International Journal of Research and Review
DOI: https://doi.org/10.52403/ijrr.20210955
Vol.8; Issue: 9; September 2021
Website: www.ijrrjournal.com
Research Paper *E-ISSN: 2349-9788; P-ISSN: 2454-2237*

Predict App Rank on Google Play Using the Random Forest Method

Abdul Khaliq[1], Eko Hariyanto[2], Supina Batubara[3]

[1,2,3]Faculty of Science and Technology, University of Pembangunan Panca Budi, Medan, Indonesia

Corresponding Author: Abdul Khaliq

ABSTRACT

Application developers and users are the keys to the market impact on application development. In application development, developers need to predict applications in the market accurately, accurate prediction results are very important in showing user ratings that affect the success of an application. Ratings are given by users to judge that the application is good or not. The higher the rating given by the user, it means that the user likes the application and can be a benchmark for other users to download the application. It is undeniable that there are so many apps available on the Google play store, it is impossible for users to select one by one app on the Google play store. Therefore, a rating prediction system is needed to determine the right application based on the rating given by the user to an application. Predictions will be made using the random forest algorithm as the method used to predict application ratings. This study using the Google Play Store dataset. This dataset has 10840 rows and 13 attributes. The results of this study can be seen from the use of the random forest algorithm with an average accuracy of 93.8%.

***Keywords*:** Google Play Store, Rating, Prediction, Random Forest

INTRODUCTION

At this time the development of technology is developing very fast, one of which is in the field of providing information, information technology can be used to complete data, and is commonly used as a basis for making a decision. In the Google Play Store, there is information in the form of descriptions, comments from users, and ratings regarding the applications in it with the aim of knowing the shortcomings or advantages of the applications made.

The significant growth of the mobile application market has a major impact on digital technology with the number of applications available on the Google Play store until March 2021 at around 2.8 million and will continue to grow over time (Appbrain, 2021). Application developers and users are key to the market impact on application development (Hengshu Zhu et al., 2014). In developing applications, developers need to predict applications on the market accurately, because accurate prediction results are very important in determining application development on Google Play (Shen, Lu and Hu, 2017). In 2017, Hartmann-Boyce et al conducted a review of the Google Play Store app to explore what users like and dislike about weight loss and weight tracking apps. Hartmann-Boyce et al's research results show that user ratings affect the success of an application (Hartmann-Boyce et al., 2017). Application ratings also affect the application's popular recommendation system on the Google Play market with criteria using category parameters, number of installs, ratings, reviews (Zhu et al., 2014).

To predict application ratings, there are several methods used, such as in 2017, Chen et al compared the Logistic Model Tree (LMT), Random Forest (RF), and Decision Tree (CART) methods to predict

landslide susceptibility. The results showed that the results of the comparison of the three methods resulted in the random forest model having the best prediction compared to the LMT or CART models with an Area Under Curve (UAC) value of 0.837 and a predictive accuracy value of 0.772. in another study that compared Random Forest with K-Nearest Neighbors on the HAR (human activity recognition) dataset, the results of this comparison obtained the best accuracy results using the Random forest method with a value of 93.13% (Bindu, BhanuJyothi and Suryanarayana, 2017). In another study where comparisons were made between SVM combined with other classifiers such as BayesNet, AdaBoost, Logistics, IBK, J48, Random Forest, JRip, OneR, and SimpleCart, the results of this study found that SVM combined with Random Forest got good results with a score of 97, 50% compared to using the only SVM with a value of 91.81% (Chand et al., 2016).

Based on the description above, this research will predict the rating of the application on Google Play using the Random Forest method so that it is hoped that it can help find the weaknesses of the application in a short time from the user's point of view as an ingredient to improve the product.

LITERATURE REVIEW

Machine Learning

Machine learning is the field of computer science that involves building algorithms that usefully rely on a collection of examples of certain phenomena. These examples can be natural, man-made, or generated by other algorithms. Machine learning can also be defined as the process of 1) collecting data sets and 2) building statistical models based on these data sets to solve practical problems through algorithms. Assume that the statistical model is used in some way to solve real problems. To save keystrokes, I use the terms "learning" and "machine learning" interchangeably (Burkov, 2019).

Supervised Learning

Supervised learning is an approach where there is already trained data, and there are targeted variables so that the purpose of this approach is to group data into existing data (Andreas Chandra, 2017).

Random Forest

Random Forest (RF) is a tree-based ensemble method designed to overcome the shortcomings of classification and regression tree (CART) methods. RF consists of a large number of classification and regression decision tree weaknesses, which grow in parallel to reduce model bias and variance at the same time (Breiman, 2001).

The following is the formula for random forest:

Entropy (Y) = $-\Sigma_i p(c|Y) log^2$! " $p(c|Y)$, (1)

Information :

Y = Case Set

P(c|Y) = The proportion of the value of Y to class c.

