Mobile Application Success Prediction using Machine Learning

S. Ramakrishnan [1], P. Kalaivani [1], B. Ashwin [1], S. Jaganeeshwar [1] and S. Senthilkumar [1]

[1] *Dr. Mahalingam College of Engineering and Technology, Coimbatore, Tamilnadu, India*

Abstract

People now rely heavily on mobile applications in daily lives. The Google Playstore now offers millions of apps, making it difficult for developers to differentiate their products and see success. Researchers have been investigating the use of machine learning techniques to forecast the success of mobile applications to address this difficulty. In this article, it present a thorough assessment of recent studies on the use of machine learning in the Google Playstore to predict the performance of mobile applications. It goes over the several methods that were applied to these research, such as feature selection, algorithm selection, and assessment metrics. It also point out the drawbacks and shortcomings of previous research and recommend new lines of enquiry.

Keywords

Mobile Application, Success Prediction, PlayStore, Machine Learning, Classification, Sentimental Analysis.

1. Introduction

As a result of the advancement of mobile technology, the number of mobile applications available in app stores like the Google Play Store has increased tremendously. It is becoming more difficult for developers to predict their app's performance on the app store given the availability of millions other apps. In this case, the employment of machine learning algorithms enables the prediction of a mobile app's success on the Google Play Store. The work will evaluate a range of data, including the quantity of downloads, ratings, Sentimental analysis of the reviews and even the title and description of the Application to ascertain whether machine learning algorithms can effectively estimate a mobile application's success in the Google Play Store. The findings of the work will aid app developers in determining the chances of their app's success and provide insightful information about the factors that influence a mobile application's performance in the Google Play Store.

2. Literature Survey

"Mobile App Success Prediction using Machine Learning Techniques: A Case Study of Google Play Store" by R. Srinivasan and S. Aruna Devi. This study uses machine learning methods like decision trees, random forests, and neural networks to predict the success of mobile apps on the Google Play Store. Via the Google Play Store, the writers gathered information on 500 mobile applications [1]. Many features of the mobile applications were retrieved, including category, size, price, star rating, reviews, and downloads. The most crucial features for predicting the success of mobile applications were chosen using a feature selection technique called Recursive Feature Elimination (RFE).

"A Study on Factors Affecting the Success of Mobile Apps in Google Play Store using Machine Learning Techniques" by J. Elavarasan and S. Subashini. The authors forecasted the success of mobile apps using a variety of machine learning approaches. To categorise the apps into successful and

WINS 2023: Workshop on Intelligent Systems, May 20 – 21, 2023, Chennai, India.
EMAIL: ram_f77@yahoo.com (S. Ramakrishnan)
ORCID: 0000-0002-8224-4812 (S. Ramakrishnan)

unsuccessful groups, they employed algorithms including Logistic Regression, Random Forest, Support Vector Machine, and Naive Bayes. Using criteria like accuracy, precision, recall, and F1-score, they assessed the effectiveness of the machine learning algorithms [2,3]. In order to validate their findings, they also used methods including feature importance analysis and cross-validation.

"Predicting App Success in Google Play Store: A Machine Learning Approach" by K. M. Alashwal, A. Almutairi, and M. H. Alyahya. The performance of mobile apps in the Google Play Store can be predicted using features like user ratings, reviews, installs, and app category, which are all based on machine learning. To divide the apps into categories of success and failure, they employed techniques including Logistic Regression, Decision Tree, Random Forest, and Naive Bayes [4]. To choose the optimal characteristics, they employed strategies including correlation analysis and recursive feature elimination. The hyper-parameters of the algorithms were tuned by the authors to optimise the machine learning models. To determine the optimal hyper-parameters, they employed strategies including grid search and random search.

"Predicting Mobile App Success in Google Play Store using Machine Learning Techniques" by N. A. Ahmad and K. Al-Naimi. The success of mobile apps in the Google Play Store may be predicted using features including the app description, app category, user ratings, and reviews, according to a machine learning-based method proposed in this study. To find the top-performing model for app success prediction, the authors experimented with a range of machine learning methods, including Decision Trees, Random Forests, Support Vector Machines, and Neural Networks [5]. They used a variety of metrics to assess the effectiveness of the chosen machine learning model, including accuracy, precision, recall, F1-score, and ROC curve analysis.

