West Science Information System and Technology
Vol. 2, No. 01, April 2024, pp. 182~187

Housing Value Predicted Modeling using Random Forest Regression: Case study California Housing Dataset

Firman Matiinu Sigit[1], Haniel Rangga Pramuditya Putra[2]
[1,2]Tulungagung University

Article Info

Article history:

Received April, 2024
Revised April, 2024
Accepted April, 2024

Keywords:

Decision Tree Regression
Housing
Linear Regression
Random Forrest Regression.

ABSTRACT

Housing price comes from many factors which are location, population, style of house, age of house, and people income. Many real estate developer companies use this data to predict the price of houses and give the amount of investment for potential housing prices. In this study, we try to help the developer companies to predict the price of house based on the dataset. We try to build machine learning that can predict for housing price. There are three machine learning models that are used for this study, namely Linear Regression Modeling, Decision Three Regression Modeling, and Random Forest Regression Modeling. Each of those machine learning models is trained using California Housing Dataset (1990) which is split into training set and testing set that training set contains 16512 instances and testing set contains 4128 instances. The training dataset is trained into each of the machine learning models (Linear Regression, Decision Tree Regression, and Random Forrest Regression) after finishing the training followed by evaluating the prediction error using K-Folds Cross Validation and shown by using Root Mean Square Error (RMSE). In this study, Random Forest Regression gives a better performance than two others (Linear Regression and Decision Tree Regression models) with error RMSE =49642.12.

Corresponding Author:

Name: Firman Matiinu Sigit
Institution: Tulungagung University
Email: firman.matiinu@gmail.com

1. INTRODUCTION

The property sector which includes areas, and housing is now entering a phase of business that is growing so rapidly, both in terms of function/infrastructure and price. The price of a property will increase over time, one of the main causes is because other prices have also increased. This is an inflationary phenomenon that is difficult to avoid.

Property prices in certain areas will be different from other areas and vice versa, for example property prices that have close access to public facilities will be different from those far from access to public facilities. We just take a simple example, namely a residential area close to toll road access will have a relatively higher price than a residential area far from toll road access, not to mention areas close to public facilities such as hospitals, shopping centers, entertainment venues and others, which certainly affect prices that are relatively higher than property prices far from these public facilities. Of course, the value of the property itself must

also be seen, such as the size of the building, the number of rooms, building materials, the shape and size of the land and buildings, as well as the model and age of the building [1].

The property area development companies certainly examine all these factors starting from the location of the area to be developed, the appropriate form of building in the area, the facilities to be built in the area, access to public facilities, and the general price of property in the area, so that the company is able to calculate or predict the most appropriate property price in the area is expected to target the appropriate consumers (based on income).

In this research, a 1990 dataset from the StatLib repository [2] is used, this dataset is based on a census in the California region about the price / value of a building in the area. In this dataset there are features that can affect the price of a property or building in the area ranging from location, age of the building, population, number of rooms, number of rooms, number of families, average income, and close / far to the sea, all of which can affect the value / price of property in the California area based on this dataset.

2. METHODS

In this research, a house price prediction model will be designed in the California region based on a dataset in 1990. This modeling uses linear regression as an objective function to predict the price of a building based on existing features (location, age of the building, population, total family, income, and distance of the building location to the beach) to be able to predict the price of the house. After the modeling is complete, the performance of the prediction model will be measured using the RMSE error measure so that it can be concluded whether the modeling is underfitting or overfitting.

The following is a block diagram of this research,

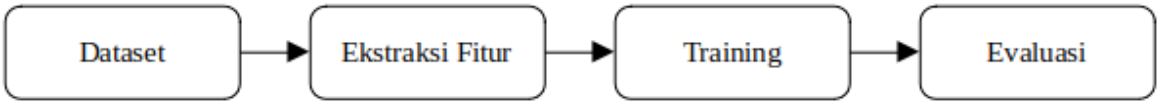

Figure 1: Block diagram of the study

2.1 Dataset

This study used a dataset consisting of 20640 rows of data consisting of columns in the form of features namely location, building age, population, total families, income, distance of building location to the sea, with the target feature being Median Housing Value. Each column consists of 20640 rows, only total_bedrooms reach 20433 and not 20640.

