4/11/2020

MINI PROJECT

BOSTON HOUSING PRICE ANALYSIS

Sakshi Agrawal

DATTA MEGHE COLLEGE OF ENGINEERING, AIROLI

PROJECT REPORT ON

"BOSTON HOUSING PRICE ANALYSIS"

SUBMITTED BY:

1. SAKSHI AGRAWAL (2017DSIT067)
2. ANIKET BRAMHE (2017DSIT001)

UNDER THE GUIDANCE OF:

Prof. NEHA THAKUR

DEPARTMENT OF INFORMATION TECHNOLOGY
DATTA MEGHE COLLEGE OF ENGINEERING, AIROLI

Report submitted in a fulfilment of

MUMBAI UNIVERSITY

In

R (MINI PROJECT) LAB

2019-2020

DATTA MEGHE COLLEGE OF ENGINEERING, AIROLI

Plot No 98, Cidco, Sector-3 Post Box No 15 Airoli Navi Mumbai-400078

CERTIFICATE OF APPROVAL

Project Entitled: "BOSTON HOUSING PRICE ANALYSIS"

SUBMITTED BY

1. SAKSHI AGRAWAL (01)
2. ANIKET BRAMHE (08)

In this partial fulfilment of the degree of B.E. in "INFORMATION TECHNOLOGY" is approved.

Signature of Internal Examiner **Signature of External Examiner**

Signature of Head of Department **Signature of Principal**

ACKNOWLEDGEMENT

I would like to thank all those who have contributed to the completion of the project and helped me with valuable suggestions for improvement. I am extremely grateful to our Prof. Shila Jawale and Prof. Neha Thakur Department of Information Technology, for providing me with the atmosphere for the creative work, guidance and encouragement.

TABLE OF CONTENTS

INTRODUCTION

In this project, we will evaluate the performance and predictive power of a model that has been trained and tested on data collected from homes in suburbs of Boston, Massachusetts. A model trained on this data that is seen as a good fit could then be used to make certain predictions about a home — in particular, its monetary value. This model would prove to be invaluable for someone like a real estate agent who could make use of such information on a daily basis.

The Boston Housing Dataset consists of price of houses in various places in Boston. Alongside with price, the dataset also provides the following information:

Code	Description
CRIM	per capita crime rate by town
ZN	proportion of residential land zoned for lots over 25,000 sq.ft.
INDUS	proportion of non-retail business acres per town
CHAS	Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
NOX	nitric oxides concentration (parts per 10 million)
RM	average number of rooms per dwelling
AGE	proportion of owner-occupied units built prior to 1940
DIS	weighted distances to five Boston employment centres
RAD	index of accessibility to radial highways
TAX	full-value property-tax rate per $10,000
PTRATIO	pupil-teacher ratio by town
B	1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
LSTAT	% lower status of the population
MEDV	Median value of owner-occupied homes in $1000's

The prices of the house indicated by the variable MEDV is our *target variable* and the remaining are the *feature variables* based on which we will predict the value of a house.

INPUT DATASET

CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0.00632	18	2.31	0	0.538	6.575	65.2	4.09	1	296	15.3	396.9	4.98	24
0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.9	9.14	21.6
0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.9	5.33	36.2
0.02985	0	2.18	0	0.458	6.43	58.7	6.0622	3	222	18.7	394.12	5.21	28.7
0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311	15.2	395.6	12.43	22.9
0.14455	12.5	7.87	0	0.524	6.172	96.1	5.9505	5	311	15.2	396.9	19.15	27.1
0.21124	12.5	7.87	0	0.524	5.631	100	6.0821	5	311	15.2	386.63	29.93	16.5
0.17004	12.5	7.87	0	0.524	6.004	85.9	6.5921	5	311	15.2	386.71	17.1	18.9
0.22489	12.5	7.87	0	0.524	6.377	94.3	6.3467	5	311	15.2	392.52	20.45	15
0.11747	12.5	7.87	0	0.524	6.009	82.9	6.2267	5	311	15.2	396.9	13.27	18.9
0.09378	12.5	7.87	0	0.524	5.889	39	5.4509	5	311	15.2	390.5	15.71	21.7
0.62976	0	8.14	0	0.538	5.949	61.8	4.7075	4	307	21	396.9	8.26	20.4
0.63796	0	8.14	0	0.538	6.096	84.5	4.4619	4	307	21	380.02	10.26	18.2
0.62739	0	8.14	0	0.538	5.834	56.5	4.4986	4	307	21	395.62	8.47	19.9
1.05393	0	8.14	0	0.538	5.935	29.3	4.4986	4	307	21	386.85	6.58	23.1
0.7842	0	8.14	0	0.538	5.99	81.7	4.2579	4	307	21	386.75	14.67	17.5
0.80271	0	8.14	0	0.538	5.456	36.6	3.7965	4	307	21	288.99	11.69	20.2
0.7258	0	8.14	0	0.538	5.727	69.5	3.7965	4	307	21	390.95	11.28	18.2
1.25179	0	8.14	0	0.538	5.57	98.1	3.7979	4	307	21	376.57	21.02	13.6
0.85204	0	8.14	0	0.538	5.965	89.2	4.0123	4	307	21	392.53	13.83	19.6
1.23247	0	8.14	0	0.538	6.142	91.7	3.9769	4	307	21	396.9	18.72	15.2
0.98843	0	8.14	0	0.538	5.813	100	4.0952	4	307	21	394.54	19.88	14.5
0.75026	0	8.14	0	0.538	5.924	94.1	4.3996	4	307	21	394.33	16.3	15.6

Boston_Housing_Data.xlsx

CODE

```
library(MASS)
library(ISLR)
#install.packages("ISLR")
data("Boston")
#print head
head(Boston)

#rows for dataset
nrow(Boston)

summary(Boston)

set.seed(2)
library(caTools)

#split using 70 percent
split<-sample.split(Boston$medv ,SplitRatio = 0.7)
split

training_data <- subset(Boston, split == "TRUE")
testing_data<-subset(Boston,split=="FALSE")


###Exploratory Data Analysis###

#creating scatterplot matrix
attach(Boston)
library(lattice)
splom(~ Boston[c(1 : 6, 14)], groups = NULL, data = Boston,
axis.line.tck = 0, axis.text.aplha = 0)
splom(~ Boston[c(7:14)], groups = NULL, data = Boston, axis.line.tck
= 0, axis.text.aplha = 0)

#corplot to visualize
#install.packages("corrplot", dependencies = 1)
library(corrplot)
cr <- cor(Boston)
corrplot(cr, type = "lower")
corrplot(cr, method = "number")
#to view corelation of variables
plot(Boston$crim ,Boston$medv, cex = 0.5, xlab = "CrimeRate", ylab =
"Price")
cr <- cor(Boston)
pairs(~ medv + ptratio + black + lstat + dis + rm + crim, data =
Boston, main = "Boston Data")

## crim is not acceptable to be a linear variable
```

```
#studying crim and medv
plot(crim, medv)
fit1 <- lm(medv ~ crim, data = Boston)
abline(fit1, col = "red")

# regression fit line

#studying rm and medv
plot(rm, medv)
fit1 <- lm(medv ~ rm, data = Boston)
abline(fit1, col = "red")
# regression fit line

#studying lstat and medv
plot(lstat, medv)
fit1 <- lm(medv ~ lstat, data = Boston)
abline(fit1, col = "red")

# regression fit line

##Creating Model

#Since line is acceptable through rm and lstat variable we use rm,
lstat to model to predict data
#Using rm, lstat as they are good linear variables.

#Rm
model_regx_rm <- lm(medv ~ rm, data = training_data)
#summary
summary(model_regx_rm)
#prediction
predic_rm <- predict(model_regx_rm, testing_data)
predic_rm
#compare actual values and prediction
plot(testing_data$medv, type = "l", lty = 1.8, col = "green")
lines(predic_rm,type = "l", col = "blue")

#lstat
model_regx_lstat <- lm(medv ~ lstat, data = training_data)
#summary
summary(model_regx_lstat)
#prediction
predic_lstat <- predict(model_regx_lstat, testing_data)
predic_lstat
#compare actual values and prediction
plot(testing_data$medv, type = "l", lty = 1.8, col = "green")
lines(predic_lstat, type = "l", col = "blue")
```

```
# finding root mean sq. error
rmse <- sqrt(mean(predic_rm-testing_data$medv) ^ 2)
rmse
rmse <-sqrt(mean(predic_lstat-testing_data$medv) ^ 2)
rmse


#### NoW we try multi linear regression ####

#selecting only variables
model_regx_ml <- lm(medv~ rm + lstat, data = Boston)
#summary
summary(model_regx_ml)

#selecting all variables
model_regx_all <- lm(medv~., data = training_data)
#summary
summary(model_regx_all)

#removing age and indus
model_regx_selected <- lm(medv~ crim +zn +tax +chas +rm +rad +dis +nox
+
           ptratio +black +lstat, data = training_data)
#summary
summary(model_regx_selected)

#prediction
predic_selected <- predict(model_regx_selected, testing_data)
predic_selected


# finding root mean sq. error
rmse <- sqrt(mean(predic_selected - testing_data$medv)^2)
rmse

#compare actual values and prediction
plot(testing_data$medv, type = "l", lty = 1.8, col = "green")
lines(predic_selected, type = "l", col = "blue")


#since rmse value is still high we need to optimize the model

f1 = lm(medv ~ lstat +I(lstat^2), Boston)
summary(fit1)
attach(Boston)
f11 = lm(medv ~ poly(lstat, 4))
plot(medv ~ lstat)
points(lstat, fitted(f11), col = "blue", pch = 20)
```

```
f2 = lm(medv ~ rm +I(rm^2), Boston)
summary(f2)
attach(Boston)
fit22 = lm(medv ~ poly(rm, 4))
plot(medv ~ rm)
points(rm,  fitted(fit22), col = "blue", pch = 20)


#building final model
fit_final = lm(medv ~ lstat +crim +rm +dis +black +chas +nox +rad
+tax +ptratio +I(lstat ^ 2) +I(rm ^ 2))
summary(fit_final)

#prediction
predic_fit_final <- predict(fit_final, testing_data)
predic_fit_final

# finding root mean sq. error
rmse <- sqrt(mean(predic_fit_final - testing_data$medv)^2)
rmse

#compare actual values and prediction
plot(testing_data$medv, type = "l", lty = 1.8, col = "green")
lines(predic_fit_final, type = "l", col = "blue")
```

