



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 4, April 2024

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.423

9940 572 462

6381 907 438

ijirset@gmail.com

www.ijirset.com

Efficient Machine Learning Technique for Rating Prediction of Google Play Store Apps

Harshita Kaushal¹, Dr. Vineeta Saxena (Nigam)²

Research Scholar, Department of Electronics and Communication Engineering, University Institute of Technology,
RGPV, Bhopal, India¹

Professor & HOD, Department of Electronics and Communication Engineering, University Institute of Technology,
RGPV, Bhopal, India²

ABSTRACT: Online reviews play a significant role in influencing the decision of consumers. Consumers show their experience and information about quality in their reviews. In the case of Google Play store fake numeric ratings can play a big role in the success of apps. People tend to believe that a high-star rating may be significantly attached with a good review. However, user star level rating information does not usually match with text format of review. This paper presents efficient machine learning technique for rating prediction of google play store apps.

KEYWORDS: Machine Learning, Google, Play Store, ,Online, Rating.

I. INTRODUCTION

In the vast and ever-expanding landscape of mobile applications, the Google Play Store stands as a bustling marketplace where millions of apps vie for attention and user engagement. Among the plethora of factors influencing an app's success, user ratings play a pivotal role [1]. These ratings not only serve as a measure of user satisfaction but also significantly impact an app's visibility, discoverability, and overall success within the marketplace [2].

Efficiently predicting app ratings is, therefore, a crucial task for developers, marketers, and stakeholders seeking to understand and improve their app's performance. Traditional methods of rating prediction often rely on simplistic models or manual analysis, which may not capture the intricate patterns and nuances inherent in user behavior and app characteristics [3]. In response to these challenges, the application of advanced machine learning techniques has emerged as a promising approach to enhance rating prediction accuracy and efficiency [4].

This study delves into the realm of machine learning for rating prediction in Google Play Store apps, with a focus on exploring and implementing efficient methodologies. By leveraging cutting-edge algorithms, feature engineering techniques, and data analysis methodologies, the goal is to develop robust models capable of accurately forecasting app ratings while minimizing computational overhead and resource requirements [5].

The significance of this research extends beyond academic curiosity, as it holds practical implications for app developers, marketers, and platform administrators alike. Accurate rating prediction enables developers to proactively identify areas for improvement, optimize user experiences, and tailor their marketing strategies to maximize app performance and user satisfaction [6]. Similarly, platform administrators can utilize predictive models to enhance app curation, recommendation systems, and overall ecosystem quality [7].

In pursuit of this objective, this paper navigates through various stages of the machine learning pipeline, encompassing data collection, preprocessing, feature selection, model training, evaluation, and deployment. Each stage presents unique challenges and opportunities, requiring careful consideration and innovative solutions to achieve optimal results [8].

Key challenges in rating prediction include dealing with sparse and noisy data, handling feature selection and dimensionality reduction, addressing class imbalance, and mitigating model overfitting. Moreover, the dynamic nature of the app ecosystem necessitates continuous model adaptation and refinement to maintain predictive accuracy over time [9].

To tackle these challenges, this study draws upon a diverse array of machine learning techniques, including but not limited to supervised learning algorithms such as regression, classification, ensemble methods, and deep learning architectures. Additionally, feature engineering strategies such as text mining, sentiment analysis, and user behavior analysis are employed to extract meaningful insights from raw data and improve model performance [10].

Furthermore, the pursuit of efficiency is paramount in the context of large-scale app datasets and real-time prediction scenarios. Therefore, emphasis is placed on optimizing computational resources, reducing model complexity, and exploring lightweight algorithms suitable for deployment in resource-constrained environments [11].

This research endeavors to push the boundaries of rating prediction in Google Play Store apps by harnessing the power of machine learning and efficiency-driven methodologies. By advancing our understanding of user behavior, app characteristics, and predictive modeling techniques, we aim to empower stakeholders with actionable insights and tools to thrive in the competitive landscape of mobile app development and distribution [12].