2.2 Feature Extraction

The feature extraction section consists of steps taken with the aim of deepening the information contained in the dataset.

2.2.1 Separation of training set and testing set

This dataset is separated by 80 percent for the training set and 20 percent for the testing set, so it consists of 16512 data for training and 4128 data for testing. The testing dataset is temporarily left alone because it is used as the final evaluation material of the prediction modeling.

2.2.2 Find the correlation/relationship between the target column (Median Housing Value) and other feature columns

At this stage, namely looking for correlations between target features and existing category features using the correlation matrix function and obtained information that median_income has a close correlation with the median_house_value feature as a target (0.68), this correlation informs that the higher the value of median_income, the higher the value of median_house_value. The higher the correlation value is close to positive 1, the closer the correlation relationship or positive linear, on the other hand, the further the correlation number is from positive 1 and

even closer to -1, the correlation relationship will be more opposite / negative linear.

2.2.3 Stratification on the dataset

Since median_income has a close correlation with median_house_value as the target, we will pay more attention to this median_income feature. The distribution of the number of median_income values can be seen through a histogram consisting of 5 bins/histogram bars as shown below,

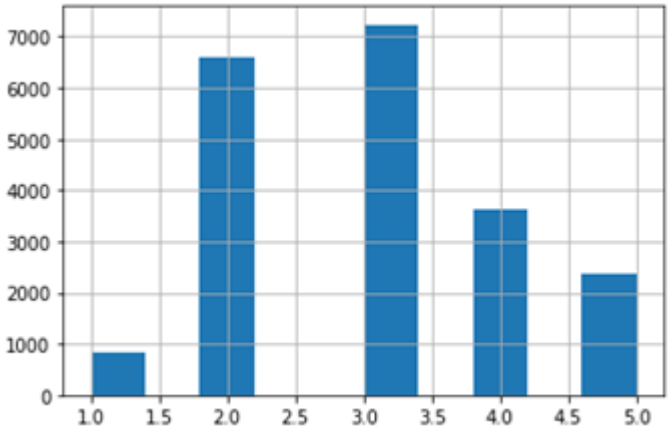

Figure 2. Stratification Graph on Median_Income

It can be seen from the histogram above that the largest distribution of median_income is in 2.5 to 3.5-bins 3 with more than 7000 data, followed by bins 2, bins 4, bin 5, and finally bin 1. This information can be used as a reference for stratifying the training set and the next test set.

It can also be seen that the composition of the distribution in the training set after stratification refers to the median_income stratification, as below,

3 0.350594
2 0.318859
4 0.176296
5 0.114402
1 0.039850

It can also be seen that the composition of the distribution in the test set after stratification based on median_income is as below,

3 0.350533
2 0.318798
4 0.176357
5 0.114583
1 0.039729

So that at this stage we have obtained the composition of the distribution of category data in the training set and test set which has a relatively close correlation with median_income.

2.2.4 Adding an Imputer

As mentioned above, the total_bedroom category data consists of 20433 data which does not match the data in the other categories which is 20640. Therefore, to add the missing data in the test set section, a simple imputer with a median strategy is needed which will be added to the test set section during modeling evaluation later.

2.2.5 Handles features that are text

In the dataset above there are features that are not in the form of numbers but in the form of text, such as in the Ocean Proximity category column. The Ocean Proximity feature consists of 5 types, namely Ocean, Inland, Island, Near bay, and Near Ocean. So, to handle data in the form of text can be done by converting the text data into data in the form of sequential numbers (ordinal), but in this case the Ocean Proximity category each data is not sequential between each other, so it can be categorized independently, therefore it can use binary numbers to categorize it by using One hot encoding to produce a sparse matrix.

2.2.6 Feature scaling

In this preprocessing stage, the common feature scaling has two types of actions, namely min-max scaling and standardization. At the min-max scaling stage all categories/features are standardized to a value between 0 and 1, this is commonly

referred to as normalization. In the standardization action the feature/category data is reduced by the Mean so that the Mean data is always 0, then divided by the standard deviation. The data resulting from feature scaling is in the form of array data.

