Data Mining & Machine Learning 1

Final Project

Appendix

Submitted to:

Luis Gustavo Nardin

Submitted by:

Sobil Dalal

x19148496

Master of Science in Data Analytics

National College of Ireland

Table of Contents

[Appendix 1 - King County house price prediction 4](#_Toc39369017)

[Part 1 - Multi-Linear Regression Model 4](#_Toc39369018)

[Step 1 : Collecting Data 4](#_Toc39369019)

[Step 2 : Exploring, preprocessing and cleaning the data 4](#_Toc39369020)

[Step 3.1 - Data transformation - default price 7](#_Toc39369021)

[Step 4.1 : Training a model on the data - default price 8](#_Toc39369022)

[Step 5.1 : Evaluating the model 9](#_Toc39369023)

[Step 6.1 : Improving the model 10](#_Toc39369024)

[Step 3.2 - Data transformation - log(price) and using all variables 21](#_Toc39369025)

[Step 4.2 : Training a model on the data - log(price) and using all variables 24](#_Toc39369026)

[Step 5.2 : Evaluating the model - log(price) and using all variables 24](#_Toc39369027)

[Step 6.2 : Improving the model - log(price) and using all variables 25](#_Toc39369028)

[Step 3.3 - Data transformation - log(price) conversion and using only numeric type parameters 46](#_Toc39369029)

[Step 4.3 : Training a model on the data - log(price) conversion and using only numeric type parameters 46](#_Toc39369030)

[Step 5.3 : Evaluating the model - log(price) conversion and using only numeric type parameters 46](#_Toc39369031)

[Step 6.3 : Improving the model - log(price) conversion and using only numeric type parameters 49](#_Toc39369032)

[Part 2 - Decision Tree Regression Model 75](#_Toc39369033)

[Step 1 : Collecting Data 75](#_Toc39369034)

[Step 2 : Exploring, preprocessing and cleaning the data 75](#_Toc39369035)

[Step 3 - Data transformation & preparation 76](#_Toc39369036)

[Step 4 : Training a model 76](#_Toc39369037)

[Step 5 : Evaluating the model 76](#_Toc39369038)

[Step 6 : Improving the model 82](#_Toc39369039)

[Appendix 2 - House price prediction from 0.5 million record data set 83](#_Toc39369040)

[Step 1 : Collecting Data 83](#_Toc39369041)

[Step 2 : Exploring, preprocessing and cleaning the data 83](#_Toc39369042)

[Step 3 - Data transformation 88](#_Toc39369043)

[Step 4 : Training a model on the data 91](#_Toc39369044)

[Step 5 : Evaluating the model 91](#_Toc39369045)

[Step 6 : Improving the model 91](#_Toc39369046)

[Appendix 3 - Customer’s decision to open a term deposit account 101](#_Toc39369047)

[Part 1 - KNN Model 101](#_Toc39369048)

[Step 1 : Collecting Data 101](#_Toc39369049)

[Step 2 : Exploring, preprocessing and cleaning the data 101](#_Toc39369050)

[Step 3.1 - Data transformation & preparation - Normalizing the data with custom function 104](#_Toc39369051)

[Step 4.1 : Training a model on the data - Normalizing the data with custom function 106](#_Toc39369052)

[Step 5.1 : Evaluating the model - Normalizing the data with custom function 106](#_Toc39369053)

[Step 6.1 : Improving the model 107](#_Toc39369054)

[Step 3.2 - Data transformation & preparation - Normalizing the data with inbuilt R function 120](#_Toc39369055)

[Step 4.2 : Training a model on the data - Normalizing the data with inbuilt R function 124](#_Toc39369056)

[Step 5.2 : Evaluating the model - Normalizing the data with inbuilt R function 124](#_Toc39369057)

[Step 6.2 : Improving the model - Normalizing the data with inbuilt R function 124](#_Toc39369058)

[Step 3.3 - Data transformation & preparation - Only numeric predictors 134](#_Toc39369059)

[Step 4.3 : Training a model on the data - Only numeric predictors 136](#_Toc39369060)

[Step 5.3 : Evaluating the model - Only numeric predictors 136](#_Toc39369061)

[Step 6.3 : Improving the model - Only numeric predictors 137](#_Toc39369062)

[Part 2 - Naive Bayes 148](#_Toc39369063)

[Step 1 : Collecting Data 148](#_Toc39369064)

[Step 2 : Exploring, preprocessing and cleaning the data 148](#_Toc39369065)

[Step 3 - Data transformation & preparation 149](#_Toc39369066)

[Step 4 : Train the model 160](#_Toc39369067)

[Step 5 : Evaluate the model 160](#_Toc39369068)

[Step 6 : Improving the model 160](#_Toc39369069)

[Part 3 - Decision Tree 163](#_Toc39369070)

[Step 1 : Collecting Data 163](#_Toc39369071)

[Step 2 : Exploring, preprocessing and cleaning the data 163](#_Toc39369072)

[Step 3 - Data transformation & preparation 164](#_Toc39369073)

[Step 4 : Train the model 164](#_Toc39369074)

[Step 5 : Evaluate the model 164](#_Toc39369075)

[Step 6 : Improving model performance 165](#_Toc39369076)

[Part 4 - SVM model 174](#_Toc39369077)

[Step 1 : Collecting Data 174](#_Toc39369078)

[Step 2 : Exploring, preprocessing and cleaning the data 174](#_Toc39369079)

[Step 3 - Data transformation & preparation 175](#_Toc39369080)

[Step 4 : Training a model on the data 175](#_Toc39369081)

[Step 5 : Evaluating model performance 176](#_Toc39369082)

[Step 6 : Improving model performance 177](#_Toc39369083)

# Appendix 1 - King County house price prediction

## Part 1 - Multi-Linear Regression Model

### Step 1 : Collecting Data

The data set is collected in CSV format from kaggele and below is the reference for the same.

Harlfoxem. (2016) House sales in king county, usa. [Online]. Available:<https://www.kaggle.com/harlfoxem/housesalesprediction>

### Step 2 : Exploring, preprocessing and cleaning the data

Primary setup

knitr::opts\_knit$set(root.dir = '/Users/sobil/Documents/MSC/Sem 1/Data Mining & Machine Learning/Project/Kc\_house/')  
remove(list = ls())

#### 1) reading the raw csv file

kc\_house <- read.csv("kc\_house\_data.csv", stringsAsFactors = FALSE)

#### 2) exploratory analysis

Structure of the kc\_house data frame

str(kc\_house)

## 'data.frame': 21613 obs. of 21 variables:  
## $ id : num 7.13e+09 6.41e+09 5.63e+09 2.49e+09 1.95e+09 ...  
## $ date : chr "20141013T000000" "20141209T000000" "20150225T000000" "20141209T000000" ...  
## $ price : num 221900 538000 180000 604000 510000 ...  
## $ bedrooms : int 3 3 2 4 3 4 3 3 3 3 ...  
## $ bathrooms : num 1 2.25 1 3 2 4.5 2.25 1.5 1 2.5 ...  
## $ sqft\_living : int 1180 2570 770 1960 1680 5420 1715 1060 1780 1890 ...  
## $ sqft\_lot : int 5650 7242 10000 5000 8080 101930 6819 9711 7470 6560 ...  
## $ floors : num 1 2 1 1 1 1 2 1 1 2 ...  
## $ waterfront : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ view : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ condition : int 3 3 3 5 3 3 3 3 3 3 ...  
## $ grade : int 7 7 6 7 8 11 7 7 7 7 ...  
## $ sqft\_above : int 1180 2170 770 1050 1680 3890 1715 1060 1050 1890 ...  
## $ sqft\_basement: int 0 400 0 910 0 1530 0 0 730 0 ...  
## $ yr\_built : int 1955 1951 1933 1965 1987 2001 1995 1963 1960 2003 ...  
## $ yr\_renovated : int 0 1991 0 0 0 0 0 0 0 0 ...  
## $ zipcode : int 98178 98125 98028 98136 98074 98053 98003 98198 98146 98038 ...  
## $ lat : num 47.5 47.7 47.7 47.5 47.6 ...  
## $ long : num -122 -122 -122 -122 -122 ...  
## $ sqft\_living15: int 1340 1690 2720 1360 1800 4760 2238 1650 1780 2390 ...  
## $ sqft\_lot15 : int 5650 7639 8062 5000 7503 101930 6819 9711 8113 7570 ...

Summary of the kc\_house data frame

summary(kc\_house)

## id date price bedrooms   
## Min. :1.000e+06 Length:21613 Min. : 75000 Min. : 0.000   
## 1st Qu.:2.123e+09 Class :character 1st Qu.: 321950 1st Qu.: 3.000   
## Median :3.905e+09 Mode :character Median : 450000 Median : 3.000   
## Mean :4.580e+09 Mean : 540088 Mean : 3.371   
## 3rd Qu.:7.309e+09 3rd Qu.: 645000 3rd Qu.: 4.000   
## Max. :9.900e+09 Max. :7700000 Max. :33.000   
## bathrooms sqft\_living sqft\_lot floors   
## Min. :0.000 Min. : 290 Min. : 520 Min. :1.000   
## 1st Qu.:1.750 1st Qu.: 1427 1st Qu.: 5040 1st Qu.:1.000   
## Median :2.250 Median : 1910 Median : 7618 Median :1.500   
## Mean :2.115 Mean : 2080 Mean : 15107 Mean :1.494   
## 3rd Qu.:2.500 3rd Qu.: 2550 3rd Qu.: 10688 3rd Qu.:2.000   
## Max. :8.000 Max. :13540 Max. :1651359 Max. :3.500   
## waterfront view condition grade   
## Min. :0.000000 Min. :0.0000 Min. :1.000 Min. : 1.000   
## 1st Qu.:0.000000 1st Qu.:0.0000 1st Qu.:3.000 1st Qu.: 7.000   
## Median :0.000000 Median :0.0000 Median :3.000 Median : 7.000   
## Mean :0.007542 Mean :0.2343 Mean :3.409 Mean : 7.657   
## 3rd Qu.:0.000000 3rd Qu.:0.0000 3rd Qu.:4.000 3rd Qu.: 8.000   
## Max. :1.000000 Max. :4.0000 Max. :5.000 Max. :13.000   
## sqft\_above sqft\_basement yr\_built yr\_renovated   
## Min. : 290 Min. : 0.0 Min. :1900 Min. : 0.0   
## 1st Qu.:1190 1st Qu.: 0.0 1st Qu.:1951 1st Qu.: 0.0   
## Median :1560 Median : 0.0 Median :1975 Median : 0.0   
## Mean :1788 Mean : 291.5 Mean :1971 Mean : 84.4   
## 3rd Qu.:2210 3rd Qu.: 560.0 3rd Qu.:1997 3rd Qu.: 0.0   
## Max. :9410 Max. :4820.0 Max. :2015 Max. :2015.0   
## zipcode lat long sqft\_living15   
## Min. :98001 Min. :47.16 Min. :-122.5 Min. : 399   
## 1st Qu.:98033 1st Qu.:47.47 1st Qu.:-122.3 1st Qu.:1490   
## Median :98065 Median :47.57 Median :-122.2 Median :1840   
## Mean :98078 Mean :47.56 Mean :-122.2 Mean :1987   
## 3rd Qu.:98118 3rd Qu.:47.68 3rd Qu.:-122.1 3rd Qu.:2360   
## Max. :98199 Max. :47.78 Max. :-121.3 Max. :6210   
## sqft\_lot15   
## Min. : 651   
## 1st Qu.: 5100   
## Median : 7620   
## Mean : 12768   
## 3rd Qu.: 10083   
## Max. :871200

Plotting histogram of the house prices

library(ggplot2)  
library(ggthemes)  
theme\_set(theme\_gdocs())  
ggplot(kc\_house, aes(x=price)) + geom\_histogram(bins = 10, color = 'blue', aes(fill=..count..), alpha = 0.4) + xlab('House Price') + ylab('Count') + ggtitle('House Price distribution Plot')

A screenshot of a cell phone

Description automatically generated

#### 3) cleaning & removing the unrelated and insignificant columns

removing id

kc\_house <- kc\_house[,-c(1)]

removing the lat, long, zipcode

kc\_house <- kc\_house[,-c(18,16,17)]

sqft\_living, sqft\_lot are related and can calculate sqft\_living15, sqft\_lot15 resp.

kc\_house <- kc\_house[,-c(17,16)]

cheking NAs

apply(X = kc\_house,MARGIN = 2, FUN = function(col) any(is.na(col))) # no NAs

## date price bedrooms bathrooms sqft\_living   
## FALSE FALSE FALSE FALSE FALSE   
## sqft\_lot floors waterfront view condition   
## FALSE FALSE FALSE FALSE FALSE   
## grade sqft\_above sqft\_basement yr\_built yr\_renovated   
## FALSE FALSE FALSE FALSE FALSE

#### 4) exploring the colmuns for outliers and influential points

table(kc\_house$bedrooms)

##   
## 0 1 2 3 4 5 6 7 8 9 10 11 33   
## 13 199 2760 9824 6882 1601 272 38 13 6 3 1 1

33 bedroom seems to be a influential point

kc\_house <- kc\_house[kc\_house$bedrooms != 33,]

Column - bathrooms

table(kc\_house$bathrooms)

##   
## 0 0.5 0.75 1 1.25 1.5 1.75 2 2.25 2.5 2.75 3 3.25 3.5 3.75 4   
## 10 4 72 3852 9 1446 3047 1930 2047 5380 1185 753 589 731 155 136   
## 4.25 4.5 4.75 5 5.25 5.5 5.75 6 6.25 6.5 6.75 7.5 7.75 8   
## 79 100 23 21 13 10 4 6 2 2 2 1 1 2

Column - View

table(kc\_house$view)

##   
## 0 1 2 3 4   
## 19488 332 963 510 319

Resetting the row names to index

row.names(kc\_house) <- 1:21612

### Step 3.1 - Data transformation - default price

converting the date field type from chr to date

kc\_house$date <- as.Date(kc\_house$date, format = "%Y%m%d")  
summary(kc\_house$date)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## "2014-05-02" "2014-07-22" "2014-10-16" "2014-10-29" "2015-02-17" "2015-05-27"

Date is almost for the same duration of time, so the sample is unbiased remove date

kc\_house <- kc\_house[,-c(1)]

converting the waterfront to factors (0,1)

table(kc\_house$waterfront)

##   
## 0 1   
## 21449 163

kc\_house$waterfront <- as.factor(kc\_house$waterfront)

### Step 4.1 : Training a model on the data - default price

adding the libraries required

library(boot)  
library(car)

## Loading required package: carData

##   
## Attaching package: 'car'

## The following object is masked from 'package:boot':  
##   
## logit

library(leaps)  
library(psych)

##   
## Attaching package: 'psych'

## The following object is masked from 'package:car':  
##   
## logit

## The following object is masked from 'package:boot':  
##   
## logit

## The following objects are masked from 'package:ggplot2':  
##   
## %+%, alpha

checking correlation

pairs.panels(kc\_house)

A screenshot of a computer

Description automatically generated

Model the linear model simple model 1

house.fit1 <- glm(price ~ ., data = kc\_house)

### Step 5.1 : Evaluating the model

Cheking the summary of the model

summary(house.fit1)

##   
## Call:  
## glm(formula = price ~ ., data = kc\_house)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1333250 -109639 -9586 89402 4224404   
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6.219e+06 1.383e+05 44.966 < 2e-16 \*\*\*  
## bedrooms -4.257e+04 2.110e+03 -20.173 < 2e-16 \*\*\*  
## bathrooms 4.698e+04 3.499e+03 13.430 < 2e-16 \*\*\*  
## sqft\_living 1.739e+02 4.612e+00 37.707 < 2e-16 \*\*\*  
## sqft\_lot -2.607e-01 3.661e-02 -7.121 1.10e-12 \*\*\*  
## floors 2.413e+04 3.736e+03 6.458 1.08e-10 \*\*\*  
## waterfront1 5.724e+05 1.865e+04 30.695 < 2e-16 \*\*\*  
## view 4.491e+04 2.256e+03 19.908 < 2e-16 \*\*\*  
## condition 1.877e+04 2.498e+03 7.512 6.03e-14 \*\*\*  
## grade 1.238e+05 2.166e+03 57.132 < 2e-16 \*\*\*  
## sqft\_above -1.547e+00 4.455e+00 -0.347 0.7284   
## sqft\_basement NA NA NA NA   
## yr\_built -3.578e+03 7.087e+01 -50.490 < 2e-16 \*\*\*  
## yr\_renovated 8.362e+00 3.914e+00 2.136 0.0327 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 46805833545)  
##   
## Null deviance: 2.9129e+15 on 21611 degrees of freedom  
## Residual deviance: 1.0110e+15 on 21599 degrees of freedom  
## AIC: 592338  
##   
## Number of Fisher Scoring iterations: 2

### Step 6.1 : Improving the model

#### 1) Adding all the possible relation to the model

Model 2 - step backward

house.fit2 <- step(house.fit1)

## Start: AIC=592338.3  
## price ~ bedrooms + bathrooms + sqft\_living + sqft\_lot + floors +   
## waterfront + view + condition + grade + sqft\_above + sqft\_basement +   
## yr\_built + yr\_renovated  
##   
##   
## Step: AIC=592338.3  
## price ~ bedrooms + bathrooms + sqft\_living + sqft\_lot + floors +   
## waterfront + view + condition + grade + sqft\_above + yr\_built +   
## yr\_renovated  
##   
## Df Deviance AIC  
## - sqft\_above 1 1.0110e+15 592336  
## <none> 1.0110e+15 592338  
## - yr\_renovated 1 1.0112e+15 592341  
## - floors 1 1.0129e+15 592378  
## - sqft\_lot 1 1.0133e+15 592387  
## - condition 1 1.0136e+15 592393  
## - bathrooms 1 1.0194e+15 592516  
## - view 1 1.0295e+15 592729  
## - bedrooms 1 1.0300e+15 592740  
## - waterfront 1 1.0551e+15 593259  
## - sqft\_living 1 1.0775e+15 593714  
## - yr\_built 1 1.1303e+15 594747  
## - grade 1 1.1637e+15 595378  
##   
## Step: AIC=592336.5  
## price ~ bedrooms + bathrooms + sqft\_living + sqft\_lot + floors +   
## waterfront + view + condition + grade + yr\_built + yr\_renovated  
##   
## Df Deviance AIC  
## <none> 1.0110e+15 592336  
## - yr\_renovated 1 1.0112e+15 592339  
## - floors 1 1.0132e+15 592382  
## - sqft\_lot 1 1.0134e+15 592386  
## - condition 1 1.0136e+15 592391  
## - bathrooms 1 1.0198e+15 592522  
## - bedrooms 1 1.0300e+15 592738  
## - view 1 1.0301e+15 592740  
## - waterfront 1 1.0551e+15 593257  
## - yr\_built 1 1.1319e+15 594777  
## - sqft\_living 1 1.1400e+15 594930  
## - grade 1 1.1674e+15 595443

summary(house.fit2)

##   
## Call:  
## glm(formula = price ~ bedrooms + bathrooms + sqft\_living + sqft\_lot +   
## floors + waterfront + view + condition + grade + yr\_built +   
## yr\_renovated, data = kc\_house)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1333829 -109621 -9637 89527 4224470   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6.225e+06 1.372e+05 45.354 < 2e-16 \*\*\*  
## bedrooms -4.257e+04 2.110e+03 -20.173 < 2e-16 \*\*\*  
## bathrooms 4.720e+04 3.444e+03 13.705 < 2e-16 \*\*\*  
## sqft\_living 1.728e+02 3.291e+00 52.502 < 2e-16 \*\*\*  
## sqft\_lot -2.622e-01 3.638e-02 -7.206 5.93e-13 \*\*\*  
## floors 2.361e+04 3.425e+03 6.893 5.61e-12 \*\*\*  
## waterfront1 5.722e+05 1.864e+04 30.700 < 2e-16 \*\*\*  
## view 4.504e+04 2.226e+03 20.229 < 2e-16 \*\*\*  
## condition 1.882e+04 2.494e+03 7.549 4.55e-14 \*\*\*  
## grade 1.237e+05 2.139e+03 57.806 < 2e-16 \*\*\*  
## yr\_built -3.581e+03 7.045e+01 -50.834 < 2e-16 \*\*\*  
## yr\_renovated 8.360e+00 3.914e+00 2.136 0.0327 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 46803927929)  
##   
## Null deviance: 2.9129e+15 on 21611 degrees of freedom  
## Residual deviance: 1.0110e+15 on 21600 degrees of freedom  
## AIC: 592336  
##   
## Number of Fisher Scoring iterations: 2

Model 3 - adding interaction

house.fit3 <- update(house.fit2, ~ . + bedrooms:bathrooms + bedrooms:grade + bedrooms:sqft\_above + bedrooms:sqft\_living + bathrooms:grade + bathrooms:sqft\_above + sqft\_above:sqft\_living + grade:sqft\_living + bathrooms:sqft\_living + floors:sqft\_living + view:sqft\_living + floors:condition + floors:grade + floors:sqft\_above + waterfront:view + view:grade)  
summary(house.fit3)

##   
## Call:  
## glm(formula = price ~ bedrooms + bathrooms + sqft\_living + sqft\_lot +   
## floors + waterfront + view + condition + grade + yr\_built +   
## yr\_renovated + bedrooms:bathrooms + bedrooms:grade + bedrooms:sqft\_above +   
## bedrooms:sqft\_living + bathrooms:grade + bathrooms:sqft\_above +   
## sqft\_living:sqft\_above + sqft\_living:grade + bathrooms:sqft\_living +   
## sqft\_living:floors + sqft\_living:view + floors:condition +   
## floors:grade + floors:sqft\_above + waterfront:view + view:grade,   
## data = kc\_house)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3001706 -103399 -11999 83139 3054376   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.716e+06 1.375e+05 41.564 < 2e-16 \*\*\*  
## bedrooms -3.110e+04 1.394e+04 -2.231 0.025709 \*   
## bathrooms -6.255e+04 2.517e+04 -2.485 0.012954 \*   
## sqft\_living -6.570e+01 2.229e+01 -2.947 0.003210 \*\*   
## sqft\_lot -2.723e-01 3.425e-02 -7.951 1.93e-15 \*\*\*  
## floors 1.035e+03 3.284e+04 0.032 0.974851   
## waterfront1 3.368e+05 1.080e+05 3.120 0.001813 \*\*   
## view -1.058e+05 1.331e+04 -7.950 1.95e-15 \*\*\*  
## condition -1.935e+04 6.975e+03 -2.774 0.005548 \*\*   
## grade 1.932e+04 7.935e+03 2.434 0.014924 \*   
## yr\_built -2.908e+03 7.076e+01 -41.097 < 2e-16 \*\*\*  
## yr\_renovated 2.198e+01 3.668e+00 5.992 2.10e-09 \*\*\*  
## bedrooms:bathrooms 4.392e+02 2.876e+03 0.153 0.878633   
## bedrooms:grade 2.865e+03 2.327e+03 1.231 0.218324   
## bedrooms:sqft\_above 1.413e+01 3.542e+00 3.988 6.68e-05 \*\*\*  
## bedrooms:sqft\_living -1.365e+01 3.814e+00 -3.579 0.000346 \*\*\*  
## bathrooms:grade 1.202e+04 3.670e+03 3.276 0.001056 \*\*   
## bathrooms:sqft\_above 9.760e+00 6.220e+00 1.569 0.116638   
## sqft\_living:sqft\_above 1.589e-02 3.240e-03 4.903 9.48e-07 \*\*\*  
## sqft\_living:grade 2.554e+01 2.769e+00 9.224 < 2e-16 \*\*\*  
## bathrooms:sqft\_living -2.797e+00 4.820e+00 -0.580 0.561662   
## sqft\_living:floors 1.981e+01 7.309e+00 2.710 0.006736 \*\*   
## sqft\_living:view 3.290e+00 2.522e+00 1.305 0.192005   
## floors:condition 3.679e+04 5.154e+03 7.138 9.74e-13 \*\*\*  
## floors:grade 5.240e+03 4.043e+03 1.296 0.194987   
## floors:sqft\_above -9.259e+01 8.524e+00 -10.862 < 2e-16 \*\*\*  
## waterfront1:view 5.474e+04 2.844e+04 1.925 0.054250 .   
## view:grade 1.596e+04 2.075e+03 7.688 1.56e-14 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 40618975037)  
##   
## Null deviance: 2.9129e+15 on 21611 degrees of freedom  
## Residual deviance: 8.7672e+14 on 21584 degrees of freedom  
## AIC: 589289  
##   
## Number of Fisher Scoring iterations: 2

Model 4 - step backward

house.fit4 <- step(house.fit3)

## Start: AIC=589289.3  
## price ~ bedrooms + bathrooms + sqft\_living + sqft\_lot + floors +   
## waterfront + view + condition + grade + yr\_built + yr\_renovated +   
## bedrooms:bathrooms + bedrooms:grade + bedrooms:sqft\_above +   
## bedrooms:sqft\_living + bathrooms:grade + bathrooms:sqft\_above +   
## sqft\_living:sqft\_above + sqft\_living:grade + bathrooms:sqft\_living +   
## sqft\_living:floors + sqft\_living:view + floors:condition +   
## floors:grade + floors:sqft\_above + waterfront:view + view:grade  
##   
## Df Deviance AIC  
## - bedrooms:bathrooms 1 8.7672e+14 589287  
## - bathrooms:sqft\_living 1 8.7673e+14 589288  
## - bedrooms:grade 1 8.7678e+14 589289  
## - floors:grade 1 8.7679e+14 589289  
## - sqft\_living:view 1 8.7679e+14 589289  
## <none> 8.7672e+14 589289  
## - bathrooms:sqft\_above 1 8.7682e+14 589290  
## - waterfront:view 1 8.7687e+14 589291  
## - sqft\_living:floors 1 8.7702e+14 589295  
## - bathrooms:grade 1 8.7716e+14 589298  
## - bedrooms:sqft\_living 1 8.7724e+14 589300  
## - bedrooms:sqft\_above 1 8.7737e+14 589303  
## - sqft\_living:sqft\_above 1 8.7770e+14 589311  
## - yr\_renovated 1 8.7818e+14 589323  
## - floors:condition 1 8.7879e+14 589338  
## - view:grade 1 8.7912e+14 589346  
## - sqft\_lot 1 8.7929e+14 589351  
## - sqft\_living:grade 1 8.8018e+14 589372  
## - floors:sqft\_above 1 8.8151e+14 589405  
## - yr\_built 1 9.4533e+14 590916  
##   
## Step: AIC=589287.4  
## price ~ bedrooms + bathrooms + sqft\_living + sqft\_lot + floors +   
## waterfront + view + condition + grade + yr\_built + yr\_renovated +   
## bedrooms:grade + bedrooms:sqft\_above + bedrooms:sqft\_living +   
## bathrooms:grade + bathrooms:sqft\_above + sqft\_living:sqft\_above +   
## sqft\_living:grade + bathrooms:sqft\_living + sqft\_living:floors +   
## sqft\_living:view + floors:condition + floors:grade + floors:sqft\_above +   
## waterfront:view + view:grade  
##   
## Df Deviance AIC  
## - bathrooms:sqft\_living 1 8.7673e+14 589286  
## - bedrooms:grade 1 8.7678e+14 589287  
## - sqft\_living:view 1 8.7679e+14 589287  
## - floors:grade 1 8.7679e+14 589287  
## <none> 8.7672e+14 589287  
## - bathrooms:sqft\_above 1 8.7683e+14 589288  
## - waterfront:view 1 8.7687e+14 589289  
## - sqft\_living:floors 1 8.7702e+14 589293  
## - bathrooms:grade 1 8.7716e+14 589296  
## - bedrooms:sqft\_living 1 8.7733e+14 589300  
## - bedrooms:sqft\_above 1 8.7737e+14 589301  
## - sqft\_living:sqft\_above 1 8.7781e+14 589312  
## - yr\_renovated 1 8.7818e+14 589321  
## - floors:condition 1 8.7879e+14 589336  
## - view:grade 1 8.7913e+14 589345  
## - sqft\_lot 1 8.7929e+14 589349  
## - sqft\_living:grade 1 8.8021e+14 589371  
## - floors:sqft\_above 1 8.8151e+14 589403  
## - yr\_built 1 9.4566e+14 590921  
##   
## Step: AIC=589285.7  
## price ~ bedrooms + bathrooms + sqft\_living + sqft\_lot + floors +   
## waterfront + view + condition + grade + yr\_built + yr\_renovated +   
## bedrooms:grade + bedrooms:sqft\_above + bedrooms:sqft\_living +   
## bathrooms:grade + bathrooms:sqft\_above + sqft\_living:sqft\_above +   
## sqft\_living:grade + sqft\_living:floors + sqft\_living:view +   
## floors:condition + floors:grade + floors:sqft\_above + waterfront:view +   
## view:grade  
##   
## Df Deviance AIC  
## - sqft\_living:view 1 8.7679e+14 589285  
## - bedrooms:grade 1 8.7681e+14 589285  
## <none> 8.7673e+14 589286  
## - floors:grade 1 8.7682e+14 589286  
## - bathrooms:sqft\_above 1 8.7685e+14 589287  
## - waterfront:view 1 8.7688e+14 589287  
## - sqft\_living:floors 1 8.7704e+14 589291  
## - bathrooms:grade 1 8.7716e+14 589294  
## - bedrooms:sqft\_above 1 8.7762e+14 589305  
## - bedrooms:sqft\_living 1 8.7772e+14 589308  
## - sqft\_living:sqft\_above 1 8.7781e+14 589310  
## - yr\_renovated 1 8.7819e+14 589319  
## - floors:condition 1 8.7882e+14 589335  
## - view:grade 1 8.7926e+14 589346  
## - sqft\_lot 1 8.7930e+14 589347  
## - sqft\_living:grade 1 8.8021e+14 589369  
## - floors:sqft\_above 1 8.8217e+14 589417  
## - yr\_built 1 9.4567e+14 590919  
##   
## Step: AIC=589285.1  
## price ~ bedrooms + bathrooms + sqft\_living + sqft\_lot + floors +   
## waterfront + view + condition + grade + yr\_built + yr\_renovated +   
## bedrooms:grade + bedrooms:sqft\_above + bedrooms:sqft\_living +   
## bathrooms:grade + bathrooms:sqft\_above + sqft\_living:sqft\_above +   
## sqft\_living:grade + sqft\_living:floors + floors:condition +   
## floors:grade + floors:sqft\_above + waterfront:view + view:grade  
##   
## Df Deviance AIC  
## - floors:grade 1 8.7687e+14 589285  
## <none> 8.7679e+14 589285  
## - bedrooms:grade 1 8.7687e+14 589285  
## - bathrooms:sqft\_above 1 8.7691e+14 589286  
## - waterfront:view 1 8.7695e+14 589287  
## - sqft\_living:floors 1 8.7712e+14 589291  
## - bathrooms:grade 1 8.7721e+14 589293  
## - bedrooms:sqft\_above 1 8.7764e+14 589304  
## - bedrooms:sqft\_living 1 8.7776e+14 589307  
## - sqft\_living:sqft\_above 1 8.7803e+14 589314  
## - yr\_renovated 1 8.7825e+14 589319  
## - floors:condition 1 8.7888e+14 589334  
## - sqft\_lot 1 8.7937e+14 589347  
## - sqft\_living:grade 1 8.8023e+14 589368  
## - floors:sqft\_above 1 8.8242e+14 589421  
## - view:grade 1 8.8392e+14 589458  
## - yr\_built 1 9.4573e+14 590919  
##   
## Step: AIC=589285.1  
## price ~ bedrooms + bathrooms + sqft\_living + sqft\_lot + floors +   
## waterfront + view + condition + grade + yr\_built + yr\_renovated +   
## bedrooms:grade + bedrooms:sqft\_above + bedrooms:sqft\_living +   
## bathrooms:grade + bathrooms:sqft\_above + sqft\_living:sqft\_above +   
## sqft\_living:grade + sqft\_living:floors + floors:condition +   
## floors:sqft\_above + waterfront:view + view:grade  
##   
## Df Deviance AIC  
## - bedrooms:grade 1 8.7695e+14 589285  
## <none> 8.7687e+14 589285  
## - bathrooms:sqft\_above 1 8.7696e+14 589285  
## - waterfront:view 1 8.7704e+14 589287  
## - sqft\_living:floors 1 8.7733e+14 589294  
## - bathrooms:grade 1 8.7746e+14 589297  
## - bedrooms:sqft\_above 1 8.7768e+14 589303  
## - bedrooms:sqft\_living 1 8.7780e+14 589306  
## - sqft\_living:sqft\_above 1 8.7812e+14 589314  
## - yr\_renovated 1 8.7832e+14 589319  
## - floors:condition 1 8.7889e+14 589333  
## - sqft\_lot 1 8.7944e+14 589346  
## - sqft\_living:grade 1 8.8033e+14 589368  
## - floors:sqft\_above 1 8.8251e+14 589422  
## - view:grade 1 8.8393e+14 589456  
## - yr\_built 1 9.4608e+14 590925  
##   
## Step: AIC=589284.9  
## price ~ bedrooms + bathrooms + sqft\_living + sqft\_lot + floors +   
## waterfront + view + condition + grade + yr\_built + yr\_renovated +   
## bedrooms:sqft\_above + bedrooms:sqft\_living + bathrooms:grade +   
## bathrooms:sqft\_above + sqft\_living:sqft\_above + sqft\_living:grade +   
## sqft\_living:floors + floors:condition + floors:sqft\_above +   
## waterfront:view + view:grade  
##   
## Df Deviance AIC  
## - bathrooms:sqft\_above 1 8.7700e+14 589284  
## <none> 8.7695e+14 589285  
## - waterfront:view 1 8.7711e+14 589287  
## - sqft\_living:floors 1 8.7739e+14 589294  
## - bathrooms:grade 1 8.7767e+14 589301  
## - bedrooms:sqft\_living 1 8.7781e+14 589304  
## - bedrooms:sqft\_above 1 8.7800e+14 589309  
## - sqft\_living:sqft\_above 1 8.7812e+14 589312  
## - yr\_renovated 1 8.7840e+14 589319  
## - floors:condition 1 8.7897e+14 589333  
## - sqft\_lot 1 8.7953e+14 589347  
## - sqft\_living:grade 1 8.8102e+14 589383  
## - floors:sqft\_above 1 8.8251e+14 589420  
## - view:grade 1 8.8394e+14 589455  
## - yr\_built 1 9.4615e+14 590924  
##   
## Step: AIC=589284.3  
## price ~ bedrooms + bathrooms + sqft\_living + sqft\_lot + floors +   
## waterfront + view + condition + grade + yr\_built + yr\_renovated +   
## bedrooms:sqft\_above + bedrooms:sqft\_living + bathrooms:grade +   
## sqft\_living:sqft\_above + sqft\_living:grade + sqft\_living:floors +   
## floors:condition + floors:sqft\_above + waterfront:view +   
## view:grade  
##   
## Df Deviance AIC  
## <none> 8.7700e+14 589284  
## - waterfront:view 1 8.7717e+14 589286  
## - sqft\_living:floors 1 8.7741e+14 589292  
## - bedrooms:sqft\_living 1 8.7791e+14 589305  
## - bedrooms:sqft\_above 1 8.7825e+14 589313  
## - yr\_renovated 1 8.7844e+14 589318  
## - floors:condition 1 8.7902e+14 589332  
## - bathrooms:grade 1 8.7909e+14 589334  
## - sqft\_lot 1 8.7959e+14 589346  
## - sqft\_living:sqft\_above 1 8.8056e+14 589370  
## - sqft\_living:grade 1 8.8174e+14 589399  
## - floors:sqft\_above 1 8.8259e+14 589419  
## - view:grade 1 8.8404e+14 589455  
## - yr\_built 1 9.4657e+14 590932

summary(house.fit4)

##   
## Call:  
## glm(formula = price ~ bedrooms + bathrooms + sqft\_living + sqft\_lot +   
## floors + waterfront + view + condition + grade + yr\_built +   
## yr\_renovated + bedrooms:sqft\_above + bedrooms:sqft\_living +   
## bathrooms:grade + sqft\_living:sqft\_above + sqft\_living:grade +   
## sqft\_living:floors + floors:condition + floors:sqft\_above +   
## waterfront:view + view:grade, data = kc\_house)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2997702 -103016 -11864 83295 3035153   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.669e+06 1.318e+05 43.009 < 2e-16 \*\*\*  
## bedrooms -1.499e+04 4.625e+03 -3.242 0.00119 \*\*   
## bathrooms -8.380e+04 1.843e+04 -4.548 5.44e-06 \*\*\*  
## sqft\_living -7.516e+01 1.678e+01 -4.481 7.48e-06 \*\*\*  
## sqft\_lot -2.732e-01 3.423e-02 -7.980 1.54e-15 \*\*\*  
## floors 3.257e+04 1.851e+04 1.760 0.07848 .   
## waterfront1 3.275e+05 1.078e+05 3.039 0.00238 \*\*   
## view -1.113e+05 1.158e+04 -9.615 < 2e-16 \*\*\*  
## condition -1.828e+04 6.918e+03 -2.642 0.00825 \*\*   
## grade 2.817e+04 4.598e+03 6.126 9.18e-10 \*\*\*  
## yr\_built -2.911e+03 7.034e+01 -41.384 < 2e-16 \*\*\*  
## yr\_renovated 2.183e+01 3.664e+00 5.957 2.61e-09 \*\*\*  
## bedrooms:sqft\_above 1.637e+01 2.950e+00 5.550 2.89e-08 \*\*\*  
## bedrooms:sqft\_living -1.254e+01 2.651e+00 -4.732 2.24e-06 \*\*\*  
## bathrooms:grade 1.655e+04 2.309e+03 7.166 7.96e-13 \*\*\*  
## sqft\_living:sqft\_above 1.806e-02 1.930e-03 9.356 < 2e-16 \*\*\*  
## sqft\_living:grade 2.488e+01 2.305e+00 10.798 < 2e-16 \*\*\*  
## sqft\_living:floors 1.994e+01 6.295e+00 3.167 0.00154 \*\*   
## floors:condition 3.602e+04 5.111e+03 7.048 1.88e-12 \*\*\*  
## floors:sqft\_above -8.625e+01 7.357e+00 -11.723 < 2e-16 \*\*\*  
## waterfront1:view 5.746e+04 2.837e+04 2.026 0.04281 \*   
## view:grade 1.766e+04 1.341e+03 13.166 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 40620783423)  
##   
## Null deviance: 2.9129e+15 on 21611 degrees of freedom  
## Residual deviance: 8.7700e+14 on 21590 degrees of freedom  
## AIC: 589284  
##   
## Number of Fisher Scoring iterations: 2

Model 5 - removing insigificant floors predictor

house.fit5 <- update(house.fit2, ~ . - floors)  
summary(house.fit5)

##   
## Call:  
## glm(formula = price ~ bedrooms + bathrooms + sqft\_living + sqft\_lot +   
## waterfront + view + condition + grade + yr\_built + yr\_renovated,   
## data = kc\_house)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1333039 -111146 -9348 90687 4226547   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6.044e+06 1.349e+05 44.814 < 2e-16 \*\*\*  
## bedrooms -4.318e+04 2.111e+03 -20.460 < 2e-16 \*\*\*  
## bathrooms 5.268e+04 3.354e+03 15.706 < 2e-16 \*\*\*  
## sqft\_living 1.712e+02 3.287e+00 52.096 < 2e-16 \*\*\*  
## sqft\_lot -2.778e-01 3.635e-02 -7.642 2.23e-14 \*\*\*  
## waterfront1 5.749e+05 1.866e+04 30.820 < 2e-16 \*\*\*  
## view 4.425e+04 2.226e+03 19.878 < 2e-16 \*\*\*  
## condition 1.674e+04 2.478e+03 6.756 1.46e-11 \*\*\*  
## grade 1.262e+05 2.109e+03 59.858 < 2e-16 \*\*\*  
## yr\_built -3.481e+03 6.901e+01 -50.441 < 2e-16 \*\*\*  
## yr\_renovated 9.696e+00 3.914e+00 2.477 0.0132 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 46904710295)  
##   
## Null deviance: 2.9129e+15 on 21611 degrees of freedom  
## Residual deviance: 1.0132e+15 on 21601 degrees of freedom  
## AIC: 592382  
##   
## Number of Fisher Scoring iterations: 2

#### 2) check the best fit by ignoring uncorrelated variables

checking the best predictors to predict the house price

bstFits1 <- regsubsets(price ~ bedrooms+bathrooms+sqft\_living+sqft\_lot+floors+waterfront+view+condition+grade+sqft\_above+sqft\_basement+yr\_built+yr\_renovated, data = kc\_house, nbest = 1, nvmax = 6)

## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax, force.in =  
## force.in, : 1 linear dependencies found

## Reordering variables and trying again:

par(mfrow = c(1,1))  
#subsets(bstFits1, statistic = "adjr2", max.size = 6)  
plot(bstFits1, scale = "adjr2")

A screenshot of a cell phone

Description automatically generated

Doing cross-validation for 4 predictors using K-Fold

house.k10.4 <- glm(price ~ sqft\_living + waterfront + yr\_renovated + grade, data = kc\_house)  
summary(house.k10.4) # 596532

##   
## Call:  
## glm(formula = price ~ sqft\_living + waterfront + yr\_renovated +   
## grade, data = kc\_house)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1342077 -132453 -21816 99973 4761201   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -5.970e+05 1.268e+04 -47.09 <2e-16 \*\*\*  
## sqft\_living 1.737e+02 2.741e+00 63.37 <2e-16 \*\*\*  
## waterfront1 7.950e+05 1.892e+04 42.02 <2e-16 \*\*\*  
## yr\_renovated 7.346e+01 4.063e+00 18.08 <2e-16 \*\*\*  
## grade 9.973e+04 2.135e+03 46.70 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 56850059198)  
##   
## Null deviance: 2.9129e+15 on 21611 degrees of freedom  
## Residual deviance: 1.2284e+15 on 21607 degrees of freedom  
## AIC: 596532  
##   
## Number of Fisher Scoring iterations: 2

house.k10.4.err <- cv.glm(data = kc\_house,glmfit = house.k10.4, K = 10)  
round(house.k10.4.err$delta[1], 4) # 56952103727

## [1] 57022968716

Checking the r square and adj. r square values

house.k10.4.lm <- lm(price ~ sqft\_living + waterfront + yr\_renovated + grade, data = kc\_house)  
summary(house.k10.4.lm)

