Math 448 – Final Project Report
House price prediction

Student ID: 921476449
Name: Nguyen Quoc Hung

I. Executive summary

While many people can have a clear vision about their dream house, little do they know about the value of having two more bathrooms or living 5km nearer to the station. Affordability, or satisfaction are more preferable factors, not because people have no concern about the value of the house, but because unlike shopping in the mall and supermarket, buying house is not on a regular basis and pricing it tend to be difficult for most people. It is not rare to see researches attempting to reach the answer for the connection between various variable and house pricing. However, most cases have their conclusion come right after using linear regression model without comparing different methods.

I used data published by Kaggle.com which includes indepth information about each house price from 322 suburbs in Perth. Furthermore, I went far beyond the scope of linear regression, applying a variety of different regression methods in order to assess which method predicts the price the best and how sufficient the relationships among the variables are. With this data, I applied multiple linear regression models, variable selection models, dimension reduction models, model regularization, and tree-based methods.

Firstly, I start by using the multiple linear regression model and it perform quite good in prediction. Secondly, by using multiple linear regression, I realized that there are some variables which coefficients are small compared to others, therefore, I use the variable selection models to improve the prediction accuracy. However, the result is become almost the same, therefore, my next thought is to use the dimensional reduction models. Finally, because the predictor variables are not all numeric, so I use the tree-based methods to verify the relationship between the predictor and the response variable.

After using methods that are mentioned from above, it can be concluded that all the predictor variables have a relationship with the response variables. Moreover, I found that the number of rooms in a house is the most rural factor to decide the price of the house. In contrast, the land area does not seem to decide the house is expensive or not.

With future work, I believe that it might be interesting to further investigate and understand the reinforcing effect of other external factors to Perth House Prices and especially if house with other stage of the art equipment and living condition such as swimming pool or garden can provide higher price or not.

II. Introduction.

Real estate market is always followed attentively by the government in every country because it directly relates to huge amount of assets in terms of size, nature as well as many other factors in the national economy. Also, buying a real estate, or to be more specific, a house, is one of the most imperative assets that every people want to own.

In this project, I wish to explore the relationship between various internal and external factors of a house and its price in Perth, a biggest city in West Australia. The internal factors are the area of the house, the number of rooms and the year the house was built, while the external factors are the location, the nearest station, the nearest school and the rank of the nearest school. My objective of this project is prediction because I want to find out which house is suitable for people that can spend a specific amount of money.

To my knowledge, all study methods are being used more than ten years old and only based on linear models. However, by using different types of methods, I want to find out which one will be the most consistent for the following data.

III. Description of the data

The title of the dataset is "Perth House Price" from the https://www.kaggle.com/syuzai/perth-house-prices. The website is public and it is free so I do not need permission to use this data. This data includes data from 322 Perth suburbs, resulting in an average of about 100 rows per suburb. It contains 19 variables, of which are 4 "categorical" and 15 "numeric". "Price" is a dependent variable which I want to predict based on other 18 independent variables, which is: address, suburb, bedrooms, bathrooms, garage, land area, floor area, build year, cbd distance, nearest station, nearest station distance, date sold, postcode, latitude, longitude, nearest school, nearest school distance and nearest school rank. However, in this report, I will concentrate only 11 variables are considered in this project, which is the response Y "price" and 10 predictors X: "BEDROOMS", "BATHROOMS", "GARAGE", "LAND_AREA", "FLOOR_AREA", "NEAREST_SCH_DIST", "BUILD_YEAR", "NEAREST_STN_DIST", "POSTCODE" and "NEAREST_SCH_RANK". The remaining 8 variables are not being used because it do not relate to the data I want to predict, in my point of view.

This Perth house price data set provides both inputs (internal and external factors) and outputs (the value of a house). This project is a regression problem, when it is interested in prediction. The dataset will be separate into 2: 90% for training dataset and 10% for testing dataset.

