Exploring the effect of different features on house prices using multiple machine learning algorithms, and analyzing how accurate a model can predict the sales price of houses in Illinois metropolitan area.

DSCI 401 Machine Learning 2

Nisi Mohan Kuniyil

2021-12-19

Contents

Executive Summary

Real estate is one of the complex markets in the world and there is a growing interest in applying data science practices to understand the factors that are driving the sales price of houses. The availability of a vast amount of data opens up the possibilities of training machine learning models to extract insights and better understand property pricing in the market. This project aims to research these possibilities by applying multiple machine learning algorithms; decision tree, random forest, and gradient boosting on the Illinois data.

The Illinois real estate data records observations of different houses in the Illinois area. The dataset includes a wide range of features and their corresponding sales price. To gain insights into these features and how they impact property prices, we train multiple regression models with different algorithms and analyze how different features help the model predict the prices. To effectively evaluate different models and understand how the models will perform in a real-world setting, the dataset is split into two groups (training and test set) and only the training set is made accessible to the model. After training, predictions are made against the test set and the model performance is evaluated using a metric called RMSE (Root Mean Squared Error). The metric RMSE helps us understand how much the model was off in predicting the correct sales prices in the test set. The lower the RMSE the closer the model predictions are to the actual house prices. Setting up the training experiments in this fashion gives us a reliable method to understand how the model will perform in the real world. Also, this helps us avoid the model overfitting problem, where the model shows good performance on the data set it was trained on, but when deployed and evaluated on unseen data, the model becomes useless. Apart from the quantitative RMSE metric, we also evaluate the predictions made by the models using visualizations to figure out how close each prediction is to its corresponding actual prices. While the RMSE metric gives us an overall performance of the model, the visualizations enable us to individually inspect each data point in the test set and understand how the model performs for different segments of houses (eg: expensive vs non-expensive).

Using the experiment and evaluation strategies mentioned above, three types of regression models were trained using a decision tree, random forest, and gradient boosting algorithms. The random forest regressor model performed the best on the unseen test data set with the lowest RMSE score as well as having consistent predictions for all the data points in the test set. Also, using random forest we were able to extract the most important features the model used to make its decisions and predict the house prices. Using this random forest feature importance analysis, we are able to get a solid understanding of the factors that has a major impact on the sales prices of houses in the Illinois area.

To summarise, this project explored three different machine learning models to check if there exist meaningful features in the Illinois dataset that can be used to accurately predict the sales prices of houses. Although we were able to achieve decent predictive performance with all three of the models we trained, the random forest algorithm gave us the best performance on the test set as well as provided insights into the most important features that are driving the market.

Introduction

The real estate market is a highly variable market where property prices are driven by a lot of factors. No matter what the circumstances are now, the demand for housing continues to grow as same as before the pandemic. The housing bubble is a serious concern for the property-buying people, and it will help the buyers and sellers if they can understand the factors driving the housing price. Now that we live in a world of Big Data, there is a lot of data available to analyze the statistical and contextual factors that have an influence on housing prices.

This project will analyze the Illinois real estate data and identify meaningful features from a large set of potential factors that has a major impact on property prices. Multiple machine learning algorithms will be applied to the dataset to compare and contrast the importance of different features in predicting the market prices of properties.

Objective

To build multiple machine learning models like Decision Tree, Random Forest, and Gradient Boosting to predict the housing price and analyze the feature importance in predicting the price. Before building a model Exploratory Data Analysis(EDA) will be conducted to obtain a better understanding of our data. This will help in choosing the important features and thus help improve the performance of our models.

Source

The Illinois real estate dataset is used for the analysis of housing price.

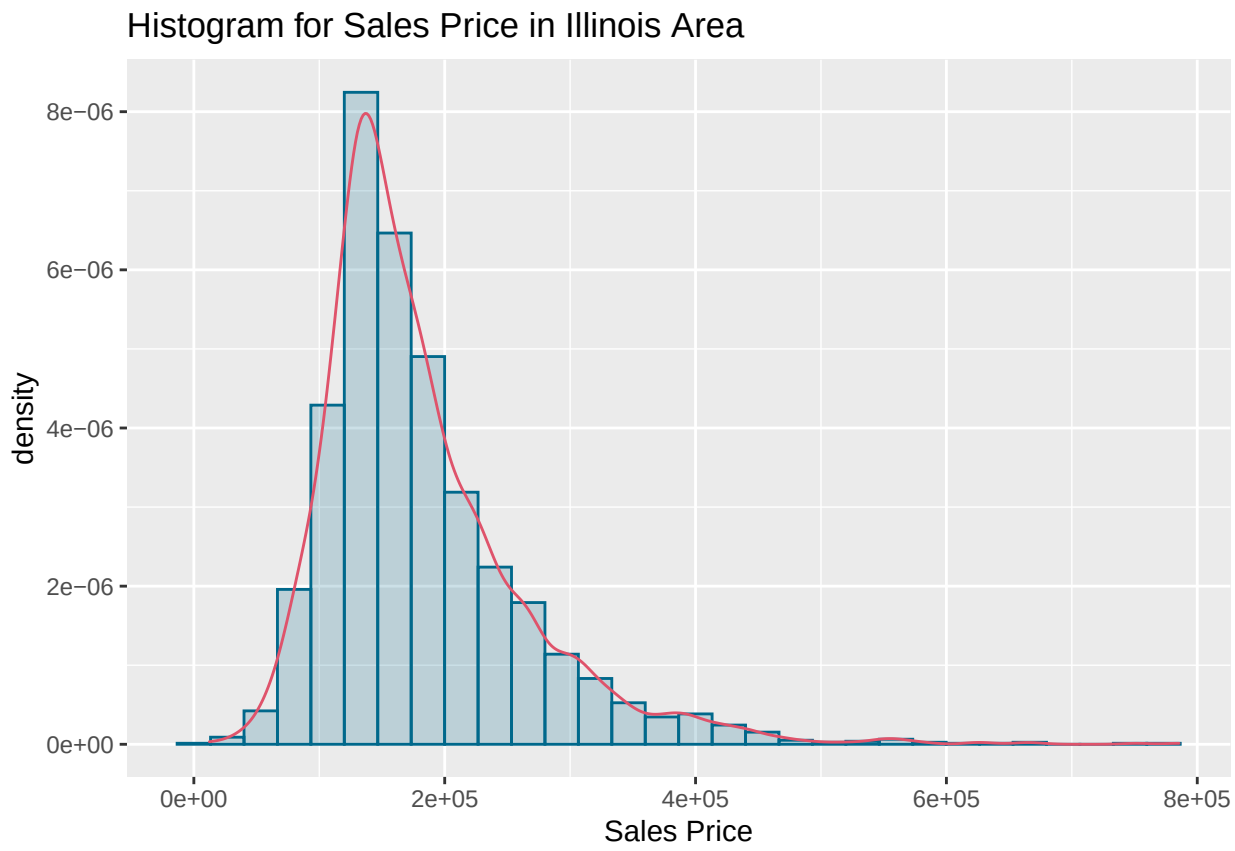
Understanding the raw data:

(a) There are 82 variables (81 independent variables — Features and 1 dependent variable — Target Variable).
(b) The data types are either integers or characters.
(c) There are 13997 missing values in our dataset.

Tidying and Exploratory Data Analysis

To get insights into the sale price we produce a distribution plot of housing price from our data. The distribution of sales price is positively skewed, where the mean(186298) is usually to the right of the median (165500).

NA's are handled by checking all the features in our dataset, and we remove columns that are having the highest number of NA's. "Alley, Pool.QC,Fence,Misc.Feature, Fireplace.Qu, and Lot.Frontage" are the features we remove from the dataset. Once we remove these columns, the rest of the NA's are handled by just omitting them. We didn't do the NA omit before removing the above-mentioned columns. Because if we had handled NA's by just removing all NA's from our original dataset, it would have affected our total number of rows leaving us with very few rows for our modeling. By handling NA's like this we only lost 6% of our data.



Feature Selection

Our data has 82 features in total. It is not advised to use all of the features in our model prediction due to the risk of overfitting. Therefore, the best thing to do is to find out the correlation between these features and "SalePrice" and remove the features which do not have a high correlation

Since we have 82 columns in our dataset, it will not be ideal to plot a correlation matrix. Here we used a manual feature selection by splitting the dataset into numeric and factor (all character columns are typecasted to factor for modeling). Then the numeric dataset is used to find out the correlation of features with respect to "SalePrice". We selected features that are having more than 0.4 value from the correlation table. An alternative option is to reduce the feature dimension using **PCA**.

After extracting these columns, we combine the numerical and categorical datasets for our modeling.

Modeling

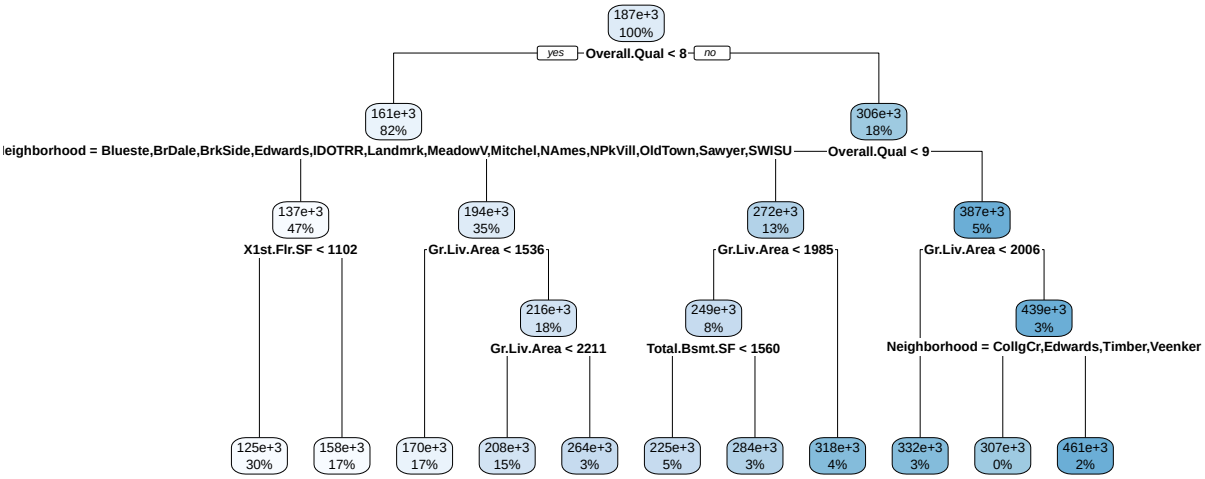
Data is partitioned to 80-20 as training and test set before applying machine learning algorithm.