##   
## Call:  
## lm(formula = price ~ sqft\_living + waterfront + yr\_renovated +   
## grade, data = kc\_house)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1342077 -132453 -21816 99973 4761201   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -5.970e+05 1.268e+04 -47.09 <2e-16 \*\*\*  
## sqft\_living 1.737e+02 2.741e+00 63.37 <2e-16 \*\*\*  
## waterfront1 7.950e+05 1.892e+04 42.02 <2e-16 \*\*\*  
## yr\_renovated 7.346e+01 4.063e+00 18.08 <2e-16 \*\*\*  
## grade 9.973e+04 2.135e+03 46.70 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 238400 on 21607 degrees of freedom  
## Multiple R-squared: 0.5783, Adjusted R-squared: 0.5782   
## F-statistic: 7408 on 4 and 21607 DF, p-value: < 2.2e-16

Plot the model

par(mfrow = c(2,2))  
plot(house.k10.4)

A close up of text on a white background

Description automatically generated

### Step 3.2 - Data transformation - log(price) and using all variables

After looking at the results of the plot and previously from histogram in step 2 - taking log of the price

checking distribution & outliers of dependent value

par(mfrow = c(1,2))  
hist(kc\_house$price,  
 col="orange",   
 border="black",  
 prob = TRUE,  
 xlab = "House prices",  
 main = "Histogram")  
lines(density(kc\_house$price),  
 lwd = 2,  
 col = "chocolate3") # not distributed normally - right skewed  
plot(kc\_house$price, main = "Scatter plot", ylab = "House prices")

A close up of text on a white background

Description automatically generated

# covert to log  
hist(log(kc\_house$price),  
 col="orange",   
 border="black",  
 prob = TRUE,  
 xlab = "House prices",  
 main = "Histogram")  
lines(density(log(kc\_house$price)),  
 lwd = 2,  
 col = "chocolate3") # not distributed normally  
plot(log(kc\_house$price), main = "Scatter plot", ylab = "House prices")

A close up of text on a black background

Description automatically generated

creating a new data frame - coverting price to log(price)

kc\_house\_log <- kc\_house   
kc\_house\_log$price <- log(kc\_house\_log$price)  
summary(kc\_house\_log)

## price bedrooms bathrooms sqft\_living   
## Min. :11.23 Min. : 0.000 Min. :0.000 Min. : 290   
## 1st Qu.:12.68 1st Qu.: 3.000 1st Qu.:1.750 1st Qu.: 1426   
## Median :13.02 Median : 3.000 Median :2.250 Median : 1910   
## Mean :13.05 Mean : 3.369 Mean :2.115 Mean : 2080   
## 3rd Qu.:13.38 3rd Qu.: 4.000 3rd Qu.:2.500 3rd Qu.: 2550   
## Max. :15.86 Max. :11.000 Max. :8.000 Max. :13540   
## sqft\_lot floors waterfront view condition   
## Min. : 520 Min. :1.000 0:21449 Min. :0.0000 Min. :1.000   
## 1st Qu.: 5040 1st Qu.:1.000 1: 163 1st Qu.:0.0000 1st Qu.:3.000   
## Median : 7619 Median :1.500 Median :0.0000 Median :3.000   
## Mean : 15107 Mean :1.494 Mean :0.2343 Mean :3.409   
## 3rd Qu.: 10688 3rd Qu.:2.000 3rd Qu.:0.0000 3rd Qu.:4.000   
## Max. :1651359 Max. :3.500 Max. :4.0000 Max. :5.000   
## grade sqft\_above sqft\_basement yr\_built   
## Min. : 1.000 Min. : 290 Min. : 0.0 Min. :1900   
## 1st Qu.: 7.000 1st Qu.:1190 1st Qu.: 0.0 1st Qu.:1951   
## Median : 7.000 Median :1560 Median : 0.0 Median :1975   
## Mean : 7.657 Mean :1788 Mean : 291.5 Mean :1971   
## 3rd Qu.: 8.000 3rd Qu.:2210 3rd Qu.: 560.0 3rd Qu.:1997   
## Max. :13.000 Max. :9410 Max. :4820.0 Max. :2015   
## yr\_renovated   
## Min. : 0.00   
## 1st Qu.: 0.00   
## Median : 0.00   
## Mean : 84.41   
## 3rd Qu.: 0.00   
## Max. :2015.00

### Step 4.2 : Training a model on the data - log(price) and using all variables

checking correlation

pairs.panels(kc\_house\_log)

A screenshot of a computer

Description automatically generated

model 1

house.glm1 <- glm(price ~ bedrooms+bathrooms+sqft\_living+sqft\_lot+floors+waterfront+view+condition+grade+sqft\_above+sqft\_basement+yr\_built+yr\_renovated, data = kc\_house\_log)

### Step 5.2 : Evaluating the model - log(price) and using all variables

summary(house.glm1)

##   
## Call:  
## glm(formula = price ~ bedrooms + bathrooms + sqft\_living + sqft\_lot +   
## floors + waterfront + view + condition + grade + sqft\_above +   
## sqft\_basement + yr\_built + yr\_renovated, data = kc\_house\_log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.87591 -0.21125 0.01574 0.21034 1.40554   
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.115e+01 2.001e-01 105.713 <2e-16 \*\*\*  
## bedrooms -2.540e-02 3.054e-03 -8.318 <2e-16 \*\*\*  
## bathrooms 7.598e-02 5.062e-03 15.008 <2e-16 \*\*\*  
## sqft\_living 2.140e-04 6.674e-06 32.064 <2e-16 \*\*\*  
## sqft\_lot -2.477e-08 5.298e-08 -0.467 0.6402   
## floors 1.044e-01 5.406e-03 19.307 <2e-16 \*\*\*  
## waterfront1 3.309e-01 2.698e-02 12.264 <2e-16 \*\*\*  
## view 4.709e-02 3.264e-03 14.426 <2e-16 \*\*\*  
## condition 4.089e-02 3.615e-03 11.311 <2e-16 \*\*\*  
## grade 2.271e-01 3.135e-03 72.434 <2e-16 \*\*\*  
## sqft\_above -6.335e-05 6.446e-06 -9.829 <2e-16 \*\*\*  
## sqft\_basement NA NA NA NA   
## yr\_built -5.358e-03 1.025e-04 -52.252 <2e-16 \*\*\*  
## yr\_renovated 9.325e-06 5.664e-06 1.646 0.0997 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.09799911)  
##   
## Null deviance: 5995.0 on 21611 degrees of freedom  
## Residual deviance: 2116.7 on 21599 degrees of freedom  
## AIC: 11147  
##   
## Number of Fisher Scoring iterations: 2

### Step 6.2 : Improving the model - log(price) and using all variables

#### 1) Adding all the possible relation to the model

model 2

house.glm2 <- update(house.glm1, ~ . - sqft\_basement)  
summary(house.glm2)

##   
## Call:  
## glm(formula = price ~ bedrooms + bathrooms + sqft\_living + sqft\_lot +   
## floors + waterfront + view + condition + grade + sqft\_above +   
## yr\_built + yr\_renovated, data = kc\_house\_log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.87591 -0.21125 0.01574 0.21034 1.40554   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.115e+01 2.001e-01 105.713 <2e-16 \*\*\*  
## bedrooms -2.540e-02 3.054e-03 -8.318 <2e-16 \*\*\*  
## bathrooms 7.598e-02 5.062e-03 15.008 <2e-16 \*\*\*  
## sqft\_living 2.140e-04 6.674e-06 32.064 <2e-16 \*\*\*  
## sqft\_lot -2.477e-08 5.298e-08 -0.467 0.6402   
## floors 1.044e-01 5.406e-03 19.307 <2e-16 \*\*\*  
## waterfront1 3.309e-01 2.698e-02 12.264 <2e-16 \*\*\*  
## view 4.709e-02 3.264e-03 14.426 <2e-16 \*\*\*  
## condition 4.089e-02 3.615e-03 11.311 <2e-16 \*\*\*  
## grade 2.271e-01 3.135e-03 72.434 <2e-16 \*\*\*  
## sqft\_above -6.335e-05 6.446e-06 -9.829 <2e-16 \*\*\*  
## yr\_built -5.358e-03 1.025e-04 -52.252 <2e-16 \*\*\*  
## yr\_renovated 9.325e-06 5.664e-06 1.646 0.0997 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.09799911)  
##   
## Null deviance: 5995.0 on 21611 degrees of freedom  
## Residual deviance: 2116.7 on 21599 degrees of freedom  
## AIC: 11147  
##   
## Number of Fisher Scoring iterations: 2

model 3

house.glm3 <- update(house.glm2, ~ . - sqft\_lot)  
summary(house.glm3)

##   
## Call:  
## glm(formula = price ~ bedrooms + bathrooms + sqft\_living + floors +   
## waterfront + view + condition + grade + sqft\_above + yr\_built +   
## yr\_renovated, data = kc\_house\_log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.88039 -0.21112 0.01575 0.21031 1.40560   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.116e+01 2.001e-01 105.755 <2e-16 \*\*\*  
## bedrooms -2.529e-02 3.044e-03 -8.308 <2e-16 \*\*\*  
## bathrooms 7.602e-02 5.062e-03 15.018 <2e-16 \*\*\*  
## sqft\_living 2.139e-04 6.670e-06 32.068 <2e-16 \*\*\*  
## floors 1.046e-01 5.378e-03 19.455 <2e-16 \*\*\*  
## waterfront1 3.311e-01 2.698e-02 12.272 <2e-16 \*\*\*  
## view 4.703e-02 3.261e-03 14.419 <2e-16 \*\*\*  
## condition 4.088e-02 3.615e-03 11.308 <2e-16 \*\*\*  
## grade 2.271e-01 3.131e-03 72.533 <2e-16 \*\*\*  
## sqft\_above -6.369e-05 6.405e-06 -9.945 <2e-16 \*\*\*  
## yr\_built -5.360e-03 1.025e-04 -52.296 <2e-16 \*\*\*  
## yr\_renovated 9.303e-06 5.664e-06 1.643 0.1   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.09799556)  
##   
## Null deviance: 5995.0 on 21611 degrees of freedom  
## Residual deviance: 2116.7 on 21600 degrees of freedom  
## AIC: 11145  
##   
## Number of Fisher Scoring iterations: 2

model 4

house.glm4 <- update(house.glm3, ~ . - yr\_renovated)  
summary(house.glm4)

##   
## Call:  
## glm(formula = price ~ bedrooms + bathrooms + sqft\_living + floors +   
## waterfront + view + condition + grade + sqft\_above + yr\_built,   
## data = kc\_house\_log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.88657 -0.21126 0.01605 0.21025 1.40499   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.126e+01 1.891e-01 112.472 <2e-16 \*\*\*  
## bedrooms -2.544e-02 3.042e-03 -8.363 <2e-16 \*\*\*  
## bathrooms 7.712e-02 5.017e-03 15.372 <2e-16 \*\*\*  
## sqft\_living 2.138e-04 6.670e-06 32.056 <2e-16 \*\*\*  
## floors 1.050e-01 5.373e-03 19.546 <2e-16 \*\*\*  
## waterfront1 3.337e-01 2.694e-02 12.386 <2e-16 \*\*\*  
## view 4.715e-02 3.261e-03 14.459 <2e-16 \*\*\*  
## condition 3.991e-02 3.566e-03 11.190 <2e-16 \*\*\*  
## grade 2.272e-01 3.131e-03 72.548 <2e-16 \*\*\*  
## sqft\_above -6.367e-05 6.405e-06 -9.940 <2e-16 \*\*\*  
## yr\_built -5.414e-03 9.712e-05 -55.741 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.09800327)  
##   
## Null deviance: 5995 on 21611 degrees of freedom  
## Residual deviance: 2117 on 21601 degrees of freedom  
## AIC: 11146  
##   
## Number of Fisher Scoring iterations: 2

adding interaction by checking the graph model 5 - sqft\_living interaction

house.glm5 <- update(house.glm4, ~ . + sqft\_above:sqft\_living + grade:sqft\_living + bathrooms:sqft\_living + floors:sqft\_living + view:sqft\_living)  
summary(house.glm5)

##   
## Call:  
## glm(formula = price ~ bedrooms + bathrooms + sqft\_living + floors +   
## waterfront + view + condition + grade + sqft\_above + yr\_built +   
## sqft\_living:sqft\_above + sqft\_living:grade + bathrooms:sqft\_living +   
## sqft\_living:floors + sqft\_living:view, data = kc\_house\_log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.33853 -0.20881 0.01564 0.20800 1.44116   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.135e+01 1.914e-01 111.534 < 2e-16 \*\*\*  
## bedrooms -3.738e-02 3.144e-03 -11.889 < 2e-16 \*\*\*  
## bathrooms 7.743e-02 9.722e-03 7.964 1.74e-15 \*\*\*  
## sqft\_living 4.128e-04 1.799e-05 22.947 < 2e-16 \*\*\*  
## floors 1.400e-01 1.250e-02 11.204 < 2e-16 \*\*\*  
## waterfront1 3.507e-01 2.682e-02 13.076 < 2e-16 \*\*\*  
## view 8.881e-02 7.575e-03 11.723 < 2e-16 \*\*\*  
## condition 3.644e-02 3.556e-03 10.248 < 2e-16 \*\*\*  
## grade 2.687e-01 6.840e-03 39.289 < 2e-16 \*\*\*  
## sqft\_above -6.156e-05 1.094e-05 -5.627 1.86e-08 \*\*\*  
## yr\_built -5.640e-03 9.883e-05 -57.067 < 2e-16 \*\*\*  
## sqft\_living:sqft\_above 7.185e-09 3.254e-09 2.208 0.0273 \*   
## sqft\_living:grade -1.937e-05 2.564e-06 -7.555 4.36e-14 \*\*\*  
## bathrooms:sqft\_living -1.711e-06 3.546e-06 -0.483 0.6294   
## sqft\_living:floors -2.326e-05 5.885e-06 -3.952 7.78e-05 \*\*\*  
## sqft\_living:view -1.399e-05 2.462e-06 -5.684 1.34e-08 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.09670111)  
##   
## Null deviance: 5995.0 on 21611 degrees of freedom  
## Residual deviance: 2088.4 on 21596 degrees of freedom  
## AIC: 10862  
##   
## Number of Fisher Scoring iterations: 2

model 6

house.glm6 <- update(house.glm5, ~ . - bathrooms:sqft\_living)  
summary(house.glm6)

##   
## Call:  
## glm(formula = price ~ bedrooms + bathrooms + sqft\_living + floors +   
## waterfront + view + condition + grade + sqft\_above + yr\_built +   
## sqft\_living:sqft\_above + sqft\_living:grade + sqft\_living:floors +   
## sqft\_living:view, data = kc\_house\_log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.35766 -0.20882 0.01566 0.20816 1.44067   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.134e+01 1.899e-01 112.375 < 2e-16 \*\*\*  
## bedrooms -3.745e-02 3.141e-03 -11.925 < 2e-16 \*\*\*  
## bathrooms 7.341e-02 4.993e-03 14.702 < 2e-16 \*\*\*  
## sqft\_living 4.135e-04 1.794e-05 23.046 < 2e-16 \*\*\*  
## floors 1.407e-01 1.241e-02 11.345 < 2e-16 \*\*\*  
## waterfront1 3.509e-01 2.681e-02 13.086 < 2e-16 \*\*\*  
## view 8.918e-02 7.537e-03 11.832 < 2e-16 \*\*\*  
## condition 3.654e-02 3.551e-03 10.288 < 2e-16 \*\*\*  
## grade 2.695e-01 6.625e-03 40.684 < 2e-16 \*\*\*  
## sqft\_above -5.923e-05 9.821e-06 -6.031 1.65e-09 \*\*\*  
## yr\_built -5.636e-03 9.839e-05 -57.277 < 2e-16 \*\*\*  
## sqft\_living:sqft\_above 6.356e-09 2.764e-09 2.300 0.0215 \*   
## sqft\_living:grade -1.969e-05 2.475e-06 -7.955 1.88e-15 \*\*\*  
## sqft\_living:floors -2.356e-05 5.851e-06 -4.026 5.69e-05 \*\*\*  
## sqft\_living:view -1.414e-05 2.441e-06 -5.794 6.95e-09 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.09669768)  
##   
## Null deviance: 5995.0 on 21611 degrees of freedom  
## Residual deviance: 2088.4 on 21597 degrees of freedom  
## AIC: 10860  
##   
## Number of Fisher Scoring iterations: 2

model 7 - bedrooms interaction

house.glm7 <- update(house.glm6, ~ . + bedrooms:bathrooms + bedrooms:grade + bedrooms:sqft\_above)  
summary(house.glm7)

##   
## Call:  
## glm(formula = price ~ bedrooms + bathrooms + sqft\_living + floors +   
## waterfront + view + condition + grade + sqft\_above + yr\_built +   
## sqft\_living:sqft\_above + sqft\_living:grade + sqft\_living:floors +   
## sqft\_living:view + bedrooms:bathrooms + bedrooms:grade +   
## bedrooms:sqft\_above, data = kc\_house\_log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.45149 -0.20796 0.01531 0.20822 1.46309   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.146e+01 1.950e-01 110.016 < 2e-16 \*\*\*  
## bedrooms -1.194e-01 2.045e-02 -5.840 5.28e-09 \*\*\*  
## bathrooms 8.171e-02 1.384e-02 5.905 3.58e-09 \*\*\*  
## sqft\_living 4.636e-04 2.148e-05 21.586 < 2e-16 \*\*\*  
## floors 1.490e-01 1.257e-02 11.854 < 2e-16 \*\*\*  
## waterfront1 3.581e-01 2.682e-02 13.352 < 2e-16 \*\*\*  
## view 8.561e-02 7.552e-03 11.336 < 2e-16 \*\*\*  
## condition 3.716e-02 3.556e-03 10.451 < 2e-16 \*\*\*  
## grade 2.525e-01 9.858e-03 25.609 < 2e-16 \*\*\*  
## sqft\_above -9.869e-05 1.768e-05 -5.584 2.38e-08 \*\*\*  
## yr\_built -5.614e-03 9.873e-05 -56.860 < 2e-16 \*\*\*  
## sqft\_living:sqft\_above 4.101e-09 3.383e-09 1.212 0.2253   
## sqft\_living:grade -2.439e-05 3.064e-06 -7.958 1.83e-15 \*\*\*  
## sqft\_living:floors -2.896e-05 5.937e-06 -4.877 1.08e-06 \*\*\*  
## sqft\_living:view -1.286e-05 2.449e-06 -5.250 1.53e-07 \*\*\*  
## bedrooms:bathrooms -2.705e-03 3.568e-03 -0.758 0.4483   
## bedrooms:grade 8.495e-03 3.328e-03 2.553 0.0107 \*   
## bedrooms:sqft\_above 1.290e-05 5.195e-06 2.483 0.0130 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.09653253)  
##   
## Null deviance: 5995.0 on 21611 degrees of freedom  
## Residual deviance: 2084.5 on 21594 degrees of freedom  
## AIC: 10826  
##   
## Number of Fisher Scoring iterations: 2

model 8

house.glm8 <- update(house.glm7, ~ . - bedrooms:sqft\_above - bedrooms:bathrooms - sqft\_living:sqft\_above)  
summary(house.glm8)

##   
## Call:  
## glm(formula = price ~ bedrooms + bathrooms + sqft\_living + floors +   
## waterfront + view + condition + grade + sqft\_above + yr\_built +   
## sqft\_living:grade + sqft\_living:floors + sqft\_living:view +   
## bedrooms:grade, data = kc\_house\_log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.40764 -0.20845 0.01571 0.20891 1.45979   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.156e+01 1.917e-01 112.486 < 2e-16 \*\*\*  
## bedrooms -1.339e-01 1.813e-02 -7.387 1.56e-13 \*\*\*  
## bathrooms 7.307e-02 4.988e-03 14.650 < 2e-16 \*\*\*  
## sqft\_living 4.382e-04 1.718e-05 25.507 < 2e-16 \*\*\*  
## floors 1.406e-01 1.217e-02 11.548 < 2e-16 \*\*\*  
## waterfront1 3.577e-01 2.683e-02 13.334 < 2e-16 \*\*\*  
## view 8.186e-02 7.445e-03 10.996 < 2e-16 \*\*\*  
## condition 3.700e-02 3.550e-03 10.420 < 2e-16 \*\*\*  
## grade 2.303e-01 7.470e-03 30.833 < 2e-16 \*\*\*  
## sqft\_above -4.176e-05 7.010e-06 -5.958 2.59e-09 \*\*\*  
## yr\_built -5.613e-03 9.835e-05 -57.073 < 2e-16 \*\*\*  
## sqft\_living:grade -2.131e-05 2.038e-06 -10.452 < 2e-16 \*\*\*  
## sqft\_living:floors -2.374e-05 5.716e-06 -4.153 3.30e-05 \*\*\*  
## sqft\_living:view -1.171e-05 2.419e-06 -4.840 1.31e-06 \*\*\*  
## bedrooms:grade 1.264e-02 2.352e-03 5.375 7.73e-08 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.09659214)  
##   
## Null deviance: 5995.0 on 21611 degrees of freedom  
## Residual deviance: 2086.1 on 21597 degrees of freedom  
## AIC: 10836  
##   
## Number of Fisher Scoring iterations: 2

model 9 - bathrooms interaction

house.glm9 <- update(house.glm8, ~ . + bathrooms:floors + bathrooms:grade + bathrooms:sqft\_above)  
summary(house.glm9)

##   
## Call:  
## glm(formula = price ~ bedrooms + bathrooms + sqft\_living + floors +   
## waterfront + view + condition + grade + sqft\_above + yr\_built +   
## sqft\_living:grade + sqft\_living:floors + sqft\_living:view +   
## bedrooms:grade + bathrooms:floors + bathrooms:grade + bathrooms:sqft\_above,   
## data = kc\_house\_log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.47519 -0.20813 0.01544 0.20881 1.46039   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.147e+01 1.950e-01 110.117 < 2e-16 \*\*\*  
## bedrooms -1.303e-01 1.863e-02 -6.993 2.77e-12 \*\*\*  
## bathrooms 3.940e-02 2.954e-02 1.334 0.18231   
## sqft\_living 4.812e-04 2.443e-05 19.692 < 2e-16 \*\*\*  
## floors 1.600e-01 1.742e-02 9.184 < 2e-16 \*\*\*  
## waterfront1 3.588e-01 2.683e-02 13.372 < 2e-16 \*\*\*  
## view 8.346e-02 7.515e-03 11.105 < 2e-16 \*\*\*  
## condition 3.747e-02 3.556e-03 10.536 < 2e-16 \*\*\*  
## grade 2.372e-01 9.079e-03 26.126 < 2e-16 \*\*\*  
## sqft\_above -6.568e-05 1.243e-05 -5.282 1.29e-07 \*\*\*  
## yr\_built -5.586e-03 9.902e-05 -56.418 < 2e-16 \*\*\*  
## sqft\_living:grade -2.675e-05 2.994e-06 -8.933 < 2e-16 \*\*\*  
## sqft\_living:floors -2.345e-05 7.198e-06 -3.258 0.00113 \*\*   
## sqft\_living:view -1.216e-05 2.440e-06 -4.983 6.31e-07 \*\*\*  
## bedrooms:grade 1.217e-02 2.419e-03 5.030 4.93e-07 \*\*\*  
## bathrooms:floors -8.550e-03 8.956e-03 -0.955 0.33978   
## bathrooms:grade 3.518e-03 4.219e-03 0.834 0.40440   
## bathrooms:sqft\_above 9.654e-06 4.167e-06 2.317 0.02051 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.09656421)  
##   
## Null deviance: 5995.0 on 21611 degrees of freedom  
## Residual deviance: 2085.2 on 21594 degrees of freedom  
## AIC: 10833  
##   
## Number of Fisher Scoring iterations: 2

model 10

house.glm10 <- update(house.glm9, ~ . - bathrooms:grade - bathrooms:floors)  
summary(house.glm10)

##   
## Call:  
## glm(formula = price ~ bedrooms + bathrooms + sqft\_living + floors +   
## waterfront + view + condition + grade + sqft\_above + yr\_built +   
## sqft\_living:grade + sqft\_living:floors + sqft\_living:view +   
## bedrooms:grade + bathrooms:sqft\_above, data = kc\_house\_log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.4698 -0.2079 0.0157 0.2086 1.4617   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.148e+01 1.936e-01 110.978 < 2e-16 \*\*\*  
## bedrooms -1.326e-01 1.814e-02 -7.311 2.75e-13 \*\*\*  
## bathrooms 5.280e-02 8.744e-03 6.038 1.58e-09 \*\*\*  
## sqft\_living 4.744e-04 2.142e-05 22.141 < 2e-16 \*\*\*  
## floors 1.492e-01 1.255e-02 11.890 < 2e-16 \*\*\*  
## waterfront1 3.586e-01 2.683e-02 13.368 < 2e-16 \*\*\*  
## view 8.374e-02 7.474e-03 11.205 < 2e-16 \*\*\*  
## condition 3.746e-02 3.554e-03 10.541 < 2e-16 \*\*\*  
## grade 2.401e-01 8.239e-03 29.148 < 2e-16 \*\*\*  
## sqft\_above -6.602e-05 1.109e-05 -5.953 2.67e-09 \*\*\*  
## yr\_built -5.598e-03 9.850e-05 -56.830 < 2e-16 \*\*\*  
## sqft\_living:grade -2.505e-05 2.432e-06 -10.301 < 2e-16 \*\*\*  
## sqft\_living:floors -2.767e-05 5.883e-06 -4.704 2.57e-06 \*\*\*  
## sqft\_living:view -1.228e-05 2.427e-06 -5.059 4.25e-07 \*\*\*  
## bedrooms:grade 1.249e-02 2.352e-03 5.309 1.11e-07 \*\*\*  
## bathrooms:sqft\_above 1.008e-05 3.571e-06 2.822 0.00477 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.096561)  
##   
## Null deviance: 5995.0 on 21611 degrees of freedom  
## Residual deviance: 2085.3 on 21596 degrees of freedom  
## AIC: 10830  
##   
## Number of Fisher Scoring iterations: 2

model 11 - floors interaction

house.glm11 <- update(house.glm10, ~ . + floors:condition + floors:grade + floors:sqft\_above)  
summary(house.glm11)

##   
## Call:  
## glm(formula = price ~ bedrooms + bathrooms + sqft\_living + floors +   
## waterfront + view + condition + grade + sqft\_above + yr\_built +   
## sqft\_living:grade + sqft\_living:floors + sqft\_living:view +   
## bedrooms:grade + bathrooms:sqft\_above + floors:condition +   
## floors:grade + floors:sqft\_above, data = kc\_house\_log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.49751 -0.20760 0.01532 0.20902 1.45728   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.150e+01 2.003e-01 107.308 < 2e-16 \*\*\*  
## bedrooms -1.335e-01 1.815e-02 -7.353 2.00e-13 \*\*\*  
## bathrooms 4.978e-02 8.909e-03 5.588 2.33e-08 \*\*\*  
## sqft\_living 4.579e-04 2.544e-05 18.001 < 2e-16 \*\*\*  
## floors 1.681e-01 4.813e-02 3.492 0.00048 \*\*\*  
## waterfront1 3.596e-01 2.684e-02 13.398 < 2e-16 \*\*\*  
## view 8.456e-02 7.532e-03 11.227 < 2e-16 \*\*\*  
## condition 5.454e-02 1.076e-02 5.071 4.00e-07 \*\*\*  
## grade 2.328e-01 1.122e-02 20.754 < 2e-16 \*\*\*  
## sqft\_above -1.700e-05 2.209e-05 -0.769 0.44168   
## yr\_built -5.618e-03 1.026e-04 -54.784 < 2e-16 \*\*\*  
## sqft\_living:grade -2.616e-05 2.522e-06 -10.373 < 2e-16 \*\*\*  
## sqft\_living:floors -7.647e-06 1.110e-05 -0.689 0.49084   
## sqft\_living:view -1.264e-05 2.462e-06 -5.133 2.88e-07 \*\*\*  
## bedrooms:grade 1.265e-02 2.354e-03 5.371 7.89e-08 \*\*\*  
## bathrooms:sqft\_above 1.114e-05 3.645e-06 3.056 0.00225 \*\*   
## floors:condition -1.342e-02 7.939e-03 -1.690 0.09106 .   
## floors:grade 5.697e-03 5.806e-03 0.981 0.32646   
## floors:sqft\_above -3.428e-05 1.373e-05 -2.496 0.01257 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.09653497)  
##   
## Null deviance: 5995.0 on 21611 degrees of freedom  
## Residual deviance: 2084.5 on 21593 degrees of freedom  
## AIC: 10828  
##   
## Number of Fisher Scoring iterations: 2

model 12

house.glm12 <- update(house.glm11, ~ . - sqft\_living:floors - floors:condition -floors:grade)  
summary(house.glm12)

##   
## Call:  
## glm(formula = price ~ bedrooms + bathrooms + sqft\_living + floors +   
## waterfront + view + condition + grade + sqft\_above + yr\_built +   
## sqft\_living:grade + sqft\_living:view + bedrooms:grade + bathrooms:sqft\_above +   
## floors:sqft\_above, data = kc\_house\_log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.49791 -0.20771 0.01574 0.20874 1.45809   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.140e+01 1.929e-01 110.923 < 2e-16 \*\*\*  
## bedrooms -1.330e-01 1.813e-02 -7.337 2.26e-13 \*\*\*  
## bathrooms 4.979e-02 8.848e-03 5.628 1.85e-08 \*\*\*  
## sqft\_living 4.431e-04 2.070e-05 21.404 < 2e-16 \*\*\*  
## floors 1.586e-01 1.340e-02 11.832 < 2e-16 \*\*\*  
## waterfront1 3.606e-01 2.683e-02 13.444 < 2e-16 \*\*\*  
## view 8.583e-02 7.485e-03 11.467 < 2e-16 \*\*\*  
## condition 3.743e-02 3.553e-03 10.535 < 2e-16 \*\*\*  
## grade 2.409e-01 8.187e-03 29.428 < 2e-16 \*\*\*  
## sqft\_above -1.488e-05 1.518e-05 -0.980 0.32720   
## yr\_built -5.564e-03 9.807e-05 -56.732 < 2e-16 \*\*\*  
## sqft\_living:grade -2.563e-05 2.418e-06 -10.600 < 2e-16 \*\*\*  
## sqft\_living:view -1.320e-05 2.435e-06 -5.419 6.05e-08 \*\*\*  
## bedrooms:grade 1.259e-02 2.352e-03 5.353 8.73e-08 \*\*\*  
## bathrooms:sqft\_above 1.096e-05 3.598e-06 3.048 0.00231 \*\*   
## floors:sqft\_above -3.718e-05 7.278e-06 -5.109 3.27e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.09654325)  
##   
## Null deviance: 5995.0 on 21611 degrees of freedom  
## Residual deviance: 2084.9 on 21596 degrees of freedom  
## AIC: 10826  
##   
## Number of Fisher Scoring iterations: 2

model 13

house.glm13 <- update(house.glm12, ~ . - sqft\_above)  
summary(house.glm13)

##   
## Call:  
## glm(formula = price ~ bedrooms + bathrooms + sqft\_living + floors +   
## waterfront + view + condition + grade + yr\_built + sqft\_living:grade +   
## sqft\_living:view + bedrooms:grade + bathrooms:sqft\_above +   
## floors:sqft\_above, data = kc\_house\_log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.49100 -0.20763 0.01565 0.20880 1.45695   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.143e+01 1.908e-01 112.317 < 2e-16 \*\*\*  
## bedrooms -1.348e-01 1.804e-02 -7.472 8.20e-14 \*\*\*  
## bathrooms 5.262e-02 8.363e-03 6.292 3.19e-10 \*\*\*  
## sqft\_living 4.349e-04 1.893e-05 22.976 < 2e-16 \*\*\*  
## floors 1.656e-01 1.130e-02 14.653 < 2e-16 \*\*\*  
## waterfront1 3.602e-01 2.682e-02 13.430 < 2e-16 \*\*\*  
## view 8.548e-02 7.477e-03 11.433 < 2e-16 \*\*\*  
## condition 3.738e-02 3.553e-03 10.522 < 2e-16 \*\*\*  
## grade 2.379e-01 7.566e-03 31.438 < 2e-16 \*\*\*  
## yr\_built -5.576e-03 9.729e-05 -57.310 < 2e-16 \*\*\*  
## sqft\_living:grade -2.485e-05 2.282e-06 -10.889 < 2e-16 \*\*\*  
## sqft\_living:view -1.304e-05 2.430e-06 -5.368 8.05e-08 \*\*\*  
## bedrooms:grade 1.282e-02 2.340e-03 5.479 4.33e-08 \*\*\*  
## bathrooms:sqft\_above 9.601e-06 3.317e-06 2.894 0.00381 \*\*   
## floors:sqft\_above -4.206e-05 5.312e-06 -7.917 2.55e-15 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.09654307)  
##   
## Null deviance: 5995 on 21611 degrees of freedom  
## Residual deviance: 2085 on 21597 degrees of freedom  
## AIC: 10825  
##   
## Number of Fisher Scoring iterations: 2

model 14 - floors interaction

house.glm14 <- update(house.glm13, ~ . + waterfront:view + view:grade + grade:sqft\_above)  
summary(house.glm14)

##   
## Call:  
## glm(formula = price ~ bedrooms + bathrooms + sqft\_living + floors +   
## waterfront + view + condition + grade + yr\_built + sqft\_living:grade +   
## sqft\_living:view + bedrooms:grade + bathrooms:sqft\_above +   
## floors:sqft\_above + waterfront:view + view:grade + grade:sqft\_above,   
## data = kc\_house\_log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.48332 -0.20768 0.01546 0.20860 1.45654   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.144e+01 1.927e-01 111.278 < 2e-16 \*\*\*  
## bedrooms -1.364e-01 1.824e-02 -7.478 7.86e-14 \*\*\*  
## bathrooms 5.343e-02 8.935e-03 5.980 2.26e-09 \*\*\*  
## sqft\_living 4.380e-04 1.963e-05 22.311 < 2e-16 \*\*\*  
## floors 1.704e-01 1.412e-02 12.070 < 2e-16 \*\*\*  
## waterfront1 7.505e-02 1.658e-01 0.453 0.6508   
## view 7.839e-02 1.995e-02 3.929 8.57e-05 \*\*\*  
## condition 3.747e-02 3.554e-03 10.541 < 2e-16 \*\*\*  
## grade 2.356e-01 8.512e-03 27.678 < 2e-16 \*\*\*  
## yr\_built -5.581e-03 9.780e-05 -57.068 < 2e-16 \*\*\*  
## sqft\_living:grade -2.527e-05 2.406e-06 -10.503 < 2e-16 \*\*\*  
## sqft\_living:view -1.405e-05 3.593e-06 -3.912 9.20e-05 \*\*\*  
## bedrooms:grade 1.303e-02 2.367e-03 5.502 3.79e-08 \*\*\*  
## bathrooms:sqft\_above 9.179e-06 3.664e-06 2.505 0.0122 \*   
## floors:sqft\_above -4.509e-05 7.660e-06 -5.886 4.02e-09 \*\*\*  
## waterfront1:view 7.613e-02 4.365e-02 1.744 0.0812 .   
## view:grade 1.132e-03 3.026e-03 0.374 0.7084   
## grade:sqft\_above 9.250e-07 1.930e-06 0.479 0.6317   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.09654095)  
##   
## Null deviance: 5995.0 on 21611 degrees of freedom  
## Residual deviance: 2084.7 on 21594 degrees of freedom  
## AIC: 10828  
##   
## Number of Fisher Scoring iterations: 2

model 15

house.glm15 <- update(house.glm14, ~ . - grade:sqft\_above - view:grade - waterfront)  
summary(house.glm15)

##   
## Call:  
## glm(formula = price ~ bedrooms + bathrooms + sqft\_living + floors +   
## view + condition + grade + yr\_built + sqft\_living:grade +   
## sqft\_living:view + bedrooms:grade + bathrooms:sqft\_above +   
## floors:sqft\_above + waterfront:view, data = kc\_house\_log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.48990 -0.20775 0.01554 0.20877 1.45692   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.143e+01 1.908e-01 112.345 < 2e-16 \*\*\*  
## bedrooms -1.347e-01 1.804e-02 -7.465 8.61e-14 \*\*\*  
## bathrooms 5.239e-02 8.363e-03 6.265 3.81e-10 \*\*\*  
## sqft\_living 4.347e-04 1.892e-05 22.968 < 2e-16 \*\*\*  
## floors 1.660e-01 1.130e-02 14.681 < 2e-16 \*\*\*  
## view 8.587e-02 7.472e-03 11.492 < 2e-16 \*\*\*  
## condition 3.751e-02 3.553e-03 10.558 < 2e-16 \*\*\*  
## grade 2.378e-01 7.565e-03 31.433 < 2e-16 \*\*\*  
## yr\_built -5.578e-03 9.729e-05 -57.335 < 2e-16 \*\*\*  
## sqft\_living:grade -2.479e-05 2.282e-06 -10.864 < 2e-16 \*\*\*  
## sqft\_living:view -1.333e-05 2.431e-06 -5.484 4.20e-08 \*\*\*  
## bedrooms:grade 1.279e-02 2.340e-03 5.467 4.64e-08 \*\*\*  
## bathrooms:sqft\_above 9.689e-06 3.317e-06 2.921 0.0035 \*\*   
## floors:sqft\_above -4.221e-05 5.312e-06 -7.947 2.01e-15 \*\*\*  
## view:waterfront1 9.560e-02 7.062e-03 13.538 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.09653019)  
##   
## Null deviance: 5995.0 on 21611 degrees of freedom  
## Residual deviance: 2084.8 on 21597 degrees of freedom  
## AIC: 10823  
##   
## Number of Fisher Scoring iterations: 2

#### 2) as per concept of parsimony : checking best fit from above model

house.glm15.bestFit1 <- regsubsets(price ~ bedrooms + bathrooms + sqft\_living + floors +   
 view + condition + grade + yr\_built + sqft\_living:grade +   
 sqft\_living:view + bedrooms:grade + bathrooms:sqft\_above +   
 floors:sqft\_above + waterfront:view, data = kc\_house\_log, nbest = 2, nvmax = 10)  
par(mfrow = c(1,1))  
#subsets(house.glm15.bestFit1, statistic = "adjr2")  
plot(house.glm15.bestFit1, scale = "adjr2")

A screenshot of a cell phone

Description automatically generated

from above we can see 4,5,6 and later produce almost similar result

house.glm15.bestFit2 <- regsubsets(price ~ bedrooms + bathrooms + sqft\_living + floors +   
 view + condition + grade + yr\_built + sqft\_living:grade +   
 sqft\_living:view + bedrooms:grade + bathrooms:sqft\_above +   
 floors:sqft\_above + waterfront:view, data = kc\_house\_log, nbest = 1, nvmax = 6)  
par(mfrow = c(1,1))  
#subsets(house.glm15.bestFit2, statistic = "adjr2", max.size = 6, min.size = 1)  
plot(house.glm15.bestFit2, scale = "adjr2")

A screenshot of a cell phone

Description automatically generated

Doing cross-validation for 3,4,5,6 predictors using K-Fold 3 predictors predictor : 3-i

house.k10.3i <- glm(price ~ sqft\_living + grade + yr\_built, data = kc\_house\_log)  
summary(house.k10.3i) # 12786

##   
## Call:  
## glm(formula = price ~ sqft\_living + grade + yr\_built, data = kc\_house\_log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.73279 -0.21945 0.01622 0.21898 1.34761   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.068e+01 1.594e-01 129.70 <2e-16 \*\*\*  
## sqft\_living 2.087e-04 3.727e-06 56.00 <2e-16 \*\*\*  
## grade 2.473e-01 3.086e-03 80.14 <2e-16 \*\*\*  
## yr\_built -5.053e-03 8.426e-05 -59.97 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.1057657)  
##   
## Null deviance: 5995.0 on 21611 degrees of freedom  
## Residual deviance: 2285.4 on 21608 degrees of freedom  
## AIC: 12786  
##   
## Number of Fisher Scoring iterations: 2

house.k10.3i.err <- cv.glm(data = kc\_house\_log,glmfit = house.k10.3i, K = 10)  
round(house.k10.3i.err$delta[1], 4) # 0.1058

## [1] 0.1058

predictor : 3-ii

house.k10.3ii <- glm(price ~ sqft\_living:grade + grade + yr\_built, data = kc\_house\_log)  
summary(house.k10.3ii) # 13539

##   
## Call:  
## glm(formula = price ~ sqft\_living:grade + grade + yr\_built, data = kc\_house\_log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.14677 -0.22200 0.01861 0.22391 1.45162   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.061e+01 1.622e-01 127.04 <2e-16 \*\*\*  
## grade 2.226e-01 3.818e-03 58.29 <2e-16 \*\*\*  
## yr\_built -4.870e-03 8.602e-05 -56.62 <2e-16 \*\*\*  
## sqft\_living:grade 1.996e-05 4.172e-07 47.84 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.1095155)  
##   
## Null deviance: 5995.0 on 21611 degrees of freedom  
## Residual deviance: 2366.4 on 21608 degrees of freedom  
## AIC: 13539  
##   
## Number of Fisher Scoring iterations: 2

house.k10.3ii.err <- cv.glm(data = kc\_house\_log,glmfit = house.k10.3ii, K = 10)  
round(house.k10.3ii.err$delta[1], 4) # 0.1096

## [1] 0.1096

4 predictors predictor : 4-i

house.k10.4i <- glm(price ~ sqft\_living + grade + yr\_built + view, data = kc\_house\_log)  
summary(house.k10.4i) # 12272

##   
## Call:  
## glm(formula = price ~ sqft\_living + grade + yr\_built + view,   
## data = kc\_house\_log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.82869 -0.21786 0.01813 0.21939 1.28731   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.003e+01 1.601e-01 125.16 <2e-16 \*\*\*  
## sqft\_living 1.965e-04 3.722e-06 52.80 <2e-16 \*\*\*  
## grade 2.392e-01 3.071e-03 77.90 <2e-16 \*\*\*  
## yr\_built -4.689e-03 8.477e-05 -55.32 <2e-16 \*\*\*  
## view 6.933e-02 3.034e-03 22.85 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.1032754)  
##   
## Null deviance: 5995.0 on 21611 degrees of freedom  
## Residual deviance: 2231.5 on 21607 degrees of freedom  
## AIC: 12272  
##   
## Number of Fisher Scoring iterations: 2

house.k10.4i.err <- cv.glm(data = kc\_house\_log,glmfit = house.k10.4i, K = 10)  
round(house.k10.4i.err$delta[1], 4) # 0.1033