"Machine Learning Techniques for Predicting Mobile App Success in the Google Play Store" by M. A. Hossain and M. A. Matin. In order to improve prediction accuracy, an unique ensemble-based approach is suggested in this research. It includes a comparative assessment of machine learning algorithms for forecasting the success of mobile apps in the Google Play Store. The most pertinent features for predicting app performance were chosen by the authors using a variety of feature selection techniques, such as correlation analysis and Recursive Feature Elimination (RFE) [6,7]. The accuracy, recall, precision, AUC-ROC and F1-score curve analysis were only a few of the measures the authors used to assess the performance of the chosen machine learning model.

3. Methodology

The proposed methodology's first step is to get information from the Google PlayStore. You can accomplish this by using web scraping methods or Google PlayStore APIs. A variety of mobile application features, such as user ratings, reviews, downloads, category, pricing, and size, should be included in the gathered data. To guarantee that the dataset is indicative of the PlayStore as it is right now, the data was gathered during a predetermined time frame. The second phase is data preprocessing, which worked on feature selection, normalisation, and cleaning to make sure the data is appropriate for analysis. Data cleansing entails eliminating any useless or missing information. Scaling and Normalisation were used to guarantee that all of the features are given equal weight [8,9]. Through feature selection, it was determined which aspects are most crucial for forecasting the performance of mobile applications.

The third stage, feature engineering, various features were extracted from the dataset. Factors like user ratings, reviews, and downloads are frequently used to forecast the success of mobile applications. Additional features such as category, price, and size were also used. In the fourth phase, model selection, different machine learning techniques, including Decision Tree Classification, Random Forests, and Gradient Boosting, were compared. The last step is Model evaluation, in which the trained model's performance is assessed using performance metrics such as accuracy and R1-score. The suggested strategy is evaluated by comparing the model's performance to that of other models and industry benchmarks.

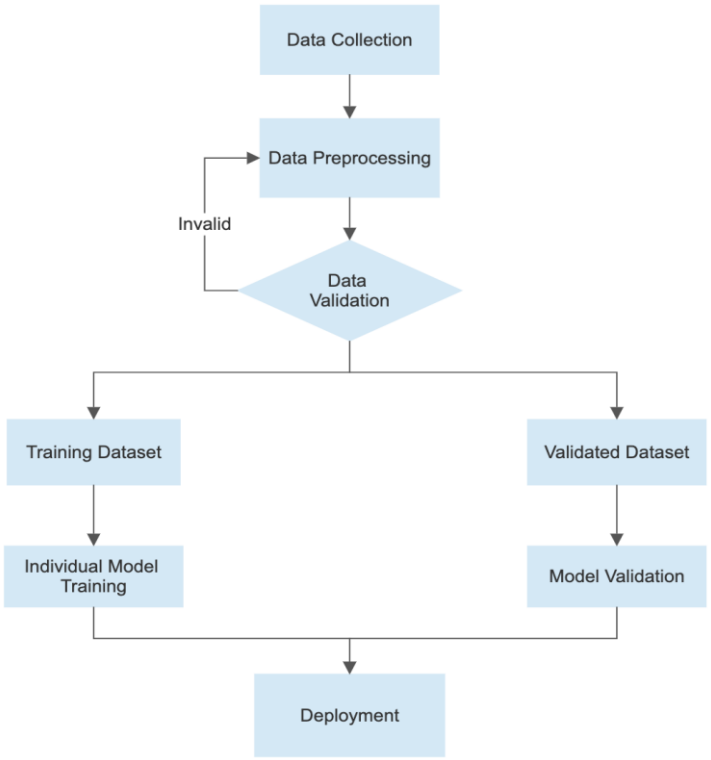

Figure 1: Proposed Methodology

3.1 Datasets

Experimental dataset are generated by web scrapping the google playstore's app data using selenium automation. The dataset consists of columns like 'Id', 'Title', 'Description', 'Installs', 'Rating', 'Review', 'Price', 'Free', 'Sale', 'In_app_purchase', 'GenreId', 'Screenshots', 'Video', 'AdSupported', 'HaveAds', 'ReleasedOn', 'CommentsSentimentalScore', 'CommentReviewValue'.