IV. Preprocessing

1. Data summary and visualization

The data includes 33,656 rows and 19 columns, which symbolize 19 variables and 33,656 values appropriate with those variables as I mentioned from above. The average price for a house in Perth is 535,000$, when the minimum price and the maximum price is $51,000 and $2,440,000, respectively.

```
   ADDRESS              SUBURB               PRICE             BEDROOMS          BATHROOMS            GARAGE         
 Length:33656       Length:33656       Min.   :  51000   Min.   : 1.000   Min.   : 1.000   Length:33656      
 Class :character   Class :character   1st Qu.: 410000   1st Qu.: 3.000   1st Qu.: 1.000   Class :character  
 Mode  :character   Mode  :character   Median : 535500   Median : 4.000   Median : 2.000   Mode  :character  
                                       Mean   : 637072   Mean   : 3.659   Mean   : 1.823                     
                                       3rd Qu.: 760000   3rd Qu.: 4.000   3rd Qu.: 2.000                     
                                       Max.   :2440000   Max.   :10.000   Max.   :16.000                     

   LAND_AREA        FLOOR_AREA     BUILD_YEAR           CBD_DIST      NEAREST_STN        NEAREST_STN_DIST
 Min.   :    61   Min.   :  1.0   Length:33656       Min.   :  681   Length:33656       Min.   :   46   
 1st Qu.:   503   1st Qu.:130.0   Class :character   1st Qu.:11200   Class :character   1st Qu.: 1800   
 Median :   682   Median :172.0   Mode  :character   Median :17500   Mode  :character   Median : 3200   
 Mean   :  2741   Mean   :183.5                      Mean   :19777                      Mean   : 4523   
 3rd Qu.:   838   3rd Qu.:222.2                      3rd Qu.:26600                      3rd Qu.: 5300   
 Max.   :999999   Max.   :870.0                      Max.   :59800                      Max.   :35500   

  DATE_SOLD            POSTCODE       LATITUDE        LONGITUDE     NEAREST_SCH        NEAREST_SCH_DIST 
 Length:33656       Min.   :6003   Min.   :-32.47   Min.   :115.6   Length:33656       Min.   : 0.07091  
 Class :character   1st Qu.:6050   1st Qu.:-32.07   1st Qu.:115.8   Class :character   1st Qu.: 0.88057  
 Mode  :character   Median :6069   Median :-31.93   Median :115.9   Mode  :character   Median : 1.34552  
                    Mean   :6089   Mean   :-31.96   Mean   :115.9                      Mean   : 1.81527  
                    3rd Qu.:6150   3rd Qu.:-31.84   3rd Qu.:116.0                      3rd Qu.: 2.09722  
                    Max.   :6558   Max.   :-31.46   Max.   :116.3                      Max.   :23.25437  

 NEAREST_SCH_RANK
 Min.   :  1.00  
 1st Qu.: 39.00  
 Median : 68.00  
 Mean   : 72.67  
 3rd Qu.:105.00  
 Max.   :139.00  
 NA's   :10952   
```

2. **Verifying and understanding variables.**

"PRICE", "BEDROOMS", "BATHROOMS", "GARAGE", "LAND_AREA", "FLOOR_AREA", "BUILD_YEAR", "NEAREST_STN_DIST", "NEAREST_SCH_DIST" and "NEAREST_SCH_RANK" are numerical. On the other hand, "POSTCODE" needs to be changed to categorical.

The reason I change "POSTCODE" as a categorical variable is that each postcode represents a small area, which is more clearly than the "SUBURB" variables. Therefore, it reflexes which types of the area is "hot" or not, so it should be changed to "categorical".

3. **Missing values and Outliers**

There are total 16,585 missing values: 2,478 missing values comes from "GARAGE" variables; 3,155 missing values comes from "BUILD_YEAR" variables; and the rest 10,952 missing values comes from "NEAREST_SCH_RANK" variables.

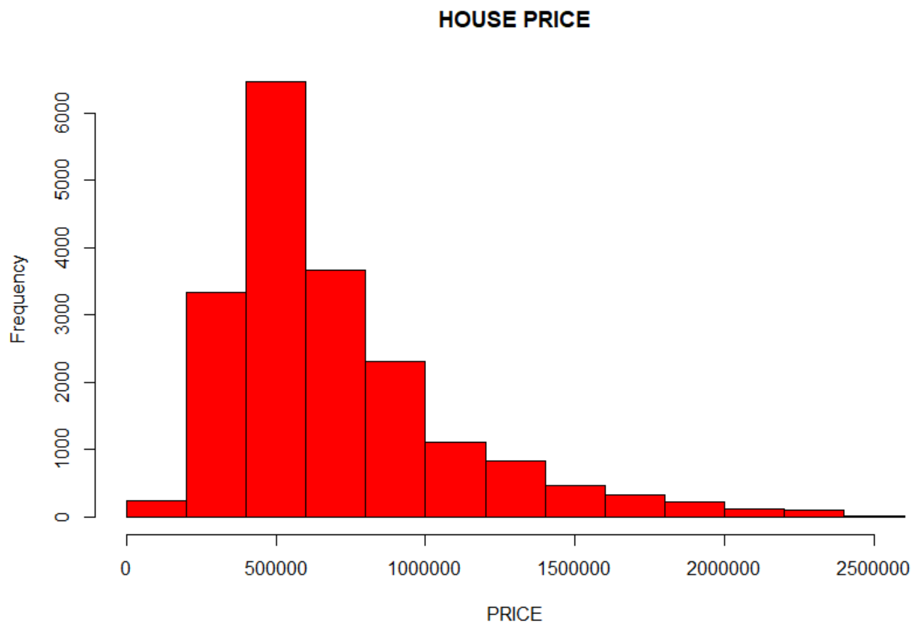
Moreover, there are some "outliers", which is also needed to be removed. For example, an observation shows that a 45m2 "FLOOR_AREA" has 5 bedrooms, 2 bathrooms and 4 garage, which is illogical. Moreover, all the observation which "LAND_AREA" are more than 500,000 also needs to be removed due to the irrelevant price. On the other hand, another observation shows that a house contains 50 "GARAGE", which is a large amount of number. However, the "LAND_AREA" is big (22,367), so it still makes sense.

