Forecasting House Prices by Regression and Classification

Abstract – There are two main parts of this paper. First part consists of regression models developed in order to predict continuous house prices. Second part consist of classification model in order to predict binary labels. Labels are introduced as below and above average of overall dataset. Performance metrics for regression models are defined as root means squared error (RMSE) and R squared. Classification part have only one performance metric to evaluate models as accuracy.

For both parts of the modelling part 3 attempts were made. Regression part have decision tree, random forest and bagging approaches. Classification part have three algorithms as decision tree, random forest and gradient boosting. A comparative analysis made in conclusion segments that shows a progressive improvement with each more complex algorithm with one exception. According our conclusion, random forest classifier provide an optimal solution for complexity / accuracy trade off. On the other hand superior performance achieved with bagging approach for regression.

Keywords—Random Forest, Linear Regression, R-Square, RMSE, Forecasting, Model Deployment, Model Validation

1. INTRODUCTION

The dataset consists of 1456 observations and 81 features in total. Before doing any analysis, it would be useful to have a close look to features. 38 features are in integer type while 43 features are character. Examples of features which are in character format are MSZoning, Street and Alley. All these features are categorical variables representing location related information of houses. There are also numeric features such as FullBath which is indeed ordinal format. For instance, 0 means there is no bath while 3 belongs three baths available in the house. Lastly, there are a few continuous features as well. For instance, GrLivArea represents size of total living area above the ground.

The histogram below shows how our target feature which is SalePrice is distributed. We can see that it ranges from $34,900 to $755,000 and has mean of $182,261. When we investigate the skewness of the distribution, we observe that it is right skewed. Therefore, it can improve the model if we apply logarithmic transformation to this feature.

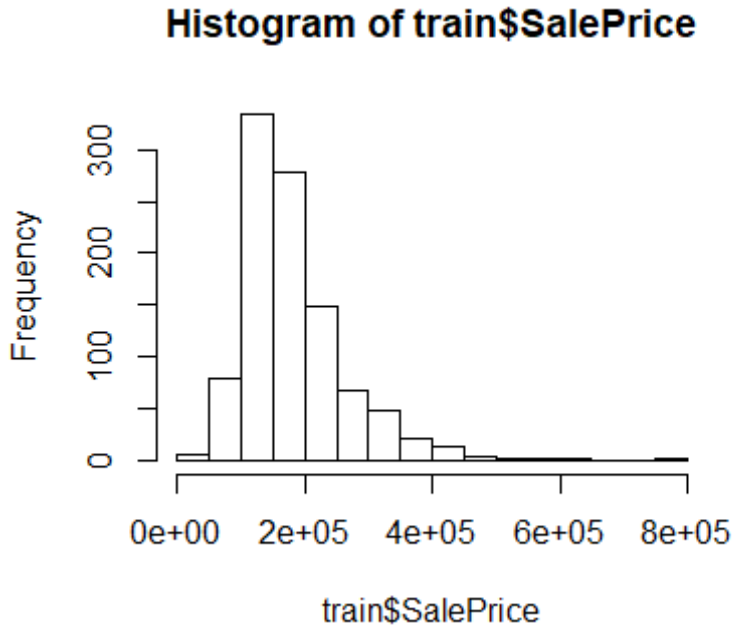

Figure 1: Histogram for target variable

2. DATA AND ANALYSIS

Figure 2 shows most significant relationships between all numerical variables. Firstly, it is worth noting that there are some numerical variables which has strong correlation between each other. For example, correlation coefficient between GarageCars and GarageArea is greater than 0.6. Secondly, some features are highly correlated with target variable such as TotalBsmtSF, GrLivArea, X1stFlrSF and GarageArea. It is better to look for outliers within these features since they may be highly important explanatory variables in the regression model.