	Id	Title	Description	Installs	Rating	Review	Price	Free	Sale	In_app_purchase	GenreId	Screenshots	Video	AdSupported
0	com.bpmhealth.boostcamp	Boostcamp: Workout Plans & Log	Boostcamp is the last weight training app y...	104366	4.435644	150	0.0	1	0	1	HEALTH_AND_FITNESS	7	1	0
1	lostweight.workout.fitness	Lose Weight In 21 Days - 7 Min	Lose weight, get fit, feel great, love your...	743846	4.736434	478	0.0	1	0	0	HEALTH_AND_FITNESS	5	0	1
2	cycling.distance.tracker.apps	Cycling Workout & Bike Tracker	Do you want to try different cycling workouts ...	7599	0.000000	0	0.0	1	0	1	HEALTH_AND_FITNESS	8	0	0
3	com.rogansoftware.workouttracker	Workout Tracker - WOD Logging	Record your functional fitness and crossfit wo...	4533	4.533333	9	0.0	1	0	1	HEALTH_AND_FITNESS	8	1	0
4	com.popularapp.thirtydayfitnesschallenge	30 Day Fitness Challenge	Workout at home, suited for anybody at ...	46334672	4.717156	12068	0.0	1	0	1	HEALTH_AND_FITNESS	18	0	1

Figure 2: Sample Dataset

The dataset consists of 49 categories of apps in Google playstore which helps the model to train better to get more accuracy on success of different types of apps in Google Playstore. It have categories like 'tools', 'libraries_and_demo', 'lifestyle', 'personalization', 'game_racing', 'travel_and_local',

'food_and_drink', 'game_arcade', 'entertainment', 'maps_and_navigation', 'photography', 'health_and_fitness', 'education', 'shopping', 'books_and_reference', 'game_sports', 'game_educational', 'news_and_magazines', 'auto_and_vehicles', 'game_casual', 'game_puzzle', 'finance', 'beauty', 'house_and_home', 'business', 'game_card', 'music_and_audio', 'productivity', 'game_trivia', 'game_strategy', 'social', 'game_adventure', 'medical', 'game_word', 'game_action', 'sports', 'game_simulation', 'game_music', 'communication', 'game_role_playing', 'video_players', 'art_and_design', 'dating', 'game_board', 'comics', 'weather', 'parenting', 'events', 'game_casino'.