2.2.7 Pipeline

At this stage of the pipeline is the invocation of the previous stages in sequence starting from the dataset stratification stage, separating the numeric data category column with data in the form of text (Ocean_Proximity), Imputer, and Feature Scaling. At the Feature scaling stage, the training set data obtained is in the form of matrix data with a size of (16512 x 8) where 8 columns here are the number of category columns of existing features. After reaching the feature scaling stage here using standardized scaling, then combining the matrix array (16512 x 8) with the sparse matrix of the Ocean_Proximity feature produces a matrix with a size of (16512 x 13) ColumnTransformer library.

This last matrix with a size of (16512 x 13) is variableized with X_prepared which will be fed into the selected Machine Learning algorithm.

2.3 Training

At this stage, the simplest machine learning algorithm is selected, namely linear regression. After the data matrix with size (16512 x 13) as training input and the house_mean_value category as label or target data, the next step is to choose the appropriate Machine Learning Algorithm. As a start, the simplest Machine Learning algorithm is chosen, namely Linear Regression. In short, this Linear Regression algorithm estimates the most effective relationship between input and target by using a linear approach. After the Linear Regression training, the next step is to calculate the performance of the modeling or evaluate it.

2.4 Evaluation

Furthermore, the performance of Linear Regression Machine Learning Modeling will be measured using the RMSE (Root Mean Square Error) measure. An example of 5 arbitrary data is taken, namely in the 'X_prepared' matrix section and 5 data in the one column Matrix, namely 'Labels'. Furthermore, 5 arbitrary data from the X_prepared matrix will be predicted output using Linear Regression Modeling that has been formed previously (through training results) or predicted output using Pretrained Linear Regression and produce consecutive outputs are [211574.39523833 321345.10513719 210947.519838 61921.01197837 192362.32961119] compared to the target value/label which is successively [286600.0, 340600.0, 196900.0, 46300.0, 254500.0] there is a deviation between the output results of our Linear Regression modeling prediction and the actual target output.

Furthermore, alternative algorithms will be selected again, namely Decision Tree Regression and Random Forrest Regression and then evaluated / measured its performance again.

3. RESULTS AND DISCUSSION

After we get our Linear Regression Pretrained Modeling, the next step is to evaluate the performance of our Machine Learning modeling using all data from the X_prepared matrix, the deviation error is measured against all Label data after the X_prepared matrix is predicted using the existing Linear Regression Pretrained, and the RMSE error is 69050.98178244587. The size of the error indicates that the Machine Learning algorithm modeling does not have the ability to predict well or can be said to be underfitting [3].

The underfitting measure indicates that our Machine Learning modeling is not able to make good predictions of the target label or the Machine Learning Algorithm is not able to find patterns from the existing dataset. One way to overcome underfitting is to replace the Machine Learning Algorithm with another Machine Learning Algorithm that is expected to provide better prediction performance. In this case, we will try to use Decision Tree Regression and Random Forest Regression.

Furthermore, experiments were carried out using Decision Tree Regression after training using Decision Tree Regression and evaluating the error deviation using RMSE between the prediction and the target label, the RMSE was found to be 0, why can this happen? Linear Regression and Decision Tree Regression have different algorithms, if Linear Regression the basis of prediction is to use a linear line function so that there is a prediction deviation while Decision Tree Regression the prediction does not use a linear line function so that it is able to predict where the target label and the prediction value are the same / overlap [4]. So, the question arises, how to overcome the 0-value RMSE because Mcahine Learning modeling is indicated to be very overfitting. One way is to use K-Fold Cross Validation.