OUTPUT

```
Console C:/Users/Sakshi/Downloads/
> library(MASS)
> library(ISLR)
Warning message:
package 'ISLR' was built under R version 3.6.3
> #install.packages("ISLR")
> data("Boston")
> #print head
> head(Boston)
     crim zn indus chas    nox     rm  age     dis rad tax
1 0.00632 18  2.31     0 0.538 6.575 65.2 4.0900   1 296
2 0.02731  0  7.07     0 0.469 6.421 78.9 4.9671   2 242
3 0.02729  0  7.07     0 0.469 7.185 61.1 4.9671   2 242
4 0.03237  0  2.18     0 0.458 6.998 45.8 6.0622   3 222
5 0.06905  0  2.18     0 0.458 7.147 54.2 6.0622   3 222
6 0.02985  0  2.18     0 0.458 6.430 58.7 6.0622   3 222
  ptratio  black lstat medv
1    15.3 396.90  4.98 24.0
2    17.8 396.90  9.14 21.6
3    17.8 392.83  4.03 34.7
4    18.7 394.63  2.94 33.4
5    18.7 396.90  5.33 36.2
6    18.7 394.12  5.21 28.7
> #rows for dataset
> nrow(Boston)
```

```
Console C:/Users/Sakshi/Downloads/
[1] 506
> summary(Boston)
      crim                zn              indus
 Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46
 1st Qu.: 0.08204   1st Qu.:  0.00   1st Qu.: 5.19
 Median : 0.25651   Median :  0.00   Median : 9.69
 Mean   : 3.61352   Mean   : 11.36   Mean   :11.14
 3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10
 Max.   :88.97620   Max.   :100.00   Max.   :27.74
      chas              nox               rm
 Min.   :0.00000   Min.   :0.3850   Min.   :3.561
 1st Qu.:0.00000   1st Qu.:0.4490   1st Qu.:5.886
 Median :0.00000   Median :0.5380   Median :6.208
 Mean   :0.06917   Mean   :0.5547   Mean   :6.285
 3rd Qu.:0.00000   3rd Qu.:0.6240   3rd Qu.:6.623
 Max.   :1.00000   Max.   :0.8710   Max.   :8.780
      age              dis              rad
 Min.   :  2.90   Min.   : 1.130   Min.   : 1.000
 1st Qu.: 45.02   1st Qu.: 2.100   1st Qu.: 4.000
 Median : 77.50   Median : 3.207   Median : 5.000
 Mean   : 68.57   Mean   : 3.795   Mean   : 9.549
 3rd Qu.: 94.08   3rd Qu.: 5.188   3rd Qu.:24.000
 Max.   :100.00   Max.   :12.127   Max.   :24.000
      tax             ptratio          black
```

```
Console C:/Users/Sakshi/Downloads/
 Min.   :187.0   Min.   :12.60   Min.   :  0.32
 1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38
 Median :330.0   Median :19.05   Median :391.44
 Mean   :408.2   Mean   :18.46   Mean   :356.67
 3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23
 Max.   :711.0   Max.   :22.00   Max.   :396.90
     lstat            medv
 Min.   : 1.73   Min.   : 5.00
 1st Qu.: 6.95   1st Qu.:17.02
 Median :11.36   Median :21.20
 Mean   :12.65   Mean   :22.53
 3rd Qu.:16.95   3rd Qu.:25.00
 Max.   :37.97   Max.   :50.00
> set.seed(2)
> library(caTools)
Warning message:
package 'caTools' was built under R version 3.6.3
> #split using 70 percent
> split<-sample.split(Boston$medv ,SplitRatio = 0.7)
> split
  [1]  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE FALSE
  [9] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [17]  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE
 [25]  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE
```

```
Console C:/Users/Sakshi/Downloads/
 [33]  TRUE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE
 [41]  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE
 [49] FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE
 [57]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [65]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE
 [73]  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
 [81]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
 [89]  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE
 [97] FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE
[105]  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
[113] FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
[121]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
[129]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
[137]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE
[145] FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE
[153]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[161]  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE
[169]  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE
[177]  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[185]  TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE
[193]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
[201]  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[209]  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[217]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE
```

```
Console C:/Users/Sakshi/Downloads/
[225]  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE
[233]  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE
[241] FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE
[249] FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE
[257]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[265]  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE
[273] FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE
[281]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[289]  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
[297]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
[305]  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
[313] FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE
[321] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[329]  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE
[337]  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE
[345]  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE  TRUE
[353] FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE
[361]  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE
[369]  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE
[377]  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE
[385] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[393]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE
[401]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
[409] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE
```

```
Console C:/Users/Sakshi/Downloads/
[417]  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE  TRUE
[425] FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[433]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[441] FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[449] FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
[457]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
[465]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
[473] FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
[481]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
[489]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE
[497] FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE
[505] FALSE  TRUE
> training_data<-subset(Boston,split=="TRUE")
> testing_data<-subset(Boston,split=="FALSE")
> ###Exploratory Data Analysis###
> #creating scatterplot matrix
> attach(Boston)
> library(lattice)
> splom(~Boston[c(1:6,14)], groups=NULL, data=Boston,axis.l
ine.tck = 0,axis.text.aplha = 0)
> splom(~Boston[c(7:14)], groups=NULL, data=Boston,axis.lin
e.tck = 0,axis.text.aplha = 0)
> #corplot to visualize
> #install.packages("corrplot", dependencies = 1)
```

```
Console C:/Users/Sakshi/Downloads/
> library(corrplot)
corrplot 0.84 loaded
Warning message:
package 'corrplot' was built under R version 3.6.3
> corrplot(cr, type = "lower")
Error in corrplot(cr, type = "lower") : object 'cr' not fou
nd
> corrplot(cr, method = "number")
Error in corrplot(cr, method = "number") : object 'cr' not
 found
> #to view corelation of variables
> plot(Boston$crim ,Boston$medv, cex = 0.5, xlab = "CrimeRa
te", ylab = "Price")
> cr<-cor(Boston)
> corrplot(cr, type = "lower")
> corrplot(cr, method = "number")
> pairs(~ medv + ptratio + black + lstat + dis + rm + crim,
 data = Boston, main = "Boston Data")
> ## crim is not acceptable to be a linear variable
> #studying crim and medv
> plot(crim,medv)
> fit1<-lm(medv~crim, data=Boston)
> abline(fit1, col="red")
> # regression fit line
```

```
Console C:/Users/Sakshi/Downloads/
> #studying rm and medv
> plot(rm,medv)
> fit1<-lm(medv~rm, data=Boston)
> abline(fit1, col="red")
> # regression fit line
> #studying lstat and medv
> plot(lstat,medv)
> fit1<-lm(medv~lstat, data=Boston)
> abline(fit1, col="red")
> # regression fit line
> ##Creating Model
> ####Since line is acceptable through rm and lstat variabl
e we use rm, lstat to model to predict data
> ####Using rm, lstat as they are good linear variables
> #Rm
> model_regx_rm<-lm(medv~rm,data = training_data)
> #summary
> summary(model_regx_rm)

Call:
lm(formula = medv ~ rm, data = training_data)

Residuals:
     Min       1Q   Median       3Q      Max