Information Gain (Y,a)

$$= Entropy\ (Y) - \sum_{v \varepsilon Values(a)} \frac{|Y_v|}{|Y_a|} Entropy\ (Y_v). \quad (2)$$

Information :

Values(a) = Possible values of the case set a.

Y_v = Subclass of Y with class v corresponding to class a.

Y_a = All values corresponding to a.

Google Play Store

Google Play is a digital distribution service operated and developed by Google. It serves as the official app store for the Android operating system, allowing users to browse and download applications developed with the Android software development kit (SDK) and published through Google. Google Play also functions as a digital media store, offering music, books, movies, and television programs. It previously offered Google hardware for purchase until the introduction of a separate

online hardware retailer, the Google Store, on March 11, 2015, and also offered news and magazine publications prior to the Google News fix on May 15, 2018 (google, 2012a)

Apps are available through Google Play either for free or for a fee. They can be downloaded directly on Android devices via the Play Store mobile app or by deploying the app to devices from the Google Play website. Apps that exploit the hardware capabilities of a device can be targeted at users of devices with specific hardware components, such as motion sensors (for motion-dependent games) or front-facing cameras (for online video calls). Google Play store had more than 82 billion app downloads in 2016 and has reached more than 3.5 million apps published in 2017. This has been the subject of various issues regarding security, where malicious software has been approved and uploaded to the store and downloaded by users, with varying degrees of severity (google, 2012b)

Google Play was launched on March 6, 2012, bringing together Android Market, Google Music, and the Google eBookstore under one brand, marking a change in Google's digital distribution strategy. The services included in Google Play are Google Play Books, Google Play Games, Google Play Movies & TV, and Google Play Music. After re-branding, Google is gradually expanding geographic support for each service (google, 2012b)

MATERIALS & METHODS

Methodology

This research is carried out in stages which will be carried out starting from determining the dataset. The next stage will be the data preprocessing process. The next stage will be the process of predicting the application rating using Random Forest. After all stages of the process are ready, the next stage will be an analysis of the results obtained by comparing them to the RMSE (Root Mean Squared Error) value to determine the accuracy of the imputation results and the prediction accuracy results are calculated by looking at the percentage of accuracy.

Preprocessing Data

Preprocessing data used is to convert attribute values into numeric form to minimize errors. The tools used in preprocessing using the Jupyter python application.

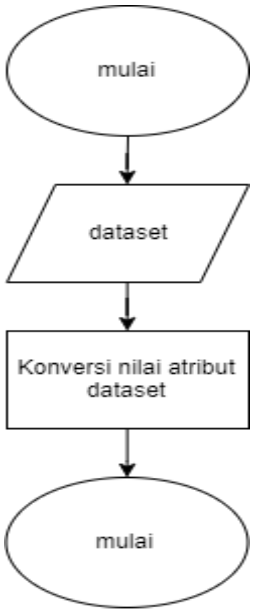

Figure 1. Preprocessing Data

Preprocessing converts attribute values with integers or floats.

a. Convert App attribute values
b. Category. attribute value conversion
c. Remove Symbol on Installs atribut attribute value
d. Conversion of types. attribute values
e. Price. attribute value conversion
f. Last Update attribute value conversion
g. Android Ver attribute value conversion
h. Conversion of Current Ver attribute values
i. Convert the Size. attribute value
j. Conversion of Content Rating attribute values
k. Conversion of Genres atribut attribute values

App Rating Prediction

The random forest method is the development of the CART (Classification and Regression Tree) method by applying

the bootstrap aggregating (bagging) and random feature selection methods by Breiman (2001). Random forest is one of the methods used for classification and regression. This method is an ensemble of learning methods using a decision tree as a base classifier that is built and combined (Kulkarni and Sinha, 2014).

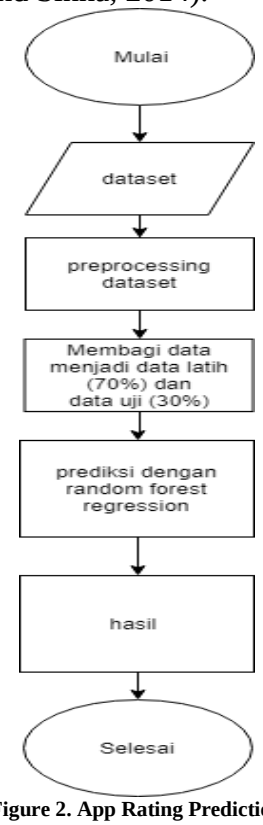

Figure 2. App Rating Prediction

There are three important aspects in using the random forest method.

a. perform bootstrap sampling to build a prediction tree.
b. each decision tree predicts with a random predictor.
c. then random forest makes predictions by combining the results of each decision tree by means of majority vote for classification or average for regression.

