Re: Own Project - Referencing Metropolitan Housing Dataset to Predict Home Pricing

Table of Contents

PHY125.9X | CAPSTONE OWN_PROJECT: Metropolitan Housing Dataset
By Nik S.

Jan8, 2020

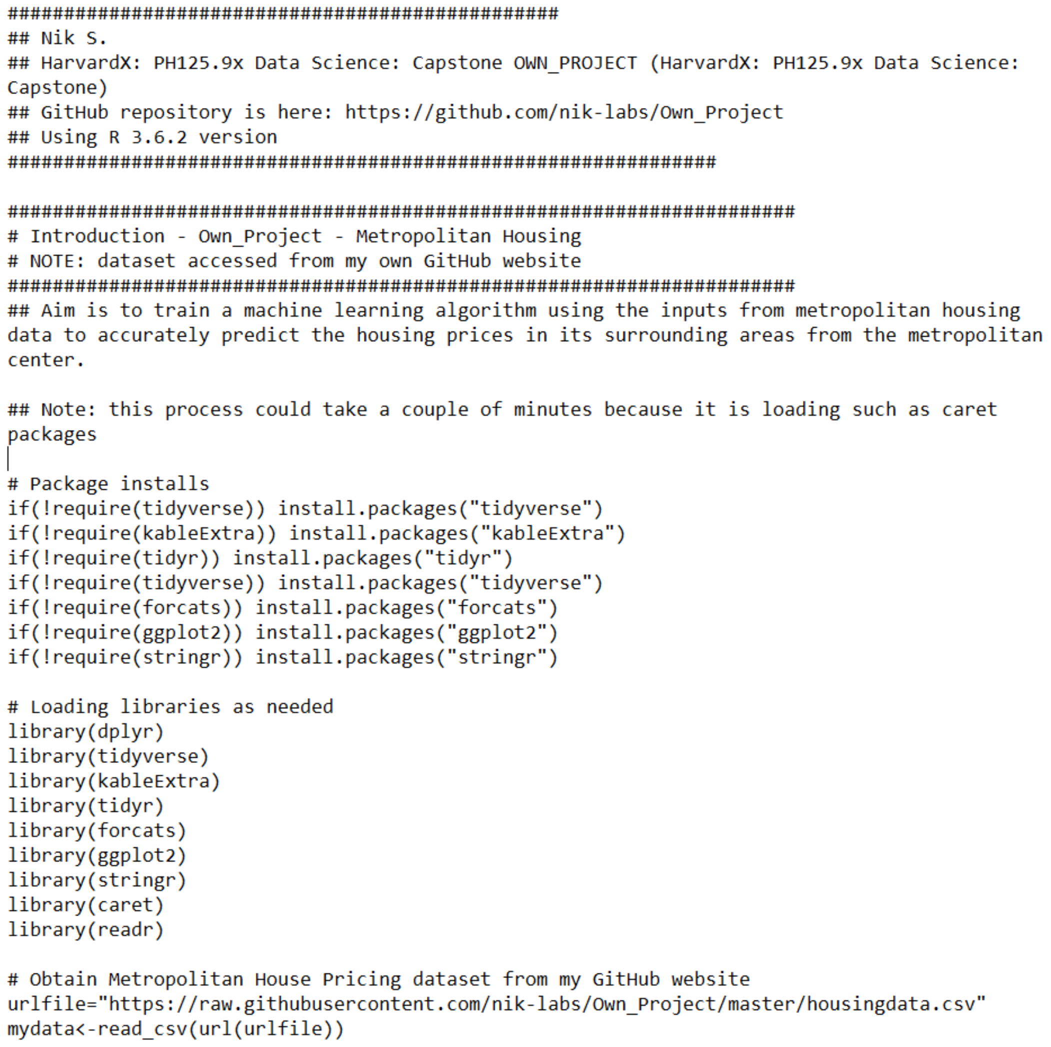
Reference List: Irizarry, A. R. (2015). Introduction to Data Science

