House Price Prediction Using Random Forest Regression Algorithm

Md. Jusef
Department of Computer Science and Engineering
East West University

***Abstract*— This study uses three machine learning algorithms including, Linear Regression, Decision Tree and Random Forest Regression algorithms for evaluating house prices. It applies that method to evaluate Boston housing data and then compares the results of these algorithm. In terms of predictive power Random Forest Regression algorithm achieved best result compared with linear and Decision tree algorithm. This study also evaluates our algorithm through mean, mean square root and standard deviation value. However, my study found that Random Forest Regression algorithm provide good result. Machine learning offers a promising, property evaluation and evaluate research specially to house price prediction.**

Keywords*— *Machine Learning Algorithm, House Price Prediction, Random Forest Regression

I. Introduction

Investment is the business policy that most people interested to this system. Machine learning is the best process for house price prediction. Having a housing price prediction model can be a very important tool for both the seller and the buyer as it can aid them in making well informed decision. For sellers, it may help them to determine the average price at which they should put their house for sale while for buyers, it may help them find out the right average price to purchase the house.

The problem falls under the category of supervised learning algorithms. There are so many processes to predict house prices, Random Forest Regression algorithm is one of them.

II. DATASET

In this study, I used **Boston Housing** dataset. The dataset comprises 13 input features and 1 target feature. The Boston Data frame has 506 rows and 14 columns.

Attribute information are given below:

Features Description

1. CRIM per capita crime rate by town
2. ZN proportion of residential land zoned for lots over 25,000 sq.ft.
3. INDUS proportion of non-retail business acres per town
4. CHAS Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
5. NOX nitric oxides concentration (parts per 10 million)

6. RM average number of rooms per dwelling
7. AGE proportion of owner-occupied units built prior to 1940
8. DIS weighted distances to five Boston employment centres
9. RAD index of accessibility to radial highways
10. TAX full-value property-tax rate per $10,000

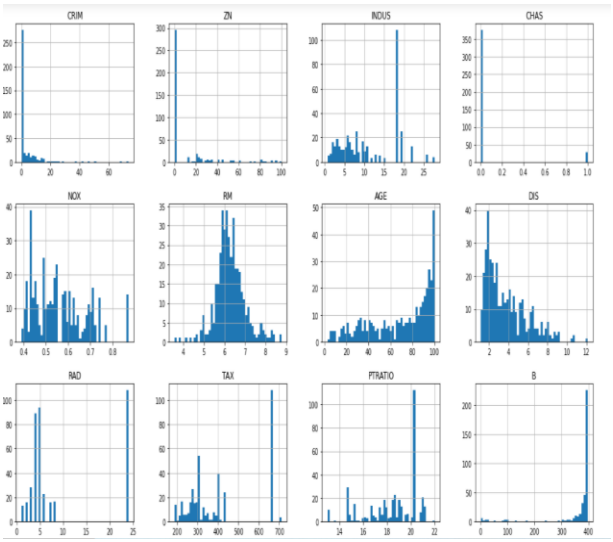
11. PTRATIO pupil-teacher ratio by town
12. B 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
13. LSTAT % lower status of the population
14. MEDV Median value of owner-occupied homes in $1000's

III. Methodology

I was using three algorithms for evaluate the model. For Figuring out best model I take some process like as linear regression, Decision tree and random forest algorithm.

Plotting Histogram:

This study I used plotting Histogram for data analyze easily. All features histogram are here. From "CHAS" features I see that counts of 1 is greater than 0. From "age" features I see that more than 60 building age is 100 years. So we can easily read dataset through histogram.



Train-Test Splitting:

For training this dataset, I used sklearn model.

```
from sklearn.model_selection import train_test_split
train_set, test_set  = train_test_split(data_field, test_size=0.2, random_s
print(f"Rows in train set: {len(train_set)}\nRows in test set: {len(test_se
```

```
Rows in train set: 404
Rows in test set: 102
```

Missing Attributes:

To take care of missing attributes we have three options

1. Get rid of the missing data points
2. Get rid of the whole attribute
3. Set the value to some value (0, mean or median)

Following this option, we can easily handle with missing data.

Picking the right evaluation metric

Picking the right evaluation metric will help us to evaluate whether our model's performance is good. Through root mean square error, mean and standard deviation I evaluate my model.

RMSE: It measured difference between actual and predicted value.

$$RMSE(y) = \sqrt{\sum_{i=1}^{n} (y' - y)^2}$$

Figure: Root Mean Square Error Formula

Where y': Predicted value, y: Actual value

Mean: The mean is the average or the most common value in a collection of numbers.

Standard Deviation: the measure of dispersion of a set of data from its mean.

Linear Regression: linear regression uses a traditional slope-intercept form, where *a* and *b* are the coefficients that we try to "learn" and produce the most accurate predictions. *X* represents our input data and *Y* is our prediction.

$$Y = bX + a$$

Applying Linear Regression algorithm, I get mean, root mean square error and standard deviation value which are:

Mean = 4.221894675406022

Standard Deviation = 0.7520304927151

Decision Tree : Decision tree algorithms is a supervised learning algorithms. The objective of Decision tree is to create a training model that can used to predict the value of the target variable by learning decision rule from training data.

Applying Decision Tree algorithm, I get mean, root mean square error and standard deviation value which are:

Mean: 4.189504502474483

Standard deviation: 0.848096620323

Random Forest Regression:

Random Forest Regression is also supervised learning algorithm. A Random forest operates by constructing several decision trees during training time and outputting the mean of the classes as the prediction of all the trees.

Applying Decision Tree algorithm, I get mean, root mean square error and standard deviation value which are:

Mean: 3.494650261111624

Standard deviation: 0.762041223886

RMSE : 2.9311088813808674

IV. Discussion and Result

After evaluating Linear regression, Decision tree and Random forest regression algorithms, I observe that Random forest regression give me best result both of them. So I used Random forest regression algorithm in my model. I separated my model as ABC.joblib file.

```
In [57]: #check prediction
         from joblib import dump, load
         import numpy as np
         model = load('ABC.joblib')
         features = np.array([[-5.43942006, 4.12628155, -1.6165
                -1.44443979304, -49.31238772,  7.61111401, -26.
                -0.97491834,  0.41164221, -66.86091034]])
         model.predict(features)

Out[57]: array([23.578])
```

I input all features value and my model predict me price is 23.578.

V. Conclusion

New analytical techniques of machine learning can be used in new globalization era. In this study our models are trained with Boston housing data utilizing model based linear regression, decision tree and random forest regression algorithm.

Given my dataset used in this report, my main conclusion is that Random Forest is able to generate comparably accurate price. My model has shown that machine learning algorithms like Linear Regression, Decision Tree and Random Forest are promising tools for researchers to use housing price prediction.

VI. References

1. https://towardsdatascience.com/predicting-housing-prices-using-a-scikit-learns-random-forest-model-e736b59d56c5
2. https://www.tandfonline.com/doi/full/10.1080/09599916.2020.1832558
3. https://www.youtube.com/watch?v=iIkJrwVUl1c&t=1303s
4. https://www.kaggle.com/ammar111/house-price-prediction-an-end-to-end-ml-project
5. https://towardsdatascience.com/predicting-housing-prices-using-a-scikit-learns-random-forest-model-e736b59d56c5