International Journal of Research and Review
Vol.7; Issue: 3; March 2020
Website: www.ijrrjournal.com
Research Paper *E-ISSN: 2349-9788; P-ISSN: 2454-2237*

KNN Imputation Missing Value For Predictor App Rating on Google Play Using Random Forest Method

Abdul Khaliq[1], Pahala Sirait[2], Andri[2]

[1]Postgraduate Student at STMIK Mikroskil, Medan, Indonesia, 20212
[2]Lecturer at STMIK Mikroskil, Medan, Indonesia, 20212

Corresponding Author: Abdul Khaliq

ABSTRACT

Developers and application users are key to the market's impact on application development. In the development of application, developers need to accurately predict applications in the market, accurate prediction results are crucial in showing the rating of the user affects the success of an application. In data retrieval, there is missing data. Lost Data is done by the process of missing value Imputasi using KNN Imputation. Predictions will be done using the random forest algorithm as a method used to predict app ratings. The combination of the KNN method for the first imputation of using a random forest algorithm is better than without imputation. It can be seen from using a random forest algorithm with an average of 91,4465% accuracy results, the result is better than the prediction without the imputation of the missing value with an accuracy result of 75,8465%.

Keywords: Imputation Missing Value, App Rating, Prediction, KNN Imputation, Random Forest

INTRODUCTION

Significant mobile app market growth has a huge impact on digital technology with the number of applications available in the Google Play store until March 2019 around 2.6 million and will continue to grow as time goes on (Appbrain, 2019). Developers and application users are key to the market's impact on application development (Hengshu Zhu et al., 2014). In developing application developers need to accurately predict applications in the market, because accurate prediction results are crucial 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 liked and disliked on weight loss applications and weight tracking. The results of the Hartmann-Boyce et al study showed that the rating of the user affects the success of an application (Hartmann-Boyce et al., 2017). App Rating also affects the application's popular recommendation system on Google Play market with criteria using category parameters, number of install, rating, reviews (Zhu et al., 2014).

To predict the rating of the application there are several methods used as in 2017, Chen et al performs a comparison between Logistic Model Tree (LMT), Random Forest (RF) and Decision Tree (CART) methods to make predictions on insecurity Landslides. The results showed that the results of the third comparison of the method resulted in a random forest model has the best prediction compared with LMT or CART model with Area Under Curve (UAC) value of 0837 and with value predictive Accuracy of 0772. In another study comparing the Random Forest with K-Nearest Neighbors to the HAR dataset (human activity recognition, the result of the comparison is achieved by using the Random forest method. With a value of 93.13% (Bindu, BhanuJyothi and Suryanarayana, 2017). On other studies where comparison between SVM was combined with other classfiers such as BayesNet, AdaBoost, Logistic, IBK, J48, Random Forest, JRip, OneR and SimpleCart, the results of the study gained SVM Combined with Random Forest get good results with a value of 97.50% compared to only using SVM with a value of 91.81% (Chand et al., 2016)

To be able to predict the rating of an app properly it requires complete data. However, the problems that often arise in a given data are the incompleteness of the data in a variable or often referred to as a missing value. To solve the missing value in the dataset used is to fill out a missing value with a possible value based on the information available on the data or usually referred to as an imputation (Aljuaid and Sasi, 2017). Year 2019, Jadhav, Pramod and Ramanathan do research on the comparison of method mean imputation, median imputation, KNN imputation, predictive mean maching, Bayesian linear regression, linear regression and random sample to overcome the missing Value on numeric datasets (Jadhav, Pramod and Ramanathan, 2019).

The seven methods are tested with several datasets such as wine, concrete, liver patient and seed for the process of data imputation. The results of the tests were conducted indicating that the method KNN imputation had the lowest NRMSE value of 0.087871 compared to other methods. In other research is to do a comparison between KNN, litewise deletion and Mean Imputation to address multiple imputation on the dataset. The result of a third comparison of the KNN method is an imputation method that gets high accuracy results with a value of 74.5% (Minakshi, Vohra and Gimpy, 2014)

LITERATURE REVIEW

Missing Value

Missing value is data or information that is missing or unavailable on the research subject in certain variables due to non-sampling error factors. Missing value has a small impact on the end result when the amount of the missing value is small or small in size. However when the number of missing values is very large then they greatly affect the outcome of the final data analysis or lower the accuracy.

Machine Learning

Machine Learning is a computer science field related to building algorithms that, useful, rely on a set of examples from several phenomena. These examples can come from nature, created by humans or produced by other algorithms. Machine learning can also be defined as a practical problem solving process by 1) collecting datasets, and 2) algorithmically build a statistical model based on the dataset. The statistical Model was assumed to be used somehow to solve practical problems. To save keystrokes, I used the term "learn " and "Machine learning" alternately (Burkov, 2019).

