DATA MINING II CASE STUDY 1: Boston Housing Data

Sagar Sahoo
MS Business Analytics

EXECUTIVE SUMMARY

PROBLEM AND APPROACH

The Boston housing dataset is a classic benchmark dataset in data mining area. Using exploratory data analysis we understood the data, distribution of the features, presence of potential outliers and used machine learning models to identify the most important features for predicting the median housing prices.

RESULTS

Various machine learning models i.e Linear Regression , CART, Bagging, Random Forests, Boosting and XG-Boost were implemented in R for predicting the median housing prices.

In order to compare the performance of all models, MSE was used as the evaluation metric.

The results are summarized below:

Model	In Sample MSE	Out of Sample MSE
Linear Regression	20.58	25.92
CART	21.62	14.61
Bagging	17.03	15.47
Random Forests	11.9	10.02
Boosting	0.99	11.6
XG-Boost	2.27	10.64

Table 1: Model Evaluation using MSE

It was found that Random Forests, Gradient Boosting and XG-Boost did good both for in-sample and out of sample predictions. Gradient Boosting however outperformed Random Forests for in-sample data leaving us further scope for checking possible case of overfitting.

The most important features highlighted by Ensemble Methods were closely similar. **Lsat, rm** and **dis** were the most important features in predicting the median housing prices in Boston.

INTRODUCTION

The dataset "Boston" on Boston Housing Price is taken from the MASS library of R. It has 506 observations and 14 attributes. Below is a brief description of each feature of the dataset:

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; else 0)
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

The report is organized in such a way as to demonstrate the entire process right from importing the data, to exploratory data analysis of the dataset to understand the distribution and correlation of various features in influencing the algorithm, to designing / training ML models, and evaluation of the models.

EXPLORTORY DATA ANALYSIS

The dataset has 506 observations and 14 attributes.

crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
0.00632	18	2.31	0	0.538	6.575	65.2	4.09	1	296	15.3	396.9	4.98	24
0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.9	9.14	21.6
0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.9	5.33	36.2
0.02985	0	2.18	0	0.458	6.43	58.7	6.0622	3	222	18.7	394.12	5.21	28.7

Table 2 5 Sample Observations from the Dataset

NATURE OF ATTRIBUTES

All the attributes are continuous in nature except for the column CHAS which is categorical and coded as 0 and 1 . The column RAD is an ordinal variable.

Attribute	Type	Sample Values
$ crim	num	0.00632 0.02731 0.02729 0.03237 0.06905 ...
$ zn	num	18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
$ indus	num	2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
$ chas	int	0 0 0 0 0 0 0 0 0 0 ...
$ nox	num	0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
$ rm	num	6.58 6.42 7.18 7 7.15 ...
$ age	num	65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
$ dis	num	4.09 4.97 4.97 6.06 6.06 ...
$ rad	int	1 2 2 3 3 3 5 5 5 5 ...
$ tax	num	296 242 242 222 222 222 311 311 311 311 ...
$ ptratio	num	15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
$ black	num	397 397 393 395 397 ...
$ lstat	num	4.98 9.14 4.03 2.94 5.33 ...
$ medv	num	24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...

Table 3 Datatype of the Attributes

CORRELATION

In order to understand the correlation between the attributes, we generate the correlation plot

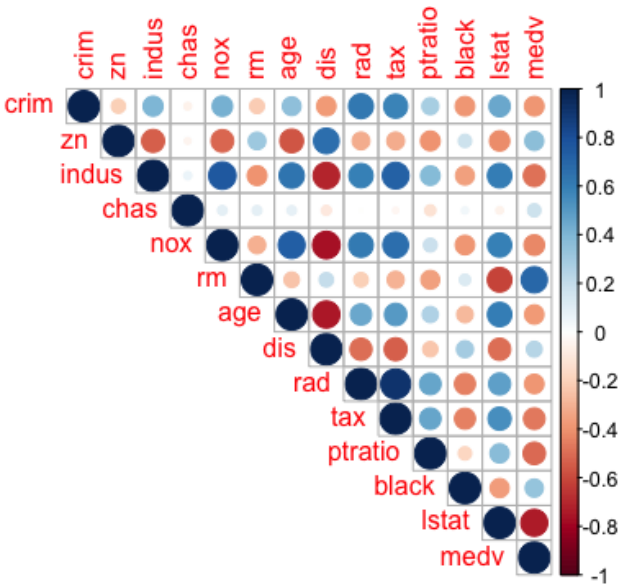

Fig 1 Correlation Plot

From correlation plot, some of the observations made are as follows:

