Journal of Applied and Research Computer Science and Information Systems
Vol. 2, No. 1, June 2024, pp. 132~148
ISSN: 2988-294X, DOI: 10.61098/jarcis.v2i1.173

Laptop Price Prediction Using Extreme Gradient Boosting Algorithm

Syahrani Adrianty[1], Febri Maspiyanti[1]
[1] Universitas Pancasila, Jakarta, Indonesia
syahraniadrianty30@gmail.com

Article Info

Article history:

Received May 31, 2024
Revised June 1, 2024
Accepted June 3, 2024

Keywords:

Laptop Prices
Extreme Gradient Boosting
Cross Validation
Outlier

ABSTRACT

The laptop is a support for many people in doing all activities. The number of laptop outputs with various models can affect the price of laptops. The presence of various online and offline stores causes different laptop prices and it becomes difficult to compare prices that are close to the low price range. Based on these problems, a system is needed that can predict laptop prices based on laptop specifications that are useful for people in finding a cheap price range. Data collection in this study came from bhinneka.com with 560 data and pemmz.com with 319 data collected by scrapping method. This research uses the Extreme Gradient Boosting method with evaluation techniques in the form of cross-validation resulting in an R2 score at the Bhinneka store of 0.98 and RMSE of 1250363.29 with the best cross-validation of 8. At Pemmz store produces an R2 score of 0.98 and RMSE of 1073090.92 with the best cross-validation of 6. Both results use data with outliers.

Corresponding Author:

Syahrani Adrianty
Universitas Pancasila, Jakarta, Indonesia
Email: syahraniadrianty30@gmail.com

1. INTRODUCTION

Laptops have become an important device for many people who support all activities in the digital era. From groups of students, students, workers, and business people need the capabilities of laptops [1]. The use of notebooks, in particular, has increased significantly in recent years, beating the popularity of desktops, tablets, and other mobile devices according to data released by IDC in 2024. In terms of small size and portability, laptops are a practical choice for users who need easy access to computing wherever they are [2].

Competition between different laptop brands has fuelled the provision of quality products for consumers. For some people, understanding how laptop specifications affect its price can be complicated. Factors such as brand, laptop series, storage capacity, memory, processor, graphics processor, and screen size play a role in determining the price of a laptop [3]. The presence of various e-commerce platforms adds to the complexity, with different laptop prices in each store [4]. This makes people have to check price variations from several stores to find cheap laptop prices. The impact of this habit takes a long time when comparing laptop prices from several stores and consumers find it difficult to find prices that are close to the low price range or close to the official price from the distributor. In this case, 2 official distributors that are used as a comparison to provide an overview of the prediction of the price of this laptop, namely Bhinneka distributors

and Pemmz distributors. Both distributors are large distributors that officially sell various types of laptops and equipment.

To overcome the price uncertainty felt by the public, the right approach is needed. One solution is to implement an automated prediction system using a machine-learning approach. However, some traditional methods such as multiple linear regression have limitations in handling non-linear relationships between input and output variables [5]. In addition, decision tree regressors can suffer from overfitting, where the model fits the training data too well and cannot generalize well to new data [6]. To overcome this problem, the extreme gradient boosting (XGBoost) algorithm can be a good choice. XGBoost is an effective algorithm for overcoming overfitting and maximizing accuracy [7]. Through careful tuning, XGBoost can produce price predictions with a high degree of accuracy.

Previous research shows that there is still space to improve the performance of laptop price prediction. The use of an extreme gradient boosting algorithm in predicting laptop prices with web-based applications is a solution to the previous problem. Features such as the use of laptop specifications (brand, series, CPU_type, RAM_GB, storage_GB, screen_size, and weight) as input variables for price prediction is the right approach. Users can input the desired laptop specification data, and will immediately get a price estimate without having to perform complicated analyses. This allows users to get laptop price estimates based on specific characteristics according to their needs. Therefore, this research uses the measurement of R2-score, RMSE, and MAPE [8]. Each evaluation metric will determine how well the model performs in predicting a laptop price.

2. METHOD

At this stage there is a description of the data and the flow of the method in the form of algorithms used to support research in predicting laptop prices.

A. Data

The data source in this research is data on laptop specifications and prices from the websites bhinneka.com and pemmz.com. The data that will be collected using the scraping method consists of laptop data through bhinneka.com which has 560 rows and 8 columns. While laptop data through pemmz.com has 319 rows and 8 columns. There are 8 features used such as brand, series, CPU_type, RAM_GB, storage_GB, screen_size, weight, and price. The description of these features can be seen in table 1.

Table 1. Feature description

Nama Kolom	Deskripsi
Brand	The brand column describes the corporate body of the laptop being sold.
Series	The model column describes the model name of the laptop being sold.
CPU_type	The CPU_type column describes the type of processor used.
RAM_GB	The RAM_GB column describes the amount of memory on the laptop. (in GB units)
Storage_GB	The storage_GB column describes the amount of storage capacity. (in GB units)
Screen_size	The screen_size column describes the size of the screen you have. (in inch)
weight	The weight column describes the weight of the laptop.
price	The price column describes the price of the laptop being sold. (in rupiah)

B. Algorithm

The stages of research carried out in the form of a method flow using the extreme gradient boosting algorithm in predicting laptop prices can be seen in Figure 1.

