Final Project

ALY 6015 – Intermediate Analytics

College of Professional Studies
Northeastern University - Vancouver

REPRESENTATIVES

Murtaza Vora.

Price prediction of Melbourne housing

The report is created based on the Melbourne housing condition such as area, rooms, bathroom and many other important aspects that a particular person or a customer looks for before buying a house or a property. This is a snapshot of a dataset taken from the Kaggle website. It was scraped from publicly available results posted every week from Domain.com.au. Our main aim or goal of this project is to find the factor affecting the price of a house using various methods such as multilinear regression, random forest, Lasso regression and K- nearest neighbor.

1. Introduction:

The dataset contains important information about the retail market of Melbourne. It is very important to look at the dataset and its variables and values as it has the past marketing trends of the Melbourne housing market.

1.1 Motives :

- **To find out the variables that actually have a significant influence on the price of the houses in Melbourne. Moreover, training a model to predict the price of houses using the variable.**
- **To Predict the Region where the house is situated using different variables.**

2. Materials and Methods:

2.1 Dataset

- The data set is all about the real estate market in Melbourne, Australia. It does have around 13,580 rows and 21 columns.
- The dataset includes variables such as
    - SellerG: Real Estate Agent
    - Date: Date sold
    - Distance: Distance from CBD
    - Regionname: General Region (West, North West, North, North east …etc)
    - Propertycount: Number of properties that exist in the suburb.
    - Bedroom2 : Scraped # of Bedrooms (from different source)
    - Bathroom: Number of Bathrooms
    - Car: Number of carspots
    - Landsize: Land Size
    - BuildingArea: Building Size
    - CouncilArea: Governing council for the area

Dataset:

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Date	Distan
	All	All	All	All	All	All	All	All	All
1	Abbotsford	85 Turner St	2	h	1480000	S	Biggin	3/12/2016	
2	Abbotsford	25 Bloomburg St	2	h	1035000	S	Biggin	4/02/2016	
3	Abbotsford	5 Charles St	3	h	1465000	SP	Biggin	4/03/2017	
4	Abbotsford	40 Federation La	3	h	850000	PI	Biggin	4/03/2017	
5	Abbotsford	55a Park St	4	h	1600000	VB	Nelson	4/06/2016	
		129							

Showing 1 to 10 of 13,580 entries

Dataset

2.2Methods :

2.2.1 **Multiple linear regression model**: Multiple linear regression is a regression model that estimates the relationship between a quantitative dependent variable and two or more independent variables using a straight line.

2.2.2 **Lasso Regression:** Lasso regression is also called Penalized regression method. This method is usually used in machine learning for the selection of the subset of variables. It provides greater prediction accuracy as compared to other regression models. Lasso Regularization helps to increase model interpretation.

2.2.3 **Random Forest:** Random Forest Regression is a supervised learning algorithm that uses an ensemble learning method for regression. The ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.

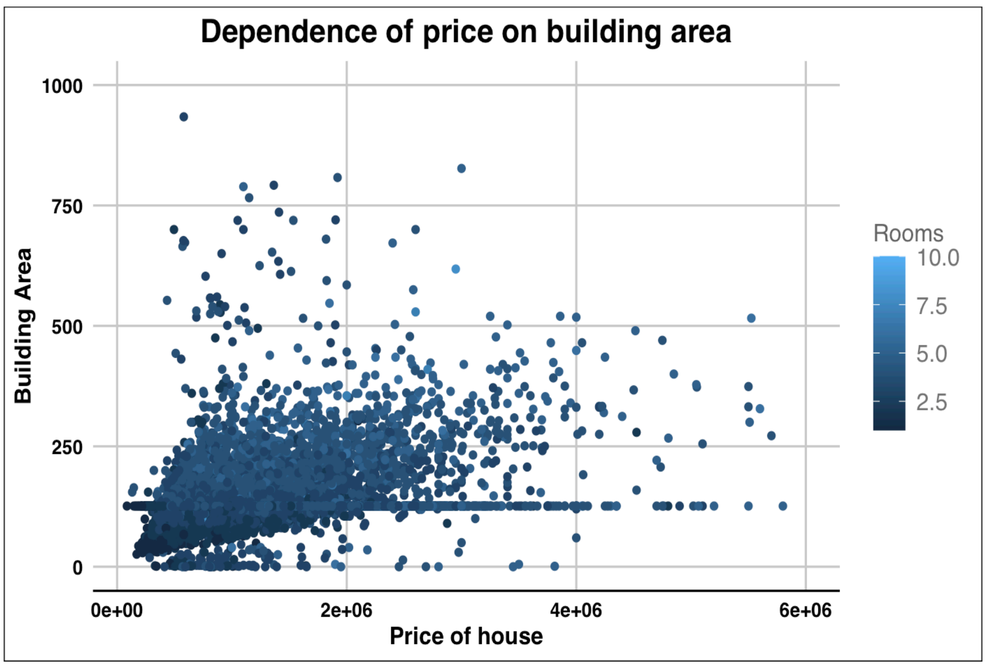
2.2.4 **K-nearest neighbor:** The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.

2.2.5 **Statistical software used :** Mainly R programming I used along with the libraries such as car, mass, Magritte, caret, glmnet, ggplot2, leaps, qqplotr, ggthemes, corplot and class.