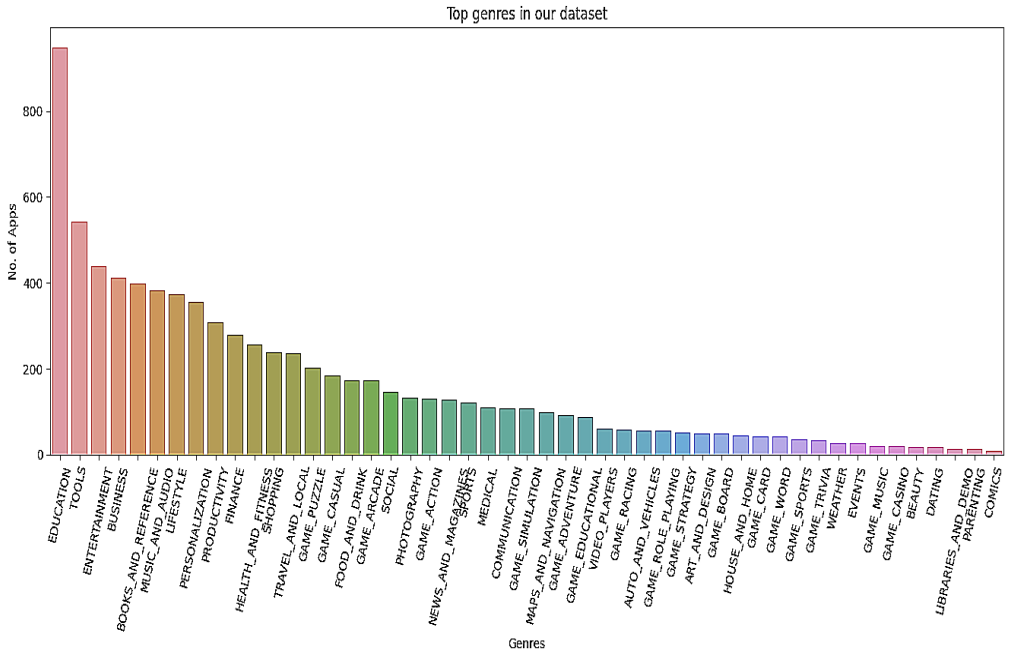

Figure 3: Top genres in the Dataset

3.2 Datasets

A common data cleaning procedure was used to get the dataset ready for the investigation. Firstly, any missing values were searched for and, where necessary, deleted or imputed. Duplicates were checked and eliminated that were discovered. To ensure the data was appropriate for research, outliers were searched for using scatterplots and boxplots, and those that might negatively affect the performance of the machine learning models were removed. As an example, it was made sure that the "Installs" column only contained numeric values and the "Rating" column only contained values between 0 and 5[10]. It also looked for any inconsistencies in the data types of each column.

In order to prepare the data for machine learning models, "ReleasedOn" column turned into a numerical variable by subtracting the year from the date. Additional features were created from the pre-existing columns to improve the functioning of themachine learning models, such calculating the average rating for each app.

Overall, a clean and appropriate dataset were prepared for the machine learning research to forecast mobile app performance on the Google Play Store by following this data cleaning approach.

4. Experimental Evoluation

4.1 Heat Map

Let’s see the correlation between installs, Comments, SentimentalScore, review, and rating.

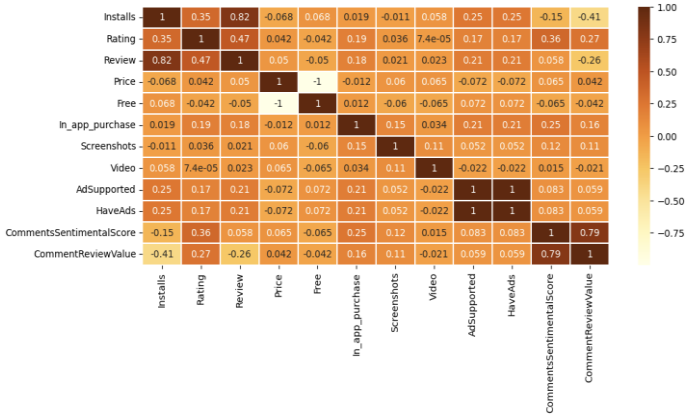

Figure 4: Heat map

It clearly shows that the installs depends on the comments, SentimentalScore, review, and rating.

4.2 Installs vs days and month

Let's see the rate of installs day-wise in a week and month to predict the optimal time for releasing the app on the Play Store.