- Median value of owner-occupied homes (in 1000$) increases as average number of rooms per dwelling (RM) increases and it decreases if LSTAT increases
- rad and tax have strong positive correlation.
- crim is strongly correlated with rad and tax.
- indus has strong positive correlation with nox and tax.
- nox or nitrogen oxides concentration (ppm) increases with increase in proportion of non-retail business acres per town and proportion of owner-occupied units built prior to 1940.

SCATTER PLOT

We used scatter plot to get a visual intuition of the relationship between mdev and dependent variables as well:

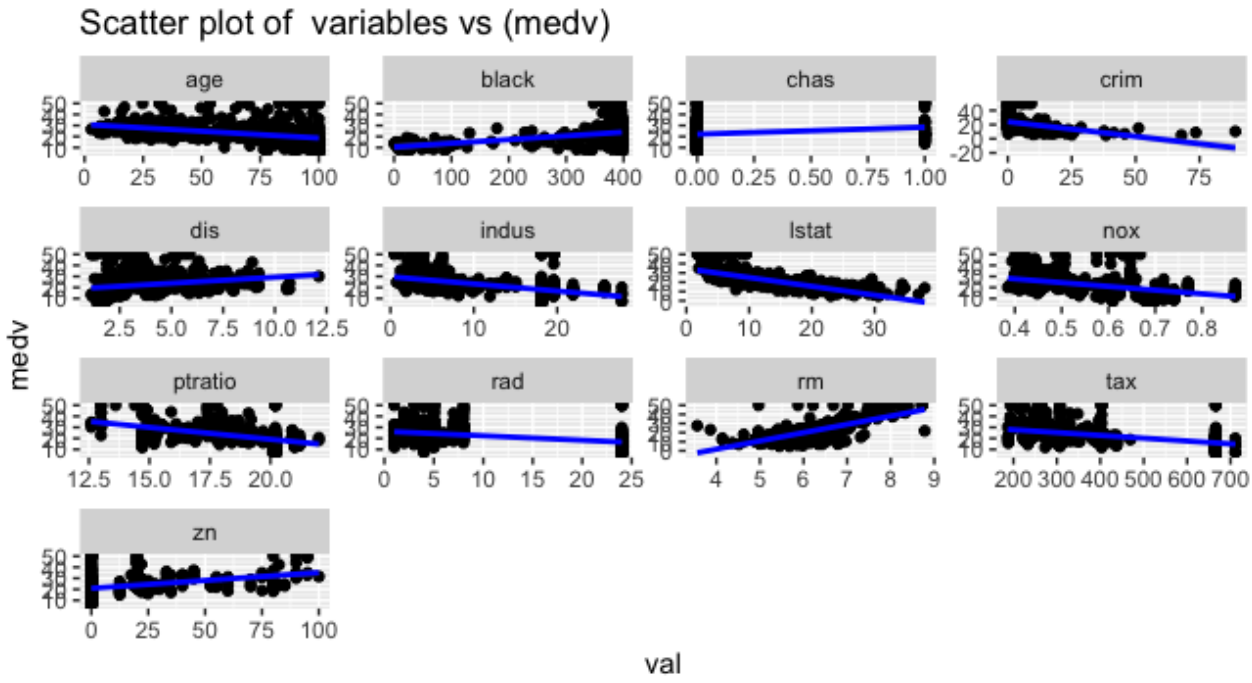

Fig 2. Scatter Plot of all variables against medv

OUTLIERS

Box Plots are used to detect the outliers in the dataset. Attributes black, crim, dis, lstat, ptratio, rm and zn shows outliers.

