

|  |
| --- |
| Boston Housing Data   * Linear Model and Regression Tree * Bagging, Random Forest and Boosting |
|  |
| May 5  Hardik Gupta |

# Boston Housing Data Executive Summary

# Executive Summary

# 1.1 Goal and Background

The Boston housing dataset is a dataset that contains median value of the house and 13 other variables including number of rooms, tax rate, etc. that could be related to the housing prices in Boston. The dataset has 506 observations and 14 variables, and we split the data into 70% (354 obs.) training data and 30% (152 obs.) testing data. We use the training data to create our linear and various tree models and the testing data to compare the different models. The aim of this project is to compare different model performance to estimate the median price value of owner-occupied houses in Boston.

The dataset is first analyzed and understood. We performed a regression analysis to build a linear regression model with all the 13 variables and then selecting the final best linear model. We then proceeded to fit the various tree models including regression tree, bagging, random forests and boosting trees. In the later part, we have compared all the tree models built so far to predict the median house price using in-sample and out-of-sample performance.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.2 Major Results  Based on the analysis, it was observed that the in-sample and out-of-sample performance was the best for Boosting model. Below is the summary of results:   |  |  |  |  | | --- | --- | --- | --- | | S.No. | Model | In-Sample MSE | Out-of-sample MSE | | 1 | Multiple Linear Regression | 17.90897 | 36.4963 | | 2 | Regression Tree (CART) | 15.00729 | 32.42087 | | 3 | Bagging Tree | 14.04 | 29.78703 | | 4 | Random Forest | 9.675908 | 28.41273 | | 5 | Boosting | ~0 | 19.12495 | |
|  |
| Boston sample data The original Boston Housing dataset contains 506 records and 14 variables. We split this dataset in two parts with 70% and 30% of original data and stored them in 2 new datasets, namely, train and test sets, respectively. We train the model with 70% of the samples and test with the remaining 30%. We do this to assess the model’s performance on unseen data. We will be using train data which contains 70% data of the original dataset (354 obs.) for further analysis. Linear Regression The prices of the house indicated by the variable MEDV is our target variable and the remaining are the feature variables based on which we will predict the value of a house. **Initial Model** We will build our initial model by considering all the 13 feature variables and MEDV as response variable.  Our initial model –  *medv ~ crim + zn + indus + chas + nox + rm + age + dis + rad + tax + ptratio + black + lstat*  The p-values for INDUS and AGE is too high and indicates that these two variables are insignificant in the model.   |  |  | | --- | --- | | **Parameters** | **Values** | | **R-squared** | 0.7576 | | **Adjusted R-squared** | 0.7483 | | **MSE** | 22.477 | | **AIC** | 2122.148 | | **BIC** | 2180.188 | | **Test MAE** | 3.684 | | **Test MSPE** | 24.463 |   *Table 1: Parameter values of initial model*  AIC and BIC of the model, these are information criteria. Smaller values indicate better fit.  As per the result our model is only 75.76% accurate. So, the prepared model is not very good for predicting the housing prices. One can improve the prediction results using many other possible machine learning algorithms and techniques. **Best Model selection using Stepwise, LASSO, Best Subset Criteria** We select the best subset selection model as our best model: Full model - AGE - INDUS:  *medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio + black + lstat*  MSE = 17.90897  Test MSPE = 36.4963 Fitting Various Tree models**Regression Tree** We will now fit a regression tree on the training data. Using default cp=0.01 and no other constraints, tree is formed with 6 terminal nodes.    We calculate the MSE and MSPE using training and testing data, respectively.  MSE=  15.00729  MSPE= **32.42087**  We can compare both the methods – linear regression and regression tree to predict the median house price (medv).  We see that the MSE and MSPE of regression tree is similar to that of the linear regression model. But the MSE of regression tree is still less than the MSE of the linear model. **Bagging**  Bagging stands for Bootstrap and Aggregating. It employs the idea of bootstrap but the purpose is not to study bias and standard errors of estimates. Instead, the goal of Bagging is to improve prediction accuracy. It fits a tree for each bootstrap sample, and then aggregate the predicted values from all these different trees.  Results of Bagging Tree model-  Number of trees – 100  OOB Estimate - 14.04  Test MSE - 29.78703 **Random Forest** Random forest is an extension of Bagging, but it makes significant improvement in terms of prediction. The idea of random forests is to randomly select $m$ out of $p$ predictors as candidate variables for each split in each tree. Commonly, $m=\sqrt{p}$. The reason of doing this is that it can \*decorrelates\* the trees such that it reduces variance when we aggregate the trees.  Results of Random Forest Tree model-  Mean of squared residuals – 9.675908  % of Variance explained – 88.25  Number of trees – 500(default)  No. of variables tried at each split: 4    *Figure 2 – Impact of predictors(included) on the increasing of MSE and Node purity* |
| *Figure 3 – OOB error vs # of trees*    *Figure 4 – OOB and test error vs ‘mtry’* **Boosting** Boosting builds a number of small trees, and each time, the response is the residual from last tree. It is a sequential procedure. We use `gbm` package to build boosted trees.  The fitted boosted tree also gives the relative influence between each predictors and response variables. The below figure and table show that the two important influencers for our response variable are “lstat” and “rm” variables.    Figure 5 – Influence between response and predictors  *var rel.inf*  *rm rm 38.9701336*  *lstat lstat 34.4523407*  *dis dis 4.8528916*  *crim crim 4.7746512*  *ptratio ptratio 4.4436631*  *nox nox 2.9858404*  *age age 2.5725375*  *tax tax 2.2338527*  *black black 2.1334944*  *indus indus 1.4132176*  *chas chas 0.5838916*  *rad rad 0.4508102*  *zn zn 0.1326755*  *Table 1 – Relative influence of predictors on response*    *Figure 6 – Test error vs # trees*  The horizontal line is the best prediction error from random forests we obtained earlier. It is clear that Boosting provides lower out-of-sample mean squared prediction error. Comparing different model performance  |  |  |  |  | | --- | --- | --- | --- | | S.No. | Model | In-Sample MSE | Out-of-sample MSE | | 1 | Multiple Linear Regression | 17.90897 | 36.4963 | | 2 | Regression Tree (CART) | 15.00729 | 32.42087 | | 3 | Bagging Tree | 14.04 | 29.78703 | | 4 | Random Forest | 9.675908 | 28.41273 | | 5 | Boosting | ~0 | 19.12495 |   *Table 2 – Comparison of various tree models and linear model* |

It can be easily deducted that Boosting tree model fits the data in the best possible way with the lowest test error.