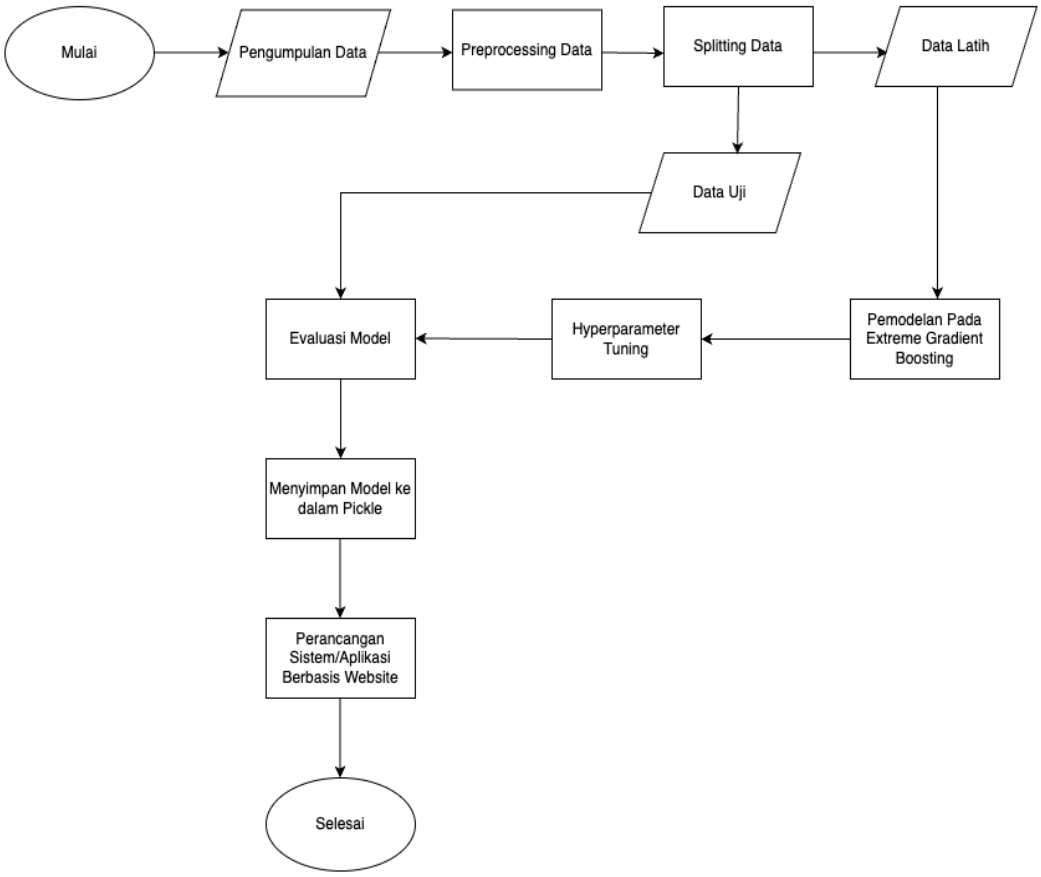

Figure 1. Stages of research

Based on the CRIPS-DM technique used in this research, there are several stages such as data collection, data preprocessing, data splitting, modelling on extreme gradient boosting, hyperparameter tuning, and model evaluation.

1. Data Collection
   At the initial stage is the collection of datasets that will be used in research. Datasets come from the bhinneka.com and pemmz.com websites obtained from the scraping method.
2. Data Preprocessing
   Datasets that have been obtained, then data preprocessing is carried out for data cleaning such as duplicates, missing values, and outliers. In addition, it performs data transformation, feature selection, and handles unbalanced classes. After going through this process, the amount of data is reduced on the Bhinneka laptop dataset from 560 to 490. In addition, the pemmz laptop dataset was reduced from 319 to 265.
3. Splitting Data
   The data that has been prepared is then separated into training data and testing data. There is a comparison for training data and testing data using k subsets or folds in k-fold experiments of 5 to 10.
4. Modelling on extreme gradient boosting.
   This stage performs modeling using Extreme Gradient Boosting. This research uses the sci-kit learn module from Python for the modeling process by installing the xgboost library first.
5. Hyperparameter Tuning
   Parameter tuning using GridSearchCV which aims to find the best combination of parameters such as max_depth, min_child_weight, gamma, subsample, comlsample_bytree, learning_rate, and n_estimators that provide optimal performance based on the selected evaluation metric.

6. Model Evaluation
   Based on the performance scores obtained from the model such as R-score, RMSE, and MAPE. Next, check the performance of the model using two evaluation techniques, namely cross-validation and train test splitting.

XGBoost has several stages in manual calculation [9]:

1. Determine the initial prediction value which can be taken randomly or from the average target value in the observation data, but there is a default value that is commonly used is 0.5. The initial prediction calculation can be seen in equation 2.1.
   $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} yi$ (2.1)
   Description:
   $\bar{y}$: average target value / initial prediction
   n : number of samples
   y_i : i-th target value
2. Finding the residual value using the target value in the observation data minus the predicted value. Residual calculations can be seen in equation 2.2
   $Residual = y - \bar{y}$ (2.2)
   Description:
   y : target values
   $\bar{y}$: average target value / initial prediction
3. Calculating similarity score

   where λ: lambda regularization parameter with a default value of 0. The similarity score calculation is used to calculate the gain value which can be seen in equation 2.3.

   $$Similarity\ score = \frac{(sum\ of\ residuals)^2}{Number\ of\ residuals + \lambda} \quad (2.3)$$

4. Calculating the gain value

   The gain value will be used for tree pruning in XGBoost. The calculation of the gain value can be seen in equation 2.4.

   $$Gain = Left_{similarity} + Right_{similarity} - Root_{similarity} \quad (2.4)$$

5. The calculation in tree pruning in XGBoost can be seen in equation 2.5
   $Gain - \gamma$ (2.5)

   If the result of equation 4 is positive then the branch is not removed. However, if the result of the calculation is negative then the branch will be removed. There is an exception if the first branch of the calculation is positive and the calculation for the root is negative, then the root is not removed. Tree creation ends with pruning.

6. After pruning is complete, then calculate the prediction result on each leaf which can be seen in equation 2.6.
   $$Output\ Value = \frac{sum\ of\ residuals}{Number\ of\ residuals + \lambda} \quad (2.6)$$
7. Calculate the new prediction of 1 tree using equation 2.7.
   $New\ prediction = \bar{y} + (learning\ rate \times output\ value)$ (2.7)

In the scenario of applying the method in laptop price prediction research using the extreme gradient boosting algorithm. Due to the limitations of researchers in calculations, researchers will display calculations using 8 samples and 7 variables that produce 1 decision tree. Table 2 contains sample data used to make decision trees.