1. Introduction

NOTE: GitHub repository is here: https://github.com/nik-labs/Own_Project

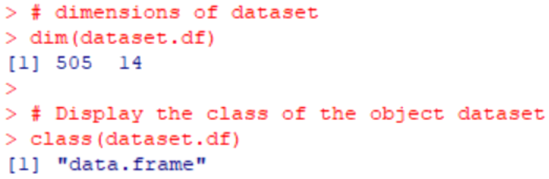
Project Objective:
The focus is to train a machine learning algorithm using the inputs from metropolitan housing dataset to accurately predict the housing prices in its surrounding areas from the metropolitan center. This approach will look to leverage the best model for predicting actual home prices. To clarify, RMSE measures accuracy by stating differences between prediction values and observed values.

The Metropolitan Housing dataset was uploaded to my GitHub website and downloaded for there directly.



```
##################################################
## Nik S.
## HarvardX: PH125.9x Data Science: Capstone OWN_PROJECT (HarvardX: PH125.9x Data Science:
Capstone)
## GitHub repository is here: https://github.com/nik-labs/Own_Project
## Using R 3.6.2 version
#################################################################

########################################################################
# Introduction - Own_Project - Metropolitan Housing
# NOTE: dataset accessed from my own GitHub website
########################################################################
## Aim is to train a machine learning algorithm using the inputs from metropolitan housing
data to accurately predict the housing prices in its surrounding areas from the metropolitan
center.

## Note: this process could take a couple of minutes because it is loading such as caret
packages

# Package installs
if(!require(tidyverse)) install.packages("tidyverse")
if(!require(kableExtra)) install.packages("kableExtra")
if(!require(tidyr)) install.packages("tidyr")
if(!require(tidyverse)) install.packages("tidyverse")
if(!require(forcats)) install.packages("forcats")
if(!require(ggplot2)) install.packages("ggplot2")
if(!require(stringr)) install.packages("stringr")

# Loading libraries as needed
library(dplyr)
library(tidyverse)
library(kableExtra)
library(tidyr)
library(forcats)
library(ggplot2)
library(stringr)
library(caret)
library(readr)

# Obtain Metropolitan House Pricing dataset from my GitHub website
urlfile="https://raw.githubusercontent.com/nik-labs/Own_Project/master/housingdata.csv"
mydata<-read_csv(url(urlfile))
```

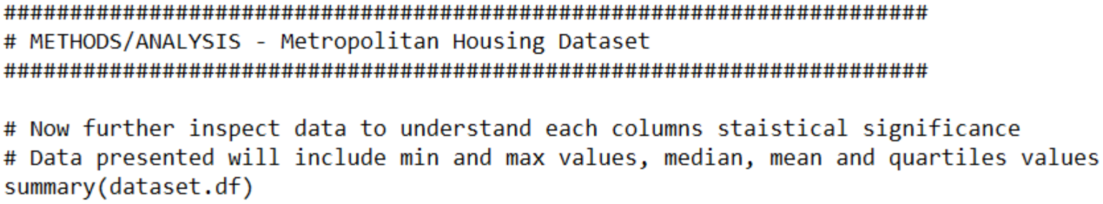
```
# Rename mydata to dataframe called dataset
dataset.df <- mydata

# set the column names in the dataset
colnames(dataset.df) <- c
("CRIM","ZN","INDUS","CHAS","NOX","RM","AGE","DIS","RAD","TAX","PTRATIO","B","LSTAT","MEDV")
## Definitions of Columns are as follows:
# Column 1: CRIM = per capita crime rate by town
# Column 2: ZN = proportion of residential land zoned for lots > 25,000 sq.ft
# Column 3: INDUS = proportion of non-retail business acres per town
# Column 4: CHAS = Charles River dummy variable (1 if tract bounds river; else 0)
# Column 5: NOX = nitric oxides concentration (10 ppm)
# Column 6: RM = average number of rooms per dwelling
# Column 7: AGE = proportion of owner-occupied units built prior to 1940
# Column 8: DIS = weighted distances to five Boston employment centres
# Column 9: RAD = index of accessibility to radial highways
# Column 10: TAX = full-value property-tax rate per $10k
# Column 11: PTRATIO = pupil-teacher ratio by town
# Column 12: B = 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
# Column 13: LSTAT = % lower status of the population
# Column 14: MEDV = Median value of owner-occupied homes in $1000's
```



```
> # dimensions of dataset
> dim(dataset.df)
[1] 505  14
>
> # Display the class of the object dataset
> class(dataset.df)
[1] "data.frame"
```

