Project_1_regression

Muhammad Zubair

10/14/2021

Link to the dataset: https://www.kaggle.com/zaynshahbaz/pakistan-car-prices

```r
# Reading in the data
df <- read.csv("updated_pakwheels.csv")
```

Data Exploration

```r
# Viewing the first 5 rows of dataset
head(df)
```

```
##     Ad.No                                       Name   Price Model.Year
## 1 4096758                       Toyota Vitz F 1.0 2017 2385000       2017
## 2 4168305 Toyota Corolla GLi Automatic 1.3 VVTi 2019  111000       2019
## 3 4168298                         Suzuki Alto VXL 2019 1530000       2019
## 4 4168307                         Suzuki Alto VXR 2019 1650000       2019
## 5 4168306                 Toyota Corolla XLi VVTi 2010 1435000       2010
## 6 4168303                   Honda Civic 1.5 RS Turbo 2017 3850000       2017
##                           Location Mileage Registered.City Engine.Type
## 1         G- 8, Islamabad Islamabad    9869   Un-Registered           1
## 2                      Peshawar KPK   11111       Islamabad           1
## 3       Akora Khattak, Nowshera KPK   17500   Un-Registered           1
## 4    Abdullahpur, Faisalabad Punjab    9600          Lahore           1
## 5   9th Avenue, Islamabad Islamabad  120000       Islamabad           1
## 6  Peshawar Road, Rawalpindi Punjab   22000       Islamabad           1
##   Engine.Capacity Transmission  Color Assembly Body.Type
## 1            1000            1 Silver Imported         1
## 2            1300            1  White    Local         2
## 3             660            1  White    Local         1
## 4             660            2  White    Local         1
## 5            1300            2  Black    Local         2
## 6            1500            1  Black    Local         2
##
## 1                                         ABS, AM/FM Radio, Air Bags, Air Conditioning, CD Player, DVD
## 2                                                                ABS, AM/FM Radio, Air Bags, Air Condition
## 3                                                                   ABS, AM/FM Radio, Air Bags, Air Condi
## 4
## 5
```

```
## 6  ABS, AM/FM Radio, Air Bags, Air Conditioning, Alloy Rims, CD Player, Cruise Control, DVD Player, 
##   Last.Updated
## 1    11-Jul-20
## 2    12-Jul-20
## 3    12-Jul-20
## 4    12-Jul-20
## 5    12-Jul-20
## 6    12-Jul-20
##                                                                                           URL
## 1    https://www.pakwheels.com/used-cars/toyota-vitz-2017-for-sale-in-islamabad-4096758
## 2  https://www.pakwheels.com/used-cars/toyota-corolla-2019-for-sale-in-peshawar-4168305
## 3     https://www.pakwheels.com/used-cars/suzuki-alto-2019-for-sale-in-nowshera-4168298
## 4   https://www.pakwheels.com/used-cars/suzuki-alto-2019-for-sale-in-faisalabad-4168307
## 5 https://www.pakwheels.com/used-cars/toyota-corolla-2010-for-sale-in-islamabad-4168306
## 6   https://www.pakwheels.com/used-cars/honda-civic-2017-for-sale-in-rawalpindi-4168303
```

```r
# Viewing the last 5 rows of data
tail(df)
```

```
##         Ad.No                                       Name   Price Model.Year
## 46018 3806954      Honda Civic Oriel 1.8 i-VTEC CVT 2017 3300000       2017
## 46019 3448128                   Honda Vezel Hybrid X 2015 3400000       2015
## 46020 3737684                          Toyota Aqua S 2015 2450000       2015
## 46021 3349017  Honda Civic VTi Prosmatec 1.8 i-VTEC 2015 3250000       2015
## 46022 3748215                          Toyota Aqua G 2016 3000000       2016
## 46023 3806951 Toyota Corolla GLi Automatic 1.3 VVTi 2015 2250000       2015
##                  Location Mileage Registered.City Engine.Type Engine.Capacity
## 46018  Gujranwala Punjab   40000          Lahore           1            1800
## 46019      Lahore Punjab   32000   Un-Registered           1            1500
## 46020  Rawalpindi Punjab   52000   Un-Registered           1            1500
## 46021      Lahore Punjab  125000          Lahore           1            1800
## 46022  Gujranwala Punjab   60000          Lahore           1            1500
## 46023  Gujranwala Punjab   77000      Gujranwala           1            1300
##       Transmission    Color Assembly Body.Type Features Last.Updated
## 46018            1    Black    Local         2             14-Jan-20
## 46019            1    Black Imported         5             28-Jul-19
## 46020            1     Blue Imported         1             18-Dec-19
## 46021            1    Black    Local         2              4-Jun-19
## 46022            1    Black Imported         1             22-Dec-19
## 46023            1 Assembly    Local         2             14-Jan-20
##                                                                                             URL
## 46018    https://www.pakwheels.com/used-cars/honda-civic-2017-for-sale-in-gujranwala-3806954
## 46019        https://www.pakwheels.com/used-cars/honda-vezel-2015-for-sale-in-lahore-3448128
## 46020    https://www.pakwheels.com/used-cars/toyota-aqua-2015-for-sale-in-rawalpindi-3737684
## 46021        https://www.pakwheels.com/used-cars/honda-civic-2015-for-sale-in-lahore-3349017
## 46022    https://www.pakwheels.com/used-cars/toyota-aqua-2016-for-sale-in-gujranwala-3748215
## 46023 https://www.pakwheels.com/used-cars/toyota-corolla-2015-for-sale-in-gujranwala-3806951
```

```r
# Dimensions of our data
dim(df)
```

