Regression Modeling of Real Estate Data

Mayur Nagare

MSc Part II

All About Dataset

```
Name of Dataset :- Real Estate data
Source :- https://www.kaggle.com/dcw8161/real-estate-priceprediction/data
Variables :-
1) X1 transaction date (Date at which home is bought)
2) X2 house age (age of house from when it was built)
3) X3 distance to the nearest MRT station
4) X4 number of convenience stores
5) X5 latitude ( represents the geographical position of property)
6) X6 longitude ( represents geographical position of property )
7) Y house price of unit area
```

Import Dataset

```
data=read.csv(file.choose(), header = T)
head(data)
```

```
##   No       X1   X2         X3 X4       X5       X6    Y
## 1  1 2012.917 32.0   84.87882 10 24.98298 121.5402 37.9
## 2  2 2012.917 19.5  306.59470  9 24.98034 121.5395 42.2
## 3  3 2013.583 13.3  561.98450  5 24.98746 121.5439 47.3
## 4  4 2013.500 13.3  561.98450  5 24.98746 121.5439 54.8
## 5  5 2012.833  5.0  390.56840  5 24.97937 121.5425 43.1
## 6  6 2012.667  7.1 2175.03000  3 24.96305 121.5125 32.1
```

```
str(data)
```

```
## 'data.frame':    414 obs. of  8 variables:
##  $ No: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ X1: num  2013 2013 2014 2014 2013 ...
##  $ X2: num  32 19.5 13.3 13.3 5 7.1 34.5 20.3 31.7 17.9 ...
##  $ X3: num  84.9 306.6 562 562 390.6 ...
##  $ X4: int  10 9 5 5 5 3 7 6 1 3 ...
##  $ X5: num  25 25 25 25 25 ...
##  $ X6: num  122 122 122 122 122 ...
##  $ Y : num  37.9 42.2 47.3 54.8 43.1 32.1 40.3 46.7 18.8 22.1 ...
```

```
names(data)
```

```
## [1] "No" "X1" "X2" "X3" "X4" "X5" "X6" "Y"
```

```
dim(data)
```

```
## [1] 414   8
```

To check if NA values is present or not in the dataset

```
sum(is.na(data))
```

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

Correlation Matrix

```
cor(data)
```

```
##             No           X1          X2          X3           X4          X5
## No  1.00000000 -0.048657949 -0.03280811 -0.01357349 -0.012698946 -0.01010966
## X1 -0.04865795  1.000000000  0.01754877  0.06087995  0.009635445  0.03505776
## X2 -0.03280811  0.017548767  1.00000000  0.02562205  0.049592513  0.05441990
## X3 -0.01357349  0.060879953  0.02562205  1.00000000 -0.602519145 -0.59106657
## X4 -0.01269895  0.009635445  0.04959251 -0.60251914  1.000000000  0.44414331
## X5 -0.01010966  0.035057756  0.05441990 -0.59106657  0.444143306  1.00000000
## X6 -0.01105928 -0.041081778 -0.04852005 -0.80631677  0.449099007  0.41292394
## Y  -0.02858717  0.087490606 -0.21056705 -0.67361286  0.571004911  0.54630665
##             X6           Y
## No -0.01105928 -0.02858717
## X1 -0.04108178  0.08749061
## X2 -0.04852005 -0.21056705
## X3 -0.80631677 -0.67361286
## X4  0.44909901  0.57100491
## X5  0.41292394  0.54630665
## X6  1.00000000  0.52328651
## Y   0.52328651  1.00000000
```

Correlation Plot

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.1.2
```

```
## corrplot 0.92 loaded
```

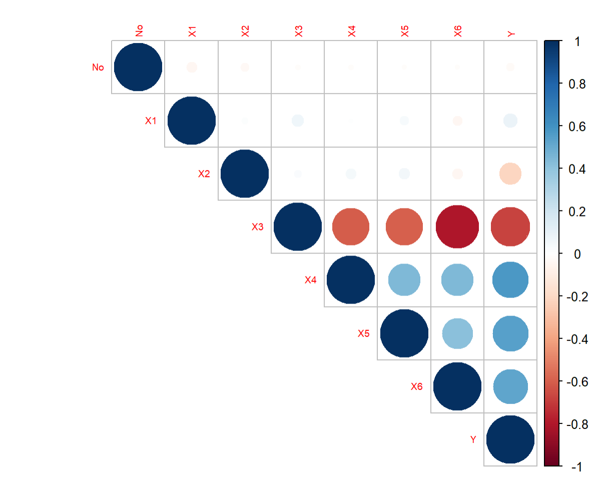
```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.2
```

