**Project**

**“House Price Prediction for King County Seattle”**

**Prof-John Russo**

**(Data Mining and Prd. Analytic-51)**

Submitted By

Radhika Kasu

Email\_Id- rkasu@fitchburgstate.edu

Reshma Kapadnis

Email\_Id- rkapadnis@student.fitchburgstate.edu

Lina MI

Email\_id- lmi@student.fitchburgstate.edu

# Introduction

In this rapid developing economic environment, real estate properties are becoming more attractive option for safe investment especially for a common person, compared with other investment alternates such as stock and fund. Real estate has property of keeping its value from decline quickly in the worse economic situation and increasing its value in the time of economic prosperity economic environment, so the investors will not lose their investment rapidly in bad economy and have good rewards in good economy.

Buying or selling a house, however, is a huge decision which can impact personal or family’s finances and future plans in critical way. When investing on real estate, what the investors concern most is the value of real estate. The price of real estate, however, is influenced by many factors and hard to estimate. Predicting the value of real estate precisely and identifying the factors that affect the value of real estate correctly could provide investors with deep insight on prospering house and help investors avoid overpaying for the candidate properties. So it is important to develop practical data mining models to give valuable estimation on the real estate market and identify the influential factors on the value of each property based on the available data. The fact that many house were transacted at the prices very close to the one estimated by some real estate websites such as Zillow and Red fin shows that accurate estimation on the house could give both seller and buyer a reliable reference for their house transactions.

In this project, random forest algorithm is chosen to analyze the dataset which contains sale prices and features of houses sold between May 2014 and May 2015 for King County, including Seattle, and then give prediction on the prices of houses and identify the important factors that influence the values of properties. The test result shows that the model is effective on identifying the influential factors and providing good estimation on the value of properties.

# Research goals and Data preprocessing

## Research Goals

In order to provide real estate investors who have little knowledge of real estate with profound insight on properties in the area of Kings County, one of goals of this research is to answer the question: what are most important features that determine the value of properties in Kings County? The other research goal concerns the question of what are the fare prices of houses in Kings County in the period from May 2014 to May 2015 under the given market situation? Answering this question successfully could provide a valuable reference to investors, help investors avoid overpaying or identify the properties with potential increase of value.

## Data Preprocessing

### Data prescription

The dataset used in this project contains 21613 observations of houses sold between May 2014 and May 2015 in area of King County, including Seattle. In each observation, 18 features of house, house ID, sale price and sale date are included. The variables in the dataset are described in the following table.

|  |  |  |
| --- | --- | --- |
| Feature name | Type of data | description |
| ID | Numeric | The house identification number |
| Date | Factor | The date house was sold |
| Price | Numeric | The price at which the house was sold |
| Bedrooms | Int | Number of bedrooms in the house |
| Bathrooms | Numeric | Number of bathrooms in the house |
| Sqft\_living | Int | Square footage of the living room in the house |
| Sqft\_lot | Int | Square footage of the lot |
| Floors | Numeric | Total floors(levels) in the house |
| Waterfront | Int | House which has a view to a waterfront |
| View | Int | Has been viewed |
| Condition | Int | How good the condition is (overall) |
| Grade | int | The level of how the house is graded |
| Sqft\_above | Int | Square footage of house apart from basement |
| Sqft\_basement | Int | Square footage of the basement |
| Yr\_built | Int | Year the house was built |
| Yr\_renovation | Int | Year when the house was renovated |
| Zipcode | int | The zipcode of house address |
| Lat | Numeric | The latitude coordination of the house address |
| Long | Numeric | Longitude coordination of the house address |
| Sqft\_living15 | int | Living room area in 2015(implies some renovation), this might or might not have affected the lot size of the house |
| Sqft\_lot15 | int | Lot size of the house in 2015(implies some renovations) |

### Data preprocessing

1. Loading the dataset into the R studio. Check if there are missing values in the dataset

getwd()

setwd("M:/milina/Registered\_Courses\_2018\_Spring/Data Mining/project/")

getwd()

house\_data<-read.csv("kc\_house\_data.csv", header = TRUE, sep = ",")

> sum(is.na(house\_data))

[1] 0

The result above shows that there is no missing data in the dataset,

II. Preprocessing

Among these variables, the types of some variables are not appropriate for analyzing directly, it is necessary to transfer these types into the ones suitable for data analysis.