3. Result:

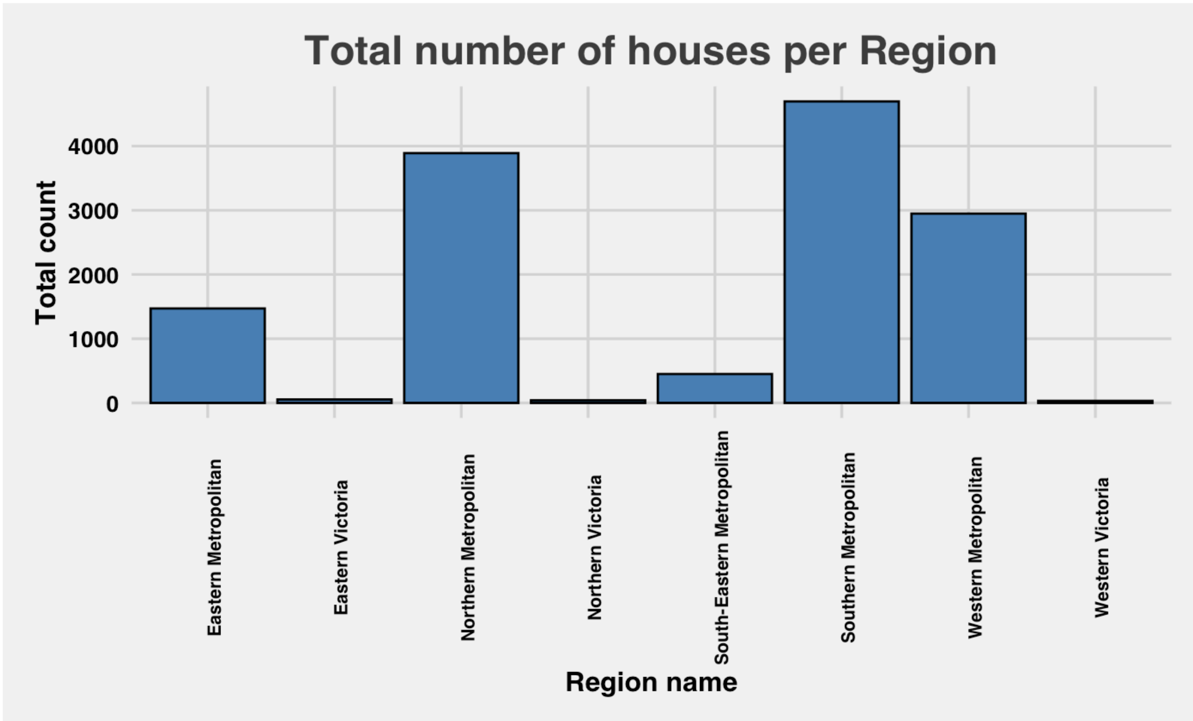
3.1 Exploratory Data Analysis (EDA) :

- The data set is all about the real estate market of Melbourne, Australia. It does have around 13,580 rows and 21 columns.
- The dataset includes Address, Type of Real estate, Suburb, Method of Selling, Rooms, Price, Real Estate Agent, Date of Sale, and distance from C.B.D. (the central business district of Melbourne).
- Also, variables such as Suburb, Address, Rooms, Type, Price, Method, SellerG, Date, Distance, Postcode, Bedroom2, Bathroom, Car, Landsize, BuildingArea, YearBuilt, CouncilArea, Lattitude, Longtitude, Regionname, and Propertycount are present.



Price vs Building area

- As the graph depicts, there is a general trend that if the area increases the price of the houses also increases. That is they are directly proportional to each other.



Total houses in a particular region.

- According to the graph western Victoria and eastern Victoria have the least number of houses in Melbourne.

3.2 Predicting price using different variables.

3.2.1 Creating model 1

```
Call:
lm(formula = Price ~ Distance + Bedroom2 + Rooms + Bathroom +
    Landsize + Car + Landsize + BuildingArea + YearBuilt + Lattitude +
    Longtitude, data = dataset)

Residuals:
     Min       1Q   Median       3Q      Max
-3261330  -258615   -66861   161646  8033898

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.822e+08  5.582e+06 -32.635  < 2e-16 ***
Distance     -3.954e+04  7.416e+02 -53.311  < 2e-16 ***
Bedroom2      3.914e+04  1.214e+04   3.224 0.001268 **
Rooms         2.317e+05  1.242e+04  18.659  < 2e-16 ***
Bathroom      2.113e+05  7.203e+03  29.330  < 2e-16 ***
Landsize      3.732e+00  9.656e-01   3.865 0.000112 ***
Car           6.489e+04  4.479e+03  14.489  < 2e-16 ***
BuildingArea  6.454e+01  9.892e+00   6.524 7.08e-11 ***
YearBuilt    -4.258e+03  1.399e+02 -30.444  < 2e-16 ***
Lattitude    -1.533e+06  5.249e+04 -29.206  < 2e-16 ***
Longtitude    9.161e+05  4.060e+04  22.563  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 446600 on 13507 degrees of freedom
  (62 observations deleted due to missingness)
Multiple R-squared:  0.5133,    Adjusted R-squared:  0.5129
F-statistic:  1424 on 10 and 13507 DF,  p-value: < 2.2e-16
```

Model1

- Creating our first models to test the different variables and their dependencies.
- Every variable is significantly influencing our dependent variable.

3.2.2 Outlier test: after training our model it is important to check for outliers and remove them from our that set

Description: df [10 x 3]

	rstudient <dbl>	unadjusted p-value <dbl>	Bonferroni p <dbl>
12095	18.214865	3.9339e-74	5.3178e-70
13246	-14.037936	9.1334e-45	1.2347e-40
9576	12.002235	3.4583e-33	4.6749e-29
7693	11.166634	5.9389e-29	8.0282e-25
6373	10.445384	1.5383e-25	2.0795e-21
12558	9.412930	4.8250e-21	6.5224e-17
3581	8.296246	1.0745e-16	1.4525e-12
5632	8.213158	2.1545e-16	2.9124e-12
6341	8.156817	3.4397e-16	4.6498e-12
3115	7.941519	1.9972e-15	2.6998e-11

Outlier test result

- As we can observe that our dataset has some outliers it is important to remove these outliers. Because it is highly influencing our model.

3.2.3 Multicollinearity:

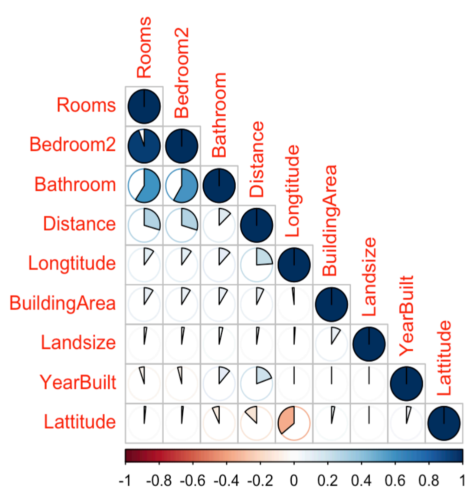
```
```{r}
vif(model1)
```
```

```
    Distance     Bedroom2        Rooms     Bathroom     Landsize          Car BuildingArea    YearBuilt    Lattitude   Longtitude 
    1.280842     9.335054     9.557256     1.685477     1.010357     1.260031     1.025029     1.113335     1.176146     1.208623 
```

```
```{r}
vif(model1)%>%sqrt>2
```
```

```
    Distance     Bedroom2        Rooms     Bathroom     Landsize          Car BuildingArea    YearBuilt    Lattitude   Longtitude 
       FALSE         TRUE         TRUE        FALSE        FALSE        FALSE        FALSE        FALSE        FALSE        FALSE 
```