```

```
Console C:/Users/Sakshi/Downloads/
-22.979  -3.111   0.102   3.032  39.099

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -33.423      3.303  -10.12   <2e-16 ***
rm             8.918      0.519   17.18   <2e-16 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.058 on 365 degrees of freedom
Multiple R-squared:  0.4472,    Adjusted R-squared:  0.4457

F-statistic: 295.3 on 1 and 365 DF,  p-value: < 2.2e-16

> #prediction
> predic_rm<-predict(model_regx_rm, testing_data)
> predic_rm
       2        4        5        8        9
23.840892 28.986720 30.315539 21.620249 16.795477
      19       21       23       24       26
15.234784 16.251464 21.352701 18.418598 16.510094
      27       31       34       35       36
18.418598 17.526774 17.419755 20.942462 19.488788
```

```
Console C:/Users/Sakshi/Downloads/
        42          44          49          53          56
26.953360 21.968060 14.726444 24.643534 31.225200
        67          72          74          80          87
18.186724 19.738499 22.271281 18.962611 20.220084
        90          94          97         100         104
29.709098 21.968060 21.539985 32.714547 21.308110
       106         109         113         114         119
18.757492 24.313559 19.310423 20.906789 18.944775
       123         130         131         132         133
19.738499 16.848987 24.170867 22.993659 23.403898
       134         135         139         143         145
18.498862 17.919176 18.811001 14.762117 10.302994
       148         149         150         151         164
10.508114 12.826858 16.492257 21.174336 41.267146
       167         170         171         173         174
37.289608 23.671445 18.971530 16.269301 23.796301
       178         186         188         189         190
22.895558 21.450802 27.060379 25.044855 30.654432
       195         202         210         223         224
25.472931 21.531066 14.235941 27.925449 25.597787
       226         228         230         235         236
44.388532 30.458231 25.009182 26.560957 20.853280
       240         241         242         244         246
25.490768 28.085977 20.933544 23.591181 16.563603
```

```
Console C:/Users/Sakshi/Downloads/
20.255757 25.374831 21.156500 27.167398
> #compare actual values and prediction
> plot(testing_data$medv, type = "l", lty = 1.8, col = "gre
en")
> lines(predic_rm,type = "l", col = "blue")
> #lstat
> model_regx_lstat<-lm(medv~lstat,data = training_data)
> summary
function (object, ...)
UseMethod("summary")
<bytecode: 0x000002afbf9ca370>
<environment: namespace:base>
> summary(model_regx_lstat)

Call:
lm(formula = medv ~ lstat, data = training_data)

Residuals:
    Min      1Q  Median      3Q     Max
-15.582  -4.319  -1.413   2.072  24.191

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 35.46712    0.67760   52.34   <2e-16 ***
```

```
Console C:/Users/Sakshi/Downloads/
lstat       -1.01339    0.04791  -21.15   <2e-16 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.363 on 365 degrees of freedom
Multiple R-squared:  0.5507,    Adjusted R-squared:  0.5494

F-statistic: 447.3 on 1 and 365 DF,  p-value: < 2.2e-16

> #prediction
> predic_lstat<-predict(model_regx_lstat, testing_data)
> predic_lstat
        2         4         5         8         9
26.2047347 32.4877507 30.0657494 16.0607039  5.1363631
       19        21        23        24        26
23.6205910 14.1656652 16.4964615 15.3209295 18.7360527
       27        31        34        35        36
20.4588152 12.5645095 16.8714157 14.8547702 25.6575043
       42        44        49        53        56
30.5623103 27.9274972  4.2445802 30.1164189 30.5927120
       67        72        74        80        87
25.0900060 25.4548263 27.8261582 26.2452703 22.4349251
       90        94        97       100       104
```

```
Console C:/Users/Sakshi/Downloads/
30.7041849 26.6810278 26.4175465 30.8663272 23.5901893
       316        320        321        330        334
23.8131350 22.5666657 28.1707107 28.0288361 29.7110630
       340        344        346        348        351
25.5967009 28.1909785 24.7961230 29.0219580 29.4070461
       353        355        356        358        359
27.5728108 27.3093294 29.8225359 22.0194353 23.8334028
       364        371        374        376        380
20.6310914 32.4674829  0.2315571 21.8471590 13.3954891
       382        385        398        399        407
14.1048618  4.4269904 15.2803939  4.4675259 11.8146012
       409        415        416        418        420
 8.7136287 -2.0080341  6.0281461  8.4704152 12.4226350
       423        425        426        441        442
21.1783218 18.0773494 10.7505420 13.0610705 15.6857498
       449        453        459        471        473
17.0943614 17.9658765 19.0198018 18.9589984 20.9148405
       477        487        494        497        498
16.5369971 20.2865389 23.2963063 14.0440585 21.1783218
       499        502        503        505
22.3741217 25.6676382 26.2655381 28.9003512
> #compare actual values and prediction
> plot(testing_data$medv, type = "l", lty = 1.8, col = "gre
en")
```

```
Console C:/Users/Sakshi/Downloads/
> lines(predic_lstat,type = "l", col = "blue")
> # finding root mean sq. error
> rmse<-sqrt(mean(predic_rm-testing_data$medv)^2)
> rmse
[1] 0.3347014
> rmse<-sqrt(mean(predic_lstat-testing_data$medv)^2)
> rmse
[1] 0.40708
> #### NOW we try multi linear regression ####
> #selecting only variables
> model_regx_ml<-lm(medv~ rm + lstat,data = Boston)
> #summary
> summary(model_regx_ml)

Call:
lm(formula = medv ~ rm + lstat, data = Boston)

Residuals:
     Min       1Q   Median       3Q      Max
-18.076   -3.516   -1.010    1.909   28.131

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.35827    3.17283  -0.428    0.669
```

```
Console C:/Users/Sakshi/Downloads/
rm           5.09479    0.44447  11.463   <2e-16 ***
lstat       -0.64236    0.04373 -14.689   <2e-16 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.54 on 503 degrees of freedom
Multiple R-squared:  0.6386,    Adjusted R-squared:  0.6371

F-statistic: 444.3 on 2 and 503 DF,  p-value: < 2.2e-16

> #selecting all variables
> model_regx_all<-lm(medv~.,data = training_data)
> #summary
> summary(model_regx_all)

Call:
lm(formula = medv ~ ., data = training_data)

Residuals:
     Min       1Q   Median       3Q      Max
-15.4036  -2.8472  -0.5166    1.8768  24.3219

Coefficients:
```

```
Console C:/Users/Sakshi/Downloads/
                Estimate Std. Error t value Pr(>|t|)
(Intercept)  41.975065   6.171855   6.801 4.45e-11 ***
crim         -0.124709   0.036198  -3.445 0.000639 ***
zn            0.060213   0.016620   3.623 0.000334 ***
indus         0.007313   0.073690   0.099 0.921005
chas          2.366739   1.074791   2.202 0.028308 *
nox         -21.354016   4.722318  -4.522 8.38e-06 ***
rm            3.483738   0.487656   7.144 5.23e-12 ***
age          -0.006623   0.015952  -0.415 0.678241
dis          -1.859822   0.245499  -7.576 3.17e-13 ***
rad           0.337442   0.076949   4.385 1.53e-05 ***
tax          -0.012126   0.004312  -2.812 0.005201 **
ptratio      -0.886151   0.164662  -5.382 1.35e-07 ***
black         0.008751   0.003099   2.824 0.005009 **
lstat        -0.590932   0.060140  -9.826  < 2e-16 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.914 on 353 degrees of freedom
Multiple R-squared:  0.7408,    Adjusted R-squared:  0.7312

F-statistic:  77.6 on 13 and 353 DF,  p-value: < 2.2e-16
```

```
Console C:/Users/Sakshi/Downloads/
> #removing age and indus
> model_regx_selected<-lm(medv~ crim + zn + tax + chas + rm
 + rad + dis + nox +
+                  ptratio + black + lstat,data = training_
data)
> #summary
> summary(model_regx_selected)

Call:
lm(formula = medv ~ crim + zn + tax + chas + rm + rad + dis
 +
    nox + ptratio + black + lstat, data = training_data)

Residuals:
     Min       1Q   Median       3Q      Max
-15.2945  -2.8405  -0.5236   1.9133  24.1194

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)  42.071518   6.151038   6.840 3.48e-11 ***
crim         -0.125202   0.036065  -3.472 0.000581 ***
zn            0.060850   0.016354   3.721 0.000231 ***
tax          -0.011962   0.003853  -3.105 0.002058 **
chas          2.358995   1.063458   2.218 0.027171 *
```

```
Console C:/Users/Sakshi/Downloads/
rm            3.443443   0.474914   7.251 2.62e-12 ***
rad           0.336977   0.073816   4.565 6.89e-06 ***
dis          -1.835740   0.229007  -8.016 1.59e-14 ***
nox         -21.819869   4.404734  -4.954 1.13e-06 ***
ptratio      -0.886786   0.163254  -5.432 1.04e-07 ***
black         0.008685   0.003086   2.814 0.005161 **
lstat        -0.597079   0.057328 -10.415  < 2e-16 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.902 on 355 degrees of freedom