After cleaning all the missing values and outliers, there are 19,197 observations left, which is ready to process.

```
     PRICE             BEDROOMS          BATHROOMS          GARAGE            LAND_AREA          FLOOR_AREA         BUILD_YEAR
 Min.   :  52000   Min.   : 1.000   Min.   :1.000   Min.   : 1.000   Min.   :     61   Min.   : 52.0   Min.   :1870
 1st Qu.: 438000   1st Qu.: 3.000   1st Qu.:2.000   1st Qu.: 2.000   1st Qu.:    494   1st Qu.:134.0   1st Qu.:1977
 Median : 585000   Median : 4.000   Median :2.000   Median : 2.000   Median :    675   Median :177.0   Median :1995
 Mean   : 699807   Mean   : 3.676   Mean   :1.862   Mean   : 2.183   Mean   :   2236   Mean   :187.6   Mean   :1989
 3rd Qu.: 850000   3rd Qu.: 4.000   3rd Qu.:2.000   3rd Qu.: 2.000   3rd Qu.:    809   3rd Qu.:228.0   3rd Qu.:2005
 Max.   :2440000   Max.   :10.000   Max.   :7.000   Max.   :50.000   Max.   :496919   Max.   :849.0   Max.   :2017
 NEAREST_STN_DIST  DATE_SOLD           POSTCODE          NEAREST_SCH_DIST   NEAREST_SCH_RANK
 Min.   :   46    Length:19197       Length:19197       Min.   : 0.07091   Min.   :  1.00
 1st Qu.: 1600    Class :character   Class :character   1st Qu.: 0.86583   1st Qu.: 38.00
 Median : 3000    Mode  :character   Mode  :character   Median : 1.30145   Median : 65.00
 Mean   : 4188                                          Mean   : 1.68422   Mean   : 72.11
 3rd Qu.: 5100                                          3rd Qu.: 1.95707   3rd Qu.:105.00
 Max.   :34300                                          Max.   :20.72091   Max.   :139.00
```

4. **Additional visualization based on the cleaning data.**

The table shows the frequently of the house's price based on the remaining cleaning data. It can be seen that more than 6000 houses is around $50000, which occupies the biggest proportion of all Perth houses' prices.

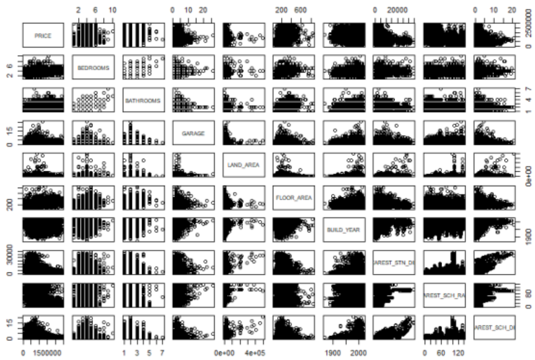


Also, the graph skewed to the right, which means that mean > median > mode.

V. Model selection.

1. Linear Regression

To begin the analysis, we start with the scatter plot matrix with all numeric variables. Overall, it is hard to conclude with this picture except that there are negative relationship between the price and the year built, which can explained the older the house is built, the more expensive the house is.



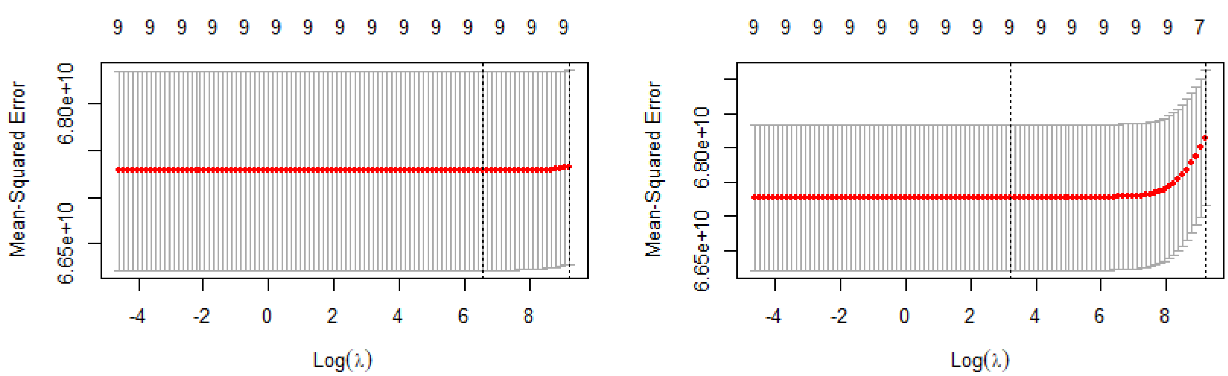
```
Call:
lm(formula = PRICE ~ BEDROOMS + BATHROOMS + GARAGE + LAND_AREA +
    FLOOR_AREA + BUILD_YEAR + NEAREST_STN_DIST + NEAREST_SCH_RANK +
    NEAREST_SCH_DIST, data = fin.train)

Residuals:
     Min       1Q   Median       3Q      Max
-2109302  -144255   -24354   107848  1619684

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       1.009e+07  1.934e+05  52.194  < 2e-16 ***
BEDROOMS         -2.424e+04  3.420e+03  -7.088 1.42e-12 ***
BATHROOMS         1.287e+05  4.622e+03  27.833  < 2e-16 ***
GARAGE            7.803e+03  1.782e+03   4.378 1.21e-05 ***
LAND_AREA         1.918e+00  1.821e-01  10.532  < 2e-16 ***
FLOOR_AREA        2.549e+03  3.623e+01  70.339  < 2e-16 ***
BUILD_YEAR       -4.924e+03  9.864e+01 -49.921  < 2e-16 ***
NEAREST_STN_DIST -1.087e+01  5.934e-01 -18.320  < 2e-16 ***
NEAREST_SCH_RANK -2.985e+03  5.166e+01 -57.789  < 2e-16 ***
NEAREST_SCH_DIST  6.159e+03  1.608e+03   3.829 0.000129 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 259200 on 17267 degrees of freedom
Multiple R-squared:  0.5526,	Adjusted R-squared:  0.5523
F-statistic:  2369 on 9 and 17267 DF,  p-value: < 2.2e-16
```