Decision Tree

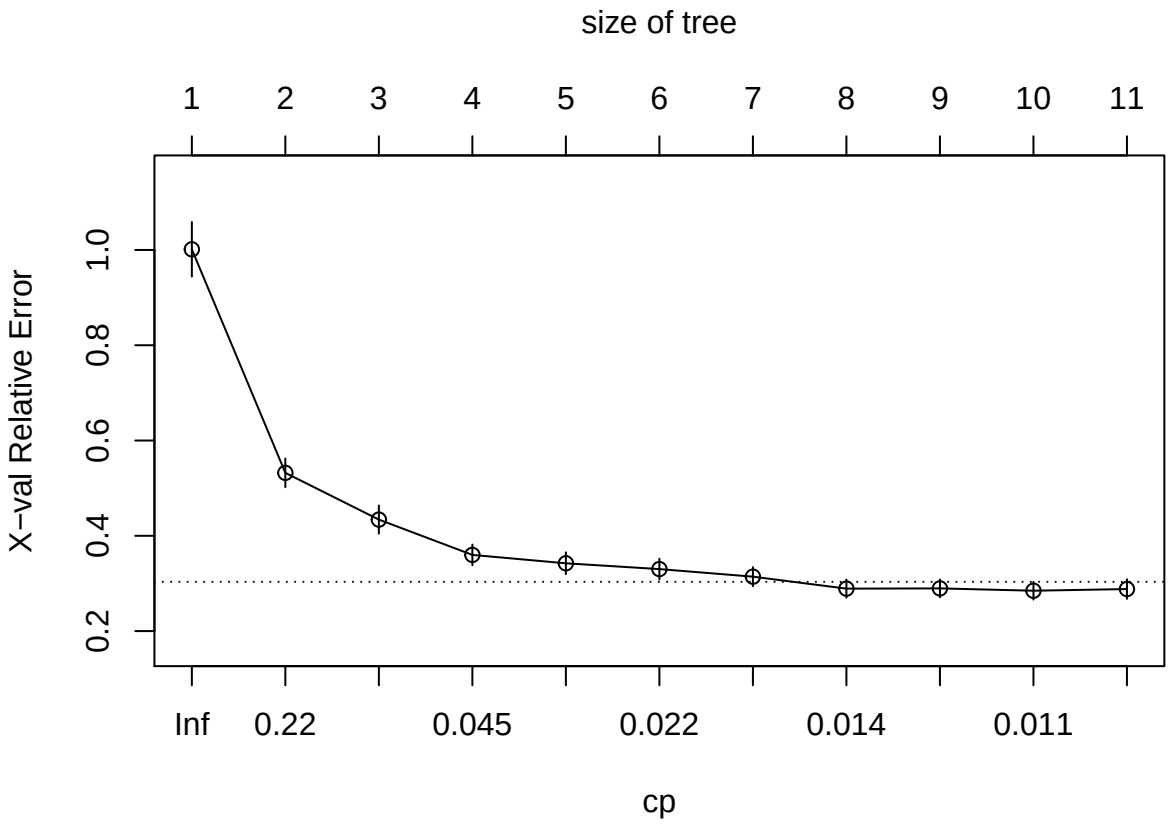
To first step in the decision tree is to build a full model with all the features. 10-fold cross-validation is applied in the decision tree algorithm. Since it is a regression tree we selected **ANOVA** as our method.

From the rpart.plot visualization below we can see how the tree is partitioned. Node 1 includes all the rows and the first split is based on **Overall.Qual**. The first split separates our dataset to a node with Overall.Qual < 8 "Yes" and a node with Overall.Qual>8 "No". The first split creates a left node with 82% and a right node with 18%.

The left node split is based on the neighborhood criteria, if the **neighborhood** falls under these categories; Blueste, BrDale, BrkSide, Edwards, IDOTRR, Landmark, MeadowV, Mitchel, NAmes, NPkVill, OldTown, Sawyer, SWISU, whereas the right node split is again based on the **Overall.Qual**. Similarly, each node is partitioned based on **Gr.Liv.Area, X1st.Flr.SF, Total.Bsmt.SF** , having a total of 21 nodes and has a depth of 5.



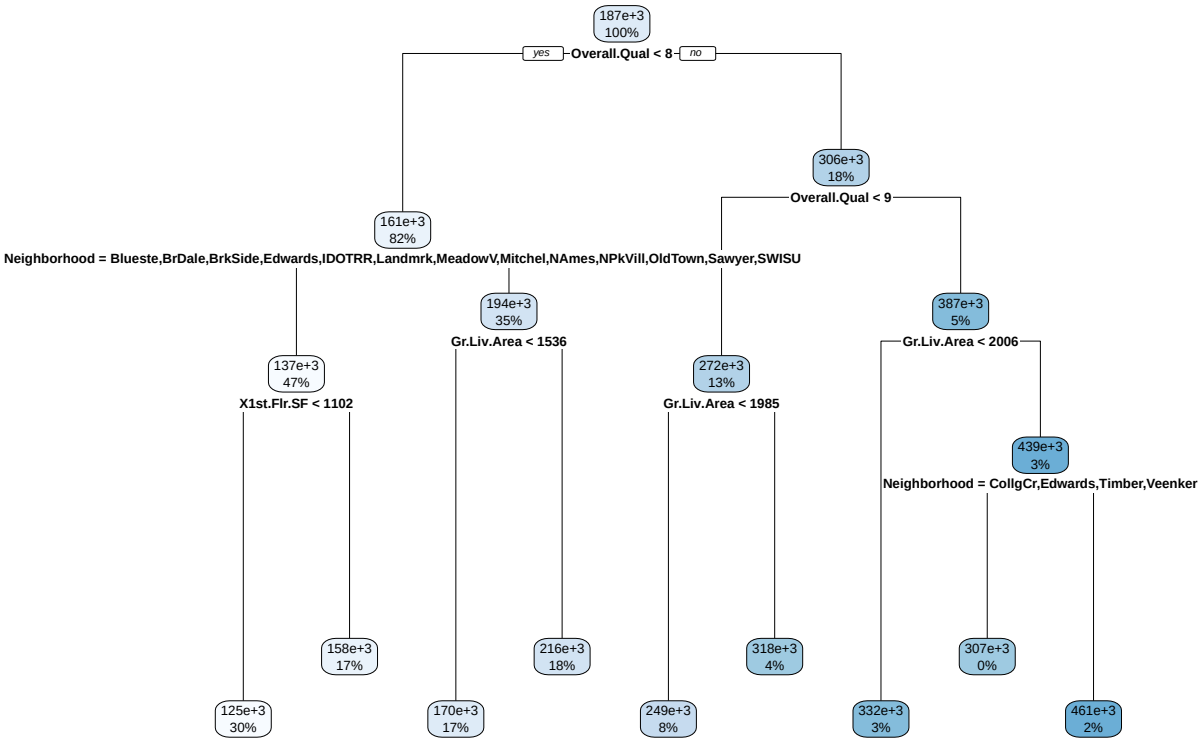
Pruning complexity parameter (cp) plot illustrating the relative cross-validation error (y-axis) for various cp values (lower x-axis). Smaller cp values lead to larger trees. Using the 1-SE rule, a tree size of 10-12 provides optimal cross-validation results. From the plot, we can see that a tree size of 10 has the lowest CP value. From the regression tree output, we get the least xerror **0.28473** that corresponds to tree size 10.

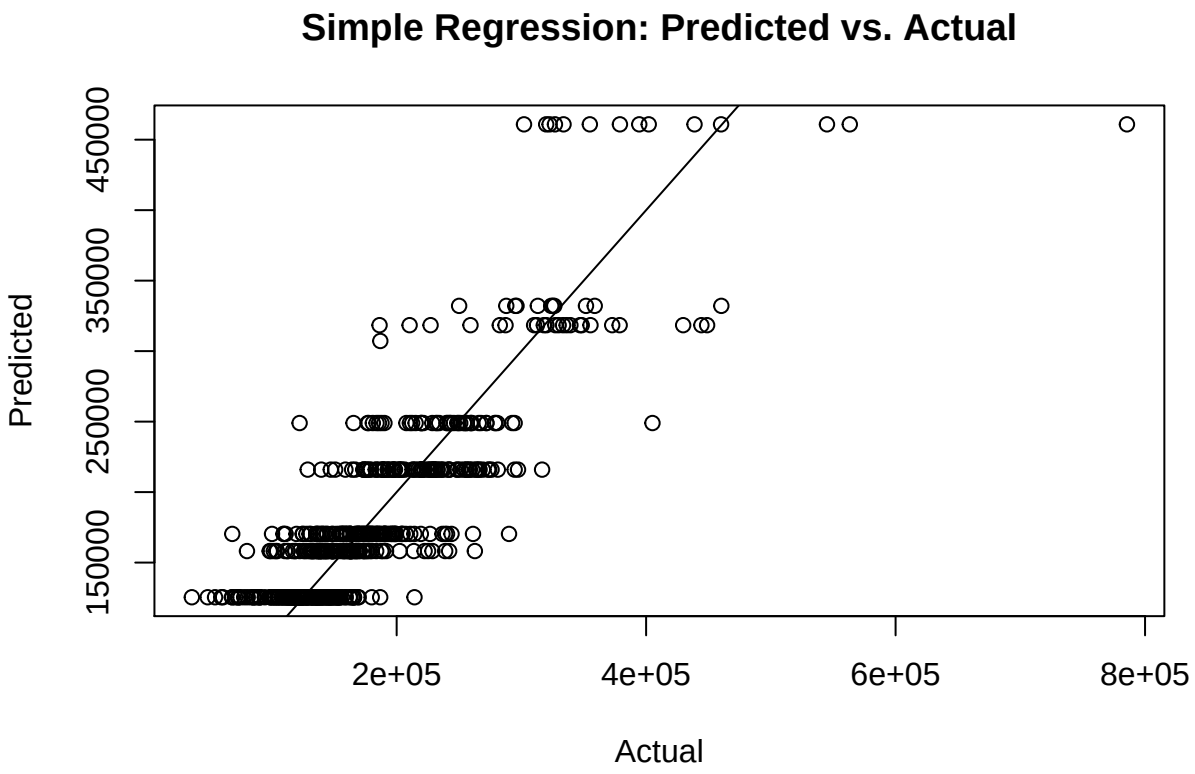


To compare the performance of the model we have chosen the loss function RMSE(Root Mean Squared Error). We will be using RMSE to explain the performance of the models throughout our project. RMSE is a mostly used error function in explaining the performance. We can compare RMSE with the standard deviation to understand how well the model can make predictions.