Supervised Learning

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

K-Nearest Neighbours Imputation

K-Nearest Neighbor Imputation (K-NNI) is a method that implements the closest neighbor technique. This method is commonly used in the process of imputation or fulfillment of missing values. It takes the closest neighbor value to populate the missing value that has the same attributes. The number of neighbors taken is the same depending on the value of K entered the number of closest observations (Minakshi, Vohra and Gimpy, 2014).

Missing Value K-Nearest Neighbours Imputation

The missing value handling with KNN begins with determining a number of nearby neighbors or nearby observations symbolized by K, then calculating the smallest distance from each observation that does not contain a missing value.

Random Forest

Random Forest (RF) is a tree-based ensemble method and developed to overcome the shortcomings of the Classification and Regression Tree (CART) methods. RF consists of a large number of Classification and Regression decision tree weaknesses, which are grown in parallel to reduce bias and variance of models at the same time (Breiman, 2001).

MATERIALS & METHODS

Methodology

This research is done with the stages that will be done starting from determining the dataset that is missing value. The next step will be the missing value process using K-NN. The next stage will be the process of predicting the app rating using Random Forest. After all process stages are ready, the next stage will be conducted analysis of the results obtained by comparing to the value of RMSE (Root Mean Squared Error) to determine the accuracy of the outcome of the results and prediction accuracy results are calculated By looking at the accuracy percentage.

Preprocessing Data

The preprocessing of the data used is to convert an attribute value into a numeric form to minimize the error. As for the tools used in preprocessing by using a Python jupyter application.

a. Conversion of APP attribute values
b. Conversion of Category attribute values
c. Delete a Symbol on the value of Installs attribute
d. Conversion of types attribute values
e. Conversion of price attribute value
f. Convesi Last Update attribute values
g. Convert attribute value to Android Ver
h. Conversion of Current Ver attribute values
i. Convert attribute value to Size
j. Conversion of Content Rating attribute values
k. Convert Genres attribute value

KNN Imputation

As described in the previous chapter, KNN Imputation is used to evaluate the lost data. Data is obtained from Google Play DataSet with attributes such as: APP, Category, Rating, Reviews, Size, Installs, Type, Price, Content Rating, Genres, Last Updated, Current Ver, Android Ver.

App Rating Predictions

The random forest method is the development of the CART (Classification and Regression Tree) method by implementing the bootstrap aggregating (bagging) and random feature Selection Breiman (2001) methods. A Random forest is one of the methods used for classification and regression. This method is an ensemble of learning methods using the decision tree as a base classifier built and combined (Kulkarni and Sinha, 2014). There are three important aspects to using a random forest method.

a. Perform bootstrap sampling to build the prediction tree.
b. Each decision tree predicts with a random predictor.
c. Then random forest make predictions by combining the results of each decision tree by majority vote for classification or average for regression.

RESULT

This research will later use a dataset that comes from Google Play. This test will use a dataset that has been split based on the results of the imputation and without the imputation as described in the test will be conducted into two phases, namely by using a dataset that has been balanced with the KNN method Imputation and remove, predictive process using Python with random forest method.

Data Sharing Results

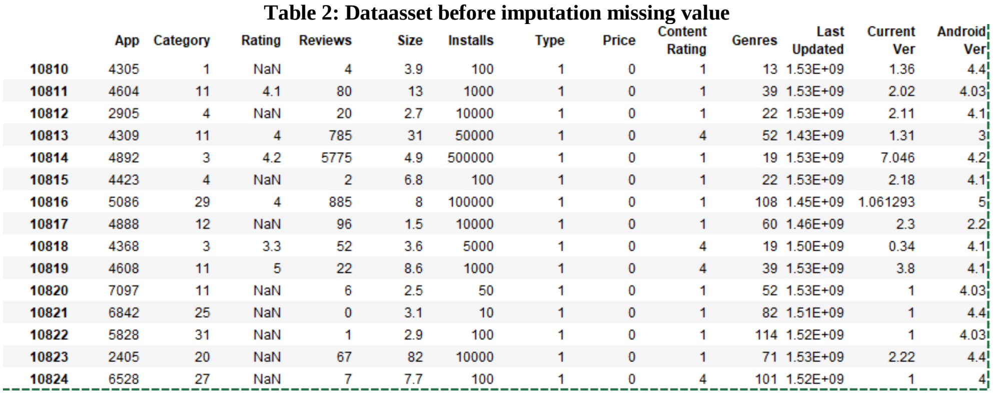
Table 1: Information Without Imputation Dataset

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

At this stage the data sharing has been done preprocessing as described in the previous chapter with a data amount of 10840 and with 13 attributes. Here is the dataset information used can be seen in the following table:

From the table 4-1 it can be seen that from the 10840 there is a missing value in the Rating attribute with a value of 1416 data. Here are some examples of missing values that can be used on the dataset as shown in the following simulation table:

Table 2: Dataasset before imputation 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 can be seen on the red mark there is an empty value or NaN value in the Rating attribute. As for the data sharing will be divided by deleting the empty data and imputation of missing value with KNN with the parameter K. At this stage will do the deletion of the empty data on the dataset by Dropdata 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

From the table 3 shows the values of all the same meanings which means that the empty values in the Rating attribute will delete the entire row of data.