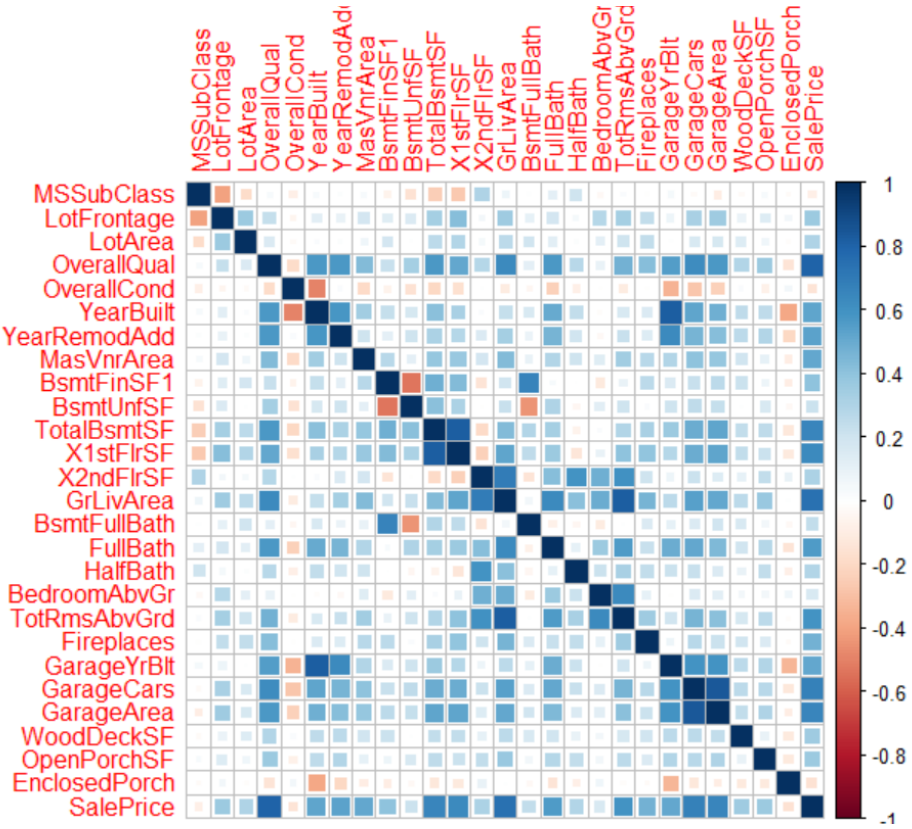

Figure 2: Heatmap of numerical features

It is observed that there are certain data points which seems as outlier in all scatter plots below. Excluding points which is above 2,500 for TotalBsmtSF yields dataset without outliers. So our model will not be affected by those extreme points.

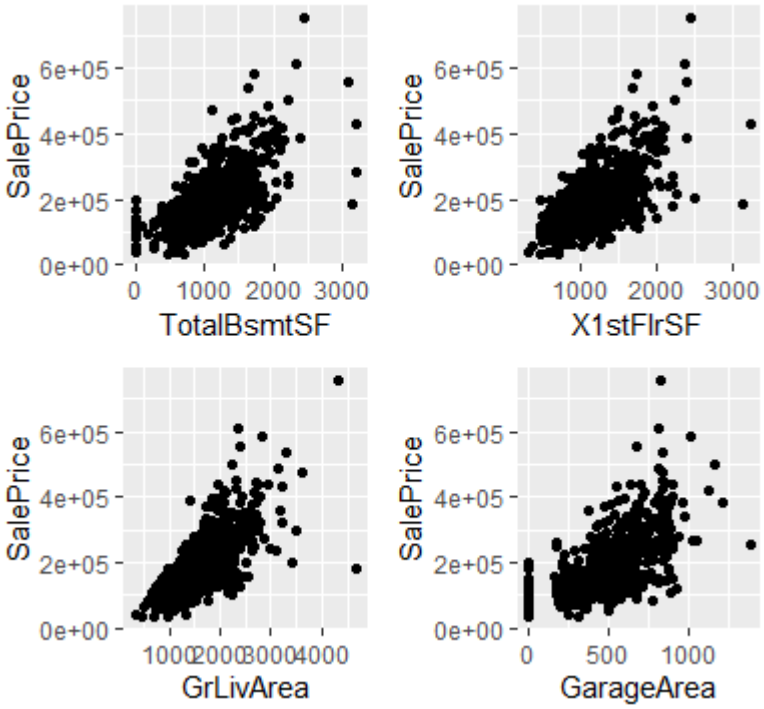

Figure 3: Outlier exploration in top correlated features

The next step for pre-processing part is checking whether there exist any null values. There are some features which has almost 1,000 missing values. This is high number considering that we have 1,456 observations in total. Some of those null values represent that the house does not have such a feature. Thus, null values hold information and should be replaced with ‘None’.