Our dataset have 46k rows and 16 attributes

```
## [1] 46023    16
```

```
# Data types for each column in our dataset
str(df)
```

Columns with long descriptions and sentences will need to be droped, also the rest of the columns will be converted into factor variable

```
## 'data.frame':    46023 obs. of  16 variables:
##  $ Ad.No          : int  4096758 4168305 4168298 4168307 4168306 4168303 4168304 4168309 4168310 4168
##  $ Name           : chr  "Toyota Vitz F 1.0 2017" "Toyota Corolla GLi Automatic 1.3 VVTi 2019" "Suzul
##  $ Price          : int  2385000 111000 1530000 1650000 1435000 3850000 1440000 1425000 2650000 33500
##  $ Model.Year     : int  2017 2019 2019 2019 2010 2017 2017 2012 1998 2017 ...
##  $ Location       : chr  " G- 8, Islamabad Islamabad" " Peshawar KPK" " Akora Khattak, Nowshera KPK"
##  $ Mileage        : int  9869 11111 17500 9600 120000 22000 31000 101000 110000 60000 ...
##  $ Registered.City: chr  "Un-Registered" "Islamabad" "Un-Registered" "Lahore" ...
##  $ Engine.Type    : int  1 1 1 1 1 1 1 1 2 1 ...
##  $ Engine.Capacity: int  1000 1300 660 660 1300 1500 1000 1000 3000 1800 ...
##  $ Transmission   : int  1 1 1 2 2 1 2 1 1 1 ...
##  $ Color          : chr  "Silver" "White" "White" "White" ...
##  $ Assembly       : chr  "Imported" "Local" "Local" "Local" ...
##  $ Body.Type      : int  1 2 1 1 2 2 1 1 3 2 ...
##  $ Features       : chr  " ABS, AM/FM Radio, Air Bags, Air Conditioning, CD Player, DVD Player, Immol
##  $ Last.Updated   : chr  "11-Jul-20" "12-Jul-20" "12-Jul-20" "12-Jul-20" ...
##  $ URL            : chr  "https://www.pakwheels.com/used-cars/toyota-vitz-2017-for-sale-in-islamabad-
```

```
# Running some stats on the dataset
summary(df)
```

```
##      Ad.No             Name               Price             Model.Year
##  Min.   :  13381   Length:46023       Min.   :  111000   Min.   :1990
##  1st Qu.:4051758   Class :character   1st Qu.:  850000   1st Qu.:2007
##  Median :4103354   Mode  :character   Median : 1450000   Median :2013
##  Mean   :4070389                      Mean   : 2014144   Mean   :2011
##  3rd Qu.:4142396                      3rd Qu.: 2300000   3rd Qu.:2016
##  Max.   :4168339                      Max.   :77500000   Max.   :2019
##    Location            Mileage       Registered.City     Engine.Type
##  Length:46023       Min.   :     1   Length:46023       Min.   :1.000
##  Class :character   1st Qu.: 48892   Class :character   1st Qu.:1.000
##  Mode  :character   Median : 80000   Mode  :character   Median :1.000
##                     Mean   : 90964                      Mean   :1.084
##                     3rd Qu.:120000                      3rd Qu.:1.000
##                     Max.   :999999                      Max.   :3.000
##  Engine.Capacity  Transmission      Color             Assembly
##  Min.   :  16    Min.   :1.000   Length:46023       Length:46023
##  1st Qu.:1000    1st Qu.:1.000   Class :character   Class :character
##  Median :1300    Median :2.000   Mode  :character   Mode  :character
##  Mean   :1313    Mean   :1.535
```

```
##   3rd Qu.:1500    3rd Qu.:2.000
##   Max.   :6600    Max.   :2.000
##     Body.Type       Features         Last.Updated           URL
##   Min.   :1.000   Length:46023      Length:46023      Length:46023
##   1st Qu.:1.000   Class :character   Class :character   Class :character
##   Median :2.000   Mode  :character   Mode  :character   Mode  :character
##   Mean   :1.772
##   3rd Qu.:2.000
##   Max.   :6.000
```

```r
# Checking for null values in our data
sapply(df, function(x) sum(is.na(x)))
```

```
##           Ad.No            Name           Price      Model.Year        Location
##               0               0               0               0               0
##         Mileage Registered.City     Engine.Type Engine.Capacity    Transmission
##               0               0               0               0               0
##           Color        Assembly       Body.Type        Features    Last.Updated
##               0               0               0               0               0
##             URL
##               0
```

