Boston Housing Data

Estimating median value of a house through Regression Model and Tree

Contents

Boston Housing Data

Background

Boston Housing is a data set that contains the details of houses in Boston along with several parameters. To be more precise, it has the data related to the median value of the houses in Boston area and values of other factors that may or may not affect the value of house.

Goal

The main aim of this report is to find a method with which we can estimate the Median Value of a house in Boston area given other parameters. The performance of the solution is a key factor since the accuracy of the prediction is desired.

Approach

In this report, we primarily concentrated on designing a Linear Regression model and Regression Tree for predicting the Median Value of the houses. We divided the data set into train and test data sets, built the model/tree on train data and validated it through the test data.

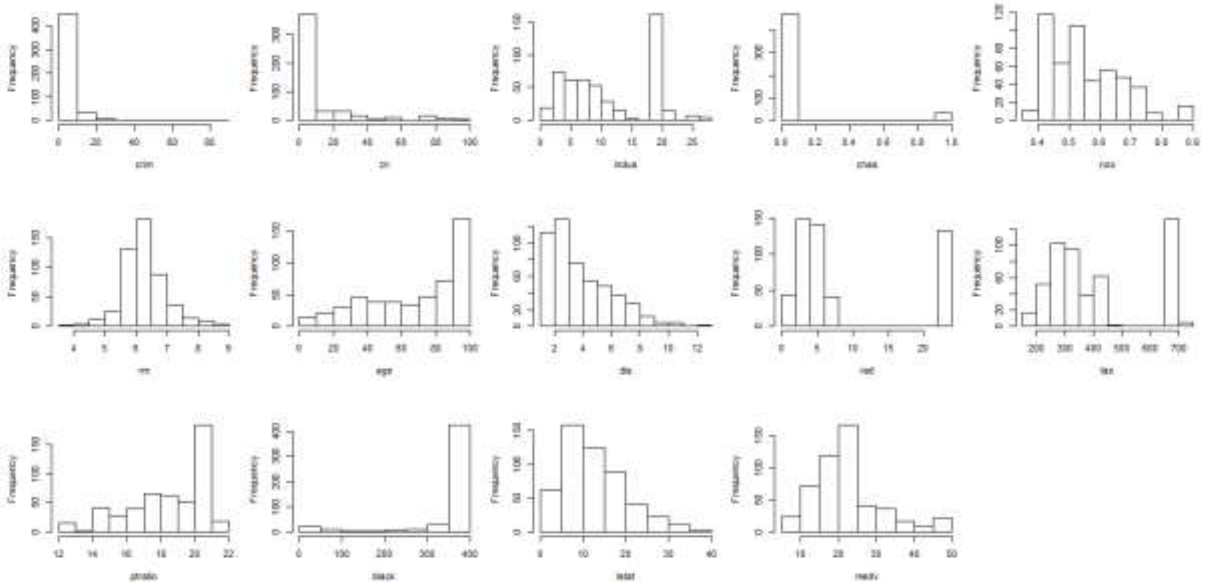
Major Findings

Through our analysis in the report, we found that the median value can be estimated either by using a regression model or by a tree with considerably good accuracy. But by comparing some key parameters, we identified that regression tree gives some better results. Hence we feel that using Regression Tree for predicting the median value of housing in Boston might yield good results.

The below report has detailed analysis and approach we followed for this purpose.