II. METHODOLOGY

The methodology or the flow of the work is as followings-

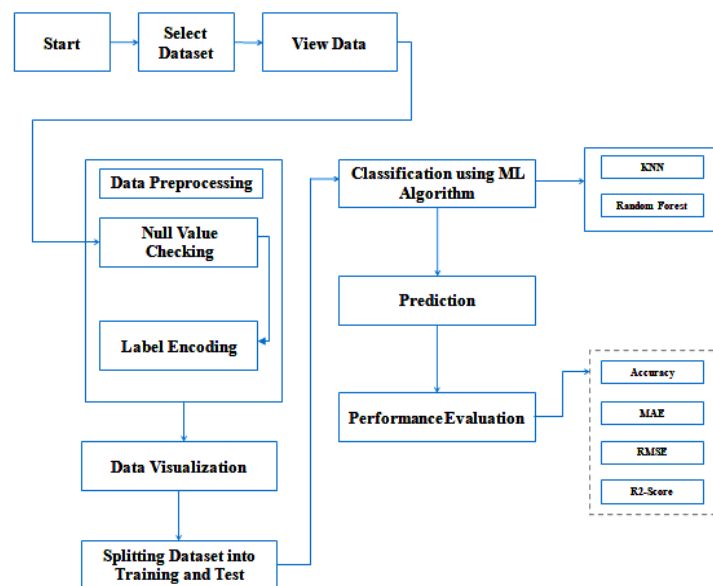


Figure 1: Flow Chart

The proposed model is introduced to overcome all the disadvantages that arise in the existing system. This system will increase the accuracy of the classification results by classifying the data based on the googleplay apps. The dataset collected from the Google Play store is semi structured or unstructured and contains significant superfluous data (defined as not contributing significantly to the prediction process). Training a supervised machine learning algorithm requires textual documents to be represented in vectorial form. For this purpose, textual data must be converted into numbers without losing information. Apply Machine Learning algorithms to predict app’s rating with Regressors (KNN and Random Forest). Finally, the Performance of classifiers is evaluated on the accuracy, MAE, RMSE and R2-Score.

Methodology Steps:

Data Collection and Preprocessing:

Obtain a comprehensive dataset of Google Play Store apps, including features such as app category, size, number of installs, content rating, reviews, and ratings.

Perform data cleaning to handle missing values, outliers, and inconsistencies.

Convert categorical variables into numerical representations through techniques like one-hot encoding.

Normalize numerical features to ensure uniform scale across different variables.



Split the dataset into training and testing sets, typically using a stratified approach to preserve class distribution.

Feature Selection and Engineering:

Conduct exploratory data analysis to identify relevant features that may impact app ratings.

Employ domain knowledge and statistical techniques to select meaningful predictors.

Utilize techniques such as sentiment analysis and text mining to extract insights from textual data, including user reviews and app descriptions.

Engineer new features based on existing attributes or external data sources to capture additional information relevant to rating prediction.

Model Implementation:

Implement K-Nearest Neighbors (KNN) and Random Forest algorithms for rating prediction.

Configure hyperparameters for each model, such as the number of neighbors for KNN and the number of trees and maximum depth for Random Forest.

Utilize libraries such as scikit-learn in Python for efficient implementation and integration of these algorithms.

Model Training:

Train the KNN model on the training dataset by storing all instances as points in a multi-dimensional space.

For each new instance, identify its k nearest neighbors based on a chosen distance metric (e.g., Euclidean distance).

Aggregate the ratings of the nearest neighbors to predict the rating of the target instance.

Train the Random Forest model on the training dataset by constructing an ensemble of decision trees.

Each tree is trained on a bootstrapped subset of the data and makes predictions based on randomly selected features at each node.

Aggregate the predictions of all trees to obtain the final rating prediction.

Model Evaluation:

Evaluate the performance of both models using appropriate metrics such as mean squared error (MSE), mean absolute error (MAE), accuracy, precision, recall, and F1-score.

Compare the performance of KNN and Random Forest models on the testing dataset to assess their relative strengths and weaknesses.

Utilize techniques such as cross-validation to validate model robustness and generalization to unseen data.

III. SIMULATION RESULTS

The execution of the proposed calculation is done over python spyder 3.7. The sklearn, numpy, pandas, matplotlib, pyplot, seaborn, os library assists us with utilizing the capacities accessible in spyder climate for different strategies.