K-Fold Cross Validation

K-Folds Cross Validation is commonly used to measure the performance of a Machine Learning Algorithm that will be used with existing datasets. The treatment is that the dataset is divided into training sets and test sets, then in the training set the data is divided into k-fold subdatasets, following the description,

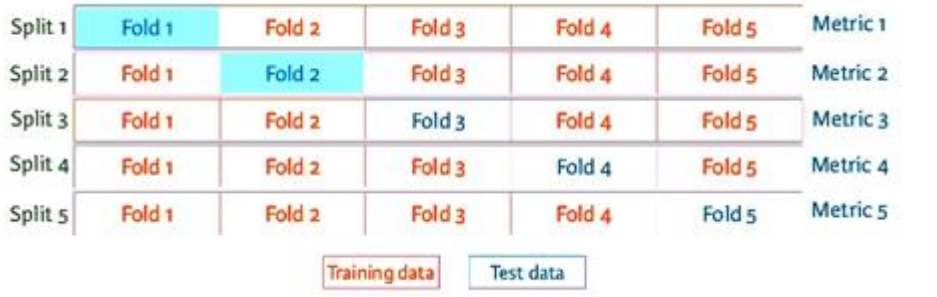

Figure 3. K-Folds Cross Validation

In the K-Folds Cross Validation, there is training data and test data, each used as training and test material as much as k-Fold data, resulting in modeling as much as k-Fold data as well. This treatment is used to generalize modeling with datasets, K-Folds Cross Validation is very effectively used to detect overfitting of a model [5].

In Decision Tree Regression after evaluation using K-Folds Cross Validation with K = 10, it produces consecutive RMSE values of [65971.50042857 66569.54062412 72599.35396605 70219.17065156 67671.19215212 76360.83208189 65696.83457799 69693.55929053 71098.4393938 69163.50497912] with the Mean value being 69504.39281457648. While the Linear Regression produces consecutive RMSE values of [67450.42057782 67329.50264436 68361.84864912 74639.88837894 68314.56738182 71628.61410355 65361.14176205 68571.62738037 72476.18028894 68098.06828865] with the Mean being 69223.18594556303. If it is further noted that Linear Regression provides better performance than Decision Tree Regression because the RMSE value is smaller, it can also be interpreted that Decision Tree modeling is more overfitting than Linear Regression modeling [3].

Furthermore, Machine Learning modeling selection is carried out using Random Forest Regression. After training using the training dataset with the target label, the modeling is obtained, then the evaluation is carried out using the training dataset with the target label and there is a deviation of the RMSE is 18372.5927988511, this deviation value is relatively smaller than the RMSE value of the previous regression model (linear regression and Decison Tree Regression). Then this modeling is evaluated using K-Folds Cross Validation with K=10 resulting in consecutive RMSE scores of [47883.88961127 45850.89967749 49243.06863356 49990.34167616 49312.73630331 53184.51869311 48969.4086803 50579.91661717 51720.74593177 49685.67774438] with a Mean value of 49642.12035685319. From the data it can be

seen that the RMSE score is smaller in training validation than the RMSE value in validation using K Fold Cross Validation, this indicates that this modeling (Random Forest Regression) is still overfitting on training data. One way to overcome overfitting is to increase the dataset again.

4. CONCLUSIONS

Of the three Regression modeling used in this experiment, it is concluded that Random Forest Regression provides the best performance compared to the other two Regression modeling, namely Linear Regression and Decision Tree Regression, with an RMSE value of 18372.5927988511 and evaluation using Cross Validation gives an average Mean deviation error RMSE of 49642.12035685319.

REFERENCES

[1] Five factors that affect the selling price of property. (2019). *Valuation.* Retrieved July 28, 2023, from https://penilaian.id/2019/03/25/5-faktor-yang-mempengaruhi-harga-jual-properti/

[2] California Housing Dataset. (1990). *California Housing Price*. Retrieved July 25, 2023, from https://www.kaggle.com/datasets/harrywang/housing

[3] Aurelien Geron. (2019). *Hands-on Machine Learning with Scikit-Learn and TensorFlow*. O'Reilly.

[4] Decision Tree (2023). *Decision Tree Regressor - A Visual Guide with Scikit Learn.* Retrieved July 25, 2023, from https://towardsdatascience.com/decision-tree-regressor-a-visual-guide-with-scikit-learn-2aa9e01f5d7f

[5] Cross Validation (2019). *Cross Validation - Why & How*. Retrieved July 25, 2023, from https://towardsdatascience.com/cross-validation-430d9a5fee22