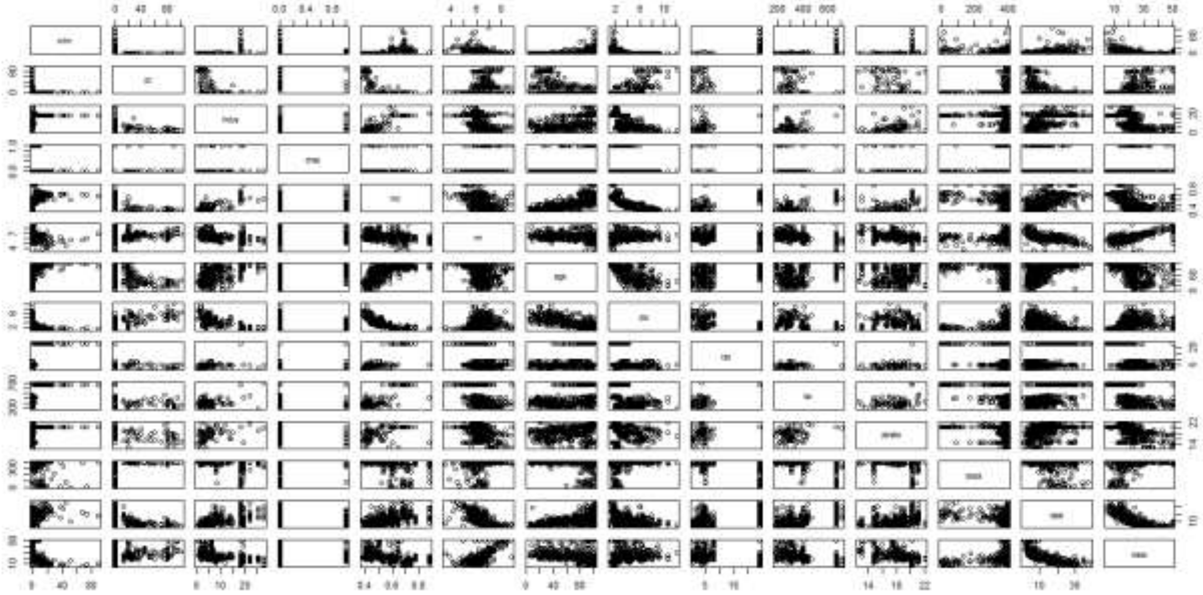
Boston housing Data- Exploratory Analysis

Boston data contains the data with respect to housing value of suburbs in Boston. Below is some exploratory data analysis of the Boston data.



Above are the histograms of all the variables that indicate the frequency or distribution of variables. Some variables are distributed over their respective range but some variables like zn, chas are concentrated at one value.

Below are the scatter plots between different variables of the Boston data set.



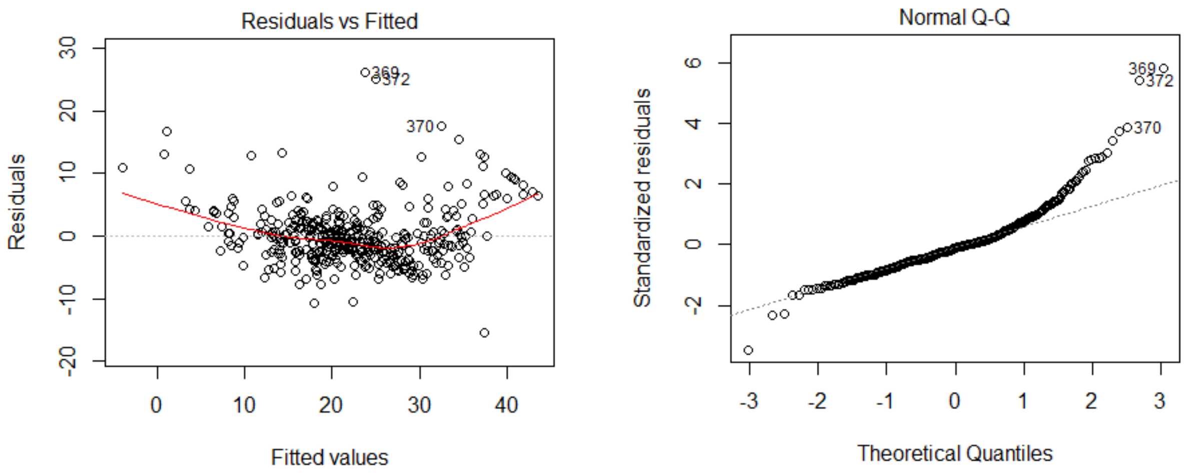
Basically the data is divided into 2 data sets training (which as 80% of original data) and test set(which has rest 20% of data). For finding the best linear model, let us construct the full model and other models using Forward, Backward and stepwise. From them we can pick one best model.