## [1] 0.1033

predictor : 4-ii

house.k10.4ii <- glm(price ~ sqft\_living + grade + yr\_built + bathrooms, data = kc\_house\_log)  
summary(house.k10.4ii) # 12351

##   
## Call:  
## glm(formula = price ~ sqft\_living + grade + yr\_built + bathrooms,   
## data = kc\_house\_log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.69517 -0.21893 0.01573 0.21689 1.39528   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.207e+01 1.711e-01 128.99 <2e-16 \*\*\*  
## sqft\_living 1.573e-04 4.428e-06 35.52 <2e-16 \*\*\*  
## grade 2.422e-01 3.065e-03 79.01 <2e-16 \*\*\*  
## yr\_built -5.790e-03 9.049e-05 -63.99 <2e-16 \*\*\*  
## bathrooms 1.011e-01 4.810e-03 21.02 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.1036514)  
##   
## Null deviance: 5995.0 on 21611 degrees of freedom  
## Residual deviance: 2239.6 on 21607 degrees of freedom  
## AIC: 12351  
##   
## Number of Fisher Scoring iterations: 2

house.k10.4ii.err <- cv.glm(data = kc\_house\_log,glmfit = house.k10.4ii, K = 10)  
round(house.k10.4ii.err$delta[1], 4) # 0.1037

## [1] 0.1037

5 predictors predictor : 5-i

house.k10.5i <- glm(price ~ sqft\_living + grade + yr\_built + view + bathrooms, data = kc\_house\_log)  
summary(house.k10.5i) # 11844

##   
## Call:  
## glm(formula = price ~ sqft\_living + grade + yr\_built + view +   
## bathrooms, data = kc\_house\_log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.79020 -0.21669 0.01738 0.21526 1.33784   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.140e+01 1.716e-01 124.75 <2e-16 \*\*\*  
## sqft\_living 1.463e-04 4.403e-06 33.23 <2e-16 \*\*\*  
## grade 2.343e-01 3.049e-03 76.83 <2e-16 \*\*\*  
## yr\_built -5.418e-03 9.093e-05 -59.59 <2e-16 \*\*\*  
## view 6.816e-02 3.005e-03 22.68 <2e-16 \*\*\*  
## bathrooms 9.908e-02 4.754e-03 20.84 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.101245)  
##   
## Null deviance: 5995.0 on 21611 degrees of freedom  
## Residual deviance: 2187.5 on 21606 degrees of freedom  
## AIC: 11844  
##   
## Number of Fisher Scoring iterations: 2

house.k10.5i.err <- cv.glm(data = kc\_house\_log,glmfit = house.k10.5i, K = 10)  
round(house.k10.5i.err$delta[1], 4) # 0.1013

## [1] 0.1013

predictor : 5-ii

house.k10.5ii <- glm(price ~ sqft\_living + grade + yr\_built + view + bathrooms, data = kc\_house\_log)  
summary(house.k10.5ii) # 11844

##   
## Call:  
## glm(formula = price ~ sqft\_living + grade + yr\_built + view +   
## bathrooms, data = kc\_house\_log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.79020 -0.21669 0.01738 0.21526 1.33784   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.140e+01 1.716e-01 124.75 <2e-16 \*\*\*  
## sqft\_living 1.463e-04 4.403e-06 33.23 <2e-16 \*\*\*  
## grade 2.343e-01 3.049e-03 76.83 <2e-16 \*\*\*  
## yr\_built -5.418e-03 9.093e-05 -59.59 <2e-16 \*\*\*  
## view 6.816e-02 3.005e-03 22.68 <2e-16 \*\*\*  
## bathrooms 9.908e-02 4.754e-03 20.84 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.101245)  
##   
## Null deviance: 5995.0 on 21611 degrees of freedom  
## Residual deviance: 2187.5 on 21606 degrees of freedom  
## AIC: 11844  
##   
## Number of Fisher Scoring iterations: 2

house.k10.5ii.err <- cv.glm(data = kc\_house\_log,glmfit = house.k10.5ii, K = 10)  
round(house.k10.5ii.err$delta[1], 4) # 0.1013

## [1] 0.1013

6 predictors predictor : 6-i

house.k10.6i <- glm(price ~ sqft\_living + grade + yr\_built + view + bathrooms + floors, data = kc\_house\_log)  
summary(house.k10.6i) # 11587

##   
## Call:  
## glm(formula = price ~ sqft\_living + grade + yr\_built + view +   
## bathrooms + floors, data = kc\_house\_log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.84521 -0.21320 0.01657 0.21441 1.34020   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.212e+01 1.763e-01 125.49 <2e-16 \*\*\*  
## sqft\_living 1.538e-04 4.401e-06 34.95 <2e-16 \*\*\*  
## grade 2.249e-01 3.087e-03 72.85 <2e-16 \*\*\*  
## yr\_built -5.796e-03 9.337e-05 -62.07 <2e-16 \*\*\*  
## view 7.025e-02 2.990e-03 23.50 <2e-16 \*\*\*  
## bathrooms 8.078e-02 4.860e-03 16.62 <2e-16 \*\*\*  
## floors 7.971e-02 4.942e-03 16.13 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.1000448)  
##   
## Null deviance: 5995.0 on 21611 degrees of freedom  
## Residual deviance: 2161.5 on 21605 degrees of freedom  
## AIC: 11587  
##   
## Number of Fisher Scoring iterations: 2

house.k10.6i.err <- cv.glm(data = kc\_house\_log,glmfit = house.k10.6i, K = 10)  
round(house.k10.6i.err$delta[1], 4) # 0.1001

## [1] 0.1001

predictor : 6-ii

house.k10.6ii <- glm(price ~ sqft\_living + grade + yr\_built + view + sqft\_living:grade + floors, data = kc\_house\_log)  
summary(house.k10.6ii) # 11589

##   
## Call:  
## glm(formula = price ~ sqft\_living + grade + yr\_built + view +   
## sqft\_living:grade + floors, data = kc\_house\_log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.31134 -0.21063 0.01459 0.21335 1.33178   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.119e+01 1.681e-01 125.99 <2e-16 \*\*\*  
## sqft\_living 3.881e-04 1.225e-05 31.69 <2e-16 \*\*\*  
## grade 2.787e-01 4.411e-03 63.17 <2e-16 \*\*\*  
## yr\_built -5.510e-03 8.967e-05 -61.45 <2e-16 \*\*\*  
## view 7.464e-02 2.994e-03 24.93 <2e-16 \*\*\*  
## floors 9.287e-02 4.819e-03 19.27 <2e-16 \*\*\*  
## sqft\_living:grade -2.237e-05 1.349e-06 -16.58 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.100051)  
##   
## Null deviance: 5995.0 on 21611 degrees of freedom  
## Residual deviance: 2161.6 on 21605 degrees of freedom  
## AIC: 11589  
##   
## Number of Fisher Scoring iterations: 2

house.k10.6ii.err <- cv.glm(data = kc\_house\_log,glmfit = house.k10.6ii, K = 10)  
round(house.k10.6ii.err$delta[1], 4) # 0.1001

## [1] 0.1001

Checking the r square and adj. r square values for house.k10.6i model

house.k10.6i.lm <- lm(price ~ sqft\_living + grade + yr\_built + view + bathrooms + floors, data = kc\_house\_log)  
summary(house.k10.6i.lm)

##   
## Call:  
## lm(formula = price ~ sqft\_living + grade + yr\_built + view +   
## bathrooms + floors, data = kc\_house\_log)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.84521 -0.21320 0.01657 0.21441 1.34020   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.212e+01 1.763e-01 125.49 <2e-16 \*\*\*  
## sqft\_living 1.538e-04 4.401e-06 34.95 <2e-16 \*\*\*  
## grade 2.249e-01 3.087e-03 72.85 <2e-16 \*\*\*  
## yr\_built -5.796e-03 9.337e-05 -62.07 <2e-16 \*\*\*  
## view 7.025e-02 2.990e-03 23.50 <2e-16 \*\*\*  
## bathrooms 8.078e-02 4.860e-03 16.62 <2e-16 \*\*\*  
## floors 7.971e-02 4.942e-03 16.13 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3163 on 21605 degrees of freedom  
## Multiple R-squared: 0.6395, Adjusted R-squared: 0.6394   
## F-statistic: 6386 on 6 and 21605 DF, p-value: < 2.2e-16

Plot the model for house.k10.6i.lm model

par(mfrow = c(2,2))  
plot(house.k10.4)

A close up of text on a white background

Description automatically generated

### Step 3.3 - Data transformation - log(price) conversion and using only numeric type parameters

removing very less correlated variable by creating a new df

house <- kc\_house\_log

remove sqft\_lot, floors, waterfront, view, condition, sqft\_basement, yr\_built, yr\_renovated i.e columns (5,6,7,8,9,13,14)

house <- house[,-c(5,6,7,8,9,12,13,14)]

### Step 4.3 : Training a model on the data - log(price) conversion and using only numeric type parameters

Model the linear model model fit 1

house.fit1 <- lm(price ~ bedrooms\*bathrooms\*sqft\_living\*grade\*sqft\_above, data = house)

### Step 5.3 : Evaluating the model - log(price) conversion and using only numeric type parameters

Cheking the summary and plot of the model

summary(house.fit1)

##   
## Call:  
## lm(formula = price ~ bedrooms \* bathrooms \* sqft\_living \* grade \*   
## sqft\_above, data = house)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.26325 -0.24011 0.00189 0.22829 1.43154   
##   
## Coefficients:  
## Estimate Std. Error t value  
## (Intercept) 1.148e+01 2.331e-01 49.265  
## bedrooms -2.793e-01 8.482e-02 -3.292  
## bathrooms 4.630e-01 2.202e-01 2.103  
## sqft\_living 5.994e-05 3.614e-04 0.166  
## grade 1.482e-01 3.475e-02 4.265  
## sqft\_above 9.753e-06 4.268e-04 0.023  
## bedrooms:bathrooms -1.211e-01 5.912e-02 -2.048  
## bedrooms:sqft\_living 9.137e-05 8.738e-05 1.046  
## bathrooms:sqft\_living 5.056e-05 1.341e-04 0.377  
## bedrooms:grade 2.815e-02 1.300e-02 2.166  
## bathrooms:grade -5.309e-02 2.732e-02 -1.943  
## sqft\_living:grade 5.085e-05 4.708e-05 1.080  
## bedrooms:sqft\_above 1.406e-04 1.030e-04 1.366  
## bathrooms:sqft\_above -5.450e-04 1.696e-04 -3.213  
## sqft\_living:sqft\_above -1.512e-08 9.403e-08 -0.161  
## grade:sqft\_above 8.829e-06 5.111e-05 0.173  
## bedrooms:bathrooms:sqft\_living 5.504e-06 2.865e-05 0.192  
## bedrooms:bathrooms:grade 1.611e-02 7.480e-03 2.154  
## bedrooms:sqft\_living:grade -1.593e-05 1.113e-05 -1.431  
## bathrooms:sqft\_living:grade -3.997e-06 1.568e-05 -0.255  
## bedrooms:bathrooms:sqft\_above 6.886e-05 3.678e-05 1.872  
## bedrooms:sqft\_living:sqft\_above -1.914e-08 2.137e-08 -0.895  
## bathrooms:sqft\_living:sqft\_above 8.932e-08 3.315e-08 2.695  
## bedrooms:grade:sqft\_above -1.828e-05 1.199e-05 -1.524  
## bathrooms:grade:sqft\_above 4.461e-05 1.836e-05 2.429  
## sqft\_living:grade:sqft\_above -9.490e-09 9.240e-09 -1.027  
## bedrooms:bathrooms:sqft\_living:grade -9.638e-07 3.272e-06 -0.295  
## bedrooms:bathrooms:sqft\_living:sqft\_above -1.462e-08 6.172e-09 -2.368  
## bedrooms:bathrooms:grade:sqft\_above -5.015e-06 3.901e-06 -1.286  
## bedrooms:sqft\_living:grade:sqft\_above 4.328e-09 2.051e-09 2.111  
## bathrooms:sqft\_living:grade:sqft\_above -6.510e-09 2.880e-09 -2.260  
## bedrooms:bathrooms:sqft\_living:grade:sqft\_above 9.753e-10 5.145e-10 1.896  
## Pr(>|t|)   
## (Intercept) < 2e-16 \*\*\*  
## bedrooms 0.000995 \*\*\*  
## bathrooms 0.035498 \*   
## sqft\_living 0.868257   
## grade 2.01e-05 \*\*\*  
## sqft\_above 0.981770   
## bedrooms:bathrooms 0.040605 \*   
## bedrooms:sqft\_living 0.295707   
## bathrooms:sqft\_living 0.706119   
## bedrooms:grade 0.030312 \*   
## bathrooms:grade 0.051991 .   
## sqft\_living:grade 0.280196   
## bedrooms:sqft\_above 0.172031   
## bathrooms:sqft\_above 0.001313 \*\*   
## sqft\_living:sqft\_above 0.872259   
## grade:sqft\_above 0.862857   
## bedrooms:bathrooms:sqft\_living 0.847661   
## bedrooms:bathrooms:grade 0.031237 \*   
## bedrooms:sqft\_living:grade 0.152302   
## bathrooms:sqft\_living:grade 0.798775   
## bedrooms:bathrooms:sqft\_above 0.061185 .   
## bedrooms:sqft\_living:sqft\_above 0.370659   
## bathrooms:sqft\_living:sqft\_above 0.007052 \*\*   
## bedrooms:grade:sqft\_above 0.127485   
## bathrooms:grade:sqft\_above 0.015133 \*   
## sqft\_living:grade:sqft\_above 0.304371   
## bedrooms:bathrooms:sqft\_living:grade 0.768316   
## bedrooms:bathrooms:sqft\_living:sqft\_above 0.017901 \*   
## bedrooms:bathrooms:grade:sqft\_above 0.198577   
## bedrooms:sqft\_living:grade:sqft\_above 0.034814 \*   
## bathrooms:sqft\_living:grade:sqft\_above 0.023816 \*   
## bedrooms:bathrooms:sqft\_living:grade:sqft\_above 0.058035 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3452 on 21580 degrees of freedom  
## Multiple R-squared: 0.571, Adjusted R-squared: 0.5704   
## F-statistic: 926.5 on 31 and 21580 DF, p-value: < 2.2e-16

par(mfrow = c(2,2))  
plot(house.fit1)

A screenshot of a cell phone

Description automatically generated

### Step 6.3 : Improving the model - log(price) conversion and using only numeric type parameters

#### 1) Adding all the possible relation to the model

fit 2

house.fit2 <- update(house.fit1, ~ . + I(bedrooms^2) + I(bathrooms^2) + I(sqft\_living^2) + I(grade^2) + I(sqft\_above^2))  
summary(house.fit2)

##   
## Call:  
## lm(formula = price ~ bedrooms + bathrooms + sqft\_living + grade +   
## sqft\_above + I(bedrooms^2) + I(bathrooms^2) + I(sqft\_living^2) +   
## I(grade^2) + I(sqft\_above^2) + bedrooms:bathrooms + bedrooms:sqft\_living +   
## bathrooms:sqft\_living + bedrooms:grade + bathrooms:grade +   
## sqft\_living:grade + bedrooms:sqft\_above + bathrooms:sqft\_above +   
## sqft\_living:sqft\_above + grade:sqft\_above + bedrooms:bathrooms:sqft\_living +   
## bedrooms:bathrooms:grade + bedrooms:sqft\_living:grade + bathrooms:sqft\_living:grade +   
## bedrooms:bathrooms:sqft\_above + bedrooms:sqft\_living:sqft\_above +   
## bathrooms:sqft\_living:sqft\_above + bedrooms:grade:sqft\_above +   
## bathrooms:grade:sqft\_above + sqft\_living:grade:sqft\_above +   
## bedrooms:bathrooms:sqft\_living:grade + bedrooms:bathrooms:sqft\_living:sqft\_above +   
## bedrooms:bathrooms:grade:sqft\_above + bedrooms:sqft\_living:grade:sqft\_above +   
## bathrooms:sqft\_living:grade:sqft\_above + bedrooms:bathrooms:sqft\_living:grade:sqft\_above,   
## data = house)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.33778 -0.24000 0.00412 0.22684 1.43264   
##   
## Coefficients:  
## Estimate Std. Error t value  
## (Intercept) 1.154e+01 2.346e-01 49.202  
## bedrooms -2.988e-01 8.473e-02 -3.526  
## bathrooms 5.023e-01 2.234e-01 2.249  
## sqft\_living 5.186e-05 3.622e-04 0.143  
## grade 1.068e-01 4.264e-02 2.505  
## sqft\_above 2.669e-04 4.267e-04 0.626  
## I(bedrooms^2) 2.004e-02 2.607e-03 7.685  
## I(bathrooms^2) 4.356e-02 6.684e-03 6.517  
## I(sqft\_living^2) -1.136e-07 1.234e-08 -9.210  
## I(grade^2) 8.768e-03 3.035e-03 2.889  
## I(sqft\_above^2) -1.182e-07 1.591e-08 -7.432  
## bedrooms:bathrooms -1.222e-01 5.926e-02 -2.061  
## bedrooms:sqft\_living 9.658e-05 8.794e-05 1.098  
## bathrooms:sqft\_living 1.489e-04 1.344e-04 1.108  
## bedrooms:grade 1.963e-02 1.296e-02 1.515  
## bathrooms:grade -8.622e-02 2.830e-02 -3.047  
## sqft\_living:grade 6.238e-05 4.747e-05 1.314  
## bedrooms:sqft\_above 2.741e-05 1.030e-04 0.266  
## bathrooms:sqft\_above -4.752e-04 1.690e-04 -2.812  
## sqft\_living:sqft\_above 2.241e-07 9.698e-08 2.311  
## grade:sqft\_above -4.245e-05 5.130e-05 -0.827  
## bedrooms:bathrooms:sqft\_living -2.540e-05 2.864e-05 -0.887  
## bedrooms:bathrooms:grade 1.640e-02 7.515e-03 2.182  
## bedrooms:sqft\_living:grade -1.618e-05 1.114e-05 -1.452  
## bathrooms:sqft\_living:grade -1.123e-05 1.568e-05 -0.717  
## bedrooms:bathrooms:sqft\_above 6.659e-05 3.667e-05 1.816  
## bedrooms:sqft\_living:sqft\_above -8.551e-09 2.139e-08 -0.400  
## bathrooms:sqft\_living:sqft\_above 5.533e-08 3.314e-08 1.670  
## bedrooms:grade:sqft\_above -3.056e-06 1.199e-05 -0.255  
## bathrooms:grade:sqft\_above 4.323e-05 1.830e-05 2.363  
## sqft\_living:grade:sqft\_above -8.691e-09 9.265e-09 -0.938  
## bedrooms:bathrooms:sqft\_living:grade 1.854e-06 3.266e-06 0.568  
## bedrooms:bathrooms:sqft\_living:sqft\_above -8.613e-09 6.166e-09 -1.397  
## bedrooms:bathrooms:grade:sqft\_above -6.720e-06 3.886e-06 -1.729  
## bedrooms:sqft\_living:grade:sqft\_above 2.896e-09 2.058e-09 1.407  
## bathrooms:sqft\_living:grade:sqft\_above -3.889e-09 2.881e-09 -1.350  
## bedrooms:bathrooms:sqft\_living:grade:sqft\_above 5.724e-10 5.143e-10 1.113  
## Pr(>|t|)   
## (Intercept) < 2e-16 \*\*\*  
## bedrooms 0.000422 \*\*\*  
## bathrooms 0.024527 \*   
## sqft\_living 0.886147   
## grade 0.012238 \*   
## sqft\_above 0.531596   
## I(bedrooms^2) 1.59e-14 \*\*\*  
## I(bathrooms^2) 7.35e-11 \*\*\*  
## I(sqft\_living^2) < 2e-16 \*\*\*  
## I(grade^2) 0.003874 \*\*   
## I(sqft\_above^2) 1.11e-13 \*\*\*  
## bedrooms:bathrooms 0.039268 \*   
## bedrooms:sqft\_living 0.272135   
## bathrooms:sqft\_living 0.267742   
## bedrooms:grade 0.129811   
## bathrooms:grade 0.002317 \*\*   
## sqft\_living:grade 0.188830   
## bedrooms:sqft\_above 0.790110   
## bathrooms:sqft\_above 0.004927 \*\*   
## sqft\_living:sqft\_above 0.020854 \*   
## grade:sqft\_above 0.408039   
## bedrooms:bathrooms:sqft\_living 0.375069   
## bedrooms:bathrooms:grade 0.029130 \*   
## bedrooms:sqft\_living:grade 0.146550   
## bathrooms:sqft\_living:grade 0.473688   
## bedrooms:bathrooms:sqft\_above 0.069388 .   
## bedrooms:sqft\_living:sqft\_above 0.689373   
## bathrooms:sqft\_living:sqft\_above 0.095015 .   
## bedrooms:grade:sqft\_above 0.798911   
## bathrooms:grade:sqft\_above 0.018148 \*   
## sqft\_living:grade:sqft\_above 0.348259   
## bedrooms:bathrooms:sqft\_living:grade 0.570285   
## bedrooms:bathrooms:sqft\_living:sqft\_above 0.162499   
## bedrooms:bathrooms:grade:sqft\_above 0.083774 .   
## bedrooms:sqft\_living:grade:sqft\_above 0.159504   
## bathrooms:sqft\_living:grade:sqft\_above 0.177085   
## bedrooms:bathrooms:sqft\_living:grade:sqft\_above 0.265788   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3435 on 21575 degrees of freedom  
## Multiple R-squared: 0.5753, Adjusted R-squared: 0.5746   
## F-statistic: 811.8 on 36 and 21575 DF, p-value: < 2.2e-16

fit 3

house.fit3 <- update(house.fit2, ~ . + I(bedrooms^3) + I(bathrooms^3) + I(sqft\_living^3) + I(grade^3) + I(sqft\_above^3))  
summary(house.fit3)

##   
## Call:  
## lm(formula = price ~ bedrooms + bathrooms + sqft\_living + grade +   
## sqft\_above + I(bedrooms^2) + I(bathrooms^2) + I(sqft\_living^2) +   
## I(grade^2) + I(sqft\_above^2) + I(bedrooms^3) + I(bathrooms^3) +   
## I(sqft\_living^3) + I(grade^3) + I(sqft\_above^3) + bedrooms:bathrooms +   
## bedrooms:sqft\_living + bathrooms:sqft\_living + bedrooms:grade +   
## bathrooms:grade + sqft\_living:grade + bedrooms:sqft\_above +   
## bathrooms:sqft\_above + sqft\_living:sqft\_above + grade:sqft\_above +   
## bedrooms:bathrooms:sqft\_living + bedrooms:bathrooms:grade +   
## bedrooms:sqft\_living:grade + bathrooms:sqft\_living:grade +   
## bedrooms:bathrooms:sqft\_above + bedrooms:sqft\_living:sqft\_above +   
## bathrooms:sqft\_living:sqft\_above + bedrooms:grade:sqft\_above +   
## bathrooms:grade:sqft\_above + sqft\_living:grade:sqft\_above +   
## bedrooms:bathrooms:sqft\_living:grade + bedrooms:bathrooms:sqft\_living:sqft\_above +   
## bedrooms:bathrooms:grade:sqft\_above + bedrooms:sqft\_living:grade:sqft\_above +   
## bathrooms:sqft\_living:grade:sqft\_above + bedrooms:bathrooms:sqft\_living:grade:sqft\_above,   
## data = house)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.25589 -0.23960 0.00441 0.22727 1.42833   
##   
## Coefficients:  
## Estimate Std. Error t value  
## (Intercept) 1.197e+01 2.887e-01 41.476  
## bedrooms -2.639e-01 9.357e-02 -2.820  
## bathrooms 6.206e-01 2.444e-01 2.539  
## sqft\_living 4.076e-04 3.707e-04 1.100  
## grade -1.608e-01 1.222e-01 -1.315  
## sqft\_above 2.160e-04 4.333e-04 0.499  
## I(bedrooms^2) 2.891e-02 8.770e-03 3.296  
## I(bathrooms^2) 4.553e-02 2.209e-02 2.062  
## I(sqft\_living^2) -2.169e-07 2.335e-08 -9.290  
## I(grade^2) 4.968e-02 1.899e-02 2.617  
## I(sqft\_above^2) -1.956e-07 3.053e-08 -6.407  
## I(bedrooms^3) -7.685e-04 6.277e-04 -1.224  
## I(bathrooms^3) -6.732e-05 2.759e-03 -0.024  
## I(sqft\_living^3) 8.428e-12 1.682e-12 5.012  
## I(grade^3) -1.729e-03 7.704e-04 -2.244  
## I(sqft\_above^3) 7.898e-12 2.704e-12 2.921  
## bedrooms:bathrooms -7.184e-02 6.459e-02 -1.112  
## bedrooms:sqft\_living -4.373e-05 9.231e-05 -0.474  
## bathrooms:sqft\_living 5.435e-05 1.367e-04 0.398  
## bedrooms:grade 1.471e-02 1.426e-02 1.031  
## bathrooms:grade -9.961e-02 3.241e-02 -3.073  
## sqft\_living:grade 2.431e-05 4.870e-05 0.499  
## bedrooms:sqft\_above 4.833e-05 1.071e-04 0.451  
## bathrooms:sqft\_above -5.136e-04 1.712e-04 -3.000  
## sqft\_living:sqft\_above 3.784e-07 1.071e-07 3.533  
## grade:sqft\_above -3.911e-05 5.215e-05 -0.750  
## bedrooms:bathrooms:sqft\_living 9.702e-06 3.011e-05 0.322  
## bedrooms:bathrooms:grade 3.861e-03 8.673e-03 0.445  
## bedrooms:sqft\_living:grade 3.613e-06 1.177e-05 0.307  
## bathrooms:sqft\_living:grade 4.884e-06 1.617e-05 0.302  
## bedrooms:bathrooms:sqft\_above 5.201e-05 3.860e-05 1.347  
## bedrooms:sqft\_living:sqft\_above 9.354e-09 2.234e-08 0.419  
## bathrooms:sqft\_living:sqft\_above 4.802e-08 3.329e-08 1.442  
## bedrooms:grade:sqft\_above -7.358e-06 1.267e-05 -0.581  
## bathrooms:grade:sqft\_above 4.906e-05 1.876e-05 2.615  
## sqft\_living:grade:sqft\_above -1.142e-08 9.895e-09 -1.154  
## bedrooms:bathrooms:sqft\_living:grade -1.591e-06 3.454e-06 -0.461  
## bedrooms:bathrooms:sqft\_living:sqft\_above -1.414e-08 6.303e-09 -2.243  
## bedrooms:bathrooms:grade:sqft\_above -2.794e-06 4.240e-06 -0.659  
## bedrooms:sqft\_living:grade:sqft\_above -9.698e-11 2.153e-09 -0.045  
## bathrooms:sqft\_living:grade:sqft\_above -5.495e-09 2.937e-09 -1.871  
## bedrooms:bathrooms:sqft\_living:grade:sqft\_above 1.039e-09 5.300e-10 1.961  
## Pr(>|t|)   
## (Intercept) < 2e-16 \*\*\*  
## bedrooms 0.004807 \*\*   
## bathrooms 0.011118 \*   
## sqft\_living 0.271462   
## grade 0.188375   
## sqft\_above 0.618083   
## I(bedrooms^2) 0.000982 \*\*\*  
## I(bathrooms^2) 0.039250 \*   
## I(sqft\_living^2) < 2e-16 \*\*\*  
## I(grade^2) 0.008887 \*\*   
## I(sqft\_above^2) 1.52e-10 \*\*\*  
## I(bedrooms^3) 0.220830   
## I(bathrooms^3) 0.980535   
## I(sqft\_living^3) 5.42e-07 \*\*\*  
## I(grade^3) 0.024839 \*   
## I(sqft\_above^3) 0.003497 \*\*   
## bedrooms:bathrooms 0.266079   
## bedrooms:sqft\_living 0.635697   
## bathrooms:sqft\_living 0.690996   
## bedrooms:grade 0.302581   
## bathrooms:grade 0.002120 \*\*   
## sqft\_living:grade 0.617617   
## bedrooms:sqft\_above 0.651802   
## bathrooms:sqft\_above 0.002703 \*\*   
## sqft\_living:sqft\_above 0.000412 \*\*\*  
## grade:sqft\_above 0.453250   
## bedrooms:bathrooms:sqft\_living 0.747243   
## bedrooms:bathrooms:grade 0.656198   
## bedrooms:sqft\_living:grade 0.758880   
## bathrooms:sqft\_living:grade 0.762695   
## bedrooms:bathrooms:sqft\_above 0.177864   
## bedrooms:sqft\_living:sqft\_above 0.675488   
## bathrooms:sqft\_living:sqft\_above 0.149229   
## bedrooms:grade:sqft\_above 0.561503   
## bathrooms:grade:sqft\_above 0.008941 \*\*   
## sqft\_living:grade:sqft\_above 0.248490   
## bedrooms:bathrooms:sqft\_living:grade 0.644967   
## bedrooms:bathrooms:sqft\_living:sqft\_above 0.024907 \*   
## bedrooms:bathrooms:grade:sqft\_above 0.509874   
## bedrooms:sqft\_living:grade:sqft\_above 0.964072   
## bathrooms:sqft\_living:grade:sqft\_above 0.061315 .   
## bedrooms:bathrooms:sqft\_living:grade:sqft\_above 0.049855 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3433 on 21570 degrees of freedom  
## Multiple R-squared: 0.5761, Adjusted R-squared: 0.5753   
## F-statistic: 714.9 on 41 and 21570 DF, p-value: < 2.2e-16

fit 4

house.fit4 <- update(house.fit3, ~ . + I(bedrooms^0.5) + I(bathrooms^0.5) + I(sqft\_living^0.5) + I(grade^0.5) + I(sqft\_above^0.5))  
summary(house.fit4)

##   
## Call:  
## lm(formula = price ~ bedrooms + bathrooms + sqft\_living + grade +   
## sqft\_above + I(bedrooms^2) + I(bathrooms^2) + I(sqft\_living^2) +   
## I(grade^2) + I(sqft\_above^2) + I(bedrooms^3) + I(bathrooms^3) +   
## I(sqft\_living^3) + I(grade^3) + I(sqft\_above^3) + I(bedrooms^0.5) +   
## I(bathrooms^0.5) + I(sqft\_living^0.5) + I(grade^0.5) + I(sqft\_above^0.5) +   
## bedrooms:bathrooms + bedrooms:sqft\_living + bathrooms:sqft\_living +   
## bedrooms:grade + bathrooms:grade + sqft\_living:grade + bedrooms:sqft\_above +   
## bathrooms:sqft\_above + sqft\_living:sqft\_above + grade:sqft\_above +   
## bedrooms:bathrooms:sqft\_living + bedrooms:bathrooms:grade +   
## bedrooms:sqft\_living:grade + bathrooms:sqft\_living:grade +   
## bedrooms:bathrooms:sqft\_above + bedrooms:sqft\_living:sqft\_above +   
## bathrooms:sqft\_living:sqft\_above + bedrooms:grade:sqft\_above +   
## bathrooms:grade:sqft\_above + sqft\_living:grade:sqft\_above +   
## bedrooms:bathrooms:sqft\_living:grade + bedrooms:bathrooms:sqft\_living:sqft\_above +   
## bedrooms:bathrooms:grade:sqft\_above + bedrooms:sqft\_living:grade:sqft\_above +   
## bathrooms:sqft\_living:grade:sqft\_above + bedrooms:bathrooms:sqft\_living:grade:sqft\_above,   
## data = house)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.25905 -0.23959 0.00499 0.22729 1.43408   
##   
## Coefficients:  
## Estimate Std. Error t value  
## (Intercept) 1.426e+01 1.907e+00 7.478  
## bedrooms -9.121e-01 1.604e-01 -5.687  
## bathrooms -1.703e-01 3.755e-01 -0.454  
## sqft\_living 6.555e-04 7.832e-04 0.837  
## grade 7.781e-01 9.200e-01 0.846  
## sqft\_above 7.509e-04 8.917e-04 0.842  
## I(bedrooms^2) 9.462e-02 1.509e-02 6.271  
## I(bathrooms^2) 1.652e-01 3.874e-02 4.264  
## I(sqft\_living^2) -2.210e-07 4.566e-08 -4.841  
## I(grade^2) 1.575e-02 4.744e-02 0.332  
## I(sqft\_above^2) -2.356e-07 5.803e-08 -4.059  
## I(bedrooms^3) -4.098e-03 8.738e-04 -4.689  
## I(bathrooms^3) -1.021e-02 3.717e-03 -2.748  
## I(sqft\_living^3) 8.514e-12 2.407e-12 3.537  
## I(grade^3) -1.058e-03 1.420e-03 -0.745  
## I(sqft\_above^3) 1.033e-11 3.912e-12 2.641  
## I(bedrooms^0.5) 1.039e+00 2.199e-01 4.726  
## I(bathrooms^0.5) 9.900e-01 3.312e-01 2.989  
## I(sqft\_living^0.5) -6.821e-03 2.428e-02 -0.281  
## I(grade^0.5) -3.045e+00 2.484e+00 -1.226  
## I(sqft\_above^0.5) -2.037e-02 2.388e-02 -0.853  
## bedrooms:bathrooms -1.816e-02 7.194e-02 -0.252  
## bedrooms:sqft\_living -6.483e-05 1.079e-04 -0.601  
## bathrooms:sqft\_living 8.072e-05 1.557e-04 0.518  
## bedrooms:grade 2.619e-02 1.559e-02 1.680  
## bathrooms:grade -8.354e-02 3.456e-02 -2.417  
## sqft\_living:grade 7.332e-06 5.660e-05 0.130  
## bedrooms:sqft\_above 1.004e-05 1.258e-04 0.080  
## bathrooms:sqft\_above -5.436e-04 1.947e-04 -2.791  
## sqft\_living:sqft\_above 3.242e-07 1.582e-07 2.049  
## grade:sqft\_above -3.872e-05 6.244e-05 -0.620  
## bedrooms:bathrooms:sqft\_living -5.539e-06 3.335e-05 -0.166  
## bedrooms:bathrooms:grade -3.836e-03 9.796e-03 -0.392  
## bedrooms:sqft\_living:grade 5.763e-06 1.313e-05 0.439  
## bathrooms:sqft\_living:grade 3.084e-06 1.786e-05 0.173  
## bedrooms:bathrooms:sqft\_above 5.586e-05 4.299e-05 1.299  
## bedrooms:sqft\_living:sqft\_above 2.314e-08 2.998e-08 0.772  
## bathrooms:sqft\_living:sqft\_above 4.431e-08 4.187e-08 1.059  
## bedrooms:grade:sqft\_above -7.381e-06 1.443e-05 -0.511  
## bathrooms:grade:sqft\_above 4.657e-05 2.131e-05 2.185  
## sqft\_living:grade:sqft\_above -5.911e-09 1.423e-08 -0.415  
## bedrooms:bathrooms:sqft\_living:grade -2.011e-07 3.765e-06 -0.053  
## bedrooms:bathrooms:sqft\_living:sqft\_above -1.314e-08 7.483e-09 -1.755  
## bedrooms:bathrooms:grade:sqft\_above -1.946e-06 4.778e-06 -0.407  
## bedrooms:sqft\_living:grade:sqft\_above -1.557e-09 2.691e-09 -0.579  
## bathrooms:sqft\_living:grade:sqft\_above -5.124e-09 3.563e-09 -1.438  
## bedrooms:bathrooms:sqft\_living:grade:sqft\_above 9.440e-10 6.060e-10 1.558  
## Pr(>|t|)   
## (Intercept) 7.80e-14 \*\*\*  
## bedrooms 1.31e-08 \*\*\*  
## bathrooms 0.650154   
## sqft\_living 0.402592   
## grade 0.397673   
## sqft\_above 0.399747   
## I(bedrooms^2) 3.64e-10 \*\*\*  
## I(bathrooms^2) 2.01e-05 \*\*\*  
## I(sqft\_living^2) 1.30e-06 \*\*\*  
## I(grade^2) 0.739883   
## I(sqft\_above^2) 4.95e-05 \*\*\*  
## I(bedrooms^3) 2.76e-06 \*\*\*  
## I(bathrooms^3) 0.005997 \*\*   
## I(sqft\_living^3) 0.000405 \*\*\*  
## I(grade^3) 0.456467   
## I(sqft\_above^3) 0.008284 \*\*   
## I(bedrooms^0.5) 2.30e-06 \*\*\*  
## I(bathrooms^0.5) 0.002800 \*\*   
## I(sqft\_living^0.5) 0.778774   
## I(grade^0.5) 0.220275   
## I(sqft\_above^0.5) 0.393708   
## bedrooms:bathrooms 0.800666   
## bedrooms:sqft\_living 0.547958   
## bathrooms:sqft\_living 0.604261   
## bedrooms:grade 0.092947 .   
## bathrooms:grade 0.015651 \*   
## sqft\_living:grade 0.896926   
## bedrooms:sqft\_above 0.936435   
## bathrooms:sqft\_above 0.005253 \*\*   
## sqft\_living:sqft\_above 0.040505 \*   
## grade:sqft\_above 0.535122   
## bedrooms:bathrooms:sqft\_living 0.868082   
## bedrooms:bathrooms:grade 0.695335   
## bedrooms:sqft\_living:grade 0.660834   
## bathrooms:sqft\_living:grade 0.862923   
## bedrooms:bathrooms:sqft\_above 0.193829   
## bedrooms:sqft\_living:sqft\_above 0.440187   
## bathrooms:sqft\_living:sqft\_above 0.289833   
## bedrooms:grade:sqft\_above 0.609104   
## bathrooms:grade:sqft\_above 0.028880 \*   
## sqft\_living:grade:sqft\_above 0.677789   
## bedrooms:bathrooms:sqft\_living:grade 0.957409   
## bedrooms:bathrooms:sqft\_living:sqft\_above 0.079227 .   
## bedrooms:bathrooms:grade:sqft\_above 0.683760   
## bedrooms:sqft\_living:grade:sqft\_above 0.562815   
## bathrooms:sqft\_living:grade:sqft\_above 0.150388   
## bedrooms:bathrooms:sqft\_living:grade:sqft\_above 0.119307   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3428 on 21565 degrees of freedom  
## Multiple R-squared: 0.5772, Adjusted R-squared: 0.5763   
## F-statistic: 640 on 46 and 21565 DF, p-value: < 2.2e-16

fit 4 : step function

house.fit5 <- step(house.fit4)