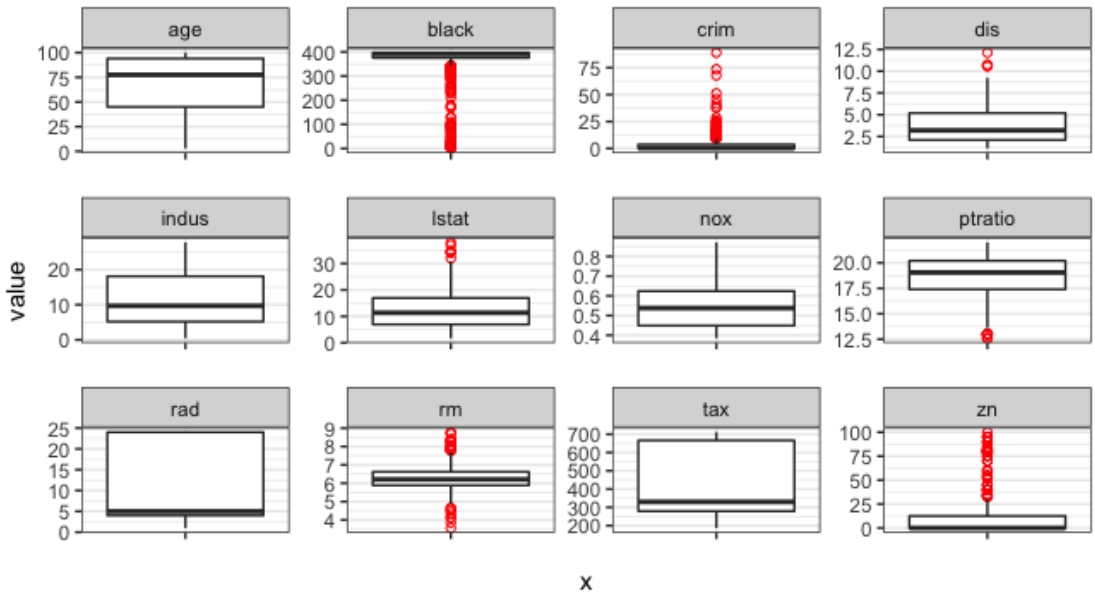

Fig 3. Box plot of all independent variables

HISTOGRAMS

The histograms of the dependent variables gives the following insights:

- Rad and Tax seem to have two different peaks
- rm follows normal distribution
- Most of the distribution of variables are skewed