```
library(ggfortify)
```

```
## Warning: package 'ggfortify' was built under R version 4.1.2
```

```
corrplot(cor(data),type = "upper",method="circle",
         mar=c(0.7,0.7,0.7,0.7),tl.cex = 0.6)
```



Regression Model

```
model1=lm(Y~. , data = data)
summary(model1)
```

```
##
## Call:
## lm(formula = Y ~ ., data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -36.003  -5.196  -0.990   4.181  75.384
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.404e+04  6.788e+03  -2.068  0.03927 *
## No          -3.593e-03  3.653e-03  -0.984  0.32590
## X1           5.079e+00  1.559e+00   3.259  0.00121 **
## X2          -2.708e-01  3.855e-02  -7.026 9.04e-12 ***
## X3          -4.521e-03  7.189e-04  -6.289 8.28e-10 ***
## X4           1.129e+00  1.882e-01   6.000 4.37e-09 ***
## X5           2.247e+02  4.458e+01   5.040 7.02e-07 ***
## X6          -1.442e+01  4.863e+01  -0.297  0.76691
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.858 on 406 degrees of freedom
## Multiple R-squared:  0.5834, Adjusted R-squared:  0.5762
## F-statistic: 81.21 on 7 and 406 DF,  p-value: < 2.2e-16
```

```
From the above model we can see that variable No and X6 are insignificant, as there P Value is not less than 0.0
5. So we remove those variables and run the model again
```

```
model2=lm(Y ~ .-No -X6 , data = data)
summary(model2)
```

```
##
## Call:
## lm(formula = Y ~ . - No - X6, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -35.623  -5.371  -1.020   4.244  75.346
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.596e+04  3.233e+03  -4.936 1.17e-06 ***
## X1           5.135e+00  1.555e+00   3.303  0.00104 **
## X2          -2.694e-01  3.847e-02  -7.003 1.04e-11 ***
## X3          -4.353e-03  4.899e-04  -8.887  < 2e-16 ***
## X4           1.136e+00  1.876e-01   6.056 3.17e-09 ***
## X5           2.269e+02  4.417e+01   5.136 4.36e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.848 on 408 degrees of freedom
## Multiple R-squared:  0.5823, Adjusted R-squared:  0.5772
## F-statistic: 113.8 on 5 and 408 DF,  p-value: < 2.2e-16
```

Checking The Assumption Of MLR Model

```
library(tidyverse)
```

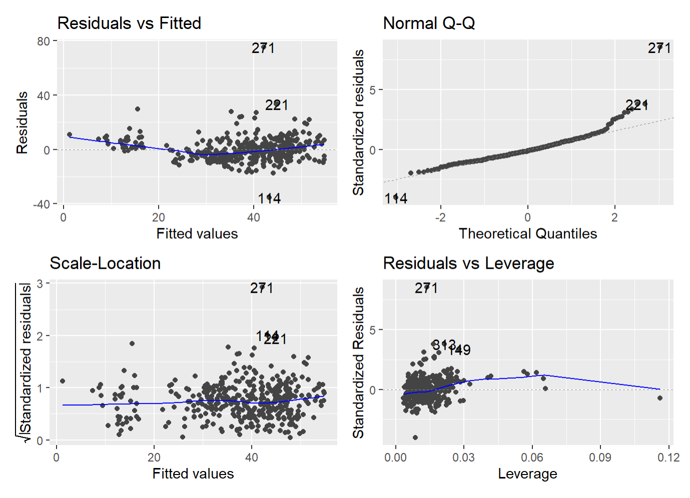
```
## Warning: package 'tidyverse' was built under R version 4.1.2
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(ggfortify)
autoplot(model2)
```



Multicollinearity

To Check Multicollinearity using VIF function