	AIC	BIC	Variables
Full Model	2411.11	2471.13	All variables
Forward	1259.89	2460.40	Crim, zn, chas, nox, rm, dis, rad, tax, ptratio, black, lstat
Backward	1259.89	2460.40	Crim, zn, chas, nox, rm, dis, rad, tax, ptratio, black, lstat
Stepwise	1259.89	2460.40	Crim, zn, chas, nox, rm, dis, rad, tax, ptratio, black, lstat

From above we can take those 11 variables for estimating the medv. Using these variables, the model that is generated has the following intercept and coefficients.

Intercept	lstat	Rm	ptratio	dis	nox	black	zn	crim	rad	tax	chas
43.30	-0.53	3.13	-1.00	-1.56	-19.16	0.01	0.05	-0.14	0.33	-0.01	1.67

The model statistics for this model are as follows



From above plots, we can infer that the Residual vs Fitted values are scattered around 0 line indicating that the model is fairly good. Q-Q plot is reasonably straight. So we can say that residuals are normally distributed. By this we can say that this model is fairly good and we can consider it for further analysis.

Out of Sample performance

To evaluate how the model performs on the future data, we need to do some out of sample testing for the selected model. From above, we have test data which can be used to test the model that is generated from train data.

Now let us calculate the Mean Square Error (MSE) and Mean Absolute Model(MAE) for our model. Also let us calculate these values to Full Model just to compare these parameters for both the models. Below are the details of the same-

	MSE	MAE
Final Model	21.80	3.34
Full Model	22.39	3.53

From above, it is clear that the Final Model’s MSE and MAE are slightly better that that of Full Model. So we can clearly say that the selected Final Model is slightly better at predicting the outcome when compared with the Full Model.

Cross Validation

In cross validating the Final Model, we check the performance of the Model in predicting the outcome variables. In above section, the model is generated on the train data and the MSE is calculated on the train data. The problem here is that-

- Train data and the Test data are randomly split from the original data.
- Because of this random split, both the data sets might not have proper conformity data.
- I.e. data such as outliers might be concentrated in only one data set.

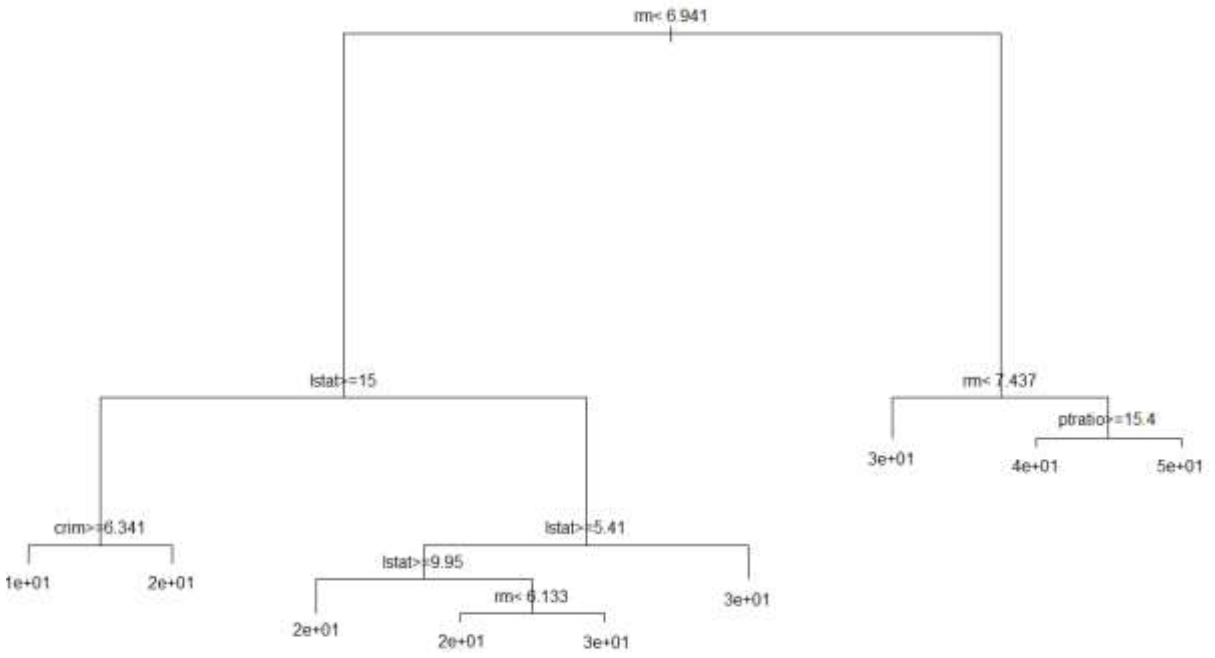
So to avoid the extent of these issues, we can calculate the MSE of our model through k-fold cross validation. In k-fold cross-validation, the original sample is randomly partitioned into k equal size subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k − 1 subsamples are used as training data. The cross-validation process is then repeated k times (the folds), with each of the k subsamples used exactly once as the validation data. The k results from the folds can then be averaged (or otherwise combined) to produce a single estimation[1].