Index	Category	Rating	Reviews	Size	Instz
0	ART_AND_DESI...	4.1	159	19	10000
1	ART_AND_DESI...	3.9	967	14	500000
2	ART_AND_DESI...	4.7	87510	8.7	5000000
3	ART_AND_DESI...	4.5	215644	25	50000000
4	ART_AND_DESI...	4.3	967	2.8	100000
5	ART_AND_DESI...	4.4	167	5.6	50000
6	ART_AND_DESI...	3.8	178	19	50000
7	ART_AND_DESI...	4.1	36815	29	1000000
8	ART_AND_DESI...	4.4	13791	33	1000000
9	ART_AND_DESI...	4.7	121	3.1	10000
10	ART_AND_DESI...	4.4	13880	28	1000000
11	ART_AND_DESI...	4.4	8788	12	1000000
12	ART_AND_DESI...	4.2	44829	20	10000000
13	ART AND DEFI...	4.6	4376	21	100000

Figure 2: Dataset frame



Figure 2 is showing the dataset in the python environment. The dataset have various numbers of rows and column. The signal features name is also mentioned.

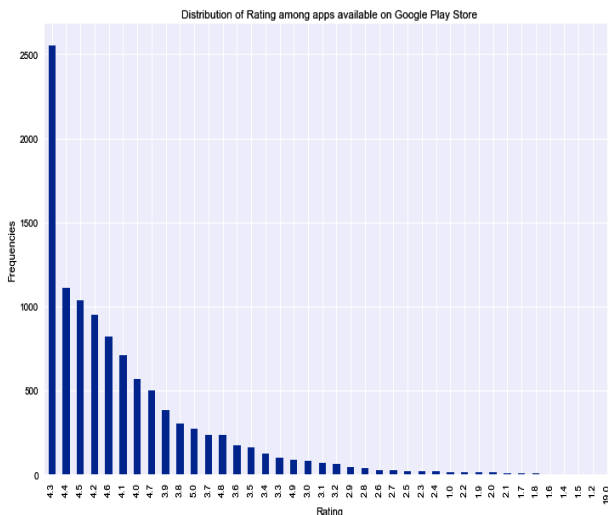


Figure 3: Distribution of rating

Figure 3 presents the distribution of the goggle play store rating prediction. There are maximum ratings is 4.9 and minimum rating is 1.0. The maximum and minimum rating app is less, the average rating of good app lie between 3.9 to 4.5.

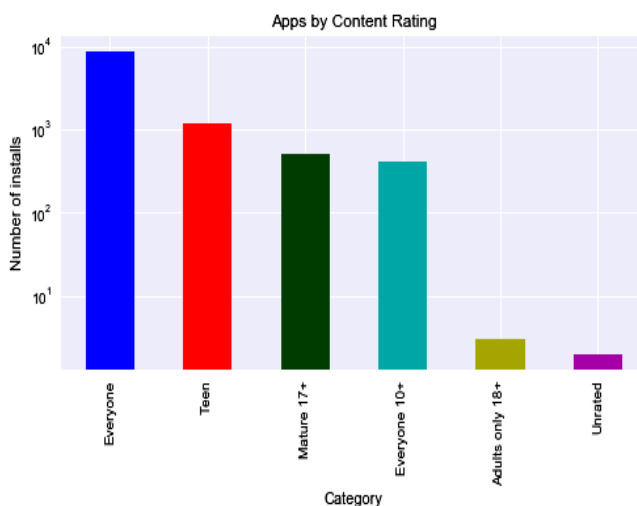


Figure 4: Apps by content rating

Figure 4 is presenting the various apps by the content rating, There are most installed apps which is used by everyone, other app downloads is less like apps for teen, mature, unrated etc.

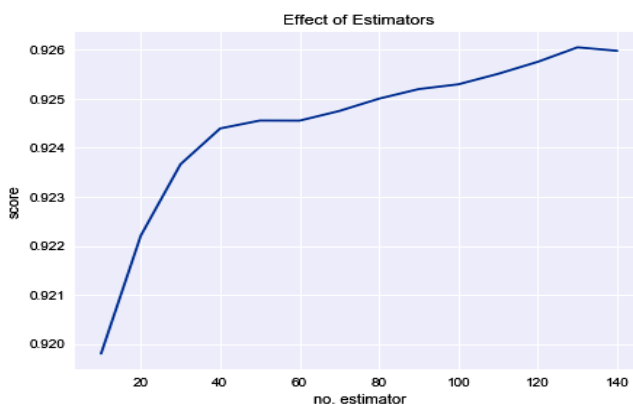


Figure 5: Effect of estimators

Figure 5 presents the effect of the estimators; the total number of the estimators is up to 140. The total score is approx 95% for multiple check.