To make an accurate prediction of home pricing, the dataset is segmented into a training subset called “training” to train the algorithm, and a “validation” subset to test pricing Testing will be performed on the “training” subset (please refer to pg5).

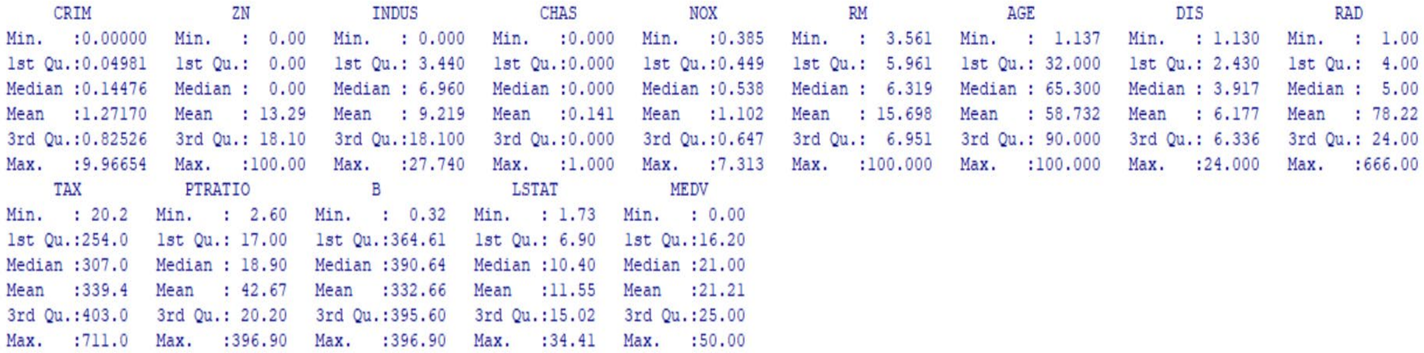
2. Methods / Analysis



```
##########################################################################
# METHODS/ANALYSIS - Metropolitan Housing Dataset
##########################################################################

# Now further inspect data to understand each columns staistical significance
# Data presented will include min and max values, median, mean and quartiles values
summary(dataset.df)
```

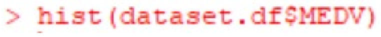
In Exhibit 1 below, we see how distribution of all individual variables that make up the dataset for each column (e.g., 1st and 3rd quartile values, etc.).