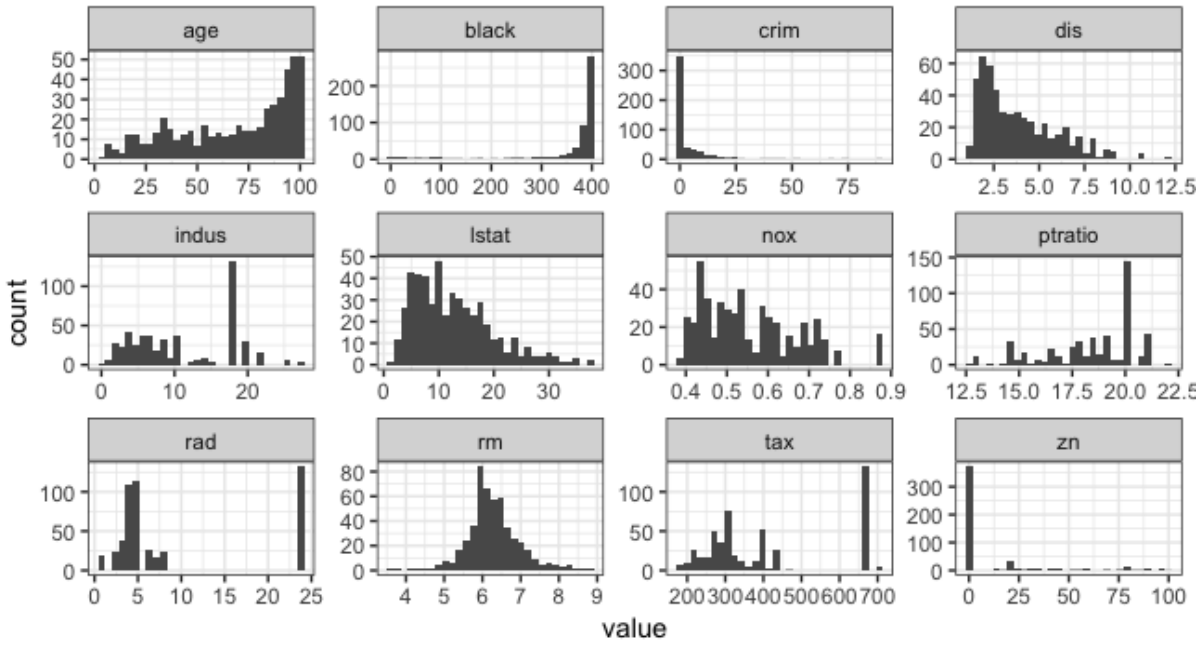

Fig 4. Histogram of all independent variables

MODEL BUILDING

LINEAR REGRESSION:

From correlation table of features against response, we could see that there exists some amount of correlation for all the responses. Hence we included all features in Linear Regression Model.

Features	Correlation with MEDV
CRIM	-0.3883046
ZN	0.3604453
INDUS	-0.4837252
CHAS	0.1752602
NOX	-0.4273208
RM	0.6953599
AGE	-0.3769546
DIS	0.2499287
RAD	-0.3816262
TAX	-0.4685359
PTRATIO	-0.5077867
B	0.3334608
LSTAT	-0.7376627
MEDV	1

Table 4: Correlation of Features with Response

The results of linear regression are summarized below:

Total No of Features	No of Significant Features	In Sample MSE	Out of Sample MSE
13	11	20.58	25.92

Table 5: Linear Regression Summary

Indus and Age were the two features flagged as In-significant by the regression model based on p-value at 95% level of significance. But correlation table showed that these features are negatively correlated, hence we cannot drop these features from our model.

CART:

We used CART (Classification and Regression Trees) Model to regress MEDV against all the features.

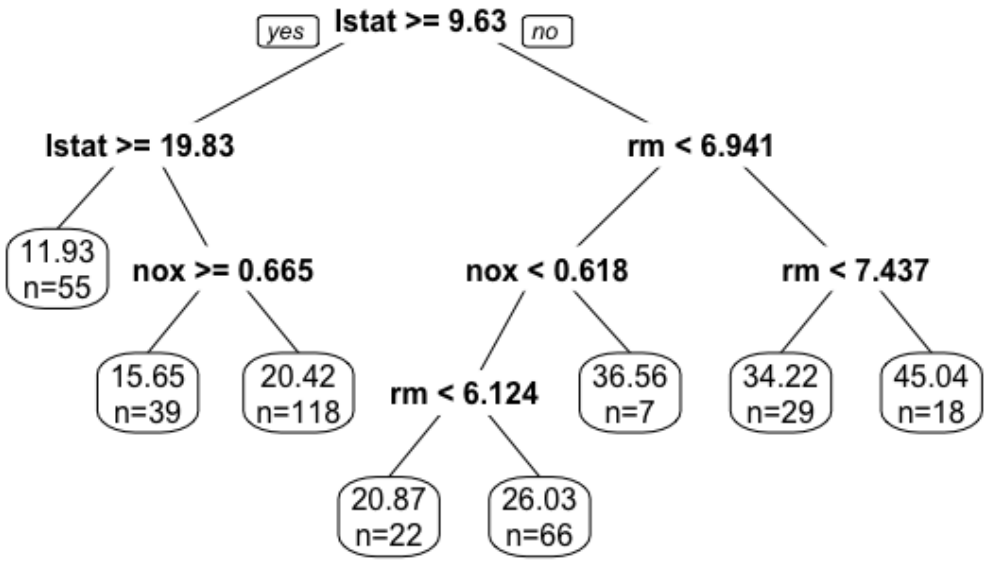

Fig 5: Plot of Trained Decision Tree

Lstat>=9.63 was the root node and only 3 features were used to build the tree.