The advantage of this method over repeated random sub-sampling is that all observations are used for both training and validation, and each observation is used for validation exactly once. 10-fold cross-validation is commonly used.

The calculated MSE using 10-fold cross validation is **23.30** but the MSE from out sample is **21.80.** We can say that the MSE resulted from cross validation is a bit more accurate that the MSE from out sample since the cross validations takes 10 sub samples and calculates MSE for each one. Where as in Out sample we use only 1 sample of test data.

Fitting a regression tree (CART)

The regression tree for the Boston data using the final model that is calculated above is as follows-

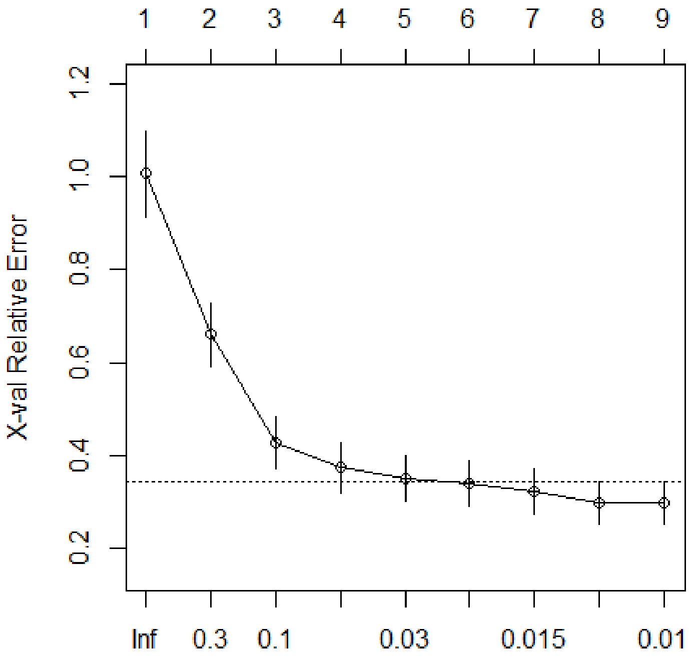


[1] http://en.wikipedia.org/wiki/Cross-validation_(statistics)

From above it is clear that the generated tree has 9 leaves and the order of variable importance is as follows-