## Start: AIC=-46224.9  
## price ~ bedrooms + bathrooms + sqft\_living + grade + sqft\_above +   
## I(bedrooms^2) + I(bathrooms^2) + I(sqft\_living^2) + I(grade^2) +   
## I(sqft\_above^2) + I(bedrooms^3) + I(bathrooms^3) + I(sqft\_living^3) +   
## I(grade^3) + I(sqft\_above^3) + I(bedrooms^0.5) + I(bathrooms^0.5) +   
## I(sqft\_living^0.5) + I(grade^0.5) + I(sqft\_above^0.5) + bedrooms:bathrooms +   
## bedrooms:sqft\_living + bathrooms:sqft\_living + bedrooms:grade +   
## bathrooms:grade + sqft\_living:grade + bedrooms:sqft\_above +   
## bathrooms:sqft\_above + sqft\_living:sqft\_above + grade:sqft\_above +   
## bedrooms:bathrooms:sqft\_living + bedrooms:bathrooms:grade +   
## bedrooms:sqft\_living:grade + bathrooms:sqft\_living:grade +   
## bedrooms:bathrooms:sqft\_above + bedrooms:sqft\_living:sqft\_above +   
## bathrooms:sqft\_living:sqft\_above + bedrooms:grade:sqft\_above +   
## bathrooms:grade:sqft\_above + sqft\_living:grade:sqft\_above +   
## bedrooms:bathrooms:sqft\_living:grade + bedrooms:bathrooms:sqft\_living:sqft\_above +   
## bedrooms:bathrooms:grade:sqft\_above + bedrooms:sqft\_living:grade:sqft\_above +   
## bathrooms:sqft\_living:grade:sqft\_above + bedrooms:bathrooms:sqft\_living:grade:sqft\_above  
##   
## Df Sum of Sq RSS AIC  
## - I(sqft\_living^0.5) 1 0.0093 2534.6 -46227  
## - I(grade^2) 1 0.0130 2534.6 -46227  
## - I(grade^3) 1 0.0652 2534.7 -46226  
## - I(sqft\_above^0.5) 1 0.0855 2534.7 -46226  
## - I(grade^0.5) 1 0.1766 2534.8 -46225  
## <none> 2534.6 -46225  
## - bedrooms:bathrooms:sqft\_living:grade:sqft\_above 1 0.2852 2534.9 -46224  
## - I(sqft\_above^3) 1 0.8195 2535.4 -46220  
## - I(bathrooms^3) 1 0.8877 2535.5 -46219  
## - I(bathrooms^0.5) 1 1.0502 2535.7 -46218  
## - I(sqft\_living^3) 1 1.4706 2536.1 -46214  
## - I(sqft\_above^2) 1 1.9363 2536.6 -46210  
## - I(bathrooms^2) 1 2.1374 2536.8 -46209  
## - I(bedrooms^3) 1 2.5847 2537.2 -46205  
## - I(bedrooms^0.5) 1 2.6253 2537.2 -46205  
## - I(sqft\_living^2) 1 2.7539 2537.4 -46203  
## - I(bedrooms^2) 1 4.6228 2539.2 -46188  
##   
## Step: AIC=-46226.82  
## price ~ bedrooms + bathrooms + sqft\_living + grade + sqft\_above +   
## I(bedrooms^2) + I(bathrooms^2) + I(sqft\_living^2) + I(grade^2) +   
## I(sqft\_above^2) + I(bedrooms^3) + I(bathrooms^3) + I(sqft\_living^3) +   
## I(grade^3) + I(sqft\_above^3) + I(bedrooms^0.5) + I(bathrooms^0.5) +   
## I(grade^0.5) + I(sqft\_above^0.5) + bedrooms:bathrooms + bedrooms:sqft\_living +   
## bathrooms:sqft\_living + bedrooms:grade + bathrooms:grade +   
## sqft\_living:grade + bedrooms:sqft\_above + bathrooms:sqft\_above +   
## sqft\_living:sqft\_above + grade:sqft\_above + bedrooms:bathrooms:sqft\_living +   
## bedrooms:bathrooms:grade + bedrooms:sqft\_living:grade + bathrooms:sqft\_living:grade +   
## bedrooms:bathrooms:sqft\_above + bedrooms:sqft\_living:sqft\_above +   
## bathrooms:sqft\_living:sqft\_above + bedrooms:grade:sqft\_above +   
## bathrooms:grade:sqft\_above + sqft\_living:grade:sqft\_above +   
## bedrooms:bathrooms:sqft\_living:grade + bedrooms:bathrooms:sqft\_living:sqft\_above +   
## bedrooms:bathrooms:grade:sqft\_above + bedrooms:sqft\_living:grade:sqft\_above +   
## bathrooms:sqft\_living:grade:sqft\_above + bedrooms:bathrooms:sqft\_living:grade:sqft\_above  
##   
## Df Sum of Sq RSS AIC  
## - I(grade^2) 1 0.0111 2534.7 -46229  
## - I(grade^3) 1 0.0607 2534.7 -46228  
## - I(sqft\_above^0.5) 1 0.0889 2534.7 -46228  
## - I(grade^0.5) 1 0.1821 2534.8 -46227  
## <none> 2534.6 -46227  
## - bedrooms:bathrooms:sqft\_living:grade:sqft\_above 1 0.2816 2534.9 -46226  
## - I(sqft\_above^3) 1 0.8368 2535.5 -46222  
## - I(bathrooms^3) 1 0.9223 2535.6 -46221  
## - I(bathrooms^0.5) 1 1.0620 2535.7 -46220  
## - I(sqft\_above^2) 1 1.9489 2536.6 -46212  
## - I(bathrooms^2) 1 2.1900 2536.8 -46210  
## - I(bedrooms^3) 1 2.5778 2537.2 -46207  
## - I(bedrooms^0.5) 1 2.6165 2537.2 -46207  
## - I(sqft\_living^3) 1 2.6244 2537.3 -46206  
## - I(bedrooms^2) 1 4.6238 2539.3 -46189  
## - I(sqft\_living^2) 1 9.3137 2543.9 -46150  
##   
## Step: AIC=-46228.72  
## price ~ bedrooms + bathrooms + sqft\_living + grade + sqft\_above +   
## I(bedrooms^2) + I(bathrooms^2) + I(sqft\_living^2) + I(sqft\_above^2) +   
## I(bedrooms^3) + I(bathrooms^3) + I(sqft\_living^3) + I(grade^3) +   
## I(sqft\_above^3) + I(bedrooms^0.5) + I(bathrooms^0.5) + I(grade^0.5) +   
## I(sqft\_above^0.5) + bedrooms:bathrooms + bedrooms:sqft\_living +   
## bathrooms:sqft\_living + bedrooms:grade + bathrooms:grade +   
## sqft\_living:grade + bedrooms:sqft\_above + bathrooms:sqft\_above +   
## sqft\_living:sqft\_above + grade:sqft\_above + bedrooms:bathrooms:sqft\_living +   
## bedrooms:bathrooms:grade + bedrooms:sqft\_living:grade + bathrooms:sqft\_living:grade +   
## bedrooms:bathrooms:sqft\_above + bedrooms:sqft\_living:sqft\_above +   
## bathrooms:sqft\_living:sqft\_above + bedrooms:grade:sqft\_above +   
## bathrooms:grade:sqft\_above + sqft\_living:grade:sqft\_above +   
## bedrooms:bathrooms:sqft\_living:grade + bedrooms:bathrooms:sqft\_living:sqft\_above +   
## bedrooms:bathrooms:grade:sqft\_above + bedrooms:sqft\_living:grade:sqft\_above +   
## bathrooms:sqft\_living:grade:sqft\_above + bedrooms:bathrooms:sqft\_living:grade:sqft\_above  
##   
## Df Sum of Sq RSS AIC  
## - I(sqft\_above^0.5) 1 0.0896 2534.7 -46230  
## <none> 2534.7 -46229  
## - bedrooms:bathrooms:sqft\_living:grade:sqft\_above 1 0.2924 2534.9 -46228  
## - I(grade^3) 1 0.6227 2535.3 -46225  
## - I(sqft\_above^3) 1 0.8548 2535.5 -46223  
## - I(bathrooms^3) 1 0.9191 2535.6 -46223  
## - I(bathrooms^0.5) 1 1.0837 2535.7 -46221  
## - I(grade^0.5) 1 1.6559 2536.3 -46217  
## - I(sqft\_above^2) 1 1.9678 2536.6 -46214  
## - I(bathrooms^2) 1 2.1919 2536.8 -46212  
## - I(bedrooms^3) 1 2.6345 2537.3 -46208  
## - I(sqft\_living^3) 1 2.6478 2537.3 -46208  
## - I(bedrooms^0.5) 1 2.6635 2537.3 -46208  
## - I(bedrooms^2) 1 4.7197 2539.4 -46191  
## - I(sqft\_living^2) 1 9.3456 2544.0 -46151  
##   
## Step: AIC=-46229.96  
## price ~ bedrooms + bathrooms + sqft\_living + grade + sqft\_above +   
## I(bedrooms^2) + I(bathrooms^2) + I(sqft\_living^2) + I(sqft\_above^2) +   
## I(bedrooms^3) + I(bathrooms^3) + I(sqft\_living^3) + I(grade^3) +   
## I(sqft\_above^3) + I(bedrooms^0.5) + I(bathrooms^0.5) + I(grade^0.5) +   
## bedrooms:bathrooms + bedrooms:sqft\_living + bathrooms:sqft\_living +   
## bedrooms:grade + bathrooms:grade + sqft\_living:grade + bedrooms:sqft\_above +   
## bathrooms:sqft\_above + sqft\_living:sqft\_above + grade:sqft\_above +   
## bedrooms:bathrooms:sqft\_living + bedrooms:bathrooms:grade +   
## bedrooms:sqft\_living:grade + bathrooms:sqft\_living:grade +   
## bedrooms:bathrooms:sqft\_above + bedrooms:sqft\_living:sqft\_above +   
## bathrooms:sqft\_living:sqft\_above + bedrooms:grade:sqft\_above +   
## bathrooms:grade:sqft\_above + sqft\_living:grade:sqft\_above +   
## bedrooms:bathrooms:sqft\_living:grade + bedrooms:bathrooms:sqft\_living:sqft\_above +   
## bedrooms:bathrooms:grade:sqft\_above + bedrooms:sqft\_living:grade:sqft\_above +   
## bathrooms:sqft\_living:grade:sqft\_above + bedrooms:bathrooms:sqft\_living:grade:sqft\_above  
##   
## Df Sum of Sq RSS AIC  
## - bedrooms:bathrooms:sqft\_living:grade:sqft\_above 1 0.2162 2534.9 -46230  
## <none> 2534.7 -46230  
## - I(grade^3) 1 0.5991 2535.3 -46227  
## - I(bathrooms^3) 1 0.9298 2535.7 -46224  
## - I(sqft\_above^3) 1 1.0462 2535.8 -46223  
## - I(bathrooms^0.5) 1 1.0930 2535.8 -46223  
## - I(grade^0.5) 1 1.6219 2536.4 -46218  
## - I(bathrooms^2) 1 2.2003 2536.9 -46213  
## - I(bedrooms^3) 1 2.6029 2537.3 -46210  
## - I(bedrooms^0.5) 1 2.6373 2537.4 -46209  
## - I(sqft\_living^3) 1 2.8736 2537.6 -46207  
## - I(bedrooms^2) 1 4.6757 2539.4 -46192  
## - I(sqft\_above^2) 1 4.7710 2539.5 -46191  
## - I(sqft\_living^2) 1 9.8476 2544.6 -46148  
##   
## Step: AIC=-46230.11  
## price ~ bedrooms + bathrooms + sqft\_living + grade + sqft\_above +   
## I(bedrooms^2) + I(bathrooms^2) + I(sqft\_living^2) + I(sqft\_above^2) +   
## I(bedrooms^3) + I(bathrooms^3) + I(sqft\_living^3) + I(grade^3) +   
## I(sqft\_above^3) + I(bedrooms^0.5) + I(bathrooms^0.5) + I(grade^0.5) +   
## bedrooms:bathrooms + bedrooms:sqft\_living + bathrooms:sqft\_living +   
## bedrooms:grade + bathrooms:grade + sqft\_living:grade + bedrooms:sqft\_above +   
## bathrooms:sqft\_above + sqft\_living:sqft\_above + grade:sqft\_above +   
## bedrooms:bathrooms:sqft\_living + bedrooms:bathrooms:grade +   
## bedrooms:sqft\_living:grade + bathrooms:sqft\_living:grade +   
## bedrooms:bathrooms:sqft\_above + bedrooms:sqft\_living:sqft\_above +   
## bathrooms:sqft\_living:sqft\_above + bedrooms:grade:sqft\_above +   
## bathrooms:grade:sqft\_above + sqft\_living:grade:sqft\_above +   
## bedrooms:bathrooms:sqft\_living:grade + bedrooms:bathrooms:sqft\_living:sqft\_above +   
## bedrooms:bathrooms:grade:sqft\_above + bedrooms:sqft\_living:grade:sqft\_above +   
## bathrooms:sqft\_living:grade:sqft\_above  
##   
## Df Sum of Sq RSS AIC  
## - bedrooms:bathrooms:sqft\_living:grade 1 0.0002 2534.9 -46232  
## - bathrooms:sqft\_living:grade:sqft\_above 1 0.0076 2535.0 -46232  
## - bedrooms:bathrooms:grade:sqft\_above 1 0.0108 2535.0 -46232  
## - bedrooms:sqft\_living:grade:sqft\_above 1 0.0584 2535.0 -46232  
## - bedrooms:bathrooms:sqft\_living:sqft\_above 1 0.2057 2535.2 -46230  
## <none> 2534.9 -46230  
## - I(grade^3) 1 0.5248 2535.5 -46228  
## - I(sqft\_above^3) 1 0.9322 2535.9 -46224  
## - I(bathrooms^3) 1 1.0929 2536.1 -46223  
## - I(bathrooms^0.5) 1 1.1951 2536.2 -46222  
## - I(grade^0.5) 1 1.4852 2536.4 -46219  
## - I(bathrooms^2) 1 2.4687 2537.4 -46211  
## - I(bedrooms^3) 1 2.5286 2537.5 -46211  
## - I(bedrooms^0.5) 1 2.6393 2537.6 -46210  
## - I(sqft\_living^3) 1 2.8390 2537.8 -46208  
## - I(sqft\_above^2) 1 4.6053 2539.6 -46193  
## - I(bedrooms^2) 1 4.6084 2539.6 -46193  
## - I(sqft\_living^2) 1 9.7931 2544.8 -46149  
##   
## Step: AIC=-46232.11  
## price ~ bedrooms + bathrooms + sqft\_living + grade + sqft\_above +   
## I(bedrooms^2) + I(bathrooms^2) + I(sqft\_living^2) + I(sqft\_above^2) +   
## I(bedrooms^3) + I(bathrooms^3) + I(sqft\_living^3) + I(grade^3) +   
## I(sqft\_above^3) + I(bedrooms^0.5) + I(bathrooms^0.5) + I(grade^0.5) +   
## bedrooms:bathrooms + bedrooms:sqft\_living + bathrooms:sqft\_living +   
## bedrooms:grade + bathrooms:grade + sqft\_living:grade + bedrooms:sqft\_above +   
## bathrooms:sqft\_above + sqft\_living:sqft\_above + grade:sqft\_above +   
## bedrooms:bathrooms:sqft\_living + bedrooms:bathrooms:grade +   
## bedrooms:sqft\_living:grade + bathrooms:sqft\_living:grade +   
## bedrooms:bathrooms:sqft\_above + bedrooms:sqft\_living:sqft\_above +   
## bathrooms:sqft\_living:sqft\_above + bedrooms:grade:sqft\_above +   
## bathrooms:grade:sqft\_above + sqft\_living:grade:sqft\_above +   
## bedrooms:bathrooms:sqft\_living:sqft\_above + bedrooms:bathrooms:grade:sqft\_above +   
## bedrooms:sqft\_living:grade:sqft\_above + bathrooms:sqft\_living:grade:sqft\_above  
##   
## Df Sum of Sq RSS AIC  
## - bathrooms:sqft\_living:grade:sqft\_above 1 0.0075 2535.0 -46234  
## - bedrooms:bathrooms:grade:sqft\_above 1 0.0248 2535.0 -46234  
## - bedrooms:sqft\_living:grade:sqft\_above 1 0.0587 2535.0 -46234  
## - bedrooms:bathrooms:sqft\_living:sqft\_above 1 0.2273 2535.2 -46232  
## <none> 2534.9 -46232  
## - I(grade^3) 1 0.5317 2535.5 -46230  
## - I(sqft\_above^3) 1 0.9726 2535.9 -46226  
## - I(bathrooms^3) 1 1.1051 2536.1 -46225  
## - I(bathrooms^0.5) 1 1.2007 2536.2 -46224  
## - I(grade^0.5) 1 1.5145 2536.5 -46221  
## - I(bathrooms^2) 1 2.4920 2537.4 -46213  
## - I(bedrooms^3) 1 2.5349 2537.5 -46213  
## - I(bedrooms^0.5) 1 2.6420 2537.6 -46212  
## - I(sqft\_living^3) 1 2.9055 2537.9 -46209  
## - I(bedrooms^2) 1 4.6163 2539.6 -46195  
## - I(sqft\_above^2) 1 4.7665 2539.7 -46194  
## - I(sqft\_living^2) 1 9.8910 2544.8 -46150  
##   
## Step: AIC=-46234.05  
## price ~ bedrooms + bathrooms + sqft\_living + grade + sqft\_above +   
## I(bedrooms^2) + I(bathrooms^2) + I(sqft\_living^2) + I(sqft\_above^2) +   
## I(bedrooms^3) + I(bathrooms^3) + I(sqft\_living^3) + I(grade^3) +   
## I(sqft\_above^3) + I(bedrooms^0.5) + I(bathrooms^0.5) + I(grade^0.5) +   
## bedrooms:bathrooms + bedrooms:sqft\_living + bathrooms:sqft\_living +   
## bedrooms:grade + bathrooms:grade + sqft\_living:grade + bedrooms:sqft\_above +   
## bathrooms:sqft\_above + sqft\_living:sqft\_above + grade:sqft\_above +   
## bedrooms:bathrooms:sqft\_living + bedrooms:bathrooms:grade +   
## bedrooms:sqft\_living:grade + bathrooms:sqft\_living:grade +   
## bedrooms:bathrooms:sqft\_above + bedrooms:sqft\_living:sqft\_above +   
## bathrooms:sqft\_living:sqft\_above + bedrooms:grade:sqft\_above +   
## bathrooms:grade:sqft\_above + sqft\_living:grade:sqft\_above +   
## bedrooms:bathrooms:sqft\_living:sqft\_above + bedrooms:bathrooms:grade:sqft\_above +   
## bedrooms:sqft\_living:grade:sqft\_above  
##   
## Df Sum of Sq RSS AIC  
## - bathrooms:sqft\_living:grade 1 0.0046 2535.0 -46236  
## - bedrooms:bathrooms:grade:sqft\_above 1 0.0324 2535.0 -46236  
## - bedrooms:sqft\_living:grade:sqft\_above 1 0.0852 2535.1 -46235  
## - bedrooms:bathrooms:sqft\_living:sqft\_above 1 0.2296 2535.2 -46234  
## <none> 2535.0 -46234  
## - I(grade^3) 1 0.5252 2535.5 -46232  
## - I(sqft\_above^3) 1 0.9651 2535.9 -46228  
## - I(bathrooms^3) 1 1.0975 2536.1 -46227  
## - I(bathrooms^0.5) 1 1.1944 2536.2 -46226  
## - I(grade^0.5) 1 1.5080 2536.5 -46223  
## - I(bathrooms^2) 1 2.4885 2537.4 -46215  
## - I(bedrooms^3) 1 2.5357 2537.5 -46214  
## - I(bedrooms^0.5) 1 2.6747 2537.6 -46213  
## - I(sqft\_living^3) 1 2.8983 2537.9 -46211  
## - I(bedrooms^2) 1 4.6185 2539.6 -46197  
## - I(sqft\_above^2) 1 4.7643 2539.7 -46195  
## - I(sqft\_living^2) 1 9.8835 2544.8 -46152  
##   
## Step: AIC=-46236.01  
## price ~ bedrooms + bathrooms + sqft\_living + grade + sqft\_above +   
## I(bedrooms^2) + I(bathrooms^2) + I(sqft\_living^2) + I(sqft\_above^2) +   
## I(bedrooms^3) + I(bathrooms^3) + I(sqft\_living^3) + I(grade^3) +   
## I(sqft\_above^3) + I(bedrooms^0.5) + I(bathrooms^0.5) + I(grade^0.5) +   
## bedrooms:bathrooms + bedrooms:sqft\_living + bathrooms:sqft\_living +   
## bedrooms:grade + bathrooms:grade + sqft\_living:grade + bedrooms:sqft\_above +   
## bathrooms:sqft\_above + sqft\_living:sqft\_above + grade:sqft\_above +   
## bedrooms:bathrooms:sqft\_living + bedrooms:bathrooms:grade +   
## bedrooms:sqft\_living:grade + bedrooms:bathrooms:sqft\_above +   
## bedrooms:sqft\_living:sqft\_above + bathrooms:sqft\_living:sqft\_above +   
## bedrooms:grade:sqft\_above + bathrooms:grade:sqft\_above +   
## sqft\_living:grade:sqft\_above + bedrooms:bathrooms:sqft\_living:sqft\_above +   
## bedrooms:bathrooms:grade:sqft\_above + bedrooms:sqft\_living:grade:sqft\_above  
##   
## Df Sum of Sq RSS AIC  
## - bedrooms:bathrooms:grade:sqft\_above 1 0.0301 2535.0 -46238  
## - bedrooms:sqft\_living:grade:sqft\_above 1 0.0837 2535.1 -46237  
## - bedrooms:bathrooms:sqft\_living:sqft\_above 1 0.2343 2535.2 -46236  
## <none> 2535.0 -46236  
## - I(grade^3) 1 0.5515 2535.5 -46233  
## - I(bathrooms^0.5) 1 1.2220 2536.2 -46228  
## - I(bathrooms^3) 1 1.2580 2536.2 -46227  
## - I(sqft\_above^3) 1 1.3709 2536.3 -46226  
## - I(grade^0.5) 1 1.5881 2536.6 -46224  
## - I(bedrooms^3) 1 2.5335 2537.5 -46216  
## - I(bedrooms^0.5) 1 2.6849 2537.7 -46215  
## - I(bathrooms^2) 1 2.7456 2537.7 -46215  
## - I(sqft\_living^3) 1 3.0341 2538.0 -46212  
## - I(bedrooms^2) 1 4.6174 2539.6 -46199  
## - I(sqft\_above^2) 1 5.9123 2540.9 -46188  
## - I(sqft\_living^2) 1 9.9709 2544.9 -46153  
##   
## Step: AIC=-46237.75  
## price ~ bedrooms + bathrooms + sqft\_living + grade + sqft\_above +   
## I(bedrooms^2) + I(bathrooms^2) + I(sqft\_living^2) + I(sqft\_above^2) +   
## I(bedrooms^3) + I(bathrooms^3) + I(sqft\_living^3) + I(grade^3) +   
## I(sqft\_above^3) + I(bedrooms^0.5) + I(bathrooms^0.5) + I(grade^0.5) +   
## bedrooms:bathrooms + bedrooms:sqft\_living + bathrooms:sqft\_living +   
## bedrooms:grade + bathrooms:grade + sqft\_living:grade + bedrooms:sqft\_above +   
## bathrooms:sqft\_above + sqft\_living:sqft\_above + grade:sqft\_above +   
## bedrooms:bathrooms:sqft\_living + bedrooms:bathrooms:grade +   
## bedrooms:sqft\_living:grade + bedrooms:bathrooms:sqft\_above +   
## bedrooms:sqft\_living:sqft\_above + bathrooms:sqft\_living:sqft\_above +   
## bedrooms:grade:sqft\_above + bathrooms:grade:sqft\_above +   
## sqft\_living:grade:sqft\_above + bedrooms:bathrooms:sqft\_living:sqft\_above +   
## bedrooms:sqft\_living:grade:sqft\_above  
##   
## Df Sum of Sq RSS AIC  
## - bedrooms:bathrooms:grade 1 0.0028 2535.0 -46240  
## - bedrooms:bathrooms:sqft\_living:sqft\_above 1 0.2073 2535.2 -46238  
## <none> 2535.0 -46238  
## - bedrooms:sqft\_living:grade:sqft\_above 1 0.2843 2535.3 -46237  
## - I(grade^3) 1 0.6794 2535.7 -46234  
## - I(bathrooms^0.5) 1 1.1959 2536.2 -46230  
## - I(bathrooms^3) 1 1.2383 2536.2 -46229  
## - I(sqft\_above^3) 1 1.3447 2536.3 -46228  
## - I(grade^0.5) 1 1.9517 2536.9 -46223  
## - I(bedrooms^3) 1 2.6065 2537.6 -46218  
## - I(bathrooms^2) 1 2.7162 2537.7 -46217  
## - I(bedrooms^0.5) 1 2.7557 2537.8 -46216  
## - I(sqft\_living^3) 1 3.3579 2538.3 -46211  
## - bathrooms:grade:sqft\_above 1 3.5940 2538.6 -46209  
## - I(bedrooms^2) 1 4.7258 2539.7 -46200  
## - I(sqft\_above^2) 1 5.9419 2540.9 -46189  
## - I(sqft\_living^2) 1 10.7905 2545.8 -46148  
##   
## Step: AIC=-46239.73  
## price ~ bedrooms + bathrooms + sqft\_living + grade + sqft\_above +   
## I(bedrooms^2) + I(bathrooms^2) + I(sqft\_living^2) + I(sqft\_above^2) +   
## I(bedrooms^3) + I(bathrooms^3) + I(sqft\_living^3) + I(grade^3) +   
## I(sqft\_above^3) + I(bedrooms^0.5) + I(bathrooms^0.5) + I(grade^0.5) +   
## bedrooms:bathrooms + bedrooms:sqft\_living + bathrooms:sqft\_living +   
## bedrooms:grade + bathrooms:grade + sqft\_living:grade + bedrooms:sqft\_above +   
## bathrooms:sqft\_above + sqft\_living:sqft\_above + grade:sqft\_above +   
## bedrooms:bathrooms:sqft\_living + bedrooms:sqft\_living:grade +   
## bedrooms:bathrooms:sqft\_above + bedrooms:sqft\_living:sqft\_above +   
## bathrooms:sqft\_living:sqft\_above + bedrooms:grade:sqft\_above +   
## bathrooms:grade:sqft\_above + sqft\_living:grade:sqft\_above +   
## bedrooms:bathrooms:sqft\_living:sqft\_above + bedrooms:sqft\_living:grade:sqft\_above  
##   
## Df Sum of Sq RSS AIC  
## - bedrooms:bathrooms:sqft\_living:sqft\_above 1 0.2057 2535.2 -46240  
## <none> 2535.0 -46240  
## - bedrooms:sqft\_living:grade:sqft\_above 1 0.2815 2535.3 -46239  
## - I(grade^3) 1 0.6773 2535.7 -46236  
## - I(bathrooms^0.5) 1 1.2240 2536.2 -46231  
## - I(bathrooms^3) 1 1.3652 2536.4 -46230  
## - I(sqft\_above^3) 1 1.4293 2536.4 -46230  
## - I(grade^0.5) 1 1.9567 2536.9 -46225  
## - I(bedrooms^3) 1 2.6108 2537.6 -46219  
## - I(bedrooms^0.5) 1 2.7588 2537.8 -46218  
## - I(bathrooms^2) 1 2.8813 2537.9 -46217  
## - I(sqft\_living^3) 1 3.4652 2538.5 -46212  
## - bathrooms:grade:sqft\_above 1 4.2362 2539.2 -46206  
## - I(bedrooms^2) 1 4.7248 2539.7 -46201  
## - I(sqft\_above^2) 1 6.1993 2541.2 -46189  
## - I(sqft\_living^2) 1 11.1073 2546.1 -46147  
##   
## Step: AIC=-46239.98  
## price ~ bedrooms + bathrooms + sqft\_living + grade + sqft\_above +   
## I(bedrooms^2) + I(bathrooms^2) + I(sqft\_living^2) + I(sqft\_above^2) +   
## I(bedrooms^3) + I(bathrooms^3) + I(sqft\_living^3) + I(grade^3) +   
## I(sqft\_above^3) + I(bedrooms^0.5) + I(bathrooms^0.5) + I(grade^0.5) +   
## bedrooms:bathrooms + bedrooms:sqft\_living + bathrooms:sqft\_living +   
## bedrooms:grade + bathrooms:grade + sqft\_living:grade + bedrooms:sqft\_above +   
## bathrooms:sqft\_above + sqft\_living:sqft\_above + grade:sqft\_above +   
## bedrooms:bathrooms:sqft\_living + bedrooms:sqft\_living:grade +   
## bedrooms:bathrooms:sqft\_above + bedrooms:sqft\_living:sqft\_above +   
## bathrooms:sqft\_living:sqft\_above + bedrooms:grade:sqft\_above +   
## bathrooms:grade:sqft\_above + sqft\_living:grade:sqft\_above +   
## bedrooms:sqft\_living:grade:sqft\_above  
##   
## Df Sum of Sq RSS AIC  
## - bedrooms:sqft\_living:grade:sqft\_above 1 0.0893 2535.3 -46241  
## <none> 2535.2 -46240  
## - bedrooms:bathrooms:sqft\_living 1 0.9857 2536.2 -46234  
## - I(grade^3) 1 1.0088 2536.2 -46233  
## - bedrooms:bathrooms:sqft\_above 1 1.0799 2536.3 -46233  
## - I(bathrooms^3) 1 1.3254 2536.5 -46231  
## - I(sqft\_above^3) 1 1.3664 2536.6 -46230  
## - I(bathrooms^0.5) 1 1.3991 2536.6 -46230  
## - bathrooms:sqft\_living:sqft\_above 1 2.0040 2537.2 -46225  
## - I(bedrooms^3) 1 2.4741 2537.7 -46221  
## - I(bedrooms^0.5) 1 2.6622 2537.9 -46219  
## - I(grade^0.5) 1 2.8316 2538.0 -46218  
## - I(bathrooms^2) 1 2.9139 2538.1 -46217  
## - I(sqft\_living^3) 1 3.2595 2538.5 -46214  
## - I(bedrooms^2) 1 4.5854 2539.8 -46203  
## - bathrooms:grade:sqft\_above 1 4.6015 2539.8 -46203  
## - I(sqft\_above^2) 1 6.1192 2541.3 -46190  
## - I(sqft\_living^2) 1 10.9348 2546.1 -46149  
##   
## Step: AIC=-46241.21  
## price ~ bedrooms + bathrooms + sqft\_living + grade + sqft\_above +   
## I(bedrooms^2) + I(bathrooms^2) + I(sqft\_living^2) + I(sqft\_above^2) +   
## I(bedrooms^3) + I(bathrooms^3) + I(sqft\_living^3) + I(grade^3) +   
## I(sqft\_above^3) + I(bedrooms^0.5) + I(bathrooms^0.5) + I(grade^0.5) +   
## bedrooms:bathrooms + bedrooms:sqft\_living + bathrooms:sqft\_living +   
## bedrooms:grade + bathrooms:grade + sqft\_living:grade + bedrooms:sqft\_above +   
## bathrooms:sqft\_above + sqft\_living:sqft\_above + grade:sqft\_above +   
## bedrooms:bathrooms:sqft\_living + bedrooms:sqft\_living:grade +   
## bedrooms:bathrooms:sqft\_above + bedrooms:sqft\_living:sqft\_above +   
## bathrooms:sqft\_living:sqft\_above + bedrooms:grade:sqft\_above +   
## bathrooms:grade:sqft\_above + sqft\_living:grade:sqft\_above  
##   
## Df Sum of Sq RSS AIC  
## - bedrooms:sqft\_living:sqft\_above 1 0.0563 2535.3 -46243  
## <none> 2535.3 -46241  
## - bedrooms:sqft\_living:grade 1 0.2439 2535.5 -46241  
## - bedrooms:grade:sqft\_above 1 0.2808 2535.6 -46241  
## - bedrooms:bathrooms:sqft\_living 1 1.0104 2536.3 -46235  
## - bedrooms:bathrooms:sqft\_above 1 1.0105 2536.3 -46235  
## - I(grade^3) 1 1.1153 2536.4 -46234  
## - I(bathrooms^3) 1 1.2838 2536.6 -46232  
## - I(sqft\_above^3) 1 1.2959 2536.6 -46232  
## - I(bathrooms^0.5) 1 1.3617 2536.7 -46232  
## - bathrooms:sqft\_living:sqft\_above 1 1.9322 2537.2 -46227  
## - sqft\_living:grade:sqft\_above 1 2.3718 2537.7 -46223  
## - I(bedrooms^3) 1 2.6096 2537.9 -46221  
## - I(bedrooms^0.5) 1 2.6691 2538.0 -46220  
## - I(bathrooms^2) 1 2.8523 2538.1 -46219  
## - I(grade^0.5) 1 3.0860 2538.4 -46217  
## - I(sqft\_living^3) 1 3.2561 2538.6 -46215  
## - bathrooms:grade:sqft\_above 1 4.5127 2539.8 -46205  
## - I(bedrooms^2) 1 4.7496 2540.0 -46203  
## - I(sqft\_above^2) 1 6.0410 2541.3 -46192  
## - I(sqft\_living^2) 1 11.0111 2546.3 -46150  
##   
## Step: AIC=-46242.73  
## price ~ bedrooms + bathrooms + sqft\_living + grade + sqft\_above +   
## I(bedrooms^2) + I(bathrooms^2) + I(sqft\_living^2) + I(sqft\_above^2) +   
## I(bedrooms^3) + I(bathrooms^3) + I(sqft\_living^3) + I(grade^3) +   
## I(sqft\_above^3) + I(bedrooms^0.5) + I(bathrooms^0.5) + I(grade^0.5) +   
## bedrooms:bathrooms + bedrooms:sqft\_living + bathrooms:sqft\_living +   
## bedrooms:grade + bathrooms:grade + sqft\_living:grade + bedrooms:sqft\_above +   
## bathrooms:sqft\_above + sqft\_living:sqft\_above + grade:sqft\_above +   
## bedrooms:bathrooms:sqft\_living + bedrooms:sqft\_living:grade +   
## bedrooms:bathrooms:sqft\_above + bathrooms:sqft\_living:sqft\_above +   
## bedrooms:grade:sqft\_above + bathrooms:grade:sqft\_above +   
## sqft\_living:grade:sqft\_above  
##   
## Df Sum of Sq RSS AIC  
## - bedrooms:sqft\_living:grade 1 0.2093 2535.6 -46243  
## <none> 2535.3 -46243  
## - bedrooms:grade:sqft\_above 1 0.3720 2535.7 -46242  
## - bedrooms:bathrooms:sqft\_above 1 0.9559 2536.3 -46237  
## - bedrooms:bathrooms:sqft\_living 1 1.0387 2536.4 -46236  
## - I(grade^3) 1 1.0816 2536.4 -46236  
## - I(bathrooms^3) 1 1.2276 2536.6 -46234  
## - I(sqft\_above^3) 1 1.2454 2536.6 -46234  
## - I(bathrooms^0.5) 1 1.3381 2536.7 -46233  
## - bathrooms:sqft\_living:sqft\_above 1 2.0597 2537.4 -46227  
## - sqft\_living:grade:sqft\_above 1 2.3326 2537.7 -46225  
## - I(bedrooms^0.5) 1 2.7175 2538.1 -46222  
## - I(bedrooms^3) 1 2.7853 2538.1 -46221  
## - I(bathrooms^2) 1 2.7962 2538.1 -46221  
## - I(grade^0.5) 1 3.0299 2538.4 -46219  
## - I(sqft\_living^3) 1 3.3839 2538.7 -46216  
## - bathrooms:grade:sqft\_above 1 4.6595 2540.0 -46205  
## - I(bedrooms^2) 1 4.9998 2540.3 -46202  
## - I(sqft\_above^2) 1 6.1775 2541.5 -46192  
## - I(sqft\_living^2) 1 11.7812 2547.1 -46145  
##   
## Step: AIC=-46242.95  
## price ~ bedrooms + bathrooms + sqft\_living + grade + sqft\_above +   
## I(bedrooms^2) + I(bathrooms^2) + I(sqft\_living^2) + I(sqft\_above^2) +   
## I(bedrooms^3) + I(bathrooms^3) + I(sqft\_living^3) + I(grade^3) +   
## I(sqft\_above^3) + I(bedrooms^0.5) + I(bathrooms^0.5) + I(grade^0.5) +   
## bedrooms:bathrooms + bedrooms:sqft\_living + bathrooms:sqft\_living +   
## bedrooms:grade + bathrooms:grade + sqft\_living:grade + bedrooms:sqft\_above +   
## bathrooms:sqft\_above + sqft\_living:sqft\_above + grade:sqft\_above +   
## bedrooms:bathrooms:sqft\_living + bedrooms:bathrooms:sqft\_above +   
## bathrooms:sqft\_living:sqft\_above + bedrooms:grade:sqft\_above +   
## bathrooms:grade:sqft\_above + sqft\_living:grade:sqft\_above  
##   
## Df Sum of Sq RSS AIC  
## - bedrooms:grade:sqft\_above 1 0.1828 2535.7 -46243  
## <none> 2535.6 -46243  
## - bedrooms:bathrooms:sqft\_above 1 0.7676 2536.3 -46238  
## - bedrooms:bathrooms:sqft\_living 1 0.8347 2536.4 -46238  
## - I(grade^3) 1 1.0143 2536.6 -46236  
## - I(sqft\_above^3) 1 1.0926 2536.7 -46236  
## - I(bathrooms^3) 1 1.1957 2536.8 -46235  
## - I(bathrooms^0.5) 1 1.2625 2536.8 -46234  
## - sqft\_living:grade:sqft\_above 1 2.1440 2537.7 -46227  
## - bathrooms:sqft\_living:sqft\_above 1 2.1695 2537.7 -46226  
## - I(bathrooms^2) 1 2.7223 2538.3 -46222  
## - I(bedrooms^0.5) 1 2.7507 2538.3 -46222  
## - I(bedrooms^3) 1 2.8883 2538.4 -46220  
## - I(grade^0.5) 1 2.9032 2538.5 -46220  
## - I(sqft\_living^3) 1 3.4627 2539.0 -46215  
## - bathrooms:grade:sqft\_above 1 4.5839 2540.1 -46206  
## - I(bedrooms^2) 1 5.0657 2540.6 -46202  
## - I(sqft\_above^2) 1 5.9819 2541.5 -46194  
## - I(sqft\_living^2) 1 11.9013 2547.5 -46144  
##   
## Step: AIC=-46243.39  
## price ~ bedrooms + bathrooms + sqft\_living + grade + sqft\_above +   
## I(bedrooms^2) + I(bathrooms^2) + I(sqft\_living^2) + I(sqft\_above^2) +   
## I(bedrooms^3) + I(bathrooms^3) + I(sqft\_living^3) + I(grade^3) +   
## I(sqft\_above^3) + I(bedrooms^0.5) + I(bathrooms^0.5) + I(grade^0.5) +   
## bedrooms:bathrooms + bedrooms:sqft\_living + bathrooms:sqft\_living +   
## bedrooms:grade + bathrooms:grade + sqft\_living:grade + bedrooms:sqft\_above +   
## bathrooms:sqft\_above + sqft\_living:sqft\_above + grade:sqft\_above +   
## bedrooms:bathrooms:sqft\_living + bedrooms:bathrooms:sqft\_above +   
## bathrooms:sqft\_living:sqft\_above + bathrooms:grade:sqft\_above +   
## sqft\_living:grade:sqft\_above  
##   
## Df Sum of Sq RSS AIC  
## <none> 2535.7 -46243  
## - bedrooms:bathrooms:sqft\_above 1 0.6213 2536.4 -46240  
## - bedrooms:grade 1 0.6280 2536.4 -46240  
## - bedrooms:bathrooms:sqft\_living 1 0.9198 2536.7 -46238  
## - I(grade^3) 1 1.0168 2536.8 -46237  
## - I(bathrooms^3) 1 1.0375 2536.8 -46237  
## - I(sqft\_above^3) 1 1.1028 2536.8 -46236  
## - I(bathrooms^0.5) 1 1.2293 2537.0 -46235  
## - bathrooms:sqft\_living:sqft\_above 1 1.9929 2537.7 -46228  
## - sqft\_living:grade:sqft\_above 1 2.2561 2538.0 -46226  
## - I(bathrooms^2) 1 2.5595 2538.3 -46224  
## - I(bedrooms^0.5) 1 2.6484 2538.4 -46223  
## - I(bedrooms^3) 1 2.8359 2538.6 -46221  
## - I(grade^0.5) 1 2.9839 2538.7 -46220  
## - I(sqft\_living^3) 1 3.3160 2539.1 -46217  
## - bathrooms:grade:sqft\_above 1 4.5167 2540.3 -46207  
## - I(bedrooms^2) 1 5.0270 2540.8 -46203  
## - I(sqft\_above^2) 1 5.9609 2541.7 -46195  
## - I(sqft\_living^2) 1 11.7208 2547.5 -46146

summary(house.fit5)

##   
## Call:  
## lm(formula = price ~ bedrooms + bathrooms + sqft\_living + grade +   
## sqft\_above + I(bedrooms^2) + I(bathrooms^2) + I(sqft\_living^2) +   
## I(sqft\_above^2) + I(bedrooms^3) + I(bathrooms^3) + I(sqft\_living^3) +   
## I(grade^3) + I(sqft\_above^3) + I(bedrooms^0.5) + I(bathrooms^0.5) +   
## I(grade^0.5) + bedrooms:bathrooms + bedrooms:sqft\_living +   
## bathrooms:sqft\_living + bedrooms:grade + bathrooms:grade +   
## sqft\_living:grade + bedrooms:sqft\_above + bathrooms:sqft\_above +   
## sqft\_living:sqft\_above + grade:sqft\_above + bedrooms:bathrooms:sqft\_living +   
## bedrooms:bathrooms:sqft\_above + bathrooms:sqft\_living:sqft\_above +   
## bathrooms:grade:sqft\_above + sqft\_living:grade:sqft\_above,   
## data = house)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.27715 -0.23949 0.00509 0.22722 1.43122   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.486e+01 8.452e-01 17.584 < 2e-16 \*\*\*  
## bedrooms -8.647e-01 1.246e-01 -6.938 4.10e-12 \*\*\*  
## bathrooms -3.195e-01 2.218e-01 -1.441 0.14973   
## sqft\_living 2.677e-04 9.256e-05 2.892 0.00383 \*\*   
## grade 1.171e+00 1.949e-01 6.009 1.89e-09 \*\*\*  
## sqft\_above 2.628e-04 9.840e-05 2.671 0.00757 \*\*   
## I(bedrooms^2) 9.484e-02 1.450e-02 6.541 6.26e-11 \*\*\*  
## I(bathrooms^2) 1.597e-01 3.421e-02 4.667 3.07e-06 \*\*\*  
## I(sqft\_living^2) -1.972e-07 1.975e-08 -9.987 < 2e-16 \*\*\*  
## I(sqft\_above^2) -1.770e-07 2.485e-08 -7.122 1.09e-12 \*\*\*  
## I(bedrooms^3) -4.082e-03 8.309e-04 -4.913 9.06e-07 \*\*\*  
## I(bathrooms^3) -9.309e-03 3.133e-03 -2.971 0.00297 \*\*   
## I(sqft\_living^3) 6.878e-12 1.295e-12 5.312 1.09e-07 \*\*\*  
## I(grade^3) -6.566e-04 2.232e-04 -2.942 0.00327 \*\*   
## I(sqft\_above^3) 6.187e-12 2.020e-12 3.063 0.00219 \*\*   
## I(bedrooms^0.5) 1.019e+00 2.147e-01 4.747 2.07e-06 \*\*\*  
## I(bathrooms^0.5) 1.001e+00 3.095e-01 3.234 0.00122 \*\*   
## I(grade^0.5) -4.122e+00 8.180e-01 -5.039 4.72e-07 \*\*\*  
## bedrooms:bathrooms -7.234e-03 9.490e-03 -0.762 0.44589   
## bedrooms:sqft\_living -7.325e-06 1.720e-05 -0.426 0.67025   
## bathrooms:sqft\_living 1.361e-04 2.794e-05 4.871 1.12e-06 \*\*\*  
## bedrooms:grade 1.100e-02 4.760e-03 2.312 0.02080 \*   
## bathrooms:grade -8.246e-02 1.350e-02 -6.110 1.01e-09 \*\*\*  
## sqft\_living:grade 3.264e-05 1.208e-05 2.702 0.00690 \*\*   
## bedrooms:sqft\_above 1.851e-06 1.952e-05 0.095 0.92446   
## bathrooms:sqft\_above -3.248e-04 4.819e-05 -6.741 1.62e-11 \*\*\*  
## sqft\_living:sqft\_above 4.278e-07 4.682e-08 9.138 < 2e-16 \*\*\*  
## grade:sqft\_above -5.266e-05 1.187e-05 -4.436 9.20e-06 \*\*\*  
## bedrooms:bathrooms:sqft\_living -1.569e-05 5.607e-06 -2.798 0.00515 \*\*   
## bedrooms:bathrooms:sqft\_above 1.577e-05 6.859e-06 2.299 0.02150 \*   
## bathrooms:sqft\_living:sqft\_above -1.936e-08 4.701e-09 -4.118 3.83e-05 \*\*\*  
## bathrooms:grade:sqft\_above 3.112e-05 5.019e-06 6.200 5.76e-10 \*\*\*  
## sqft\_living:grade:sqft\_above -1.404e-08 3.205e-09 -4.382 1.18e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3428 on 21579 degrees of freedom  
## Multiple R-squared: 0.577, Adjusted R-squared: 0.5764   
## F-statistic: 919.9 on 32 and 21579 DF, p-value: < 2.2e-16

fit 6 : removing non - significant coefficients manually

house.fit6 <- update(house.fit5, ~ . - bedrooms:sqft\_above - bedrooms:sqft\_living - bedrooms:bathrooms)  
summary(house.fit6)