Data cleaning

Link to dataset: https://www.kaggle.com/zaynshahbaz/pakistan-car-prices

```r
df <- subset(df, select = -c(Ad.No, Last.Updated, URL, Features, Color))
head(df)
```

Dropping the URL, last updated, ad.no columns, and features because they will be no use in doing regression

```
##                                         Name   Price Model.Year
## 1                     Toyota Vitz F 1.0 2017 2385000       2017
## 2 Toyota Corolla GLi Automatic 1.3 VVTi 2019  111000       2019
## 3                       Suzuki Alto VXL 2019 1530000       2019
## 4                       Suzuki Alto VXR 2019 1650000       2019
## 5               Toyota Corolla XLi VVTi 2010 1435000       2010
## 6                Honda Civic 1.5 RS Turbo 2017 3850000       2017
##                             Location Mileage Registered.City Engine.Type
## 1         G- 8, Islamabad Islamabad    9869   Un-Registered           1
## 2                      Peshawar KPK   11111       Islamabad           1
## 3       Akora Khattak, Nowshera KPK   17500   Un-Registered           1
## 4    Abdullahpur, Faisalabad Punjab    9600          Lahore           1
## 5   9th Avenue, Islamabad Islamabad  120000       Islamabad           1
## 6  Peshawar Road, Rawalpindi Punjab   22000       Islamabad           1
##   Engine.Capacity Transmission Assembly Body.Type
## 1            1000            1 Imported         1
## 2            1300            1    Local         2
```

```
## 3              660           1    Local         1
## 4              660           2    Local         1
## 5             1300           2    Local         2
## 6             1500           1    Local         2
```

```
library(stringr)
df$Name = sub("\\ .*", "", as.character(df$Name))
```

Splitting the name column to only have the value for car brand

```
df <- subset(df, select = -c(Location))
df["Registered"] <- FALSE
df$Registered[df$Registered.City!="Un-Registered"] <- TRUE
df <- subset(df, select = -c(Registered.City))
```

Making registered_city into a True or false column, then dropping registered.city col

```
df["Local"] <- FALSE
df$Local[df$Assembly=="Local"] <- TRUE
df <- subset(df, select = -c(Assembly))
```

Turning imported and local into a true or false column, then dropping the Assembly col

```
df$Registered <- as.factor(df$Registered)
df$Transmission <- as.factor(df$Transmission)
df$Engine.Type <- as.factor(df$Engine.Type)
df$Body.Type <- as.factor(df$Body.Type)
df$Local <- as.factor(df$Local)
df$Name <- as.factor(df$Name)
df[sapply(df, is.integer)] <- lapply(df[sapply(df, is.integer)], as.numeric)
str(df)
```