Multicollinearity



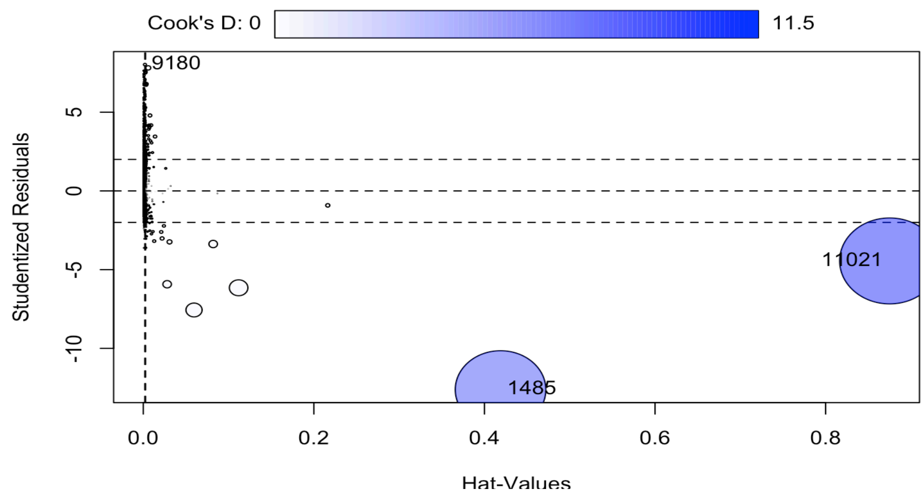
Correlation Plot

- As we test for correlation between the independent variable. It is found that "Rooms" and "Bedrooms" have a high positive correlation between them.

3.2.4 Influence plot:

	StudRes <dbl>	Hat <dbl>	CookD <dbl>
1485	-12.621104	0.418847619	9.4561714
9180	8.022105	0.001926629	0.0103038
11021	-4.446084	0.875024527	11.5177074

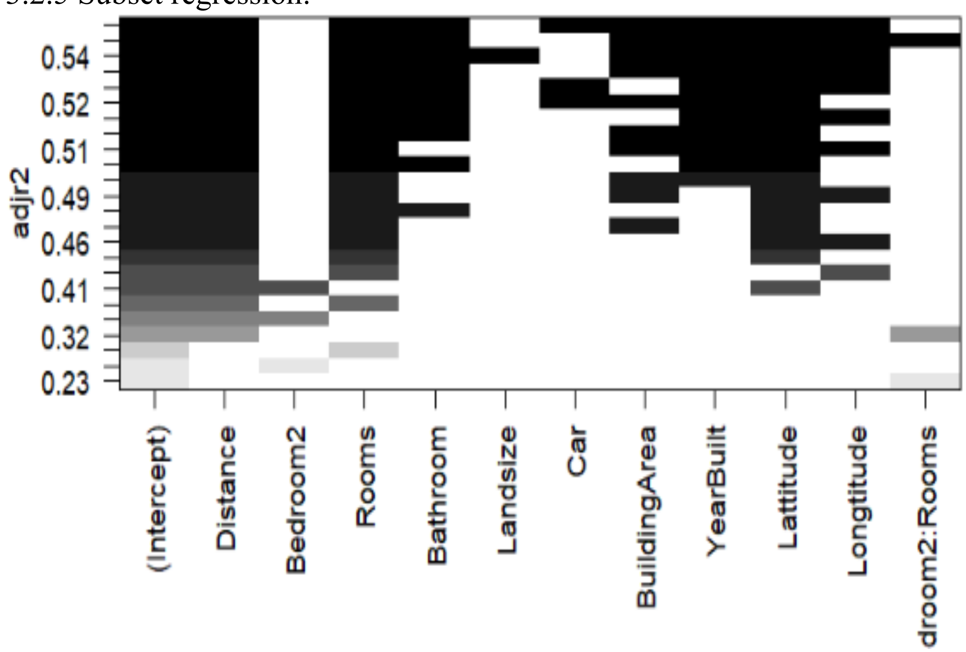
Influential points



Influential graph

- As we can see that we have high influence points thus it is important to remove all the influence points so that our model is affected by them.

3.2.5 Subset regression:



Subset regression.

- Regsubsets are used to select the variable that has very less influence on our dependent variable.
- From the leaps plot it is clear that we should remove bedroom2, land size, and bedroom: rooms.

3.2.6 Model 2

- A new model has been created without the “Bedroom2” and “Landsize” variables.

```
Call:
lm(formula = Price ~ Distance + Rooms + Bathroom + Car + BuildingArea +
    YearBuilt + Lattitude + Longtitude, data = dataset3)

Residuals:
     Min       1Q   Median       3Q      Max
-3540723  -247869   -63390   157905  3196088

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.774e+08  5.282e+06  -33.58   <2e-16 ***
Distance     -3.882e+04  7.007e+02  -55.40   <2e-16 ***
Rooms         2.421e+05  5.351e+03   45.24   <2e-16 ***
Bathroom      1.820e+05  6.908e+03   26.34   <2e-16 ***
Car           6.109e+04  4.239e+03   14.41   <2e-16 ***
BuildingArea  1.328e+03  5.991e+01   22.16   <2e-16 ***
YearBuilt    -4.143e+03  1.325e+02  -31.27   <2e-16 ***
Lattitude    -1.486e+06  4.964e+04  -29.93   <2e-16 ***
Longtitude    8.934e+05  3.841e+04   23.26   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 422300 on 13495 degrees of freedom
  (62 observations deleted due to missingness)
Multiple R-squared:  0.5417,    Adjusted R-squared:  0.5414
F-statistic:  1994 on 8 and 13495 DF,  p-value: < 2.2e-16
```

Model2

3.2.7 Anova:

```
Analysis of Deviance Table

Model 1: Price ~ Distance + Bedroom2 + Rooms + Bathroom + Landsize + Car +
    Landsize + BuildingArea + YearBuilt + Lattitude + Longtitude +
    Bedroom2:Rooms
Model 2: Price ~ Distance + Rooms + Bathroom + Car + BuildingArea + YearBuilt +
    Lattitude + Longtitude
  Resid. Df Resid. Dev Df   Deviance  Pr(>Chi)
1     13492 2.3568e+15
2     13495 2.4072e+15 -3 -5.043e+13 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Analysis of variance

- As we can see we need to reject the null hypothesis as the p-value is less than the alpha so we reject the null hypothesis. Thus, there is a significant difference between the two models

3.2.8 Step AIC:

```
Start:  AIC=387904.2
Price ~ Distance + Bedroom2 + Rooms + Bathroom + Landsize + Car +
    Landsize + BuildingArea + YearBuilt + Lattitude + Longtitude +
    Bedroom2:Rooms