Exhibit 1: Statistical Details per Variable



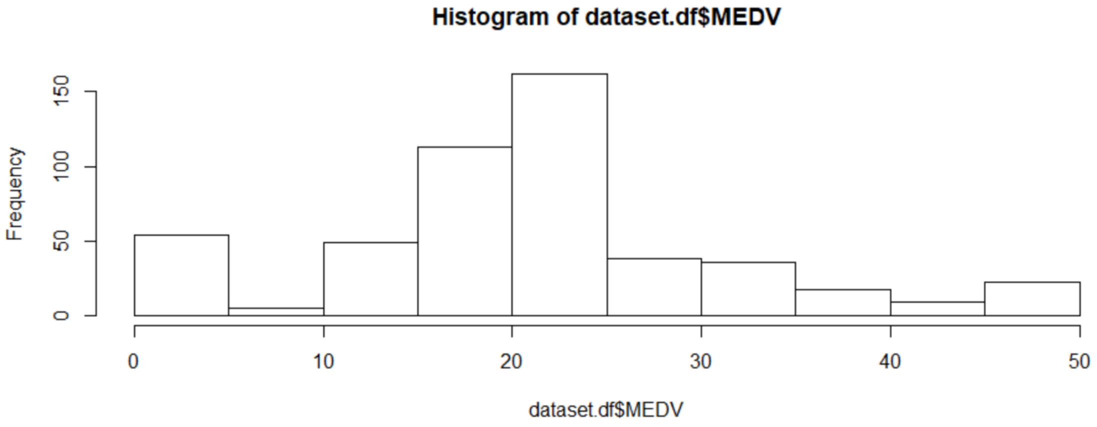
```
      CRIM              ZN             INDUS             CHAS             NOX              RM               AGE               DIS              RAD
 Min.   :0.00000   Min.   :  0.00   Min.   : 0.000   Min.   :0.000   Min.   :0.385   Min.   :  3.561   Min.   :  1.137   Min.   : 1.130   Min.   :  1.00
 1st Qu.:0.04981   1st Qu.:  0.00   1st Qu.: 3.440   1st Qu.:0.000   1st Qu.:0.449   1st Qu.:  5.961   1st Qu.: 32.000   1st Qu.: 2.430   1st Qu.:  4.00
 Median :0.14476   Median :  0.00   Median : 6.960   Median :0.000   Median :0.538   Median :  6.319   Median : 65.300   Median : 3.917   Median :  5.00
 Mean   :1.27170   Mean   : 13.29   Mean   : 9.219   Mean   :0.141   Mean   :1.102   Mean   : 15.698   Mean   : 58.732   Mean   : 6.177   Mean   : 78.22
 3rd Qu.:0.82526   3rd Qu.: 18.10   3rd Qu.:18.100   3rd Qu.:0.000   3rd Qu.:0.647   3rd Qu.:  6.951   3rd Qu.: 90.000   3rd Qu.: 6.336   3rd Qu.: 24.00
 Max.   :9.96654   Max.   :100.00   Max.   :27.740   Max.   :1.000   Max.   :7.313   Max.   :100.000   Max.   :100.000   Max.   :24.000   Max.   :666.00
      TAX            PTRATIO              B              LSTAT            MEDV
 Min.   : 20.2   Min.   :  2.60   Min.   :  0.32   Min.   : 1.73   Min.   : 0.00
 1st Qu.:254.0   1st Qu.: 17.00   1st Qu.:364.61   1st Qu.: 6.90   1st Qu.:16.20
 Median :307.0   Median : 18.90   Median :390.64   Median :10.40   Median :21.00
 Mean   :339.4   Mean   : 42.67   Mean   :332.66   Mean   :11.55   Mean   :21.21
 3rd Qu.:403.0   3rd Qu.: 20.20   3rd Qu.:395.60   3rd Qu.:15.02   3rd Qu.:25.00
 Max.   :711.0   Max.   :396.90   Max.   :396.90   Max.   :34.41   Max.   :50.00
```

In Exhibit 2 below, there is a depiction of a histogram for MEDV (median home pricing). It can be observed that there is a slight uneven distribution of values (skewed to the right). A natural log may be useful to normalize the data when modelling is employed in later sections.



```
> hist(dataset.df$MEDV)
```

Exhibit 2: Histogram of MEDV (median value of home pricing)