##   
## Call:  
## lm(formula = price ~ bedrooms + bathrooms + sqft\_living + grade +   
## sqft\_above + I(bedrooms^2) + I(bathrooms^2) + I(sqft\_living^2) +   
## I(sqft\_above^2) + I(bedrooms^3) + I(bathrooms^3) + I(sqft\_living^3) +   
## I(grade^3) + I(sqft\_above^3) + I(bedrooms^0.5) + I(bathrooms^0.5) +   
## I(grade^0.5) + bathrooms:sqft\_living + bedrooms:grade + bathrooms:grade +   
## sqft\_living:grade + bathrooms:sqft\_above + sqft\_living:sqft\_above +   
## grade:sqft\_above + bedrooms:bathrooms:sqft\_living + bedrooms:bathrooms:sqft\_above +   
## bathrooms:sqft\_living:sqft\_above + bathrooms:grade:sqft\_above +   
## sqft\_living:grade:sqft\_above, data = house)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.27917 -0.23935 0.00519 0.22747 1.43123   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.484e+01 8.434e-01 17.592 < 2e-16 \*\*\*  
## bedrooms -8.459e-01 1.202e-01 -7.037 2.02e-12 \*\*\*  
## bathrooms -3.563e-01 2.141e-01 -1.665 0.095988 .   
## sqft\_living 2.495e-04 7.701e-05 3.240 0.001199 \*\*   
## grade 1.167e+00 1.942e-01 6.007 1.92e-09 \*\*\*  
## sqft\_above 2.707e-04 7.115e-05 3.804 0.000143 \*\*\*  
## I(bedrooms^2) 9.066e-02 1.334e-02 6.794 1.12e-11 \*\*\*  
## I(bathrooms^2) 1.596e-01 3.347e-02 4.769 1.86e-06 \*\*\*  
## I(sqft\_living^2) -1.978e-07 1.873e-08 -10.558 < 2e-16 \*\*\*  
## I(sqft\_above^2) -1.752e-07 2.423e-08 -7.232 4.93e-13 \*\*\*  
## I(bedrooms^3) -3.870e-03 7.814e-04 -4.952 7.40e-07 \*\*\*  
## I(bathrooms^3) -9.183e-03 3.003e-03 -3.058 0.002232 \*\*   
## I(sqft\_living^3) 6.915e-12 1.230e-12 5.621 1.92e-08 \*\*\*  
## I(grade^3) -6.586e-04 2.227e-04 -2.957 0.003105 \*\*   
## I(sqft\_above^3) 5.895e-12 1.922e-12 3.067 0.002164 \*\*   
## I(bedrooms^0.5) 9.919e-01 2.105e-01 4.713 2.46e-06 \*\*\*  
## I(bathrooms^0.5) 1.028e+00 3.023e-01 3.401 0.000673 \*\*\*  
## I(grade^0.5) -4.093e+00 8.139e-01 -5.028 4.99e-07 \*\*\*  
## bathrooms:sqft\_living 1.404e-04 1.929e-05 7.280 3.46e-13 \*\*\*  
## bedrooms:grade 9.865e-03 4.200e-03 2.349 0.018845 \*   
## bathrooms:grade -8.105e-02 1.337e-02 -6.064 1.35e-09 \*\*\*  
## sqft\_living:grade 3.290e-05 1.195e-05 2.753 0.005917 \*\*   
## bathrooms:sqft\_above -3.287e-04 4.006e-05 -8.207 2.40e-16 \*\*\*  
## sqft\_living:sqft\_above 4.255e-07 4.670e-08 9.110 < 2e-16 \*\*\*  
## grade:sqft\_above -5.234e-05 1.180e-05 -4.436 9.20e-06 \*\*\*  
## bedrooms:bathrooms:sqft\_living -1.795e-05 2.416e-06 -7.432 1.11e-13 \*\*\*  
## bedrooms:bathrooms:sqft\_above 1.626e-05 2.796e-06 5.816 6.11e-09 \*\*\*  
## bathrooms:sqft\_living:sqft\_above -1.835e-08 4.489e-09 -4.088 4.37e-05 \*\*\*  
## bathrooms:grade:sqft\_above 3.092e-05 4.999e-06 6.186 6.27e-10 \*\*\*  
## sqft\_living:grade:sqft\_above -1.398e-08 3.203e-09 -4.365 1.28e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3428 on 21582 degrees of freedom  
## Multiple R-squared: 0.577, Adjusted R-squared: 0.5764   
## F-statistic: 1015 on 29 and 21582 DF, p-value: < 2.2e-16

fit 7 : step function

house.fit7 <- update(house.fit6, ~ . - bathrooms)  
summary(house.fit7)

##   
## Call:  
## lm(formula = price ~ bedrooms + sqft\_living + grade + sqft\_above +   
## I(bedrooms^2) + I(bathrooms^2) + I(sqft\_living^2) + I(sqft\_above^2) +   
## I(bedrooms^3) + I(bathrooms^3) + I(sqft\_living^3) + I(grade^3) +   
## I(sqft\_above^3) + I(bedrooms^0.5) + I(bathrooms^0.5) + I(grade^0.5) +   
## bathrooms:sqft\_living + bedrooms:grade + bathrooms:grade +   
## sqft\_living:grade + bathrooms:sqft\_above + sqft\_living:sqft\_above +   
## grade:sqft\_above + bedrooms:bathrooms:sqft\_living + bedrooms:bathrooms:sqft\_above +   
## bathrooms:sqft\_living:sqft\_above + bathrooms:grade:sqft\_above +   
## sqft\_living:grade:sqft\_above, data = house)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.28967 -0.23949 0.00514 0.22730 1.43077   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.475e+01 8.418e-01 17.522 < 2e-16 \*\*\*  
## bedrooms -9.301e-01 1.090e-01 -8.529 < 2e-16 \*\*\*  
## sqft\_living 2.074e-04 7.274e-05 2.851 0.004362 \*\*   
## grade 1.139e+00 1.935e-01 5.885 4.04e-09 \*\*\*  
## sqft\_above 2.875e-04 7.043e-05 4.082 4.48e-05 \*\*\*  
## I(bedrooms^2) 9.738e-02 1.272e-02 7.657 1.99e-14 \*\*\*  
## I(bathrooms^2) 1.150e-01 2.003e-02 5.740 9.61e-09 \*\*\*  
## I(sqft\_living^2) -2.056e-07 1.813e-08 -11.345 < 2e-16 \*\*\*  
## I(sqft\_above^2) -1.849e-07 2.351e-08 -7.868 3.78e-15 \*\*\*  
## I(bedrooms^3) -4.163e-03 7.613e-04 -5.468 4.59e-08 \*\*\*  
## I(bathrooms^3) -5.729e-03 2.171e-03 -2.639 0.008329 \*\*   
## I(sqft\_living^3) 7.541e-12 1.172e-12 6.436 1.25e-10 \*\*\*  
## I(grade^3) -6.110e-04 2.208e-04 -2.767 0.005669 \*\*   
## I(sqft\_above^3) 6.728e-12 1.856e-12 3.625 0.000289 \*\*\*  
## I(bedrooms^0.5) 1.154e+00 1.867e-01 6.178 6.59e-10 \*\*\*  
## I(bathrooms^0.5) 5.685e-01 1.232e-01 4.615 3.95e-06 \*\*\*  
## I(grade^0.5) -3.946e+00 8.092e-01 -4.877 1.09e-06 \*\*\*  
## sqft\_living:bathrooms 1.471e-04 1.887e-05 7.800 6.49e-15 \*\*\*  
## bedrooms:grade 1.057e-02 4.179e-03 2.529 0.011451 \*   
## grade:bathrooms -9.019e-02 1.218e-02 -7.402 1.39e-13 \*\*\*  
## sqft\_living:grade 3.779e-05 1.159e-05 3.261 0.001110 \*\*   
## sqft\_above:bathrooms -3.475e-04 3.845e-05 -9.037 < 2e-16 \*\*\*  
## sqft\_living:sqft\_above 4.571e-07 4.268e-08 10.710 < 2e-16 \*\*\*  
## grade:sqft\_above -5.600e-05 1.159e-05 -4.831 1.37e-06 \*\*\*  
## bedrooms:sqft\_living:bathrooms -1.793e-05 2.416e-06 -7.421 1.21e-13 \*\*\*  
## bedrooms:sqft\_above:bathrooms 1.598e-05 2.790e-06 5.725 1.05e-08 \*\*\*  
## sqft\_living:sqft\_above:bathrooms -2.140e-08 4.097e-09 -5.225 1.76e-07 \*\*\*  
## grade:sqft\_above:bathrooms 3.449e-05 4.516e-06 7.639 2.28e-14 \*\*\*  
## sqft\_living:grade:sqft\_above -1.603e-08 2.957e-09 -5.422 5.96e-08 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3428 on 21583 degrees of freedom  
## Multiple R-squared: 0.577, Adjusted R-squared: 0.5764   
## F-statistic: 1051 on 28 and 21583 DF, p-value: < 2.2e-16

#### 2) check the best fit by ignoring uncorrelated variables

from model 7 formula : lm(formula = price ~ bedrooms + sqft\_living + grade + sqft\_above + I(bedrooms^2) + I(bathrooms^2) + I(sqft\_living^2) + I(sqft\_above^2) + I(bedrooms^3) + I(bathrooms^3) + I(sqft\_living^3) + I(grade^3) + I(sqft\_above^3) + I(bedrooms^0.5) + I(bathrooms^0.5) + I(grade^0.5) + bathrooms:sqft\_living + bedrooms:grade + bathrooms:grade + sqft\_living:grade + bathrooms:sqft\_above + sqft\_living:sqft\_above + grade:sqft\_above + bedrooms:bathrooms:sqft\_living + bedrooms:bathrooms:sqft\_above + bathrooms:sqft\_living:sqft\_above + bathrooms:grade:sqft\_above + sqft\_living:grade:sqft\_above, data = house)

check the best fit

bstFits1 <- regsubsets(price ~ bedrooms + sqft\_living + grade + sqft\_above +   
 I(bedrooms^2) + I(bathrooms^2) + I(sqft\_living^2) + I(sqft\_above^2) +   
 I(bedrooms^3) + I(bathrooms^3) + I(sqft\_living^3) + I(grade^3) +   
 I(sqft\_above^3) + I(bedrooms^0.5) + I(bathrooms^0.5) + I(grade^0.5) +   
 bathrooms:sqft\_living + bedrooms:grade + bathrooms:grade +   
 sqft\_living:grade + bathrooms:sqft\_above + sqft\_living:sqft\_above +   
 grade:sqft\_above + bedrooms:bathrooms:sqft\_living + bedrooms:bathrooms:sqft\_above +   
 bathrooms:sqft\_living:sqft\_above + bathrooms:grade:sqft\_above +   
 sqft\_living:grade:sqft\_above, data = house, nbest = 1, nvmax = 4)  
par(mfrow = c(1,1))  
#subsets(bstFits1, statistic = "adjr2")  
plot(bstFits1, scale = "adjr2")

A close up of a logo

Description automatically generated

Final model -

house.final <- lm (price ~ sqft\_living + grade + sqft\_above, data = house)  
summary(house.final)

##   
## Call:  
## lm(formula = price ~ sqft\_living + grade + sqft\_above, data = house)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.76366 -0.24677 0.00374 0.22913 1.41674   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.108e+01 1.893e-02 585.46 <2e-16 \*\*\*  
## sqft\_living 3.022e-04 5.643e-06 53.55 <2e-16 \*\*\*  
## grade 2.049e-01 3.241e-03 63.22 <2e-16 \*\*\*  
## sqft\_above -1.309e-04 6.183e-06 -21.17 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3477 on 21608 degrees of freedom  
## Multiple R-squared: 0.5644, Adjusted R-squared: 0.5643   
## F-statistic: 9331 on 3 and 21608 DF, p-value: < 2.2e-16

par(mfrow = c(2,2))  
plot(house.final)

A close up of text on a white background

Description automatically generated

removing the influencial points

house <- house[- c(12778),]

same model again

house.final <- lm (price ~ sqft\_living + grade + sqft\_above, data = house)  
summary(house.final)

##   
## Call:  
## lm(formula = price ~ sqft\_living + grade + sqft\_above, data = house)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.34968 -0.24682 0.00352 0.22932 1.41620   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.109e+01 1.893e-02 585.66 <2e-16 \*\*\*  
## sqft\_living 3.044e-04 5.656e-06 53.82 <2e-16 \*\*\*  
## grade 2.040e-01 3.244e-03 62.87 <2e-16 \*\*\*  
## sqft\_above -1.311e-04 6.179e-06 -21.22 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3475 on 21607 degrees of freedom  
## Multiple R-squared: 0.5647, Adjusted R-squared: 0.5646   
## F-statistic: 9344 on 3 and 21607 DF, p-value: < 2.2e-16

par(mfrow = c(2,2))  
plot(house.final)

A close up of a map

Description automatically generated

checking the accuracy using k-Fold method

library(boot)  
house.glm <- glm(price ~ sqft\_living + grade + sqft\_above, data = house)  
summary(house.glm) # AIC = 15670

##   
## Call:  
## glm(formula = price ~ sqft\_living + grade + sqft\_above, data = house)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.34968 -0.24682 0.00352 0.22932 1.41620   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.109e+01 1.893e-02 585.66 <2e-16 \*\*\*  
## sqft\_living 3.044e-04 5.656e-06 53.82 <2e-16 \*\*\*  
## grade 2.040e-01 3.244e-03 62.87 <2e-16 \*\*\*  
## sqft\_above -1.311e-04 6.179e-06 -21.22 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.120723)  
##   
## Null deviance: 5992.5 on 21610 degrees of freedom  
## Residual deviance: 2608.5 on 21607 degrees of freedom  
## AIC: 15644  
##   
## Number of Fisher Scoring iterations: 2

house.glm.err<- cv.glm(data = house,glmfit = house.glm, K = 10)  
round(house.glm.err$delta[1],4)# 0.1209

## [1] 0.1208

## Part 2 - Decision Tree Regression Model

### Step 1 : Collecting Data

Same as of Part 1 : Step 1

### Step 2 : Exploring, preprocessing and cleaning the data

Primary setup

knitr::opts\_knit$set(root.dir = '/Users/sobil/Documents/MSC/Sem 1/Data Mining & Machine Learning/Project/Kc\_house/')  
remove(list = ls())  
set.seed(1)

adding alll the libraries required

library(RWeka)  
library(rpart)  
library(rpart.plot)  
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

Rest is same as of Part 1 : Step 2

source("Kc\_Import\_Explore\_Clean.R")

##   
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':  
##   
## %+%, alpha

## 'data.frame': 21613 obs. of 21 variables:  
## $ id : num 7.13e+09 6.41e+09 5.63e+09 2.49e+09 1.95e+09 ...  
## $ date : chr "20141013T000000" "20141209T000000" "20150225T000000" "20141209T000000" ...  
## $ price : num 221900 538000 180000 604000 510000 ...  
## $ bedrooms : int 3 3 2 4 3 4 3 3 3 3 ...  
## $ bathrooms : num 1 2.25 1 3 2 4.5 2.25 1.5 1 2.5 ...  
## $ sqft\_living : int 1180 2570 770 1960 1680 5420 1715 1060 1780 1890 ...  
## $ sqft\_lot : int 5650 7242 10000 5000 8080 101930 6819 9711 7470 6560 ...  
## $ floors : num 1 2 1 1 1 1 2 1 1 2 ...  
## $ waterfront : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ view : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ condition : int 3 3 3 5 3 3 3 3 3 3 ...  
## $ grade : int 7 7 6 7 8 11 7 7 7 7 ...  
## $ sqft\_above : int 1180 2170 770 1050 1680 3890 1715 1060 1050 1890 ...  
## $ sqft\_basement: int 0 400 0 910 0 1530 0 0 730 0 ...  
## $ yr\_built : int 1955 1951 1933 1965 1987 2001 1995 1963 1960 2003 ...  
## $ yr\_renovated : int 0 1991 0 0 0 0 0 0 0 0 ...  
## $ zipcode : int 98178 98125 98028 98136 98074 98053 98003 98198 98146 98038 ...  
## $ lat : num 47.5 47.7 47.7 47.5 47.6 ...  
## $ long : num -122 -122 -122 -122 -122 ...  
## $ sqft\_living15: int 1340 1690 2720 1360 1800 4760 2238 1650 1780 2390 ...  
## $ sqft\_lot15 : int 5650 7639 8062 5000 7503 101930 6819 9711 8113 7570 ...

### Step 3 - Data transformation & preparation

creating training and testing dataset from exisitng sample

indx <- createDataPartition(kc\_house$price, p = 0.8, list = FALSE)  
  
house\_train <- kc\_house[indx,]  
house\_test <- kc\_house[- indx,]

### Step 4 : Training a model

modeling a regression tree

house.rpart <- rpart(price ~ ., data = house\_train)

### Step 5 : Evaluating the model

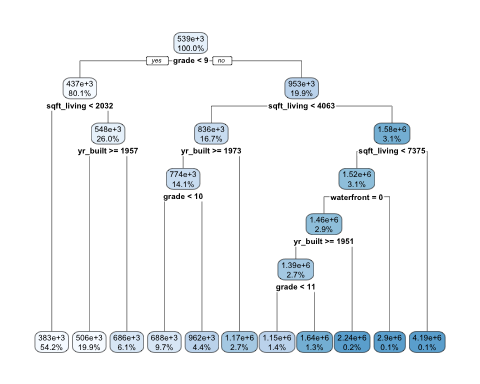
Summary of the model and description

#house.rpart  
summary(house.rpart)

## Call:  
## rpart(formula = price ~ ., data = house\_train)  
## n= 17291   
##   
## CP nsplit rel error xerror xstd  
## 1 0.32282388 0 1.0000000 1.0000754 0.04302218  
## 2 0.11069233 1 0.6771761 0.6772682 0.03386504  
## 3 0.03693204 2 0.5664838 0.5715869 0.02547574  
## 4 0.03632547 3 0.5295517 0.5720321 0.02594972  
## 5 0.02628165 4 0.4932263 0.5331213 0.02476824  
## 6 0.01922944 5 0.4669446 0.4885913 0.01712738  
## 7 0.01733461 6 0.4477152 0.4749996 0.01683980  
## 8 0.01179755 7 0.4303806 0.4546777 0.01584424  
## 9 0.01150364 9 0.4067855 0.4469484 0.01577295  
## 10 0.01000000 10 0.3952818 0.4319201 0.01521171  
##   
## Variable importance  
## grade sqft\_living sqft\_above bathrooms yr\_built   
## 33 26 20 9 4   
## sqft\_basement waterfront bedrooms view yr\_renovated   
## 4 2 1 1 1   
##   
## Node number 1: 17291 observations, complexity param=0.3228239  
## mean=539193.4, MSE=1.312817e+11   
## left son=2 (13855 obs) right son=3 (3436 obs)  
## Primary splits:  
## grade < 8.5 to the left, improve=0.3228239, (0 missing)  
## sqft\_living < 3087.5 to the left, improve=0.3101538, (0 missing)  
## sqft\_above < 2829 to the left, improve=0.2211162, (0 missing)  
## bathrooms < 3.125 to the left, improve=0.2104690, (0 missing)  
## view < 0.5 to the left, improve=0.1290508, (0 missing)  
## Surrogate splits:  
## sqft\_above < 2495.5 to the left, agree=0.885, adj=0.420, (0 split)  
## sqft\_living < 2915 to the left, agree=0.880, adj=0.394, (0 split)  
## bathrooms < 3.125 to the left, agree=0.838, adj=0.184, (0 split)  
## sqft\_basement < 1615 to the left, agree=0.807, adj=0.031, (0 split)  
## view < 2.5 to the left, agree=0.807, adj=0.027, (0 split)  
##   
## Node number 2: 13855 observations, complexity param=0.03632547  
## mean=436673.5, MSE=3.827863e+10   
## left son=4 (9365 obs) right son=5 (4490 obs)  
## Primary splits:  
## sqft\_living < 2032 to the left, improve=0.15547930, (0 missing)  
## grade < 7.5 to the left, improve=0.15026070, (0 missing)  
## sqft\_above < 1417 to the left, improve=0.08633045, (0 missing)  
## view < 0.5 to the left, improve=0.07755471, (0 missing)  
## bathrooms < 1.625 to the left, improve=0.06831389, (0 missing)  
## Surrogate splits:  
## sqft\_above < 2035 to the left, agree=0.845, adj=0.523, (0 split)  
## sqft\_basement < 725 to the left, agree=0.770, adj=0.292, (0 split)  
## bedrooms < 3.5 to the left, agree=0.769, adj=0.286, (0 split)  
## bathrooms < 2.375 to the left, agree=0.760, adj=0.259, (0 split)  
## grade < 7.5 to the left, agree=0.737, adj=0.188, (0 split)  
##   
## Node number 3: 3436 observations, complexity param=0.1106923  
## mean=952584.9, MSE=2.93025e+11   
## left son=6 (2896 obs) right son=7 (540 obs)  
## Primary splits:  
## sqft\_living < 4062.5 to the left, improve=0.2495652, (0 missing)  
## grade < 10.5 to the left, improve=0.2215809, (0 missing)  
## bathrooms < 3.625 to the left, improve=0.1625699, (0 missing)  
## sqft\_above < 4165 to the left, improve=0.1616369, (0 missing)  
## view < 0.5 to the left, improve=0.1142973, (0 missing)  
## Surrogate splits:  
## sqft\_above < 4062.5 to the left, agree=0.925, adj=0.524, (0 split)  
## grade < 10.5 to the left, agree=0.884, adj=0.265, (0 split)  
## bathrooms < 3.875 to the left, agree=0.883, adj=0.256, (0 split)  
## sqft\_basement < 1585 to the left, agree=0.862, adj=0.124, (0 split)  
## bedrooms < 5.5 to the left, agree=0.846, adj=0.020, (0 split)  
##   
## Node number 4: 9365 observations  
## mean=383255.9, MSE=2.356423e+10   
##   
## Node number 5: 4490 observations, complexity param=0.01150364  
## mean=548088.9, MSE=5.060422e+10   
## left son=10 (3440 obs) right son=11 (1050 obs)  
## Primary splits:  
## yr\_built < 1956.5 to the right, improve=0.11492820, (0 missing)  
## view < 0.5 to the left, improve=0.08605582, (0 missing)  
## waterfront splits as LR, improve=0.08538260, (0 missing)  
## sqft\_living < 2557.5 to the left, improve=0.06326877, (0 missing)  
## grade < 7.5 to the left, improve=0.05369639, (0 missing)  
## Surrogate splits:  
## yr\_renovated < 972.5 to the left, agree=0.797, adj=0.130, (0 split)  
## bathrooms < 1.625 to the right, agree=0.781, adj=0.064, (0 split)  
## condition < 4.5 to the left, agree=0.775, adj=0.037, (0 split)  
## grade < 6.5 to the right, agree=0.773, adj=0.030, (0 split)  
## sqft\_above < 1095 to the right, agree=0.768, adj=0.007, (0 split)  
##   
## Node number 6: 2896 observations, complexity param=0.02628165  
## mean=835812.1, MSE=1.265127e+11   
## left son=12 (2437 obs) right son=13 (459 obs)  
## Primary splits:  
## yr\_built < 1972.5 to the right, improve=0.16283370, (0 missing)  
## sqft\_living < 3155 to the left, improve=0.11520040, (0 missing)  
## grade < 9.5 to the left, improve=0.11088580, (0 missing)  
## view < 2.5 to the left, improve=0.09847294, (0 missing)  
## waterfront splits as LR, improve=0.09561904, (0 missing)  
## Surrogate splits:  
## yr\_renovated < 978 to the left, agree=0.877, adj=0.224, (0 split)  
## condition < 4.5 to the left, agree=0.864, adj=0.142, (0 split)  
## bedrooms < 5.5 to the left, agree=0.844, adj=0.015, (0 split)  
## bathrooms < 1.875 to the right, agree=0.844, adj=0.013, (0 split)  
## sqft\_basement < 1920 to the left, agree=0.843, adj=0.011, (0 split)  
##   
## Node number 7: 540 observations, complexity param=0.03693204  
## mean=1578833, MSE=7.207079e+11   
## left son=14 (528 obs) right son=15 (12 obs)  
## Primary splits:  
## sqft\_living < 7375 to the left, improve=0.2154143, (0 missing)  
## waterfront splits as LR, improve=0.1584211, (0 missing)  
## view < 3.5 to the left, improve=0.1430094, (0 missing)  
## sqft\_above < 6115 to the left, improve=0.1417749, (0 missing)  
## grade < 11.5 to the left, improve=0.1282572, (0 missing)  
## Surrogate splits:  
## sqft\_above < 6625 to the left, agree=0.989, adj=0.500, (0 split)  
## bathrooms < 6.125 to the left, agree=0.983, adj=0.250, (0 split)  
## sqft\_basement < 3815 to the left, agree=0.981, adj=0.167, (0 split)  
## grade < 12.5 to the left, agree=0.980, adj=0.083, (0 split)  
##   
## Node number 10: 3440 observations  
## mean=505955.9, MSE=3.61513e+10   
##   
## Node number 11: 1050 observations  
## mean=686124.5, MSE=7.308504e+10   
##   
## Node number 12: 2437 observations, complexity param=0.01733461  
## mean=773522.1, MSE=9.185468e+10   
## left son=24 (1675 obs) right son=25 (762 obs)  
## Primary splits:  
## grade < 9.5 to the left, improve=0.17578480, (0 missing)  
## sqft\_living < 3155 to the left, improve=0.14302750, (0 missing)  
## bathrooms < 3.125 to the left, improve=0.11465870, (0 missing)  
## view < 3.5 to the left, improve=0.09042453, (0 missing)  
## waterfront splits as LR, improve=0.08946337, (0 missing)  
## Surrogate splits:  
## sqft\_above < 3325 to the left, agree=0.734, adj=0.148, (0 split)  
## sqft\_living < 3425 to the left, agree=0.732, adj=0.144, (0 split)  
## bathrooms < 3.625 to the left, agree=0.699, adj=0.038, (0 split)  
## view < 3.5 to the left, agree=0.693, adj=0.018, (0 split)  
## waterfront splits as LR, agree=0.690, adj=0.009, (0 split)  
##   
## Node number 13: 459 observations  
## mean=1166532, MSE=1.805485e+11   
##   
## Node number 14: 528 observations, complexity param=0.01922944  
## mean=1519433, MSE=5.177857e+11   
## left son=28 (506 obs) right son=29 (22 obs)  
## Primary splits:  
## waterfront splits as LR, improve=0.1596640, (0 missing)  
## view < 3.5 to the left, improve=0.1358158, (0 missing)  
## grade < 10.5 to the left, improve=0.1061851, (0 missing)  
## yr\_built < 1944.5 to the right, improve=0.1057782, (0 missing)  
## sqft\_living < 5015 to the left, improve=0.0856600, (0 missing)  
##   
## Node number 15: 12 observations  
## mean=4192458, MSE=2.663e+12   
##   
## Node number 24: 1675 observations  
## mean=687816.1, MSE=4.943799e+10   
##   
## Node number 25: 762 observations  
## mean=961918, MSE=1.334538e+11   
##   
## Node number 28: 506 observations, complexity param=0.01179755  
## mean=1459479, MSE=4.272531e+11   
## left son=56 (467 obs) right son=57 (39 obs)  
## Primary splits:  
## yr\_built < 1950.5 to the right, improve=0.11954010, (0 missing)  
## grade < 10.5 to the left, improve=0.11449980, (0 missing)  
## sqft\_living < 5015 to the left, improve=0.07874945, (0 missing)  
## sqft\_lot < 35070.5 to the right, improve=0.05617333, (0 missing)  
## bathrooms < 4.625 to the left, improve=0.04976594, (0 missing)  
##   
## Node number 29: 22 observations  
## mean=2898364, MSE=6.159149e+11   
##   
## Node number 56: 467 observations, complexity param=0.01179755  
## mean=1394170, MSE=3.803135e+11   
## left son=112 (236 obs) right son=113 (231 obs)  
## Primary splits:  
## grade < 10.5 to the left, improve=0.15606020, (0 missing)  
## sqft\_living < 5015 to the left, improve=0.08001487, (0 missing)  
## bathrooms < 4.625 to the left, improve=0.05171274, (0 missing)  
## sqft\_above < 4227.5 to the left, improve=0.05108456, (0 missing)  
## view < 0.5 to the left, improve=0.04756573, (0 missing)  
## Surrogate splits:  
## sqft\_above < 4255 to the left, agree=0.664, adj=0.320, (0 split)  
## sqft\_living < 4655 to the left, agree=0.640, adj=0.273, (0 split)  
## sqft\_lot < 11499.5 to the left, agree=0.612, adj=0.216, (0 split)  
## yr\_built < 1987.5 to the left, agree=0.591, adj=0.173, (0 split)  
## sqft\_basement < 890 to the right, agree=0.587, adj=0.165, (0 split)  
##   
## Node number 57: 39 observations  
## mean=2241513, MSE=3.266734e+11   
##   
## Node number 112: 236 observations  
## mean=1153143, MSE=2.157654e+11   
##   
## Node number 113: 231 observations  
## mean=1640415, MSE=4.284349e+11

visulazing decision tree

rpart.plot(house.rpart, digits = 3)



Evaluating the model on testing data

house.predict.rpart <- predict(house.rpart, house\_test)  
summary(house.predict.rpart)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 383256 383256 383256 542083 686124 4192458

summary(house\_test$price)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 82500 322200 450000 543646 645000 7700000

checking correlation

cor(house.predict.rpart, house\_test$price)

## [1] 0.7713274

cheking performace with MAE creating a function mae to calculate Mean Absolute error

mae <- function(actual, pred) {  
 mean(abs(actual-pred))  
}

MAE with respect to predicted values

mae(house\_test$price, house.predict.rpart)

## [1] 158941.5

MAE with respect to mean values

mae(mean(house\_train$price), house.predict.rpart)

## [1] 187045.4

RMSE - Root Mean Squared Error - log values to compare to multi-linear model

RMSE(log(house\_test$price), log(house.predict.rpart))

## [1] 0.3780299

### Step 6 : Improving the model

modeling a model tree

house.m5p <- M5P(price ~ . , data = house\_train)  
#house.m5p  
summary(house.m5p)

##   
## === Summary ===  
##   
## Correlation coefficient 0.6249  
## Mean absolute error 8705723.0228  
## Root mean squared error 25833978.3661  
## Relative absolute error 3735.2195 %  
## Root relative squared error 7129.9947 %  
## Total Number of Instances 17291

house.predict.m5p <- predict(house.m5p, house\_test)  
summary(house.predict.m5p)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 241503 639213 798946 9295952 1286236 283228764

cor(house.predict.m5p, house\_test$price)

## [1] 0.6551532

mae(house\_test$price, house.predict.m5p)

## [1] 8756021

RMSE(log(house\_test$price), log(house.predict.m5p))

## [1] 1.794102

# Appendix 2 - House price prediction from 0.5 million record data set

### Step 1 : Collecting Data

The data set is collected in CSV format from kaggele and below is the reference for the same.

A.Sleem.(2018)Housepricing.[Online]. Available:<https://www.kaggle.com/greenwing1985/housepricing>

### Step 2 : Exploring, preprocessing and cleaning the data

Primary setup

knitr::opts\_knit$set(root.dir = '/Users/sobil/Documents/MSC/Sem 1/Data Mining & Machine Learning/Project/L5\_house/')  
remove(list = ls())  
set.seed(1)  
options(scipen=1)

loading alll the libraries required

library(data.table)  
library(boot)  
library(fpp2)

## Loading required package: ggplot2

## Loading required package: forecast

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

## Loading required package: fma

## Loading required package: expsmooth

library(car)

## Loading required package: carData

##   
## Attaching package: 'car'

## The following object is masked from 'package:boot':  
##   
## logit

library(leaps)

#### 1) reading the raw csv file

house <- fread("HousePrices\_HalfMil.csv")

#### 2) exploratory analysis

Structure of the house data frame

str(house)

## Classes 'data.table' and 'data.frame': 500000 obs. of 16 variables:  
## $ Area : int 164 84 190 75 148 124 58 249 243 242 ...  
## $ Garage : int 2 2 2 2 1 3 1 2 1 1 ...  
## $ FirePlace : int 0 0 4 4 4 3 0 1 0 2 ...  
## $ Baths : int 2 4 4 4 2 3 2 1 2 4 ...  
## $ White Marble : int 0 0 1 0 1 0 0 1 0 0 ...  
## $ Black Marble : int 1 0 0 0 0 1 0 0 0 0 ...  
## $ Indian Marble: int 0 1 0 1 0 0 1 0 1 1 ...  
## $ Floors : int 0 1 0 1 1 1 0 1 1 0 ...  
## $ City : int 3 2 2 1 2 1 3 1 1 2 ...  
## $ Solar : int 1 0 0 1 1 0 0 0 0 1 ...  
## $ Electric : int 1 0 0 1 0 0 1 1 0 0 ...  
## $ Fiber : int 1 0 1 1 0 1 1 0 0 0 ...  
## $ Glass Doors : int 1 1 0 1 1 1 1 1 0 0 ...  
## $ Swiming Pool : int 0 1 0 1 1 1 0 1 1 1 ...  
## $ Garden : int 0 1 0 1 1 1 1 0 0 0 ...  
## $ Prices : int 43800 37550 49500 50075 52400 54300 34400 50425 29575 22300 ...  
## - attr(\*, ".internal.selfref")=<externalptr>

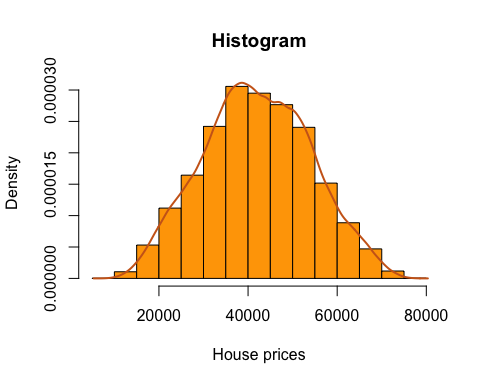
Summary of the house data frame

summary(house)

## Area Garage FirePlace Baths   
## Min. : 1.0 Min. :1.000 Min. :0.000 Min. :1.000   
## 1st Qu.: 63.0 1st Qu.:1.000 1st Qu.:1.000 1st Qu.:2.000   
## Median :125.0 Median :2.000 Median :2.000 Median :3.000   
## Mean :124.9 Mean :2.001 Mean :2.003 Mean :2.998   
## 3rd Qu.:187.0 3rd Qu.:3.000 3rd Qu.:3.000 3rd Qu.:4.000   
## Max. :249.0 Max. :3.000 Max. :4.000 Max. :5.000   
## White Marble Black Marble Indian Marble Floors   
## Min. :0.000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000   
## Median :0.000 Median :0.0000 Median :0.0000 Median :0.0000   
## Mean :0.333 Mean :0.3327 Mean :0.3343 Mean :0.4994   
## 3rd Qu.:1.000 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:1.0000   
## Max. :1.000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
## City Solar Electric Fiber   
## Min. :1.000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:1.000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000   
## Median :2.000 Median :0.0000 Median :1.0000 Median :1.0000   
## Mean :2.001 Mean :0.4987 Mean :0.5007 Mean :0.5005   
## 3rd Qu.:3.000 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:1.0000   
## Max. :3.000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
## Glass Doors Swiming Pool Garden Prices   
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. : 7725   
## 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:33500   
## Median :0.0000 Median :1.0000 Median :1.0000 Median :41850   
## Mean :0.4999 Mean :0.5004 Mean :0.5016 Mean :42050   
## 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:50750   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :77975

#### 3) check normal distribution of Prices

hist(house$Prices,  
 col = "orange",  
 border = "black",  
 prob = TRUE,  
 xlab = "House prices",  
 main = "Histogram")  
lines(density(house$Prices),  
 lwd = 2,  
 col = "chocolate3")



house.boxplot <- boxplot(house$Prices, main = "Box plot", ylab = "House prices")

A screenshot of a cell phone

Description automatically generated

#### 4) removing outliers to make more genarlised model

house.boxplot$out

## [1] 76975 77225 77000 77175 77375 77525 76825 77700 76950 77075 77250 77975  
## [13] 76750 77225 76775 76800

house <- subset(house, ! house$Prices %in% house.boxplot$out)  
house.boxplot <- boxplot(house$Prices, main = "Box plot", ylab = "House prices")

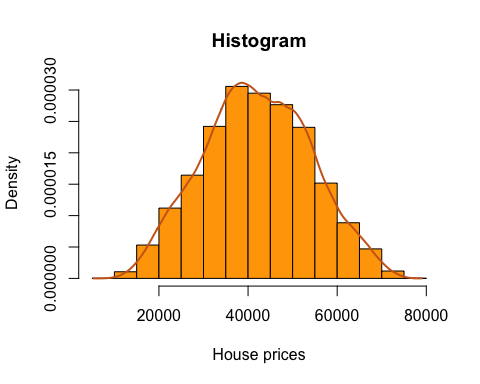
A screenshot of a cell phone

Description automatically generated

house.boxplot$out

## numeric(0)

hist(house$Prices,  
 col = "orange",  
 border = "black",  
 prob = TRUE,  
 xlab = "House prices",  
 main = "Histogram")  
lines(density(house$Prices),  
 lwd = 2,  
 col = "chocolate3")



house.boxplot$out

## numeric(0)

re-numbering the rows names

row.names(house) <- 1:499984

### Step 3 - Data transformation

cheking correlation

cor(house)

## Area Garage FirePlace Baths  
## Area 1.00000000000 -0.0009561749 3.171512e-04 -0.0004593863  
## Garage -0.00095617488 1.0000000000 1.260204e-03 -0.0036970899  
## FirePlace 0.00031715120 0.0012602041 1.000000e+00 0.0005954250  
## Baths -0.00045938627 -0.0036970899 5.954250e-04 1.0000000000  
## White Marble 0.00245686201 0.0004860091 8.982235e-04 0.0024346689  
## Black Marble -0.00144316109 0.0018742155 -8.954114e-04 -0.0027102665  
## Indian Marble -0.00101296442 -0.0023574470 -2.991059e-06 0.0002747674  
## Floors -0.00082493326 -0.0009699422 1.472555e-04 -0.0009210395  
## City -0.00351443467 0.0007451225 -2.573800e-04 -0.0009081267  
## Solar 0.00050591175 0.0014655403 -3.219511e-04 -0.0007673861  
## Electric -0.00016954633 0.0007447016 1.309616e-03 0.0010115692  
## Fiber 0.00006559031 -0.0006011681 1.780089e-03 -0.0007284917  
## Glass Doors -0.00127853023 -0.0022101528 -4.038464e-04 -0.0017088395  
## Swiming Pool 0.00061805058 0.0010929364 1.299237e-03 0.0022159658  
## Garden 0.00142202932 -0.0006733884 2.326840e-04 0.0017158806  
## Prices 0.14760173932 0.1001963924 8.904238e-02 0.1449912494  
## White Marble Black Marble Indian Marble Floors  
## Area 0.0024568620 -0.00144316109 -1.012964e-03 -0.00082493326  
## Garage 0.0004860091 0.00187421552 -2.357447e-03 -0.00096994219  
## FirePlace 0.0008982235 -0.00089541137 -2.991059e-06 0.00014725553  
## Baths 0.0024346689 -0.00271026646 2.747674e-04 -0.00092103948  
## White Marble 1.0000000000 -0.49888101866 -5.007113e-01 0.00003298180  
## Black Marble -0.4988810187 1.00000000000 -5.004070e-01 -0.00034552061  
## Indian Marble -0.5007113337 -0.50040700678 1.000000e+00 0.00031215124  
## Floors 0.0000329818 -0.00034552061 3.121512e-04 1.00000000000  
## City -0.0009160093 -0.00029667802 1.211395e-03 -0.00067982507  
## Solar -0.0016005727 0.00177222179 -1.711149e-04 -0.00266340286  
## Electric 0.0005187391 -0.00052889174 1.003497e-05 0.00005360457  
## Fiber -0.0006217479 0.00009601436 5.252191e-04 0.00134118361  
## Glass Doors -0.0004472523 0.00010855764 3.383727e-04 -0.00004036926  
## Swiming Pool -0.0018921204 0.00071005734 1.181007e-03 -0.00020690307  
## Garden 0.0009531764 0.00013600455 -1.088047e-03 -0.00049587593  
## Prices 0.4480976230 -0.07799497376 -3.697426e-01 0.61945121764  
## City Solar Electric Fiber  
## Area -0.0035144347 0.0005059118 -0.00016954633 0.00006559031  
## Garage 0.0007451225 0.0014655403 0.00074470160 -0.00060116814  
## FirePlace -0.0002573800 -0.0003219511 0.00130961628 0.00178008869  
## Baths -0.0009081267 -0.0007673861 0.00101156924 -0.00072849172  
## White Marble -0.0009160093 -0.0016005727 0.00051873907 -0.00062174794  
## Black Marble -0.0002966780 0.0017722218 -0.00052889174 0.00009601436  
## Indian Marble 0.0012113955 -0.0001711149 0.00001003497 0.00052521914  
## Floors -0.0006798251 -0.0026634029 0.00005360457 0.00134118361  
## City 1.0000000000 0.0004734082 0.00075332543 -0.00275531766  
## Solar 0.0004734082 1.0000000000 0.00187540582 0.00022638029  
## Electric 0.0007533254 0.0018754058 1.00000000000 -0.00033716112  
## Fiber -0.0027553177 0.0002263803 -0.00033716112 1.00000000000  
## Glass Doors 0.0007305603 -0.0008287956 0.00106040627 -0.00229981066  
## Swiming Pool 0.0003273864 -0.0004577177 0.00057090464 0.00413134355  
## Garden 0.0012019720 -0.0042715465 0.00077184703 -0.00002697347  
## Prices 0.2331825181 0.0083951719 0.05236927271 0.48460589251  
## Glass Doors Swiming Pool Garden Prices  
## Area -0.00127853023 0.0006180506 0.00142202932 0.147601739  
## Garage -0.00221015284 0.0010929364 -0.00067338843 0.100196392  
## FirePlace -0.00040384643 0.0012992366 0.00023268400 0.089042379  
## Baths -0.00170883950 0.0022159658 0.00171588060 0.144991249  
## White Marble -0.00044725227 -0.0018921204 0.00095317639 0.448097623  
## Black Marble 0.00010855764 0.0007100573 0.00013600455 -0.077994974  
## Indian Marble 0.00033837269 0.0011810067 -0.00108804715 -0.369742607  
## Floors -0.00004036926 -0.0002069031 -0.00049587593 0.619451218  
## City 0.00073056032 0.0003273864 0.00120197205 0.233182518  
## Solar -0.00082879564 -0.0004577177 -0.00427154650 0.008395172  
## Electric 0.00106040627 0.0005709046 0.00077184703 0.052369273  
## Fiber -0.00229981066 0.0041313435 -0.00002697347 0.484605893  
## Glass Doors 1.00000000000 0.0004002688 0.00332508465 0.181907887  
## Swiming Pool 0.00040026878 1.0000000000 -0.00019888790 0.001798530  
## Garden 0.00332508465 -0.0001988879 1.00000000000 0.001529416  
## Prices 0.18190788732 0.0017985297 0.00152941620 1.000000000

making a new df with only correlated columns

chouse <- house  
chouse <- chouse[,-c(2,3,6,10,11,14,15)]  
  
chouse.colnames <-colnames(chouse)  
chouse.colnames[3] <- "White\_Marbel"  
chouse.colnames[4] <- "Indian\_Marbel"  
colnames(chouse) <- chouse.colnames

checking correlation

cor(chouse)