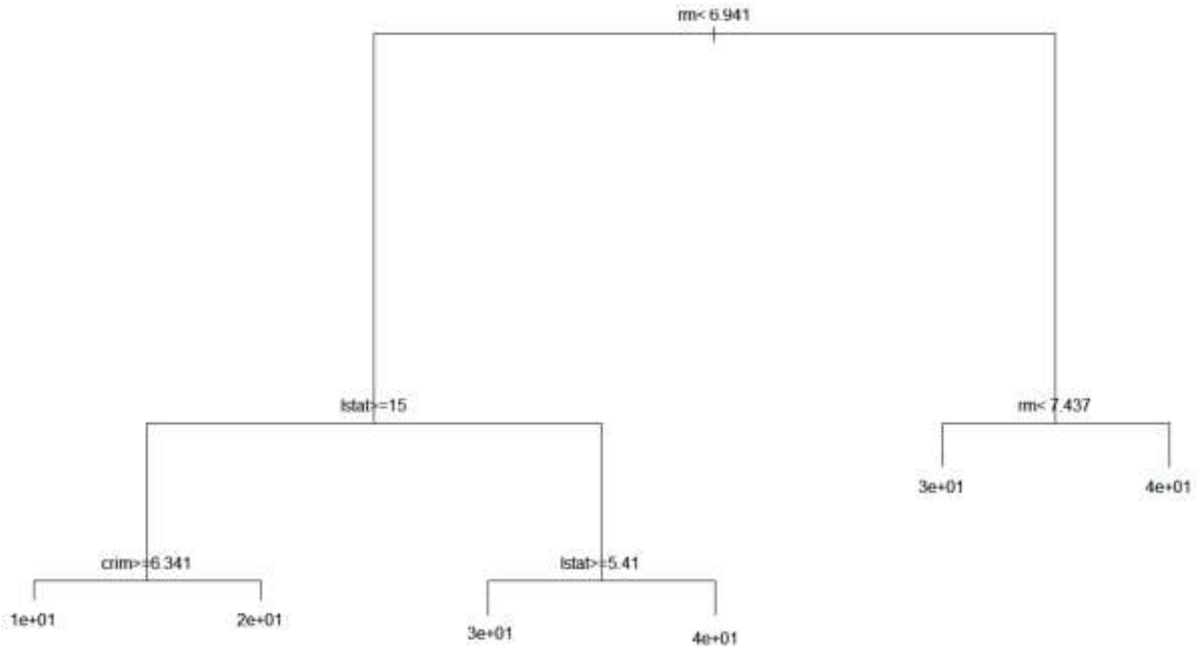
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
rm	lstat	crim	dis	nox	tax	rad	ptratio	zn	black	Chas
18847	13991	4210	4040	3678	3477	3099	2973	1381	274	235

In the above tree, we are including all the parameters of the model, this can lead to the tree becoming more complex. We can reduce the complexity of the tree by selecting the optimum Complexity Parameter (cp) and there by identifying the number of leaves of the tree.



From above plot, it is clear that at Cp of 0.03, the cross validation error is just below 0.4 and from then on it is getting reduced very low. So the optimum Complexity Parameter would be the point after 0.03 which is **0.0225.**

Now with this Complexity Parameter, let us prune the regression tree to get the optimum number of leaves-



Above is the Regression tree with the complexity parameter 0.0225 and 6 leaves. This can be considered as the optimized tree for the selected final model.

Now we can check the out sample performance of this regression tree by applying the test data and calculating the mean square error.

The mean square error (MSE) of this pruned regression tree is **17.55** on the out sample / test data.

Comparison of Regression Model and Regression Tree

On simple comparison of MSE's of generated Regression Tree and Regression Linear Model for the Boston data, it is clear that Regression Tree has lower MSE and hence better out of sample performance when compared with Linear Regression model.

	Out sample MSE
Regression Model	21.80
Regression Tree	17.55

Our analysis shows that the classification and regression trees method is better suited than the classical linear regression approach for explaining the median value of the houses in Boston area. The quality of the performance of regression tree out of the training data is good when compared with that of the Regression model. Hence we can say that CART approach is worth being further investigated.

Comparison of different samples

The entire analysis we did in the above chapters is based on single set of randomly sampled train and test data (Sample1). Now let us repeat the same analysis for different other randomly sampled train and test data from the Boston data.

Below is the table that show the comparison of different parameters that came out of the analysis-

	80/20 Sample1	80/20 Sample2	70/30 Sample1	70/30 Sample2
Full Model AIC	2411	2409	2141	2115
Full Model BIC	2471	2469	2199	2173
Final Model AIC	1259	1256	1131	1150
Final Model BIC	2460	2457	2188	2162
Final Model MSE	21.80	25.56	20.20	24.5
CV Final Model MSE	23.30	23.40	23.30	23.50
Full tree MSE	15.00	20.30	18.00	21.9
Complexity Parameter	0.0225	0.03	0.03	0.025
Pruned leaves	5	5	5	5
Pruned tree MSE	17.50	28.30	20.80	23.00

Observations from the above table-

- Since the test data is randomly generated every time, the out sample MSE is little bit different from each set.
- For cross validation, since we use the entire credit data, the CV MSE is almost the same in each case.
- When considering the MSE, the performance of regression tree is considerably very good when compared with that of the linear regression model.
- When AUC is considered, the regression model has little bit better values when compared with regression tree.
- As we can observe the full tree has lower MSE when compared with Pruned tree. So we either need to compromise either on complexity of tree or on MSE while selecting the regression tree.