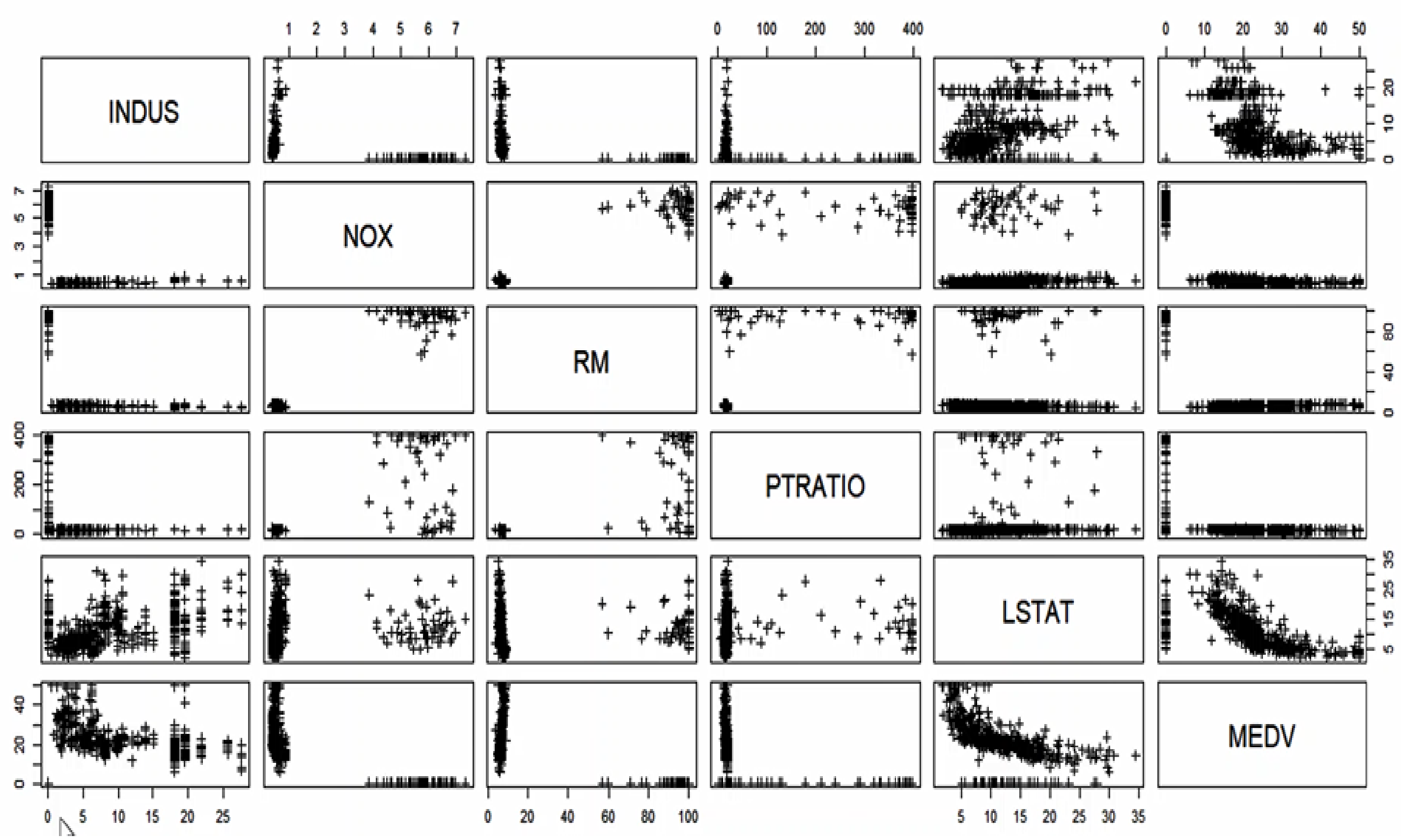
In Exhibit 3 below, this is a matrix scatterplot to illustrate whether there are any correlations between the variables, specifically for INDUS, NOX, RM, PTRATIO. LSTAT, and MDEV.

Here are some observations:

1. LSTAT and MEDV are strongly negatively correlated = as the % of lower status of the population increases, the median value of owner -occupied homes in $1000s decreases (no “gentrification” happening here)
2. LSTAT and INDUS are strongly positively correlated = as the % of lower status of the population increases, so does the proportion of non-retail business acres per town (perhaps businesses are biased not to do business with lower income populated areas?)
3. INDUS and MEDV are strongly positively correlated = as the median value of owner -occupied homes in $1000s increases, so does the proportion of non-retail business acres per town (perhaps businesses are biased to do business with high income populated areas?)

```
# Employ scatterplot to view attributes with an illustration of correlation or lack there
of.
plot(dataset.df[,c(3,5,6,11,13,14)],pch=3)
```

Exhibit 3: Matrix Scatterplot to view attributes across all variables

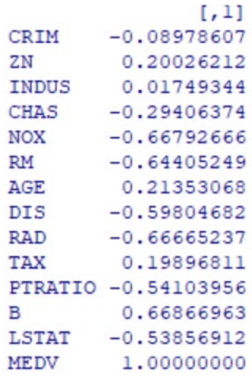


In Exhibit 4 below, the correlations of each independent variable with the dependent variable are illustrated. It is observed that the number of minorities, B, has the strongest positive correlation with the median value of the housing price, MEDV. This means that when the number of minorities in the area goes up, so does the median housing prices.