The first model I use is a standard multiple linear regression – a good model performance to use when there are many numeric predictor variables - on the full dataset to see what predictors may be significant. It can be seen that the p-value for all of the predictors are small, which means that there is a significant evidence to conclude that all the numeric predictor variables have a relationship with the response variable “PRICE”. The number of bedrooms, the distance of the nearest station, the rank of the nearest school decrease will make the “Price” of the house increase and opposite, the number of bathrooms, garage, the area of the house increase will make the “Price” increase due to the positive coefficient of the above variables. On the other hand, the higher school rank makes the house that near from that school become more expensive.

The R-squared is 0.5526, which means that using Multiple Linear Model Regression explained 55.26% of the data set, while the squared – root test MSE is 247,126.

2. The Ridge Regression and Lasso

Ridge regression is the method of estimating the coefficient of multiple regression models when the independent variables are highly correlated. When multiple regression occurs, the variance is large and far from the true value. By increasing the bias, the ridge regression decreases the variance. The best Lambda when using Ridge Regression is 705.4802, which gives the best MSE. The squared-root Test MSE is 247,153.7.



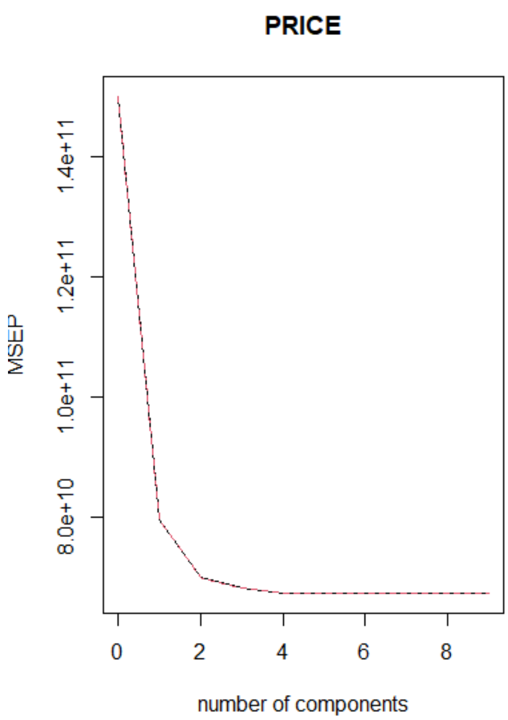
The correlation between Lambda and MSE

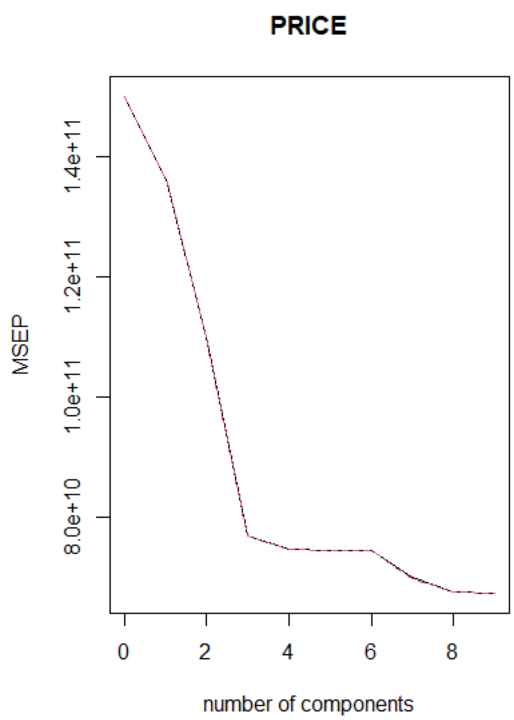
Right: Ridge Regression – Left: The Lasso

The Lasso - stands for Least Absolute Shrinkage and Selection Operator – is a method that is used when there is a high difference between the coefficients' values. Ridge regression is to shrink the value of the coefficients closely to 0, but in fact, when there are too many independent variables, the model is too complicated to calculate. Lasso regression solves the problem by assuming all the coefficients that are pretty small to others remaining coefficients by 0. The best Lambda when using The Lasso model is 24.77076, while the squared – root test MSE is 247,132, which is a little bit better than the Ridge Regression. However, considering that both abovementioned methods are neither superior concerning prediction nor superior from an inferential point of view, multiple linear regression seems to be easier to use in this application.