## Area Baths White\_Marbel Indian\_Marbel  
## Area 1.00000000000 -0.0004593863 0.0024568620 -0.0010129644  
## Baths -0.00045938627 1.0000000000 0.0024346689 0.0002747674  
## White\_Marbel 0.00245686201 0.0024346689 1.0000000000 -0.5007113337  
## Indian\_Marbel -0.00101296442 0.0002747674 -0.5007113337 1.0000000000  
## Floors -0.00082493326 -0.0009210395 0.0000329818 0.0003121512  
## City -0.00351443467 -0.0009081267 -0.0009160093 0.0012113955  
## Fiber 0.00006559031 -0.0007284917 -0.0006217479 0.0005252191  
## Glass Doors -0.00127853023 -0.0017088395 -0.0004472523 0.0003383727  
## Prices 0.14760173932 0.1449912494 0.4480976230 -0.3697426073  
## Floors City Fiber Glass Doors  
## Area -0.00082493326 -0.0035144347 0.00006559031 -0.00127853023  
## Baths -0.00092103948 -0.0009081267 -0.00072849172 -0.00170883950  
## White\_Marbel 0.00003298180 -0.0009160093 -0.00062174794 -0.00044725227  
## Indian\_Marbel 0.00031215124 0.0012113955 0.00052521914 0.00033837269  
## Floors 1.00000000000 -0.0006798251 0.00134118361 -0.00004036926  
## City -0.00067982507 1.0000000000 -0.00275531766 0.00073056032  
## Fiber 0.00134118361 -0.0027553177 1.00000000000 -0.00229981066  
## Glass Doors -0.00004036926 0.0007305603 -0.00229981066 1.00000000000  
## Prices 0.61945121764 0.2331825181 0.48460589251 0.18190788732  
## Prices  
## Area 0.1476017  
## Baths 0.1449912  
## White\_Marbel 0.4480976  
## Indian\_Marbel -0.3697426  
## Floors 0.6194512  
## City 0.2331825  
## Fiber 0.4846059  
## Glass Doors 0.1819079  
## Prices 1.0000000

### Step 4 : Training a model on the data

model 1

house.fit1 <- glm(Prices ~ . + White\_Marbel:Indian\_Marbel , data = chouse)

### Step 5 : Evaluating the model

Cheking the summary of the model

summary(house.fit1)

##   
## Call:  
## glm(formula = Prices ~ . + White\_Marbel:Indian\_Marbel, data = chouse)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3769.7 -1254.9 -4.7 1245.7 3761.8   
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 11265.35304 10.91311 1032.3 <2e-16 \*\*\*  
## Area 24.98769 0.03434 727.6 <2e-16 \*\*\*  
## Baths 1247.61897 1.74349 715.6 <2e-16 \*\*\*  
## White\_Marbel 9000.64343 6.04412 1489.2 <2e-16 \*\*\*  
## Indian\_Marbel -5005.84224 6.03795 -829.1 <2e-16 \*\*\*  
## Floors 14997.32899 4.93130 3041.3 <2e-16 \*\*\*  
## City 3501.43505 3.02091 1159.1 <2e-16 \*\*\*  
## Fiber 11751.93391 4.93132 2383.1 <2e-16 \*\*\*  
## `Glass Doors` 4444.83567 4.93131 901.4 <2e-16 \*\*\*  
## White\_Marbel:Indian\_Marbel NA NA NA NA   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 3039597)  
##   
## Null deviance: 7.3309e+13 on 499983 degrees of freedom  
## Residual deviance: 1.5197e+12 on 499975 degrees of freedom  
## AIC: 8882283  
##   
## Number of Fisher Scoring iterations: 2

### Step 6 : Improving the model

#### 1) Adding all the possible relation to the model

model 2 adding the correlation independent variabels

house.fit2 <- update(house.fit1, ~ . - White\_Marbel:Indian\_Marbel)  
summary(house.fit2) #8882283

##   
## Call:  
## glm(formula = Prices ~ Area + Baths + White\_Marbel + Indian\_Marbel +   
## Floors + City + Fiber + `Glass Doors`, data = chouse)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3769.7 -1254.9 -4.7 1245.7 3761.8   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 11265.35304 10.91311 1032.3 <2e-16 \*\*\*  
## Area 24.98769 0.03434 727.6 <2e-16 \*\*\*  
## Baths 1247.61897 1.74349 715.6 <2e-16 \*\*\*  
## White\_Marbel 9000.64343 6.04412 1489.2 <2e-16 \*\*\*  
## Indian\_Marbel -5005.84224 6.03795 -829.1 <2e-16 \*\*\*  
## Floors 14997.32899 4.93130 3041.3 <2e-16 \*\*\*  
## City 3501.43505 3.02091 1159.1 <2e-16 \*\*\*  
## Fiber 11751.93391 4.93132 2383.1 <2e-16 \*\*\*  
## `Glass Doors` 4444.83567 4.93131 901.4 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 3039597)  
##   
## Null deviance: 7.3309e+13 on 499983 degrees of freedom  
## Residual deviance: 1.5197e+12 on 499975 degrees of freedom  
## AIC: 8882283  
##   
## Number of Fisher Scoring iterations: 2

accuracy(house.fit2)

## ME RMSE MAE MPE MAPE MASE  
## Training set 5.791451e-09 1743.428 1427.536 -0.2387077 3.763029 0.1442308

house.k10.fit2.err <- cv.glm(data = chouse,glmfit = house.fit2, K = 10)  
house.k10.fit2.err$delta# 3039673

## [1] 3039655 3039649

#checking r2  
house.fit2.lm <- lm(Prices ~ ., data = chouse)  
summary(house.fit2.lm) # 0.9793

##   
## Call:  
## lm(formula = Prices ~ ., data = chouse)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3769.7 -1254.9 -4.7 1245.7 3761.8   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 11265.35304 10.91311 1032.3 <2e-16 \*\*\*  
## Area 24.98769 0.03434 727.6 <2e-16 \*\*\*  
## Baths 1247.61897 1.74349 715.6 <2e-16 \*\*\*  
## White\_Marbel 9000.64343 6.04412 1489.2 <2e-16 \*\*\*  
## Indian\_Marbel -5005.84224 6.03795 -829.1 <2e-16 \*\*\*  
## Floors 14997.32899 4.93130 3041.3 <2e-16 \*\*\*  
## City 3501.43505 3.02091 1159.1 <2e-16 \*\*\*  
## Fiber 11751.93391 4.93132 2383.1 <2e-16 \*\*\*  
## `Glass Doors` 4444.83567 4.93131 901.4 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1743 on 499975 degrees of freedom  
## Multiple R-squared: 0.9793, Adjusted R-squared: 0.9793   
## F-statistic: 2.952e+06 on 8 and 499975 DF, p-value: < 2.2e-16

#### 2) check the best fit

house.fit2.bestFit1 <- regsubsets(Prices ~ Area + Baths + White\_Marbel + Indian\_Marbel +   
 Floors + City + Fiber + `Glass Doors`, data = chouse, nbest = 1, nvmax = 6)  
par(mfrow = c(1,1))  
#subsets(house.fit2.bestFit1, statistic = "adjr2", max.size = 6, min.size = 1)  
plot(house.fit2.bestFit1, scale = "adjr2")

A close up of a logo

Description automatically generated

4 predictors

house.fit4 <- glm(Prices ~ White\_Marbel + Floors + City + Fiber, data = chouse)  
summary(house.fit4) #9783831

##   
## Call:  
## glm(formula = Prices ~ White\_Marbel + Floors + City + Fiber,   
## data = chouse)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -14061.6 -2997.2 -4.2 3009.2 14020.8   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 17865.211 18.748 952.9 <2e-16 \*\*\*  
## White\_Marbel 11526.007 12.889 894.3 <2e-16 \*\*\*  
## Floors 14989.394 12.148 1233.9 <2e-16 \*\*\*  
## City 3491.541 7.442 469.2 <2e-16 \*\*\*  
## Fiber 11738.342 12.148 966.3 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 18446727)  
##   
## Null deviance: 7.3309e+13 on 499983 degrees of freedom  
## Residual deviance: 9.2230e+12 on 499979 degrees of freedom  
## AIC: 9783831  
##   
## Number of Fisher Scoring iterations: 2

house.k10.fit4.err <- cv.glm(data = chouse,glmfit = house.fit4, K = 10)  
house.k10.fit4.err$delta# 18446855

## [1] 18446975 18446952

#checking r2  
house.fit4.lm <- lm(Prices ~ White\_Marbel + Floors + City + Fiber, data = chouse)  
summary(house.fit4.lm) # 0.8742

##   
## Call:  
## lm(formula = Prices ~ White\_Marbel + Floors + City + Fiber, data = chouse)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -14061.6 -2997.2 -4.2 3009.2 14020.8   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 17865.211 18.748 952.9 <2e-16 \*\*\*  
## White\_Marbel 11526.007 12.889 894.3 <2e-16 \*\*\*  
## Floors 14989.394 12.148 1233.9 <2e-16 \*\*\*  
## City 3491.541 7.442 469.2 <2e-16 \*\*\*  
## Fiber 11738.342 12.148 966.3 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4295 on 499979 degrees of freedom  
## Multiple R-squared: 0.8742, Adjusted R-squared: 0.8742   
## F-statistic: 8.685e+05 on 4 and 499979 DF, p-value: < 2.2e-16

5 predictors

house.fit5 <- glm(Prices ~ White\_Marbel + Floors + City + Fiber + `Glass Doors`, data = chouse)  
summary(house.fit5) #9628934

##   
## Call:  
## glm(formula = Prices ~ White\_Marbel + Floors + City + Fiber +   
## `Glass Doors`, data = chouse)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -11849.7 -2573.4 7.6 2569.7 11835.7   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 15647.064 16.880 927.0 <2e-16 \*\*\*  
## White\_Marbel 11528.115 11.039 1044.3 <2e-16 \*\*\*  
## Floors 14989.557 10.405 1440.6 <2e-16 \*\*\*  
## City 3489.575 6.374 547.5 <2e-16 \*\*\*  
## Fiber 11748.531 10.405 1129.1 <2e-16 \*\*\*  
## `Glass Doors` 4433.692 10.405 426.1 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 13532329)  
##   
## Null deviance: 7.3309e+13 on 499983 degrees of freedom  
## Residual deviance: 6.7659e+12 on 499978 degrees of freedom  
## AIC: 9628934  
##   
## Number of Fisher Scoring iterations: 2

house.k10.fit5.err <- cv.glm(data = chouse,glmfit = house.fit5, K = 10)  
house.k10.fit5.err$delta# 13532564

## [1] 13532487 13532470

#checking r2  
house.fit5.lm <- lm(Prices ~ White\_Marbel + Floors + City + Fiber + `Glass Doors`, data = chouse)  
summary(house.fit5.lm) # 0.9077

##   
## Call:  
## lm(formula = Prices ~ White\_Marbel + Floors + City + Fiber +   
## `Glass Doors`, data = chouse)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -11849.7 -2573.4 7.6 2569.7 11835.7   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 15647.064 16.880 927.0 <2e-16 \*\*\*  
## White\_Marbel 11528.115 11.039 1044.3 <2e-16 \*\*\*  
## Floors 14989.557 10.405 1440.6 <2e-16 \*\*\*  
## City 3489.575 6.374 547.5 <2e-16 \*\*\*  
## Fiber 11748.531 10.405 1129.1 <2e-16 \*\*\*  
## `Glass Doors` 4433.692 10.405 426.1 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3679 on 499978 degrees of freedom  
## Multiple R-squared: 0.9077, Adjusted R-squared: 0.9077   
## F-statistic: 9.835e+05 on 5 and 499978 DF, p-value: < 2.2e-16

6 predictors

house.fit6 <- glm(Prices ~ White\_Marbel + Indian\_Marbel + Floors + City + Fiber + `Glass Doors`, data = chouse)  
summary(house.fit6) #9445048

##   
## Call:  
## glm(formula = Prices ~ White\_Marbel + Indian\_Marbel + Floors +   
## City + Fiber + `Glass Doors`, data = chouse)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -9370.3 -2152.8 0.8 2147.3 9351.6   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 18145.883 15.012 1208.8 <2e-16 \*\*\*  
## White\_Marbel 9023.390 10.611 850.4 <2e-16 \*\*\*  
## Indian\_Marbel -4997.261 10.600 -471.4 <2e-16 \*\*\*  
## Floors 14991.108 8.657 1731.6 <2e-16 \*\*\*  
## City 3491.752 5.303 658.4 <2e-16 \*\*\*  
## Fiber 11749.548 8.657 1357.2 <2e-16 \*\*\*  
## `Glass Doors` 4434.231 8.657 512.2 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 9367959)  
##   
## Null deviance: 7.3309e+13 on 499983 degrees of freedom  
## Residual deviance: 4.6838e+12 on 499977 degrees of freedom  
## AIC: 9445048  
##   
## Number of Fisher Scoring iterations: 2

house.k10.fit6.err <- cv.glm(data = chouse,glmfit = house.fit6, K = 10)  
house.k10.fit6.err$delta# 9368131

## [1] 9368083 9368069

#checking r2  
house.fit6.lm <- lm(Prices ~ White\_Marbel + Indian\_Marbel + Floors + City + Fiber + `Glass Doors`, data = chouse)  
summary(house.fit6.lm) # 0.9361

##   
## Call:  
## lm(formula = Prices ~ White\_Marbel + Indian\_Marbel + Floors +   
## City + Fiber + `Glass Doors`, data = chouse)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9370.3 -2152.8 0.8 2147.3 9351.6   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 18145.883 15.012 1208.8 <2e-16 \*\*\*  
## White\_Marbel 9023.390 10.611 850.4 <2e-16 \*\*\*  
## Indian\_Marbel -4997.261 10.600 -471.4 <2e-16 \*\*\*  
## Floors 14991.108 8.657 1731.6 <2e-16 \*\*\*  
## City 3491.752 5.303 658.4 <2e-16 \*\*\*  
## Fiber 11749.548 8.657 1357.2 <2e-16 \*\*\*  
## `Glass Doors` 4434.231 8.657 512.2 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3061 on 499977 degrees of freedom  
## Multiple R-squared: 0.9361, Adjusted R-squared: 0.9361   
## F-statistic: 1.221e+06 on 6 and 499977 DF, p-value: < 2.2e-16

#### 3) Best model performance is house.fit2

par(mfrow = c(2,2))  
summary(house.fit2)

##   
## Call:  
## glm(formula = Prices ~ Area + Baths + White\_Marbel + Indian\_Marbel +   
## Floors + City + Fiber + `Glass Doors`, data = chouse)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3769.7 -1254.9 -4.7 1245.7 3761.8   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 11265.35304 10.91311 1032.3 <2e-16 \*\*\*  
## Area 24.98769 0.03434 727.6 <2e-16 \*\*\*  
## Baths 1247.61897 1.74349 715.6 <2e-16 \*\*\*  
## White\_Marbel 9000.64343 6.04412 1489.2 <2e-16 \*\*\*  
## Indian\_Marbel -5005.84224 6.03795 -829.1 <2e-16 \*\*\*  
## Floors 14997.32899 4.93130 3041.3 <2e-16 \*\*\*  
## City 3501.43505 3.02091 1159.1 <2e-16 \*\*\*  
## Fiber 11751.93391 4.93132 2383.1 <2e-16 \*\*\*  
## `Glass Doors` 4444.83567 4.93131 901.4 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 3039597)  
##   
## Null deviance: 7.3309e+13 on 499983 degrees of freedom  
## Residual deviance: 1.5197e+12 on 499975 degrees of freedom  
## AIC: 8882283  
##   
## Number of Fisher Scoring iterations: 2

summary(house.fit2.lm)

##   
## Call:  
## lm(formula = Prices ~ ., data = chouse)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3769.7 -1254.9 -4.7 1245.7 3761.8   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 11265.35304 10.91311 1032.3 <2e-16 \*\*\*  
## Area 24.98769 0.03434 727.6 <2e-16 \*\*\*  
## Baths 1247.61897 1.74349 715.6 <2e-16 \*\*\*  
## White\_Marbel 9000.64343 6.04412 1489.2 <2e-16 \*\*\*  
## Indian\_Marbel -5005.84224 6.03795 -829.1 <2e-16 \*\*\*  
## Floors 14997.32899 4.93130 3041.3 <2e-16 \*\*\*  
## City 3501.43505 3.02091 1159.1 <2e-16 \*\*\*  
## Fiber 11751.93391 4.93132 2383.1 <2e-16 \*\*\*  
## `Glass Doors` 4444.83567 4.93131 901.4 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1743 on 499975 degrees of freedom  
## Multiple R-squared: 0.9793, Adjusted R-squared: 0.9793   
## F-statistic: 2.952e+06 on 8 and 499975 DF, p-value: < 2.2e-16

plot(house.fit2)

A screenshot of a cell phone

Description automatically generated

# Appendix 3 - Customer’s decision to open a term deposit account

## Part 1 - KNN Model

### Step 1 : Collecting Data

The data set is collected in CSV format from UCI.edu and below is the reference for the same.

S. Moro, P. Cortez, and P. Rita. (2014) A data-driven approach to predictthe success of bank telemarketing. Decision Support Systems,. [Online].Available: <https://archive.ics.uci.edu/ml/datasets/bank> marketing

### Step 2 : Exploring, preprocessing and cleaning the data

Primary setup

knitr::opts\_knit$set(root.dir = '/Users/sobil/Documents/MSC/Sem 1/Data Mining & Machine Learning/Project/Bank/')  
remove(list = ls())  
set.seed(1)

loading all the libraries required

library(gmodels)  
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(class)  
library(fastDummies)

#### 1) reading the raw csv file

bank <- read.csv("bank-additional-full.csv", sep = ";")

#### 2) exploratory analysis

Structure of the bank data frame

str(bank)

## 'data.frame': 41188 obs. of 21 variables:  
## $ age : int 56 57 37 40 56 45 59 41 24 25 ...  
## $ job : Factor w/ 12 levels "admin.","blue-collar",..: 4 8 8 1 8 8 1 2 10 8 ...  
## $ marital : Factor w/ 4 levels "divorced","married",..: 2 2 2 2 2 2 2 2 3 3 ...  
## $ education : Factor w/ 8 levels "basic.4y","basic.6y",..: 1 4 4 2 4 3 6 8 6 4 ...  
## $ default : Factor w/ 3 levels "no","unknown",..: 1 2 1 1 1 2 1 2 1 1 ...  
## $ housing : Factor w/ 3 levels "no","unknown",..: 1 1 3 1 1 1 1 1 3 3 ...  
## $ loan : Factor w/ 3 levels "no","unknown",..: 1 1 1 1 3 1 1 1 1 1 ...  
## $ contact : Factor w/ 2 levels "cellular","telephone": 2 2 2 2 2 2 2 2 2 2 ...  
## $ month : Factor w/ 10 levels "apr","aug","dec",..: 7 7 7 7 7 7 7 7 7 7 ...  
## $ day\_of\_week : Factor w/ 5 levels "fri","mon","thu",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ duration : int 261 149 226 151 307 198 139 217 380 50 ...  
## $ campaign : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ pdays : int 999 999 999 999 999 999 999 999 999 999 ...  
## $ previous : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ poutcome : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ emp.var.rate : num 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 ...  
## $ cons.price.idx: num 94 94 94 94 94 ...  
## $ cons.conf.idx : num -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 ...  
## $ euribor3m : num 4.86 4.86 4.86 4.86 4.86 ...  
## $ nr.employed : num 5191 5191 5191 5191 5191 ...  
## $ y : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...

Summary of the bank data frame

summary(bank)

## age job marital   
## Min. :17.00 admin. :10422 divorced: 4612   
## 1st Qu.:32.00 blue-collar: 9254 married :24928   
## Median :38.00 technician : 6743 single :11568   
## Mean :40.02 services : 3969 unknown : 80   
## 3rd Qu.:47.00 management : 2924   
## Max. :98.00 retired : 1720   
## (Other) : 6156   
## education default housing loan   
## university.degree :12168 no :32588 no :18622 no :33950   
## high.school : 9515 unknown: 8597 unknown: 990 unknown: 990   
## basic.9y : 6045 yes : 3 yes :21576 yes : 6248   
## professional.course: 5243   
## basic.4y : 4176   
## basic.6y : 2292   
## (Other) : 1749   
## contact month day\_of\_week duration   
## cellular :26144 may :13769 fri:7827 Min. : 0.0   
## telephone:15044 jul : 7174 mon:8514 1st Qu.: 102.0   
## aug : 6178 thu:8623 Median : 180.0   
## jun : 5318 tue:8090 Mean : 258.3   
## nov : 4101 wed:8134 3rd Qu.: 319.0   
## apr : 2632 Max. :4918.0   
## (Other): 2016   
## campaign pdays previous poutcome   
## Min. : 1.000 Min. : 0.0 Min. :0.000 failure : 4252   
## 1st Qu.: 1.000 1st Qu.:999.0 1st Qu.:0.000 nonexistent:35563   
## Median : 2.000 Median :999.0 Median :0.000 success : 1373   
## Mean : 2.568 Mean :962.5 Mean :0.173   
## 3rd Qu.: 3.000 3rd Qu.:999.0 3rd Qu.:0.000   
## Max. :56.000 Max. :999.0 Max. :7.000   
##   
## emp.var.rate cons.price.idx cons.conf.idx euribor3m   
## Min. :-3.40000 Min. :92.20 Min. :-50.8 Min. :0.634   
## 1st Qu.:-1.80000 1st Qu.:93.08 1st Qu.:-42.7 1st Qu.:1.344   
## Median : 1.10000 Median :93.75 Median :-41.8 Median :4.857   
## Mean : 0.08189 Mean :93.58 Mean :-40.5 Mean :3.621   
## 3rd Qu.: 1.40000 3rd Qu.:93.99 3rd Qu.:-36.4 3rd Qu.:4.961   
## Max. : 1.40000 Max. :94.77 Max. :-26.9 Max. :5.045   
##   
## nr.employed y   
## Min. :4964 no :36548   
## 1st Qu.:5099 yes: 4640   
## Median :5191   
## Mean :5167   
## 3rd Qu.:5228   
## Max. :5228   
##

checking the overall distribution of prediction/ target charcterstics/feature

gmodels::CrossTable(bank$y)

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 41188   
##   
##   
## | no | yes |   
## |-----------|-----------|  
## | 36548 | 4640 |   
## | 0.887 | 0.113 |   
## |-----------|-----------|  
##   
##   
##   
##

converting the target feature into factor with adding labels for more information

bank$y <- factor(bank$y, levels = c("yes", "no"), labels = c("Yes", "No"))

cheking NAs

apply(X = bank,MARGIN = 2, FUN = function(col) any(is.na(col))) # no NAs

## age job marital education default   
## FALSE FALSE FALSE FALSE FALSE   
## housing loan contact month day\_of\_week   
## FALSE FALSE FALSE FALSE FALSE   
## duration campaign pdays previous poutcome   
## FALSE FALSE FALSE FALSE FALSE   
## emp.var.rate cons.price.idx cons.conf.idx euribor3m nr.employed   
## FALSE FALSE FALSE FALSE FALSE   
## y   
## FALSE

### Step 3.1 - Data transformation & preparation - Normalizing the data with custom function

normalizing the data to standardize range of values of various characterstics #### 1) creating normalized method

nor <- function(val) {  
 return((val - min(val))/(max(val) - min(val)))  
}

getting normalised variables in a data frame

bank\_n\_cont <- as.data.frame(lapply(bank[c(1,11,12,13,14,16,17,18,19,20)], nor))  
summary(bank\_n\_cont)

## age duration campaign pdays   
## Min. :0.0000 Min. :0.00000 Min. :0.00000 Min. :0.0000   
## 1st Qu.:0.1852 1st Qu.:0.02074 1st Qu.:0.00000 1st Qu.:1.0000   
## Median :0.2593 Median :0.03660 Median :0.01818 Median :1.0000   
## Mean :0.2842 Mean :0.05252 Mean :0.02850 Mean :0.9634   
## 3rd Qu.:0.3704 3rd Qu.:0.06486 3rd Qu.:0.03636 3rd Qu.:1.0000   
## Max. :1.0000 Max. :1.00000 Max. :1.00000 Max. :1.0000   
## previous emp.var.rate cons.price.idx cons.conf.idx   
## Min. :0.00000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.00000 1st Qu.:0.3333 1st Qu.:0.3406 1st Qu.:0.3389   
## Median :0.00000 Median :0.9375 Median :0.6033 Median :0.3766   
## Mean :0.02471 Mean :0.7254 Mean :0.5357 Mean :0.4309   
## 3rd Qu.:0.00000 3rd Qu.:1.0000 3rd Qu.:0.6988 3rd Qu.:0.6025   
## Max. :1.00000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
## euribor3m nr.employed   
## Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.1610 1st Qu.:0.5123   
## Median :0.9574 Median :0.8597   
## Mean :0.6772 Mean :0.7691   
## 3rd Qu.:0.9810 3rd Qu.:1.0000   
## Max. :1.0000 Max. :1.0000

#### 2) creating a new df of only independent variables with normalised columns

bank\_n <- bank[,-21]  
bank\_n$age <- bank\_n\_cont$age  
bank\_n$duration <- bank\_n\_cont$duration  
bank\_n$campaign <- bank\_n\_cont$campaign  
bank\_n$pdays <- bank\_n\_cont$pdays  
bank\_n$previous <- bank\_n\_cont$previous  
bank\_n$emp.var.rate <- bank\_n\_cont$emp.var.rate  
bank\_n$cons.price.idx <- bank\_n\_cont$cons.price.idx  
bank\_n$cons.conf.idx <- bank\_n\_cont$cons.conf.idx  
bank\_n$euribor3m <- bank\_n\_cont$euribor3m  
bank\_n$nr.employed <- bank\_n\_cont$nr.employed  
summary(bank\_n)

## age job marital   
## Min. :0.0000 admin. :10422 divorced: 4612   
## 1st Qu.:0.1852 blue-collar: 9254 married :24928   
## Median :0.2593 technician : 6743 single :11568   
## Mean :0.2842 services : 3969 unknown : 80   
## 3rd Qu.:0.3704 management : 2924   
## Max. :1.0000 retired : 1720   
## (Other) : 6156   
## education default housing loan   
## university.degree :12168 no :32588 no :18622 no :33950   
## high.school : 9515 unknown: 8597 unknown: 990 unknown: 990   
## basic.9y : 6045 yes : 3 yes :21576 yes : 6248   
## professional.course: 5243   
## basic.4y : 4176   
## basic.6y : 2292   
## (Other) : 1749   
## contact month day\_of\_week duration   
## cellular :26144 may :13769 fri:7827 Min. :0.00000   
## telephone:15044 jul : 7174 mon:8514 1st Qu.:0.02074   
## aug : 6178 thu:8623 Median :0.03660   
## jun : 5318 tue:8090 Mean :0.05252   
## nov : 4101 wed:8134 3rd Qu.:0.06486   
## apr : 2632 Max. :1.00000   
## (Other): 2016   
## campaign pdays previous poutcome   
## Min. :0.00000 Min. :0.0000 Min. :0.00000 failure : 4252   
## 1st Qu.:0.00000 1st Qu.:1.0000 1st Qu.:0.00000 nonexistent:35563   
## Median :0.01818 Median :1.0000 Median :0.00000 success : 1373   
## Mean :0.02850 Mean :0.9634 Mean :0.02471   
## 3rd Qu.:0.03636 3rd Qu.:1.0000 3rd Qu.:0.00000   
## Max. :1.00000 Max. :1.0000 Max. :1.00000   
##   
## emp.var.rate cons.price.idx cons.conf.idx euribor3m   
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.3333 1st Qu.:0.3406 1st Qu.:0.3389 1st Qu.:0.1610   
## Median :0.9375 Median :0.6033 Median :0.3766 Median :0.9574   
## Mean :0.7254 Mean :0.5357 Mean :0.4309 Mean :0.6772   
## 3rd Qu.:1.0000 3rd Qu.:0.6988 3rd Qu.:0.6025 3rd Qu.:0.9810   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
##   
## nr.employed   
## Min. :0.0000   
## 1st Qu.:0.5123   
## Median :0.8597   
## Mean :0.7691   
## 3rd Qu.:1.0000   
## Max. :1.0000   
##

#### 3) creating dummy columns for categorical columns

bank\_n <- fastDummies::dummy\_cols(bank\_n, remove\_first\_dummy = TRUE) # remove\_first\_dummy = TRUE to avoid multi-collinearity

removing categorical columns from df (dummy are included)

bank\_n <- bank\_n[,-c(2:10,15)]

#### 4) creating training and testing dataset from exisitng sample

indx <- createDataPartition(bank$y, p = 0.8, list = FALSE)  
bank\_train <- bank\_n[indx,]  
bank\_test <- bank\_n[- indx,]

creating lables for test and training data sets

bank\_train\_labels <- bank[indx,21]  
bank\_test\_labels <- bank[- indx,21]

### Step 4.1 : Training a model on the data - Normalizing the data with custom function

using the knn method of class package with K value equalent to square root of total train observation & odd number to eliminate tie vote issue for 2 factor classification i.e.

bank$y %>% length %>% sqrt %>% round # 203

## [1] 203

bank\_test\_predict <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = 203)

### Step 5.1 : Evaluating the model - Normalizing the data with custom function

“Model 1 : k = 203”

confusionMatrix(data = bank\_test\_predict,reference = bank\_test\_labels)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 143 35  
## No 785 7274  
##   
## Accuracy : 0.9004   
## 95% CI : (0.8938, 0.9068)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 7.001e-05   
##   
## Kappa : 0.2307   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.15409   
## Specificity : 0.99521   
## Pos Pred Value : 0.80337   
## Neg Pred Value : 0.90259   
## Prevalence : 0.11266   
## Detection Rate : 0.01736   
## Detection Prevalence : 0.02161   
## Balanced Accuracy : 0.57465   
##   
## 'Positive' Class : Yes   
##

### Step 6.1 : Improving the model

#### 1) Changing k values

bank\_test\_predict2 <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = 151)  
print("Model 2 : k = 151")

## [1] "Model 2 : k = 151"

print(confusionMatrix(data = bank\_test\_predict2,reference = bank\_test\_labels))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 144 43  
## No 784 7266  
##   
## Accuracy : 0.8996   
## 95% CI : (0.8929, 0.906)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.0001886   
##   
## Kappa : 0.2292   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.15517   
## Specificity : 0.99412   
## Pos Pred Value : 0.77005   
## Neg Pred Value : 0.90261   
## Prevalence : 0.11266   
## Detection Rate : 0.01748   
## Detection Prevalence : 0.02270   
## Balanced Accuracy : 0.57464   
##   
## 'Positive' Class : Yes   
##

predict = 3 : k = 101

bank\_test\_predict3 <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = 101)  
print("Model 3 : k = 101")

## [1] "Model 3 : k = 101"

print(confusionMatrix(data = bank\_test\_predict3,reference = bank\_test\_labels))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 164 45  
## No 764 7264  
##   
## Accuracy : 0.9018   
## 95% CI : (0.8952, 0.9081)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 1.298e-05   
##   
## Kappa : 0.2577   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.17672   
## Specificity : 0.99384   
## Pos Pred Value : 0.78469   
## Neg Pred Value : 0.90483   
## Prevalence : 0.11266   
## Detection Rate : 0.01991   
## Detection Prevalence : 0.02537   
## Balanced Accuracy : 0.58528   
##   
## 'Positive' Class : Yes   
##

bank\_test\_predict4 <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = 75)  
print("Model 4 : k = 75")

## [1] "Model 4 : k = 75"

print(confusionMatrix(data = bank\_test\_predict4,reference = bank\_test\_labels))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 164 47  
## No 764 7262  
##   
## Accuracy : 0.9015   
## 95% CI : (0.8949, 0.9079)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 1.785e-05   
##   
## Kappa : 0.257   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.17672   
## Specificity : 0.99357   
## Pos Pred Value : 0.77725   
## Neg Pred Value : 0.90481   
## Prevalence : 0.11266   
## Detection Rate : 0.01991   
## Detection Prevalence : 0.02562   
## Balanced Accuracy : 0.58515   
##   
## 'Positive' Class : Yes   
##

bank\_test\_predict5 <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = 51)  
print("Model 5 : k = 51")

## [1] "Model 5 : k = 51"

print(confusionMatrix(data = bank\_test\_predict5,reference = bank\_test\_labels))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 173 52  
## No 755 7257  
##   
## Accuracy : 0.902   
## 95% CI : (0.8954, 0.9084)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 9.396e-06   
##   
## Kappa : 0.2679   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.18642   
## Specificity : 0.99289   
## Pos Pred Value : 0.76889   
## Neg Pred Value : 0.90577   
## Prevalence : 0.11266   
## Detection Rate : 0.02100   
## Detection Prevalence : 0.02732   
## Balanced Accuracy : 0.58965   
##   
## 'Positive' Class : Yes   
##

bank\_test\_predict6 <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = 31)  
print("Model 6 : k = 31")

## [1] "Model 6 : k = 31"

print(confusionMatrix(data = bank\_test\_predict6,reference = bank\_test\_labels))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 179 63  
## No 749 7246  
##   
## Accuracy : 0.9014   
## 95% CI : (0.8948, 0.9078)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 2.088e-05   
##   
## Kappa : 0.2721   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.19289   
## Specificity : 0.99138   
## Pos Pred Value : 0.73967   
## Neg Pred Value : 0.90632   
## Prevalence : 0.11266   
## Detection Rate : 0.02173   
## Detection Prevalence : 0.02938   
## Balanced Accuracy : 0.59213   
##   
## 'Positive' Class : Yes   
##

bank\_test\_predict7 <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = 21)  
print("Model 7 : k = 21")

## [1] "Model 7 : k = 21"

print(confusionMatrix(data = bank\_test\_predict7,reference = bank\_test\_labels))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 193 89  
## No 735 7220  
##   
## Accuracy : 0.9   
## 95% CI : (0.8933, 0.9064)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.0001243   
##   
## Kappa : 0.2813   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.20797   
## Specificity : 0.98782   
## Pos Pred Value : 0.68440   
## Neg Pred Value : 0.90761   
## Prevalence : 0.11266   
## Detection Rate : 0.02343   
## Detection Prevalence : 0.03424   
## Balanced Accuracy : 0.59790   
##   
## 'Positive' Class : Yes   
##

bank\_test\_predict8 <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = 11)  
print("Model 8 : k = 11")

## [1] "Model 8 : k = 11"

print(confusionMatrix(data = bank\_test\_predict8,reference = bank\_test\_labels))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 219 130  
## No 709 7179  
##   
## Accuracy : 0.8981   
## 95% CI : (0.8914, 0.9046)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.0008916   
##   
## Kappa : 0.2999   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.23599   
## Specificity : 0.98221   
## Pos Pred Value : 0.62751   
## Neg Pred Value : 0.91012   
## Prevalence : 0.11266   
## Detection Rate : 0.02659   
## Detection Prevalence : 0.04237   
## Balanced Accuracy : 0.60910   
##   
## 'Positive' Class : Yes   
##

Model n : k = 9 to 1 (only odd numbers)

i <- 9  
while (i > 0) {  
 bank\_test\_predict\_n <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = i)  
 print(i)  
 print(confusionMatrix(data = bank\_test\_predict\_n,reference = bank\_test\_labels))  
 i <- i - 2  
}

## [1] 9  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 226 148  
## No 702 7161  
##   
## Accuracy : 0.8968   
## 95% CI : (0.89, 0.9033)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.003164   
##   
## Kappa : 0.302   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.24353   
## Specificity : 0.97975   
## Pos Pred Value : 0.60428   
## Neg Pred Value : 0.91072   
## Prevalence : 0.11266   
## Detection Rate : 0.02744   
## Detection Prevalence : 0.04540   
## Balanced Accuracy : 0.61164   
##   
## 'Positive' Class : Yes   
##   
## [1] 7  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 237 152  
## No 691 7157  
##   
## Accuracy : 0.8977   
## 95% CI : (0.8909, 0.9041)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.001438   
##   
## Kappa : 0.3143   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.25539   
## Specificity : 0.97920   
## Pos Pred Value : 0.60925   
## Neg Pred Value : 0.91195   
## Prevalence : 0.11266   
## Detection Rate : 0.02877   
## Detection Prevalence : 0.04723   
## Balanced Accuracy : 0.61730   
##   
## 'Positive' Class : Yes   
##   
## [1] 5  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 255 195  
## No 673 7114  
##   
## Accuracy : 0.8946   
## 95% CI : (0.8878, 0.9012)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.01836   
##   
## Kappa : 0.3201   
##   
## Mcnemar's Test P-Value : < 2e-16   
##   
## Sensitivity : 0.27478   
## Specificity : 0.97332   
## Pos Pred Value : 0.56667   
## Neg Pred Value : 0.91357   
## Prevalence : 0.11266   
## Detection Rate : 0.03096   
## Detection Prevalence : 0.05463   
## Balanced Accuracy : 0.62405   
##   
## 'Positive' Class : Yes   
##   
## [1] 3  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 283 263  
## No 645 7046  
##   
## Accuracy : 0.8898   
## 95% CI : (0.8828, 0.8965)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.2492   
##   
## Kappa : 0.3279   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.30496   
## Specificity : 0.96402   
## Pos Pred Value : 0.51832   
## Neg Pred Value : 0.91614   
## Prevalence : 0.11266   
## Detection Rate : 0.03436   
## Detection Prevalence : 0.06629   
## Balanced Accuracy : 0.63449   
##   
## 'Positive' Class : Yes   
##   
## [1] 1  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 325 464  
## No 603 6845  
##   
## Accuracy : 0.8705   
## 95% CI : (0.863, 0.8776)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.3068   
##   
## Mcnemar's Test P-Value : 2.392e-05   
##   
## Sensitivity : 0.35022   
## Specificity : 0.93652   
## Pos Pred Value : 0.41191   
## Neg Pred Value : 0.91904   
## Prevalence : 0.11266   
## Detection Rate : 0.03946   
## Detection Prevalence : 0.09579   
## Balanced Accuracy : 0.64337   
##   
## 'Positive' Class : Yes   
##

Model n : k = 19 to 13 (only odd numbers)

i <- 19  
while (i > 11) {  
 bank\_test\_predict\_n <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = i)  
 print(i)  
 print(confusionMatrix(data = bank\_test\_predict\_n,reference = bank\_test\_labels))  
 i <- i - 2  
}

## [1] 19  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 202 91  
## No 726 7218  
##   
## Accuracy : 0.9008   
## 95% CI : (0.8942, 0.9072)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 4.491e-05   
##   
## Kappa : 0.2926   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.21767   
## Specificity : 0.98755   
## Pos Pred Value : 0.68942   
## Neg Pred Value : 0.90861   
## Prevalence : 0.11266   
## Detection Rate : 0.02452   
## Detection Prevalence : 0.03557   
## Balanced Accuracy : 0.60261   
##   
## 'Positive' Class : Yes   
##   
## [1] 17  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 202 102  
## No 726 7207  
##   
## Accuracy : 0.8995   
## 95% CI : (0.8928, 0.9059)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.0002161   
##   
## Kappa : 0.2884   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.21767   
## Specificity : 0.98604   
## Pos Pred Value : 0.66447   
## Neg Pred Value : 0.90848   
## Prevalence : 0.11266   
## Detection Rate : 0.02452   
## Detection Prevalence : 0.03691   
## Balanced Accuracy : 0.60186   
##   
## 'Positive' Class : Yes   
##   
## [1] 15  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 211 110  
## No 717 7199  
##   
## Accuracy : 0.8996   
## 95% CI : (0.8929, 0.906)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.0001886   
##   
## Kappa : 0.2972   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.22737   
## Specificity : 0.98495   
## Pos Pred Value : 0.65732   
## Neg Pred Value : 0.90942   
## Prevalence : 0.11266   
## Detection Rate : 0.02562   
## Detection Prevalence : 0.03897   
## Balanced Accuracy : 0.60616   
##   
## 'Positive' Class : Yes   
##   
## [1] 13  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 213 118  
## No 715 7191  
##   
## Accuracy : 0.8989   
## 95% CI : (0.8922, 0.9053)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.0004194   
##   
## Kappa : 0.2967   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.22953   
## Specificity : 0.98386   
## Pos Pred Value : 0.64350   
## Neg Pred Value : 0.90956   
## Prevalence : 0.11266   
## Detection Rate : 0.02586   
## Detection Prevalence : 0.04018   
## Balanced Accuracy : 0.60669   
##   
## 'Positive' Class : Yes   
##

Model n : k = 29 to 23 (only odd numbers)

i <- 29  
while (i > 21) {  
 bank\_test\_predict\_n <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = i)  
 print(i)  
 print(confusionMatrix(data = bank\_test\_predict\_n,reference = bank\_test\_labels))  
 i <- i - 2  
}