Tabel 2. Sample Data

brand	series	CPU_type	RAM_GB	storage_GB	screen_size	weight	price_ bhinneka	price_ pemmz
0	2	10	8	512	15,6	1,80	11.099.000	11.099.000
0	2	6	4	256	14	1,90	4.899.000	5.199.000
1	32	5	16	1024	14	1,40	17.799.000	17.499.000
1	28	10	12	512	14	1,50	12.399.000	12.099.000
2	14	4	32	1024	16	1,50	30.999.000	30.999.000
2	15	10	16	512	14	1,51	14.499.000	12.625.000
3	10	11	16	1024	16	2,49	26.499.000	26.499.000
3	26	9	8	256	14	1,43	6.280.000	6.299.000

The sample data that has been collected, then determine the initial prediction value which can be calculated from the average price of the Bhinneka store as y_1 and the Pemmz store price as y_2. The initial prediction calculation uses equation 2.1.

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} yi$$

Description:
$\bar{y}$: average target value / initial prediction
n : number of samples
y_i : i-th target value

$$\bar{y} = \frac{11.099.000 \times 11.099.000}{2} = 11.099.000$$
$$\bar{y} = \frac{4.899.000 \times 5.199.000}{2} = 5.049.000$$
$$\bar{y} = \frac{17.799.000 \times 17.499.000}{2} = 17.649.000$$
$$\bar{y} = \frac{12.399.000 \times 12.099.000}{2} = 12.249.000$$
$$\bar{y} = \frac{30.999.000 \times 30.999.000}{2} = 30.999.000$$
$$\bar{y} = \frac{14.499.000 \times 12.625.000}{2} = 13.562.000$$
$$\bar{y} = \frac{26.499.000 \times 26.499.000}{2} = 26.499.000$$
$$\bar{y} = \frac{26.499.000 \times 26.499.000}{2} = 26.499.000$$

Table 3 produces an initial prediction value calculated based on the average value of the price_binneka and price_pemmz targets.

Tabel 3. Initial predicted values

y_1	y_2	$\bar{y}$
11.099.000	11.099.000	11.099.000
4.899.000	5.199.000	5.049.000
17.799.000	17.499.000	17.649.000
12.399.000	12.099.000	12.249.000
30.999.000	30.999.000	30.999.000
14.499.000	12.625.000	13.562.000
26.499.000	26.499.000	26.499.000
6.280.000	6.299.000	6.289.500

After knowing the initial prediction value, the next step is to calculate the error or residual using equation 2.2.
$Residual = y - \bar{y}$
Description:

$y : target\ value$

$\bar{y}$: average target value / initial prediction

$Residual = 11.099.000 - 11.099.000 = 0$
$Residual = 4.899.000 - 5.049.000 = -150.000$
$Residual = 17.799.000 - 17.649.000 = 150.000$
$Residual = 12.399.000 - 12.249.000 = 150.000$
$Residual = 30.999.000 - 30.999.000 = 0$
$Residual = 14.499.000 - 13.562.000 = 937.000$
$Residual = 26.499.000 - 26.499.000 = 0$
$Residual = 6.280.000 - 6.289.500 = -9.500$

Table 4 produces a residual value calculated based on the target value minus the average target value or initial prediction.

Tabel 4. Calculating the Residuals

y	$\bar{y}$	residual
11.099.000	11.099.000	0
4.899.000	5.049.000	-150.000
17.799.000	17.649.000	150.000
12.399.000	12.249.000	150.000
30.999.000	30.999.000	0
14.499.000	13.562.000	937.000
26.499.000	26.499.000	0
6.280.000	6.289.500	-9.500

The residual results that have been obtained, then determine various possible splits by dividing the data into 2 partitions. The split is to determine the boundaries of the root node by finding the average value of the two branching points, and the rest goes to the leaf node. At this stage, we conducted splitting experiments on 7 variables, as follows.

1. Splitting of brand data

Tabel 5. Splitting of brand data

Brand	Pemisahan
0	(0+1)/2 = 0,5 (1+2)/2 = 1,5 (2+3)/2 = 2,5
1	
2	
3	

2. Splitting of series data

Tabel 6. Splitting of series data

Series	Pemisahan
2	(2+10)/2 = 6 (10+14)/2 =12 (14+15)/2 = 14,5 (15+26)/2 = 20,5 (26+28)/2 = 27 (28+32)/2 = 30
10	
14	
15	
26	
28	
32	

3. Splitting of CPU_type data

Tabel 7. Splitting of CPU_type data

CPU_type	Pemisahan
4	(4+5)/2 = 4,5 (5+6)/2 = 5,5 (6+9)/2 = 7,5 (9+10)/2 = 9,5 (10+11)/2 = 10,5
5	
6	
9	
10	
11	

4. Splitting of RAM_GB data

Tabel 8. Splitting of RAM_GB data

RAM_GB	Pemisahan
4	(4+8)/2 = 6
8	(8+12)/2 = 10
12	
16	(12+16)/2 = 14
32	(16+32)/2 = 24

5. Splitting of storage_GB data

Tabel 9. Splitting of storage_GB data

Storage_GB	Pemisahan
256	(256+512)/2 = 384 (512+1024)/2 = 768
512	
1024	

6. Splitting of screen_size data

Tabel 10. Splitting of screen_size data

Screen_size	Pemisahan
14	(14+15,6)/2 = 14,8 (15,6+16)/2 = 15,8
15,6	
16	

7. Splitting of weight data

Tabel 11. Splitting of weight data

Weight	Pemisahan
1,40	
1,43	(1,40+1,43)/2 = 1,42
1,50	(1,43+1,50)/2 = 1,47
1,51	(1,50+1,51)/2 = 1,51
1,80	(1,51+1,80)/2 = 1,66
1,90	(1,80+1,90)/2 = 1,85
2,49	(1,90+2,49)/2 = 2,20

After separating several variables, then calculate the similarity score value of the various separations using equation 2.3.