1. ‘Date’ variable: the type of this variable is factor which is not suitable for data analysis, so I convert it as following: first it is converted into date type in R, and then the date type is transferred to numeric type, since the sale date in the dataset is between May 2014 and May 2015, the variable of sale date is expressed as number of days to the earliest date in the dataset:

>date\_tem<-substr(house\_data$date,1,8)

> date\_tem<-as.Date(date\_tem,format = "%Y%m%d")

> date\_num<-as.numeric(date\_tem-min(date\_tem))

>house\_data$date<-date\_num

1. ‘ID’ variable: in the dataset, observations with exactly same ID number are found, it means that the same house was sold more than one times during the period of May 2014 and May 2015. These duplicated IDs are apparently not useful for the analysis, so the duplicated IDs are removed and only the observations with duplicated ID and largest value of variable DATEN were kept, that means only the observations of the houses which were sold at latest time are kept.

> sum(duplicated(house\_data$id))

[1] 177

>house\_data<-house\_data[order(house\_data$id, -house\_data$date), ]

> house\_data<-house\_data[!duplicated(house\_data$id),] #this will remove the duplicated record

Check if there are duplicated id number

> sum(duplicated(house\_data$id))

[1] 0

The result shows that duplications of the observation ID were successfully removed

1. ‘yr\_renovated’ variable: since for newly built houses and old houses which have never been renovated since they were built, the values of this variable are same, this condition definitely can not reflect the real features of properties. I convert it the variable of ‘nyrs\_since\_last\_renovate’, which reflects the number of years since last renovate to 2015, namely 2015-yr\_renovated, if the value of yr\_renovated is zero, then 2015-yr\_built.

yrs<-house\_data$yr\_renovated

yrs[yrs==0]<-house\_date$yr\_built[dat$yr\_renovated==0]

yrs<-house\_data$yr\_renovated

yrs[yrs==0]<-house\_data$yr\_built[dat$yr\_renovated==0]

house\_data<-data.frame(house\_data, nyrs\_since\_last\_renovation=as.numeric(2015-yrs))

1. Variables of ‘lat’ and ‘long’: since feature of ‘zipcode’ has indicated the location of house, variable ‘lat’ and ‘long’ which are latitude of house and longitude of house respectively are not necessary for data analysis. Therefore these two variables were not included in the final dataset, furthermore, zipcode attribute should be factor type instead of numeric type.

ycol<-"price"

xcols<-c('bedrooms', 'bathrooms', 'sqft\_living', 'sqft\_lot', 'floors', 'waterfront', 'view', 'condition', 'grade', 'sqft\_above', 'sqft\_basement', 'yr\_built', 'nyrs\_since\_last\_renovation', 'zipcode', 'sqft\_living15', 'sqft\_lot15','daten')

datafinal<-data.frame(price=house\_data$price, house\_data[, xcols])

the final dataset used for analysis is like the following:

str(datfinal)

'data.frame': 21436 obs. of 18 variables:

$ price : num 300000 647500 400000 235000 402500 ...

$ bedrooms : num 6 4 3 3 4 4 5 4 3 4 ...

$ bathrooms : num 3 1.75 1 1 2 2.75 1.5 2.5 1 2 ...

$ sqft\_living : num 2400 2060 1460 1430 1650 2220 1990 2540 1340 1980 ...

$ sqft\_lot : num 9373 26036 43000 7599 3504 ...

$ floors : num 2 1 1 1.5 1 1 1 2 1.5 1.5 ...

$ waterfront : num 0 0 0 0 0 0 0 0 0 0 ...

$ view : num 0 0 0 0 0 0 0 0 0 0 ...

$ condition : num 3 4 3 4 3 5 3 3 4 2 ...

$ grade : num 7 8 7 6 7 7 7 9 5 6 ...

$ sqft\_above : num 2400 1160 1460 1010 760 1170 1990 2540 1340 1980 ...