                 Df   Deviance    AIC
<none>              2.3568e+15 387904
- Landsize        1 2.3602e+15 387922
- Car             1 2.3889e+15 388085
- Bedroom2:Rooms  1 2.4020e+15 388159
- BuildingArea    1 2.4443e+15 388394
- Longtitude      1 2.4597e+15 388480
- Bathroom        1 2.4923e+15 388657
- YearBuilt       1 2.5197e+15 388805
- Lattitude       1 2.5233e+15 388824
- Distance        1 2.9386e+15 390882

Call:  glm(formula = Price ~ Distance + Bedroom2 + Rooms + Bathroom +
    Landsize + Car + Landsize + BuildingArea + YearBuilt + Lattitude +
    Longtitude + Bedroom2:Rooms, data = dataset3)

Coefficients:
   (Intercept)        Distance        Bedroom2           Rooms        Bathroom        Landsize             Car    BuildingArea       YearBuilt
Lattitude
    -183684958          -40415          167005          328934          192657               4           56969            1327           -4012
-1517979
    Longtitude  Bedroom2:Rooms
        924348          -40851

Degrees of Freedom: 13503 Total (i.e. Null);  13492 Residual
  (62 observations deleted due to missingness)
Null Deviance:      5.253e+15
Residual Deviance: 2.357e+15    AIC: 387900
```

StepAIC

- When we run stepAIC to check which model is better, it shows that the model with all variables is better.

3.2.9 AIC:

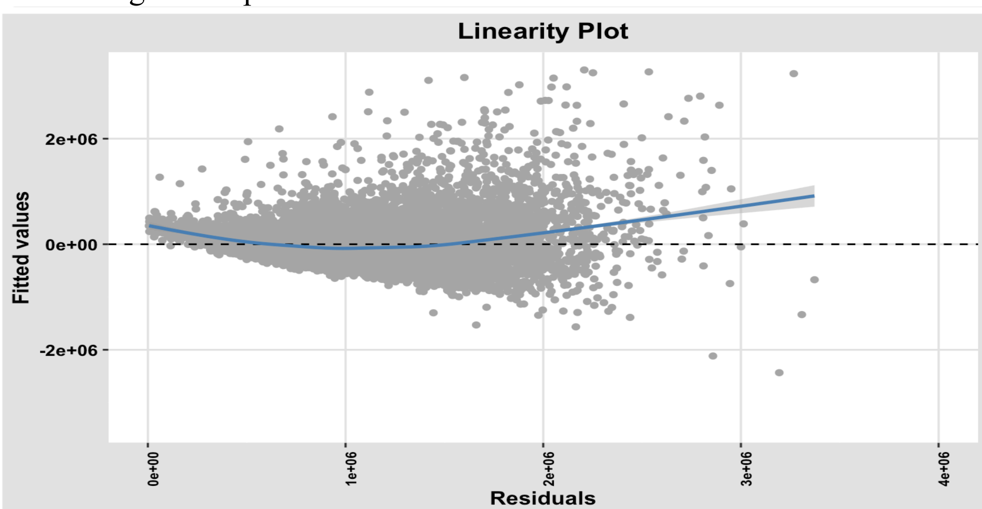
