

|  |
| --- |
| 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 | GAM | 7.562186 | 13.96477 | | 4 | Neural Network | 5.129316 | 8.604409 | |
|  |
| 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. |
| **GAM** Residual diagnostics of linear regression model showed that the relation between medv and predictor variables may not be linear. Since the correct transformation of predictor variables is not known, GAM can be used to model non-linearity. GAM is fit using smoothing splines, s(), which is available in gam library in R. In the model, smoothing spline is used for all continuous variables except ‘chas’ and ‘rad’, which are of integer type and which have less than 10 unique values. It is not recommended to use smoothing splines on such variables.    *Since edf of ZN, ptratio and age is approximately 1 we consider them as linear term and remove s and run the model.*  A close up of text on a white background  Description automatically generated  We again run the model with edf 1 variables in linear    We can see that p values of zn, age, and chas are not significant so we remove them from the model.  Key Insights:   * MSE and MSPE from the GAM model comes out to be 7.562186 and 13.96477 respectively.   1. Neural Network   The response(in regression) needs to be standardized to [0,1] interval. It’s important normalize the response. If not, most of the times the algorithm will not converge. I chose to use the min-max method and scale the data in the interval [0,1].     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 | GAM | 7.562186 | 13.96477 | | 4 | Neural Network | 5.129316 | 8.604409 |   *Table 2 – Comparison of various tree models and linear model* |

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