Lastly, i applied log-transformation to the SalePrice target variable as i mentioned above. The distribution of data now looks closer to normal distribution as can be seen in Figure 4.

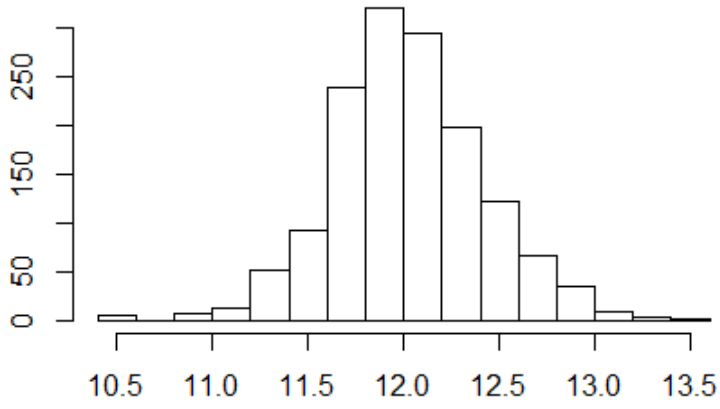

Figure 4: Distribution of target variable after log-transformation

3. MODEL BUILDING & VALIDATION

3.1 Regression Models

I applied both decision tree and random forest regressors for this project. As final step a bagging approach deployed with random forest regressor as well. My target was to create a best fit model with lowest error margin. In order to achieve this goal all models built with grid search and cross validation techniques.

Below table shows summary statistical information about our test data subset.

Min	1st Quartile	Median	Mean	3rd Quartile	Max
$52,500	$129,000	$162,000	$178,018	$206,900	$745,000

Table 1: Summary information on test data subset

3.1.1 Decision Tree Regressor

I deployed a decision tree regression model with its default parameters as a benchmark. As anticipated a larger tree has been produced with 10 terminal nodes. Root Mean Squared Error (RMSE) for this model was $46,911 for test subset and R squared value was 63%.

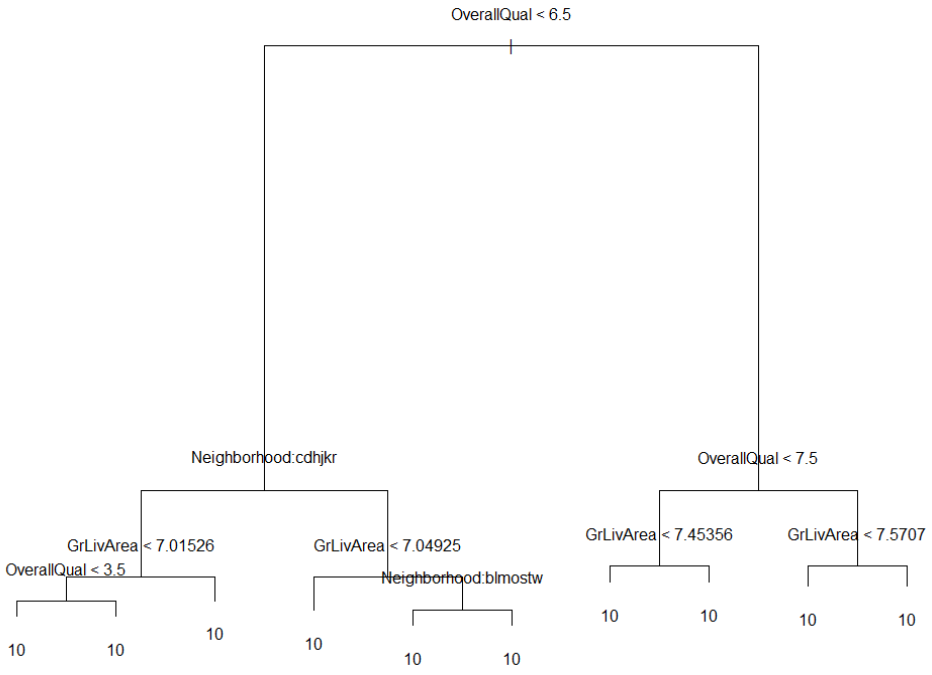

Figure 5: Decision Tree Regression with Default Parameters

In order to create a more simplistic and tree to save up run time and improve generalization of the model i explored if there is any possibility to prune the tree without losing any informative aspect of the model. Below graph shows creating more than 6 terminal nodes does not create much difference. Thus, a pruned tree with 6 terminal nodes would be still a similarly good fit model for our problem.