The results from RPART is presented below:

No of Features	In Sample MSE	Out of Sample MSE
3	21.62	14.61

Table 6: CART Summary

We could see that the Out of Sample MSE for CART is lower as compared to Linear Regression in our case.

BAGGING:

Bootstrap Aggregation or Bagging is a procedure that can be used to reduce the variance for those algorithm that have high variance. Decision trees are sensitive to the specific data on which they are trained. If the training data is changed (e.g. a tree is trained on a subset of the training data) the resulting decision tree can be quite different and in turn the predictions can be quite different.

We used R package "ipred" to build the Bagging model on the training data using all the features. The results are summarized below (nbagg i.e Number of bootstrap aggregations =100):

No of Significant Features	In Sample MSE	Out of Sample MSE
13	17.03	15.47

Table 7: Bagging Summary

We used, initially Number of Bootstrap Aggregations as 100 and got in-sample MSE of 17.03. We can check if we can improve the MSE by varying the nbagg parameter.

Below is the plot for the in-sample MSE for nbagg value ranging from 10 to 200 in steps of 10.

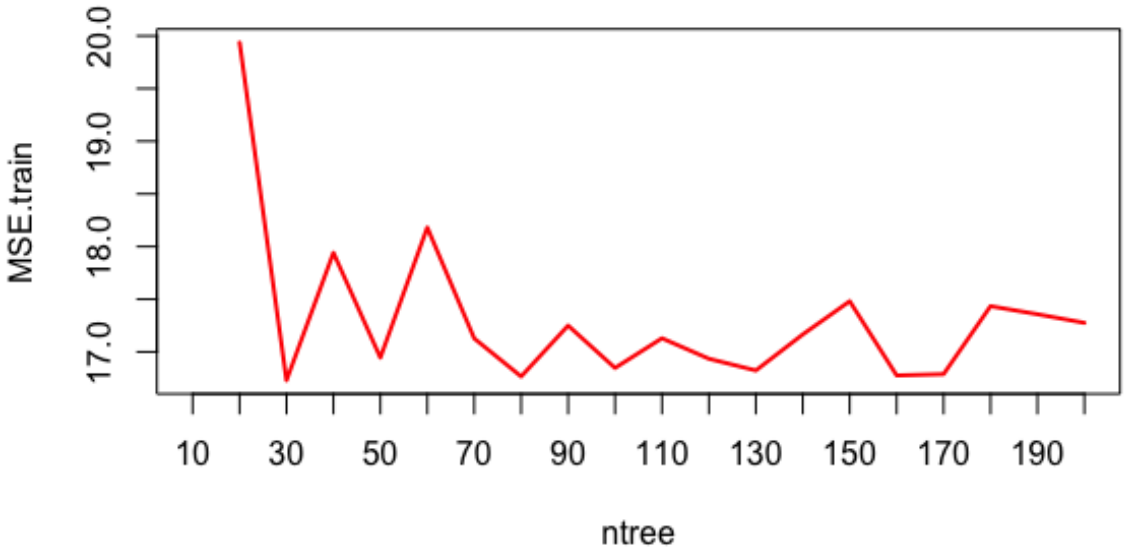

Fig 6: variation of Training MSE against ntrees

The plot shows that the MSE smoothens out somewhat round 100, hence our assumption of nbagg as 100 works fine in our training data.

RANDOM FORESTS

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, for classification trees, $m = \sqrt{p}$. And for regression trees, $m = p/3$ The reason of doing this is that it can *decorrelates* the trees such that it reduces variance when we aggregate the trees.

Random Forest algorithm was applied using the R library "randomForest". With default parameters, the number of trees was set to 500 and No of variables for each split was 4.

No of Subset Features	In Sample MSE	Out of Sample MSE
4	11.90	10.02

Table 8: Random Forest Summary

Below are the features listed in order of Importance (highest to lowest) from fitted Random Forest Model:

	%IncMSE	IncNodePurity
lstat	61.7457014	8600.7682
rm	29.5165102	7729.0114
nox	10.3015091	2330.7231
crim	9.090411	1818.2422
dis	7.7677612	1904.8554
indus	6.961315	1934.8796
ptratio	4.9931812	1389.989
age	4.1751067	895.0489
tax	2.8634733	744.3266
black	1.3248681	563.1613
rad	0.8702955	227.8545
zn	0.501094	148.2329
chas	0.4205753	200.2955

Table 8: Feature Importance

Using parameter tuning, i.e by varying the ntrees argument, we plotted the in-sample MSE and found the optimal value of ntrees around 470 close to our assumed default value of 500.