Result of determination and testing Division

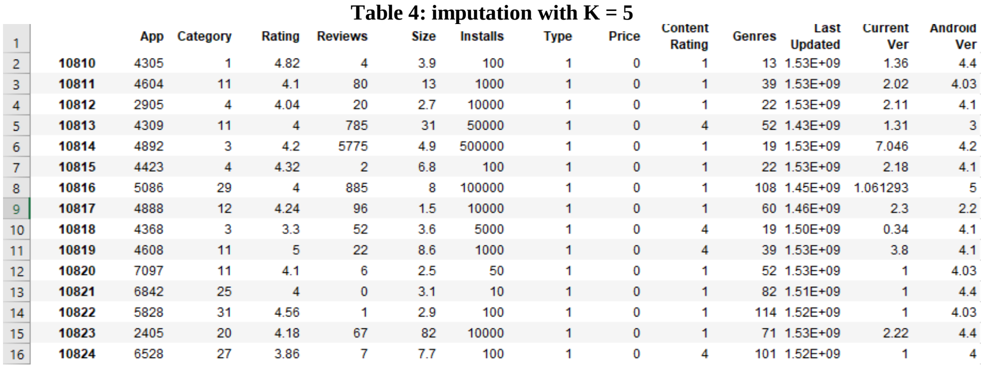
In this research will be shown the results of a missing value Imputasi research to use on the prediction rating in Google Play Store using a random forest algorithm. This research will compare the performation between the use of different K values on the missing value imputation by using values K = 5, K = 10, K = 15, K = 20 and K = 25. Data obtained from Imputasi with a different value K then done rating prediction by using the number of trees and the depth of tree in the various random forest algorithms that are: 200, 300, 400 with a tree depth of 10. Data that has been imputated by using the KNN Imputation is then divided by as much as 10840 divided into 2, with a comparison of 70:30, the amount of data training consists of 7588 data and the amount of data testing consists of 3252 data. Performance testing based on MAE, RMSE and MSE. Then conducted evaluation of the performance of the random forest by using measurement parameters that is accuracy.

The missing value imputation test is divided into five dataset-sharing criteria, as follows:

1. Imputation with K = 5

At this stage will imputasi missing value by using the KNN Imputation method using the value parameter K =. 5 Here are some data that have been done Imputasi missing value seen in the following table:

Table 4: imputation with K = 5



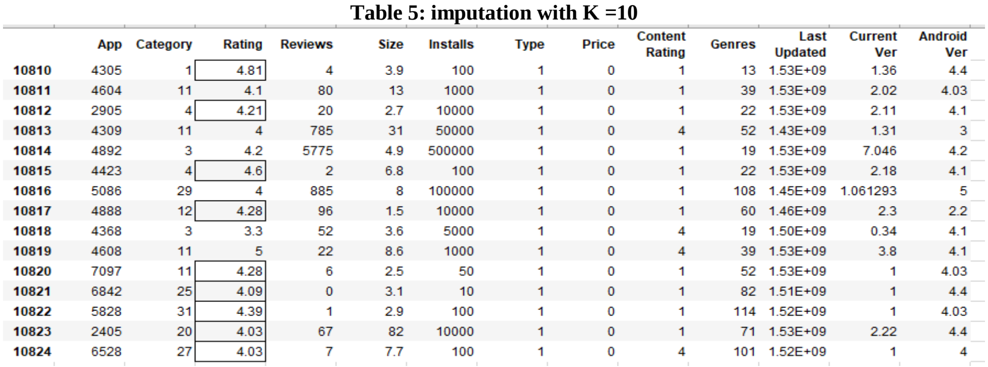
1		App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
2	**10810**	4305	1	4.82	4	3.9	100	1	0	1	13	1.53E+09	1.36	4.4
3	**10811**	4604	11	4.1	80	13	1000	1	0	1	39	1.53E+09	2.02	4.03
4	**10812**	2905	4	4.04	20	2.7	10000	1	0	1	22	1.53E+09	2.11	4.1
5	**10813**	4309	11	4	785	31	50000	1	0	4	52	1.43E+09	1.31	3
6	**10814**	4892	3	4.2	5775	4.9	500000	1	0	1	19	1.53E+09	7.046	4.2
7	**10815**	4423	4	4.32	2	6.8	100	1	0	1	22	1.53E+09	2.18	4.1
8	**10816**	5086	29	4	885	8	100000	1	0	1	108	1.45E+09	1.061293	5
9	**10817**	4888	12	4.24	96	1.5	10000	1	0	1	60	1.46E+09	2.3	2.2
10	**10818**	4368	3	3.3	52	3.6	5000	1	0	4	19	1.50E+09	0.34	4.1
11	**10819**	4608	11	5	22	8.6	1000	1	0	4	39	1.53E+09	3.8	4.1
12	**10820**	7097	11	4.1	6	2.5	50	1	0	1	52	1.53E+09	1	4.03
13	**10821**	6842	25	4	0	3.1	10	1	0	1	82	1.51E+09	1	4.4
14	**10822**	5828	31	4.56	1	2.9	100	1	0	1	114	1.52E+09	1	4.03
15	**10823**	2405	20	4.18	67	82	10000	1	0	1	71	1.53E+09	2.22	4.4
16	**10824**	6528	27	3.86	7	7.7	100	1	0	4	101	1.52E+09	1	4