$$Similarity\ score = \frac{(sum\ of\ residual)^2}{Number\ of\ residuals + \lambda}$$

- $Similarity\ score/\ Root\ similarity = \frac{(0 + (-150.000) + 0 + 0 + 150.000 + 150.000 + 937.000 + (-9.500))^2}{8+1} = 1{,}29E + 11$
- $Similarity\ score/\ Left\ similarity = \frac{(0+(-150.000)+0+0)^2}{4+1} = 4500000000$
- $Similarity\ score\ /\ Right\ similarity = \frac{(150.000 + 150.000 + 937.000 + (-9.500))^2}{4+1} = 3{,}0135E + 11$

The purpose of finding the similarity score value is to calculate the gain value. The calculation of the gain value determines the beginning of the root node and the next node using equation 2.4.

$$Gain = Left_{similarity} + Right_{similarity} - Root_{similarity}$$
$$Gain = 4500000000 + 3{,}0135E + 11 - 1{,}29E + 11 = 1{,}7685E + 11$$

If the separation has the highest gain value, then the separation of the variable becomes the root node. This stage displays the highest gain value in the variable series of 1.7685E+11 which will be used as the root node and can be seen in Figure 2.

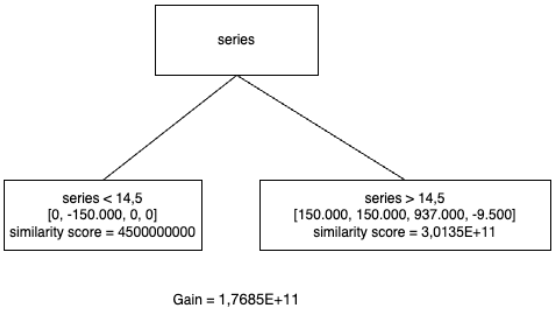

Figure 2. The highest gain value in the series in tree 1

After determining the root node on the series variable, the next stage performs a separation again for the tree that has the maximum gain. The next data separation is on the left and right nodes with the variables RAM_GB < 10 and RAM_GB < 10. This process can be seen in Figure 3.

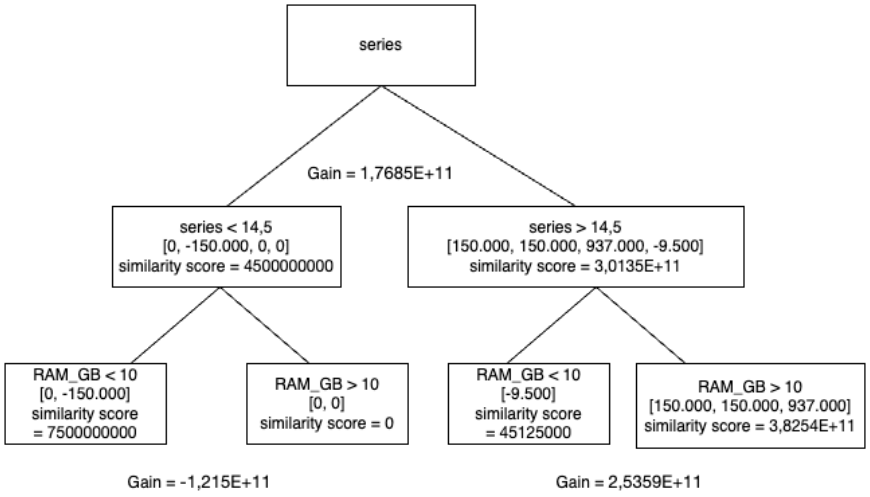

Figure 3. Splitting of continuation nodes with RAM_GB in tree 1

The separation of nodes in RAM_GB produces different gains on the left node with a gain value of -1.215E+11 and on the right node of 2.5359E+11. In addition, the right node with series > 14.5 has a further branch at RAM_GB < 10 which has become a leaf node and RAM_GB > 10 can still be separated into further nodes with variable weights. The continuation of this process can be seen in Figure 4.

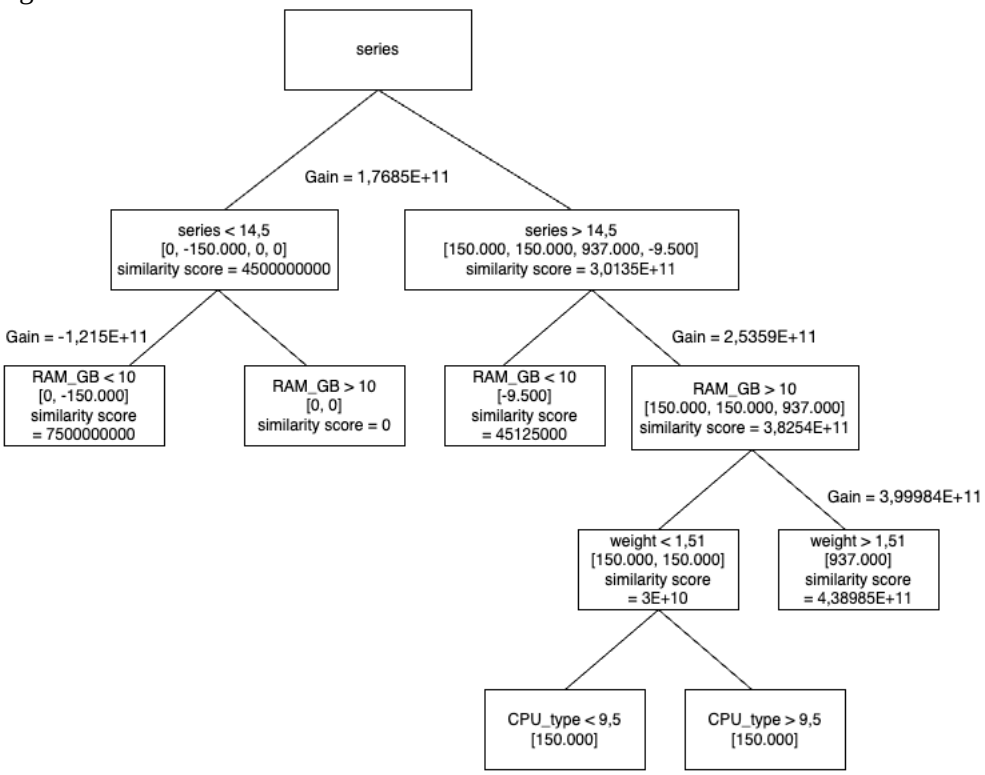

Figure 4. Splitting of continuation nodes with weight and CPU_type in tree 1

There is further branching on the weight node resulting in a CPU_type branch that ends up as a leaf node. Next, calculate the output of each leaf node using equation 2.6.