$ sqft\_basement : num 0 900 0 420 890 1050 0 0 0 0 ...

$ yr\_built : num 1991 1947 1952 1930 1951 ...

$ nyrs\_since\_last\_renovation: num 24 68 63 85 2 64 55 10 70 91 ...

$ zipcode : Factor w/ 70 levels "98001","98002",..: 2 64 64 65 60 60 67 47 21 31 ...

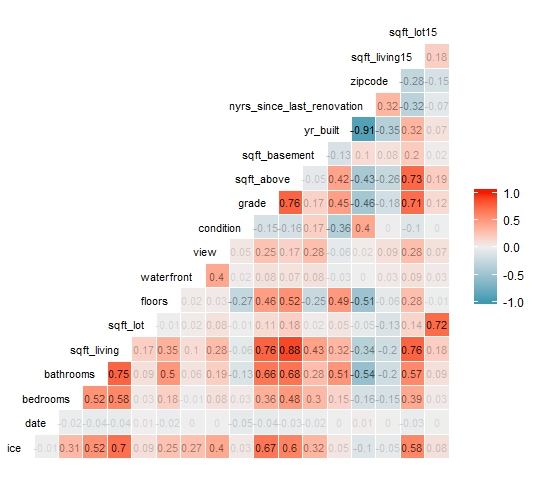
$ sqft\_living15 : num 2060 2590 2250 1290 1480 1540 1860 2360 1340 1360 ...

$ sqft\_lot15 : num 7316 21891 20023 10320 3504 ...

$ daten : num 355 6 101 334 321 332 298 68 194 186 ...

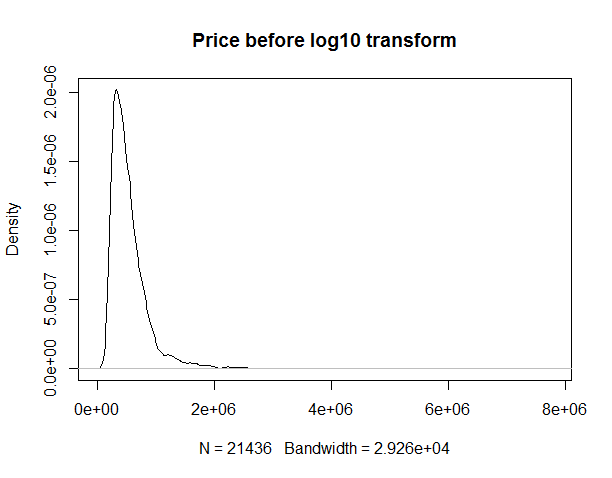
## Analysis on Variables

The degree of correlation among variables has critical effect on the model we used to analyze the dataset, so before choosing the model, I first check the collinearity between each pair of variables.

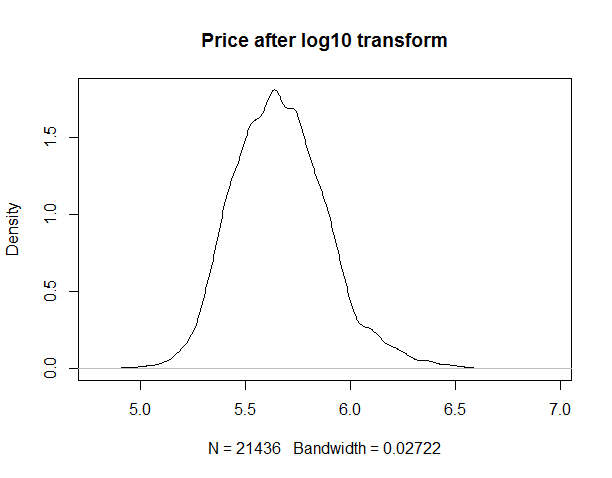


According the figures above, it is easy to tell that the attributes of “bedrooms”, “bathrooms”, “sqft\_living”, “view”,”grade”,”sqft\_above”,”sqft\_basement”, and “sqft\_living15” display strong correlations with “price”(Correlation>0.3). whereas, “Date”,”sqft\_lot”,”floors”,”waterfront”,”condition”,”yr\_built”,”yr\_renovated”, “zip code” and “sqft\_lot15”attributes have least effect on Price. (Correlation < 0.3).