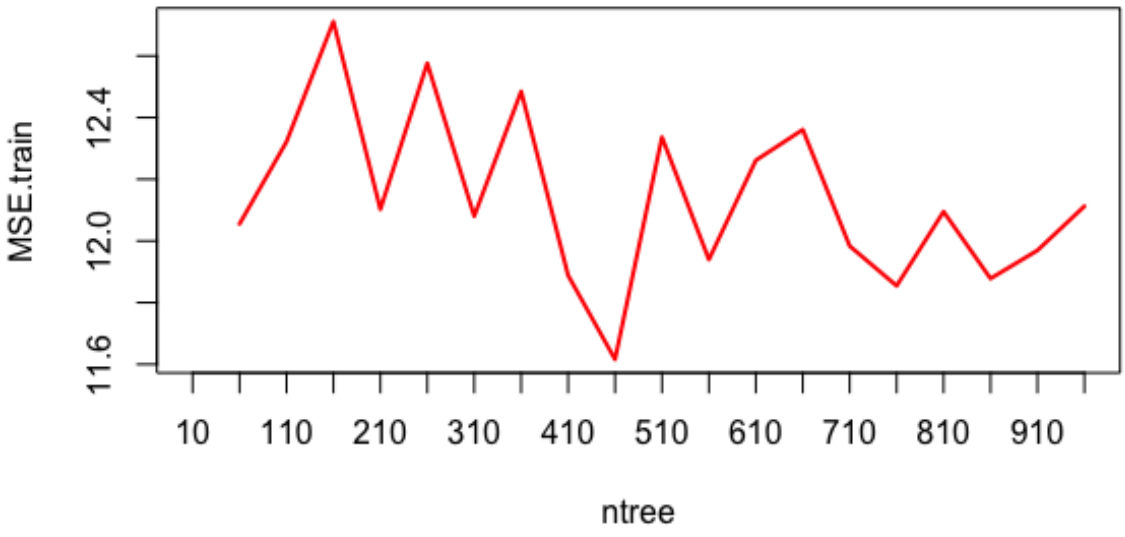

Figure 7: Variation of Train MSE against ntrees

BOOSTING

The term 'Boosting' refers to a family of algorithms which converts weak learner to strong learners. Boosting is an ensemble method for improving the model predictions of any given learning algorithm. The idea of boosting is to train weak learners sequentially, each trying to correct its predecessor.

The library "gbm" is used to build the Boosting Model in R. Below results were obtained using gbm. The arguments n.trees was set to 10000, shrinkage to 0.01 and interaction.depth to 8

The in-sample MSE is almost 0 i.e it has fit the in-sample data perfectly. The out of sample MSE is on the higher side as compared to our Random Forests.

In Sample MSE	Out of Sample MSE
0.02	12.48

Table 9: Boosting Summary

In order to check possible case of overfitting, we now vary the number of trees and then check the in-sample MSE.