$$Output\ Value = \frac{sum\ of\ residuals}{Number\ of\ residuals + \lambda}$$

$$Output\ Value = \frac{0 + (-150.000)}{2 + 1} = -75000$$

$$Output\ Value = \frac{0 + 0}{2 + 1} = 0$$

$$Output\ Value = \frac{-9500}{1 + 1} = -4750$$

$$Output\ Value = \frac{937000}{1 + 1} = 468500$$

$$Output\ Value = \frac{150000}{1 + 1} = 75000$$

$$Output\ Value = \frac{150000}{1 + 1} = 75000$$

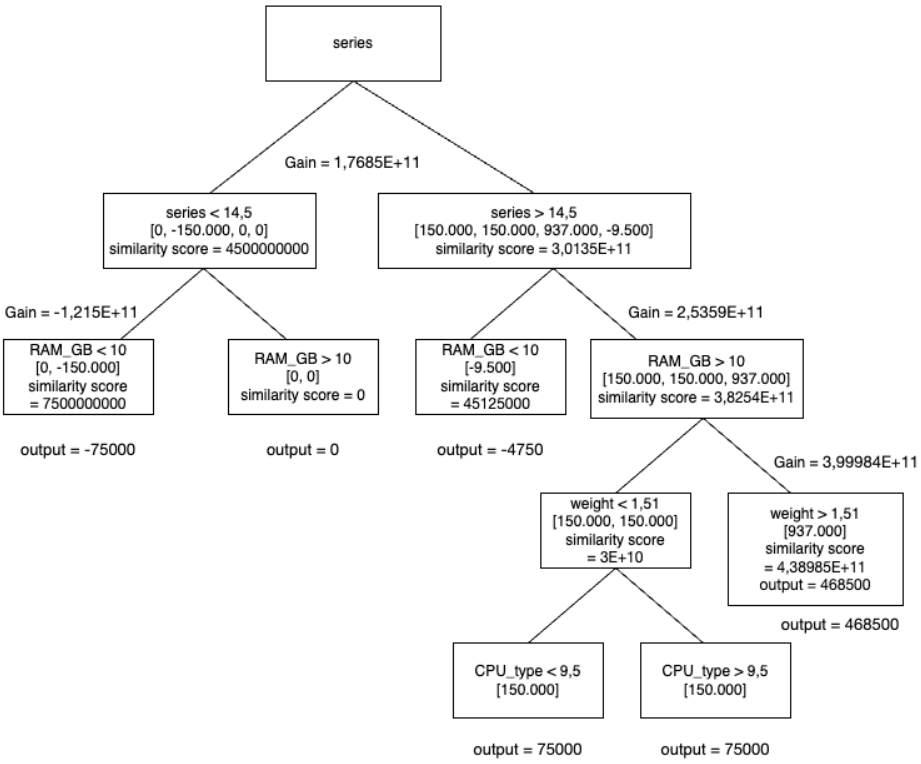

Figure 5. The output of each leaf node in tree 1

Figure 5 produces the output of each leaf node. The next step calculates the new predicted value using equation 2.7.

$$New\ prediction = \bar{y} + (learning\ rate \times output\ value)$$
$$New\ prediction = 11090000 + \big(0{,}01 \times (-75000)\big) = 11098250$$
$$New\ prediction = 5049000 + \big(0{,}01 \times (-75000)\big) = 5048250$$
$$New\ prediction = 17649000 + (0{,}01 \times 75000) = 17649750$$
$$New\ prediction = 12249000 + (0{,}10 \times 75000) = 12249750$$
$$New\ prediction = 30999000 + (0{,}01 \times 0) = 30999000$$
$$New\ prediction = 13562000 + (0{,}01 \times 468500) = 13566685$$
$$New\ prediction = 26499000 + (0{,}01 \times 0) = 26499000$$
$$New\ prediction = 6289000 + (0{,}01 \times (-4750)) = 6289452{,}5$$

After that, you can find the new residual by calculating the target value or initial price minus the new predicted value. The results at this stage can be seen in Table 12.

Tabel 12. Prediction Results and 2nd residuals

prediksi_baru	residual_2
11098250	750
5048250	-149.250
17649750	149.250
12249750	149.250
30999000	0
13566685	932.315
26499000	0
6289452,5	-9.453

After obtaining the 2rd residual there is still a large enough error. So build the tree again with new residuals by calculating the similarity score and gain value like the previous stage. There is the highest gain value in the series variable of 1,5785E+11 which will be used as the root node. The results at this stage can be seen in Table 13.

Tabel 13. Prediction Results and 3th residuals

Prediksi_baru	residual_3
11097916,25	1.084
4972916,25	-73.916
17725087,5	73.913
12325087,5	73.913
30999000	0
14037293,25	461.707
26499000	0
6284681,125	-4.681

Furthermore, it produces leaf nodes that can calculate the output of each node such as weight > 1.51 with an output of 230853.5 and CPU_type < 9.5 with an output of 36956.5 which can be seen in Figure 4.10.