Judging from the table 4 the NaN value of the rating attribute has been replaced by an imputation of the missing value using the KNN Imputation method with the value parameter K = 5. The test results in the use of parameter K = 5 parameters on missing value with RMSE value of 1.5129

2. Imputation with K = 10

At this stage will imputasi missing value by using the KNN Imputation method using the value parameter K =. 10 Here are some data that has been done Imputasi missing value seen in the following table:

Table 5: imputation with K =10



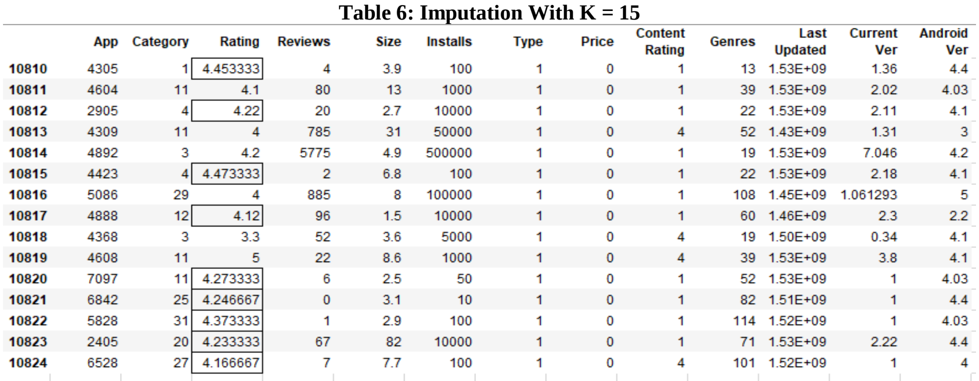
	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
10810	4305	1	4.81	4	3.9	100	1	0	1	13	1.53E+09	1.36	4.4
10811	4604	11	4.1	80	13	1000	1	0	1	39	1.53E+09	2.02	4.03
10812	2905	4	4.21	20	2.7	10000	1	0	1	22	1.53E+09	2.11	4.1
10813	4309	11	4	785	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	4.6	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	4.28	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	4.28	6	2.5	50	1	0	1	52	1.53E+09	1	4.03
10821	6842	25	4.09	0	3.1	10	1	0	1	82	1.51E+09	1	4.4
10822	5828	31	4.39	1	2.9	100	1	0	1	114	1.52E+09	1	4.03
10823	2405	20	4.03	67	82	10000	1	0	1	71	1.53E+09	2.22	4.4
10824	6528	27	4.03	7	7.7	100	1	0	4	101	1.52E+09	1	4

Judging from the table 5 the NaN value of the rating attribute has been replaced by an imputation of the missing value using the KNN Imputation method with the value parameter K = 10. The test results in the use of parameter K = 10 parameters on missing value with RMSE value of 1.5087

3. Imputation With K = 15

At this stage will imputasi missing value using KNN Imputation method using the value parameter K = 15 Here are some data that has been done Imputasi missing value seen in the following table:

Table 6: Imputation With K = 15



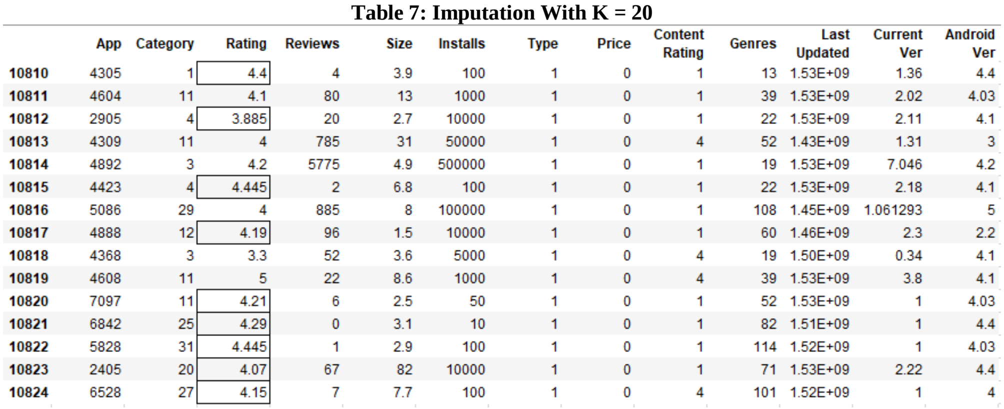
	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
10810	4305	1	4.453333	4	3.9	100	1	0	1	13	1.53E+09	1.36	4.4
10811	4604	11	4.1	80	13	1000	1	0	1	39	1.53E+09	2.02	4.03
10812	2905	4	4.22	20	2.7	10000	1	0	1	22	1.53E+09	2.11	4.1
10813	4309	11	4	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	4.473333	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	4.12	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	4.273333	6	2.5	50	1	0	1	52	1.53E+09	1	4.03
10821	6842	25	4.246667	0	3.1	10	1	0	1	82	1.51E+09	1	4.4
10822	5828	31	4.373333	1	2.9	100	1	0	1	114	1.52E+09	1	4.03
10823	2405	20	4.233333	67	82	10000	1	0	1	71	1.53E+09	2.22	4.4
10824	6528	27	4.166667	7	7.7	100	1	0	4	101	1.52E+09	1	4

Judging from the table 6 the NaN value of the rating attribute has been replaced by an imputation of the missing value using the KNN Imputation method with the value parameter K = 15. The test results in the use of parameter K = 15 in the missing value with the RMSE value of 1.5062.

4. Imputation With K = 20

At this stage will imputasi missing value using KNN Imputation method using the value parameter K = 20 Here are some data that has been done Imputasi missing value seen in the following table:

Table 7: Imputation With K = 20



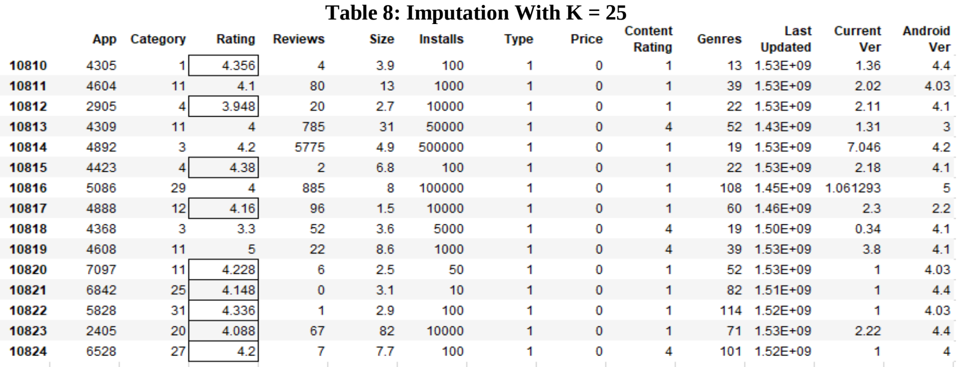
	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
10810	4305	1	4.4	4	3.9	100	1	0	1	13	1.53E+09	1.36	4.4
10811	4604	11	4.1	80	13	1000	1	0	1	39	1.53E+09	2.02	4.03
10812	2905	4	3.885	20	2.7	10000	1	0	1	22	1.53E+09	2.11	4.1
10813	4309	11	4	785	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	4.445	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	4.19	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	4.21	6	2.5	50	1	0	1	52	1.53E+09	1	4.03
10821	6842	25	4.29	0	3.1	10	1	0	1	82	1.51E+09	1	4.4
10822	5828	31	4.445	1	2.9	100	1	0	1	114	1.52E+09	1	4.03
10823	2405	20	4.07	67	82	10000	1	0	1	71	1.53E+09	2.22	4.4
10824	6528	27	4.15	7	7.7	100	1	0	4	101	1.52E+09	1	4

Judging from the table 7 the NaN value of the rating attribute has been replaced by an imputation of the missing value using the KNN Imputation method with the value parameter K = 20. The test results in the use of parameter K = 20 parameters on missing value with RMSE value of 1.5052.