Multiple R-squared:  0.7407,    Adjusted R-squared:  0.7326

F-statistic: 92.17 on 11 and 355 DF,  p-value: < 2.2e-16

> #prediction
> predic_selected<-predict(model_regx_selected, testing_dat
a)
> predic_selected
       2        4        5        8        9
24.810941 28.487147 27.588326 18.208571  9.570032
      19       21       23       24       26
16.633235 12.156905 15.350228 13.317613 13.160220
```

```
Console C:/Users/Sakshi/Downloads/
       27          31          34          35          36
15.153836 10.779911 14.079769 13.284937 24.425422
       42          44          49          53          56
27.898199 24.495516  7.481366 27.511790 31.424660
       67          72          74          80          87
26.412589 21.908279 24.280792 22.834622 22.264722
       90          94          97         100         104
31.199007 30.018000 25.195590 32.635026 20.806728
      106         109         113         114         119
19.134760 23.200215 20.982686 20.720961 20.625168
      123         130         131         132         133
20.143625 13.939241 19.990708 19.368603 20.082864
      134         135         139         143         145
15.650829 13.118455 13.525819 12.450522  6.918186
      148         149         150         151         164
 6.649556  7.844176 13.273424 19.617164 41.129334
      167         170         171         173         174
36.992985 26.391499 22.247576 23.446739 29.628114
      178         186         188         189         190
29.695711 25.010781 34.540533 33.498044 35.295779
      195         202         210         223         224
32.283674 30.199716 16.195250 32.025637 30.216747
      226         228         230         235         236
39.885801 32.812559 32.036462 31.683867 25.589785
```

```
Console C:/Users/Sakshi/Downloads/
> # finding root mean sq. error
> rmse<-sqrt(mean(predic_selected-testing_data$medv)^2)
> rmse
[1] 0.0844029
> #compare actual values and prediction
> plot(testing_data$medv, type = "l", lty = 1.8, col = "gre
en")
> lines(predic_selected,type = "l", col = "blue")
> #since rmse value is still high we need to optimize
> f1=lm(medv~lstat +I(lstat^2),Boston)
> summary(fit1)

Call:
lm(formula = medv ~ lstat, data = Boston)

Residuals:
    Min      1Q  Median      3Q     Max
-15.168  -3.990  -1.318   2.034  24.500

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 34.55384    0.56263   61.41   <2e-16 ***
lstat       -0.95005    0.03873  -24.53   <2e-16 ***
---
```

```
Console C:/Users/Sakshi/Downloads/
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.216 on 504 degrees of freedom
Multiple R-squared:  0.5441,    Adjusted R-squared:  0.5432

F-statistic: 601.6 on 1 and 504 DF,  p-value: < 2.2e-16

> attach(Boston)
The following objects are masked from Boston (pos = 5):

    age, black, chas, crim, dis, indus, lstat,
    medv, nox, ptratio, rad, rm, tax, zn

> f11=lm(medv~poly(lstat,4))
> plot(medv~lstat)
> points(lstat,fitted(f11),col="blue",pch=20)
> f2=lm(medv~rm +I(rm^2),Boston)
> summary(f2)

Call:
lm(formula = medv ~ rm + I(rm^2), data = Boston)

Residuals:
```

```
Console C:/Users/Sakshi/Downloads/
     Min       1Q   Median       3Q      Max
-35.769   -2.752    0.619    3.003   35.464

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  66.0588    12.1040   5.458 7.59e-08 ***
rm          -22.6433     3.7542  -6.031 3.15e-09 ***
I(rm^2)       2.4701     0.2905   8.502  < 2e-16 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.193 on 503 degrees of freedom
Multiple R-squared:  0.5484,    Adjusted R-squared:  0.5466

F-statistic: 305.4 on 2 and 503 DF,  p-value: < 2.2e-16

> attach(Boston)
The following objects are masked from Boston (pos = 3):

    age, black, chas, crim, dis, indus, lstat,
    medv, nox, ptratio, rad, rm, tax, zn

The following objects are masked from Boston (pos = 6):
```

```
Console C:/Users/Sakshi/Downloads/
    age, black, chas, crim, dis, indus, lstat,
    medv, nox, ptratio, rad, rm, tax, zn

> fit22=lm(medv~poly(rm,4))
> plot(medv~rm)
> points(rm,fitted(fit22),col="blue",pch=20)
> #building final model
> fit_final=lm(medv~lstat+crim+rm+dis+black+chas+nox+rad+ta
x+ptratio+I(lstat^2)+I(rm^2))
> summary(fit_final)

Call:
lm(formula = medv ~ lstat + crim + rm + dis + black + chas
 +
    nox + rad + tax + ptratio + I(lstat^2) + I(rm^2))

Residuals:
     Min       1Q   Median       3Q      Max
-26.6989  -2.2722  -0.3181   1.6910  26.5553

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 116.873849   9.277693  12.597  < 2e-16 ***
lstat        -1.283203   0.118452 -10.833  < 2e-16 ***
```

```
Console C:/Users/Sakshi/Downloads/
crim          -0.147969   0.027819  -5.319 1.59e-07 ***
rm           -21.700173   2.790414  -7.777 4.38e-14 ***
dis           -1.023585   0.137258  -7.457 4.01e-13 ***
black          0.006993   0.002260   3.094 0.002087 **
chas           2.418864   0.720510   3.357 0.000848 ***
nox          -14.970947   2.992777  -5.002 7.89e-07 ***
rad            0.229174   0.053487   4.285 2.20e-05 ***
tax           -0.007628   0.002792  -2.732 0.006513 **
ptratio       -0.750995   0.104930  -7.157 3.01e-12 ***
I(lstat^2)     0.021460   0.003234   6.635 8.58e-11 ***
I(rm^2)        1.965340   0.217641   9.030  < 2e-16 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.993 on 493 degrees of freedom
Multiple R-squared:  0.816,     Adjusted R-squared:  0.8115

F-statistic: 182.2 on 12 and 493 DF,  p-value: < 2.2e-16

> #prediction
> predic_fit_final<-predict(fit_final, testing_data)
> predic_fit_final
        2         4         5         8         9
```

```
Console C:/Users/Sakshi/Downloads/
24.541437 32.319154 30.595743 17.280817 13.775862
       19        21        23        24        26
17.679623 12.797727 14.503250 13.283436 13.837019
       27        31        34        35        36
15.337623 11.775017 13.928431 12.801007 23.351744
       42        44        49        53        56
29.913925 25.190487 13.328280 28.390293 30.979321
       67        72        74        80        87
22.521523 21.959939 24.734061 22.991411 20.918836
       90        94        97       100       104
32.106064 28.026497 23.311050 34.273404 19.034815
      106       109       113       114       119
17.190405 21.236383 18.972000 18.610446 18.898730
      123       130       131       132       133
17.722179 13.623347 18.802906 18.336454 19.330663
      134       135       139       143       145
15.174836 12.913323 12.909453 15.364537 13.528252
      148       149       150       151       164
13.375352 12.643265 13.820939 18.555713 49.633752
      167       170       171       173       174
42.083901 24.328545 20.529178 21.643495 27.532296
      178       186       188       189       190
29.140250 22.479055 31.503254 31.811673 34.365593
      195       202       210       223       224
```

```
Console C:/Users/Sakshi/Downloads/
30.826025 26.314331 17.525242 30.514806 28.996522
      226       228       230       235       236
50.181092 33.109590 32.561044 30.857076 23.862476
      240       241       242       244       246
27.543744 25.318943 21.480748 27.999905 13.453547
      247       249       252       255       256
20.421277 21.213634 27.621745 23.091245 20.150238
      268       270       271       273       278
44.648534 22.973444 20.351297 27.177260 34.608795
      293       303       306       309       313
29.843539 26.895825 27.903120 29.972619 21.845155
      316       320       321       330       334
20.453586 20.045103 25.417491 25.342405 24.284553
      340       344       346       348       351
21.333570 26.345849 18.075477 24.006235 22.137552
      353       355       356       358       359
18.194472 15.573615 18.273562 20.970563 21.142846
      364       371       374       376       380
18.979165 36.523009 14.222700 23.648009 14.070012
      382       385       398       399       407
15.597706 12.737880 14.845144  8.261543 16.173409
      409       415       416       418       420
13.999965  7.865720 10.167079  8.561722 13.484192
      423       425       426       441       442
```

```
17.495289 14.396054  9.555753 11.002167 14.960418
      449       453       459       471       473
15.286007 16.488850 15.479952 18.179679 20.446842
      477       487       494       497       498
18.063294 18.054750 20.253422 14.554289 18.284055
      499       502       503       505
19.954535 22.823904 22.022212 26.833388
> # finding root mean sq. error
> rmse<-sqrt(mean(predic_fit_final-testing_data$medv)^2)
> rmse
[1] 0.1883437
> #compare actual values and prediction
> plot(testing_data$medv, type = "l", lty = 1.8, col = "gre
en")
> lines(predic_fit_final,type = "l", col = "blue")
>
```