The RMSE of the full model is **41805.53**, which is almost half the standard deviation of the target variable **80255.26**.

The next step is to prune the tree in order to avoid overfitting and see if there is any improvement in the model. The CP table shows the relevant statistics to choose the appropriate pruning parameter. As mentioned above from the CP table, we chose the CP corresponding to the xerror 0.28473. However, from tree size 8th onward the xerror is almost similar, so we chose the CP of the 8th tree which is **0.011663** for our model.


187e+3
100%
yes
Overall.Qual < 8
no
161e+3
82%
306e+3
18%
Overall.Qual < 9
Neighborhood = Blueste,BrDale,BrkSide,Edwards,IDOTRR,Landmrk,MeadowV,Mitchel,NAmes,NPkVill,OldTown,Sawyer,SWISU
194e+3
35%
387e+3
5%
Gr.Liv.Area < 1536
Gr.Liv.Area < 2006
137e+3
47%
272e+3
13%
X1st.Flr.SF < 1102
Gr.Liv.Area < 1985
439e+3
3%
Neighborhood = CollgCr,Edwards,Timber,Veenker
158e+3
17%
216e+3
18%
318e+3
4%
307e+3
0%
125e+3
30%
170e+3
17%
249e+3
8%
332e+3
3%
461e+3
2%



The pruning process leads to an average prediction error of 42303.02 in the test data set. It is not too bad considering the standard deviation of Sales Price is 80255.26.

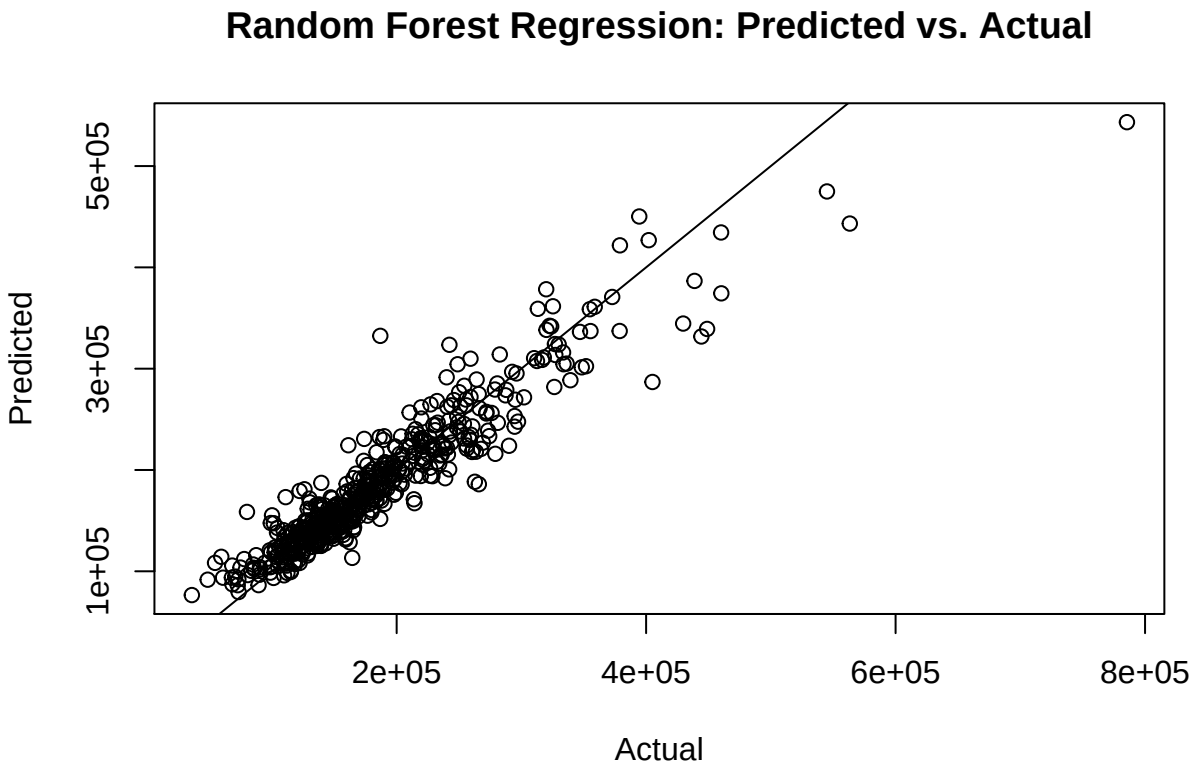
Looking at the plot of actual sales price vs predicted sales price, we can see that the variance is really high and all the predictions are not so close to its target sales price indicating a low accuracy in the model for predicting house prices.

Random Forest

Next, a different supervised learning algorithm, random forest is used to predict the **SalePrice** and understand the feature importance in predicting housing price. A random forest uses a collection of weak decision trees to greatly reduce the overfitting problem. Therefore, we should achieve better performance on the test set. The model was trained on the training set with setting the importance flag set TRUE as we wanted to analyze the most important features that contribute to the predictions of house prices. Looking at the feature importance table, we could see that features **Gr.Liv.Area, Neighborhood, Overall.Qual, BsmtFin.SF.1, Total.Bsmt.SF, X1st.Flr.SF, Garage.Area, Fireplaces, Full.Bath, and Garage.Type** are the most relevant factors that affect house prices. The model's performance on the test set is RMSE the **27595.37**, which is a great improvement over the decision tree model.

```
##                   Variables Importance
## Gr.Liv.Area     Gr.Liv.Area      47.68
## Neighborhood   Neighborhood      33.20
## Overall.Qual   Overall.Qual      26.90
## BsmtFin.SF.1   BsmtFin.SF.1      24.00
## Total.Bsmt.SF Total.Bsmt.SF      22.64
## X1st.Flr.SF     X1st.Flr.SF      20.59
## Garage.Area     Garage.Area      18.80
```

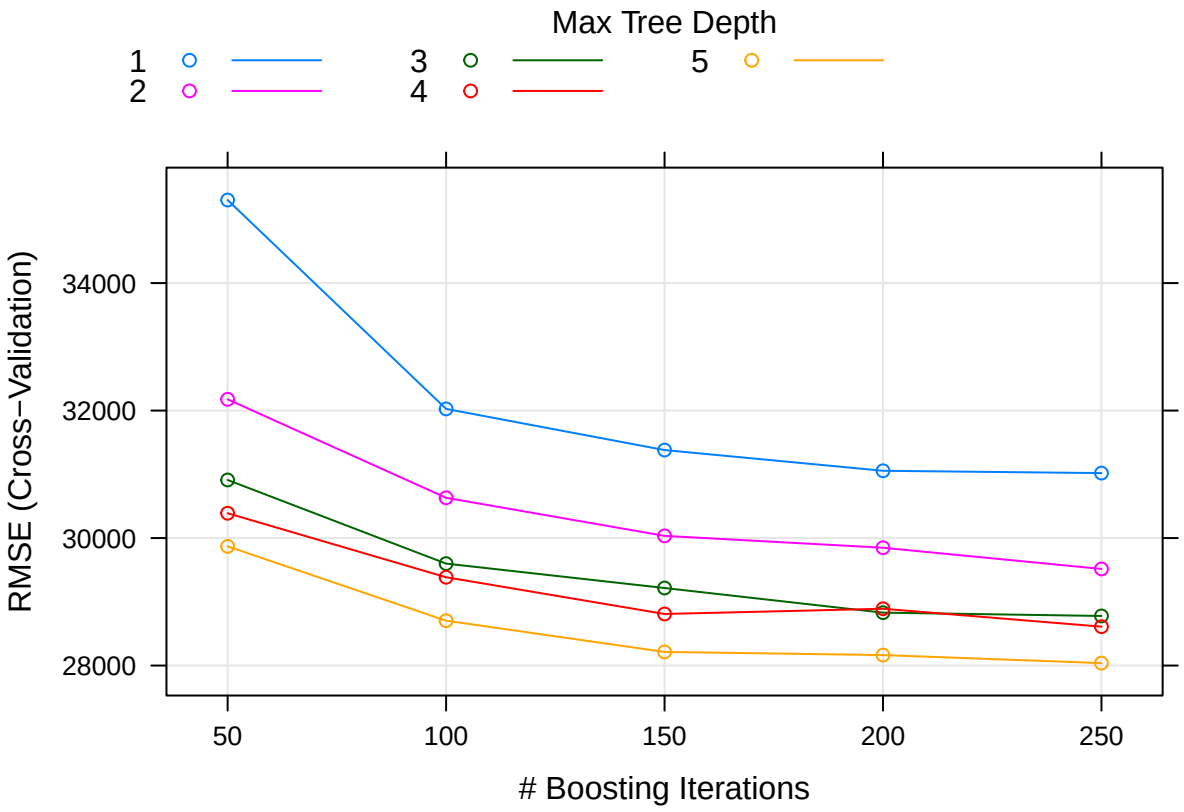
```
## Fireplaces       Fireplaces      15.21
## Full.Bath         Full.Bath      14.57
## Garage.Type     Garage.Type      14.21
```