5. Imputation With K = 25

At this stage will imputasi missing value using KNN method Imputation using the value parameter K = 25 Here are some data that has been done Imputasi missing value seen in the following table:

Table 8: Imputation With K = 25



	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
10810	4305	1	4.356	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	3.948	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	4.38	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	4.16	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	4.228	6	2.5	50	1	0	1	52	1.53E+09	1	4.03
10821	6842	25	4.148	0	3.1	10	1	0	1	82	1.51E+09	1	4.4
10822	5828	31	4.336	1	2.9	100	1	0	1	114	1.52E+09	1	4.03
10823	2405	20	4.088	67	82	10000	1	0	1	71	1.53E+09	2.22	4.4
10824	6528	27	4.2	7	7.7	100	1	0	4	101	1.52E+09	1	4

Judging from the table 8 the NaN value of the rating attribute has been replaced by an imputation of the missing value using the KNN Imputation method with the value parameter K = 25. The test results in the usability scenario of the K = 25 parameter on the missing value with the RMSE value of 1.5045. As for the general testing results by using KNN imputation with the parameters of values K = 5, K = 10, K = 15, K = 20 and K = 25 can be seen in the following table:

Table 9: Test results K = 5, K = 10, K = 15, K = 20 dan K = 25

K Value	RMSE
5	1.5129
10	1.5087
15	1.5062
20	1.5052
25	1.5045

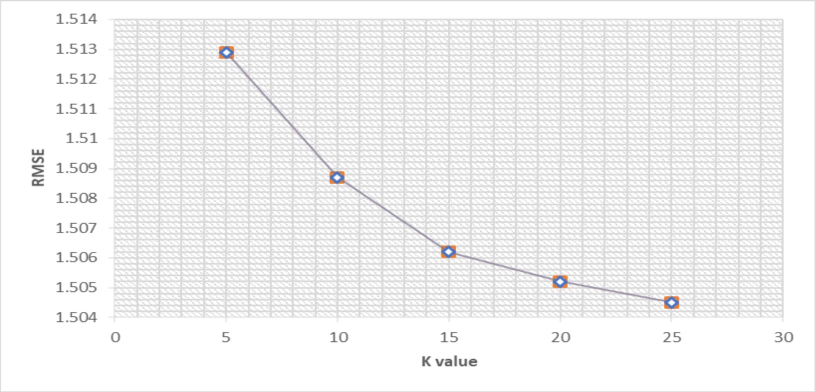


Figure 1: Chart of test results K = 5, K = 10, K = 15, K = 20 dan K = 25

At Figure 1 It is seen that in general each test stage can provide a different RMSE value. The variation is due to the difference of parameter K on the imputation of missing value using KNN Imputation. At the test stage RMSE value with a value of K = 25 can be better compared to the RMSE value by the number below. The results of testing using KNN Imputation and without the imputation of the missing value as follows:

Random algorithm test Results

Testing will be conducted using experiments with data that has been balanced using the best value of KNNI with a value parameter of K = 25. Experiments were conducted using a random forest algorithm with the number of tree parameters of 200, 400 and 600 and the tree depth of 10, 20 and 30. Testing will use accuracy and performance measurements as a comparison of results:

1. Test with the number of trees as much as 200 and the depth of tree 10, 20, and 30
2. Test with the number of trees as much as 400 and the depth of tree 10, 20, and 30
3. Test with the number of trees as much as 600 and the depth of tree 10, 20, and 30

As for the test results in general using the algorithm value Imputasi K = 25 and prediction rating with a random forest algorithm with the number of tree parameters 200, 400, 600 and the amount of tree depth 10, 20, 30 can be seen in table 4-13.

Table 10: Test result with K = 25

K Value	N_estimator	Deep Tree	MAE	RMSE	MSE	Accuracy
K = 25	**200**	**10**	0.2972	0.4756	0.2262	91,2438
		20	0.2897	0.4736	0.2243	91,3165
		30	0.2871	0.47	0.2209	91,4465
	400	**10**	0.2968	0.4758	0.2263	91,2372
		20	0.289	0.4717	0.2225	91,3844
		30	0.2892	0.4729	0.2236	91,3413
	600	**10**	0.297	0.4752	0.2258	91,2579
		20	0.2887	0.4713	0.2221	91,3999
		30	0.288	0.4714	0.2222	91,3972

Based on the table 10 can be seen that with an increase in the number of trees and the depth of tree gives better accuracy and performance, seen at the depth of tree 30 in the number of trees 200 with accuracy value 91.4465%, MAE 0.2871, RMSE 0.4700 and MSE 0.2209.

Test result Random Forest algorithm without imputation

The first test will be done by using a trial with the data that is missing value removed. Experiments were conducted using a random forest algorithm with the number of tree parameters of 200, 400 and 600 and the tree depth of 10, 20 and 30. Testing will use accuracy and performance measurements as a comparison of results.

1. Test with the number of trees as much as 200 and the depth of tree 10, 20, and 30
2. Test with the number of trees as much as 400 and the depth of tree 10, 20, and 30
3. Test with the number of trees as much as 600 and the depth of tree 10, 20, and 30

As for general testing results using missing data removed and prediction of rating with random forest algorithm with the number of tree parameters 200, 400, 600 and total tree depth 10, 20, 30 can be seen in table 4-38.