Conversely, both nitric oxides concentration, NOX and accessibility to radial highways, RAD, have strong negative correlation with the median value of the housing price, MEDV. This means that when the accessibility of radial highways goes up, the median value of the housing price, MEDV, decreases.

```
# Correlation of each independent variable with the dependent variable and to also see
whether a feature has near zero variance within each column
cor(dataset.df,dataset.df$MEDV)
```

Exhibit 4: Variable Correlation

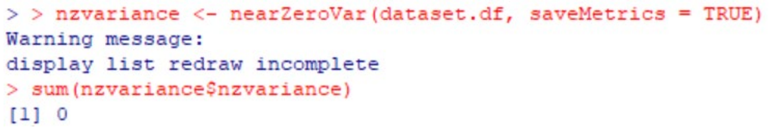


```
               [,1]
CRIM    -0.08978607
ZN       0.20026212
INDUS    0.01749344
CHAS    -0.29406374
NOX     -0.66792666
RM      -0.64405249
AGE      0.21353068
DIS     -0.59804682
RAD     -0.66665237
TAX      0.19896811
PTRATIO -0.54103956
B        0.66866963
LSTAT   -0.53856912
MEDV     1.00000000
```

In Exhibit 5 below, this performs a check to see whether there are unique values (e.g., zero variance predictors) which may skew the dataset. Examples may include having very few unique values relative to the number of samples and the ratio of the frequency of the most common value to the frequency of the second most common value is large. In this case, there are no variables with near zero variances.

```
# Calulate/confirm whether a variable has a near zero variance
nzvariance <- nearZeroVar(dataset.df, saveMetrics = TRUE)
sum(nzvarviance$nzvariance)
```

Exhibit 5: Near Zero Variance



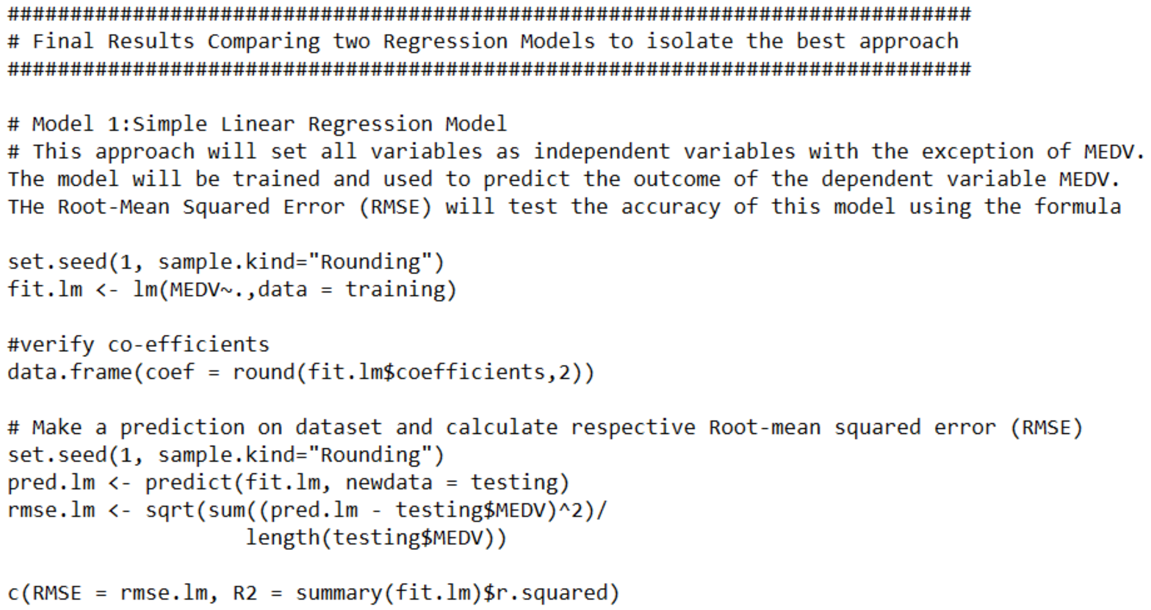
```
> > nzvariance <- nearZeroVar(dataset.df, saveMetrics = TRUE)
Warning message:
display list redraw incomplete
> sum(nzvariance$nzvariance)
[1] 0
```