Making columns into factor and numeric to enhance our model implementation for categorical and integer values

```
## 'data.frame':    46023 obs. of  10 variables:
##  $ Name          : Factor w/ 31 levels "Adam","Audi",..: 29 29 28 28 29 11 28 22 29 11 ...
##  $ Price         : num  2385000 111000 1530000 1650000 1435000 ...
##  $ Model.Year    : num  2017 2019 2019 2019 2010 ...
##  $ Mileage       : num  9869 11111 17500 9600 120000 ...
##  $ Engine.Type   : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 2 1 ...
```

```
##  $ Engine.Capacity: num  1000 1300 660 660 1300 1500 1000 1000 3000 1800 ...
##  $ Transmission   : Factor w/ 2 levels "1","2": 1 1 1 2 2 1 2 1 1 1 ...
##  $ Body.Type      : Factor w/ 6 levels "1","2","3","4",..: 1 2 1 1 2 2 1 1 3 2 ...
##  $ Registered     : Factor w/ 2 levels "FALSE","TRUE": 1 2 1 2 2 2 2 2 2 2 ...
##  $ Local          : Factor w/ 2 levels "FALSE","TRUE": 1 2 2 2 2 2 2 1 1 2 ...
```

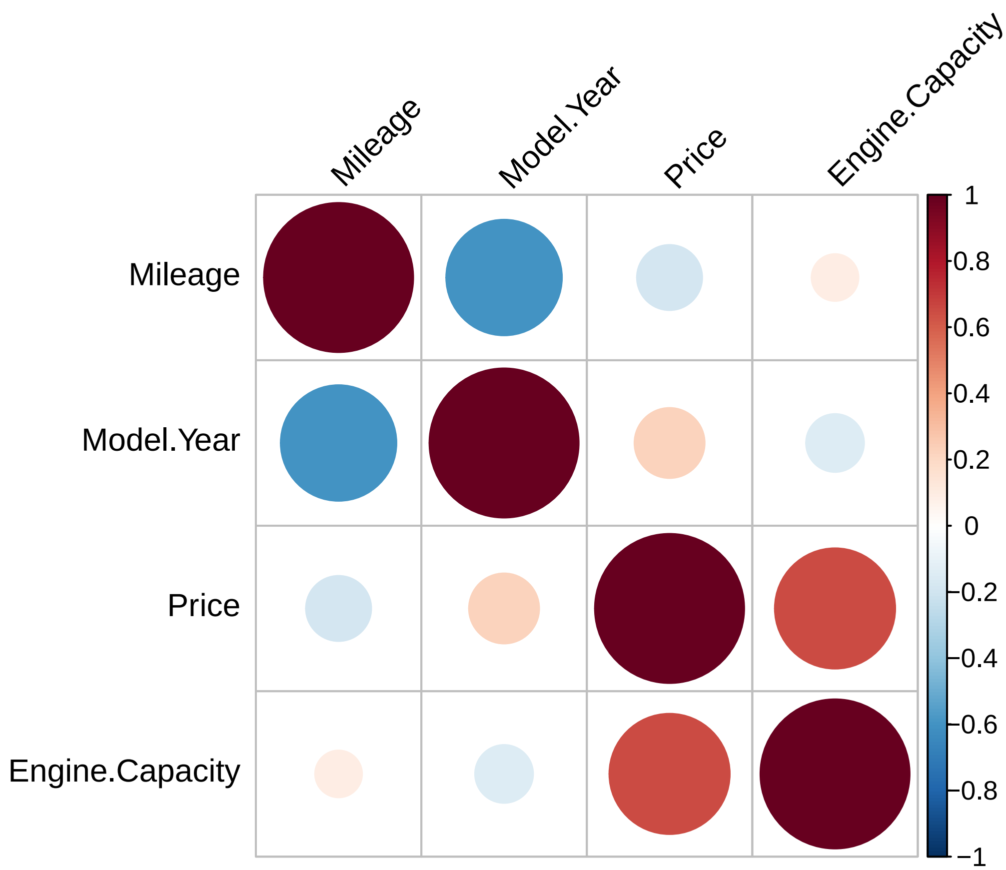
Visual Data exploration

```
library(corrplot)
```