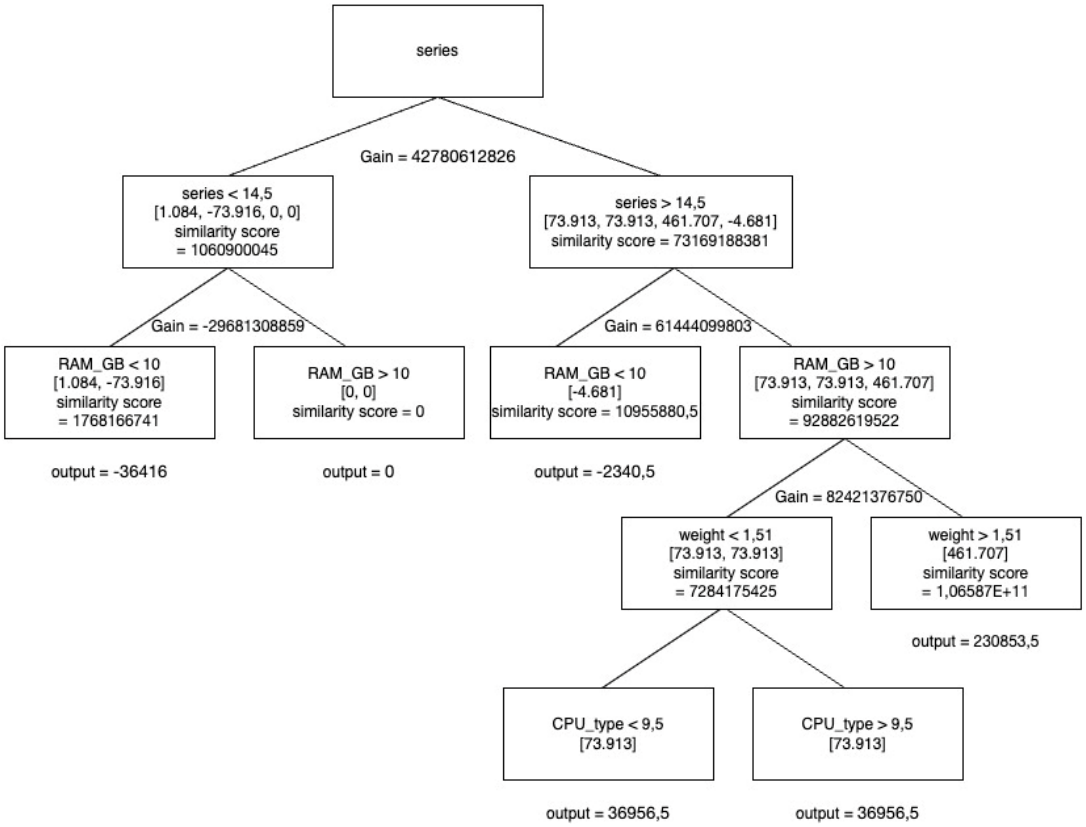

Figure 6. The output of each leaf node in tree 3

Next, the new prediction results are displayed in Tree 3 and there is a new residual. The residual value has decreased from the previous 73,913 to 36,587 with a price prediction from 17725087.5 to 17762413.32 which shows an increase in price prediction close to the original price. The results at this stage can be seen in Table 14.

Tabel 14. Prediction Results and 4th residual

Prediksi_baru	residual_4
11098093,97	906
4935593,965	-36.594
17762413,32	36.587
12362413,32	36.587
30999000	0
14270455,16	228.545
26499000	0
6282317,158	-2.317

Furthermore, it produces leaf nodes that can calculate the output of each node such as weight > 1.51 with an output of 114272.5 and CPU_type < 9.5 with an output of 18293.5 which can be seen in Figure 4.11.