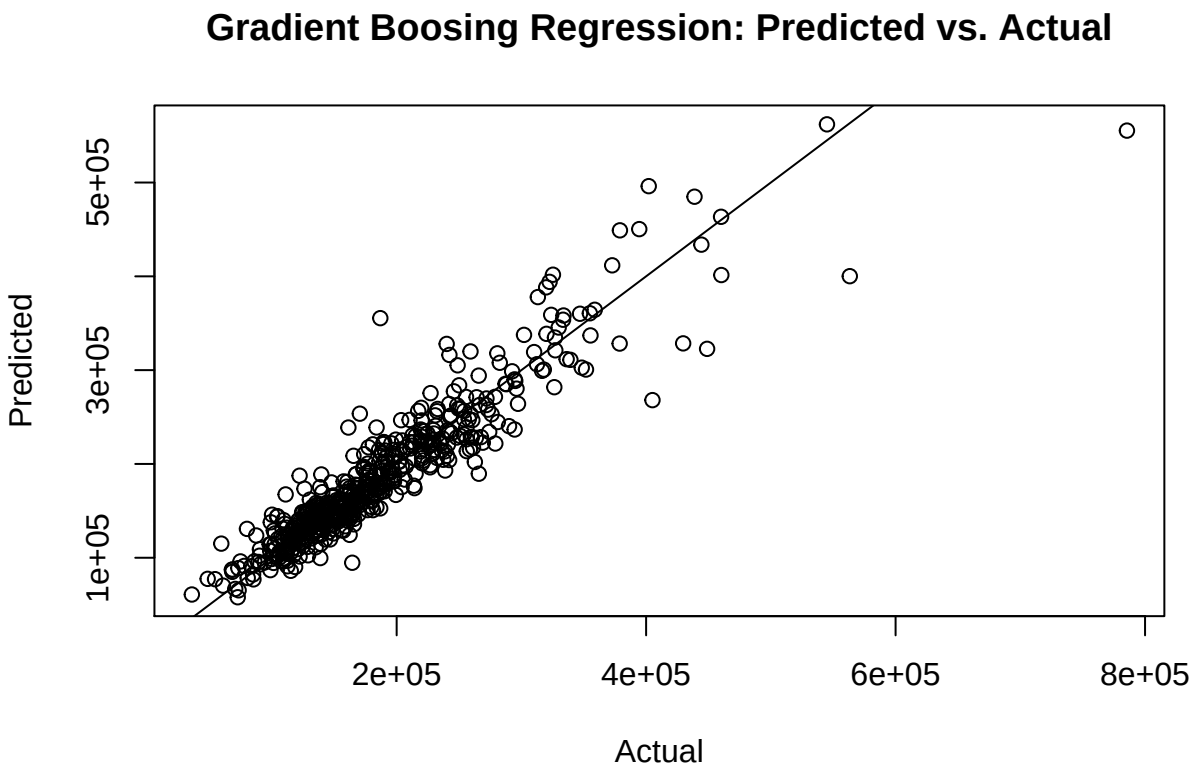
Looking at the plot of actual sales price vs predicted sales price, we can see that the variance is really low and all the predictions are really close to its target sales price indicating a great accuracy in the model for predicting house prices.

Gradient Boosting

Although we achieved great test set performance with the random forest model, we now explore a different machine learning model called gradient boosting to check if we can achieve even better accuracy in predicting the sales price. A GBM model was trained on the training set using multiple tree-depth as a hyperparameter and with a 10 fold cross-validation and achieved a test set RMSE **28259.19**. Although the model performed much better than the individual decision tree we trained. The random forest is still a better model when it comes to the test set performance.



The plot above shows the model cross-validation performance for different values of the hyperparameters "tree depth". it clearly shows that tree depth 5 outperforms all the other values.



The above target sales price vs predicted sales price also shows a good model fit and test set performance as the previous random forest model.

Conclusion

This project examined the different factors associated with the Sale Price of houses in the Illinois area using supervised learning techniques on the Illinois real estate dataset. Fairly accurate and useful models were built using regression tree, random forest, and gradient boosting. Among these three models, random forest is the best performing model in predicting the sales price.

Appendix

CODE

Data Acquisition

(a) Load Data

```
housing_data <- read.csv("HousesSoldData.csv", stringsAsFactors = TRUE)
head(housing_data)
```

```
##   Order       PID MS.SubClass MS.Zoning Lot.Frontage Lot.Area Street Alley
## 1     1 526301100          20        RL          141    31770   Pave  <NA>
## 2     2 526350040          20        RH           80    11622   Pave  <NA>
## 3     3 526351010          20        RL           81    14267   Pave  <NA>
## 4     4 526353030          20        RL           93    11160   Pave  <NA>
## 5     5 527105010          60        RL           74    13830   Pave  <NA>
```

```
## 6     6 527105030          60        RL           78     9978   Pave  <NA>
##   Lot.Shape Land.Contour Utilities Lot.Config Land.Slope Neighborhood
## 1       IR1          Lvl    AllPub     Corner        Gtl        NAmes
## 2       Reg          Lvl    AllPub     Inside        Gtl        NAmes
## 3       IR1          Lvl    AllPub     Corner        Gtl        NAmes
## 4       Reg          Lvl    AllPub     Corner        Gtl        NAmes
## 5       IR1          Lvl    AllPub     Inside        Gtl      Gilbert
## 6       IR1          Lvl    AllPub     Inside        Gtl      Gilbert
##   Condition.1 Condition.2 Bldg.Type House.Style Overall.Qual Overall.Cond
## 1        Norm        Norm      1Fam      1Story            6            5
## 2       Feedr        Norm      1Fam      1Story            5            6
## 3        Norm        Norm      1Fam      1Story            6            6
## 4        Norm        Norm      1Fam      1Story            7            5
## 5        Norm        Norm      1Fam      2Story            5            5
## 6        Norm        Norm      1Fam      2Story            6            6
##   Year.Built Year.Remod.Add Roof.Style Roof.Matl Exterior.1st Exterior.2nd
## 1       1960           1960        Hip   CompShg      BrkFace      Plywood
## 2       1961           1961      Gable   CompShg      VinylSd      VinylSd
## 3       1958           1958        Hip   CompShg      Wd Sdng      Wd Sdng
## 4       1968           1968        Hip   CompShg      BrkFace      BrkFace
## 5       1997           1998      Gable   CompShg      VinylSd      VinylSd
## 6       1998           1998      Gable   CompShg      VinylSd      VinylSd
##   Mas.Vnr.Type Mas.Vnr.Area Exter.Qual Exter.Cond Foundation Bsmt.Qual
## 1        Stone          112         TA         TA     CBlock        TA
## 2         None            0         TA         TA     CBlock        TA
## 3      BrkFace          108         TA         TA     CBlock        TA
## 4         None            0         Gd         TA     CBlock        TA
## 5         None            0         TA         TA      PConc        Gd
## 6      BrkFace           20         TA         TA      PConc        TA
##   Bsmt.Cond Bsmt.Exposure BsmtFin.Type.1 BsmtFin.SF.1 BsmtFin.Type.2
## 1        Gd            Gd            BLQ          639            Unf
## 2        TA            No            Rec          468            LwQ
## 3        TA            No            ALQ          923            Unf
## 4        TA            No            ALQ         1065            Unf
## 5        TA            No            GLQ          791            Unf
## 6        TA            No            GLQ          602            Unf
##   BsmtFin.SF.2 Bsmt.Unf.SF Total.Bsmt.SF Heating Heating.QC Central.Air
## 1            0         441          1080    GasA         Fa           Y
## 2          144         270           882    GasA         TA           Y
## 3            0         406          1329    GasA         TA           Y
## 4            0        1045          2110    GasA         Ex           Y
## 5            0         137           928    GasA         Gd           Y
## 6            0         324           926    GasA         Ex           Y
##   Electrical X1st.Flr.SF X2nd.Flr.SF Low.Qual.Fin.SF Gr.Liv.Area Bsmt.Full.Bath
## 1      SBrkr        1602           0               0        1788              1
## 2      SBrkr         898           0               0         888              0
## 3      SBrkr        1315           0               0        1327              0
## 4      SBrkr        2125           0               0        2061              1
## 5      SBrkr         855         701               0        1611              0
## 6      SBrkr         885         678               0        1553              0
##   Bsmt.Half.Bath Full.Bath Half.Bath Bedroom.AbvGr Kitchen.AbvGr Kitchen.Qual
## 1              0         1         0             3             1           TA
## 2              0         1         0             2             1           TA
## 3              0         1         1             3             1           Gd
```

```
## 4              0         2         1             3             1           Ex
## 5              0         2         1             3             1           TA
## 6              0         2         1             3             1           Gd
##   TotRms.AbvGrd Functional Fireplaces Fireplace.Qu Garage.Type Garage.Yr.Blt
## 1             7        Typ          2          Gd      Attchd          1960
## 2             5        Typ          0        <NA>      Attchd          1961
## 3             6        Typ          0        <NA>      Attchd          1958
## 4             8        Typ          2          TA      Attchd          1968
## 5             6        Typ          1          TA      Attchd          1997
## 6             7        Typ          1          Gd      Attchd          1998
##   Garage.Finish Garage.Cars Garage.Area Garage.Qual Garage.Cond Paved.Drive
## 1           Fin           2         528          TA          TA           P
## 2           Unf           1         730          TA          TA           Y
## 3           Unf           1         312          TA          TA           Y
## 4           Fin           2         522          TA          TA           Y
## 5           Fin           2         482          TA          TA           Y
## 6           Fin           2         470          TA          TA           Y
##   Wood.Deck.SF Open.Porch.SF Enclosed.Porch X3Ssn.Porch Screen.Porch Pool.Area
## 1          210            62              0           0           0         0
## 2          140             0              0           0         120         0
## 3          393            36              0           0           0         0
## 4            0             0              0           0           0         0
## 5          212            34              0           0           0         0
## 6          360            36              0           0           0         0
##   Pool.QC Fence Misc.Feature Misc.Val Mo.Sold Yr.Sold Sale.Type Sale.Condition
## 1    <NA>  <NA>         <NA>        0       5    2010       WD          Normal
## 2    <NA> MnPrv         <NA>        0       6    2010       WD          Normal
## 3    <NA>  <NA>         Gar2    12500       6    2010       WD          Normal
## 4    <NA>  <NA>         <NA>        0       4    2010       WD          Normal
## 5    <NA> MnPrv         <NA>        0       3    2010       WD          Normal
## 6    <NA>  <NA>         <NA>        0       6    2010       WD          Normal
##   SalePrice
## 1    209200
## 2    107700
## 3    163600
## 4    257900
## 5    184400
## 6    208700
```

