknown sample by considering the classifications of its nearest neighbours. The pseudo code for KNN involves calculating distances and identifying the most frequent tag among the nearest neighbors.

Table 5: Predicted User Rating by K-Nearest Neighbor Algorithm

App Name	Original Installation No.	Predicted Installation No.
Angry Birds	10,000,000+	10,129,323
Facebook	1,000,000,000+	455,465,897
Messenger	500,000,000+	359,823,985
UC Browser	500,000,000+	187,489,232
Clean Master Lite	50,000,000+	56,673,210

Table 6: Predicted Installation Number by K-Nearest Neighbor Algorithm

3. Support Vector Machine (SVM):

SVM aims to find the best decision boundary that maximizes the distance from the nearest data points in each class. Kernel SVM is used for non-linearly separable data.

App Name	Original Rating	Predicted Rating
Angry Birds	4.5	4.54
Facebook	4.1	4.09
Messenger	4.36	4.40
UC Browser	4.5	4.44
Clean Master Lite	4.58	4.55

Table 7: Predicted User Rating by Support Vector Machine Algorithm

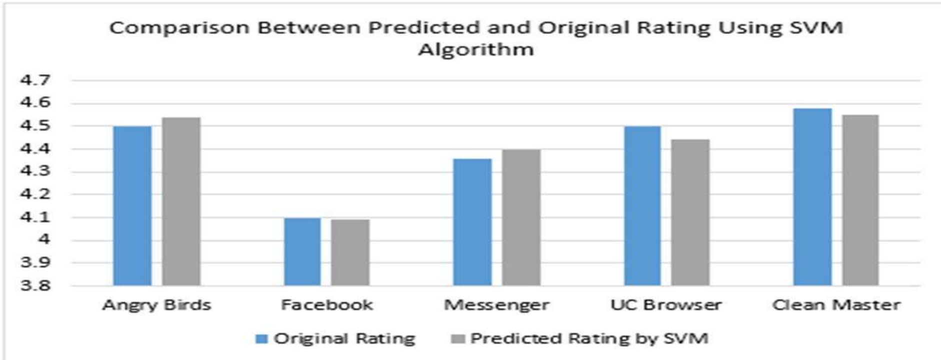


Fig. 6: Bar graph comparing actual rating with predicted rating using SVM Algorithm

App Name	Original Installation No.	Predicted Installation No.
Angry Birds	10,000,000+	10,009,323
Facebook	1,000,000,000+	975,465,897
Messenger	500,000,000+	634,324,990
UC Browser	500,000,000+	298,753,459
Clean Master Lite	50,000,000+	43,467,023

Table 8: Predicted Installation Number by Support Vector Machine Algorithm

B. Forecasting Success Metrics

The accuracy of prediction algorithms is evaluated by comparing predicted values with actual data from a subset of apps. Random Forest and SVM provided the most accurate forecasts for both ratings and installations.

FINDINGS

In this study, we analyzed app categories, installs, and ratings, using various predictive algorithms. Our dataset comprises 34 distinct categories, as illustrated in Figure 3, displaying a bar chart of app distribution. The top ten dominant categories include Games (35,508 apps), Education (33,394 apps), and Tools (21,592 apps), among others. We used three machine learning algorithms to predict user ratings and installation numbers: Random Forest, K-Nearest Neighbour (KNN), and Support Vector Machine (SVM).

Random Forest: This algorithm constructs multiple decision trees and aggregates their outputs for better predictive power. For example, it predicted Angry Birds' rating as 4.41 (original: 4.5) and Facebook's installations as 78,845,910 (original: 1,000,000,000+).

K-Nearest Neighbour (KNN): KNN classifies unknown samples based on their nearest neighbors. It predicted Angry Birds' rating as 4.61 (original: 4.5) and Facebook's installations as 455,465,897 (original: 1,000,000,000+).

Support Vector Machine (SVM): SVM aims to find the optimal decision boundary to maximize class separation. It predicted Angry Birds' rating as 4.54 (original: 4.5) and Facebook's installations as 975,465,897 (original: 1,000,000,000+).

Comparing the algorithms, Random Forest and SVM provided the most accurate predictions for both ratings and installation numbers. These results demonstrate the effectiveness of these algorithms in forecasting app success metrics, aiding developers in refining their apps prior to launch.