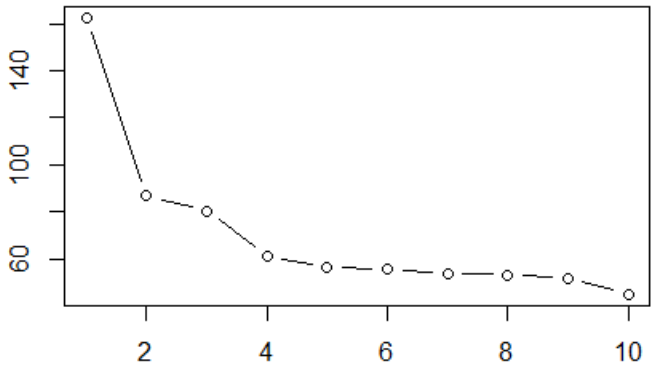

Figure 6: Contribution of each terminal node to the model

After pruning the original tree, final model includes 6 terminal nodes and 3 features. RMSE of the pruned model was $50,197 which is 7% higher compare to original model. R squared value was 58%.

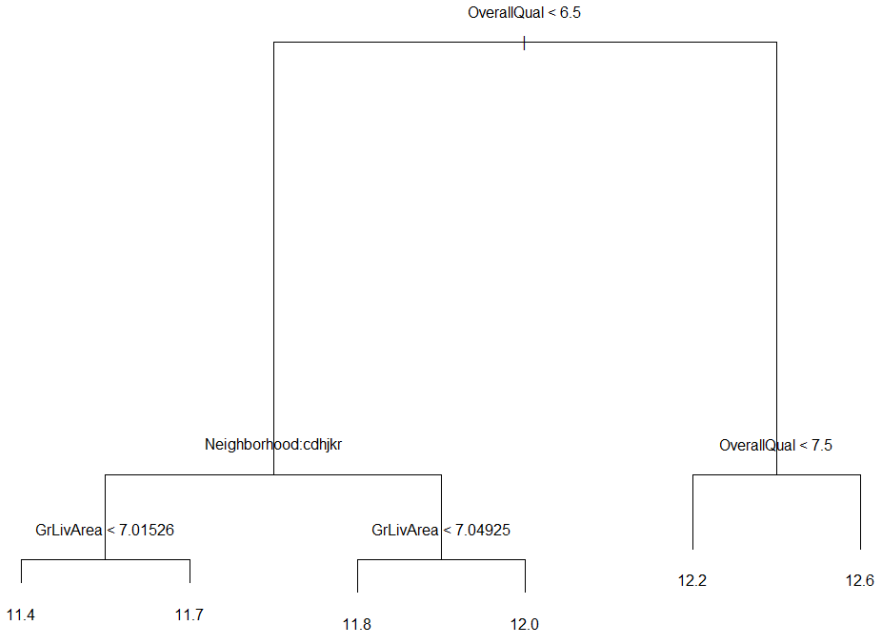

Figure 7: Pruned Decision Tree Regression with 6 Terminal Nodes

3.1.2 Random Forest Regression

Random Forest deployment was performed with a grid search. Our grid includes two parameters. Ntree is the number of trees to grow that i tried values of 100, 250, 750, and 1000. Mtry is the number of variables randomly sampled as candidates at each split which we tried the values of 2, 3 and 4.

Grid search with 10fold cross validation showed a random forest model with Ntree parameter set as 1,000 and mtry parameter set as 4 produced best result. RMSE for this parameter on test data was $34,458 and R squared was 80%. This means that we decreased the RMSE by 46% compared to decision tree model.