RESULT AND DISCUSSION

This research will use a dataset from Google Play. This test will use a dataset that has been divided based on the results of preprocessing the data as described in the test will be carried out using a dataset that has been preprocessed using integer or float units, the process of predicting the use of python with the random forest method.

Data Sharing Results

At this stage, the distribution of data has been preprocessed as described in the previous chapter with a total data of 10840 and with 13 attributes. The following is information on the dataset used, which can be seen in the following table:

Table 1. Information dataset

Atribute	Value	Status	Type
App	10840	non-null	int64
Category	10840	non-null	int64
Rating	9424	non-null	float64
Reviews	10840	non-null	int64
Size	10840	non-null	float64
Installs	10840	non-null	int64
Type	10840	non-null	int64
Price	10840	non-null	float64
Content Rating	10840	non-null	int64
Genres	10840	non-null	int64
Last Updated	10840	non-null	int64
Current Ver	10840	non-null	float64
Android Ver	10840	non-null	float64

From table 1 it can be seen that from 10840 there is a missing value in the Rating attribute with a value of 1416 data.

The following are some examples of missing values found in the dataset used can be seen in the following table simulation:

Table 2. Dataset before imputing missing value

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
10810	4305	1	NaN	4	3.9	100	1	0	1	13	1.53E+09	1.36	4.4
10811	4604	11	4.1	80	13	1000	1	0	1	39	1.53E+09	2.02	4.03
10812	2905	4	NaN	20	2.7	10000	1	0	1	22	1.53E+09	2.11	4.1
10813	4309	11	4	785	31	50000	1	0	4	52	1.43E+09	1.31	3
10814	4892	3	4.2	5775	4.9	500000	1	0	1	19	1.53E+09	7.046	4.2
10815	4423	4	NaN	2	6.8	100	1	0	1	22	1.53E+09	2.18	4.1
10816	5086	29	4	885	8	100000	1	0	1	108	1.45E+09	1.061293	5
10817	4888	12	NaN	96	1.5	10000	1	0	1	60	1.46E+09	2.3	2.2
10818	4368	3	3.3	52	3.6	5000	1	0	4	19	1.50E+09	0.34	4.1
10819	4608	11	5	22	8.6	1000	1	0	4	39	1.53E+09	3.8	4.1
10820	7097	11	NaN	6	2.5	50	1	0	1	52	1.53E+09	1	4.03
10821	6842	25	NaN	0	3.1	10	1	0	1	82	1.51E+09	1	4.4
10822	5828	31	NaN	1	2.9	100	1	0	1	114	1.52E+09	1	4.03
10823	2405	20	NaN	67	82	10000	1	0	1	71	1.53E+09	2.22	4.4
10824	6528	27	NaN	7	7.7	100	1	0	4	101	1.52E+09	1	4

From table 2, it can be seen in the red mark that there are empty values or NaN values in the Rating attribute. As for data sharing, it will be divided by deleting empty data.

At this stage, we will delete empty data in the dataset by dropping data using python. The results can be seen in the following image:

Table 3. Dataset information after deletion of missing data

Atribute	Value	Status	Type
App	10840	non-null	int64
Category	10840	non-null	int64
Rating	10840	non-null	float64
Reviews	10840	non-null	int64
Size	10840	non-null	float64
Installs	10840	non-null	int64
Type	10840	non-null	int64
Price	10840	non-null	float64
Content Rating	10840	non-null	int64
Genres	10840	non-null	int64
Last Updated	10840	non-null	int64
Current Ver	10840	non-null	float64
Android Ver	10840	non-null	float64

Table 3 shows the values for all attributes are the same, which means that an empty value in the Rating attribute will delete all data rows.

The results of the testing division

In this study, the results of the missing value imputation research will be displayed to be used in rating predictions on the Google Play Store using the random forest algorithm. This study will divide the data as much as 10840 divided into 2, with a ratio of 70:30, the number of training data consists of 7588 data and the number of testing data consists of 3252 data. Performance testing based on MAE, RMSE, and MSE. Then the performance evaluation of random forest is carried out using the measurement parameter, namely accuracy.

Random Forest Algorithm Test Results

The first test will be carried out using experiments with data whose missing values have been removed. The experiment was conducted using a random forest algorithm with the parameters of the number of trees of 200 and the depth of the tree of 10, 20, and 30. The test will use accuracy and performance measurements as a comparison of results. The test is carried out using the number of trees as much as 200 with a tree depth of 10, 20, and 30. The test will use accuracy and performance. The following are the results of the test with the number of trees 200 and into trees 10, 20, and 30.