## [1] 29  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 184 65  
## No 744 7244  
##   
## Accuracy : 0.9018   
## 95% CI : (0.8952, 0.9081)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 1.298e-05   
##   
## Kappa : 0.2783   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.19828   
## Specificity : 0.99111   
## Pos Pred Value : 0.73896   
## Neg Pred Value : 0.90686   
## Prevalence : 0.11266   
## Detection Rate : 0.02234   
## Detection Prevalence : 0.03023   
## Balanced Accuracy : 0.59469   
##   
## 'Positive' Class : Yes   
##   
## [1] 27  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 188 71  
## No 740 7238  
##   
## Accuracy : 0.9015   
## 95% CI : (0.8949, 0.9079)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 1.785e-05   
##   
## Kappa : 0.2814   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.20259   
## Specificity : 0.99029   
## Pos Pred Value : 0.72587   
## Neg Pred Value : 0.90724   
## Prevalence : 0.11266   
## Detection Rate : 0.02282   
## Detection Prevalence : 0.03144   
## Balanced Accuracy : 0.59644   
##   
## 'Positive' Class : Yes   
##   
## [1] 25  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 189 76  
## No 739 7233  
##   
## Accuracy : 0.9011   
## 95% CI : (0.8944, 0.9074)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 3.319e-05   
##   
## Kappa : 0.2809   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.20366   
## Specificity : 0.98960   
## Pos Pred Value : 0.71321   
## Neg Pred Value : 0.90730   
## Prevalence : 0.11266   
## Detection Rate : 0.02295   
## Detection Prevalence : 0.03217   
## Balanced Accuracy : 0.59663   
##   
## 'Positive' Class : Yes   
##   
## [1] 23  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 191 79  
## No 737 7230  
##   
## Accuracy : 0.9009   
## 95% CI : (0.8943, 0.9073)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 3.864e-05   
##   
## Kappa : 0.2824   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.20582   
## Specificity : 0.98919   
## Pos Pred Value : 0.70741   
## Neg Pred Value : 0.90749   
## Prevalence : 0.11266   
## Detection Rate : 0.02319   
## Detection Prevalence : 0.03278   
## Balanced Accuracy : 0.59751   
##   
## 'Positive' Class : Yes   
##

#### 2) Best among all is for K = 3

bank\_test\_predict.best <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = 3)  
print(confusionMatrix(data = bank\_test\_predict.best,reference = bank\_test\_labels))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 283 262  
## No 645 7047  
##   
## Accuracy : 0.8899   
## 95% CI : (0.8829, 0.8966)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.2382   
##   
## Kappa : 0.3282   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.30496   
## Specificity : 0.96415   
## Pos Pred Value : 0.51927   
## Neg Pred Value : 0.91615   
## Prevalence : 0.11266   
## Detection Rate : 0.03436   
## Detection Prevalence : 0.06616   
## Balanced Accuracy : 0.63456   
##   
## 'Positive' Class : Yes   
##

### Step 3.2 - Data transformation & preparation - Normalizing the data with inbuilt R function

Redoing the first 2 steps before transformation

source("/Users/sobil/Documents/MSC/Sem 1/Data Mining & Machine Learning/Project/Bank/bank\_import\_primaryExplore.R")

## 'data.frame': 41188 obs. of 21 variables:  
## $ age : int 56 57 37 40 56 45 59 41 24 25 ...  
## $ job : Factor w/ 12 levels "admin.","blue-collar",..: 4 8 8 1 8 8 1 2 10 8 ...  
## $ marital : Factor w/ 4 levels "divorced","married",..: 2 2 2 2 2 2 2 2 3 3 ...  
## $ education : Factor w/ 8 levels "basic.4y","basic.6y",..: 1 4 4 2 4 3 6 8 6 4 ...  
## $ default : Factor w/ 3 levels "no","unknown",..: 1 2 1 1 1 2 1 2 1 1 ...  
## $ housing : Factor w/ 3 levels "no","unknown",..: 1 1 3 1 1 1 1 1 3 3 ...  
## $ loan : Factor w/ 3 levels "no","unknown",..: 1 1 1 1 3 1 1 1 1 1 ...  
## $ contact : Factor w/ 2 levels "cellular","telephone": 2 2 2 2 2 2 2 2 2 2 ...  
## $ month : Factor w/ 10 levels "apr","aug","dec",..: 7 7 7 7 7 7 7 7 7 7 ...  
## $ day\_of\_week : Factor w/ 5 levels "fri","mon","thu",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ duration : int 261 149 226 151 307 198 139 217 380 50 ...  
## $ campaign : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ pdays : int 999 999 999 999 999 999 999 999 999 999 ...  
## $ previous : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ poutcome : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ emp.var.rate : num 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 ...  
## $ cons.price.idx: num 94 94 94 94 94 ...  
## $ cons.conf.idx : num -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 ...  
## $ euribor3m : num 4.86 4.86 4.86 4.86 4.86 ...  
## $ nr.employed : num 5191 5191 5191 5191 5191 ...  
## $ y : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...  
##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 41188   
##   
##   
## | no | yes |   
## |-----------|-----------|  
## | 36548 | 4640 |   
## | 0.887 | 0.113 |   
## |-----------|-----------|  
##   
##   
##   
##

normalizing the data to standardize range of values of various characterstics

getting normalised variables in a data frame

#### 1) Normalization using scale inbulit R method based upon using z score

bank\_n\_cont <- as.data.frame(scale(bank[c(1,11,12,13,14,16,17,18,19,20)]))  
summary(bank\_n\_cont)

## age duration campaign pdays   
## Min. :-2.2093 Min. :-0.9962 Min. :-0.5659 Min. :-5.1494   
## 1st Qu.:-0.7700 1st Qu.:-0.6028 1st Qu.:-0.5659 1st Qu.: 0.1954   
## Median :-0.1942 Median :-0.3019 Median :-0.2049 Median : 0.1954   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 0.6694 3rd Qu.: 0.2342 3rd Qu.: 0.1561 3rd Qu.: 0.1954   
## Max. : 5.5632 Max. :17.9718 Max. :19.2896 Max. : 0.1954   
## previous emp.var.rate cons.price.idx cons.conf.idx   
## Min. :-0.3495 Min. :-2.2164 Min. :-2.3749 Min. :-2.2249   
## 1st Qu.:-0.3495 1st Qu.:-1.1979 1st Qu.:-0.8649 1st Qu.:-0.4748   
## Median :-0.3495 Median : 0.6481 Median : 0.2995 Median :-0.2803   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.:-0.3495 3rd Qu.: 0.8391 3rd Qu.: 0.7227 3rd Qu.: 0.8864   
## Max. :13.7948 Max. : 0.8391 Max. : 2.0581 Max. : 2.9391   
## euribor3m nr.employed   
## Min. :-1.7223 Min. :-2.8157   
## 1st Qu.:-1.3130 1st Qu.:-0.9403   
## Median : 0.7125 Median : 0.3317   
## Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 0.7724 3rd Qu.: 0.8452   
## Max. : 0.8208 Max. : 0.8452

#### 2) creating a new df of only independent variables

bank\_n <- bank[,-21]  
bank\_n$age <- bank\_n\_cont$age  
bank\_n$duration <- bank\_n\_cont$duration  
bank\_n$campaign <- bank\_n\_cont$campaign  
bank\_n$pdays <- bank\_n\_cont$pdays  
bank\_n$previous <- bank\_n\_cont$previous  
bank\_n$emp.var.rate <- bank\_n\_cont$emp.var.rate  
bank\_n$cons.price.idx <- bank\_n\_cont$cons.price.idx  
bank\_n$cons.conf.idx <- bank\_n\_cont$cons.conf.idx  
bank\_n$euribor3m <- bank\_n\_cont$euribor3m  
bank\_n$nr.employed <- bank\_n\_cont$nr.employed  
summary(bank\_n)

## age job marital   
## Min. :-2.2093 admin. :10422 divorced: 4612   
## 1st Qu.:-0.7700 blue-collar: 9254 married :24928   
## Median :-0.1942 technician : 6743 single :11568   
## Mean : 0.0000 services : 3969 unknown : 80   
## 3rd Qu.: 0.6694 management : 2924   
## Max. : 5.5632 retired : 1720   
## (Other) : 6156   
## education default housing loan   
## university.degree :12168 no :32588 no :18622 no :33950   
## high.school : 9515 unknown: 8597 unknown: 990 unknown: 990   
## basic.9y : 6045 yes : 3 yes :21576 yes : 6248   
## professional.course: 5243   
## basic.4y : 4176   
## basic.6y : 2292   
## (Other) : 1749   
## contact month day\_of\_week duration   
## cellular :26144 may :13769 fri:7827 Min. :-0.9962   
## telephone:15044 jul : 7174 mon:8514 1st Qu.:-0.6028   
## aug : 6178 thu:8623 Median :-0.3019   
## jun : 5318 tue:8090 Mean : 0.0000   
## nov : 4101 wed:8134 3rd Qu.: 0.2342   
## apr : 2632 Max. :17.9718   
## (Other): 2016   
## campaign pdays previous poutcome   
## Min. :-0.5659 Min. :-5.1494 Min. :-0.3495 failure : 4252   
## 1st Qu.:-0.5659 1st Qu.: 0.1954 1st Qu.:-0.3495 nonexistent:35563   
## Median :-0.2049 Median : 0.1954 Median :-0.3495 success : 1373   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 0.1561 3rd Qu.: 0.1954 3rd Qu.:-0.3495   
## Max. :19.2896 Max. : 0.1954 Max. :13.7948   
##   
## emp.var.rate cons.price.idx cons.conf.idx euribor3m   
## Min. :-2.2164 Min. :-2.3749 Min. :-2.2249 Min. :-1.7223   
## 1st Qu.:-1.1979 1st Qu.:-0.8649 1st Qu.:-0.4748 1st Qu.:-1.3130   
## Median : 0.6481 Median : 0.2995 Median :-0.2803 Median : 0.7125   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 0.8391 3rd Qu.: 0.7227 3rd Qu.: 0.8864 3rd Qu.: 0.7724   
## Max. : 0.8391 Max. : 2.0581 Max. : 2.9391 Max. : 0.8208   
##   
## nr.employed   
## Min. :-2.8157   
## 1st Qu.:-0.9403   
## Median : 0.3317   
## Mean : 0.0000   
## 3rd Qu.: 0.8452   
## Max. : 0.8452   
##

#### 3) creating dummy columns for categorical columns

bank\_n <- fastDummies::dummy\_cols(bank\_n, remove\_first\_dummy = TRUE) # remove\_first\_dummy = TRUE to avoid multi-collinearity

removing categorical columns from df (dummy are included)

bank\_n <- bank\_n[,-c(2:10,15)]

#### 4) creating training and testing dataset from exisitng sample

indx <- createDataPartition(bank$y, p = 0.8, list = FALSE)  
bank\_train <- bank\_n[indx,]  
bank\_test <- bank\_n[- indx,]

creating lables for test and training data sets

bank\_train\_labels <- bank[indx,21]  
bank\_test\_labels <- bank[- indx,21]

### Step 4.2 : Training a model on the data - Normalizing the data with inbuilt R function

using the knn method of class package with K odd

bank\_test\_predict <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = 51)

### Step 5.2 : Evaluating the model - Normalizing the data with inbuilt R function

“Model 1 : k = 51”

confusionMatrix(data = bank\_test\_predict,reference = bank\_test\_labels)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 365 150  
## No 563 7159  
##   
## Accuracy : 0.9134   
## 95% CI : (0.9072, 0.9194)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 4.566e-15   
##   
## Kappa : 0.4627   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.39332   
## Specificity : 0.97948   
## Pos Pred Value : 0.70874   
## Neg Pred Value : 0.92709   
## Prevalence : 0.11266   
## Detection Rate : 0.04431   
## Detection Prevalence : 0.06252   
## Balanced Accuracy : 0.68640   
##   
## 'Positive' Class : Yes   
##

### Step 6.2 : Improving the model - Normalizing the data with inbuilt R function

#### 1) Changing k values

Skipped model for K values which were higher than 51 (please refer the R script for model performance) - overall the performance was poor than below

Model n : k = 31 to 1 (only odd numbers)"

i <- 31  
while (i > 0) {  
 bank\_test\_predict\_n <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = i)  
 print(i)  
 print(confusionMatrix(data = bank\_test\_predict\_n,reference = bank\_test\_labels))  
 i <- i - 2  
}

## [1] 31  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 375 160  
## No 553 7149  
##   
## Accuracy : 0.9134   
## 95% CI : (0.9072, 0.9194)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 4.566e-15   
##   
## Kappa : 0.4689   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.40409   
## Specificity : 0.97811   
## Pos Pred Value : 0.70093   
## Neg Pred Value : 0.92820   
## Prevalence : 0.11266   
## Detection Rate : 0.04553   
## Detection Prevalence : 0.06495   
## Balanced Accuracy : 0.69110   
##   
## 'Positive' Class : Yes   
##   
## [1] 29  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 382 169  
## No 546 7140  
##   
## Accuracy : 0.9132   
## 95% CI : (0.9069, 0.9192)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 8.234e-15   
##   
## Kappa : 0.4723   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.41164   
## Specificity : 0.97688   
## Pos Pred Value : 0.69328   
## Neg Pred Value : 0.92896   
## Prevalence : 0.11266   
## Detection Rate : 0.04638   
## Detection Prevalence : 0.06689   
## Balanced Accuracy : 0.69426   
##   
## 'Positive' Class : Yes   
##   
## [1] 27  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 376 170  
## No 552 7139  
##   
## Accuracy : 0.9123   
## 95% CI : (0.906, 0.9184)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 6.182e-14   
##   
## Kappa : 0.4656   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.40517   
## Specificity : 0.97674   
## Pos Pred Value : 0.68864   
## Neg Pred Value : 0.92823   
## Prevalence : 0.11266   
## Detection Rate : 0.04565   
## Detection Prevalence : 0.06629   
## Balanced Accuracy : 0.69096   
##   
## 'Positive' Class : Yes   
##   
## [1] 25  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 381 175  
## No 547 7134  
##   
## Accuracy : 0.9123   
## 95% CI : (0.906, 0.9184)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 6.182e-14   
##   
## Kappa : 0.4686   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.41056   
## Specificity : 0.97606   
## Pos Pred Value : 0.68525   
## Neg Pred Value : 0.92879   
## Prevalence : 0.11266   
## Detection Rate : 0.04625   
## Detection Prevalence : 0.06750   
## Balanced Accuracy : 0.69331   
##   
## 'Positive' Class : Yes   
##   
## [1] 23  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 372 183  
## No 556 7126  
##   
## Accuracy : 0.9103   
## 95% CI : (0.9039, 0.9164)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 6.102e-12   
##   
## Kappa : 0.4558   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.40086   
## Specificity : 0.97496   
## Pos Pred Value : 0.67027   
## Neg Pred Value : 0.92762   
## Prevalence : 0.11266   
## Detection Rate : 0.04516   
## Detection Prevalence : 0.06738   
## Balanced Accuracy : 0.68791   
##   
## 'Positive' Class : Yes   
##   
## [1] 21  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 381 189  
## No 547 7120  
##   
## Accuracy : 0.9106   
## 95% CI : (0.9043, 0.9167)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 2.799e-12   
##   
## Kappa : 0.4626   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.41056   
## Specificity : 0.97414   
## Pos Pred Value : 0.66842   
## Neg Pred Value : 0.92866   
## Prevalence : 0.11266   
## Detection Rate : 0.04625   
## Detection Prevalence : 0.06920   
## Balanced Accuracy : 0.69235   
##   
## 'Positive' Class : Yes   
##   
## [1] 19  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 383 197  
## No 545 7112  
##   
## Accuracy : 0.9099   
## 95% CI : (0.9035, 0.916)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 1.313e-11   
##   
## Kappa : 0.4613   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.41272   
## Specificity : 0.97305   
## Pos Pred Value : 0.66034   
## Neg Pred Value : 0.92882   
## Prevalence : 0.11266   
## Detection Rate : 0.04650   
## Detection Prevalence : 0.07041   
## Balanced Accuracy : 0.69288   
##   
## 'Positive' Class : Yes   
##   
## [1] 17  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 382 190  
## No 546 7119  
##   
## Accuracy : 0.9106   
## 95% CI : (0.9043, 0.9167)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 2.799e-12   
##   
## Kappa : 0.4632   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.41164   
## Specificity : 0.97400   
## Pos Pred Value : 0.66783   
## Neg Pred Value : 0.92877   
## Prevalence : 0.11266   
## Detection Rate : 0.04638   
## Detection Prevalence : 0.06944   
## Balanced Accuracy : 0.69282   
##   
## 'Positive' Class : Yes   
##   
## [1] 15  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 393 212  
## No 535 7097  
##   
## Accuracy : 0.9093   
## 95% CI : (0.9029, 0.9154)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 4.574e-11   
##   
## Kappa : 0.4652   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.42349   
## Specificity : 0.97099   
## Pos Pred Value : 0.64959   
## Neg Pred Value : 0.92990   
## Prevalence : 0.11266   
## Detection Rate : 0.04771   
## Detection Prevalence : 0.07345   
## Balanced Accuracy : 0.69724   
##   
## 'Positive' Class : Yes   
##   
## [1] 13  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 395 215  
## No 533 7094  
##   
## Accuracy : 0.9092   
## 95% CI : (0.9028, 0.9153)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 5.845e-11   
##   
## Kappa : 0.4659   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.42565   
## Specificity : 0.97058   
## Pos Pred Value : 0.64754   
## Neg Pred Value : 0.93012   
## Prevalence : 0.11266   
## Detection Rate : 0.04795   
## Detection Prevalence : 0.07406   
## Balanced Accuracy : 0.69812   
##   
## 'Positive' Class : Yes   
##   
## [1] 11  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 390 224  
## No 538 7085  
##   
## Accuracy : 0.9075   
## 95% CI : (0.901, 0.9137)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 1.559e-09   
##   
## Kappa : 0.4571   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.42026   
## Specificity : 0.96935   
## Pos Pred Value : 0.63518   
## Neg Pred Value : 0.92942   
## Prevalence : 0.11266   
## Detection Rate : 0.04735   
## Detection Prevalence : 0.07454   
## Balanced Accuracy : 0.69481   
##   
## 'Positive' Class : Yes   
##   
## [1] 9  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 411 234  
## No 517 7075  
##   
## Accuracy : 0.9088   
## 95% CI : (0.9024, 0.915)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 1.21e-10   
##   
## Kappa : 0.474   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.44289   
## Specificity : 0.96798   
## Pos Pred Value : 0.63721   
## Neg Pred Value : 0.93190   
## Prevalence : 0.11266   
## Detection Rate : 0.04990   
## Detection Prevalence : 0.07831   
## Balanced Accuracy : 0.70544   
##   
## 'Positive' Class : Yes   
##   
## [1] 7  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 410 257  
## No 518 7052  
##   
## Accuracy : 0.9059   
## 95% CI : (0.8994, 0.9121)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 2.564e-08   
##   
## Kappa : 0.4636   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.44181   
## Specificity : 0.96484   
## Pos Pred Value : 0.61469   
## Neg Pred Value : 0.93157   
## Prevalence : 0.11266   
## Detection Rate : 0.04978   
## Detection Prevalence : 0.08098   
## Balanced Accuracy : 0.70332   
##   
## 'Positive' Class : Yes   
##   
## [1] 5  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 409 287  
## No 519 7022  
##   
## Accuracy : 0.9021   
## 95% CI : (0.8955, 0.9085)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 7.978e-06   
##   
## Kappa : 0.4506   
##   
## Mcnemar's Test P-Value : 4.064e-16   
##   
## Sensitivity : 0.44073   
## Specificity : 0.96073   
## Pos Pred Value : 0.58764   
## Neg Pred Value : 0.93118   
## Prevalence : 0.11266   
## Detection Rate : 0.04965   
## Detection Prevalence : 0.08450   
## Balanced Accuracy : 0.70073   
##   
## 'Positive' Class : Yes   
##   
## [1] 3  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 421 336  
## No 507 6973  
##   
## Accuracy : 0.8977   
## 95% CI : (0.8909, 0.9041)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.001438   
##   
## Kappa : 0.4434   
##   
## Mcnemar's Test P-Value : 4.767e-09   
##   
## Sensitivity : 0.45366   
## Specificity : 0.95403   
## Pos Pred Value : 0.55614   
## Neg Pred Value : 0.93222   
## Prevalence : 0.11266   
## Detection Rate : 0.05111   
## Detection Prevalence : 0.09190   
## Balanced Accuracy : 0.70385   
##   
## 'Positive' Class : Yes   
##   
## [1] 1  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 426 479  
## No 502 6830  
##   
## Accuracy : 0.8809   
## 95% CI : (0.8737, 0.8878)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.9681   
##   
## Kappa : 0.3978   
##   
## Mcnemar's Test P-Value : 0.4824   
##   
## Sensitivity : 0.45905   
## Specificity : 0.93446   
## Pos Pred Value : 0.47072   
## Neg Pred Value : 0.93153   
## Prevalence : 0.11266   
## Detection Rate : 0.05172   
## Detection Prevalence : 0.10987   
## Balanced Accuracy : 0.69676   
##   
## 'Positive' Class : Yes   
##

#### 2) Best among all is for K = 9

bank\_test\_predict.best <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = 9)  
print(confusionMatrix(data = bank\_test\_predict.best,reference = bank\_test\_labels))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 411 235  
## No 517 7074  
##   
## Accuracy : 0.9087   
## 95% CI : (0.9023, 0.9148)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 1.537e-10   
##   
## Kappa : 0.4736   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.44289   
## Specificity : 0.96785   
## Pos Pred Value : 0.63622   
## Neg Pred Value : 0.93189   
## Prevalence : 0.11266   
## Detection Rate : 0.04990   
## Detection Prevalence : 0.07843   
## Balanced Accuracy : 0.70537   
##   
## 'Positive' Class : Yes   
##

### Step 3.3 - Data transformation & preparation - Only numeric predictors

Redoing the first 2 steps before transformation

source("/Users/sobil/Documents/MSC/Sem 1/Data Mining & Machine Learning/Project/Bank/bank\_import\_primaryExplore.R")

## 'data.frame': 41188 obs. of 21 variables:  
## $ age : int 56 57 37 40 56 45 59 41 24 25 ...  
## $ job : Factor w/ 12 levels "admin.","blue-collar",..: 4 8 8 1 8 8 1 2 10 8 ...  
## $ marital : Factor w/ 4 levels "divorced","married",..: 2 2 2 2 2 2 2 2 3 3 ...  
## $ education : Factor w/ 8 levels "basic.4y","basic.6y",..: 1 4 4 2 4 3 6 8 6 4 ...  
## $ default : Factor w/ 3 levels "no","unknown",..: 1 2 1 1 1 2 1 2 1 1 ...  
## $ housing : Factor w/ 3 levels "no","unknown",..: 1 1 3 1 1 1 1 1 3 3 ...  
## $ loan : Factor w/ 3 levels "no","unknown",..: 1 1 1 1 3 1 1 1 1 1 ...  
## $ contact : Factor w/ 2 levels "cellular","telephone": 2 2 2 2 2 2 2 2 2 2 ...  
## $ month : Factor w/ 10 levels "apr","aug","dec",..: 7 7 7 7 7 7 7 7 7 7 ...  
## $ day\_of\_week : Factor w/ 5 levels "fri","mon","thu",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ duration : int 261 149 226 151 307 198 139 217 380 50 ...  
## $ campaign : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ pdays : int 999 999 999 999 999 999 999 999 999 999 ...  
## $ previous : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ poutcome : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ emp.var.rate : num 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 ...  
## $ cons.price.idx: num 94 94 94 94 94 ...  
## $ cons.conf.idx : num -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 ...  
## $ euribor3m : num 4.86 4.86 4.86 4.86 4.86 ...  
## $ nr.employed : num 5191 5191 5191 5191 5191 ...  
## $ y : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...  
##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 41188   
##   
##   
## | no | yes |   
## |-----------|-----------|  
## | 36548 | 4640 |   
## | 0.887 | 0.113 |   
## |-----------|-----------|  
##   
##   
##   
##

#### 1) Creating new data frame with only numeric predicors and normalizaing them

From previous section, it is clear that inbuilt normalization is performing better, Thus Normalization using scale inbulit R method based upon using z score

bank\_n <- as.data.frame(scale(bank[c(1,11,12,13,14,16,17,18,19,20)]))  
summary(bank\_n)

## age duration campaign pdays   
## Min. :-2.2093 Min. :-0.9962 Min. :-0.5659 Min. :-5.1494   
## 1st Qu.:-0.7700 1st Qu.:-0.6028 1st Qu.:-0.5659 1st Qu.: 0.1954   
## Median :-0.1942 Median :-0.3019 Median :-0.2049 Median : 0.1954   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 0.6694 3rd Qu.: 0.2342 3rd Qu.: 0.1561 3rd Qu.: 0.1954   
## Max. : 5.5632 Max. :17.9718 Max. :19.2896 Max. : 0.1954   
## previous emp.var.rate cons.price.idx cons.conf.idx   
## Min. :-0.3495 Min. :-2.2164 Min. :-2.3749 Min. :-2.2249   
## 1st Qu.:-0.3495 1st Qu.:-1.1979 1st Qu.:-0.8649 1st Qu.:-0.4748   
## Median :-0.3495 Median : 0.6481 Median : 0.2995 Median :-0.2803   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.:-0.3495 3rd Qu.: 0.8391 3rd Qu.: 0.7227 3rd Qu.: 0.8864   
## Max. :13.7948 Max. : 0.8391 Max. : 2.0581 Max. : 2.9391   
## euribor3m nr.employed   
## Min. :-1.7223 Min. :-2.8157   
## 1st Qu.:-1.3130 1st Qu.:-0.9403   
## Median : 0.7125 Median : 0.3317   
## Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 0.7724 3rd Qu.: 0.8452   
## Max. : 0.8208 Max. : 0.8452

#### 2) creating training and testing dataset from exisitng sample

indx <- createDataPartition(bank$y, p = 0.8, list = FALSE)  
bank\_train <- bank\_n[indx,]  
bank\_test <- bank\_n[- indx,]

creating lables for test and training data sets

bank\_train\_labels <- bank[indx,21]  
bank\_test\_labels <- bank[- indx,21]

### Step 4.3 : Training a model on the data - Only numeric predictors

using the knn method of class package with K odd

bank\_test\_predict <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = 51)

### Step 5.3 : Evaluating the model - Only numeric predictors

“Model 1 : k = 51”

confusionMatrix(data = bank\_test\_predict,reference = bank\_test\_labels)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 417 211  
## No 511 7098  
##   
## Accuracy : 0.9123   
## 95% CI : (0.906, 0.9184)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 6.182e-14   
##   
## Kappa : 0.4896   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.44935   
## Specificity : 0.97113   
## Pos Pred Value : 0.66401   
## Neg Pred Value : 0.93284   
## Prevalence : 0.11266   
## Detection Rate : 0.05063   
## Detection Prevalence : 0.07624   
## Balanced Accuracy : 0.71024   
##   
## 'Positive' Class : Yes   
##

### Step 6.3 : Improving the model - Only numeric predictors

#### 1) Changing k values

Skipped model for K values which were higher than 51 (please refer the R script for model performance) - overall the performance was poor than below

Model n : k = 31 to 1 (only odd numbers)"

i <- 31  
while (i > 0) {  
 bank\_test\_predict\_n <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = i)  
 print(i)  
 print(confusionMatrix(data = bank\_test\_predict\_n,reference = bank\_test\_labels))  
 i <- i - 2  
}

## [1] 31  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 448 211  
## No 480 7098  
##   
## Accuracy : 0.9161   
## 95% CI : (0.9099, 0.922)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5196   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.48276   
## Specificity : 0.97113   
## Pos Pred Value : 0.67982   
## Neg Pred Value : 0.93666   
## Prevalence : 0.11266   
## Detection Rate : 0.05439   
## Detection Prevalence : 0.08000   
## Balanced Accuracy : 0.72695   
##   
## 'Positive' Class : Yes   
##   
## [1] 29  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 447 216  
## No 481 7093  
##   
## Accuracy : 0.9154   
## 95% CI : (0.9092, 0.9213)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5165   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.48168   
## Specificity : 0.97045   
## Pos Pred Value : 0.67421   
## Neg Pred Value : 0.93649   
## Prevalence : 0.11266   
## Detection Rate : 0.05427   
## Detection Prevalence : 0.08049   
## Balanced Accuracy : 0.72606   
##   
## 'Positive' Class : Yes   
##   
## [1] 27  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 449 216  
## No 479 7093  
##   
## Accuracy : 0.9156   
## 95% CI : (0.9094, 0.9215)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5184   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.48384   
## Specificity : 0.97045   
## Pos Pred Value : 0.67519   
## Neg Pred Value : 0.93674   
## Prevalence : 0.11266   
## Detection Rate : 0.05451   
## Detection Prevalence : 0.08073   
## Balanced Accuracy : 0.72714   
##   
## 'Positive' Class : Yes   
##   
## [1] 25  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 453 217  
## No 475 7092  
##   
## Accuracy : 0.916   
## 95% CI : (0.9098, 0.9219)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5218   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.48815   
## Specificity : 0.97031   
## Pos Pred Value : 0.67612   
## Neg Pred Value : 0.93723   
## Prevalence : 0.11266   
## Detection Rate : 0.05500   
## Detection Prevalence : 0.08134   
## Balanced Accuracy : 0.72923   
##   
## 'Positive' Class : Yes   
##   
## [1] 23  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 463 230  
## No 465 7079  
##   
## Accuracy : 0.9156   
## 95% CI : (0.9094, 0.9215)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5255   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.49892   
## Specificity : 0.96853   
## Pos Pred Value : 0.66811   
## Neg Pred Value : 0.93836   
## Prevalence : 0.11266   
## Detection Rate : 0.05621   
## Detection Prevalence : 0.08413   
## Balanced Accuracy : 0.73373   
##   
## 'Positive' Class : Yes   
##   
## [1] 21  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 464 221  
## No 464 7088  
##   
## Accuracy : 0.9168   
## 95% CI : (0.9107, 0.9227)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5304   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.50000   
## Specificity : 0.96976   
## Pos Pred Value : 0.67737   
## Neg Pred Value : 0.93856   
## Prevalence : 0.11266   
## Detection Rate : 0.05633   
## Detection Prevalence : 0.08316   
## Balanced Accuracy : 0.73488   
##   
## 'Positive' Class : Yes   
##   
## [1] 19  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 469 228  
## No 459 7081  
##   
## Accuracy : 0.9166   
## 95% CI : (0.9104, 0.9225)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.532   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.50539   
## Specificity : 0.96881   
## Pos Pred Value : 0.67288   
## Neg Pred Value : 0.93912   
## Prevalence : 0.11266   
## Detection Rate : 0.05694   
## Detection Prevalence : 0.08462   
## Balanced Accuracy : 0.73710   
##   
## 'Positive' Class : Yes   
##   
## [1] 17  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 469 241  
## No 459 7068  
##   
## Accuracy : 0.915   
## 95% CI : (0.9088, 0.921)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5264   
##   
## Mcnemar's Test P-Value : 2.368e-16   
##   
## Sensitivity : 0.50539   
## Specificity : 0.96703   
## Pos Pred Value : 0.66056   
## Neg Pred Value : 0.93902   
## Prevalence : 0.11266   
## Detection Rate : 0.05694   
## Detection Prevalence : 0.08620   
## Balanced Accuracy : 0.73621   
##   
## 'Positive' Class : Yes   
##   
## [1] 15  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 472 239  
## No 456 7070  
##   
## Accuracy : 0.9156   
## 95% CI : (0.9094, 0.9215)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.53   
##   
## Mcnemar's Test P-Value : 2.541e-16   
##   
## Sensitivity : 0.50862   
## Specificity : 0.96730   
## Pos Pred Value : 0.66385   
## Neg Pred Value : 0.93941   
## Prevalence : 0.11266   
## Detection Rate : 0.05730   
## Detection Prevalence : 0.08632   
## Balanced Accuracy : 0.73796   
##   
## 'Positive' Class : Yes   
##   
## [1] 13  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 466 256  
## No 462 7053  
##   
## Accuracy : 0.9128   
## 95% CI : (0.9065, 0.9188)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 1.971e-14   
##   
## Kappa : 0.5173   
##   
## Mcnemar's Test P-Value : 2.001e-14   
##   
## Sensitivity : 0.50216   
## Specificity : 0.96497   
## Pos Pred Value : 0.64543   
## Neg Pred Value : 0.93852   
## Prevalence : 0.11266   
## Detection Rate : 0.05657   
## Detection Prevalence : 0.08765   
## Balanced Accuracy : 0.73356   
##   
## 'Positive' Class : Yes   
##   
## [1] 11  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 458 270  
## No 470 7039  
##   
## Accuracy : 0.9102   
## 95% CI : (0.9038, 0.9163)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 7.889e-12   
##   
## Kappa : 0.504   
##   
## Mcnemar's Test P-Value : 2.566e-13   
##   
## Sensitivity : 0.49353   
## Specificity : 0.96306   
## Pos Pred Value : 0.62912   
## Neg Pred Value : 0.93741   
## Prevalence : 0.11266   
## Detection Rate : 0.05560   
## Detection Prevalence : 0.08838   
## Balanced Accuracy : 0.72830   
##   
## 'Positive' Class : Yes   
##   
## [1] 9  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 457 288  
## No 471 7021  
##   
## Accuracy : 0.9079   
## 95% CI : (0.9014, 0.914)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 7.895e-10   
##   
## Kappa : 0.4957   
##   
## Mcnemar's Test P-Value : 3.944e-11   
##   
## Sensitivity : 0.49246   
## Specificity : 0.96060   
## Pos Pred Value : 0.61342   
## Neg Pred Value : 0.93713   
## Prevalence : 0.11266   
## Detection Rate : 0.05548   
## Detection Prevalence : 0.09045   
## Balanced Accuracy : 0.72653   
##   
## 'Positive' Class : Yes   
##   
## [1] 7  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 461 286  
## No 467 7023  
##   
## Accuracy : 0.9086   
## 95% CI : (0.9022, 0.9147)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 1.950e-10   
##   
## Kappa : 0.5002   
##   
## Mcnemar's Test P-Value : 5.397e-11   
##   
## Sensitivity : 0.49677   
## Specificity : 0.96087   
## Pos Pred Value : 0.61714   
## Neg Pred Value : 0.93765   
## Prevalence : 0.11266   
## Detection Rate : 0.05597   
## Detection Prevalence : 0.09069   
## Balanced Accuracy : 0.72882   
##   
## 'Positive' Class : Yes   
##   
## [1] 5  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 478 308  
## No 450 7001  
##   
## Accuracy : 0.908   
## 95% CI : (0.9015, 0.9141)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 6.276e-10   
##   
## Kappa : 0.5068   
##   
## Mcnemar's Test P-Value : 3.034e-07   
##   
## Sensitivity : 0.51509   
## Specificity : 0.95786   
## Pos Pred Value : 0.60814   
## Neg Pred Value : 0.93961   
## Prevalence : 0.11266   
## Detection Rate : 0.05803   
## Detection Prevalence : 0.09542   
## Balanced Accuracy : 0.73647   
##   
## 'Positive' Class : Yes   
##   
## [1] 3  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 475 365  
## No 453 6944  
##   
## Accuracy : 0.9007   
## 95% CI : (0.894, 0.9071)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 5.214e-05   
##   
## Kappa : 0.4819   
##   
## Mcnemar's Test P-Value : 0.002351   
##   
## Sensitivity : 0.51185   
## Specificity : 0.95006   
## Pos Pred Value : 0.56548   
## Neg Pred Value : 0.93876   
## Prevalence : 0.11266   
## Detection Rate : 0.05767   
## Detection Prevalence : 0.10198   
## Balanced Accuracy : 0.73096   
##   
## 'Positive' Class : Yes   
##   
## [1] 1  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 485 479  
## No 443 6830  
##   
## Accuracy : 0.8881   
## 95% CI : (0.8811, 0.8948)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.4257   
##   
## Kappa : 0.4495   
##   
## Mcnemar's Test P-Value : 0.2490   
##   
## Sensitivity : 0.52263   
## Specificity : 0.93446   
## Pos Pred Value : 0.50311   
## Neg Pred Value : 0.93909   
## Prevalence : 0.11266   
## Detection Rate : 0.05888   
## Detection Prevalence : 0.11703   
## Balanced Accuracy : 0.72855   
##   
## 'Positive' Class : Yes   
##

#### 2) Best among all is for K = 19

bank\_test\_predict.best <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = 19)  
print(confusionMatrix(data = bank\_test\_predict.best,reference = bank\_test\_labels))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 469 227  
## No 459 7082  
##   
## Accuracy : 0.9167   
## 95% CI : (0.9105, 0.9226)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5324   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.50539   
## Specificity : 0.96894   
## Pos Pred Value : 0.67385   
## Neg Pred Value : 0.93913   
## Prevalence : 0.11266   
## Detection Rate : 0.05694   
## Detection Prevalence : 0.08450   
## Balanced Accuracy : 0.73717   
##   
## 'Positive' Class : Yes   
##

confusionMatrix (data = bank\_test\_predict.best,reference = bank\_test\_labels, mode="prec\_recall")

##Confusion Matrix and Statistics

##

## Reference

##Prediction Yes No

## Yes 469 226

## No 459 7083

##

## Accuracy : 0.9168

## 95% CI : (0.9107, 0.9227)

## No Information Rate : 0.8873

## P-Value [Acc > NIR] : < 2.2e-16

##

## Kappa : 0.5329

##

## Mcnemar's Test P-Value : < 2.2e-16

##

## Precision : 0.67482

## Recall : 0.50539

## F1 : 0.57794

## Prevalence : 0.11266

## Detection Rate : 0.05694

## Detection Prevalence : 0.08438

## Balanced Accuracy : 0.73723

##

## 'Positive' Class : Yes

##

## Part 2 - Naive Bayes

### Step 1 : Collecting Data

Same as of Part 1

### Step 2 : Exploring, preprocessing and cleaning the data

Primary setup

knitr::opts\_knit$set(root.dir = '/Users/sobil/Documents/MSC/Sem 1/Data Mining & Machine Learning/Project/Bank/')  
remove(list = ls())  
set.seed(1)

loading all the libraries required

library(ggplot2)  
library(ggthemes)  
library(caret)

## Loading required package: lattice

library(e1071)

Same as part 1

source("bank\_import\_primaryExplore.R")

## 'data.frame': 41188 obs. of 21 variables:  
## $ age : int 56 57 37 40 56 45 59 41 24 25 ...  
## $ job : Factor w/ 12 levels "admin.","blue-collar",..: 4 8 8 1 8 8 1 2 10 8 ...  
## $ marital : Factor w/ 4 levels "divorced","married",..: 2 2 2 2 2 2 2 2 3 3 ...  
## $ education : Factor w/ 8 levels "basic.4y","basic.6y",..: 1 4 4 2 4 3 6 8 6 4 ...  
## $ default : Factor w/ 3 levels "no","unknown",..: 1 2 1 1 1 2 1 2 1 1 ...  
## $ housing : Factor w/ 3 levels "no","unknown",..: 1 1 3 1 1 1 1 1 3 3 ...  
## $ loan : Factor w/ 3 levels "no","unknown",..: 1 1 1 1 3 1 1 1 1 1 ...  
## $ contact : Factor w/ 2 levels "cellular","telephone": 2 2 2 2 2 2 2 2 2 2 ...  
## $ month : Factor w/ 10 levels "apr","aug","dec",..: 7 7 7 7 7 7 7 7 7 7 ...  
## $ day\_of\_week : Factor w/ 5 levels "fri","mon","thu",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ duration : int 261 149 226 151 307 198 139 217 380 50 ...  
## $ campaign : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ pdays : int 999 999 999 999 999 999 999 999 999 999 ...  
## $ previous : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ poutcome : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ emp.var.rate : num 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 ...  
## $ cons.price.idx: num 94 94 94 94 94 ...  
## $ cons.conf.idx : num -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 ...  
## $ euribor3m : num 4.86 4.86 4.86 4.86 4.86 ...  
## $ nr.employed : num 5191 5191 5191 5191 5191 ...  
## $ y : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...  
##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 41188   
##   
##   
## | no | yes |   
## |-----------|-----------|  
## | 36548 | 4640 |   
## | 0.887 | 0.113 |   
## |-----------|-----------|  
##   
##   
##   
##

### Step 3 - Data transformation & preparation

#### 1) creating a new df to convert the numeric columns (continous) to factors by binning

bank\_n <- bank

checking histograms and creating factors

theme\_set(theme\_gdocs())

age

pl.age <- ggplot(bank\_n, aes(x=age)) + geom\_histogram(bins = 10, color = 'blue', aes(fill=..count..), alpha = 0.4) + xlab('Age') + ylab('Count') + ggtitle('Age distribution Plot')  
pl.age

A screenshot of a cell phone

Description automatically generated

bank\_n$age <- cut(bank\_n$age,  
 breaks = c(1, 10, 20, 30, 40, 50, 60, 70, 80, max(bank\_n$age)),  
 labels = c("1-10", "10-20", "20-30", "30-40", "40-50", "50-60", "60-70", "70-80", "80+"))  
table(bank\_n$age)

##   
## 1-10 10-20 20-30 30-40 40-50 50-60 60-70 70-80 80+   
## 0 140 7243 16385 10240 6270 488 303 119

duration

pl.duration <- ggplot(bank\_n, aes(x=duration)) + geom\_histogram(bins = 15, color = 'blue', aes(fill=..count..), alpha = 0.4) + xlab('Duration') + ylab('Count') + ggtitle('Duration distribution Plot') +stat\_bin(breaks=c(-0.004, seq(0.001,1.0, by=0.005)))  
pl.duration

A screenshot of a cell phone

Description automatically generated

summary(bank\_n$duration)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0 102.0 180.0 258.3 319.0 4918.0

bank\_n$duration <- cut(bank\_n$duration,  
 breaks = c(0, 10, 30, 70, 110, 150, 190, 250, 350, 600, 900, 1200, max(bank\_n$duration)),  
 labels = c("1-10","30-70", "30-70", "70-110", "110-150", "150-190", "190-250", "250-350", "350-600", "600-900", "900-1200", "1200+"))  
table(bank\_n$age)

##   
## 1-10 10-20 20-30 30-40 40-50 50-60 60-70 70-80 80+   
## 0 140 7243 16385 10240 6270 488 303 119

campaign

pl.campaign<- ggplot(bank\_n, aes(x=campaign)) + geom\_histogram(bins = 15, color = 'blue', aes(fill=..count..), alpha = 0.4) + xlab('campaign') + ylab('Count') + ggtitle('campaign distribution Plot')  
pl.campaign