Description: df [2 x 2]

	df <dbl>	AIC <dbl>
model3	13	387904.2
model4	10	388184.1

AIC

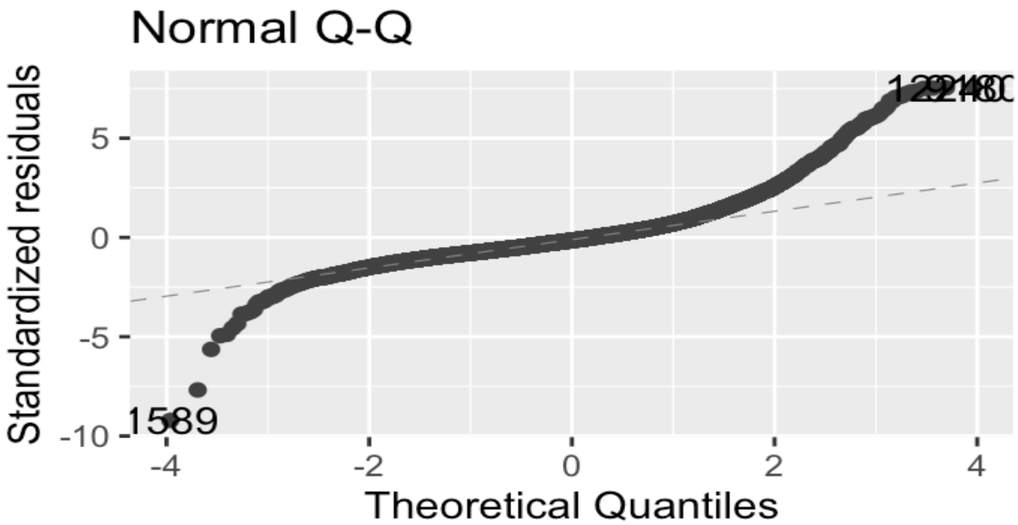
- We see that stepAIC and even AIC predicts that the model with all the variables is better, which contradicts the output of regsubsets. Nonetheless, we will select model 4 as regsubset as we select R2 when we need better prediction power of our model. Even in StepAic, not every possible model is tested. Thus it is better to go with regsubset prediction.

3.2.10 Regression plots:



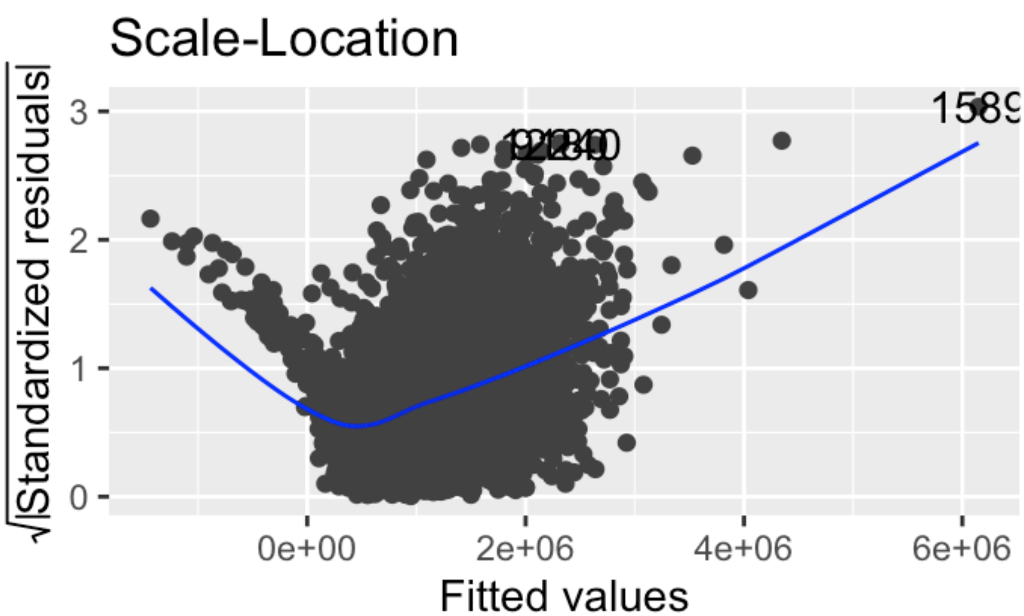
Linearity plot

- The linearity plot: This plot is used to detect linearity. Since our graph is like a curve linearity is not met.



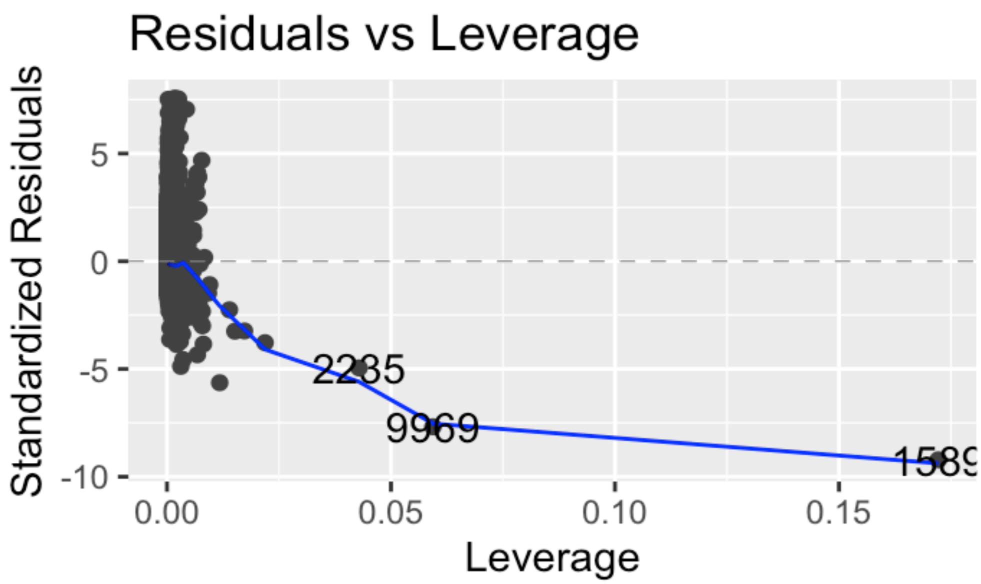
Theoretical quantiles

- Normal Q-Q: It is a normality plot. As our graph does not have a straight line we can say that we do not have a normal dataset



Scale-Location vs fitted values

- Scale-location plot : This plot is used to detect whether the variance is constant. As our plot has a trend , we have an un-equal variance in the dataset



Influence plot

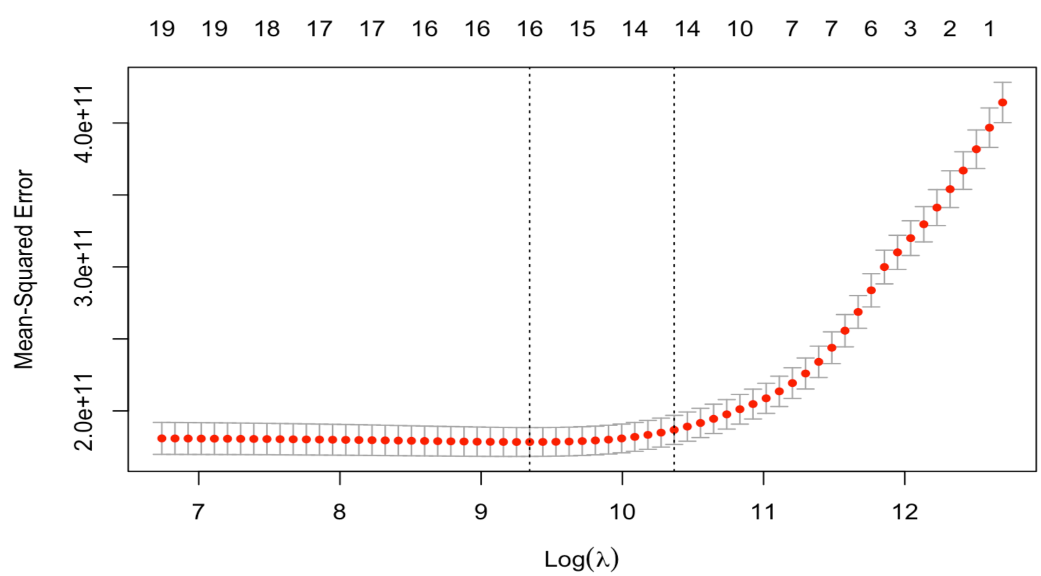
- Residual vs Leverage: It is use to detect the point which have high influence on our model. For instance, point 1589 has the highest influence on our model

3.2.11 Lasso regression:

```
```{r}
trainx<-data.matrix(sampletrain[,c("Suburb","Address","Rooms","Type","Method","SellerG","Date","Distance","Postcode","Bedroom2","Bathroom","Car","Landsize","BuildingArea","YearBuilt","CouncilArea","Lattitude","Longtitude", "Regionname","Propertycount")])
testx<- data.matrix(sampletest[,c("Suburb","Address","Rooms","Type","Method","SellerG","Date","Distance","Postcode","Bedroom2","Bathroom","Car","Landsize","BuildingArea","YearBuilt","CouncilArea","Lattitude","Longtitude", "Regionname","Propertycount")])
trainy<- sampletrain$Price
testy<- sampletest$Price
```

```{r}
lasso1 <- cv.glmnet(trainx, trainy , nfolds = 40)
plot(lasso1)
```
```

Code for lasso regression



Lasso plot

- Plot for lasso regression showing the influential or significant variables for our model.

```
```{r}
coef(modellasso)
```
```

```
21 x 1 sparse Matrix of class "dgCMatrix"
                         s0
(Intercept)   -1.353890e+08
Suburb        -2.318291e+02
Address        .
Rooms          1.597330e+05
Type          -1.886491e+05
Method         .
SellerG        .
Date           .
Distance      -3.949348e+04
Postcode       6.323445e+02
Bedroom2       8.168957e+03
Bathroom       2.130120e+05
Car            2.429794e+04
Landsize       .
BuildingArea   5.697775e-01
YearBuilt     -2.072230e+03
CouncilArea   -1.629168e+03
Lattitude     -1.121063e+06
Longtitude     6.623523e+05
Regionname     9.151284e+03
Propertycount  .
```

Influential variable

```{r}
pre.model.1se<- predict(modellasso2 , newx = trainx)
rmse(trainy , pre.model.1se)
```

```
[1] 429212.1
```

```{r}
pre.test.model.1se<- predict(modellasso2 , newx = testx)
rmse(testy , pre.test.model.1se)
```

```
[1] 405518.7
```

RMSE values

- After calculating our root mean square values(RMSE). We can say that lasso is not a good regression model for our data set as our RMSE values are large

3.2.12 Random forest:

```
rfmodel<-randomForest( x= trainx , y = trainy,mtry = 7 , importance = TRUE)
```

```
rfmodel
```

```
##
## Call:
##  randomForest(x = trainx, y = trainy, mtry = 7, importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##           Mean of squared residuals: 83205278498
##                     % Var explained: 80.14
```

Random Forest

- As seen the Mean of squared residual is high as well, thus we can say that our dataset does not support any kind of regression model on it. After trying multilinear regression, lasso regression, and random forest, we cannot produce the desired result.

3.3 Predicting Region name using suburb and council area

3.3.1 K- nearest neighbor

- K- nearest neighbors is a classification algorithm which we have used to classify our data into the region using council area and suburb.

Creating kNN modelfor prediction of Regionname

```
knnmodel<-knn(sampletrain1[,1:2], sampletest1[,1:2],sampletrain1[,3],k)
```

Confusion matrix

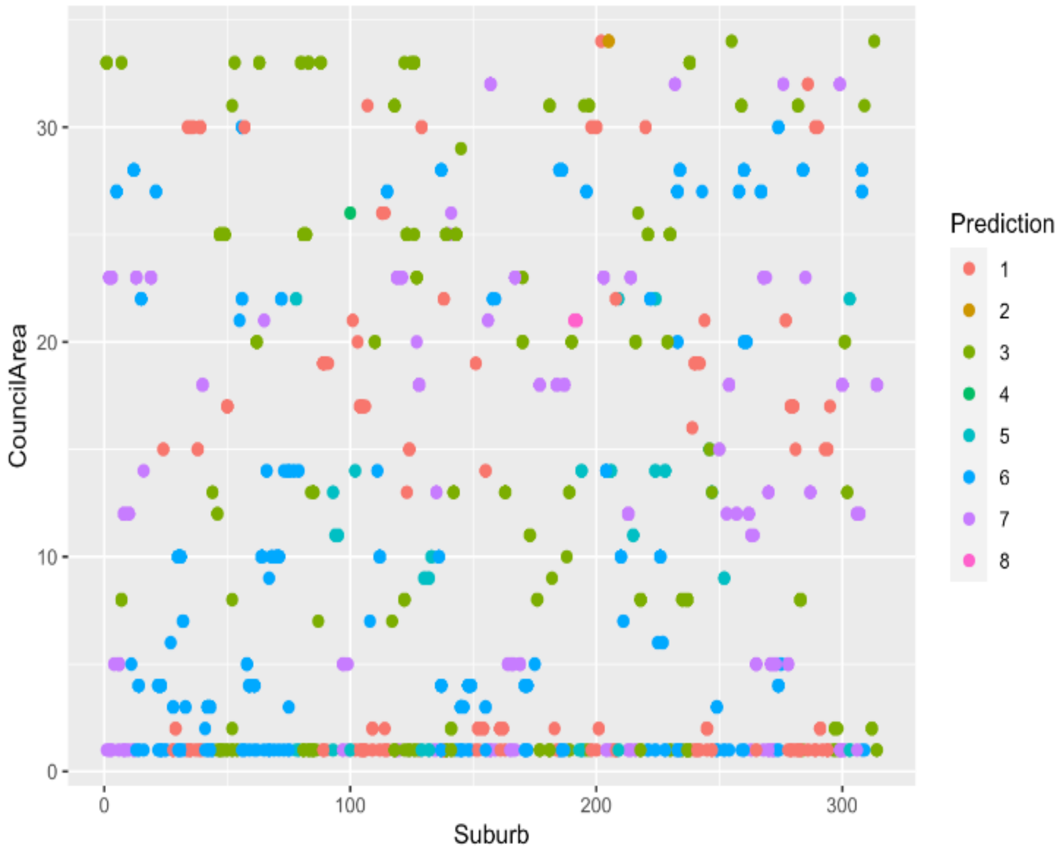
```
table(knnmodel,sampletest1[,3])
```

```
##
## knnmodel    1    2    3    4    5    6    7    8
##        1  401    2   12    3    3   12   15    0
##        2    0    3    0    0    0    0    0    0
##        3    7    2 1142    1   12    4   14    0
##        4    0    0    0    1    0    0    0    0
##        5    2    1    7    3   82    3    0    0
##        6    7   10    9    0   28 1355   17    0
##        7    2    1    7    1    6    6  866    0
##        8    0    0    0    0    0    0    0    5
```

Confusion Matrix

```
1 = eastern metropolitan
2= eastern  vitoria
3 = northern metropolitan
4 = Northearn victoria
5 = southeastern metropolitan
6 = southern metropolitan
7 = Western metropolitan
8 = western victoria
```

Reference table



Prediction plot

- The plot depicts the different points that were predicted by our model.
- The KNN model has an accuracy of 90.778%

4 Conclusion

- After implementing multiple regression models such as multilinear regression, lasso regression, and random forest. It can be concluded that our that set is not fit for regression algorithms.
- The Knn model achieved an exceptional accuracy of 90.78% while classifying the region names using other variables

Reference:

1. Kabacoff, R. (2022). R in action: Data analysis and graphics with R. Manning.
2.STHDA. (n.d.). Retrieved January 20, 2023, from http://www.sthda.com/english/articles/37-model-selection-essentials-in-r/155-best-subsets-regression-essentials-in-r.
3.Kassambara. (2018, March 11). *Multicollinearity Essentials and VIF in R*. STHDA. Retrieved January 20, 2023, from http://www.sthda.com/english/articles/39-regression-model-diagnostics/160-multicollinearity-essentials-and-vif-in-r
4 GeeksforGeeks. (2020, June 22). K-nn classifier in R programming. GeeksforGeeks. Retrieved February 18, 2023, from https://www.geeksforgeeks.org/k-nn-classifier-in-r-programming/
5 GeeksforGeeks. (2020, June 22). K-nn classifier in R programming. GeeksforGeeks. Retrieved February 18, 2023, from https://www.geeksforgeeks.org/k-nn-classifier-in-r-programming/

Appendix

````
---
title: "trying assisgnement"
author: "Murtaza Vora"
date: "`r Sys.Date()`"
output: html_document
editor_options:
  markdown:
    wrap: 72
---

```{r}
library(car)
```

```{r}
library(MASS)
```

```{r}
library(magrittr)
```

```{r}
library(DT)
```

```{r}
library(ggplot2)
```

```{r}
library(leaps)
````
````