(b) Understanding the datset.

```
str(housing_data)
```

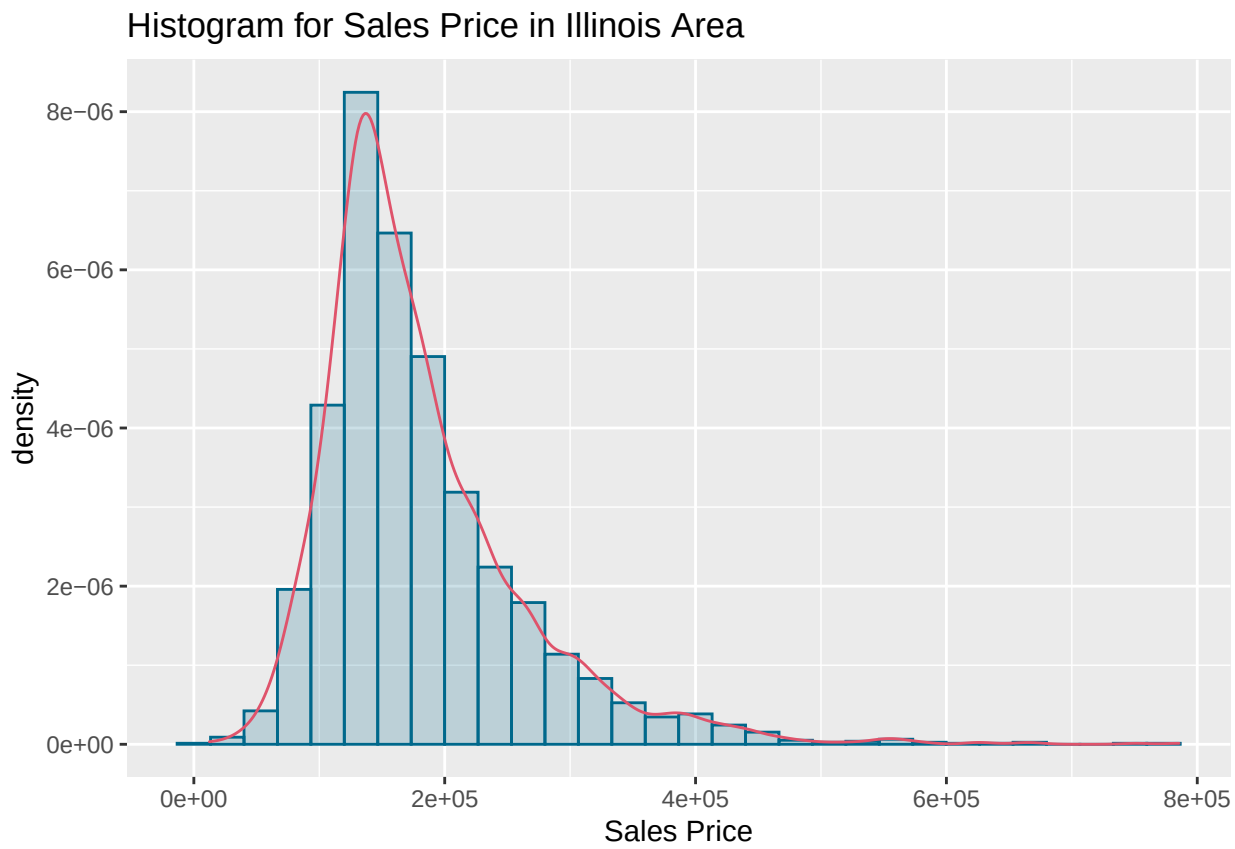
```
## 'data.frame':    2930 obs. of  82 variables:
##  $ Order          : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ PID            : int  526301100 526350040 526351010 526353030 527105010 527105030 527127150 52714
##  $ MS.SubClass    : int  20 20 20 20 60 60 120 120 120 60 ...
##  $ MS.Zoning      : Factor w/ 7 levels "A (agr)","C (all)",..: 6 5 6 6 6 6 6 6 6 6 ...
##  $ Lot.Frontage   : int  141 80 81 93 74 78 41 43 39 60 ...
##  $ Lot.Area       : int  31770 11622 14267 11160 13830 9978 4920 5005 5389 7500 ...
##  $ Street         : Factor w/ 2 levels "Grvl","Pave": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Alley          : Factor w/ 2 levels "Grvl","Pave": NA NA NA NA NA NA NA NA NA NA ...
##  $ Lot.Shape      : Factor w/ 4 levels "IR1","IR2","IR3",..: 1 4 1 4 1 1 4 1 1 4 ...
##  $ Land.Contour   : Factor w/ 4 levels "Bnk","HLS","Low",..: 4 4 4 4 4 4 4 2 4 4 ...
##  $ Utilities      : Factor w/ 3 levels "AllPub","NoSeWa",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
##  $ Lot.Config     : Factor w/ 5 levels "Corner","CulDSac",..: 1 5 1 1 5 5 5 5 5 5 ...
##  $ Land.Slope     : Factor w/ 3 levels "Gtl","Mod","Sev": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Neighborhood   : Factor w/ 28 levels "Blmngtn","Blueste",..: 16 16 16 16 9 9 25 25 25 9 ...
##  $ Condition.1    : Factor w/ 9 levels "Artery","Feedr",..: 3 2 3 3 3 3 3 3 3 3 ...
##  $ Condition.2    : Factor w/ 8 levels "Artery","Feedr",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ Bldg.Type      : Factor w/ 5 levels "1Fam","2fmCon",..: 1 1 1 1 1 1 5 5 5 1 ...
##  $ House.Style    : Factor w/ 8 levels "1.5Fin","1.5Unf",..: 3 3 3 3 6 6 3 3 3 6 ...
##  $ Overall.Qual   : int  6 5 6 7 5 6 8 8 8 7 ...
##  $ Overall.Cond   : int  5 6 6 5 5 6 5 5 5 5 ...
##  $ Year.Built     : int  1960 1961 1958 1968 1997 1998 2001 1992 1995 1999 ...
##  $ Year.Remod.Add : int  1960 1961 1958 1968 1998 1998 2001 1992 1996 1999 ...
##  $ Roof.Style     : Factor w/ 6 levels "Flat","Gable",..: 4 2 4 4 2 2 2 2 2 2 ...
##  $ Roof.Matl      : Factor w/ 8 levels "ClyTile","CompShg",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Exterior.1st   : Factor w/ 16 levels "AsbShng","AsphShn",..: 4 14 15 4 14 14 6 7 6 14 ...
##  $ Exterior.2nd   : Factor w/ 17 levels "AsbShng","AsphShn",..: 11 15 16 4 15 15 6 7 6 15 ...
##  $ Mas.Vnr.Type   : Factor w/ 5 levels "BrkCmn","BrkFace",..: 5 4 2 4 4 2 4 4 4 4 ...
##  $ Mas.Vnr.Area   : int  112 0 108 0 0 20 0 0 0 0 ...
##  $ Exter.Qual     : Factor w/ 4 levels "Ex","Fa","Gd",..: 4 4 4 3 4 4 3 3 3 4 ...
##  $ Exter.Cond     : Factor w/ 5 levels "Ex","Fa","Gd",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ Foundation     : Factor w/ 6 levels "BrkTil","CBlock",..: 2 2 2 2 3 3 3 3 3 3 ...
##  $ Bsmt.Qual      : Factor w/ 5 levels "Ex","Fa","Gd",..: 5 5 5 5 3 5 3 3 3 5 ...
##  $ Bsmt.Cond      : Factor w/ 5 levels "Ex","Fa","Gd",..: 3 5 5 5 5 5 5 5 5 5 ...
##  $ Bsmt.Exposure  : Factor w/ 4 levels "Av","Gd","Mn",..: 2 4 4 4 4 4 3 4 4 4 ...
##  $ BsmtFin.Type.1 : Factor w/ 6 levels "ALQ","BLQ","GLQ",..: 2 5 1 1 3 3 3 1 3 6 ...
##  $ BsmtFin.SF.1   : int  639 468 923 1065 791 602 616 263 1180 0 ...
##  $ BsmtFin.Type.2 : Factor w/ 6 levels "ALQ","BLQ","GLQ",..: 6 4 6 6 6 6 6 6 6 6 ...
##  $ BsmtFin.SF.2   : int  0 144 0 0 0 0 0 0 0 0 ...
##  $ Bsmt.Unf.SF    : int  441 270 406 1045 137 324 722 1017 415 994 ...
##  $ Total.Bsmt.SF  : int  1080 882 1329 2110 928 926 1338 1280 1595 994 ...
##  $ Heating        : Factor w/ 6 levels "Floor","GasA",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Heating.QC     : Factor w/ 5 levels "Ex","Fa","Gd",..: 2 5 5 1 3 1 1 1 1 3 ...
##  $ Central.Air    : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Electrical     : Factor w/ 5 levels "FuseA","FuseF",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ X1st.Flr.SF    : int  1602 898 1315 2125 855 885 1387 1334 1807 1138 ...
##  $ X2nd.Flr.SF    : int  0 0 0 0 701 678 0 0 0 776 ...
##  $ Low.Qual.Fin.SF: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Gr.Liv.Area    : int  1788 888 1327 2061 1611 1553 1165 1320 1792 1842 ...
##  $ Bsmt.Full.Bath : int  1 0 0 1 0 0 1 0 1 0 ...
##  $ Bsmt.Half.Bath : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Full.Bath      : int  1 1 1 2 2 2 2 2 2 2 ...
##  $ Half.Bath      : int  0 0 1 1 1 1 0 0 0 1 ...
##  $ Bedroom.AbvGr  : int  3 2 3 3 3 3 2 2 2 3 ...
##  $ Kitchen.AbvGr  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Kitchen.Qual   : Factor w/ 5 levels "Ex","Fa","Gd",..: 5 5 3 1 5 3 3 3 3 3 ...
##  $ TotRms.AbvGrd  : int  7 5 6 8 6 7 6 5 5 7 ...
##  $ Functional     : Factor w/ 8 levels "Maj1","Maj2",..: 8 8 8 8 8 8 8 8 8 8 ...
##  $ Fireplaces     : int  2 0 0 2 1 1 0 0 1 1 ...
##  $ Fireplace.Qu   : Factor w/ 5 levels "Ex","Fa","Gd",..: 3 NA NA 5 5 3 NA NA 5 5 ...
##  $ Garage.Type    : Factor w/ 6 levels "2Types","Attchd",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Garage.Yr.Blt  : int  1960 1961 1958 1968 1997 1998 2001 1992 1995 1999 ...
##  $ Garage.Finish  : Factor w/ 3 levels "Fin","RFn","Unf": 1 3 3 1 1 1 1 2 2 1 ...
##  $ Garage.Cars    : int  2 1 1 2 2 2 2 2 2 2 ...
##  $ Garage.Area    : int  528 730 312 522 482 470 582 506 608 442 ...
##  $ Garage.Qual    : Factor w/ 5 levels "Ex","Fa","Gd",..: 5 5 5 5 5 5 5 5 5 5 ...
```

```
##  $ Garage.Cond    : Factor w/ 5 levels "Ex","Fa","Gd",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ Paved.Drive    : Factor w/ 3 levels "N","P","Y": 2 3 3 3 3 3 3 3 3 3 ...
##  $ Wood.Deck.SF   : int  210 140 393 0 212 360 0 0 237 140 ...
##  $ Open.Porch.SF  : int  62 0 36 0 34 36 0 82 152 60 ...
##  $ Enclosed.Porch : int  0 0 0 0 0 0 170 0 0 0 ...
##  $ X3Ssn.Porch    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Screen.Porch   : int  0 120 0 0 0 0 0 144 0 0 ...
##  $ Pool.Area      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Pool.QC        : Factor w/ 4 levels "Ex","Fa","Gd",..: NA NA NA NA NA NA NA NA NA NA ...
##  $ Fence          : Factor w/ 4 levels "GdPrv","GdWo",..: NA 3 NA NA 3 NA NA NA NA NA ...
##  $ Misc.Feature   : Factor w/ 5 levels "Elev","Gar2",..: NA NA 2 NA NA NA NA NA NA NA ...
##  $ Misc.Val       : int  0 0 12500 0 0 0 0 0 0 0 ...
##  $ Mo.Sold        : int  5 6 6 4 3 6 4 1 3 6 ...
##  $ Yr.Sold        : int  2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...
##  $ Sale.Type      : Factor w/ 10 levels "COD","Con","ConLD",..: 10 10 10 10 10 10 10 10 10 10 ...
##  $ Sale.Condition : Factor w/ 6 levels "Abnorml","AdjLand",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ SalePrice      : int  209200 107700 163600 257900 184400 208700 213700 193600 241700 191200 ...
```