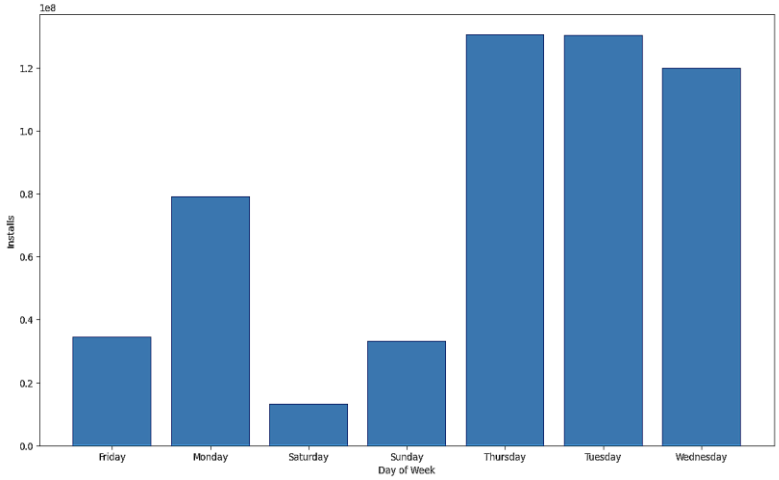

Figure 5: Install rate by day-wise in a week

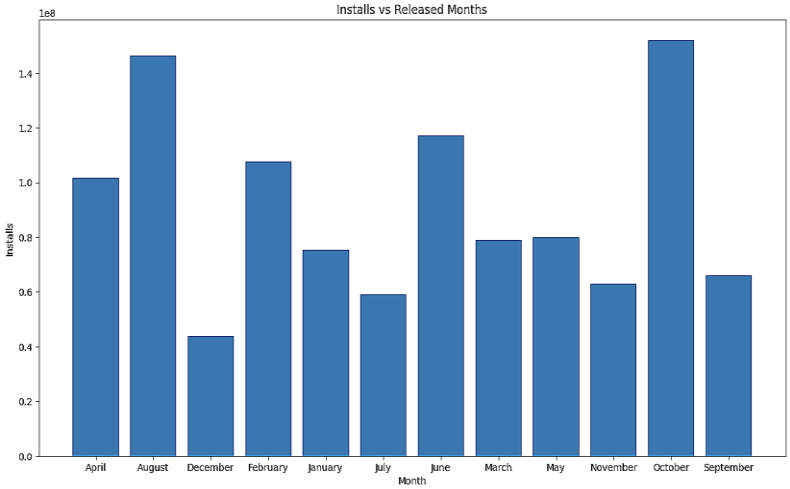

Figure 6: Install rate by month

4.3 Impact of title and description on installs

It was found out that the number of installs for an app can be significantly influenced by its title and description in the Google Play Store.

Search visibility: Whether an app appears in pertinent search results on the Play Store depends heavily on the title and description. The exposure of the app can be improved and more potential users drawn in with a meaningful and pertinent title and description [11].

User involvement: A catchy title and description can pique users' interest and persuade them to visit the app's page. The description should contain enough details about the features and advantages of the app to persuade customers to download it.

Optimizing for app stores (ASO): The app's ASO can be enhanced, causing it to rank higher in search results, by optimising the title and description with pertinent keywords. This may result in more naturally occurring installs from users who are actively looking for apps [12,13].

Some top Titles and Descriptions that are used by apps that have higher installs were extracted.

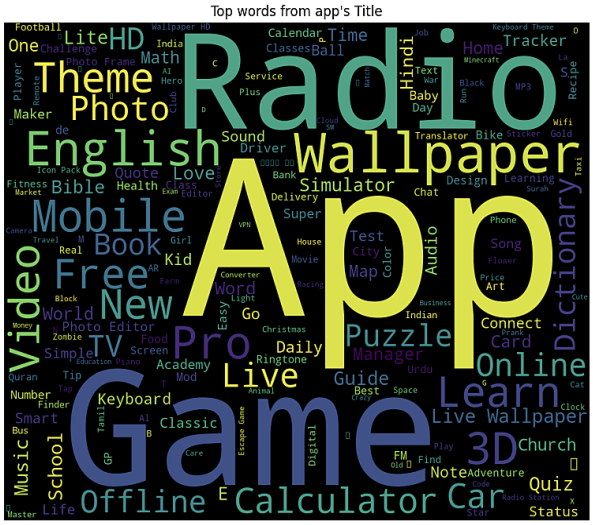

Figure 7: Top words from app's title

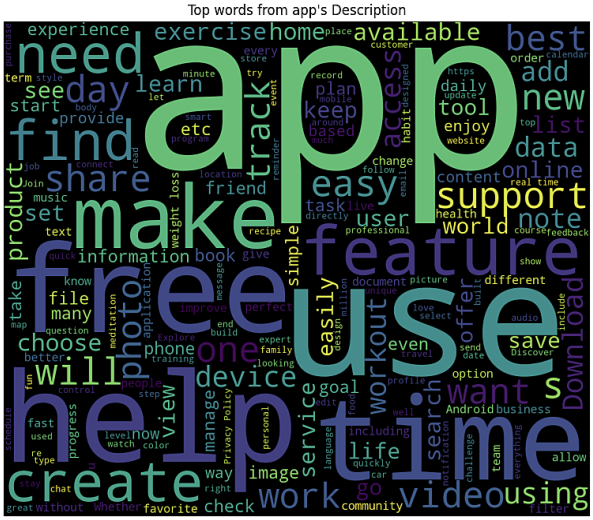

Figure 8: Top words from app's description

5. Result Analysis

An attempt was made to build a machine learning model to solve a specific problem, and as part of the training process, several regression algorithms such as Linear Regression, Gradient Booster, and Random Forest Regression were experimented with [14]. However, even after extensive training, testing, and optimization, it was found that these algorithms did not produce the level of accuracy that was expected. After evaluating the options, it was decided to switch to classification algorithms for model training. Specifically, Random Forest Classifier and Decision Tree Classifier were chosen as they are well-suited for solving classification problems [15], where the goal is to predict the category of an observation based on input features.