Correlation matrix to identify which numeirc coulumns to use in dataset

```
## corrplot 0.90 loaded
```

```
source("http://www.sthda.com/upload/rquery_cormat.r")
df_numeric <- df[sapply(df, is.numeric)]
rquery.cormat(df_numeric, type="full")
```




```
## $r
##                 Mileage Model.Year Price Engine.Capacity
## Mileage           1.000      -0.60 -0.19           0.098
```

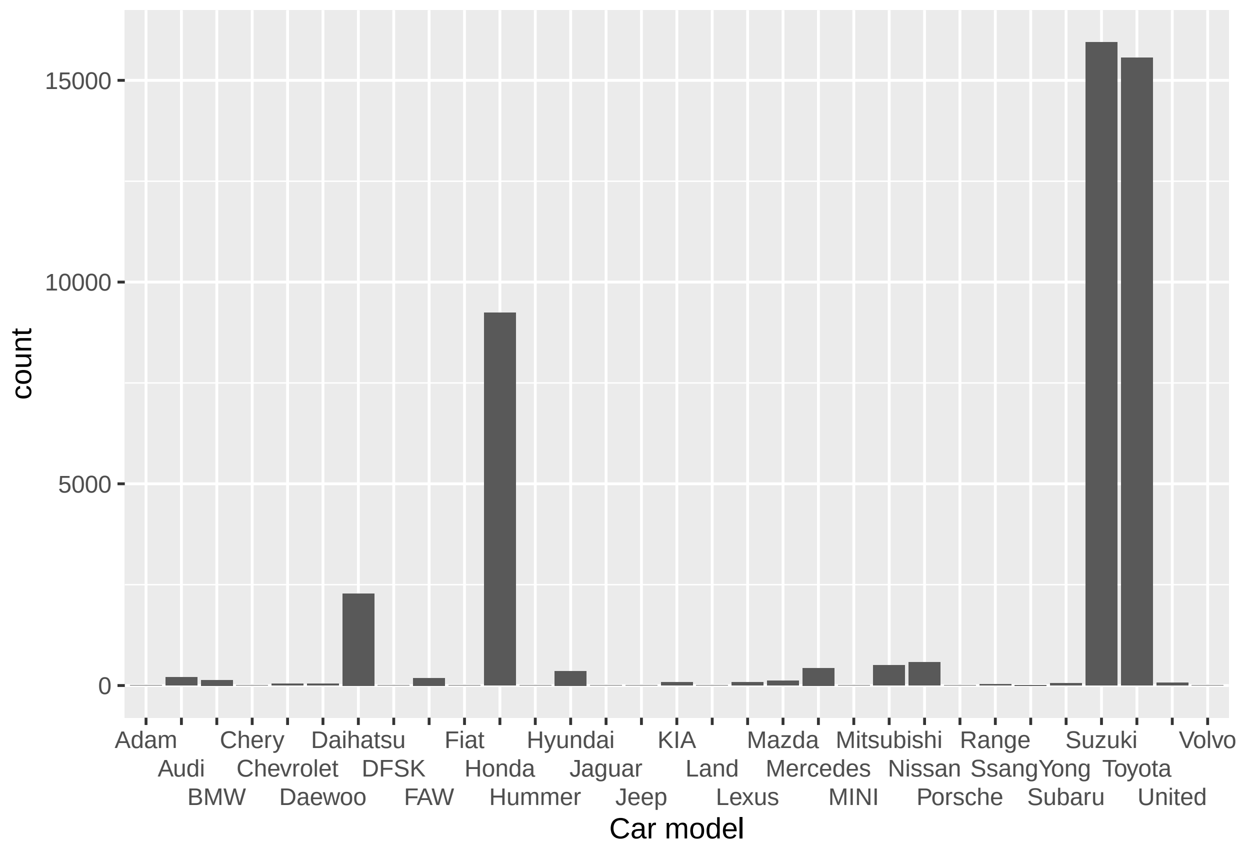
```
## Model.Year       -0.600       1.00  0.22          -0.150
## Price            -0.190       0.22  1.00           0.650
## Engine.Capacity   0.098      -0.15  0.65           1.000
##
## $p
##                 Mileage Model.Year Price Engine.Capacity
## Mileage         0.0e+00    0.0e+00     0         7.3e-99
## Model.Year      0.0e+00    0.0e+00     0        9.9e-234
## Price           0.0e+00    0.0e+00     0         0.0e+00
## Engine.Capacity 7.3e-99   9.9e-234     0         0.0e+00
##
## $sym
##                 Mileage Model.Year Price Engine.Capacity
## Mileage         1
## Model.Year      .       1
## Price                              1
## Engine.Capacity                    ,     1
## attr(,"legend")
## [1] 0 ' ' 0.3 '.' 0.6 ',' 0.8 '+' 0.9 '*' 0.95 'B' 1
```

```
library(ggplot2)
```