VISUALIZATION

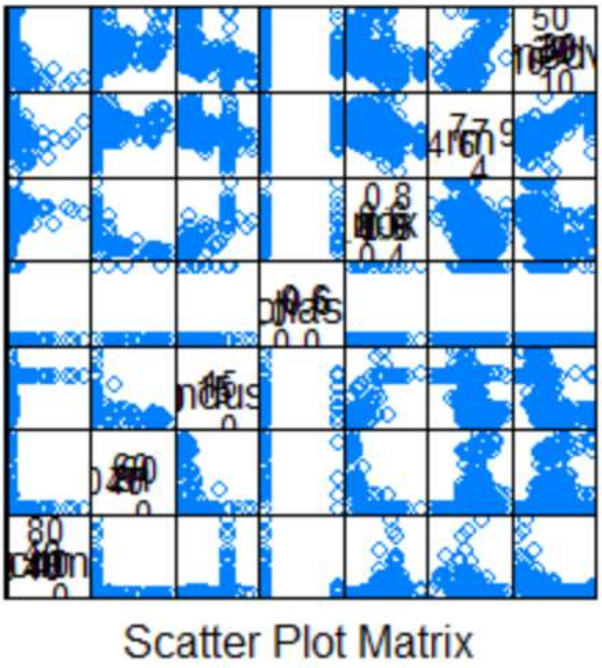

Chart 1: Scatterplot matrix of the input dataset

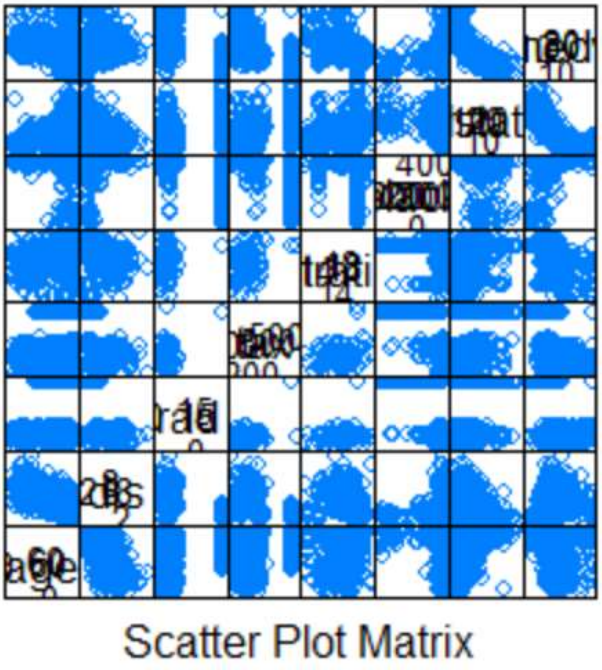

Chart 2: Scatterplot matrix of the input dataset

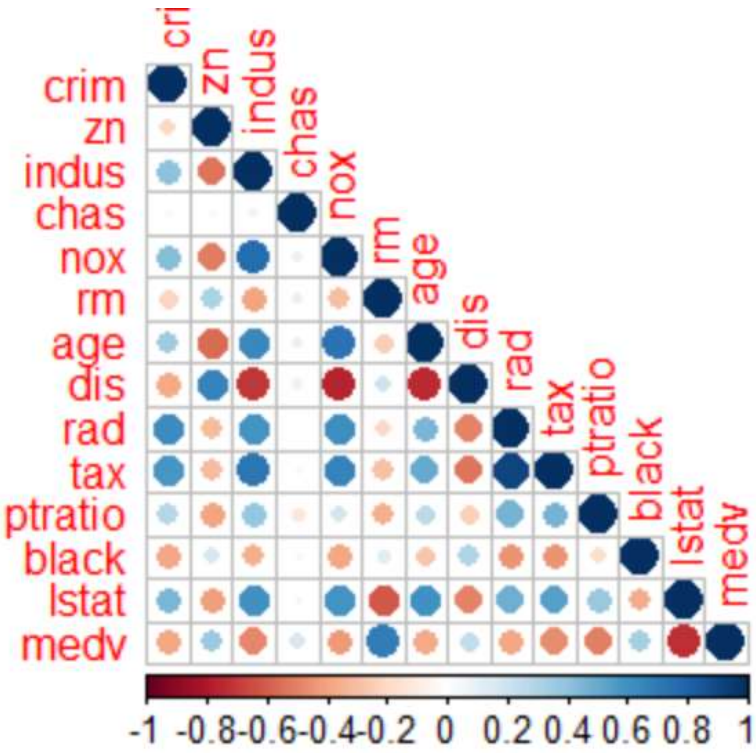

Chart 3: Correlation between the variables

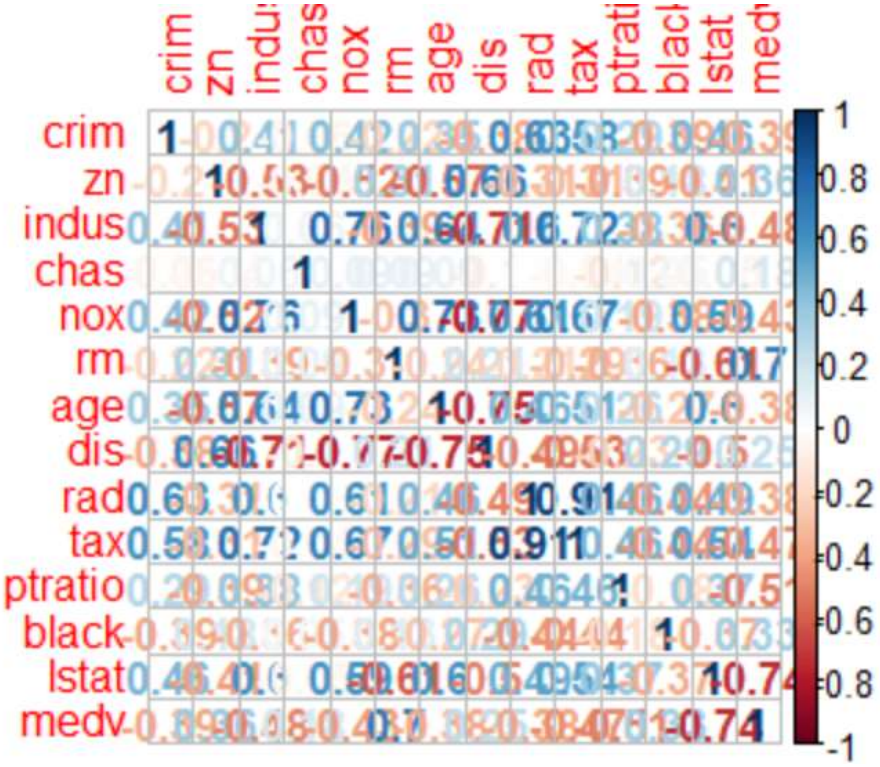

Chart 4: Numerical representation of correlation between the variables