Tidying Data and EDA

```
ggplot(data=housing_data, aes(housing_data$SalePrice)) +
  geom_histogram(aes(y =..density..),
                 col="deepskyblue4",
                 fill="deepskyblue4",
                 alpha=.2) +
  geom_density(col=2)+
  labs(title = "Histogram for Sales Price in Illinois Area", x= "Sales Price")
```



(c) Dropping the columns with highest no of NA's

```
names(which(colSums(is.na(housing_data)) > 0))
```

```
##  [1] "Lot.Frontage"    "Alley"           "Mas.Vnr.Type"    "Mas.Vnr.Area"
##  [5] "Bsmt.Qual"       "Bsmt.Cond"       "Bsmt.Exposure"   "BsmtFin.Type.1"
##  [9] "BsmtFin.SF.1"    "BsmtFin.Type.2"  "BsmtFin.SF.2"    "Bsmt.Unf.SF"
## [13] "Total.Bsmt.SF"   "Electrical"      "Bsmt.Full.Bath"  "Bsmt.Half.Bath"
## [17] "Fireplace.Qu"    "Garage.Type"     "Garage.Yr.Blt"   "Garage.Finish"
## [21] "Garage.Cars"     "Garage.Area"     "Garage.Qual"     "Garage.Cond"
## [25] "Pool.QC"         "Fence"           "Misc.Feature"
```

```
housing_data <- housing_data %>% select(-c(Alley,Pool.QC,Fence,Misc.Feature,Fireplace.Qu,Lot.Frontage))
#head(housing_data)
```

(d) Removing NA's

```
sum(is.na(housing_data))
```

```
## [1] 1254
```

```
housing_data<- na.omit(housing_data)
nrow(housing_data)
```

```
## [1] 2678
```

(e) split char and numeric

```
housing_numeric<- select_if(housing_data, is.numeric)
housing_categorical <- select_if(housing_data, is.factor)
```

(f) finding the correlation of numeric columns

```
cor_housing<-as.data.frame(cor(housing_numeric[, -39], housing_numeric$SalePrice))
corr_columns <- row.names(subset(cor_housing, V1 >=0.4))
corr_columns
```

```
##  [1] "Overall.Qual"   "Year.Built"     "Year.Remod.Add" "Mas.Vnr.Area"
##  [5] "BsmtFin.SF.1"   "Total.Bsmt.SF"  "X1st.Flr.SF"    "Gr.Liv.Area"
##  [9] "Full.Bath"      "TotRms.AbvGrd"  "Fireplaces"     "Garage.Yr.Blt"
## [13] "Garage.Cars"    "Garage.Area"    "SalePrice"
```

(g) extract the above highly correlated columns

```
housing_numeric<- housing_numeric[, corr_columns]
#head(housing_numeric)
```

(h) Combine both character and numerical dataset

```
housing_data <- cbind(housing_numeric, housing_categorical)
#housing_data
nrow(housing_data)
```

```
## [1] 2678
```

```
ncol(housing_data)
```

```
## [1] 53
```

```
sum(is.na(housing_data))
```

```
## [1] 0
```

Modeling

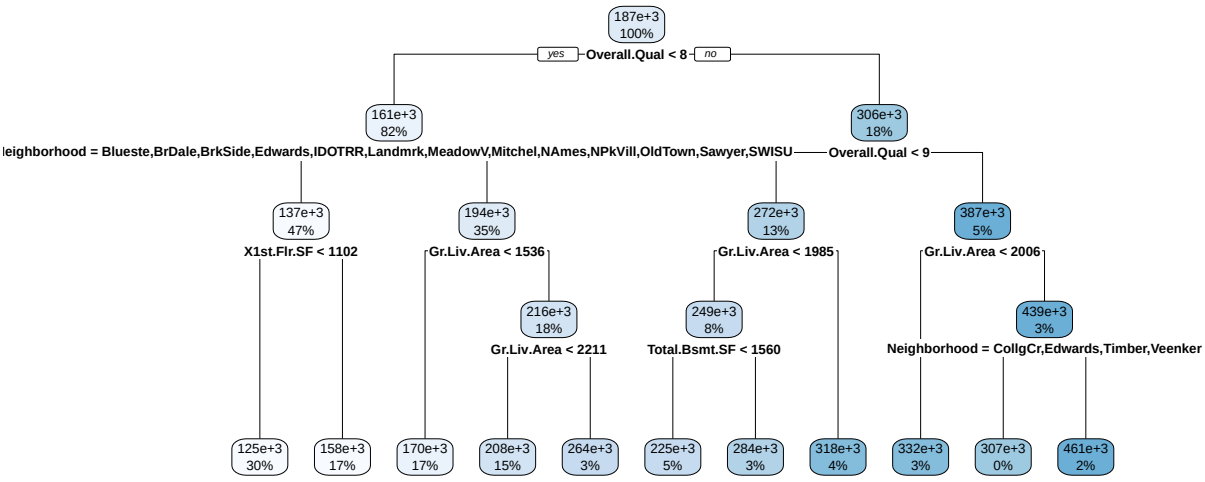
Splitting data set train and test using CV

```
set.seed(42)

partition <- createDataPartition(y= housing_data$SalePrice, p =0.80, list = FALSE)
housing_train <- housing_data[partition,]
housing_test <- housing_data[-partition,]
nrow(housing_train)/ nrow(housing_data) *100
```

```
## [1] 80.05975
```

```
set.seed(123)
housing_price <- rpart(formula = SalePrice ~ .,
                       data = housing_train,
                       method = "anova",  # regression)
                       xval = 10  # 10-fold cross-validation
                       )
rpart.plot(housing_price, yesno = TRUE)
```