Barplot to identify which cars are most popular in Pakistan

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
ggplot(df, aes(x = Name)) + geom_bar() + scale_x_discrete(guide = guide_axis(n.dodge=3)) + xlab("Car mod
```



Model Building

Linear Regression

```r
set.seed(1234)
spec <- c(train=.6, test=.2, validate=.2)
i <- sample(cut(1:nrow(df),nrow(df)*cumsum(c(0,spec)), labels=names(spec)))
train <- df[i=="train",]
test <- df[i=="test",]
vald <- df[i=="validate",]
```

```r
lm <- lm(Price~Registered+Transmission+Engine.Type+Body.Type+Local+Mileage+Engine.Capacity+Model.Year, 
summary(lm)
```

I decided to use all the features in dataset because removing the features that were not correlated with price, removed noise from the data and led to lower scores of models.

```
##
## Call:
## lm(formula = Price ~ Registered + Transmission + Engine.Type +
```

```
##     Body.Type + Local + Mileage + Engine.Capacity + Model.Year,
##     data = train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -13795014   -498905    -48445    410093  43261392
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -2.455e+08  4.643e+06 -52.881  < 2e-16 ***
## RegisteredTRUE  -1.109e+06  4.721e+04 -23.491  < 2e-16 ***
## Transmission2    2.736e+05  3.250e+04   8.418  < 2e-16 ***
## Engine.Type2    -3.963e+06  8.455e+04 -46.864  < 2e-16 ***
## Engine.Type3    -5.505e+05  7.197e+04  -7.649 2.09e-14 ***
## Body.Type2      -5.853e+05  3.163e+04 -18.503  < 2e-16 ***
## Body.Type3       1.863e+06  8.262e+04  22.553  < 2e-16 ***
## Body.Type4       1.995e+04  7.042e+04   0.283    0.777
## Body.Type5       1.617e+05  8.031e+04   2.014    0.044 *
## Body.Type6      -6.306e+05  9.186e+04  -6.865 6.81e-12 ***
## LocalTRUE       -5.313e+05  3.530e+04 -15.051  < 2e-16 ***
## Mileage         -1.968e+00  2.225e-01  -8.846  < 2e-16 ***
## Engine.Capacity  3.310e+03  3.185e+01 103.943  < 2e-16 ***
## Model.Year       1.218e+05  2.300e+03  52.946  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1838000 on 27599 degrees of freedom
## Multiple R-squared:  0.6021, Adjusted R-squared:  0.6019
## F-statistic:  3213 on 13 and 27599 DF,  p-value: < 2.2e-16
```

```r
# Testing on the data
pred <- predict(lm, newdata = test)
# Computing statistical equation to interpretate our model
print(paste('correlation:', cor(pred, test$Price)))
```

```
## [1] "correlation: 0.771521991335604"
```

```r
mse_t <- mean((pred - test$Price)^2)
print(paste("Rmse for test data: ", sqrt(mse_t)))
```

```
## [1] "Rmse for test data:  2051290.16642133"
```

Decision Tree

```r
library(tree)
```

```
## Warning: package 'tree' was built under R version 4.0.5
```

```r
tree1 <- tree(Price~Name+Registered+Transmission+Engine.Type+Body.Type+Local+Mileage+Engine.Capacity+Mo
summary(tree1)
```

```
##
## Regression tree:
## tree(formula = Price ~ Name + Registered + Transmission + Engine.Type +
##     Body.Type + Local + Mileage + Engine.Capacity + Model.Year,
##     data = train)
## Variables actually used in tree construction:
## [1] "Engine.Capacity" "Name"            "Model.Year"
## Number of terminal nodes:  10
## Residual mean deviance:  1.289e+12 = 3.559e+16 / 27600
## Distribution of residuals:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -35100000   -444800   -150800         0    371200  41080000
```

```r
pred_tree <- predict(tree1, newdata=test)
print(paste('correlation:', cor(pred_tree, test$Price)))
```

Decision Tree performed much better than linear regression because our correlation got alot higher and the RMSE got relatively lower.