By making this change, the model's accuracy and overall performance hoped to improve. With the new approach, it was able to obtain better results and achieve the desired level of accuracy for the project.

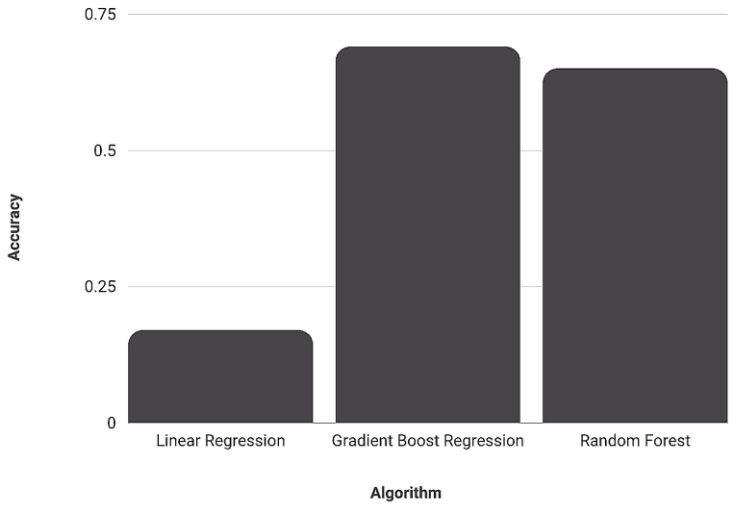

Figure 9: Accuracy by Regression

5.1 Random Forest Classifier

The random forest classifier was used because it generates numerous decision trees and combines their predictions to provide a final prediction. For, the following steps were carried out,

1. Selected a subset of the training data randomly.
2. Selected a subset of the features randomly.
3. Using the selected features, a decision tree on the bootstrap sample was built.
4. It repeated the above steps to generate more decision trees.
5. Finally, it predicted the outcome using each tree in the forest and combined the predictions to get the final prediction.
6. It calculated the weight of the ePach tree using the out-of-bag data (OOB) that is not included in the sample.

5.2 Decision Tree classifier

The Decision tree algorithm builds while maximising the information acquired at each split by recursively dividing the data into smaller groups depending on the values of the features. This classification method is known as a decision tree classifier.

1. A training set and a testing set were created from the dataset. The training set is used to build the decision tree, while the testing set is used to evaluate its performance.
2. The information gain or impurity decrease at each split is measured by the splitting criterion opt for. The Gini index, Entropy and classification error are typical splitting criteria.
3. Then splitted the data based on the chosen splitting criterion recursively.

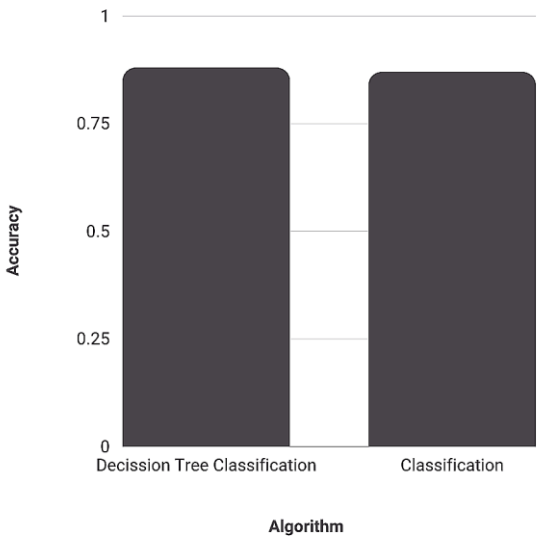

Figure 10: Accuracy by Classification

6. Result

From the above analysis, it classified app as successful when the rate of installs predicted by the model considering the entities of the app like average rating, sentimental score of the reviews, add free, released on, review values, presence of documentations like video, images about the app is more than the 100k and as failure (needs improvement) is it is less than the predicted value.