Table 11: Test result without imputation

K Value	N_estimator	Deep Tree	MAE	RMSE	MSE	Accuracy
Without Imputation	**200**	**10**	0.5233	0.9472	0.8973	75,8465
		20	0.5179	0.9498	0.9021	75,7165
		30	0.516	0.9516	0.9056	75,6242
	400	**10**	0.5227	0.948	0.8987	75,8084
		20	0.5161	0.9487	0.9001	75,7709
		30	0.5167	0.9517	0.9058	75,6183
	600	**10**	0.5219	0.9475	0.8977	75,8352
		20	0.517	0.9493	0.9013	75,7403
		30	0.517	0.9508	0.904	75,6662

According to table 11 it can be seen that with the number of trees 200 and the depth of tree 10 provides better accuracy and performance, seen at the depth of tree 10 in the number of trees 200 with an accuracy value of 75.8465% MAE 0.5233, RMSE 0.9472 and MSE 0.8973.

DISCUSSION

Misssing Value with KNN Imputation

In the previous section test Imputasi missing value by using KNN Imputation using the Parameters K = 5, K = 10, K = 15, K = 20 and K = 25. On the KNN Imputation is done Euclidean distance on each attribute with data contained missing value. To find the value of neighbors to a minimum distance, which is a Selanjukan to enter the value of Euclidean distance to the amount of K used that will be used as the parameter K to be used, the last stage is performed evalue performance with Using RMSE. The test results can be seen in the following table:

Table 12: Result of performance KNN Imputation

K Value	RMSE
5	1.5129
10	1.5087
15	1.5062
20	1.5052
25	1.5045

Based on the table 12 can be seen that generally determination of the parameter K affects the performance of the missing value Imputasi with KNN Imputation shown in the value of Root Mean Square Error (RMSE) obtained based on the Imputation results.

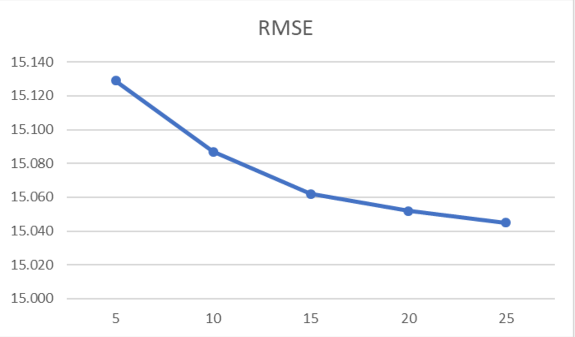


Figure 2: Graph of performance results KNN Imputation

The results of the analysis obtained according to the table 12, by comparing the performance results on the data imputasi, increase the number of K values in the test will increase the performance of KNN Imputation, the empirical value K Affected by the data type and the missing value ratio.

Rating prediction with Random Forest

In prediction rating by using random forest using data that has been in Imputasi with KNN imputation with the parameter K = 25 and done also without using the Imputasi missing value, then the result is compared. Testing of the random forest algorithm is done using the number of tree parameters 200, 400 and 600, tree depth 10, 20, 30. The results of the test are addressed to the accuracy value of the test results can be seen in the table:

Table 13: Comparison of imputation results with K = 25 and no imputation

K Value	**N_estimator**	**Deep Tree**	**MAE**	**RMSE**	**MSE**	**Accuracy**
K = 25	**200**	**10**	0.2972	0.4756	0.2262	91,2438
		20	0.2897	0.4736	0.2243	91,3165
		30	0.2871	0.47	0.2209	91,4465
	400	**10**	0.2968	0.4758	0.2263	91,2372
		20	0.2890	0.4717	0.2225	91,3844
		30	0.2892	0.4729	0.2236	91,3413
	600	**10**	0.2970	0.4752	0.2258	91,2579
		20	0.2887	0.4713	0.2221	91,3999
		30	0.2880	0.4714	0.2222	91,3972
Average			**0.2906**	**0.4727**	**0.2234**	**91,3476**
Without Imputation	**200**	**10**	0.5233	0.9472	0.8973	75,8465
		20	0.5179	0.9498	0.9021	75,7165
		30	0.516	0.9516	0.9056	75,6242
	400	**10**	0.5227	0.948	0.8987	75,8084
		20	0.5161	0.9487	0.9001	757,709
		30	0.5167	0.9517	0.9058	75,6183
	600	**10**	0.5219	0.9475	0.8977	75,8352
		20	0.517	0.9493	0.9013	75,7403
		30	0.517	0.9508	0.904	75,6662
Average			**0.5181**	**0.9496**	**0.9019**	**75,7225**