```
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

```
vif(model2)
```

```
##       X1       X2       X3       X4       X5
## 1.013834 1.013243 2.016855 1.611299 1.585635
```

Autocorrelation

To check autocorrelation we use Durbin Watson Test

```
durbinWatsonTest(model2)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1     -0.07998669      2.154158   0.128
##  Alternative hypothesis: rho != 0
```

Heteroscedasticity

To check heteroscedasticity we use breusch pagan godfrey test

```
library(lmtest)
```

```
## Warning: package 'lmtest' was built under R version 4.1.2
```

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 4.1.2
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
bptest(model2)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model2
## BP = 5.7624, df = 5, p-value = 0.33
```

Multivariate Normality

To check normality we use kolmogorov smirnov normality test

```
library(nortest)
lillie.test(residuals(model2))
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  residuals(model2)
## D = 0.078433, p-value = 2.105e-06
```

```
From the above analysis we can see that the assumption of linearity and normality is violated. So we can fix this
error by removing outliers and adding those variables in the model whose having high correlation among them
```

Remove outliers

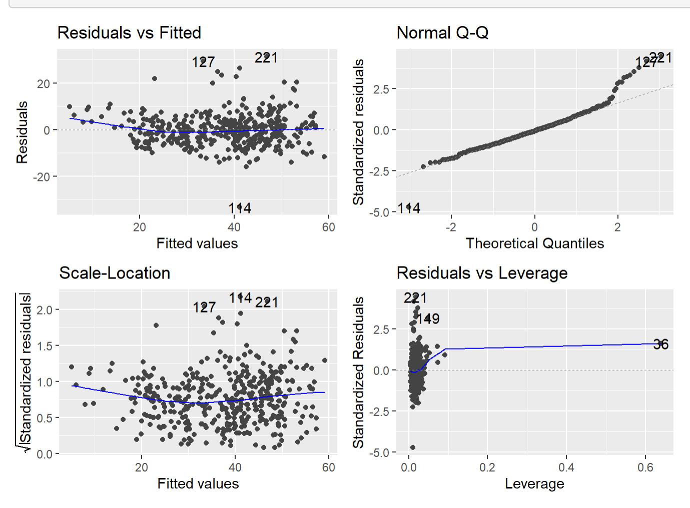
```
data=data[-c(345,313,271,229),]
```

Final Model

```
model3=lm(Y ~ .-No-X6+X4:X3+X5:X3 , data = data )
summary(model3)
```

```
##
## Call:
## lm(formula = Y ~ . - No - X6 + X4:X3 + X5:X3, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.496  -4.411  -0.537   3.674  31.438
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.010e+04  2.837e+03  -7.086 6.24e-12 ***
## X1           4.587e+00  1.257e+00   3.648 0.000299 ***
## X2          -2.888e-01  3.117e-02  -9.266  < 2e-16 ***
## X3           2.920e+00  6.623e-01   4.408 1.34e-05 ***
## X4           1.430e+00  1.789e-01   7.995 1.39e-14 ***
## X5           4.368e+02  5.236e+01   8.342 1.18e-15 ***
## X3:X4       -1.200e-03  2.287e-04  -5.246 2.52e-07 ***
## X3:X5       -1.171e-01  2.654e-02  -4.412 1.32e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.11 on 402 degrees of freedom
## Multiple R-squared:  0.7029, Adjusted R-squared:  0.6978
## F-statistic: 135.9 on 7 and 402 DF,  p-value: < 2.2e-16
```

```
autoplot(model3)
```



Conclusion Of The Final Model

1. R-squared Value Of Our Final Model Is 70.29% .

2. From The Residual Vs Fitted Graph We Can See That The Estimated Error Curve Of Our Final Model Is Almost Converge To 0.

3.From The QQ-plot We Can See That The Our Model Behaves Like Normal Except For The Tail Parts.

4. Data Is Homoscedastic.

`