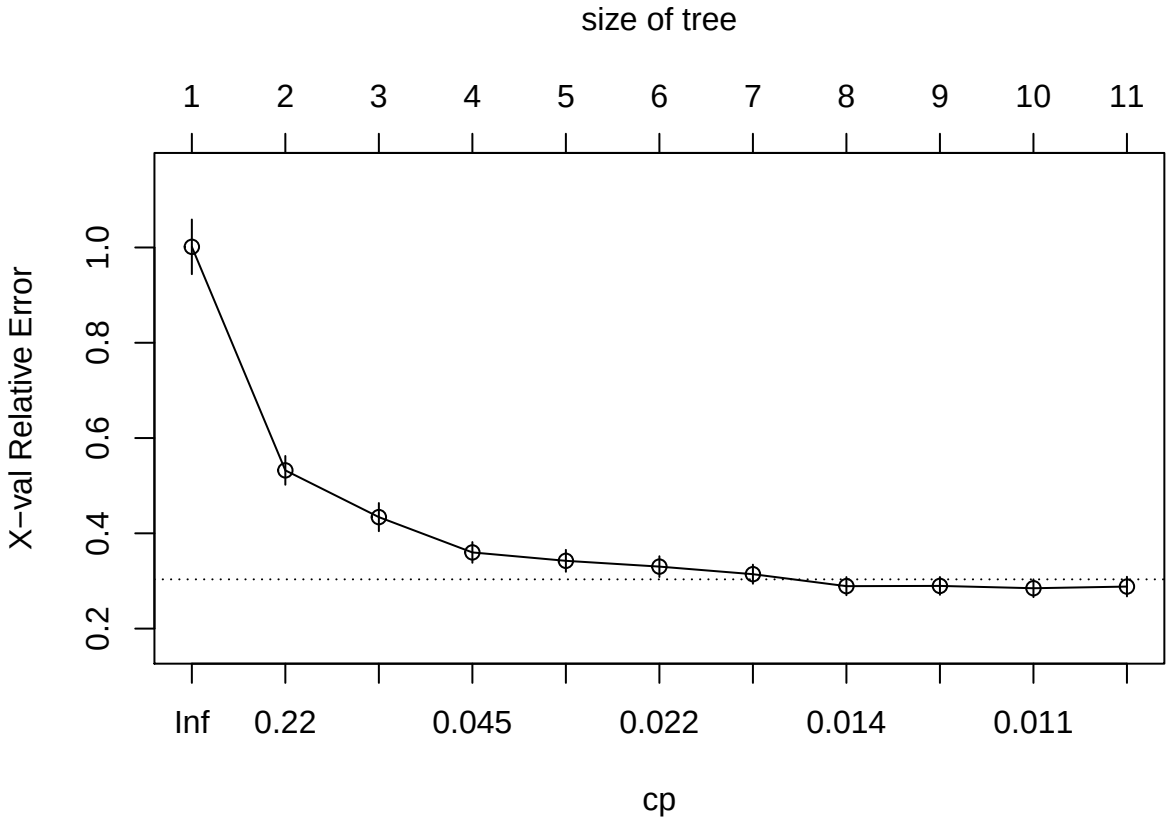


Decision Tree

```
printcp(housing_price)
```

```
##
## Regression tree:
## rpart(formula = SalePrice ~ ., data = housing_train, method = "anova",
##     xval = 10)
##
## Variables actually used in tree construction:
## [1] Gr.Liv.Area   Neighborhood  Overall.Qual  Total.Bsmt.SF X1st.Flr.SF
##
## Root node error: 1.3926e+13/2144 = 6495199497
##
## n= 2144
##
##          CP nsplit rel error  xerror     xstd
## 1  0.470697      0   1.00000 1.00135 0.057366
## 2  0.099626      1   0.52930 0.53212 0.030162
## 3  0.074212      2   0.42968 0.43400 0.029564
## 4  0.027869      3   0.35546 0.35993 0.021718
## 5  0.022874      4   0.32760 0.34227 0.023012
## 6  0.020846      5   0.30472 0.33010 0.021791
## 7  0.018009      6   0.28388 0.31421 0.020003
## 8  0.011663      7   0.26587 0.28907 0.019030
## 9  0.010946      8   0.25420 0.28960 0.018669
## 10 0.010824      9   0.24326 0.28473 0.018649
## 11 0.010000     10   0.23243 0.28820 0.020592
```

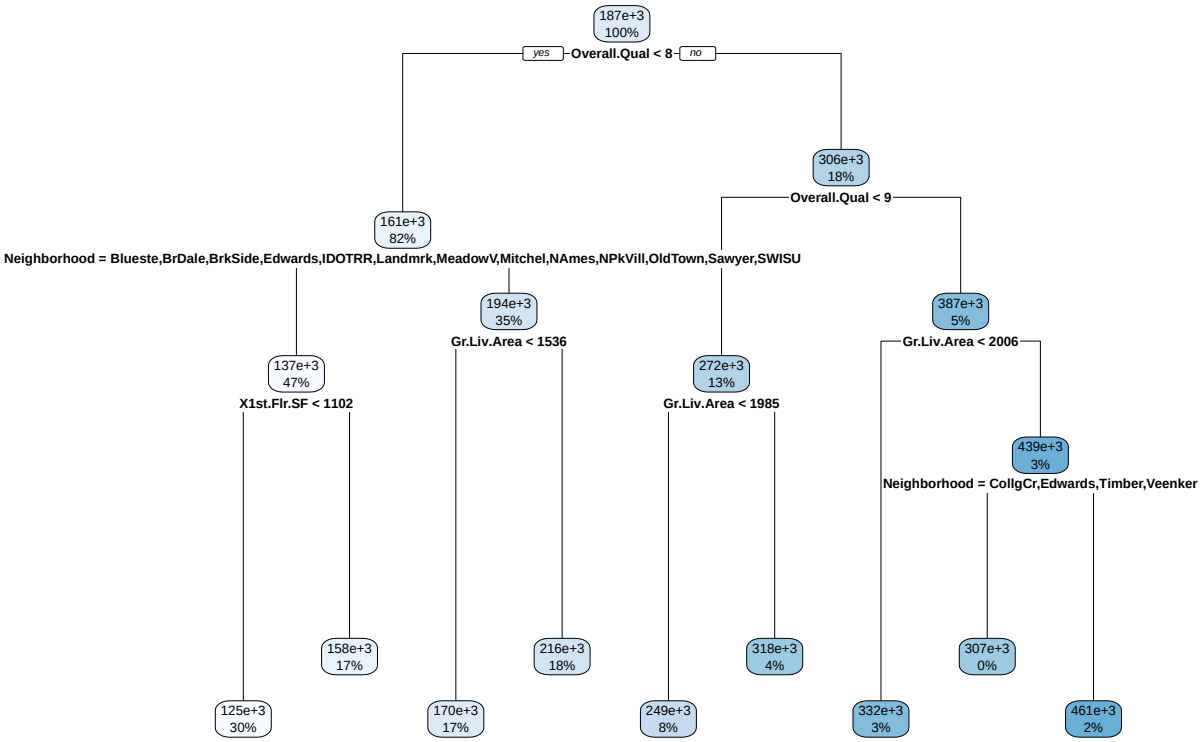
```
plotcp(housing_price)
```




```
housing_price$cptable[which.min(housing_price$cptable[,"xerror"]),"CP"]
```

```
## [1] 0.01082439
```

```
housingPrune_Anova <- prune(housing_price,
                        cp = 0.011663)
rpart.plot(housingPrune_Anova, yesno = TRUE)
```




```
housingPrice_pred <- predict( housingPrune_Anova,housing_test, type= "vector")

housingPred_RMSE <- RMSE(pred = housingPrice_pred,
                         obs = housing_test$SalePrice)
housingPred_RMSE
```

```
## [1] 42303.02
```

```
sd(housing_data$SalePrice)
```

```
## [1] 80255.26
```

```
housing_rf <- randomForest(SalePrice ~ ., data = housing_train, importance = TRUE)
housing_rf
```

Random Forest

```
##
## Call:
##  randomForest(formula = SalePrice ~ ., data = housing_train, importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 17
##
##           Mean of squared residuals: 813611545
##                     % Var explained: 87.47
```

```r
importance    <- importance(housing_rf)
varImportance <- data.frame(Variables = row.names(importance),
                            Importance = round(importance[ ,'%IncMSE'],2))
head(varImportance[order(-varImportance$Importance),],10)
```

```
##                   Variables Importance
## Gr.Liv.Area     Gr.Liv.Area      47.68
## Neighborhood   Neighborhood      33.20
## Overall.Qual   Overall.Qual      26.90
## BsmtFin.SF.1   BsmtFin.SF.1      24.00
## Total.Bsmt.SF Total.Bsmt.SF      22.64
## X1st.Flr.SF     X1st.Flr.SF      20.59
## Garage.Area     Garage.Area      18.80
## Fireplaces       Fireplaces      15.21
## Full.Bath         Full.Bath      14.57
## Garage.Type     Garage.Type      14.21
```