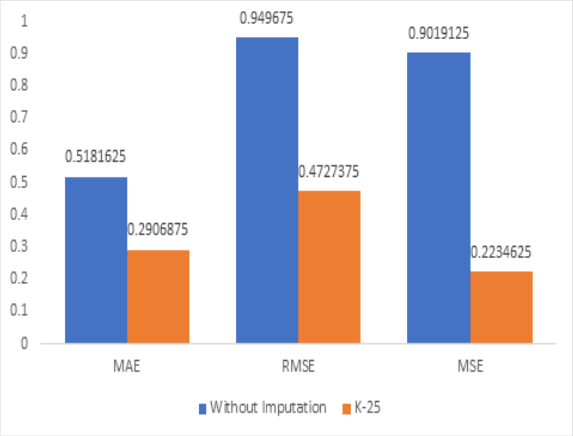


Figure 3: Graph of Comparison of imputation results with K = 25 and no imputation

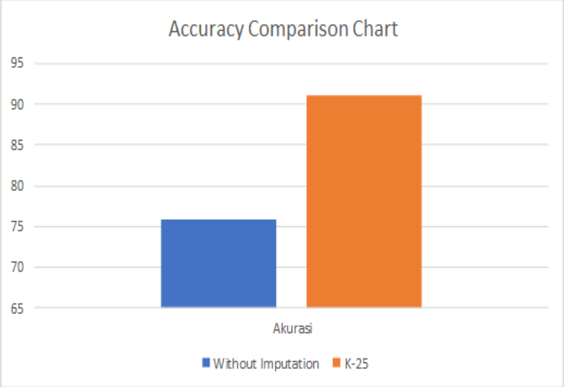


Figure 3: Graphic comparison of imputation accuracy with K = 25 and without imputation

The analysis results are obtained according to the table 13, namely:

1. Of the whole testing by increasing the number of trees on the predictions with a random forest algorithm does not get good results. It can be seen from the evaluation value with the number of trees 200 has greater results compared to the number of trees 400 and 600. It also follows by increasing the amount of depth of tree. So by increasing the depth of the tree to be larger also does not get better results.
2. According to the performance value and the rating obtained from the overall data is done for testing using the Imputasi missing value with KNN imputation better than without the Imputasi missing value by comparison 91.4465% and 75.8465%

CONCLUSION

Throughout the test results that have been done to Imputasi missing value by using KNN Imputation by predicting the rating using a random forest algorithm, it is obtained the following conclusions:

1. In this study by increasing the K parameter on the imputation of the missing value with KNN resulted in a better value. And obtained a value of K = 25 with the highest performance, so as to predict the rating on the study using K = 25
2. In the process of predictive rating by using a random forest with the number of tree parameters with the value 200 and deep tree 30 produce the highest accuracy value, it does not apply by

increasing the number of tree parameters and tree depth

3. From the experiments that have been carried out the best combination by using the value K = 25 on the Imputasi missing value and predicted attribute rating using a random forest algorithm with an average result of 91.4465% accuracy, results Better than the prediction without Imputasi missing value with an accuracy result of 75.8465%

REFERENCES

1. Aljuaid, T. and Sasi, S. (2017) ‘Proper imputation techniques for missing values in data sets’, Proceedings of the 2016 International Conference on Data Science and Engineering, ICDSE 2016. doi: 10.1109/ICDSE.2016.7823957.
2. Andreas Chandra (2017) Perbedaan Supervised And Unsupervised Learning. Available at: https://datascience.or.id/article/Perbedaan-Supervised-and-Unsupervised-Learning-5a8fa6e6.
3. Appbrain (2019) Number of Android apps on Google Play. Available at: https://www.appbrain.com/stats/number-of-android-apps.
4. Bindu, K. H., BhanuJyothi, K. and Suryanarayana, D. (2017) ‘A Comparative Study of Random Forest & K – Nearest Neighbors on HAR dataset Using Caret’, International Journal of Innovative Research in Technology, 3(9), pp. 6–9. Available at: http://ijirt.org/Article?manuscript=144228.
5. Breiman, L. (2001) ‘Random Forests’, Machine Learning, 45(1), pp. 5–32. doi: 10.1023/A:1010933404324.
6. 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.
7. 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.
8. 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.
9. 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.
10. Jadhav, A., Pramod, D. and Ramanathan, K. (2019) ‘Comparison of Performance of Data Imputation Methods for Numeric Dataset’, Applied Artificial Intelligence. Taylor & Francis, 33(10), pp. 913–933. doi: 10.1080/08839514.2019.1637138.
11. 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.
12. Minakshi, Vohra, R. and Gimpy (2014) ‘Missing Value Imputation in Multi Attribute Data Set’, International Journal of Computer Science and Information Technologies, 5(4), pp. 5315–5321.
13. 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.
14. 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, Sirait P, Andri. KNN imputation missing value for predictor app rating on google play using random forest method . International Journal of Research and Review. 2020; 7(3): 151-160.

******