Below figures shows our model successfully predicts more frequent values of house prices which is between $100,000 and $ 150,000 however with the increasing price range error amount increasing as well.

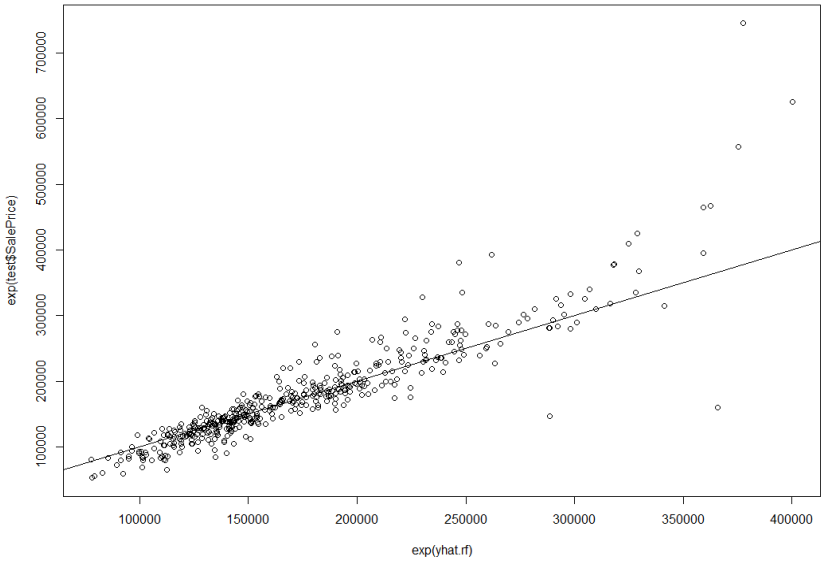

Figure 8: Random Forest Regression Model

Since not all features would have the same importance for our model it is better to present a future importance table for a reference to further studies. A more sophisticated data collection process focusing these areas would enable us to create better models. As seen in below figure there are three dominant features that defines the price of a house according to our model. First one is OveralQual which means overall quality of the house. Second important feature is GrLivAre which means above ground living area and finally the neighbourhood. It is clear that our model predicts the price of a house parallel to human common sense.

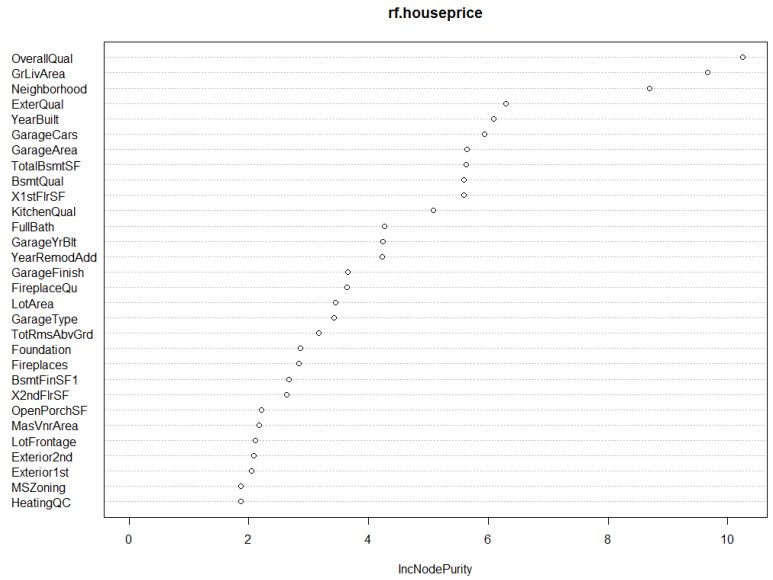

Figure 9: Feature Importance for Random Forest Model

3.1.3 Bagging Approach

Bootstrap aggregation is part of ensemble machine learning methods. Bootstrap refers to random sampling with replacement. Bootstrap allows us to better understand the bias and the variance with the dataset and stabilize trees grown.

Since our original random forest showed signs of improving results while the two grid search parameters were increasing, i applied model with ntree 1500 and mtry 8. RMSE improved dramatically to $32,156 and R squared to 87%.