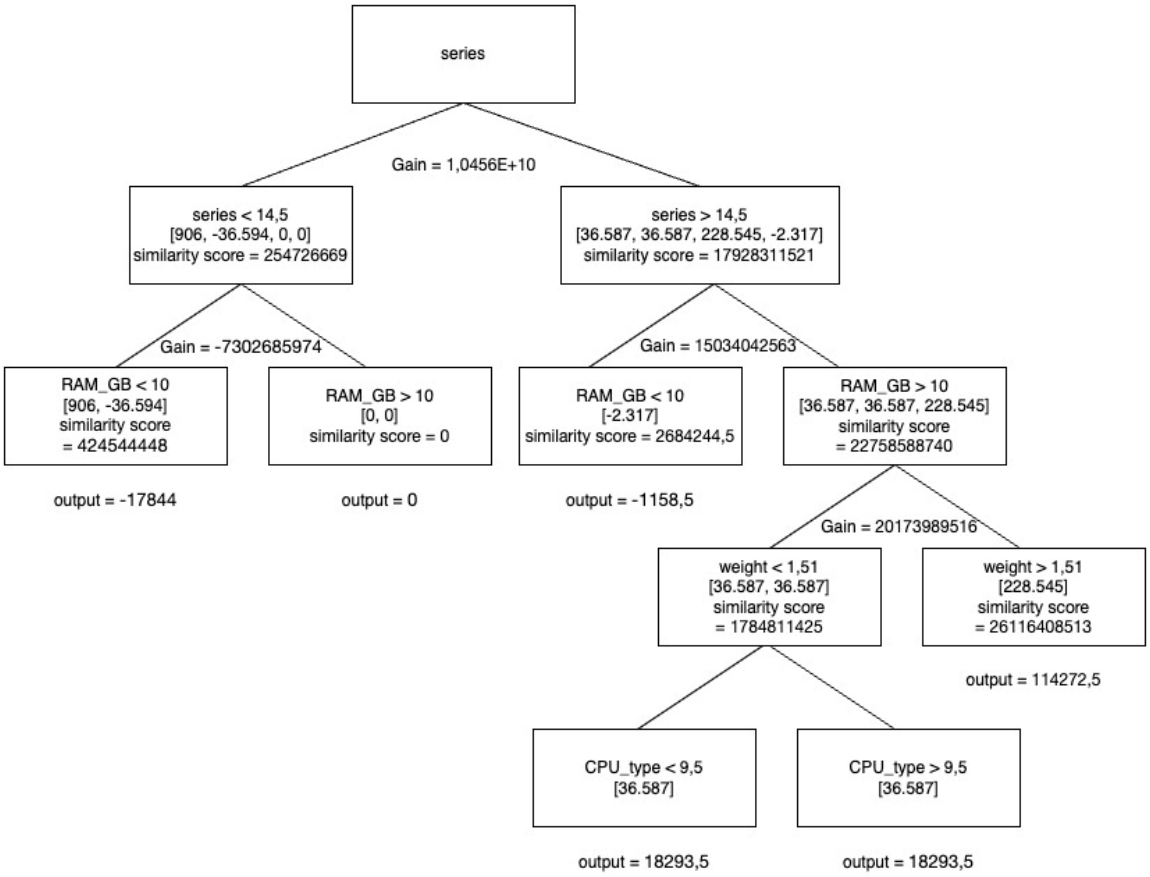

Figure 7. The output of each leaf node in tree 4

The calculation of the 4th tree output can determine the results of new predictions and new residuals which are the last experiment in this study. The residual value decreased from the previous 36,587 to 18,295 with a price prediction from 17762413.32 to 17780704.81 which shows an increase in price prediction close to the original price. The results at this stage can be seen in Table 4.15.

Tabel 15. Prediction Results and 5th residuals

Prediksi_baru	residual_5
11098550,62	449
4917300,623	-18.301
17780704,81	18.295
12380704,81	18.295
30999000	0
14384716,04	114.284
26499000	0
6281158,696	-1.159

Experimental results from residual 1 to residual 5 produce a pretty good error reduction with price predictions that are close to the original price. In one of the residuals of 150,000 decreased to 18,295 with the target value or original price of 17,799,000 predicted to be 17780704.81. This shows that there is still room to reduce the error by doing further node experiments calculating similarity and gain with new residuals. But in

this study only calculates up to the 5th residual. The next step is to test the tree with sample data which can be seen in table 16.

Tabel 16. Sample data on decision tree trials

brand	series	CPU_type	RAM_GB	storage_GB	screen_size	weight
1	28	10	12	512	14	1,50

The results of the sample data trial on the decision tree can be seen in Figure 8.

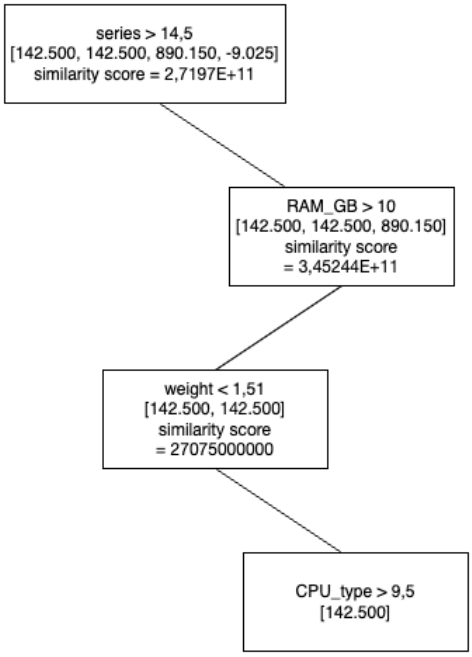

Figure 8. Sample data prediction results

After testing with the sample data in Table 4.14 in one decision tree, the price of the laptop will be predicted at 12380704.81 with the original price of 12,399,000.

3. RESULTS AND DISCUSSION

Based on the data source in this research comes from the bhinneka.com and pemmz.com websites which are collected using the scraping method containing data on laptop specifications and prices. The data on Bhinneka has 560 rows and 8 columns, while the data on Pemmz has 319 rows and 8 columns. The attributes used are brand, series, CPU_type, RAM_GB, storage_GB, screen_size, weight, and price. The data has been preprocessed to handle missing values in the weight attribute of 71 in Bhinneka and 1 missing value in Pemmz, so imputation is done with the median.

In duplicate data, there are 31 data in Bhinneka and 19 data in Pemmz, the data is deleted. The label encoding stage on categorical attributes such as brand, series, and CPU_type must be converted to numerical values to be processed in machine learning modeling. In handling outlier data, there are two experiments, namely dropping outliers and imputation with mode. If dropping outlier data is done, there are only 88 data left which will cause the dataset to be small and replace outlier data using mode and the results show that it can distort the original data which does not reflect the true variation, so it was decided not to dropping outlier data, not replacing outliers with mode, and retaining the data as before for further analysis.

In the data splitting stage, it is done by dividing based on folds using cross-validation of 5 to 10 folds. There are several parameters used to get the best model performance score and parameter tuning using GridSearchCV which can be seen in the table 17.

Tabel 17. Hyper parameter

Hyper Parameter	Nilai yang digunakan
max_depth	3, 4, 5
min_child_weight	0, 1, 2
gamma	0, 0.1, 0.2
subsample	0.7, 0.8, 0.9
colsample_bytree	0.7, 0.8, 0.9
learning_rate	0.01, 0.05, 0.1
n_estimators	100, 200, 300

Based on the best parameter results from parameter tunning can be seen in the table 3 and table 4 obtained a cross-validation of 8 at Bhinneka and a cross-validation of 6 at Pemmz while still using outlier data.

a. Results of parameter tuning with outlier data on Bhinneka

Tabel 18. Results of parameter tuning with outlier data on Bhinneka

Jumlah folds	max_depth	min_child_weight	gamma	subsample	colsample_bytree	learning_rate	n_estimators
5	5	0	0	0.8	0.7	0.05	300
6	4	2	0	0.7	0.9	0.1	200
7	4	2	0	0.8	0.8	0.1	200
8	4	2	0	0.8	0.8	0.1	200
9	4	0	0	0.8	0.8	0.1	200
10	3	0	0	0.7	0.9	0.1	300

b. Results of parameter tuning with outlier data on Pemmz

Tabel 19. Results of parameter tuning with outlier data on Pemmz

Jumlah folds	max_depth	min_child_weight	gamma	subsample	colsample_bytree	learning_rate	n_estimators
5	4	0	0	0.9	0.8	0.05	300
6	4	0	0	0.7	0.8	0.1	300
7	4	0	0	0.9	0.8	0.1	200
8	3	0	0	0.9	0.8	0.1	300
9	3	0	0	0.9	0.9	0.1	300
10	3	2	0	0.9	0.9	0.1	300