3. Principal Components Regression and Partial Least Square Regression (PCR and PLS)

Principal Components Regression and Partial Least Square Regression are 2 methods that both based on the idea of using the technique to reduce the dimension when facing a high-dimensional data, which can lead to complicated calculation, increase the test error and so on. However, unlike PCR, which is unlikely that selected principal components are associated with the outcome, PLS can identify a new principal component that not only summarizes the original predictors, but also that are related to the outcome.





Using PCR method gives a result that when the number of components = 9, the CV error will become minimize. The squared – root test MSE is 281,991.5

```
Data:   X dimension: 17277 9
        Y dimension: 17277 1
Fit method: svdpc
Number of components considered: 9

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps  9 comps
CV          387340   369186   331710   277113   273123   272795   272477   264213   259643   259381
adjCV       387340   369181   331694   277111   273113   272782   272468   264102   259622   259365

TRAINING: % variance explained
       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps  9 comps
X        28.32    48.68    60.87    71.07    79.17    86.61    91.32    95.84   100.00
PRICE     9.19    26.71    48.86    50.34    50.48    50.58    53.61    55.17    55.26
```

On the other hand, using PLS gives a result that when the number of components = 4, the CV error will become minimize. The squared – root test MSE is 251,651.7, which is less than the PCR method.

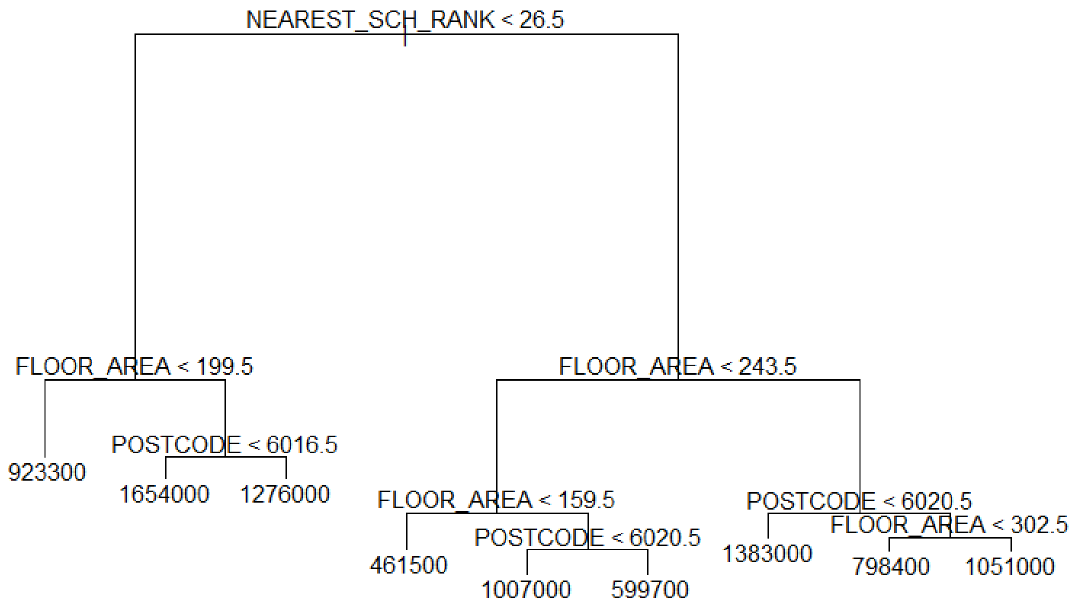
```
Data:   X dimension: 17277 9
        Y dimension: 17277 1
Fit method: kernelpls
Number of components considered: 9

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps  9 comps
CV          387340   281851   264251   260966   259412   259382   259381   259381   259381   259381
adjCV       387340   281846   264243   260950   259396   259366   259365   259365   259365   259365

TRAINING: % variance explained
       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps  9 comps
X        21.92    41.01    58.02    65.59    75.10    82.55    88.02    95.26   100.00
PRICE    47.10    53.53    54.70    55.25    55.26    55.26    55.26    55.26    55.26
```

However, as I mentioned from above, there are a significant correlation between the response Y "Price" and the other predictors X. Therefore, reducing the dimension in this case does not improve the prediction accuracy specifically. In fact, the test MSE by using multiple linear regression model is still less than reducing dimensional-model like PLS or PCR.