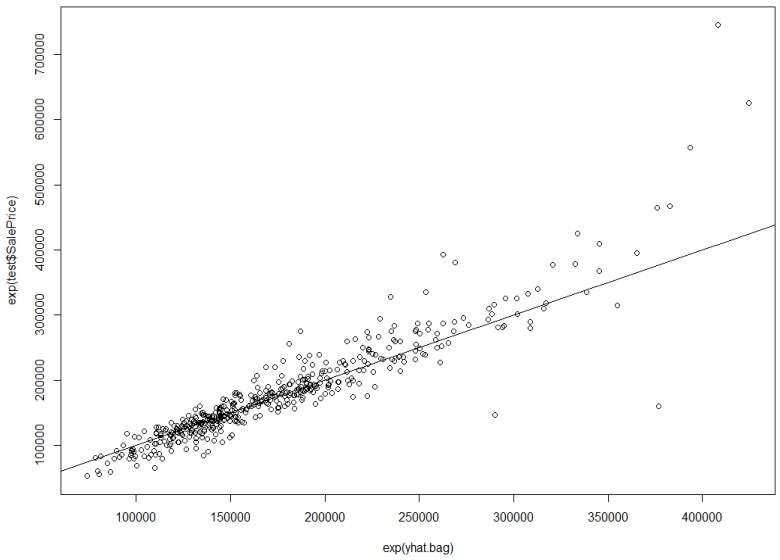

Figure 10: Random Forest Regression Model With Bagging Approach

3.2 Classification Models

Same structure of progressive evaluation of models applied to classification segment as well. Decision tree, Random Forest and Gradient Boosting classification models have been used for a binary classification.

In order to create binary labels for classification, a new column is added to dataset as an indication of if the price is below or above the mean of all dataset.

3.2.1 Decision Tree Classifier

Our first attempt to create a decision tree with default parameters resulted a complicated tree as expected. Below figure presents unpruned default tree as a benchmark for this classification problem.

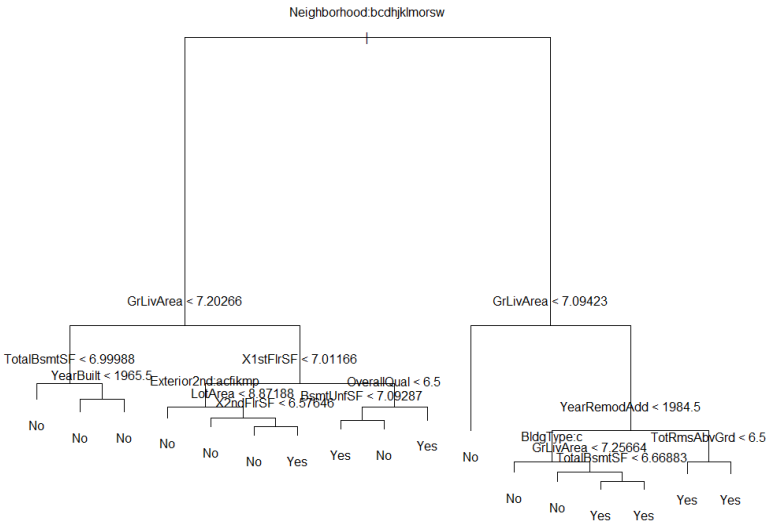

Figure 11: Unpruned Decision Tree Classifier

This complicated model yields an accuracy of 87% for our test dataset. Confusion matrix in order to observe both precision and recall is below.

Actual / Predicted	**NO**	**YES**
NO	190	18
YES	41	212

Table 2: Confusion Matrix for Unpruned Decision Tree

I continue to explore if there is any possibility to create a simpler model by pruning the original tree. As presented in the below figure, after 3 terminal nodes there is not much improvement in our model.