In the dataset, ‘price’ is dependent variable and chosen as goal variable, it is necessary to examine the distribution of ‘price’ first.



The above figure shows that the distribution of ‘price’ variable is not normal distribution, so the value of variable ‘price’ should be manipulated to get normal distribution. Here, the ‘price’ value are taken logarithm with 10 as base, the distribution of ‘price’ variable after taken logarithm is shown in following:



From the above figure, it can be seen that the distribution of ‘price’ variable after taken logarithm become normal and the manipulation is appropriate.

**DETERMINING PRICE RMSE OF VARIOUS MODELS USING CARET PACKAGE**

**IMPLEMENTATION:**

performed as a part of this:

1. Set the “id” attribute to Null as it is not required.

2. Since we are predicting prices, Price attribute will be our target variable.

3.. We check the correlation of various attributes with target variable “Price”. We need to plot the

correlation between the variables. Once we get how strong the pairs are associated we can ignore

the variables which have least effect on the price. For this we will use ggcorr to get the

correlation plot.

ggcorr(kchd,label=TRUE,label\_alpha = TRUE,label\_round = 2,label\_size = 3,label\_color = "black",hjust

=1,size = 3)

4. We can see from below plot that attributes

**bedrooms,bathrooms,sqft\_living,view,grade,sqft\_above,sqft\_basement,lat,sqft\_living15**

display strong correlation with Price (Correlation >0.3).

5. **Date,sqft\_lot,floors,waterfront,condition,yr\_built,yr\_renovated,zip code,long,sqft\_lot15**

attributes have least effect on Price. (Correlation < 0.3).

We then look for multicollinearity effect between

**bedrooms,bathrooms,sqft\_living,view,grade,sqft\_above,sqft\_basement,lat,sqft\_living15** .

6.. This can be checked by calculating VIF factor and if it is close to 4 then the attribute which

displays better correlation with Price will be retained and the other can be ignored.

7. Based on the above rules the following attributes are set to NULL.

kchd$date = NULL

kchd$sqft\_lot = NULL

kchd$floors= NULL

kchd$waterfront = NULL

kchd$condition = NULL

kchd$yr\_built = NULL

kchd$yr\_renovated = NULL

kchd$lat = NULL

kchd$long = NULL

kchd$zipcode = NULL

kchd$sqft\_above = NULL

kchd$sqft\_lot15 = NULL

Some erroneous data entries are corrected next.

8. Now our data is ready for modelling.

9. Install caret package.

10. We run the following models using caret Package. I have chosen 8 folds.

mod\_glm generalized linear None

mod\_glmnet regularized generalized linear alpha, lambda

mod\_tree random forest mtry

mod\_elastic regularized lasso ridge fraction, lambda

mod\_leap\_back linear regression with backward selection nvmax

mod\_reg\_rf regularized random forest mtry, coefReg

mod\_svm SVM with linear kernelcost

mod\_gbm stochastic gradient boosting n.trees, interaction.depth, shrinkage, n.minobsinnode

mod\_bnet bayesian neural networks neurons

11. The results of the analysis are as follows:

Call:

summary.resamples(object = mod\_resamples)

Models: mod\_glm, mod\_glmnet, mod\_tree, mod\_elastic, mod\_leap\_back, mod\_reg\_rf, mod\_svm,