```

```{r}
library(qqplotr)
```

```{r}
library(ggthemes)
```

```{r}
library(dplyr)
```

```{r}
library(corrplot)
```

loading the dataset ! Description of each group .

```{r}
dataset<-read.csv("~/Desktop/ALY 6015/melb_data.csv")
```

```{r}
#dataset<- read.csv(file.choose(), header = T , na.strings = "")
```

```{r}
dataset<-as.data.frame(dataset)
datatable((dataset),
        rownames = TRUE , extensions = 'Scroller' , filter = "top", options = list( dom ="tis" , scrollX = TRUE  ,
scrollY = 400, scrollCollapse = TRUE))

```

to check how many NA values we have

```{r}
sum(is.na(dataset$YearBuilt))
```

```{r}
dataset[!complete.cases(dataset),]
```

Using mean of different columns to subtitute instead of NA values.

```{r}
median_Buildingarea <- median(dataset$BuildingArea, na.rm = TRUE)
```

```{r}
dataset[is.na(dataset$BuildingArea) ,"BuildingArea"]<- median_Buildingarea
```
```

````
```{r}
median_Yearbuilt<- median(dataset$YearBuilt , na.rm = TRUE)
```

```{r}
dataset[is.na(dataset$YearBuilt) ,"YearBuilt"]<- median_Yearbuilt
```

```{r}
median_car <- median(dataset$Car , na.ram =TRUE )
```

```{r}
dataset[is.na(dataset$Car) , "Car"]<- median_car
```
````

Basic data analysis

````
```{r}
head(dataset)
```

```{r}
library(tidyr)
```

```{r}
dataset <- dataset %>% drop_na()
```

```{r}
ggplot(dataset , aes(Price , BuildingArea , color = Rooms)) +
  geom_point()+
  ylim(NA,1000)+
  xlim(NA,6000000)+
  xlab("Price of house")+
  ylab("Building Area") +
  ggtitle("Dependence of price on building area") +
  theme_gdocs()+
  theme(axis.title.x = element_text(size = 12 , color = "black",face="bold",),
        axis.title.y = element_text(size = 12 , color = "black",face="bold",),
        plot.title = element_text(size = 16 , color = "black",face="bold", hjust = 0.5),
        axis.text.x = element_text(face="bold", color="black", size=10),
        axis.text.y = element_text(face="bold", color="black", size=10)
        )

```

```{r}
par(las = 2)
ggplot(dataset , aes(Regionname)) +
  geom_bar(fill = "steelblue", color = "black")+
  xlab("Region name")+
  ylab("Total count")+
  ggtitle("Total number of houses per Region") +
  theme_fivethirtyeight()+
````
````

````
  theme(axis.title.x = element_text(size = 12 , color = "black",face="bold",),
        axis.title.y = element_text(size = 12 , color = "black",face="bold",),
        plot.title = element_text(face = "bold" , hjust = 0.5),
        axis.text.x = element_text(face="bold", color="black", size=8, angle = 90),
        axis.text.y = element_text(face="bold", color="black", size=10)
        )

```


```{r}
?ggplot
```
````

Creating our first models to test the different variables and there dependency

````
```{r}
model1<-lm(Price~ Distance +Bedroom2+ Rooms +Bathroom + Landsize +Car
+Landsize+BuildingArea+YearBuilt+ Lattitude+ Longtitude, data= dataset)
```

```{r}
summary(model1)
```

```{r}
murtaza<-outlierTest(model1)
murtaza
```

```{r}
dataset2 <- dataset[-c(12095 , 13246,9576, 7693,6373,12558,3581,5632,6341,3115),]
```

```{r}
dataset2["12095",]
```

```{r}
modeldata<- dataset %>% dplyr::select(Distance,Bedroom2, Rooms,Bathroom,
Landsize,Landsize,BuildingArea,YearBuilt, Lattitude, Longtitude )
modeldata
```

```{r}
corrplotmel<-cor(modeldata)
```

```{r}
corrplot(corrplotmel, method = "pie",order = 'FPC', type = 'lower')
```

```{r}
vif(model1)
```
````
````

```{r}
vif(model1)%>%sqrt>2
```

Creating model2 as we have multicolinearity

```{r}
model2<-lm(Price~ Distance +Bedroom2+ Rooms +Bathroom + Landsize +Car
+Landsize+BuildingArea+YearBuilt+ Lattitude+ Longtitude +Bedroom2:Rooms, data= dataset2)
```

```{r}
summary(model2)
```

```{r}
influencePlot(model2)
```

clearly seen that 1485 and 11021 are highly influencial points so its
better to remove them

```{r}
dataset3 <- dataset2[-c(1485 , 11021,9180 , 2561),]
```

```{r}
dataset3["1485",]
```

After removing outliers and influential points we have created another
model model3

```{r}
model3 <-lm(Price~ Distance +Bedroom2+ Rooms +Bathroom + Landsize +Car
+Landsize+BuildingArea+YearBuilt+ Lattitude+ Longtitude +Bedroom2:Rooms, data= dataset3)
```

```{r}
summary(model3)
```

```{r}
fotifieddataset<-fortify(model3)
```

do not use this graph .

```{r}
library(leaps)
```

regsubsets are used to select the variable that have high influence in
on our dependent variable.

```{r}
```

```
leaps<- regsubsets(Price~ Distance +Bedroom2+ Rooms +Bathroom + Landsize +Car
+Landsize+BuildingArea+YearBuilt+ Lattitude+ Longtitude +Bedroom2:Rooms, data= dataset3, nbest=3 )
plot(leaps , scale = "adjr2")
```

From the leaps plot it is clear that we should remove bedroom2 , landsize ,bedroom: rooms

```{r}
model4 <- lm(Price~ Distance + Rooms +Bathroom  +Car+BuildingArea+YearBuilt+ Lattitude+ Longtitude, data= dataset3)
```

```{r}
summary(model4)
```

As we can see we need to reject the null hypothesis as p-value is less than alpha. thus , there is a significant difference between the two model.

```{r}
anova(model3 , model4,test = "Chisq")
```

As we see that in stepAIC and even AIC predicts that the model with the all the variables is better , which contradicts the output of regsubsets . Nonetheless, we will select model4 as regsubset as we select R2 when we need better prediction power of our model.Even in StepAic , not every possible model is tested. Thus its better to go with regsubsts prediction.