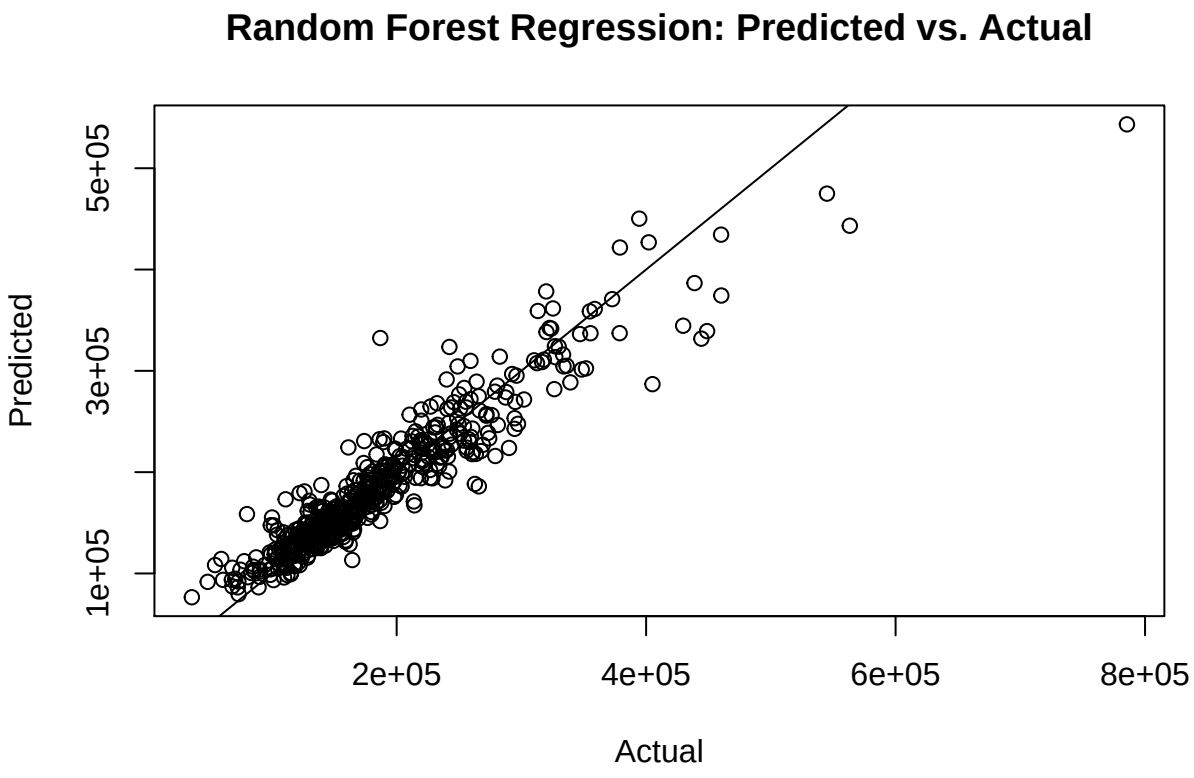
```r
housingRF_pred <- predict(housing_rf, housing_test, type = "response")
summary(housingRF_pred)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   76521  138813  166900  185318  219649  543317
```

```r
housingPredRF_RMSE <- RMSE(pred = housingRF_pred,
                           obs = housing_test$SalePrice)
housingPredRF_RMSE
```

```
## [1] 27595.37
```

```r
plot(housing_test$SalePrice, housingRF_pred,
     main = "Random Forest Regression: Predicted vs. Actual",
     xlab = "Actual",
     ylab = "Predicted")
abline(0,1)
```




```
sum(is.na(housing_train))
```

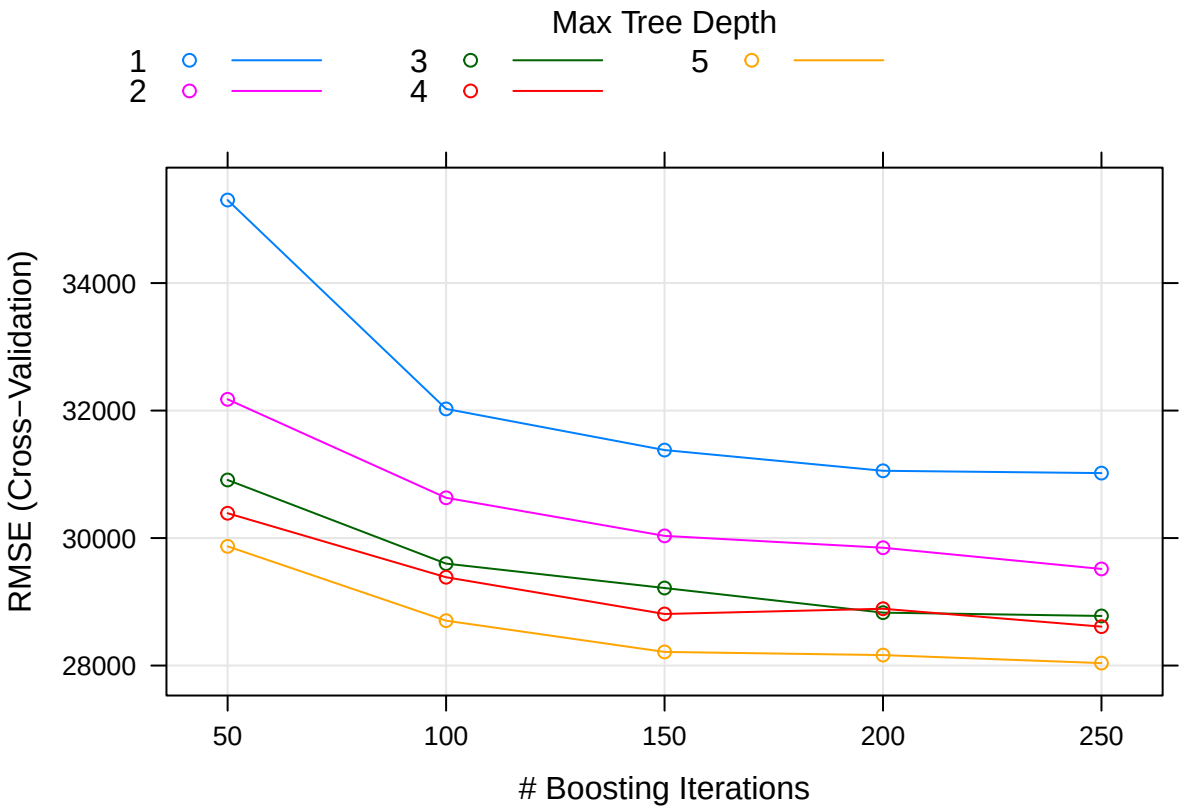
Gradient Boosting

```
## [1] 0
```

```
housing_gbm <- train(SalePrice ~ .,
                     data = housing_train,
                     method = "gbm",
                     tuneLength = 5,  # choose up to 5 combinations of tuning parameters
                     metric = "RMSE",
                     trControl = trainControl(
                       method = "cv",  # k-fold cross validation
                       number = 10,  # 10 folds
                       savePredictions = "final",
                       # verboseIter = FALSE,
                       # returnData = TRUE
                     ),
                     verbose=FALSE

                     )
```

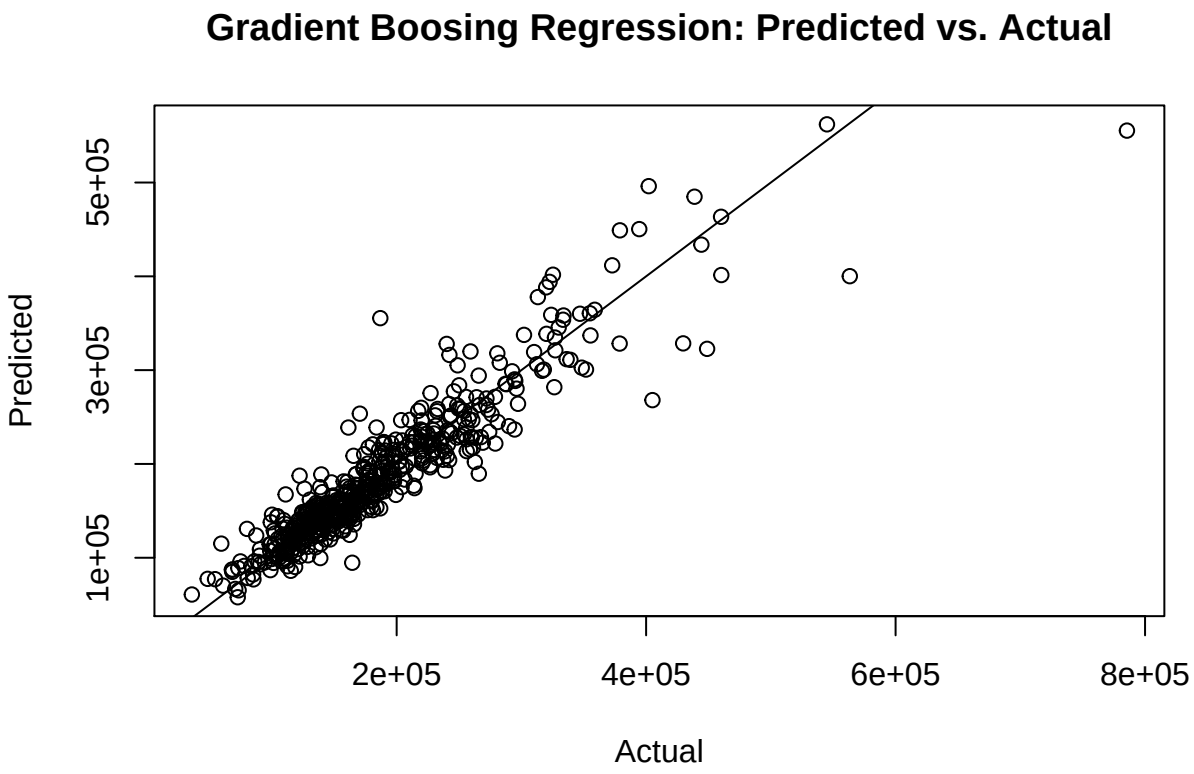
```
plot(housing_gbm)
```




```
housinggbm_pred <- predict(housing_gbm, housing_test, type = "raw")
housingPredGBM_RMSE <- RMSE(pred = housinggbm_pred,
                            obs = housing_test$SalePrice)
housingPredGBM_RMSE
```

```
## [1] 28619.38
```

```
plot(housing_test$SalePrice, housinggbm_pred,
     main = "Gradient Boosing Regression: Predicted vs. Actual",
     xlab = "Actual",
     ylab = "Predicted")
abline(0,1)
```



References

https://medium.com/analytics-vidhya/kaggle-house-prices-prediction-with-linear-regression-and-gradient boosting-c5694d9c6df4

https://towardsdatascience.com/house-prices-prediction-using-deep-learning-dea265cc3154

https://www.datacamp.com/community/tutorials/decision-trees-R