mod\_gbm, mod\_bnet

Number of resamples: 8

MAE

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

mod\_glm 0.1951601 0.1954763 0.1956949 0.1956855 0.1959455 0.1962047 0

mod\_glmnet 0.1951641 0.1954598 0.1956423 0.1956455 0.1958818 0.1961816 0

mod\_tree 0.1367878 0.1370932 0.1374694 0.1375149 0.1377659 0.1386098 0

mod\_elastic 0.1951601 0.1954763 0.1956949 0.1956855 0.1959455 0.1962047 0

mod\_leap\_back 0.2696710 0.2713264 0.2719976 0.2721700 0.2731862 0.2747076 0

mod\_reg\_rf 0.1372441 0.1372754 0.1375091 0.1377150 0.1380711 0.1387058 0

mod\_svm 0.1944865 0.1946668 0.1948347 0.1948703 0.1951034 0.1952757 0

mod\_gbm 0.1373093 0.1388117 0.1395085 0.1393859 0.1403185 0.1408180 0

mod\_bnet 0.1434312 0.1510893 0.1559311 0.1530243 0.1568821 0.1572920 0

RMSE

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

mod\_glm 0.2519210 0.2521701 0.2522885 0.2524483 0.2527045 0.2531350 0

mod\_glmnet 0.2519149 0.2521435 0.2522413 0.2524037 0.2526653 0.2530460 0

mod\_tree 0.1911446 0.1918267 0.1921104 0.1920681 0.1925063 0.1928317 0

mod\_elastic 0.2519210 0.2521701 0.2522885 0.2524483 0.2527045 0.2531350 0

mod\_leap\_back 0.3361042 0.3368770 0.3377682 0.3382484 0.3387802 0.3422098 0

mod\_reg\_rf 0.1912582 0.1921904 0.1922655 0.1923061 0.1925054 0.1930875 0

mod\_svm 0.2521333 0.2531906 0.2536248 0.2535216 0.2541046 0.2542538 0

mod\_gbm 0.1865435 0.1886761 0.1896588 0.1893533 0.1900393 0.1916768 0

mod\_bnet 0.1931231 0.2035878 0.2086957 0.2055513 0.2104688 0.2111849 0

Rsquared

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

mod\_glm 0.7687016 0.7698790 0.7706041 0.7705060 0.7712580 0.7719100 0

mod\_glmnet 0.7688509 0.7699629 0.7706207 0.7705774 0.7713920 0.7719429 0

mod\_tree 0.8674037 0.8676566 0.8687327 0.8687991 0.8696585 0.8705998 0

mod\_elastic 0.7687016 0.7698790 0.7706041 0.7705060 0.7712580 0.7719100 0

mod\_leap\_back 0.5785414 0.5853649 0.5889513 0.5877361 0.5905411 0.5929543 0

mod\_reg\_rf 0.8676732 0.8680151 0.8685855 0.8686928 0.8695081 0.8697172 0

mod\_svm 0.7662830 0.7670283 0.7685529 0.7685715 0.7697105 0.7714719 0

mod\_gbm 0.8673479 0.8700739 0.8705908 0.8709112 0.8716326 0.8747477 0

mod\_bnet 0.8390228 0.8407680 0.8437571 0.8478080 0.8501314 0.8660297 0

## Fitting Model with Linear Regression:

**IMPLEMENTATION:**

1.Training and Testing data:

Training Dataset- Training Dataset is a set of data which is used to discover potentially predictive relationships.

indexes = sample(1:nrow(HouseDataSet), size = 0.3\*nrow(HouseDataSet))

Train\_data = HouseDataSet[-indexes,] #Train dataset 70%

dim(Train\_data)

Test Dataset – Test dataset is a set of data used to access the strength and utility of a predictive relationship.

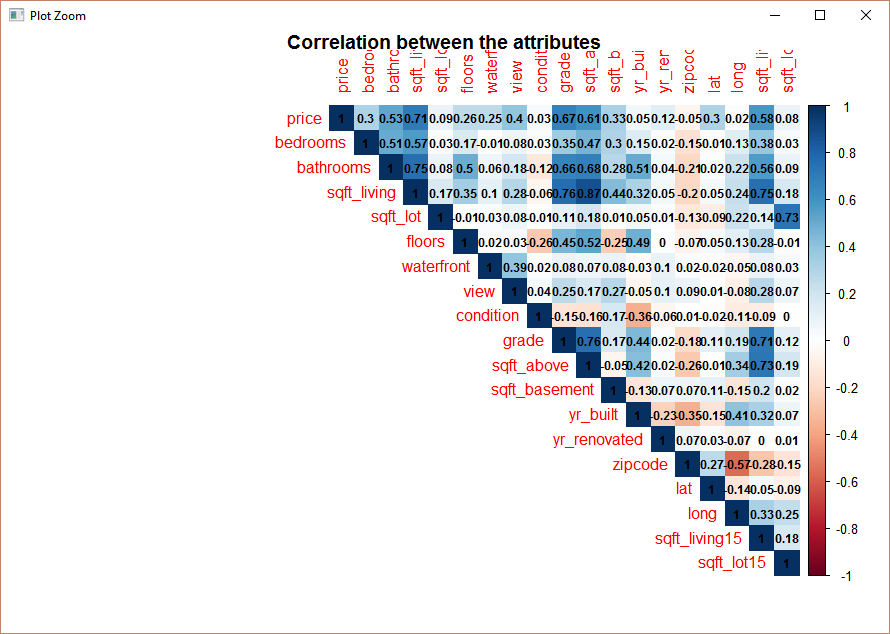
Test\_data = HouseDataSet[indexes,] #Test dataset 30%