```{r}
stepAIC(model3 , direction = "backward" )
```

```{r}
AIC(model3 , model4)
```

AIC shows that the model with all variables is better contradicting the regsubsets result . As we know that

```{r}
library(MASS)
```

```{r}
plot(model4)
```

```{r}
fotifieddataset
```

```{r}
```

```
ggplot(fotifieddataset , aes(.fitted , .resid))+geom_point(color="darkgray")
+geom_hline(yintercept=0,linetype="dashed") +
  geom_smooth(color= "steelblue")+xlim(0,4000000)+
  xlab("Residuals")+
  ylab("Fitted values")+
  ggtitle("Linearity Plot" )+
  theme_igray()+
  theme(axis.title.x = element_text(size = 12 , color = "black",face="bold"),
        axis.title.y = element_text(size = 12 , color = "black",face="bold"),
        plot.title = element_text(face = "bold" , hjust = 0.5),
        axis.text.x = element_text(face="bold", color="black", size=8, angle = 90),
        axis.text.y = element_text(face="bold", color="black", size=10)
        )
```

```{r}
library(ggfortify)
```

```{r}
autoplot(model4)
```

Trying ridge and lasso to get a better model

```{r}
library(caret)
```

```{r}
library(psych)
```

Divided the dataset into train and test. To run lasso regression

```{r}
set.seed(123)
train <- sort(sample(x= nrow(dataset) , size=nrow(dataset)*0.7))
sampletrain<- dataset[train,]
sampletest<- dataset[-train,]
```

```{r}
ncol(sampletest)
```

```{r}
print(ncol(dataset))
```

Separating "x(independent)" variables and "y(dependent)" variable to meet the condition for lasso regression

```{r}
```

trainx<-
data.matrix(sampletrain[,c("Suburb","Address","Rooms","Type","Method","SellerG","Date","Distance","Postcode",
"Bedroom2","Bathroom","Car","Landsize","BuildingArea","YearBuilt","CouncilArea","Lattitude","Longtitude",
"Regionname","Propertycount")])
testx<-
data.matrix(sampletest[,c("Suburb","Address","Rooms","Type","Method","SellerG","Date","Distance","Postcode","
Bedroom2","Bathroom","Car","Landsize","BuildingArea","YearBuilt","CouncilArea","Lattitude","Longtitude",
"Regionname","Propertycount")])
trainy<- sampletrain$Price
testy<- sampletest$Price
```

```{r}
print(ncol(trainx))
```

```{r}
print(ncol(testx))
```

Lasso

```{r}
library(glmnet)
```

Plot for lasso regression showing the influential or significant
variables for our model

```{r}
lasso1 <- cv.glmnet(trainx, trainy , nfolds = 40)
plot(lasso1)
```

```{r}
print(log(lasso1$lambda.min))
```

```{r}
print(log(lasso1$lambda.1se))
```

```{r}
modellasso<- glmnet(trainx , trainy, alpha =1 , lambda = lasso1$lambda.min)
modellasso
```

```{r}
coef(modellasso)
```

```{r}
?model.matrix()
```

```{r}

modellasso2<- glmnet(trainx , trainy, alpha =1 , lambda = lasso1$lambda.1se)
modellasso2
```

```{r}
coef(modellasso2)
```

```{r}
library(Metrics)
```

The RMSE of our model is very high , thus our is not a good fit for our dataset.

```{r}
pre.model.1se<- predict(modellasso2 , newx = trainx)
rmse(trainy , pre.model.1se)
```

```{r}
pre.test.model.1se<- predict(modellasso2 , newx = testx)
rmse(testy , pre.test.model.1se)
```

for minimum

```{r}
modellassomin<- glmnet(trainx , trainy, alpha =1 , lambda = lasso1$lambda.min)
modellassomin
```

```{r}
pre.model.min<- predict(modellassomin , newx = trainx)
rmse(trainy , pre.model.min)
```

```{r}
pre.test.model.min<- predict(modellassomin , newx = testx)
rmse(testy , pre.test.model.min)
```

```{r}
library(Metrics)
```

```{r}
plot(pre.model.1se)
```

Trying random forest

```{r}
library(randomForest)
```
```

```{r}
rfmodel<-randomForest( x= trainx , y = trainy,mtry = 7 , importance = TRUE)
```

```{r}
rfmodel
```

As seen the Mean of squared residual is high as well , thus we can say that our dataset does not support any kind of regression model on it. As after trying multilinear regression , lasso regression, and random forest we cannot produce desire result

```{r}
predict3 <- predict(rfmodel , testx)
plot(predict3)
```

```{r}
plot(testy)
```

```{r}
library(magrittr)
```

predicting region using Council area and Suhurb using KNN(k-nearest neighbour)

```{r}
question2data<- dataset3%>%dplyr::select(Suburb,CouncilArea,Regionname)
```

```{r}
question2data$Regionname<-as.factor(question2data[,"Regionname"])
question2data$CouncilArea<-as.factor(question2data[,"CouncilArea"])
question2data$Suburb<-as.factor(question2data[,"Suburb"])
```

```{r}
question2data$Regionname<-as.integer(question2data[,"Regionname"])
question2data$CouncilArea<-as.integer(question2data[,"CouncilArea"])
question2data$Suburb<-as.integer(question2data[,"Suburb"])
```

```{r}
str(question2data)
```

```{r}
unique(question2data$Regionname)
```

```{r}
question2data<- question2data%>% drop_na()
```

Dividing the data into train and test

```{r}
set.seed(123)
train <- sort(sample(x= nrow(question2data) , size=nrow(question2data)*0.7))
sampletrain1<- question2data[train,]
sampletest1<- question2data[-train,]
```

The model gives an accuracy of 90.778%

```{r}
library(class)
```

```{r}
k =7
```

Creating kNN modelfor prediction of Regionname

```{r}
knnmodel<-knn(sampletrain1[,1:2], sampletest1[,1:2],sampletrain1[,3],k)
```

Confusion matrix

```{r}
table(knnmodel,sampletest1[,3])
```

Reference Number	Region name
1	Eastern Metropolitan
2	Eastern Victoria
3	Northern Metropolitan
4	Northeastern Victoria
5	Southeastern metropolitan
6	Southern Metropolitan
7	Western Metropolitan
8	Western Victoria

```{r}
plot_prediction <- data.frame(sampletest1$Suburb , sampletest1$CouncilArea , sampletest1$Regionname , predicted
= knnmodel)
```

```{r}
colnames(plot_prediction) <- c("Suburb", "CouncilArea",
                               "Regionname", "Prediction")
```

The graph shows the points predicted by the KNN model

```{r}
```

````
library(ggplot2)
```

```{r}
library(gridExtra)
```

```{r}
ggplot(plot_prediction, aes(Suburb,CouncilArea, color= Prediction,fill = Prediction ))+
  geom_point(size=2)
```
````
````