A screenshot of a cell phone

Description automatically generated

summary(bank\_n$campaign)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 2.000 2.568 3.000 56.000

bank\_n$campaign <- cut(bank\_n$campaign,  
 breaks = c(1, 2, 3, 5, 10, 15, 20, max(bank\_n$campaign)),  
 labels = c("1-2","2-3", "3-5", "5-10", "10-15", "15-20", "20+"))  
table(bank\_n$campaign)

##   
## 1-2 2-3 3-5 5-10 10-15 15-20 20+   
## 10570 5341 4250 2516 514 198 157

pdays

summary(bank\_n$pdays)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0 999.0 999.0 962.5 999.0 999.0

table(bank\_n$pdays)

##   
## 0 1 2 3 4 5 6 7 8 9 10 11 12   
## 15 26 61 439 118 46 412 60 18 64 52 28 58   
## 13 14 15 16 17 18 19 20 21 22 25 26 27   
## 36 20 24 11 8 7 3 1 2 3 1 1 1   
## 999   
## 39673

bank\_n$pdays <- cut(bank\_n$pdays,  
 breaks = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 28, max(bank\_n$pdays)),  
 labels = c("0","1", "2", "3", "4", "5", "6", "7", "8", "9", "10-15", "15-20", "20-28", "999"))  
summary(bank\_n$pdays)

## 0 1 2 3 4 5 6 7 8 9 10-15 15-20 20-28   
## 26 61 439 118 46 412 60 18 64 52 166 30 8   
## 999 NA's   
## 39673 15

previous

summary(bank\_n$previous)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000 0.000 0.000 0.173 0.000 7.000

table(bank\_n$previous)

##   
## 0 1 2 3 4 5 6 7   
## 35563 4561 754 216 70 18 5 1

bank\_n$previous <- as.factor(bank\_n$previous)  
table(bank\_n$previous)

##   
## 0 1 2 3 4 5 6 7   
## 35563 4561 754 216 70 18 5 1

emp.var.rate

summary(bank\_n$emp.var.rate)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -3.40000 -1.80000 1.10000 0.08189 1.40000 1.40000

table(bank\_n$emp.var.rate)

##   
## -3.4 -3 -2.9 -1.8 -1.7 -1.1 -0.2 -0.1 1.1 1.4   
## 1071 172 1663 9184 773 635 10 3683 7763 16234

bank\_n$emp.var.rate <- as.factor(bank\_n$emp.var.rate)  
summary(bank\_n$emp.var.rate)

## -3.4 -3 -2.9 -1.8 -1.7 -1.1 -0.2 -0.1 1.1 1.4   
## 1071 172 1663 9184 773 635 10 3683 7763 16234

cons.price.idx

summary(bank\_n$cons.price.idx)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 92.20 93.08 93.75 93.58 93.99 94.77

table(bank\_n$cons.price.idx)

##   
## 92.201 92.379 92.431 92.469 92.649 92.713 92.756 92.843 92.893 92.963 93.075   
## 770 267 447 178 357 172 10 282 5794 715 2458   
## 93.2 93.369 93.444 93.749 93.798 93.876 93.918 93.994 94.027 94.055 94.199   
## 3616 264 5175 174 67 212 6685 7763 233 229 303   
## 94.215 94.465 94.601 94.767   
## 311 4374 204 128

hist(bank\_n$cons.price.idx, col = "lightblue")

A screenshot of a cell phone

Description automatically generated

bank\_n$cons.price.idx <- cut(bank\_n$cons.price.idx,  
 breaks = c(92,93, 93.5, 94, max(bank\_n$cons.price.idx)),  
 labels = c("92-93", "93-93.5", "93.5-94", "94+"))  
summary(bank\_n$cons.price.idx)

## 92-93 93-93.5 93.5-94 94+   
## 8992 11513 14901 5782

cons.conf.idx

summary(bank\_n$cons.conf.idx)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -50.8 -42.7 -41.8 -40.5 -36.4 -26.9

table(bank\_n$cons.conf.idx)

##   
## -50.8 -50 -49.5 -47.1 -46.2 -45.9 -42.7 -42 -41.8 -40.8 -40.4 -40.3 -40   
## 128 282 204 2458 5794 10 6685 3616 4374 715 67 311 212   
## -39.8 -38.3 -37.5 -36.4 -36.1 -34.8 -34.6 -33.6 -33 -31.4 -30.1 -29.8 -26.9   
## 229 233 303 7763 5175 264 174 178 172 770 357 267 447

hist(bank\_n$cons.conf.idx, col = "lightblue")

A screenshot of a cell phone

Description automatically generated

bank\_n$cons.conf.idx <- cut(bank\_n$cons.conf.idx,  
 breaks = c(-51, -45, -40, -35, max(bank\_n$cons.conf.idx)),  
 labels = c("(-50)-(-45)", "(-45)-(-40)", "(-40)-(-35)", "(-35)+"))  
summary(bank\_n$cons.conf.idx)

## (-50)-(-45) (-45)-(-40) (-40)-(-35) (-35)+   
## 8876 15980 13703 2629

euribor3m

summary(bank\_n$euribor3m)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.634 1.344 4.857 3.621 4.961 5.045

pl.euribor3m <- ggplot(bank\_n, aes(x=euribor3m)) + geom\_histogram(bins = 10, color = 'blue', aes(fill=..count..), alpha = 0.4) + xlab('euribor3m') + ylab('Count') + ggtitle('euribor3m distribution Plot')  
pl.euribor3m

A picture containing clock

Description automatically generated

bank\_n$euribor3m <- cut(bank\_n$euribor3m,  
 breaks = c(0,1, 2, 4, max(bank\_n$euribor3m)),  
 labels = c("0-1", "1-2", "2-4", "4+"))  
summary(bank\_n$euribor3m)

## 0-1 1-2 2-4 4+   
## 3908 9590 14 27676

nr.employed

summary(bank\_n$nr.employed)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 4964 5099 5191 5167 5228 5228

table(bank\_n$nr.employed)

##   
## 4963.6 4991.6 5008.7 5017.5 5023.5 5076.2 5099.1 5176.3 5191 5195.8 5228.1   
## 635 773 650 1071 172 1663 8534 10 7763 3683 16234

bank\_n$nr.employed <- as.factor(bank\_n$nr.employed)  
summary(bank\_n$nr.employed)

## 4963.6 4991.6 5008.7 5017.5 5023.5 5076.2 5099.1 5176.3 5191 5195.8 5228.1   
## 635 773 650 1071 172 1663 8534 10 7763 3683 16234

summary of df

summary(bank\_n)

## age job marital   
## 30-40 :16385 admin. :10422 divorced: 4612   
## 40-50 :10240 blue-collar: 9254 married :24928   
## 20-30 : 7243 technician : 6743 single :11568   
## 50-60 : 6270 services : 3969 unknown : 80   
## 60-70 : 488 management : 2924   
## 70-80 : 303 retired : 1720   
## (Other): 259 (Other) : 6156   
## education default housing loan   
## university.degree :12168 no :32588 no :18622 no :33950   
## high.school : 9515 unknown: 8597 unknown: 990 unknown: 990   
## basic.9y : 6045 yes : 3 yes :21576 yes : 6248   
## professional.course: 5243   
## basic.4y : 4176   
## basic.6y : 2292   
## (Other) : 1749   
## contact month day\_of\_week duration campaign   
## cellular :26144 may :13769 fri:7827 70-110 : 5952 1-2 :10570   
## telephone:15044 jul : 7174 mon:8514 110-150: 5521 2-3 : 5341   
## aug : 6178 thu:8623 350-600: 5452 3-5 : 4250   
## jun : 5318 tue:8090 250-350: 5325 5-10 : 2516   
## nov : 4101 wed:8134 30-70 : 5198 10-15 : 514   
## apr : 2632 (Other):13736 (Other): 355   
## (Other): 2016 NA's : 4 NA's :17642   
## pdays previous poutcome emp.var.rate   
## 999 :39673 0 :35563 failure : 4252 1.4 :16234   
## 2 : 439 1 : 4561 nonexistent:35563 -1.8 : 9184   
## 5 : 412 2 : 754 success : 1373 1.1 : 7763   
## 10-15 : 166 3 : 216 -0.1 : 3683   
## 3 : 118 4 : 70 -2.9 : 1663   
## (Other): 365 5 : 18 -3.4 : 1071   
## NA's : 15 (Other): 6 (Other): 1590   
## cons.price.idx cons.conf.idx euribor3m nr.employed y   
## 92-93 : 8992 (-50)-(-45): 8876 0-1: 3908 5228.1 :16234 Yes: 4640   
## 93-93.5:11513 (-45)-(-40):15980 1-2: 9590 5099.1 : 8534 No :36548   
## 93.5-94:14901 (-40)-(-35):13703 2-4: 14 5191 : 7763   
## 94+ : 5782 (-35)+ : 2629 4+ :27676 5195.8 : 3683   
## 5076.2 : 1663   
## 5017.5 : 1071   
## (Other): 2240

#### 2) removing NAs after transformation

bank\_n <- na.omit(bank\_n)  
summary(bank\_n)

## age job marital education   
## 30-40 :9376 admin. :6012 divorced: 2692 university.degree :6896   
## 40-50 :5885 blue-collar:5323 married :14252 high.school :5436   
## 20-30 :4122 technician :3857 single : 6551 basic.9y :3488   
## 50-60 :3649 services :2315 unknown : 44 professional.course:3046   
## 60-70 : 243 management :1642 basic.4y :2333   
## 70-80 : 131 retired : 932 basic.6y :1339   
## (Other): 133 (Other) :3458 (Other) :1001   
## default housing loan contact   
## no :18422 no :10754 no :19306 cellular :14380   
## unknown: 5116 unknown: 575 unknown: 575 telephone: 9159   
## yes : 1 yes :12210 yes : 3658   
##   
##   
##   
##   
## month day\_of\_week duration campaign pdays   
## may :8001 fri:4811 70-110 :3451 1-2 :10568 999 :22824   
## jul :4500 mon:5302 30-70 :3218 2-3 : 5338 5 : 209   
## aug :3579 thu:4593 110-150:3066 3-5 : 4249 2 : 205   
## jun :3275 tue:4379 350-600:3057 5-10 : 2515 10-15 : 80   
## nov :1981 wed:4454 250-350:2933 10-15: 514 3 : 61   
## apr :1306 190-250:2870 15-20: 198 8 : 29   
## (Other): 897 (Other):4944 20+ : 157 (Other): 131   
## previous poutcome emp.var.rate cons.price.idx  
## 0 :20768 failure : 2125 1.4 :10263 92-93 :4730   
## 1 : 2258 nonexistent:20768 -1.8 : 4978 93-93.5:6228   
## 2 : 362 success : 646 1.1 : 4622 93.5-94:9056   
## 3 : 107 -0.1 : 1743 94+ :3525   
## 4 : 34 -2.9 : 740   
## 5 : 7 -3.4 : 443   
## (Other): 3 (Other): 750   
## cons.conf.idx euribor3m nr.employed y   
## (-50)-(-45):4819 0-1: 1779 5228.1 :10263 Yes: 2338   
## (-45)-(-40):9433 1-2: 5132 5099.1 : 4658 No :21201   
## (-40)-(-35):8096 2-4: 0 5191 : 4622   
## (-35)+ :1191 4+ :16628 5195.8 : 1743   
## 5076.2 : 740   
## 5017.5 : 443   
## (Other): 1070

#### 3) creating training and testing dataset from exisitng sample

set.seed(1)  
indx <- caret::createDataPartition(bank\_n$y, p = 0.8, list = FALSE)

creating lables for test and training data sets

bank\_train\_labels <- bank\_n[indx,21]  
bank\_test\_labels <- bank\_n[- indx,21]

removing dependent variable

bank\_n <- bank\_n[,-21]  
summary(bank\_n)

## age job marital education   
## 30-40 :9376 admin. :6012 divorced: 2692 university.degree :6896   
## 40-50 :5885 blue-collar:5323 married :14252 high.school :5436   
## 20-30 :4122 technician :3857 single : 6551 basic.9y :3488   
## 50-60 :3649 services :2315 unknown : 44 professional.course:3046   
## 60-70 : 243 management :1642 basic.4y :2333   
## 70-80 : 131 retired : 932 basic.6y :1339   
## (Other): 133 (Other) :3458 (Other) :1001   
## default housing loan contact   
## no :18422 no :10754 no :19306 cellular :14380   
## unknown: 5116 unknown: 575 unknown: 575 telephone: 9159   
## yes : 1 yes :12210 yes : 3658   
##   
##   
##   
##   
## month day\_of\_week duration campaign pdays   
## may :8001 fri:4811 70-110 :3451 1-2 :10568 999 :22824   
## jul :4500 mon:5302 30-70 :3218 2-3 : 5338 5 : 209   
## aug :3579 thu:4593 110-150:3066 3-5 : 4249 2 : 205   
## jun :3275 tue:4379 350-600:3057 5-10 : 2515 10-15 : 80   
## nov :1981 wed:4454 250-350:2933 10-15: 514 3 : 61   
## apr :1306 190-250:2870 15-20: 198 8 : 29   
## (Other): 897 (Other):4944 20+ : 157 (Other): 131   
## previous poutcome emp.var.rate cons.price.idx  
## 0 :20768 failure : 2125 1.4 :10263 92-93 :4730   
## 1 : 2258 nonexistent:20768 -1.8 : 4978 93-93.5:6228   
## 2 : 362 success : 646 1.1 : 4622 93.5-94:9056   
## 3 : 107 -0.1 : 1743 94+ :3525   
## 4 : 34 -2.9 : 740   
## 5 : 7 -3.4 : 443   
## (Other): 3 (Other): 750   
## cons.conf.idx euribor3m nr.employed   
## (-50)-(-45):4819 0-1: 1779 5228.1 :10263   
## (-45)-(-40):9433 1-2: 5132 5099.1 : 4658   
## (-40)-(-35):8096 2-4: 0 5191 : 4622   
## (-35)+ :1191 4+ :16628 5195.8 : 1743   
## 5076.2 : 740   
## 5017.5 : 443   
## (Other): 1070

creating training and testing data

bank\_train <- bank\_n[indx,]  
bank\_test <- bank\_n[- indx,]

### Step 4 : Train the model

bank\_classifier <- naiveBayes(bank\_train, bank\_train\_labels)

### Step 5 : Evaluate the model

bank\_test\_pred <- predict(bank\_classifier, bank\_test)  
caret::confusionMatrix(bank\_test\_pred, bank\_test\_labels)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 277 371  
## No 190 3869  
##   
## Accuracy : 0.8808   
## 95% CI : (0.8712, 0.8899)  
## No Information Rate : 0.9008   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.4313   
##   
## Mcnemar's Test P-Value : 2.97e-14   
##   
## Sensitivity : 0.59315   
## Specificity : 0.91250   
## Pos Pred Value : 0.42747   
## Neg Pred Value : 0.95319   
## Prevalence : 0.09921   
## Detection Rate : 0.05885   
## Detection Prevalence : 0.13767   
## Balanced Accuracy : 0.75282   
##   
## 'Positive' Class : Yes   
##

### Step 6 : Improving the model

bank\_classifier2 <- naiveBayes(bank\_train, bank\_train\_labels, laplace = 1)  
bank\_test\_pred2 <- predict(bank\_classifier2, bank\_test)  
caret::confusionMatrix(bank\_test\_pred2, bank\_test\_labels)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 277 367  
## No 190 3873  
##   
## Accuracy : 0.8817   
## 95% CI : (0.8721, 0.8908)  
## No Information Rate : 0.9008   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.4335   
##   
## Mcnemar's Test P-Value : 8.827e-14   
##   
## Sensitivity : 0.59315   
## Specificity : 0.91344   
## Pos Pred Value : 0.43012   
## Neg Pred Value : 0.95324   
## Prevalence : 0.09921   
## Detection Rate : 0.05885   
## Detection Prevalence : 0.13682   
## Balanced Accuracy : 0.75330   
##   
## 'Positive' Class : Yes   
##

confusionMatrix(bank\_test\_pred2, bank\_test\_labels, mode="prec\_recall")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 277 367  
## No 190 3873  
##   
## Accuracy : 0.8817   
## 95% CI : (0.8721, 0.8908)  
## No Information Rate : 0.9008   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.4335   
##   
## Mcnemar's Test P-Value : 8.827e-14   
##   
## Precision : 0.43012   
## Recall : 0.59315   
## F1 : 0.49865   
## Prevalence : 0.09921   
## Detection Rate : 0.05885   
## Detection Prevalence : 0.13682   
## Balanced Accuracy : 0.75330   
##   
## 'Positive' Class : Yes   
##

## Part 3 - Decision Tree

### Step 1 : Collecting Data

Same as of Part 1

### Step 2 : Exploring, preprocessing and cleaning the data

Primary setup

knitr::opts\_knit$set(root.dir = '/Users/sobil/Documents/MSC/Sem 1/Data Mining & Machine Learning/Project/Bank/')  
remove(list = ls())  
set.seed(1)

loading all the libraries required

library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

library(C50)  
library(RWeka)

rest same as part 1

source("bank\_import\_primaryExplore.R")

## 'data.frame': 41188 obs. of 21 variables:  
## $ age : int 56 57 37 40 56 45 59 41 24 25 ...  
## $ job : Factor w/ 12 levels "admin.","blue-collar",..: 4 8 8 1 8 8 1 2 10 8 ...  
## $ marital : Factor w/ 4 levels "divorced","married",..: 2 2 2 2 2 2 2 2 3 3 ...  
## $ education : Factor w/ 8 levels "basic.4y","basic.6y",..: 1 4 4 2 4 3 6 8 6 4 ...  
## $ default : Factor w/ 3 levels "no","unknown",..: 1 2 1 1 1 2 1 2 1 1 ...  
## $ housing : Factor w/ 3 levels "no","unknown",..: 1 1 3 1 1 1 1 1 3 3 ...  
## $ loan : Factor w/ 3 levels "no","unknown",..: 1 1 1 1 3 1 1 1 1 1 ...  
## $ contact : Factor w/ 2 levels "cellular","telephone": 2 2 2 2 2 2 2 2 2 2 ...  
## $ month : Factor w/ 10 levels "apr","aug","dec",..: 7 7 7 7 7 7 7 7 7 7 ...  
## $ day\_of\_week : Factor w/ 5 levels "fri","mon","thu",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ duration : int 261 149 226 151 307 198 139 217 380 50 ...  
## $ campaign : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ pdays : int 999 999 999 999 999 999 999 999 999 999 ...  
## $ previous : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ poutcome : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ emp.var.rate : num 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 ...  
## $ cons.price.idx: num 94 94 94 94 94 ...  
## $ cons.conf.idx : num -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 ...  
## $ euribor3m : num 4.86 4.86 4.86 4.86 4.86 ...  
## $ nr.employed : num 5191 5191 5191 5191 5191 ...  
## $ y : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...  
##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 41188   
##   
##   
## | no | yes |   
## |-----------|-----------|  
## | 36548 | 4640 |   
## | 0.887 | 0.113 |   
## |-----------|-----------|  
##   
##   
##   
##

### Step 3 - Data transformation & preparation

creating training and testing dataset from exisitng sample

set.seed(1)  
indx <- createDataPartition(bank$y, p = 0.8, list = FALSE)  
  
bank\_train <- bank[indx,]  
bank\_test <- bank[- indx,]  
  
prop.table(table(bank\_train$y))

##   
## Yes No   
## 0.1126521 0.8873479

prop.table(table(bank\_test$y))

##   
## Yes No   
## 0.1126624 0.8873376

### Step 4 : Train the model

bank\_model <- C50::C5.0(bank\_train[-21],bank\_train$y)

### Step 5 : Evaluate the model

bank\_model

##   
## Call:  
## C5.0.default(x = bank\_train[-21], y = bank\_train$y)  
##   
## Classification Tree  
## Number of samples: 32951   
## Number of predictors: 20   
##   
## Tree size: 172   
##   
## Non-standard options: attempt to group attributes

bank\_predict <- predict(object = bank\_model, newdata = bank\_test)  
caret::confusionMatrix(bank\_predict, bank\_test$y)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 519 262  
## No 409 7047  
##   
## Accuracy : 0.9185   
## 95% CI : (0.9124, 0.9244)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5623   
##   
## Mcnemar's Test P-Value : 1.738e-08   
##   
## Sensitivity : 0.55927   
## Specificity : 0.96415   
## Pos Pred Value : 0.66453   
## Neg Pred Value : 0.94514   
## Prevalence : 0.11266   
## Detection Rate : 0.06301   
## Detection Prevalence : 0.09482   
## Balanced Accuracy : 0.76171   
##   
## 'Positive' Class : Yes   
##

### Step 6 : Improving model performance

#### 1) Method 1 - Boosting trials = 10

bank\_boost10 <- C50::C5.0(bank\_train[-21], bank\_train$y, trials = 10)  
bank\_boost10

##   
## Call:  
## C5.0.default(x = bank\_train[-21], y = bank\_train$y, trials = 10)  
##   
## Classification Tree  
## Number of samples: 32951   
## Number of predictors: 20   
##   
## Number of boosting iterations: 10   
## Average tree size: 167.1   
##   
## Non-standard options: attempt to group attributes

bank\_boost\_predict10 <- predict(object = bank\_boost10, newdata = bank\_test)  
caret::confusionMatrix(bank\_boost\_predict10, bank\_test$y)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 510 287  
## No 418 7022  
##   
## Accuracy : 0.9144   
## 95% CI : (0.9082, 0.9204)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 4.065e-16   
##   
## Kappa : 0.5438   
##   
## Mcnemar's Test P-Value : 9.777e-07   
##   
## Sensitivity : 0.54957   
## Specificity : 0.96073   
## Pos Pred Value : 0.63990   
## Neg Pred Value : 0.94382   
## Prevalence : 0.11266   
## Detection Rate : 0.06192   
## Detection Prevalence : 0.09676   
## Balanced Accuracy : 0.75515   
##   
## 'Positive' Class : Yes   
##

#### 2) Method 2 - Boosting trials = 5

bank\_boost5 <- C50::C5.0(bank\_train[-21], bank\_train$y, trials = 5)  
bank\_boost5

##   
## Call:  
## C5.0.default(x = bank\_train[-21], y = bank\_train$y, trials = 5)  
##   
## Classification Tree  
## Number of samples: 32951   
## Number of predictors: 20   
##   
## Number of boosting iterations: 5   
## Average tree size: 129.4   
##   
## Non-standard options: attempt to group attributes

bank\_boost\_predict5 <- predict(object = bank\_boost5, newdata = bank\_test)  
caret::confusionMatrix(bank\_boost\_predict5, bank\_test$y)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 521 305  
## No 407 7004  
##   
## Accuracy : 0.9136   
## 95% CI : (0.9073, 0.9195)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 3.393e-15   
##   
## Kappa : 0.5459   
##   
## Mcnemar's Test P-Value : 0.0001536   
##   
## Sensitivity : 0.56142   
## Specificity : 0.95827   
## Pos Pred Value : 0.63075   
## Neg Pred Value : 0.94508   
## Prevalence : 0.11266   
## Detection Rate : 0.06325   
## Detection Prevalence : 0.10028   
## Balanced Accuracy : 0.75985   
##   
## 'Positive' Class : Yes   
##

#### 3) Method 3 - Adding Cost matrix

creating cost matrix - “yes” more costlier than others

mtr\_dim <- list(c("no", "yes"), c("no","yes"))  
names(mtr\_dim) <- c("predict","actual")  
mtr\_dim

## $predict  
## [1] "no" "yes"  
##   
## $actual  
## [1] "no" "yes"

err\_cst <- matrix(c(0,1,0,4), nrow = 2, dimnames = mtr\_dim)  
err\_cst

## actual  
## predict no yes  
## no 0 0  
## yes 1 4

Model Costs

bank\_cost <- C50::C5.0(bank\_train[-21], bank\_train$y, costs = err\_cst)  
bank\_cost\_predict <- predict(object = bank\_cost, newdata = bank\_test)  
caret::confusionMatrix(bank\_cost\_predict, bank\_test$y)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 519 262  
## No 409 7047  
##   
## Accuracy : 0.9185   
## 95% CI : (0.9124, 0.9244)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5623   
##   
## Mcnemar's Test P-Value : 1.738e-08   
##   
## Sensitivity : 0.55927   
## Specificity : 0.96415   
## Pos Pred Value : 0.66453   
## Neg Pred Value : 0.94514   
## Prevalence : 0.11266   
## Detection Rate : 0.06301   
## Detection Prevalence : 0.09482   
## Balanced Accuracy : 0.76171   
##   
## 'Positive' Class : Yes   
##

#### 4) Method 4 - 1R algo model

bank\_1R <- OneR(y ~ ., data = bank\_train)  
bank\_1R

## duration:  
## < 457.5 -> No  
## < 458.5 -> Yes  
## < 525.5 -> No  
## < 526.5 -> Yes  
## < 554.5 -> No  
## < 555.5 -> Yes  
## < 562.5 -> No  
## < 563.5 -> Yes  
## < 582.5 -> No  
## < 583.5 -> Yes  
## < 606.5 -> No  
## < 608.5 -> Yes  
## < 617.5 -> No  
## < 618.5 -> Yes  
## < 622.5 -> No  
## < 623.5 -> Yes  
## < 644.5 -> No  
## < 646.5 -> Yes  
## < 648.5 -> No  
## < 651.5 -> Yes  
## < 653.5 -> No  
## < 655.5 -> Yes  
## < 657.5 -> No  
## < 658.5 -> Yes  
## < 661.5 -> No  
## < 664.5 -> Yes  
## < 666.5 -> No  
## < 667.5 -> Yes  
## < 668.5 -> No  
## < 671.5 -> Yes  
## < 676.5 -> No  
## < 679.5 -> Yes  
## < 699.5 -> No  
## < 701.5 -> Yes  
## < 702.5 -> No  
## < 704.5 -> Yes  
## < 706.5 -> No  
## < 712.5 -> Yes  
## < 714.5 -> No  
## < 716.5 -> Yes  
## < 718.5 -> No  
## < 721.5 -> Yes  
## < 730.5 -> No  
## < 734.5 -> Yes  
## < 738.5 -> No  
## < 741.5 -> Yes  
## < 744.5 -> No  
## < 746.5 -> Yes  
## < 761.5 -> No  
## < 763.5 -> Yes  
## < 766.5 -> No  
## < 768.5 -> Yes  
## < 780.5 -> No  
## < 788.5 -> Yes  
## < 800.5 -> No  
## < 803.5 -> Yes  
## < 807.5 -> No  
## < 809.5 -> Yes  
## < 811.5 -> No  
## < 813.5 -> Yes  
## < 816.5 -> No  
## < 820.5 -> Yes  
## < 827.5 -> No  
## < 832.5 -> Yes  
## < 835.5 -> No  
## < 842.0 -> Yes  
## < 847.5 -> No  
## < 850.5 -> Yes  
## < 853.5 -> No  
## < 883.5 -> Yes  
## < 887.0 -> No  
## < 897.5 -> Yes  
## < 899.5 -> No  
## < 904.5 -> Yes  
## < 907.5 -> No  
## < 914.0 -> Yes  
## < 919.5 -> No  
## < 924.5 -> Yes  
## < 930.5 -> No  
## < 971.5 -> Yes  
## < 977.5 -> No  
## < 985.5 -> Yes  
## < 990.5 -> No  
## < 1009.5 -> Yes  
## < 1014.5 -> No  
## < 1082.5 -> Yes  
## < 1089.5 -> No  
## < 1093.5 -> Yes  
## < 1100.5 -> No  
## < 1155.0 -> Yes  
## < 1168.0 -> No  
## < 1179.0 -> Yes  
## < 1198.0 -> No  
## < 1219.0 -> Yes  
## < 1225.5 -> No  
## < 1270.0 -> Yes  
## < 1292.5 -> No  
## < 1375.0 -> Yes  
## < 1390.5 -> No  
## < 1424.5 -> Yes  
## < 1438.5 -> No  
## < 1546.5 -> Yes  
## < 1592.0 -> No  
## < 1737.0 -> Yes  
## < 1805.5 -> No  
## < 1869.5 -> Yes  
## < 1960.0 -> No  
## < 2781.0 -> Yes  
## < 3570.0 -> No  
## >= 3570.0 -> Yes  
## (29647/32951 instances correct)

bank\_predict\_1R <- predict(object = bank\_1R, newdata = bank\_test)  
caret::confusionMatrix(bank\_predict\_1R, bank\_test$y)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 196 164  
## No 732 7145  
##   
## Accuracy : 0.8912   
## 95% CI : (0.8843, 0.8979)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.136   
##   
## Kappa : 0.2576   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.21121   
## Specificity : 0.97756   
## Pos Pred Value : 0.54444   
## Neg Pred Value : 0.90707   
## Prevalence : 0.11266   
## Detection Rate : 0.02380   
## Detection Prevalence : 0.04371   
## Balanced Accuracy : 0.59438   
##   
## 'Positive' Class : Yes   
##

#### 5) Method 5 - RIPPER algo model

bank\_JRip <- JRip(y ~ . , data = bank\_train)  
bank\_JRip

## JRIP rules:  
## ===========  
##   
## (nr.employed <= 5076.2) and (duration >= 190) and (pdays <= 15) => y=Yes (639.0/112.0)  
## (duration >= 854) => y=Yes (1155.0/485.0)  
## (duration >= 381) and (euribor3m <= 1.4) and (cons.price.idx >= 92.963) => y=Yes (363.0/105.0)  
## (nr.employed <= 5076.2) and (duration >= 251) => y=Yes (801.0/354.0)  
## (duration >= 608) and (contact = cellular) and (month = may) => y=Yes (213.0/93.0)  
## (euribor3m >= 4.866) and (duration >= 758) and (age <= 37) => y=Yes (76.0/33.0)  
## (euribor3m <= 3.053) and (duration >= 148) and (nr.employed <= 5076.2) and (pdays <= 15) => y=Yes (140.0/50.0)  
## => y=No (29564.0/1557.0)  
##   
## Number of Rules : 8

bank\_predict\_JRip <- predict(object = bank\_JRip, newdata = bank\_test)  
caret::confusionMatrix(bank\_predict\_JRip, bank\_test$y)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 543 300  
## No 385 7009  
##   
## Accuracy : 0.9168   
## 95% CI : (0.9107, 0.9227)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2e-16   
##   
## Kappa : 0.5667   
##   
## Mcnemar's Test P-Value : 0.00133   
##   
## Sensitivity : 0.58513   
## Specificity : 0.95895   
## Pos Pred Value : 0.64413   
## Neg Pred Value : 0.94793   
## Prevalence : 0.11266   
## Detection Rate : 0.06592   
## Detection Prevalence : 0.10234   
## Balanced Accuracy : 0.77204   
##   
## 'Positive' Class : Yes   
##

#kappa = 0.5667, Sensitivity = 0.58513, Specificity = 0.95895 ====> BEST

caret::confusionMatrix(bank\_predict\_JRip, bank\_test$y, mode="prec\_recall")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 543 300  
## No 385 7009  
##   
## Accuracy : 0.9168   
## 95% CI : (0.9107, 0.9227)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2e-16   
##   
## Kappa : 0.5667   
##   
## Mcnemar's Test P-Value : 0.00133   
##   
## Precision : 0.64413   
## Recall : 0.58513   
## F1 : 0.61321   
## Prevalence : 0.11266   
## Detection Rate : 0.06592   
## Detection Prevalence : 0.10234   
## Balanced Accuracy : 0.77204   
##   
## 'Positive' Class : Yes   
##

## Part 4 - SVM model

### Step 1 : Collecting Data

Same as of Part 1

### Step 2 : Exploring, preprocessing and cleaning the data

Primary setup

knitr::opts\_knit$set(root.dir = normalizePath('D:\\Sobil'))  
remove(list = ls())  
set.seed(1)

loading all the libraries required

library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

library(kernlab)

##   
## Attaching package: 'kernlab'

## The following object is masked from 'package:ggplot2':  
##   
## alpha

rest same as part 1

source("bank\_import\_primaryExplore.R")

## 'data.frame': 41188 obs. of 21 variables:  
## $ age : int 56 57 37 40 56 45 59 41 24 25 ...  
## $ job : Factor w/ 12 levels "admin.","blue-collar",..: 4 8 8 1 8 8 1 2 10 8 ...  
## $ marital : Factor w/ 4 levels "divorced","married",..: 2 2 2 2 2 2 2 2 3 3 ...  
## $ education : Factor w/ 8 levels "basic.4y","basic.6y",..: 1 4 4 2 4 3 6 8 6 4 ...  
## $ default : Factor w/ 3 levels "no","unknown",..: 1 2 1 1 1 2 1 2 1 1 ...  
## $ housing : Factor w/ 3 levels "no","unknown",..: 1 1 3 1 1 1 1 1 3 3 ...  
## $ loan : Factor w/ 3 levels "no","unknown",..: 1 1 1 1 3 1 1 1 1 1 ...  
## $ contact : Factor w/ 2 levels "cellular","telephone": 2 2 2 2 2 2 2 2 2 2 ...  
## $ month : Factor w/ 10 levels "apr","aug","dec",..: 7 7 7 7 7 7 7 7 7 7 ...  
## $ day\_of\_week : Factor w/ 5 levels "fri","mon","thu",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ duration : int 261 149 226 151 307 198 139 217 380 50 ...  
## $ campaign : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ pdays : int 999 999 999 999 999 999 999 999 999 999 ...  
## $ previous : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ poutcome : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ emp.var.rate : num 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 ...  
## $ cons.price.idx: num 94 94 94 94 94 ...  
## $ cons.conf.idx : num -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 ...  
## $ euribor3m : num 4.86 4.86 4.86 4.86 4.86 ...  
## $ nr.employed : num 5191 5191 5191 5191 5191 ...  
## $ y : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...  
##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 41188   
##   
##   
## | no | yes |   
## |-----------|-----------|  
## | 36548 | 4640 |   
## | 0.887 | 0.113 |   
## |-----------|-----------|  
##   
##   
##   
##

### Step 3 - Data transformation & preparation

creating training and testing dataset from exisitng sample

set.seed(1)  
indx <- createDataPartition(bank$y, p = 0.8, list = FALSE)  
  
bank\_train <- bank[indx,]  
bank\_test <- bank[- indx,]  
  
prop.table(table(bank\_train$y))

##   
## Yes No   
## 0.1126521 0.8873479

prop.table(table(bank\_test$y))

##   
## Yes No   
## 0.1126624 0.8873376

### Step 4 : Training a model on the data

bank\_svm <- ksvm(y ~ ., data = bank\_train, kernel = "vanilladot")

## Setting default kernel parameters

bank\_svm

## Support Vector Machine object of class "ksvm"   
##   
## SV type: C-svc (classification)   
## parameter : cost C = 1   
##   
## Linear (vanilla) kernel function.   
##   
## Number of Support Vectors : 6674   
##   
## Objective Function Value : -6639.711   
## Training error : 0.096355

### Step 5 : Evaluating model performance

bank\_predict\_svm <- predict(object = bank\_svm, newdata = bank\_test)  
caret::confusionMatrix(bank\_predict\_svm, bank\_test$y) # kappa = 0.4051, Sensitivity = 0.33297, Specificity = 0.98085

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 309 140  
## No 619 7169  
##   
## Accuracy : 0.9079   
## 95% CI : (0.9014, 0.914)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 7.895e-10   
##   
## Kappa : 0.4051   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.33297   
## Specificity : 0.98085   
## Pos Pred Value : 0.68820   
## Neg Pred Value : 0.92052   
## Prevalence : 0.11266   
## Detection Rate : 0.03751   
## Detection Prevalence : 0.05451   
## Balanced Accuracy : 0.65691   
##   
## 'Positive' Class : Yes   
##

### Step 6 : Improving model performance

#### 1) Method 1 : kernel = rbfdot

bank\_svm.rbf <- ksvm(y ~ . , data = bank\_train, kernel = "rbfdot")  
bank\_predict\_svm.rbf <- predict(object = bank\_svm.rbf, newdata = bank\_test)  
caret::confusionMatrix(bank\_predict\_svm.rbf, bank\_test$y) # kappa = 0.4907, Sensitivity = 0.42996, Specificity = 0.97715

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 399 167  
## No 529 7142  
##   
## Accuracy : 0.9155   
## 95% CI : (0.9093, 0.9214)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.4907   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.42996   
## Specificity : 0.97715   
## Pos Pred Value : 0.70495   
## Neg Pred Value : 0.93104   
## Prevalence : 0.11266   
## Detection Rate : 0.04844   
## Detection Prevalence : 0.06871   
## Balanced Accuracy : 0.70355   
##   
## 'Positive' Class : Yes   
##

#### Method 2 : kernel = polydot

bank\_svm.poly <- ksvm(y ~ . , data = bank\_train, kernel = "polydot")

## Setting default kernel parameters

bank\_predict\_svm.poly <- predict(object = bank\_svm.poly, newdata = bank\_test)  
caret::confusionMatrix(bank\_predict\_svm.poly, bank\_test$y) # kappa = 0.4051, Sensitivity = 0.33297, Specificity = 0.98085

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 309 140  
## No 619 7169  
##   
## Accuracy : 0.9079   
## 95% CI : (0.9014, 0.914)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 7.895e-10   
##   
## Kappa : 0.4051   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.33297   
## Specificity : 0.98085   
## Pos Pred Value : 0.68820   
## Neg Pred Value : 0.92052   
## Prevalence : 0.11266   
## Detection Rate : 0.03751   
## Detection Prevalence : 0.05451   
## Balanced Accuracy : 0.65691   
##   
## 'Positive' Class : Yes   
##

#### Method 3 : kernel = tanhdot

bank\_svm.tanh <- ksvm(y ~ . , data = bank\_train, kernel = "tanhdot")

## Setting default kernel parameters

bank\_predict\_svm.tanh <- predict(object = bank\_svm.tanh, newdata = bank\_test)  
caret::confusionMatrix(bank\_predict\_svm.tanh, bank\_test$y) # kappa = 0.17, Sensitivity = 0.26078, Specificity = 0.90806

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 242 672  
## No 686 6637  
##   
## Accuracy : 0.8351   
## 95% CI : (0.8269, 0.8431)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 1.0000   
##   
## Kappa : 0.17   
##   
## Mcnemar's Test P-Value : 0.7243   
##   
## Sensitivity : 0.26078   
## Specificity : 0.90806   
## Pos Pred Value : 0.26477   
## Neg Pred Value : 0.90632   
## Prevalence : 0.11266   
## Detection Rate : 0.02938   
## Detection Prevalence : 0.11096   
## Balanced Accuracy : 0.58442   
##   
## 'Positive' Class : Yes   
##

from above 3 methods - trying adding cost param in kernel = rbfdot

#### Method 4 : kernel = rbfdot & C = 10

bank\_svm.rbf10 <- ksvm(y ~ . , data = bank\_train, kernel = "rbfdot", C = 10)  
bank\_predict\_svm.rbf10 <- predict(object = bank\_svm.rbf10, newdata = bank\_test)  
caret::confusionMatrix(bank\_predict\_svm.rbf10, bank\_test$y) # kappa = 0.4991, Sensitivity = 0.47737, Specificity = 0.96607

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 444 245  
## No 484 7064  
##   
## Accuracy : 0.9115   
## 95% CI : (0.9052, 0.9175)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 4.313e-13   
##   
## Kappa : 0.5013   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.47845   
## Specificity : 0.96648   
## Pos Pred Value : 0.64441   
## Neg Pred Value : 0.93588   
## Prevalence : 0.11266   
## Detection Rate : 0.05390   
## Detection Prevalence : 0.08365   
## Balanced Accuracy : 0.72246   
##   
## 'Positive' Class : Yes   
##

#### Method 5 : kernel = rbfdot & C = 50

bank\_svm.rbf50 <- ksvm(y ~ . , data = bank\_train, kernel = "rbfdot", C = 50)  
bank\_predict\_svm.rbf50 <- predict(object = bank\_svm.rbf50, newdata = bank\_test)  
caret::confusionMatrix(bank\_predict\_svm.rbf50, bank\_test$y) # kappa = 0.4848, Sensitivity = 0.49461, Specificity = 0.95622

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 459 320  
## No 469 6989  
##   
## Accuracy : 0.9042   
## 95% CI : (0.8977, 0.9105)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 4.026e-07   
##   
## Kappa : 0.4848   
##   
## Mcnemar's Test P-Value : 1.372e-07   
##   
## Sensitivity : 0.49461   
## Specificity : 0.95622   
## Pos Pred Value : 0.58922   
## Neg Pred Value : 0.93711   
## Prevalence : 0.11266   
## Detection Rate : 0.05572   
## Detection Prevalence : 0.09457   
## Balanced Accuracy : 0.72542   
##   
## 'Positive' Class : Yes   
##

#### Method 6 : kernel = rbfdot & C = 100

bank\_svm.rbf100 <- ksvm(y ~ . , data = bank\_train, kernel = "rbfdot", C = 100)  
bank\_predict\_svm.rbf100 <- predict(object = bank\_svm.rbf100, newdata = bank\_test)  
caret::confusionMatrix(bank\_predict\_svm.rbf100, bank\_test$y) # kappa = 0.4853, Sensitivity = 0.51509, Specificity = 0.95034 ==> BEST

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 478 362  
## No 450 6947  
##   
## Accuracy : 0.9014   
## 95% CI : (0.8948, 0.9078)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 2.088e-05   
##   
## Kappa : 0.4857   
##   
## Mcnemar's Test P-Value : 0.002265   
##   
## Sensitivity : 0.51509   
## Specificity : 0.95047   
## Pos Pred Value : 0.56905   
## Neg Pred Value : 0.93916   
## Prevalence : 0.11266   
## Detection Rate : 0.05803   
## Detection Prevalence : 0.10198   
## Balanced Accuracy : 0.73278   
##   
## 'Positive' Class : Yes   
##

confusionMatrix(bank\_predict\_svm.rbf100, bank\_test$y, mode="prec\_recall")

##Confusion Matrix and Statistics

##

## Reference

## Prediction Yes No

## Yes 478 362

## No 450 6947

##

## Accuracy : 0.9014

## 95% CI : (0.8948, 0.9078)

## No Information Rate : 0.8873

## P-Value [Acc > NIR] : 2.088e-05

## Kappa : 0.4857

##

## Mcnemar's Test P-Value : 0.002265

##

## Precision : 0.56905

## Recall : 0.51509

## F1 : 0.54072

## Prevalence : 0.11266

## Detection Rate : 0.05803

## Detection Prevalence : 0.10198

## Balanced Accuracy : 0.73278

##

## 'Positive' Class : Yes

##