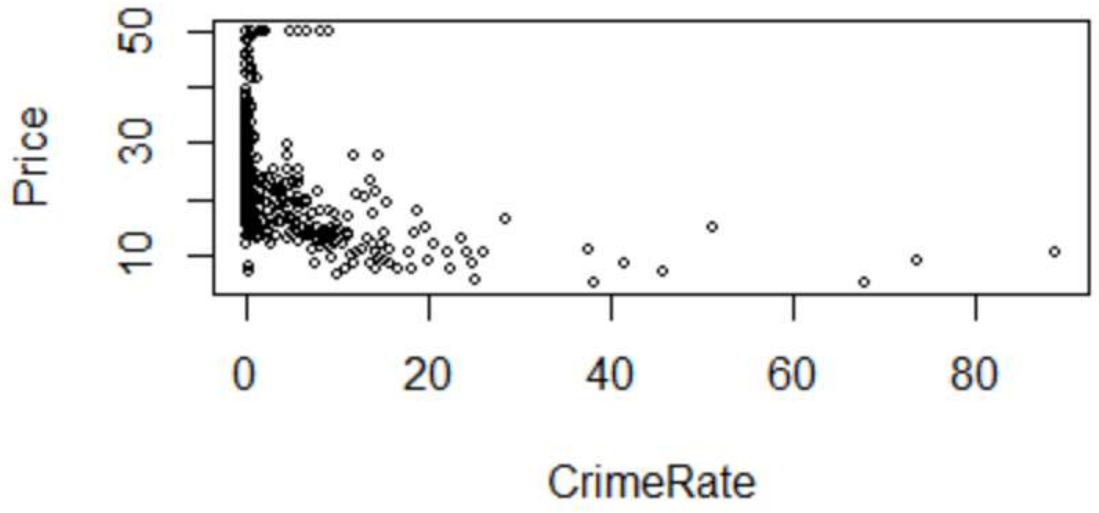

Chart 5: Correlation between price and crime rate

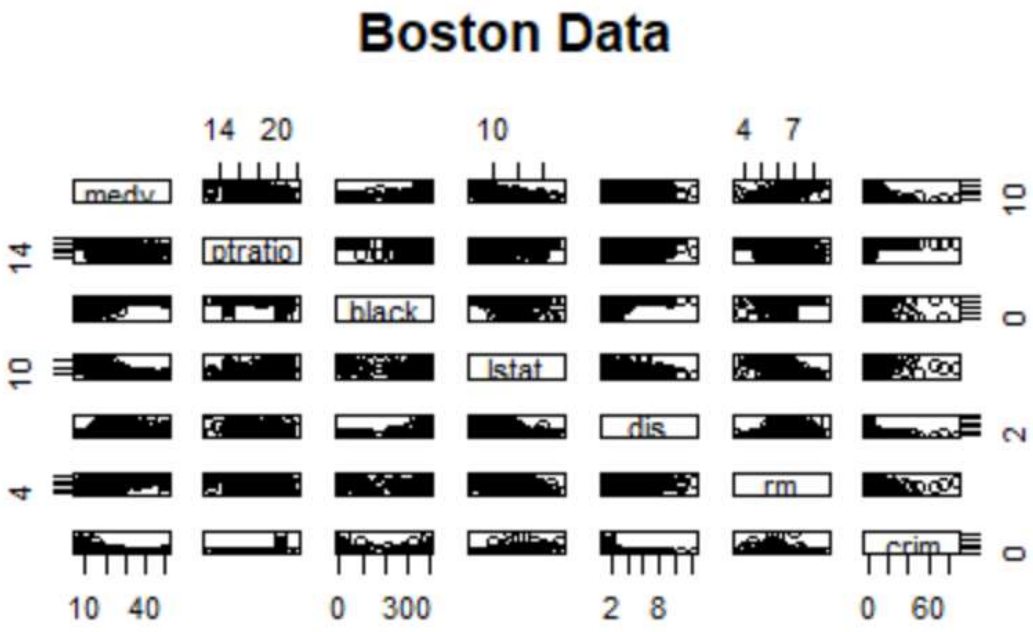

Chart 6: Correlation of price with all the feature variables

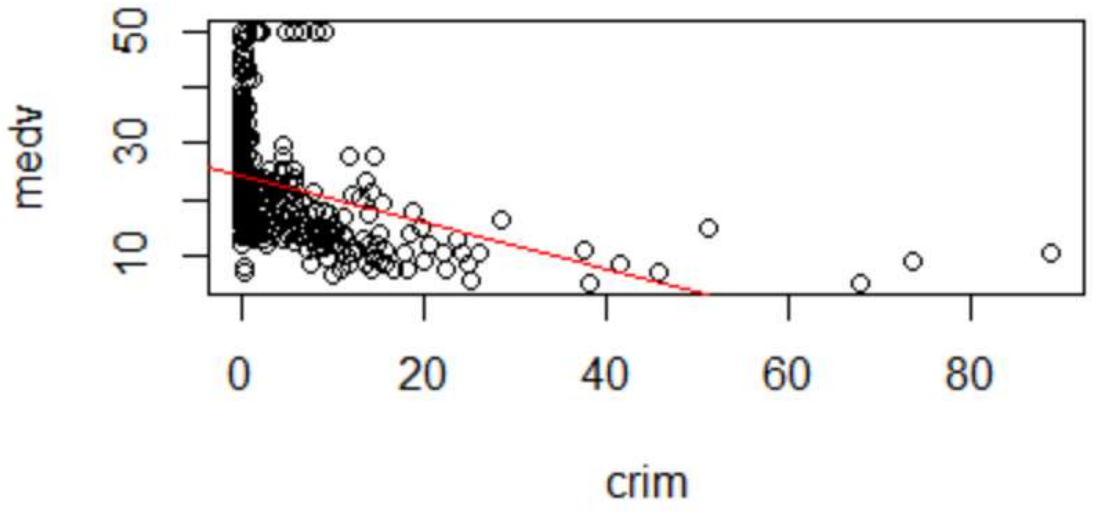

Chart 7: Linear regression fit of median price value with variable crime rate

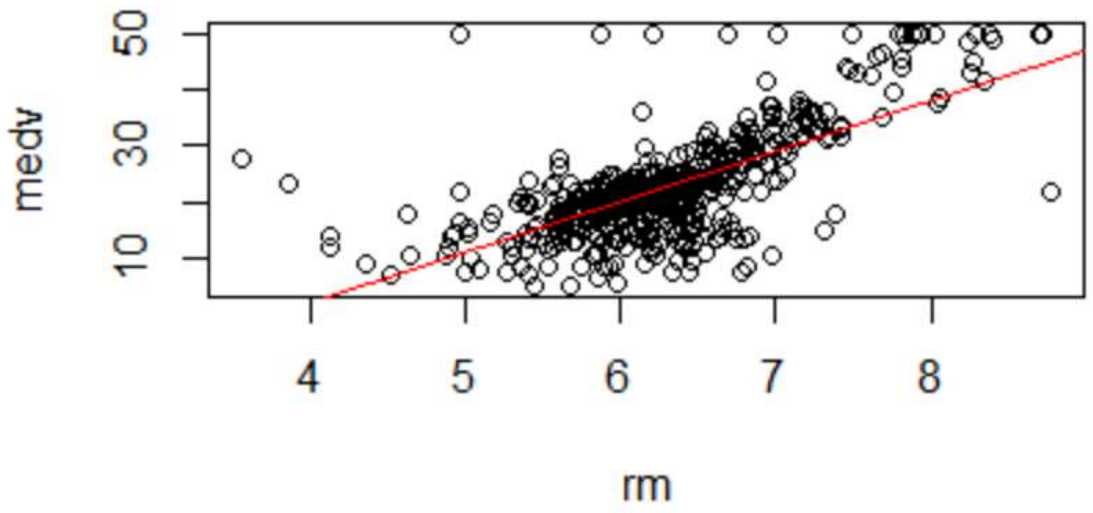