CONCLUSIONS

Predicting the success of Android apps prior to their launch can significantly benefit developers in making informed decisions and optimizing their apps. Our study demonstrates the effectiveness of machine learning algorithms like Random Forest, K-Nearest Neighbor, and Support Vector Machine in forecasting app success metrics such as user ratings and installation numbers. By leveraging historical data from the Google Play Store, developers can anticipate app performance and implement necessary improvements before release, thereby enhancing their market position. Future work could involve refining these models with additional features and exploring more advanced machine learning techniques to further improve prediction accuracy.

Acknowledgment
I would like to express my deepest gratitude to my supervisor, Noorhuzaimi Mohd Noor, for her invaluable guidance, support, and encouragement throughout the course of this research. His insights and expertise have been instrumental in the successful completion of this project. I am also profoundly grateful to my family and friends for their unwavering support and encouragement, which have been a constant source of motivation. I would like to thank all my colleagues and peers who have contributed to this research in various ways, offering their time, knowledge, and support. This accomplishment would not have been possible without the collective efforts and contributions of everyone involved.

1. References

[1] Venkata N Inukollu, Divya D Keshamoni, Taeghyun Kang and Manikanta Inukollu. “Factors Influencing Quality of Mobile Apps: Role of Mobile App Development Life Cycle” in International Journal of Software Engineering & Applications (IJSEA), Vol.5, No.5, September 2014.

[2] Abdul Mueez, Khushba Ahmed, Tuba Islam and Waqqas Iqbal, “Exploratory Data Analysis and Success Prediction of Google Play Store App,” B.Sc. Engineering in CSE thesis, BRAC University, Dhaka - 1212, Bangladesh, December 2018.

[3] Tuckerman, C.J., “Predicting Mobile Application Success,” Dec. 2014.

[4] Federica Sarro, Mark Harman, Yue Jia, and Yuanyuan Zhang, “Customer Rating Reactions Can Be Predicted Purely Using App Feature”, CREST, Department of Computer Science, University College London, London, UK.2020

[5] (2002) The IEEE website. [Online]. Available: http://www.ieee.org/

[6] M. Shell. (2002) IEEEtran homepage on CTAN. [Online]. Available: http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran/

[7] FLEXChip Signal Processor (MC68175/D), Motorola, 1996.

[8] “PDCA12-70 data sheet,” Opto Speed SA, Mezzovico, Switzerland.

[9] A. Karnik, “Performance of TCP congestion control with rate feedback: TCP/ABR and rate adaptive TCP/IP,” M. Eng. thesis, Indian Institute of Science, Bangalore, India, Jan. 2019.

[10] J. Padhye, V. Firoiu, and D. Towsley, “A stochastic model of TCP Reno congestion avoidance and control,” Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99-02, 2019.

[11] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, IEEE Std. 802.11, 1997.

[12] Liu, B. (2012). Sentiment analysis and opinion mining. Synthesis lectures on human language technologies.

[13] Luiz, W., Viegas, F., Alencar, R., Mourão, F., Salles, T., Carvalho, D., Gonçalves, M. A., and Rocha, L. (2018). A feature-oriented sentiment rating for mobile app reviews. In Proceedings of the 2018 World Wide Web Conference on World Wide Web, pages 1909–1918. International World Wide Web Conferences Steering Committee.

[14] Islam, M. R. (2014). Numeric rating of apps on google play store by sentiment analysis on user reviews. In 2014 International Conference on Electrical Engineering and Information Communication Technology.

[15] Grover, S. (2015). 3 apps that failed (and what they teach us about app marketing). [online] https://blog.placeit.net/apps-fail-teach-us-app-marketing/.

[16] Fu, B., Lin, J., Li, L., Faloutsos, C., Hong, J., and Sadeh, N. (2013). Why people hate your app: Making sense of user feedback in a mobile app store. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 1276–1284. ACM.

[17] Prasad Patil (March 23, 2018). What is Exploratory Data Analysis? [Online]. In blog, Available: https://towardsdatascience.com/exploratory-data-analysis8fc1cb20fd15.