Now looking to validate 30% of the dataset and train the balance population set.

```
# Validation set will be 30% of Metropolitan Housing data
dataset.scale <- cbind(scale(dataset.df[1:13]), dataset.df[14])

set.seed(1, sample.kind="Rounding")
#Do data partitioning
inTrain <- createDataPartition(y = dataset.scale$MEDV, p = 0.70, list = FALSE)
training <- dataset.scale[inTrain,]
testing <- dataset.scale[-inTrain,]
```

Reference List: Irizarry, A. R. (2015). Introduction to Data Science

3. Results

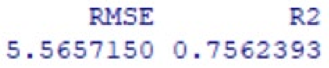


```
##############################################################################
# Final Results Comparing two Regression Models to isolate the best approach
##############################################################################

# Model 1:Simple Linear Regression Model
# This approach will set all variables as independent variables with the exception of MEDV.
The model will be trained and used to predict the outcome of the dependent variable MEDV.
THe Root-Mean Squared Error (RMSE) will test the accuracy of this model using the formula

set.seed(1, sample.kind="Rounding")
fit.lm <- lm(MEDV~.,data = training)

#verify co-efficients
data.frame(coef = round(fit.lm$coefficients,2))

# Make a prediction on dataset and calculate respective Root-mean squared error (RMSE)
set.seed(1, sample.kind="Rounding")
pred.lm <- predict(fit.lm, newdata = testing)
rmse.lm <- sqrt(sum((pred.lm - testing$MEDV)^2)/
                   length(testing$MEDV))

c(RMSE = rmse.lm, R2 = summary(fit.lm)$r.squared)
```

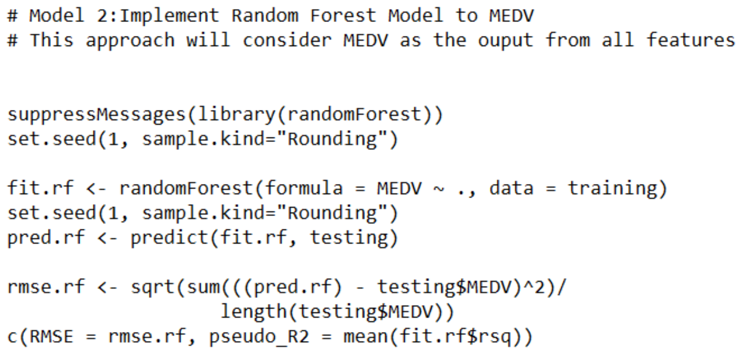
We have leveraged the linear regression model incorporating MEDV as the dependent variable and setting the balance of variables as independent variables. In Exhibit 6 below, the Root-Mean Squared Error (RMSE) has been used to test the accuracy of this model and as we can see, the value is 5.565 (which is quite high). We will now look to baseline this model with another model called the Random Forest Model.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_{pred,i} - y_{act,i})^2}{n}}$$

Exhibit 6: Linear Regression Model generated RMSE and R-squared



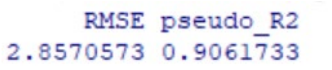
```
     RMSE        R2 
5.5657150 0.7562393 
```



```
# Model 2:Implement Random Forest Model to MEDV
# This approach will consider MEDV as the ouput from all features


suppressMessages(library(randomForest))
set.seed(1, sample.kind="Rounding")

fit.rf <- randomForest(formula = MEDV ~ ., data = training)
set.seed(1, sample.kind="Rounding")
pred.rf <- predict(fit.rf, testing)

rmse.rf <- sqrt(sum(((pred.rf) - testing$MEDV)^2)/
                  length(testing$MEDV))
c(RMSE = rmse.rf, pseudo_R2 = mean(fit.rf$rsq))
```

In Exhibit 7, the Random Forest model is being employed. Benchmarking it with the Linear Regression model above (please see Exhibit 6), we see that the RMSE value has almost been cut in half from 5.656 to now 2.857. It can be concluded from these results that the Random Forest Model is the most accurate of the two models.