dim(Test\_data)

Correlation:

Attribute\_Corr <-cor(Train\_data[3:21],method="pearson")

View(Attribute\_Corr)



2.Checking Linearity:

Simple linear regression model that describes the relationship between two variables x and y can be expressed by the following equation

Linear models make some strong assumptions concerning the data structure:

1.Independence of each data points

2.Correct distribution of the residuals

3.Correct specification of the variance structure

4.Linear relationship between the response and the linear predictor

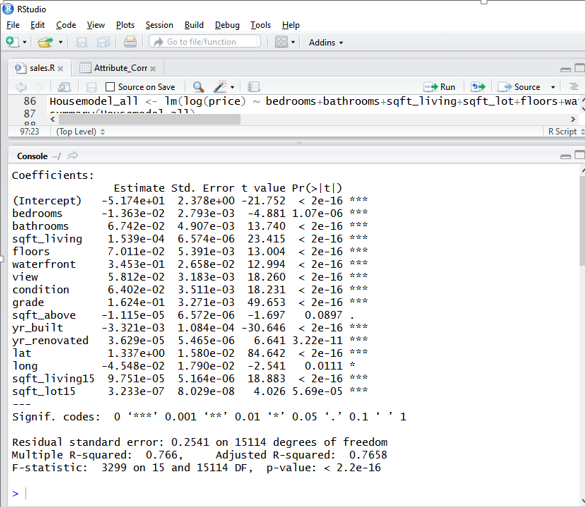
3. Model:

We tried making 7 models to check for the linear relationship between dependent and independent variables. We found the Housemodel\_1 as the fittest model for our predictive model.

R Code:

Housemodel\_i1<- lm(log(price) ~ bedrooms+bathrooms+sqft\_living+floors+waterfront+view+condition+grade+sqft\_above+yr\_built+yr\_renovated+lat+long+sqft\_living15+ sqft\_lot15, data = Train\_data)

summary(Housemodel\_i1) # Multiple R-Squared : 0.7684



4..Predicting the price:

We applied the predict() to predict the new price of the house using the best fitted linear model.

Equation

Y = a+b1X1 +b2X2 + +b kX

where Y is the dependent (Outcome) variable, and the variables X1…., X k are the explanatory (predictive) variables. The constants b1……….,b k are the regression coefficients and a is regression constant or intercept

pred <- predict(Housemodel\_i1,Test\_data)

summary(pred)

View(pred)

View (cbind("ID"=Test\_data$id,"Orginal Price"=Test\_data$price,"New predicted price"=exp(pred)))

## Fitting Model with Random Forest :

**Implementation:**

1. Splitting preprocessed dataset into training dataset and testing dataset

set.seed(1234)

ids<-sample(2,nrow(house\_data),prob=c(0.70,0.30),replace=TRUE) ###70% samples to build the mode,

train\_data<-datafinal[ids==1,]

test\_data<-datafinal[ids==2,]

1. Fitting the model with Random Forest

train\_data$price<-log10(train\_data$price)

model\_rf<-randomForest(price ~ .,train\_data,mtry=6,importance=TRUE)

summary(model\_rf)