After tunning the parameters and getting the best parameters on Bhinneka and Pemmz. Furthermore, testing is carried out to get a score of model performance which can be seen in table 5 and table 6 which is a selection of the best accuracy scores using outlier data.

a. Accuracy results with outlier data on Bhinneka

Tabel 20. Accuracy results with outlier data on Bhinneka

No.	*Cross Validation*	*RMSE*	*MAPE*	*R2 Score*
1	5	1424572.56	0.06	0.97
2	6	1777888.45	0.08	0.96
3	7	1794793.18	0.08	0.96
4	8	1250363.29	0.05	0.98
5	9	1545737.71	0.07	0.97
6	10	1763054.31	0.08	0.96

b. Accuracy results with outlier data on Pemmz

Tabel 21. Accuracy results with outlier data on Pemmz

No.	*Cross Validation*	*RMSE*	*MAPE*	*R2 Score*
1	5	1259569.91	0.06	0.98
2	6	1073090.92	0.04	0.98
3	7	1163005.41	0.05	0.98
4	8	1348570.45	0.06	0.98
5	9	1243366.75	0.06	0.98
6	10	1277210.42	0.06	0.98

Based on the results of the trials conducted in table 5 and table 6, researchers obtained the following results:

a. When using outlier data, the model produces a higher accuracy score than the score with outlier data imputation using mode. This is because when outlier data is imputed with the mode, it can cause distortion of the original data that does not reflect the true variation.
b. The model that has the highest accuracy score is in table 5 with a cross-validation of 8 resulting in an RMSE of 1250363.29, a MAPE score of 0.05, and an R2 score of 0.98. Table 6 produces an RMSE of 1073090.92, a MAPE score of 0.04, and an R2 Score of 0.98 with a cross-validation of 6.
c. Based on the test results above, the researchers decided to use the model described in point b, which is the model generated from data with outliers.

4. CONCLUSION

The application in the case of laptop price prediction using extreme gradient boosting algorithm requires a relevant and recent dataset. Furthermore, the process in the algorithm determines the initial prediction value used to find the residual value. Based on the residual value, then calculate similarity and gain through data separation on each variable which aims to select the root node and the next node. After generating a decision tree and obtaining new outputs and residuals, then repeat the previous stages of building a tree that calculates similarity and gain values until it finds a low residual value.

Based on the results of the research that has been done, the accuracy score generated on Bhinneka store laptop data using the Extreme Gradient Boosting algorithm on outlier data with cross-validation of 8 gets an R2 Score value of 0.98, and outlier data imputed with mode produces an R2 Score value of 0.93 on cross-validation of 7. In addition, on Pemmz store laptop data with outlier data with cross-validation of 6 gets an R2 Score value of 0.98, and outlier data imputed with mode produces an R2 Score value of 0.94 with cross-validation of 5.

REFERENCES

[1]. N. Safariatun, “Penerapan Algoritma MOORA Dalam Pembelian Laptop Application of the MOORA Algorithm in Purchasing Laptops,” 2023.
[2]. A. A. Rifa’i, M. Fatchan, and N. T. Kurniadi, “Penerapan Algoritma K-Medoids Dalam Klasterisasi Penjualan Laptop,” Prosiding SAINTEK: Sains dan Teknologi , vol. 1, no. 1, 2022, [Online]. Available: https://www.kaggle.com/muhammetvarl/laptop-price
[3]. M. R. Ona Sain, Y. Setyawan, and R. D. Bekti, “Analisis Positioning Merk Laptop dengan Menggunakan Metode MDS Nonmetrik dan CA,” Jurnal Matematika, vol. 12, no. 2, p. 89, May 2023, doi: 10.24843/jmat.2022.v12.i02.p152.
[4]. H. Saiful Madjid, L. Nurul Istanti, M. Pemasaran, and U. Negeri Malang, “Pengembangan Website E-Commerce Sebagai Pendukung Strategi Digital Marketing Pada UKM Amazon Laptop,” 2023. [Online]. Available: https://vinicho.id/index.php/vidheas

[5]. W. K. Majid and I. Dzikria, “Perbandingan Penerapan Regresi Linear Berganda Dan Holt-Winter Exponential Smoothing Pada Prediksi Harga Emas Perhiasan,” 2023.

[6]. M. Zahedi, D. Jamal, and A. Das, “Human Centric Computing Applications for Laptop Price Prediction,” American Journal of Advanced Computing, vol. 2, no. 1, pp. 62–68, Nov. 2023, doi: 10.15864/ajac.21021.

[7]. J. M. A. S. Dachi and P. Sitompul, “Analisis Perbandingan Algoritma XGBoost dan Algoritma Random Forest Ensemble Learning pada Klasifikasi Keputusan Kredit,” Jurnal Riset Rumpun Matematika dan Ilmu Pengetahuan Alam (JURRIMIPA), vol. 2, no. 2, pp. 87–103, Jul. 2023, doi: https://doi.org/ 10.55606/jurrimipa.v2i2.1336.

[8]. A. T. Nurani, A. Setiawan, and B. Susanto, “Perbandingan Kinerja Regresi Decision Tree dan Regresi Linear Berganda untuk Prediksi BMI pada Dataset Asthma,” Jurnal Sains dan Edukasi Sains, vol. 6, no. 1, pp. 34–43, May 2023, doi: 10.24246/juses.v6i1p34-43.

[9]. S. F. N. Islam, A. Sholahuddin, and A. S. Abdullah, “Extreme gradient boosting (XGBoost) method in making forecasting application and analysis of USD exchange rates against rupiah,” in Journal of Physics: Conference Series, IOP Publishing Ltd, Jan. 2021. doi: 10.1088/1742-6596/1722/1/012016.