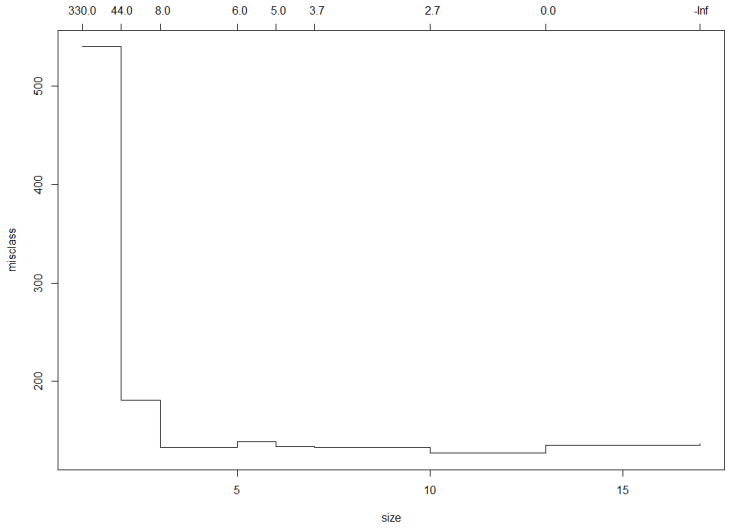

Figure 12: Misclassification per terminal nodes

Pruning the original tree to 3 terminal nodes resulted a very simple model with only 2 features. According to this model house of a price can be classified as above or below mean by only considering its neighbourhood and above ground living area.

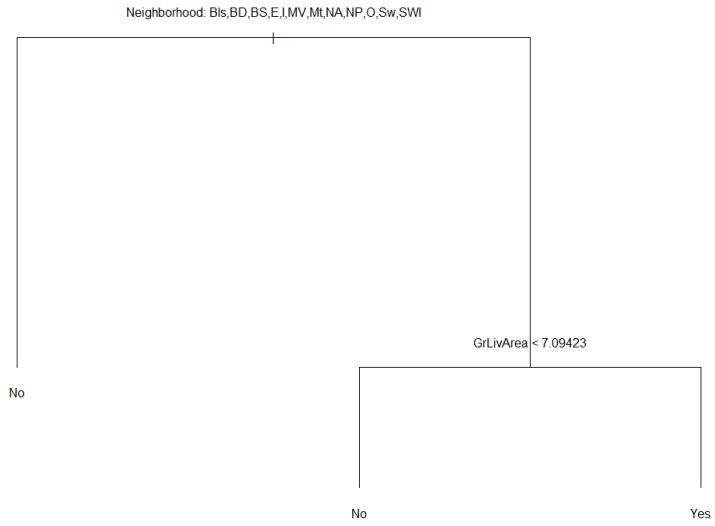

Figure 13: Pruned Tree Classifier

This simple model provided an accuracy ratio 88%. Confusion matrix can be seen below.

Actual / Predicted	**NO**	**YES**
NO	214	39
YES	17	191

Table 3: Confusion Matrix for Pruned Decision Tree

3.2.2 Random Forest Classifier

Accuracy of random forest classification with default parameters was 93% on test subset. This means an improvement of 5% compared to our pruned decision tree classifier. Feature importance and confusion matrix can be seen below.