[18] Lukas Frei (Apr 26, 2019). Speed Up Your Exploratory Data Analysis With Pandasa-Profiling [Online]. In blog, Available: https://towardsdatascience.com/speed-up-your-exploratory-data-analysis-with-pandas-profiling-88b33dc53625. [19]

[19] Dhinakaran, V. T., Pulle, R., Ajmeri, N., & Murukannaiah, P. K. (2018, August). App review analysis via active learning: reducing supervision effort without compromising classification accuracy. In 2018 IEEE 26th international requirements engineering conference (RE) (pp. 170-181). IEEE. https://doi.org/10.1109/RE.2018.00026
[20] Sarro, F., Harman, M., Jia, Y., & Zhang, Y. (2018, August). Customer rating reactions can be predicted purely using app features. In 2018 IEEE 26th International Requirements Engineering Conference (RE) (pp. 76-87). IEEE. https://doi.org/10.1109/RE.2018.00018
[21] Fernandez, V., & Gallardo-Gallardo, E. (2021). Tackling the HR digitalization challenge: key factors and barriers to HR analytics adoption. Competitiveness Review, 31(1), 162–187. https://doi.org/10.1108/CR-12-2019-0163
[22] Kumar, V., & Garg, M. L. (2018). Predictive analytics: a review of trends and techniques. International Journal of Computer Applications, 182(1), 31-37. https://doi.org/10.5120/ ijca2018917434
[23] Kuhn, M., & Johnson, K. (2019). Feature engineering and selection: A practical approach for predictive models. CRC Press. https://doi.org/10.1201/9781315108230
[24] Schmitt, J., Bönig, J., Borggräfe, T., Beitinger, G., & Deuse, J. (2020). Predictive modelbased quality inspection using Machine Learning and Edge Cloud Computing. Advanced engineering informatics, 45, 101101. https://doi.org/10.1016/j.aei.2020.101101
[25] Mahmud, O., Niloy, N. T., Rahman, M. A., & Siddik, M. S. (2019, June). Predicting an effective android application release based on user reviews and ratings. In 2019 7th International Conference on Smart Computing & Communications (ICSCC) (pp. 1-5). IEEE. https://doi.org/10.1109/ICSCC.2019.8843677
[26] Daeli, N. O. F., & Adiwijaya, A. (2020). Sentiment analysis on movie reviews using Information gain and K-nearest neighbor. Journal of Data Science and Its Applications, 3(1), 1-7.
[27] Yue, L., Chen, W., Li, X., Zuo, W., & Yin, M. (2019). A survey of sentiment analysis in social media. Knowledge and Information Systems, 60(2), 617-663. https://doi.org/ 10.1007/s10115-018-1236-4
[28] Bandana, R. (2018, May). Sentiment analysis of movie reviews using heterogeneous features. In 2018 2nd International Conference on Electronics, Materials Engineering & NanoTechnology (IEMENTech) (pp. 1-4). IEEE. https://doi.org/10.1109/IEMENTECH.2018. 8465346
[29] Dragoni, M., Poria, S., & Cambria, E. (2018). OntoSenticNet: A commonsense ontology for sentiment analysis. IEEE Intelligent Systems, 33(3), 77-85. https://doi.org/10.1109/ MIS.2018.033001419
[30] Magar, B. T., Mali, S., & Abdelfattah, E. (2021, January). App Success Classification Using Machine Learning Models. In 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC) (pp. 0642-0647). IEEE. https://doi.org/10.1109/ CCWC51732.2021.9376021
[31] Bashir, G. M. M., Hossen, M. S., Karmoker, D., & Kamal, M. J. (2019, December). Android apps success prediction before uploading on google play store. In 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI) (pp. 1-6). IEEE.
[32] Suleman, M., Malik, A., & Hussain, S. S. (2019). Google play store app ranking prediction using machine learning algorithm. Urdu News Headline, Text Classification by Using Different Machine Learning Algorithms, 57.
[33] Daimi, K., & Hazzazi, N. Using Apple Store Dataset to Predict User Rating of Mobile Applications.
[34] Umer, M., Ashraf, I., Mehmood, A., Ullah, S., & Choi, G. S. (2021). Predicting numeric ratings for google apps using text features and ensemble learning. ETRI Journal, 43(1), 95 108. https://doi.org/10.4218/etrij.2019-0443
[35] Kayalvily, T., Denis, A., Mohd Norshahriel, A. R., & Sarasvathi, N. (2022). Data Analysis and Rating Prediction on Google Play Store Using Data-Mining Techniques. Journal of Data Science, 2022(01).
[36] Sadiq, S., Umer, M., Ullah, S., Mirjalili, S., Rupapara, V., & Nappi, M. (2021). Discrepancy detection between actual user reviews and numeric ratings of Google App store using deep learning. Expert Systems with Applications, 181, 115111. https://doi.org/10.1016/ j.eswa.2021.115111
[37] Sandag, G. A., & Gara, F. (2020, October). Android Application Market Prediction Based on User Ratings Using KNN. In 2020 2nd International Conference on Cybernetics and Intelligent System (ICORIS) (pp. 1-5). IEEE.