Exhibit 7: Forest Model generated RMSE and R-squared

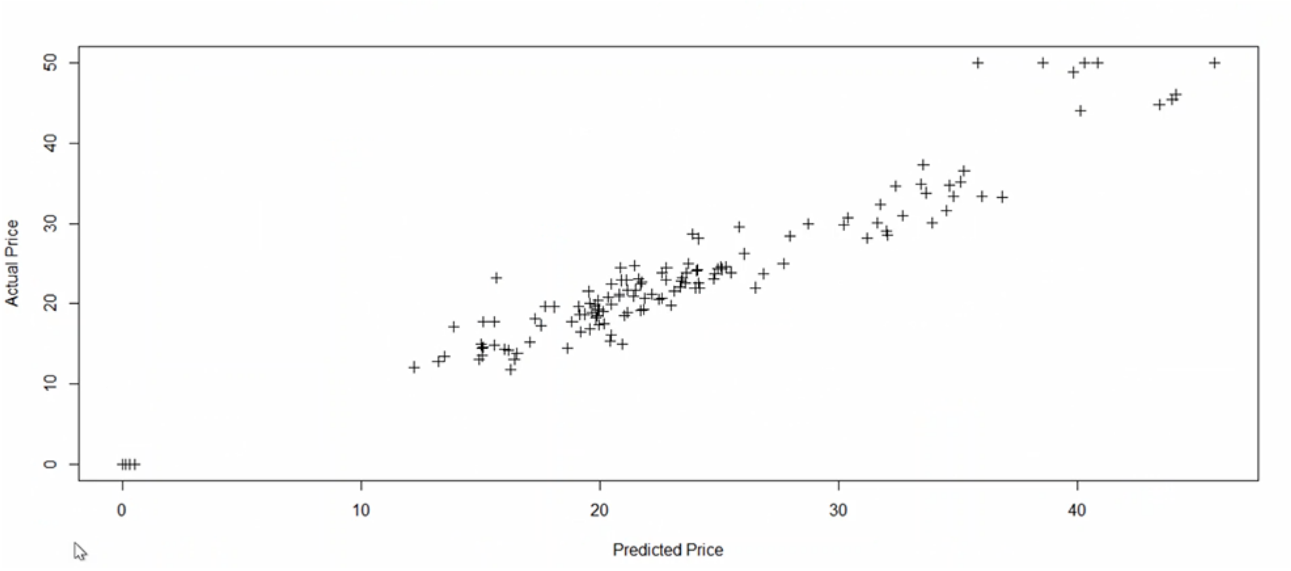


```
     RMSE pseudo_R2
2.8570573 0.9061733
```

In Exhibit 8, the correlation between actual home prices and estimated home prices are shown. It can be observed that by using the Random Forest model, this model generates a great predictor of home prices and the two variables are strongly positively correlated.

```
#Plotting Actual Price vs. Estimated Price
plot(pred.rf,testing$MEDV, xlab = "Predicted Price", ylab = "Actual Price", pch = 3)
```

Exhibit 8: Actual vs. Estimated Home Pricing



4. Conclusion

It was observed that throughout this project, there was a test and learn approach to use two predictive models to seek the most accurate method, which in this case is the random forest model. This model achieved a significantly lower Root Mean Squared Error (RMSE) of 2.857 (almost half in value) versus the linear regression model yielding a RMSE of 5.565. Also, this observation was reinforced by the observations yielded in Exhibit 8, whereby the random forest model generated accurate predicted results against actual home price data.