Length Class Mode

call 5 -none- call

type 1 -none- character

predicted 15010 -none- numeric

mse 500 -none- numeric

rsq 500 -none- numeric

oob.times 15010 -none- numeric

importance 34 -none- numeric

importanceSD 17 -none- numeric

localImportance 0 -none- NULL

proximity 0 -none- NULL

ntree 1 -none- numeric

mtry 1 -none- numeric

forest 11 -none- list

coefs 0 -none- NULL

y 15010 -none- numeric

test 0 -none- NULL

inbag 0 -none- NULL

terms 3 terms call

importance(model\_rf)

%IncMSE IncNodePurity

date 9.765698 16.487502

bedrooms 23.826195 7.863034

bathrooms 31.661745 30.929622

sqft\_living 46.408308 142.144359

sqft\_lot 68.772628 25.192899

floors 23.202933 4.416281

waterfront 32.453565 3.749189

view 35.412470 11.870716

condition 29.787021 6.028057

grade 61.667900 183.495887

sqft\_above 25.284306 51.449590

sqft\_basement 34.577731 16.543809

yr\_built 104.650943 43.678168

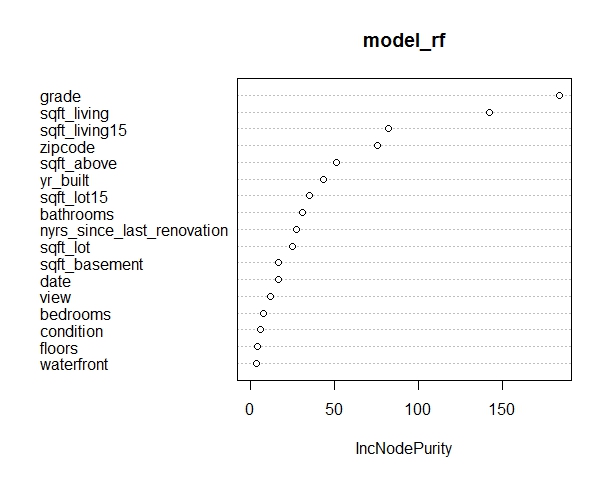
nyrs\_since\_last\_renovation 61.867635 27.604091

zipcode 114.810388 75.659382

sqft\_living15 62.446416 82.144425

sqft\_lot15 85.127854 35.060096

varImpPlot(model\_rf,type=2)



It can be seen from above figures that some X variables are highly correlated with dependable variable Y (such as variable ‘grade’ and ‘sqft\_living’), some X variables are much less correlated with Y. That means that different X variable has different degree of influence on variable Y.

1. Predicting prices of houses in testing dataset using the Random Forest Model

>test\_data$price<-log10(test\_data$price)

> test\_data$pred<-predict(model\_rf,test\_data)

> res<-test\_data$pred-test\_data$price

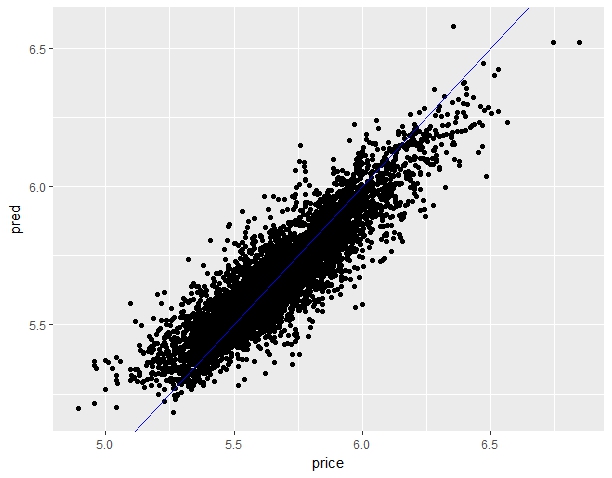
> rmse<-sqrt(mean(res^2))

> rmse\_rf

[1] 0.09962464

Plot the predicted vs. actual prices to check the prediction accuracy

ggplot(test\_data,aes(x=price,y=pred))+geom\_point()+geom\_abline(color="blue")



it can be seen from above plot that the predicted prices are fair good compared with actual prices of houses.