Chart 8: Linear regression fit of median price value with variable RM

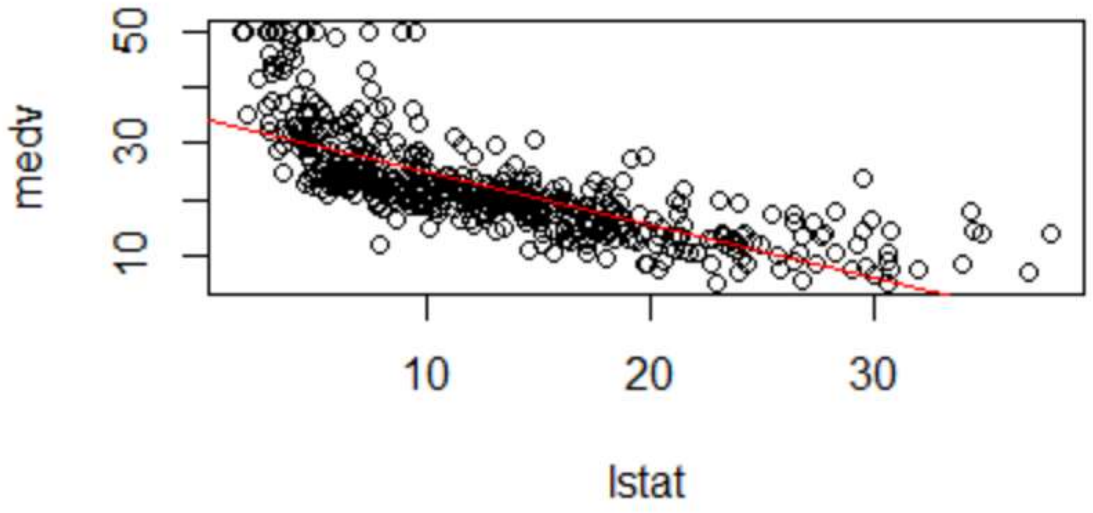

Chart 9: Linear regression fit of median price value with variable LSTAT

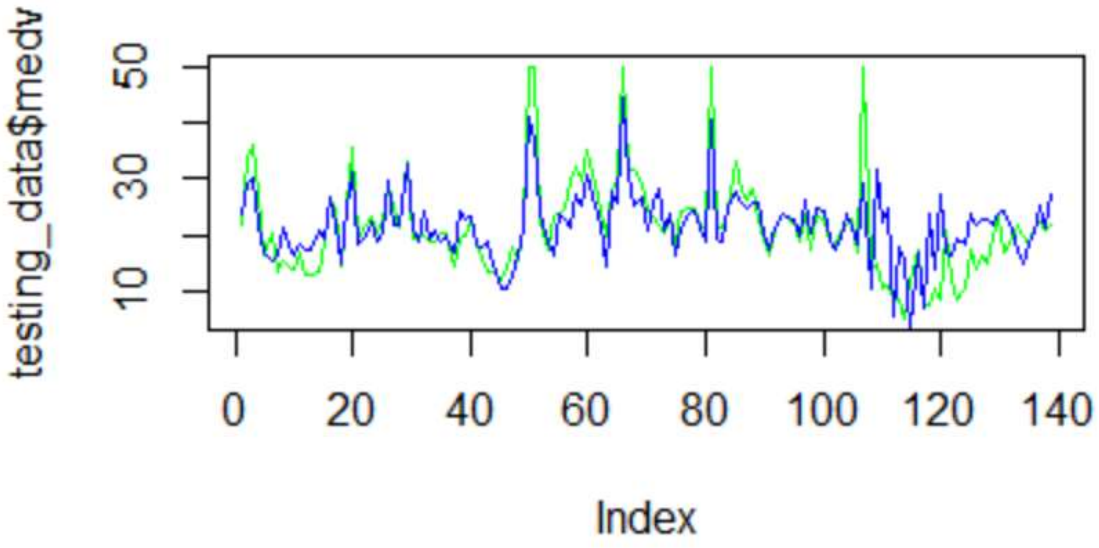

Chart 10: Prediction chart using only one feature variable (RM)

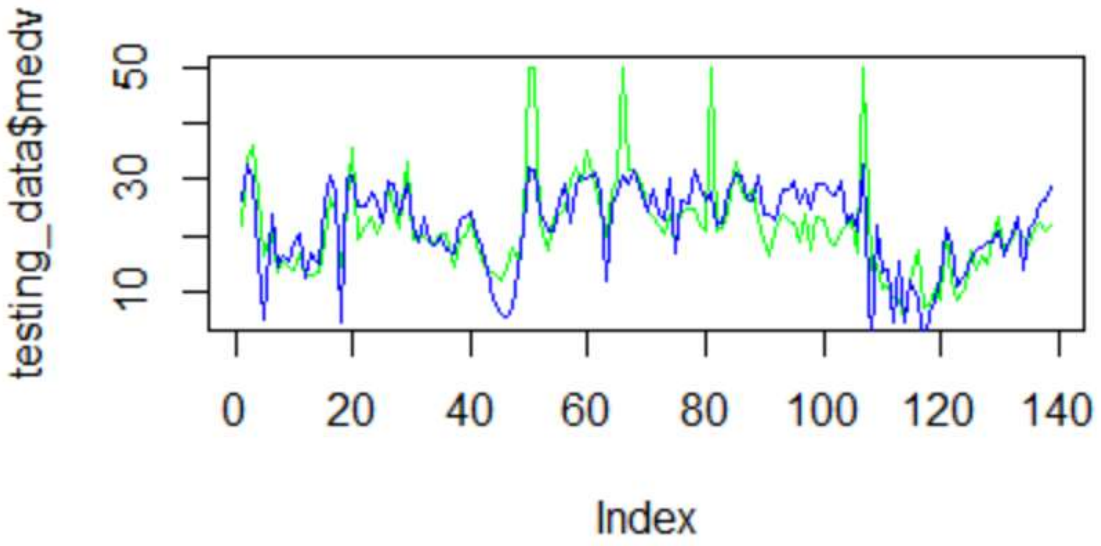

Chart 11: Prediction chart using only one feature variable (LSTAT)