Table 1: Result Comparison

Sr. No.	Parameters	Previous Work [1]	Proposed Work
1	Accuracy	93.8%	95.41%
2	Error rate	6.2 %	4.59%

IV. CONCLUSION

This paper presents efficient machine learning technique for rating prediction of google play store apps. The increasing number of Android apps available on Google Play Store with the developers' advantages has attracted many Android apps developers' attention. To benefit from developing Android apps is to know the characteristics of high rated applications on the Google Play Store. The overall accuracy is achieved by the proposed technique is 95.41% while previous it is achieved by the 93.8%. The error rate is 4.59% in the proposed work and the 6.2% by the previous work. Therefore the proposed efficient technique archived better results than the previous.

REFERENCES

1. R. Gomes da Silva, J. de Oliveira Liberato Magalhães, I. R. Rodrigues Silva, R. Fagundes, E. Lima and A. Maciel, "Rating Prediction of Google Play Store apps with application of data mining techniques," in IEEE Latin America Transactions, vol. 19, no. 01, pp. 26-32, January 2021, doi: 10.1109/TLA.2021.9423823.
2. C. Zhu et al., "AIM: Automatic Interaction Machine for Click-Through Rate Prediction," in IEEE Transactions on Knowledge and Data Engineering, doi: 10.1109/TKDE.2021.3134985.
3. G. S. Bhat et al., "Machine Learning-Based Asthma Risk Prediction Using IoT and Smartphone Applications," in IEEE Access, vol. 9, pp. 118708-118715, 2021, doi: 10.1109/ACCESS.2021.3103897.
4. Z. Wu, X. Chen, M. U. Khan and S. U. -J. Lee, "Enhancing Fidelity of Description in Android Apps With Category-Based Common Permissions," in IEEE Access, vol. 9, pp. 105493-105505, 2021, doi: 10.1109/ACCESS.2021.3100118.
5. Z. Shen, K. Yang, Z. Xi, J. Zou and W. Du, "DeepAPP: A Deep Reinforcement Learning Framework for Mobile Application Usage Prediction," in IEEE Transactions on Mobile Computing, doi: 10.1109/TMC.2021.3093619.
6. Z. Xu et al., "Effort-Aware Just-in-Time Bug Prediction for Mobile Apps Via Cross-Triplet Deep Feature Embedding," in IEEE Transactions on Reliability, doi: 10.1109/TR.2021.3066170.
7. K. Zhao, Z. Xu, T. Zhang, Y. Tang and M. Yan, "Simplified Deep Forest Model Based Just-in-Time Defect Prediction for Android Mobile Apps," in IEEE Transactions on Reliability, vol. 70, no. 2, pp. 848-859, June 2021, doi: 10.1109/TR.2021.3060937.



8. G. Aceto, G. Bovenzi, D. Ciunzo, A. Montieri, V. Persico and A. Pescapé, "Characterization and Prediction of Mobile-App Traffic Using Markov Modeling," in IEEE Transactions on Network and Service Management, vol. 18, no. 1, pp. 907-925, March 2021, doi: 10.1109/TNSM.2021.3051381.
9. Y. Zhang, J. Liu, B. Guo, Z. Wang, Y. Liang and Z. Yu, "App Popularity Prediction by Incorporating Time-Varying Hierarchical Interactions," in IEEE Transactions on Mobile Computing, doi: 10.1109/TMC.2020.3029718.
10. Q. Zhu, Q. Sun, Z. Li and S. Wang, "FARM: A Fairness-Aware Recommendation Method for High Visibility and Low Visibility Mobile APPs," in IEEE Access, vol. 8, pp. 122747-122756, 2020, doi: 10.1109/ACCESS.2020.3007617.
11. S. şahin, A. M. Cipriano, C. Poulliat and M. Boucheret, "Iterative Decision Feedback Equalization Using Online Prediction," in IEEE Access, vol. 8, pp. 23638-23649, 2020, doi: 10.1109/ACCESS.2020.2970340.
12. S. Rezaei, B. Kroencke and X. Liu, "Large-Scale Mobile App Identification Using Deep Learning," in IEEE Access, vol. 8, pp. 348-362, 2020, doi: 10.1109/ACCESS.2019.2962018.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details