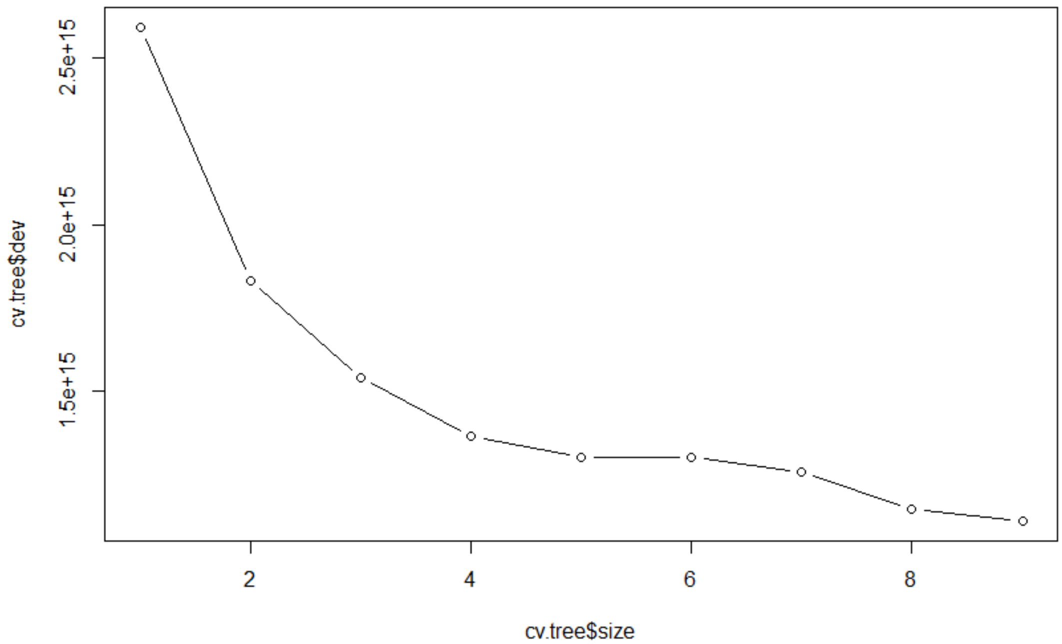
4. Decision tree

All of the methods that are demonstrated from above are only predict the response variables based on the numeric predictors. However, the "POSTCODE" variables is also important to concentrate as a predictor variables. Due to the non-numeric class of the "POSTCODE", decision tree method are used in this project.

Decision tree is a simple procedure which identifies variables that provide optimal separation of classes by splitting the data on their values. It is used for both regression problem and classification problems to make decision. Regression tree is used to predict quantitative response, while classification tree is used to predict qualitative response. In this case, we use regression tree to predict the numeric "Price" variables.



Using the function cv.tree() from the tree package to find the best node for minimize the RSS, however, the result comes to a surprise that with 9 nodes will gives the lowest MSE. Therefore, pruning the tree in this case can just make the tree look simple, but not improve the prediction accuracy. With 9 nodes, the test MSE by using regression decision tree is 252,301.6



5. Bagging

Bagging or bootstrap-aggregation is a way to improve the fit of decision trees and tree-based methods further. It is an algorithm that fit multiple models on different subsets of a training dataset then combines the predictions from the model. The idea of bagging based on 2 things:

Averaging: reduces variance

Bootstrapping: myriad training datasets

The test MSE by using bagging is 183,118.3, which is the smallest MSE compare to all of the methods that are used.

```
Call:
 randomForest(formula = PRICE ~ BEDROOMS + BATHROOMS + GARAGE +      LAND_AREA + FLOOR_AREA + BUILD_YEAR + NEAREST_STN_D
IST +      NEAREST_SCH_RANK + NEAREST_SCH_DIST + POSTCODE, data = fin.train,      importance = TRUE)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 3

          Mean of squared residuals: 28399458274
                    % Var explained: 81.07
```

```
                   %IncMSE IncNodePurity
BEDROOMS          42.82085  7.429023e+13
BATHROOMS         40.89092  1.631010e+14
GARAGE            21.15199  3.656042e+13
LAND_AREA         97.22820  1.663179e+14
FLOOR_AREA       140.97578  5.981422e+14
BUILD_YEAR        77.85908  1.627395e+14
NEAREST_STN_DIST  73.13850  1.237232e+14
NEAREST_SCH_RANK 105.37995  5.913169e+14
NEAREST_SCH_DIST  51.95750  9.690508e+13
POSTCODE          92.70406  5.173241e+14
```

Comment: I do not use random forest to compare with bagging because the calculation is really a hard test that cost a lot of time for the computer to process.

VI. Conclusion

To decide the most consistent statistical learning method to be used in this dataset, the importance of prediction accuracy and coefficient interpretation are essential factors to take into consideration. Lasso and Ridge Regression gives almost the same prediction accuracy and R-squared compared to multiple linear regression, although it takes a lot of computer resources. Same to variable selection methods, reducing dimensional methods even give poor performance than multiple linear regression. On the other hand, Tree-based methods provide a good performance with low MSE and high R-squared, especially Bagging methods.