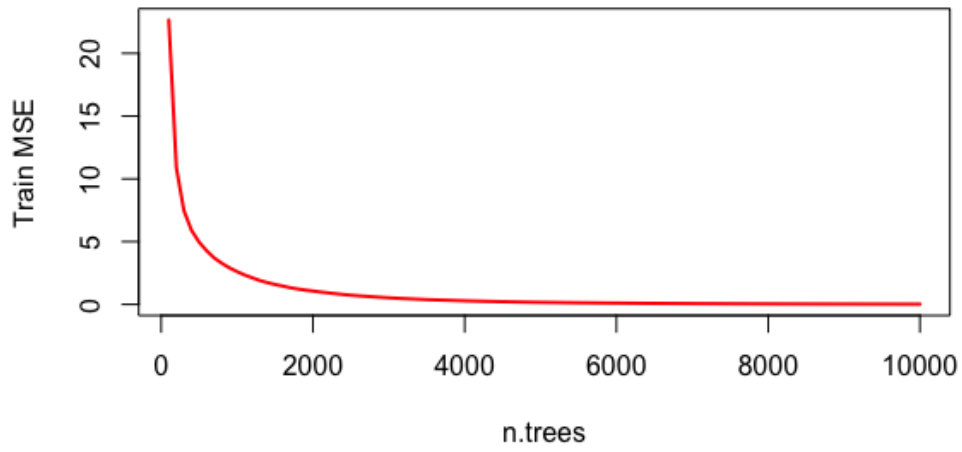

Figure 8: Variation of Train MSE against ntrees

From the plot, it could be seen that the Training MSE smoothens out with n.trees as 2000. We now choose 2000 as n.trees argument to our Boosting algorithm and we check the in-sample MSE

In Sample MSE	Out of Sample MSE
0.99	11.60

Table 10: Optimized Boost Summary

The out of sample performed better with n.trees as 2000. Using hyper parameter tuning, we can say that the overfitting has reduced.

We as well checked the importance of features of our final fitted boosting model.

Var	rel.inf
lstat	39.6778669
rm	30.224386
dis	9.7873348
nox	5.4770235
crim	3.54142
age	3.21637
black	2.3328721
ptratio	2.1276601
tax	1.4494284
indus	0.9640537
chas	0.6626671
rad	0.4006366
zn	0.1382807

Table 11: Optimized Boosting Feature Importance

XGBOOST

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way. The same code runs on major distributed environment (Hadoop, SGE, MPI) and can solve problems beyond billions of examples.

The library "**xgboost**" was used to implement the algorithm. Though the algorithm needs several arguments to build the model, below arguments were used:
max_depth=3
eta = 0.2
nthread=3
nrounds=40
lambda=0
objective="reg:linear"

The results obtained from XGBoost are summarized below:

In Sample MSE	Out of Sample MSE
2.27	10.64

Table 12: XG-Boost Summary

The feature importance along with gain are also listed below:

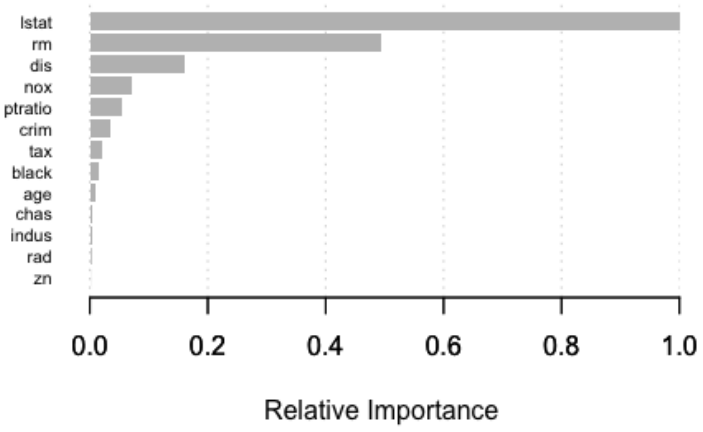

Figure 9: Feature Importance in XG-Boost

CONCLUSION

We evaluate the MSE using various machine learning models i.e Linear Regression ,CART, Bagging, Random Forests and Boosting for predicting the median housing prices.

The results are summarized below:

Model	In Sample MSE	Out of Sample MSE
Linear Regression	20.58	25.92
CART	21.62	14.61
Bagging	17.03	15.47
Random Forests	11.9	10.02
Boosting	0.99	11.6
XG-Boost	2.27	10.64

Table 13: Model Evaluation using MSE

It was found that Random Forests, Boosting and XG-Boost did good both for in-sample and out of sample predictions. Boosting however outperformed Random Forests for in-sample data leaving us further scope for checking possible case of overfitting.

The most important features highlighted by Random Forests and Boosting were closely similar. **Lsat, rm** and **dis** were the most important features in predicting the median housing prices in Boston.