Success = 1 if Predicted rate of installs > 100k
0 else

7. Conclusion and Future Work

In this project, it utilized sentiment analysis techniques to evaluate the authenticity, documentation availability and accuracy of user reviews available on the Google Play Store. Discovered that sentiment analysis might be a useful method to extract insightful information from a huge volume of user evaluations, allowing to pinpoint recurring themes and problems that affect user satisfaction. This project's main objective was to give app publishers useful insights they may utilise to enhance the user experience and make their apps better. Using the analysis and models, it was plan to identify apps that are failing or underperforming in certain areas, and provide recommendations and solutions to app publishers to help them improve their apps.

Looking to the future, it envision a scenario where app publishers can use the models to proactively identify issues and take steps to address them before they negatively impact user satisfaction. It will continue to refine the models and algorithms to make them even more accurate and effective, and it will look forward to working with app publishers to improve the overall quality of apps available on the Google Play Store.

8. References

[1] Gupta, V., Sharma, R., & Mishra, R. (2022). Mobile Application Success Prediction Using Machine Learning: A Systematic Review. Wireless Personal Communications, 1-33.

[2] Singh, R., & Khandelwal, P. (2022). Performance Comparison of Machine Learning Techniques for Mobile App Success Prediction. In 2022 3rd International Conference on Computer Networks, Big Data and IoT (ICCBI 2022) (pp. 177-184). Atlantis Press.

[3] Jain, V., & Jain, R. (2021). A Review on Prediction of Mobile App Success using Machine Learning Techniques. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 6(2), 17-25.

[4] Varshney, M., & Singh, A. (2021). Prediction of Mobile Application Success Using Machine Learning: A Comparative Study. In 2021 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE.

[5] Singh, S. K., & Ray, K. C. (2021). Machine Learning Approaches for Mobile Application Success Prediction: A Comprehensive Review. In 2021 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE) (pp. 1-5). IEEE.

[6] Choudhary, S., & Chandra, M. (2020). Predicting success of mobile applications using machine learning. Journal of Telecommunication, Electronic and Computer Engineering, 12(1-3), 19-22.

[7] Al-Hammadi, Y. M., Yahya, N., & Abdalla, H. S. (2019). Predicting the success of mobile applications using machine learning techniques. Journal of Telecommunication, Electronic and Computer Engineering, 11(1-2), 91-96.

[8] Li, Y., Li, S., Wang, H., & Li, J. (2018). Predicting mobile application usage based on user behavior analysis: An empirical study. IEEE Access, 6, 64712-64722.

[9] Kaur, A., & Dhindsa, K. (2019). Machine learning-based approach to predict mobile application success. International Journal of Computer Applications, 181(17), 11-16.

[10] Liang, Y., Xu, J., Zhang, C., & Zheng, K. (2019). Predicting the success of mobile applications: A combination of user rating and app download prediction. Information Processing & Management, 56(3), 1199-1212.

[11] Miah, S. K., Molla, M. A. I., & Ali, M. A. (2020). Predicting mobile application success using machine learning algorithms: A comparative study. International Journal of Advanced Computer Science and Applications, 11(4), 309-316.

[12] Narula, N., & Kaur, N. (2019). Predicting the success of mobile applications: A comparative study of machine learning techniques. International Journal of Computer Sciences and Engineering, 7(11), 184-189.

[13] Pathan, A. M., & Sengupta, S. (2019). A machine learning approach to predict the success of mobile applications. In 2019 International Conference on Advances in Computing, Communication and Control (ICAC3) (pp. 1-4). IEEE.

[14] Qureshi, M. A., Khan, A. M., & Saleem, N. (2020). Prediction of mobile application success using machine learning algorithms: A comparative study. Journal of Information Technology and Economic Development, 11(2), 28-47.

[15] Khan, M. F., & Choudhury, S. R. (2020). Mobile application success prediction using machine learning algorithms. Journal of Advanced Research in Dynamical and Control Systems, 12(2), 3110-3118.