R CODE

```
#### Name: Nguyen Quoc Hung
knitr::opts_chunk$set(echo = TRUE)
library(MASS)
library(dplyr)
library(ggplot2)
library(ISLR)
library(plyr)
library(stringr)
library(tidyr)
library(readr)
library(glmnet)
library(xtable)
library(plsr)
library(splines)
library(pls)
library(tree)
library(randomForest)
data <- read_csv("D:/COURSE/Math 448/perth.csv.csv")
View(data)
dim(data) #counting how many rows and columns in this data
summary(data) #data summary
hist(data$PRICE) #how frequently for the house’s price, the graph’s skewness
boxplot(data$PRICE)
data1 <-data[-1:-2]
data2 <-data1[-8:-9]
dataclean <-data2[-11:-13] #filter variables and keep the variables that is necessary
dataclean$PRICE <- as.numeric((dataclean$PRICE))
dataclean$BEDROOMS <- as.numeric((dataclean$BEDROOMS))
dataclean$BATHROOMS <- as.numeric((dataclean$BATHROOMS))
dataclean$GARAGE <- as.numeric((dataclean$GARAGE))
dataclean$LAND_AREA <- as.numeric((dataclean$LAND_AREA))
dataclean$FLOOR_AREA <- as.numeric((dataclean$FLOOR_AREA))
dataclean$BUILD_YEAR <- as.numeric((dataclean$BUILD_YEAR))
dataclean$NEAREST_STN_DIST <- as.numeric((dataclean$NEAREST_STN_DIST))
dataclean$POSTCODE <-as.character(dataclean$POSTCODE)
dataclean$NEAREST_SCH_DIST <- as.numeric(dataclean$NEAREST_SCH_DIST)
dataclean$NEAREST_SCH_RANK <- as.numeric(dataclean$NEAREST_SCH_RANK)
dataclean$DATE_SOLD <- as.character(dataclean$DATE_SOLD)
sum(is.na(dataclean)) #check the total of missing value
summary(dataclean) #look at the missing value in each variable
```

```
dataclean <-dataclean %>%
na.omit() #exclude the missing value
sum(is.na(dataclean))
dim(dataclean)
summary(dataclean)
fin<-subset(dataclean,LAND_AREA <= 500000 & FLOOR_AREA >=50) #keep all the
observations from dataclean which “LAND_AREA” are less than 500,000 and keep all the
observations from dataclean1 which “FLOOR_AREA” are more than 50
dim(fin)
summary(fin)
View(fin)
hist(fin$PRICE, main = "HOUSE PRICE", xlab = "PRICE", col = "red")
class(fin$POSTCODE)

### setting a training and test set (90 - 10)

set.seed(159)
train = sample(0.9*nrow(fin))
fin.train <- fin[train,]     #training set
fin.test <- fin[-train,]     #test set

################################
# Multiple Linear Regression ###
################################



lm.fit = lm(PRICE~BEDROOMS + BATHROOMS + GARAGE + LAND_AREA +
FLOOR_AREA + BUILD_YEAR + NEAREST_STN_DIST + NEAREST_SCH_RANK +
NEAREST_SCH_DIST,data = fin.train)
summary(lm.fit)
coef(lm.fit)
confint(lm.fit)
lm.pred <- predict(lm.fit,fin.test)
lm.pred
MSE.lm <- mean((lm.pred-fin.test$PRICE)^2) #Test MSE
MSE.lm
sqrt(MSE.lm)
summary(lm.fit)$sigma
summary(lm.fit)$r.sq
```

```
pairs(PRICE~BEDROOMS + BATHROOMS + GARAGE + LAND_AREA +
FLOOR_AREA + BUILD_YEAR + NEAREST_STN_DIST + NEAREST_SCH_RANK +
NEAREST_SCH_DIST,data = fin.train)
AIC(lm.fit)
par(mfrow=c(2,2))
plot(lm.fit)

#################################
###     Ridge regression     ###
#################################

set.seed(159)
train.mat <- model.matrix(PRICE ~ BEDROOMS + BATHROOMS + GARAGE +
LAND_AREA + FLOOR_AREA + BUILD_YEAR + NEAREST_STN_DIST +
NEAREST_SCH_RANK + NEAREST_SCH_DIST, data = fin.train)
test.mat <- model.matrix(PRICE ~ BEDROOMS + BATHROOMS + GARAGE +
LAND_AREA + FLOOR_AREA + BUILD_YEAR + NEAREST_STN_DIST +
NEAREST_SCH_RANK + NEAREST_SCH_DIST, data = fin.test)
grid <- 10 ^ seq(4, -2, length = 100)
ridge.fit <- glmnet(train.mat, fin.train$PRICE, alpha = 0, lambda = grid, thresh = 1e-12)
ridge.cv <- cv.glmnet(train.mat, fin.train$PRICE, alpha = 0, lambda = grid, thresh = 1e-12)
bestlam.ridge <- ridge.cv$lambda.min
bestlam.ridge
ridge.pred <- predict(ridge.fit, s = bestlam.ridge, newx = test.mat)
MSE.ridge <- mean((ridge.pred - fin.test$PRICE)^2)
MSE.ridge
sqrt(MSE.ridge)
summary(ridge.fit)
plot(ridge.cv)
#################################
###        The LASSO       ###
#################################

set.seed(159)
train.mat <- model.matrix(PRICE ~ BEDROOMS + BATHROOMS + GARAGE +
LAND_AREA + FLOOR_AREA + BUILD_YEAR + NEAREST_STN_DIST +
NEAREST_SCH_RANK + NEAREST_SCH_DIST, data = fin.train)
test.mat <- model.matrix(PRICE ~ BEDROOMS + BATHROOMS + GARAGE +