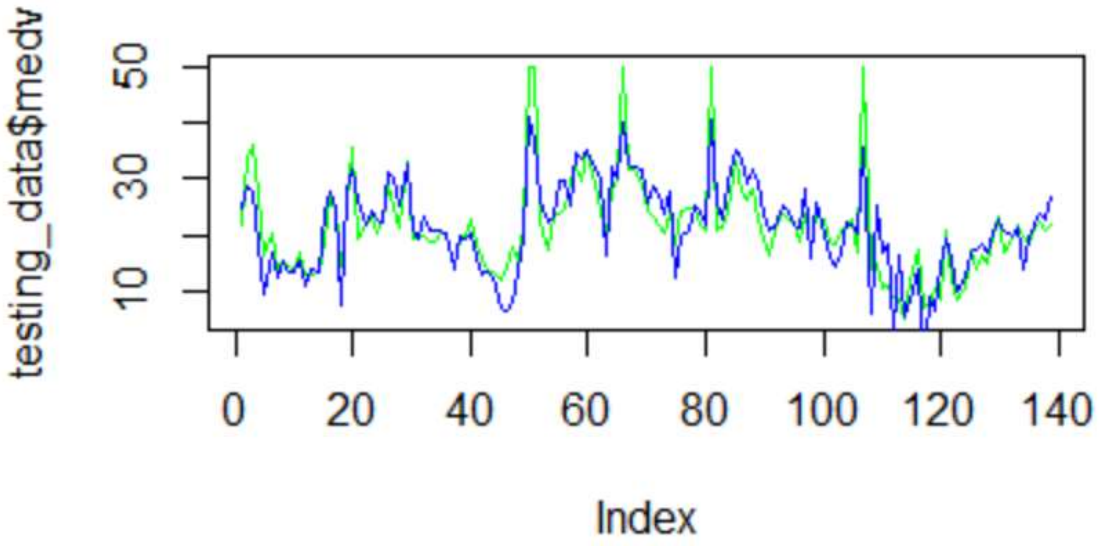

Chart 12: Multi-linear prediction model developed using RM, LSTAT and all other variables except AGE and INDUS

Note: Its root mean square error is still high. So, to adjust it, we will now apply polynomial regression of RM and LSTAT.

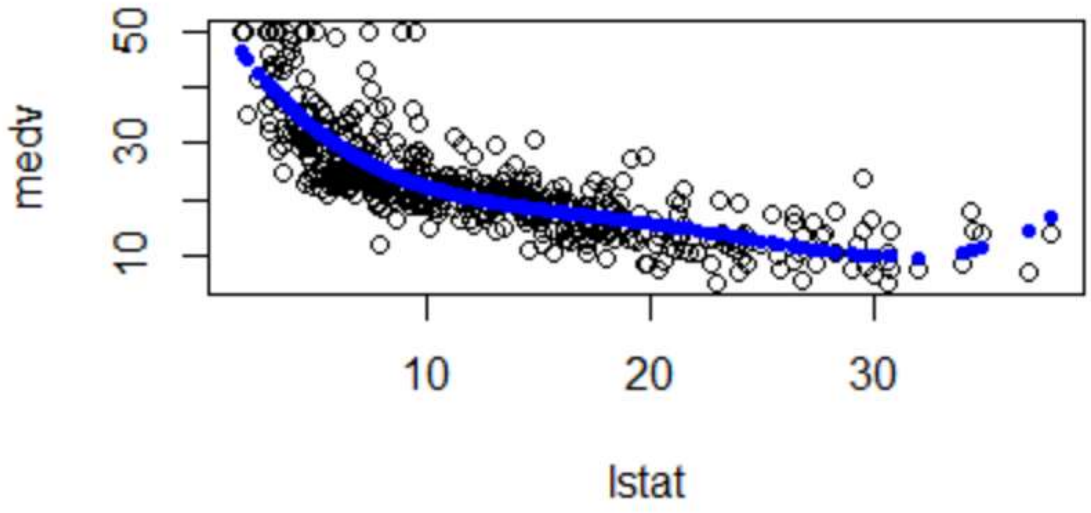

Chart 13: Polynomial regression for LSTAT

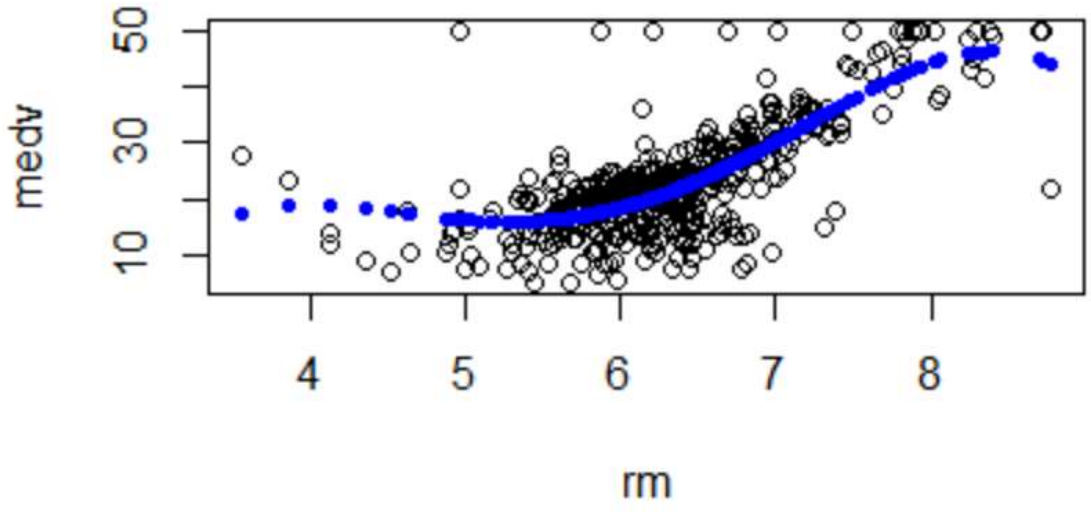

Chart 14: Polynomial regression for RM

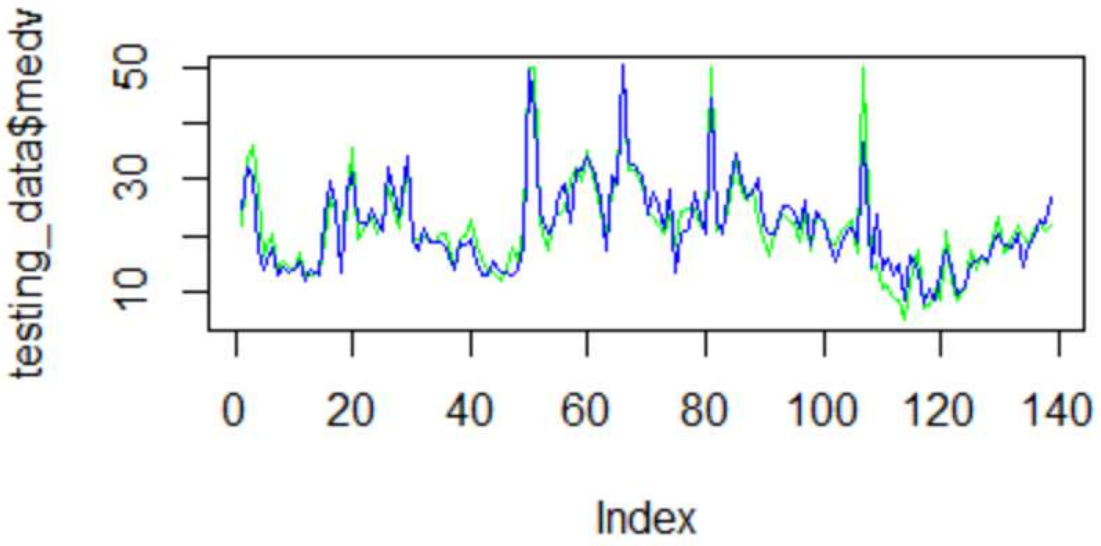

Chart 15: Final trained prediction model with a much accurate root mean square error

CONCLUSION

The goal of this report was to determine the neighbourhood attributes that best explained variation in house pricing. Various statistical techniques were used to eliminate predictors and extraneous observations. In examining the final model, one finds – quite reasonably – that house prices are higher in areas with lower crime and lower pupil-teacher ratios.

This suggests that people would prefer to live further away from their place of employment if it meant lower levels of pollution, which is an interesting point to consider. On a concluding note, it is important to note that the data for this report was collected several decades ago. In the years since, there is no doubt that pollution levels have risen and it would be interesting to examine the ways in which that affects house pricing in Boston today.

BIBLIOGRAPHY

https://towardsdatascience.com/machine-learning-project-predicting-boston-house-prices-with-regression-b4e47493633d

https://rstudio-pubs-static.s3.amazonaws.com/364346_811c9012a14847428c9b1fc1e956431a.html

https://rstudio-pubs-static.s3.amazonaws.com/364346_811c9012a14847428c9b1fc1e956431a.html

https://rpubs.com/chocka314/251613