[38] Lengkong, O., & Maringka, R. (2020, October). Apps Rating Classification on Play Store Using Gradient Boost Algorithm. In 2020 2nd International Conference on Cybernetics and Intelligent System (ICORIS) (pp. 1-5). IEEE. https://doi.org/10.1109/ICORIS50180. 2020.9320756
[39] Mahmood, A. (2020). Identifying the influence of various factor of apps on google play apps ratings. Journal of Data, Information and Management, 2(1), 15-23. https://doi.org/ 10.1007/s42488-019-00015-w
[40] Martens, D., & Maalej, W. (2019). Towards understanding and detecting fake reviews in app stores. Empirical Software Engineering, 24(6), 3316-3355. https://doi.org/10.1007/ s10664-019-09706-9
[41] Jha, N., & Mahmoud, A. (2019). Mining non-functional requirements from app store reviews. Empirical Software Engineering, 24(6), 3659-3695. https://doi.org/10.1007/s10664019-09716-7
[42] Aralikatte, R., Sridhara, G., Gantayat, N., & Mani, S. (2018, January). Fault in your stars: an analysis of android app reviews. In Proceedings of the acm india joint international conference on data science and management of data (pp. 57-66). https://doi.org/10.1145/ 3152494.3152500
[43] Luiz, W., Viegas, F., Alencar, R., Mourão, F., Salles, T., Carvalho, D., ... & Rocha, L. (2018, April). A feature-oriented sentiment rating for mobile app reviews. In Proceedings of the 2018 World Wide Web Conference (pp. 1909-1918). https://doi.org/10.1145/3178876. 3186168
[44] Ranjan, S., & Mishra, S. (2020, July). Comparative sentiment analysis of app reviews. In 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-7). IEEE https://doi.org/10.1109/ICCCNT49239.2020.9225348
[45] Rahman, M., Rahman, S. S. M.M., Allayear, S. M., Patwary, M., Karim, F., Munna, M., & Ahmed, T. (2020). A sentiment analysis based approach for understanding the user satisfaction on android application. In Data Engineering and Communication Technology (pp. 397-407). Springer, Singapore. https://doi.org/10.1007/978-981-15-1097-7_33
[46] Pratama, B. T., Utami, E., & Sunyoto, A. (2019, March). A comparison of the use of several different resources on lexicon based Indonesian sentiment analysis on app review dataset. In 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT) (pp. 282-287). IEEE. https://doi.org/10.1109/ICAIIT.2019.8834531
[47] Soumik, M. M. J., Farhavi, S. S. M., Eva, F., Sinha, T., & Alam, M. S. (2019, December). Employing machine learning techniques on sentiment analysis of google play store bangla reviews. In 2019 22nd International Conference on Computer and Information Technology (ICCIT) (pp. 1-5). IEEE.
[48] Suresh, K. P., & Urolagin, S. (2020, January). Android App Success Prediction based on Reviews. In 2020 International Conference on Computation, Automation and Knowledge Management (ICCAKM) (pp. 358-362). IEEE. https://doi.org/10.1109/ICCAKM46823. 2020.9051529
[49] Ongsulee, P. (2017, November). Artificial intelligence, machine learning and deep learning. In 2017 15th International Conference on ICT and Knowledge Engineering (ICT&KE) (pp. 1-6). IEEE. https://doi.org/10.1109/ICTKE.2017.8259629 [32] Settles, B. (2009). Active learning literature survey.
[50] Dietterich, T. G. (2000, June). Ensemble methods in machine learning. In International workshop on multiple classifier systems (pp. 1-15). Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-45014-9_1
[51] Badawood, A., & AlBadri, H. (2021). Technology Based Model of a Mobile Knowledge as a Service to Facilitate Education Community. International Journal of Interactive Mobile Technologies, 15(24). https://doi.org/10.3991/ijim.v15i24.27335
[52] Strzelecki, A. (2020). Application of Developers' and Users' Dependent Factors in App Store Optimization. International Journal of Interactive Mobile Technologies (iJIM), 14(13), pp. 91–106. https://doi.org/10.3991/ijim.v14i13.14143