Table 4. Testing with tree 200 and deep tree 10, 20, 30.

nilai K Imputasi	N_estimator	Deep Tree	MAE	RMSE	MSE	Akurasi
Tanpa Imputasi	**200**	**10**	0.242	0.4	0.16	0.938
		20	0.241	0.399	0.159	0.938
		30	0.242	0.401	0.161	0.938

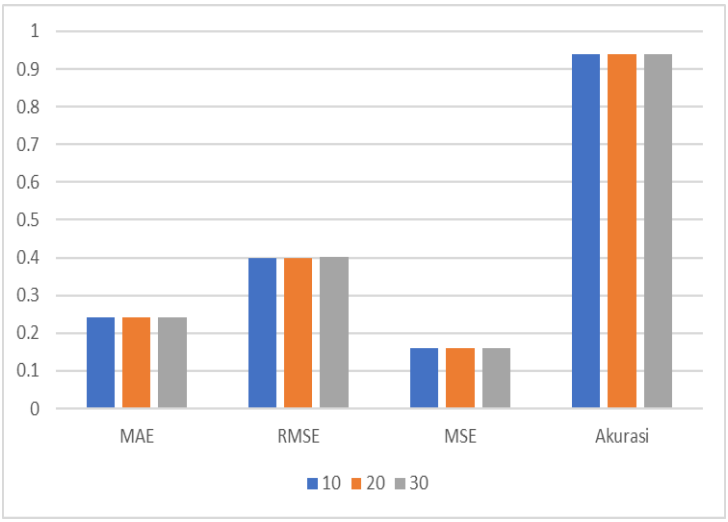

Figure 3. Grafik performance tanpa imputasi deep tree dengan tree 200

From Table 4. it can be seen that in general, the accuracy value can vary for each test. The test was carried out 3 times using deep tree values of 10, 20, 30 with a total of 200 trees. From these tests, the highest accuracy results were 93.8% MAE 0.399, RMSE 0. 9551, and MSE 0.159.

CONCLUSION

Ratings are usually given by the user and are used as a benchmark to find out whether the application made is good or there are still weaknesses. If there are weaknesses, then, by using the prediction model used, developers can find out what factors are the weaknesses of the application. Based on the problems that occur, the Random Forest algorithm has the best performance from other algorithms in helping to find weaknesses in the Google play store data set. With an accuracy of 93.8%, MAE 0.339, RMSE 0.9551 and MSE 0.159.

Acknowledgement: None

Conflict of Interest: None

Source of Funding: None

REFERENCES

1. Andreas Chandra (2017) Perbedaan Supervised And Unsupervised Learning. Available at: https://datascience.or.id/article/Perbedaan-Supervised-and-Unsupervised-Learning-5a8fa6e6.
2. Appbrain (2021) Number of Android Apps on Google Play. Available at: https://www.appbrain.com/stats/number-of-android-apps.
3. Breiman, L. (2001) 'Random Forests', Machine Learning, 45(1), pp. 5–32. doi: 10.1023/A:1010933404324.
4. Burkov, A. (2019) 'The Hundred-Page Machine Learning Book-Andriy Burkov', Expert Systems, 5(2), pp. 132–150. doi: 10.1111/j.1468-0394.1988.tb00341.x.
5. Chand, N. et al. (2016) 'A comparative analysis of SVM and its stacking with other classification algorithm for intrusion detection', Proceedings - 2016 International Conference on Advances in Computing, Communication and Automation, ICACCA 2016. doi: 10.1109/ICACCA.2016.7578859.
6. google (2012) Introducing Google Play: All your entertainment, anywhere you go, googleblog.
7. Hartmann-Boyce, J. et al. (2017) 'Insights From Google Play Store User Reviews for the Development of Weight Loss Apps: Mixed-Method Analysis', JMIR mHealth and uHealth, 5(12), p. e203. doi: 10.2196/mhealth.8791.
8. Hengshu Zhu et al. (2014) 'Popularity Modeling for Mobile Apps: A Sequential Approach', IEEE Transactions on Cybernetics, 45(7), pp. 1303–1314. doi: 10.1109/tcyb.2014.2349954.
9. Kulkarni, V. Y. and Sinha, P. K. (2014) 'Effective Learning and Classification using Random Forest Algorithm', International Journal of Engineering and Innovative Technolgy, 3(11), pp. 267–273.
10. Shen, S., Lu, X. and Hu, Z. (2017) 'Towards Release Strategy Optimization for Apps in Google Play'. Available at: http://arxiv.org/abs/1707.06022.
11. Zhu, H. et al. (2014) 'Mobile App Recommendations with Security and Privacy Awareness Categories and Subject Descriptors', Proc. of the 20th ACM SIGKDD international conference on Knowledge Discovery and Data mining (KDD), pp. 951–960. doi: 10.1145/2623330.2623705.

How to cite this article: Khaliq A, Hariyanto E, Batubara S. Predict app rank on Google play using the random forest method. *International Journal of Research and Review*. 2021; 8(9): 436-441. DOI: *https://doi.org/10.52403/ijrr.20210955*

******