# Task Distribution:

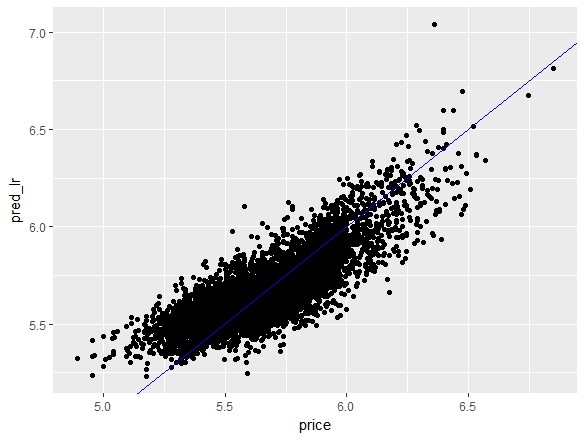
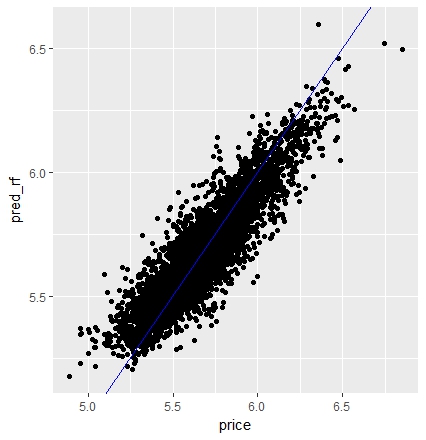
We discussed each step of the project implementation from selecting Data set to what methods we should use for cleaning data and analyzing it. Also, below is the list of implementation of algorithms.

Random Forest algorithm implementation: Lina MI

Linear Regression algorithm implementation: RESHMA KAPADNIS

Determining price RMSE of various models using CARET package: RADHIKA KASU

**Comparison of two model results**

The RMSE of both algorithms:

1)Linear regression model: RMSE\_lr=0.1324297

2) random forest model: RMSE\_rf=0.09962464

It can be seen from above two plots that random forest gave the better predictions on the price of king county houses.

**Conclusion:**

In this project, we chose king county house dataset as our target dataset, all three team members explored different algorithms to give predictions on the house prices based on the attributes of houses in the dataset. In the preprocessing of dataset, the attributes ‘date’, ‘yr\_renovated’ are transformed into more meaningful type, the duplicated the observations with same ID are eliminated, and the target variable ‘price’ are taken logarithm to get the normal distribution of ‘price’. Furthermore, the correlations among attributes and correlations of each attribute with target variable ‘price’ are analyzed, the analysis results reveals complex correlations among the attributes and target variable ‘price’. Then training dataset are used to fit linear regression and random forest individually, then used the trained models to make prediction on the prices of houses in test dataset, the results from both models show that the predictions made by two models are fair good, and the random forest model gave a little bit better result compared with linear regression. It can also be seen from the prediction results of both models that the preprocessing on dataset and model parameters selection are successful. We deepened our understanding on the random forest algorithm and linear regression model during complement of this project.

**Challenges:**

1.It was very difficult to clean data which suits for every model.

2.Calculating RMSE value with considering different attributes each algorithm considers and then compare them.

**Future Enhancement:**

Further we decided to create models for our dataset using different regression methods like- Ridge Regression, Lasso Regression. For as of now, we predicted the price of house sales using only multiple regression because of the time constraint.

Different parameters can be taken into consideration:

1. We can be considering more parameters to get the correct results as per the user requirements.

2. We can consider different types of homes like Colonial, Raised Ranches, Condominiums etc.

3.We can also consider and classify if the house is FHA approved or not.

Area demands:

Each area has different demands like in California and Arizona most people prefer to have a swimming pool where as in New England there isn’t a big demand for swimming pools. So we can consider and sort data according to similar factors for more precise results.

# References

* Kaggle.com
* Data Mining With Rattle and R- Graham William
* Data Mining –concepts, Models & Techniques.
* R Data Mining -Yanchang Zhao