LAND_AREA + FLOOR_AREA + BUILD_YEAR + NEAREST_STN_DIST +
NEAREST_SCH_RANK + NEAREST_SCH_DIST, data = fin.test)
lasso.fit <- glmnet(train.mat, fin.train$PRICE, alpha = 1, lambda = grid, thresh = 1e-12)
```

```
lasso.cv <- cv.glmnet(train.mat, fin.train$PRICE, alpha = 1, lambda = grid, thresh = 1e-12)
bestlam.lasso <- lasso.cv$lambda.min
bestlam.lasso
lasso.pred <- predict(lasso.fit, s = bestlam.lasso, newx = test.mat)
MSE.lasso <- mean((lasso.pred - fin.test$PRICE)^2)
sqrt(MSE.lasso)
plot(lasso.cv)

#################################
###           PLS           ###
#################################
par(mfrow=c(1,2))
set.seed(159)
fit.pls <- plsr(PRICE ~ BEDROOMS + BATHROOMS + GARAGE + LAND_AREA +
FLOOR_AREA + BUILD_YEAR + NEAREST_STN_DIST + NEAREST_SCH_RANK +
NEAREST_SCH_DIST, data = fin.train, scale = TRUE, validation = "CV")
validationplot(fit.pls, val.type = "MSEP")
summary(fit.pls)
pred.pls <- predict(fit.pls, fin.test)
MSE.pls<-mean((pred.pls - fin.test$PRICE)^2)
sqrt(MSE.pls)

#################################
###           PCR           ###
#################################

set.seed(159)
fit.pcr <- pcr(PRICE ~ BEDROOMS + BATHROOMS + GARAGE + LAND_AREA +
FLOOR_AREA + BUILD_YEAR + NEAREST_STN_DIST + NEAREST_SCH_RANK +
NEAREST_SCH_DIST, data = fin.train, scale = TRUE, validation = "CV")
validationplot(fit.pcr, val.type = "MSEP")
summary(fit.pcr)
pred.pcr <- predict(fit.pcr, fin.test)
MSE.pcr <- mean((pred.pcr - fin.test$PRICE)^2)
sqrt(MSE.pcr)

#comparision pls and pcr

ggplot(fit.pls)
```

```
##################################
###      decision tree       ###
##################################

tree <- tree(PRICE ~ BEDROOMS + BATHROOMS + GARAGE + LAND_AREA +
FLOOR_AREA + BUILD_YEAR + NEAREST_STN_DIST + NEAREST_SCH_RANK +
NEAREST_SCH_DIST + POSTCODE, data = fin.train)
summary(tree)
plot(tree)
text(tree, pretty = 0)
pred.tree <- predict(tree, fin.test)
MSE.tree <- mean((fin.test$PRICE - pred.tree)^2)
sqrt(MSE.tree)

#pruning the tree
cv.tree <- cv.tree(tree, FUN=prune.tree)
par(mfrow=c(1, 1))
plot(cv.tree$size, cv.tree$dev, type="b")
#with 9 nodes give the lowest CV error means that this model cannot be prunned.

#check the CV error with 6 nodes
prune.tree <- prune.tree(tree, best = 6)
par(mfrow = c(1, 1))
plot(prune.tree)
text(prune.tree, pretty = 0)
pred.prunetree <- predict(prune.tree, fin.test)
MSE.prunetree <- mean((fin.test$PRICE - pred.prunetree)^2)
sqrt(MSE.prunetree)

##################################
##       Bagging            ##
##################################

set.seed(159)
bag <- randomForest(PRICE ~ BEDROOMS + BATHROOMS + GARAGE +
LAND_AREA + FLOOR_AREA + BUILD_YEAR + NEAREST_STN_DIST +
NEAREST_SCH_RANK + NEAREST_SCH_DIST + POSTCODE, data=fin.train,
importance=TRUE)
bag
pred.bag = predict(bag, fin.test)
MSE.bag <- mean((fin.test$PRICE - pred.bag)^2)
```

```
sqrt(MSE.bag)
importance(bag)


#comparision
test.avg <- mean(fin.test$PRICE)
lm.r2 <- 1 - mean((lm.pred - fin.test$PRICE)^2) / mean((test.avg - fin.test$PRICE)^2)
ridge.r2 <- 1 - mean((ridge.pred - fin.test$PRICE)^2) / mean((test.avg - fin.test$PRICE)^2)
lasso.r2 <- 1 - mean((lasso.pred - fin.test$PRICE)^2) / mean((test.avg - fin.test$PRICE)^2)
pls.r2 <- 1 - mean((pred.pls - fin.test$PRICE)^2) / mean((test.avg - fin.test$PRICE)^2)
pcr.r2 <- 1 - mean((pred.pcr - fin.test$PRICE)^2) / mean((test.avg - fin.test$PRICE)^2)
tree.r2 <- 1 - mean((pred.tree - fin.test$PRICE)^2) / mean((test.avg - fin.test$PRICE)^2)
bag.r2 <- 1 - mean((pred.bag - fin.test$PRICE)^2) / mean((test.avg - fin.test$PRICE)^2)

print(lm.r2)
print(ridge.r2)
print(lasso.r2)
print(pls.r2)
print(pcr.r2)
print(tree.r2)
print(bag.r2)

all = c(lm.r2, ridge.r2, lasso.r2, pls.r2, pcr.r2, tree.r2, bag.r2)
names(all) = c("lm", "rid", "las", "pls", "pcr", "tree", "bag")
par(mfrow = c(1,2))
barplot(all,main = "test R-Squared",col = "green")

all2 = c(sqrt(MSE.lm), sqrt(MSE.ridge), sqrt(MSE.lasso), sqrt(MSE.pls), sqrt(MSE.pcr),
sqrt(MSE.tree), sqrt(MSE.bag))
names(all2) = c("lm", "rid", "las", "pls", "pcr", "tree", "bag")
barplot(all2,main = "test MSE",col = "red")
```