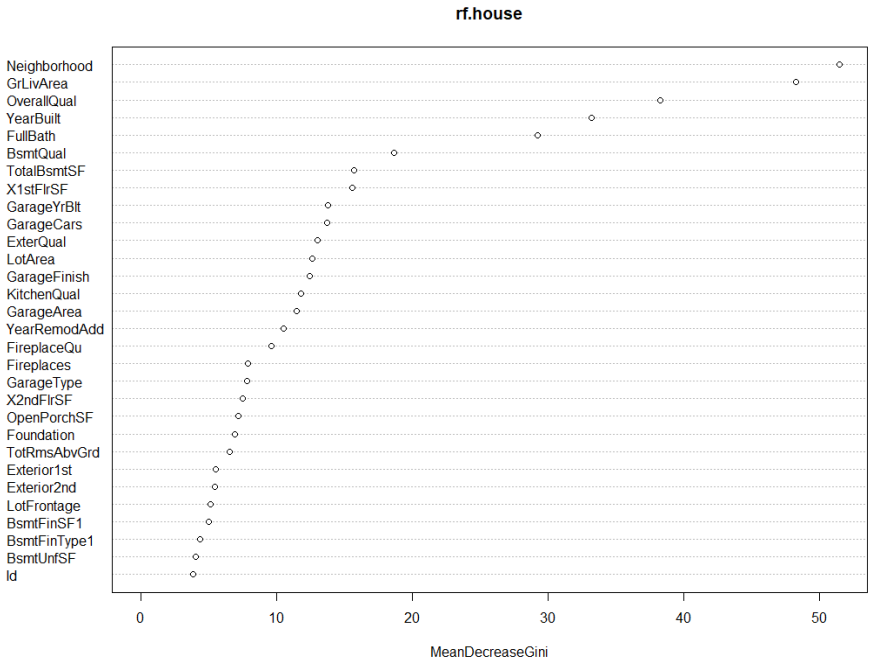

Figure 14: Feature Importance for Random Forest Model

Actual / Predicted	**NO**	**YES**
NO	218	20
YES	13	210

Table 4: Confusion Matrix for Random Forest Model

3.3.3 Gradient Boosting Classifier

Let's continue to explore if there is any room for further improvement on accuracy score with gradient boosting algorithm. Below figure shows ideal number of trees as 2193 under bagging approach. Shrinkage parameter was set to 0.01 and 10 fold cross validation was applied.

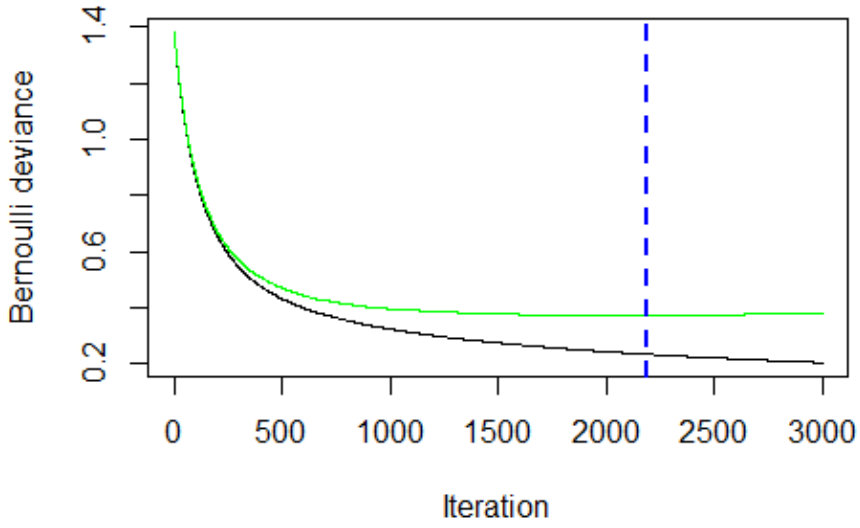

Figure 15: Bernoulli Deviance per Number of Trees

We have an accuracy score of 93% again. Unfortunately, we could not accomplish a better accuracy. However, we had better performing model by using less trees. Confusion matrix for our final model can be seen below.

Actual / Predicted	**NO**	**YES**
NO	214	16
YES	17	214

Table 5: Confusion Matrix for Bagging Approach

4. CONCLUSION

In this project we tried to predict house prices with 2 different methodology. Regression by its nature can provide continuous prediction which is more suitable for pricing prediction compare to a binary classification. However, choice shall be defined by the need. If we try to predict house price as an ultimate target, then regression algorithms should be used. On the other hand, with classification we can understand if a house is below or above the mean price range of its neighbourhood and drivers of this result.

A clearer choice can be made on the which classification or regression algorithm performs better on this domain with objective performance metrics. A comparison between models can be found in table 6.

Regression Model	RMSE	R Squared	Classification Model	Accuracy
Decision Tree	$50,197	58%	**Decision Tree**	88%
Random Forest	$34,458	80%	**Random Forest**	93%
Bagging	$32,156	87%	**Gradient Boosting**	93%

Table 6: Model Comparison

Regression models shows a clear improvement in every step. 87% R squared value shows our model with bagging approach is quite a good fit for this problem. Thus, it would be the ultimate choice amongst regression model.

Choice between classification models seems harder with this result. Although gradient boosting shows a minor improvement compared to random forest model it is less than 1%. Moreover, gradient boosting algorithms are trickier to tune with its hyper parameter set compare to random forest. If there is a need for absolute accuracy with more invested time gradient boosting can yield better performance. However, 93% accuracy should be enough for such problem and deploying a random forest model is much easier.