```
## [1] "correlation: 0.896548849760948"
```

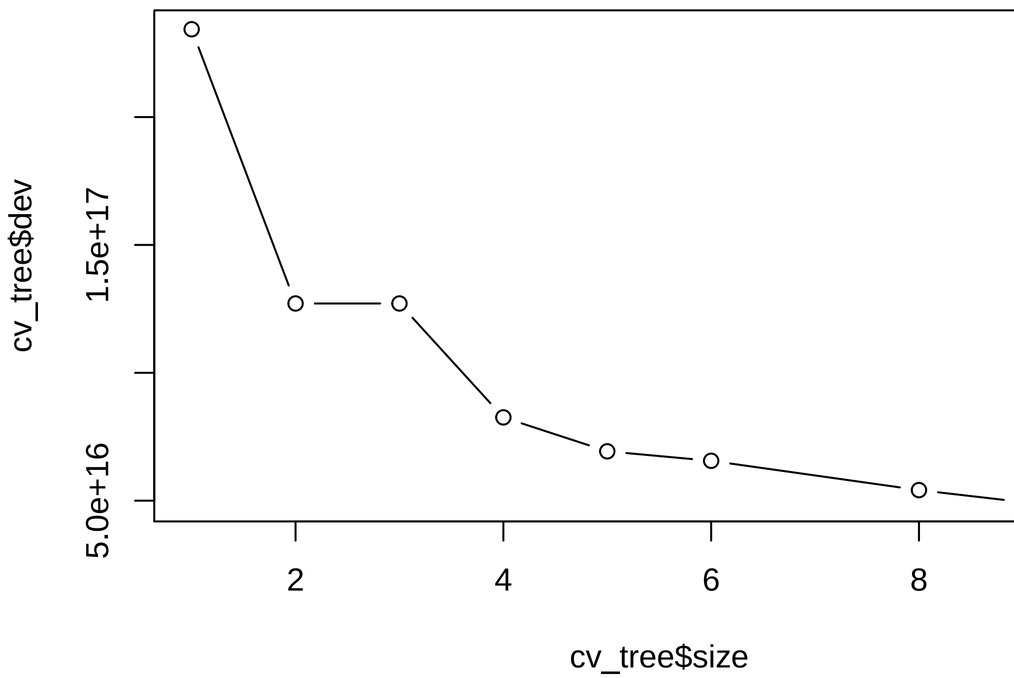
```r
rmse_tree <- sqrt(mean((pred_tree-test$Price)^2))
print(paste('rmse:', rmse_tree))
```

```
## [1] "rmse: 1429424.45115798"
```

Cross validation

```r
cv_tree <- cv.tree(tree1)
plot(cv_tree$size, cv_tree$dev, type='b')
```

We will prune the tree to 5 terminal nodes because we want to avoid overfitting by pruning it to



a node with smallest deviance.

Pruning the tree, and then testing.

```
tree_pruned <- prune.tree(tree1, best=5)
pred_pruned <- predict(tree_pruned, newdata=test)
print(paste('correlation:', cor(pred_pruned, test$Price)))
```

In this case, the pruning did not improve results on test data because we got a higher correlation and a lower RMSE for the unpruned Tree.

```
## [1] "correlation: 0.846419959763747"
```

```
rmse_pruned <- sqrt(mean((pred_pruned-test$Price)^2))
print(paste('rmse pruned:', rmse_pruned))
```

```
## [1] "rmse pruned: 1714840.04713052"
```

Support Vector machines

```r
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.0.5
```

```r
svm1 <- svm(Price~Registered+Transmission+Engine.Type+Body.Type+Local+Mileage+Engine.Capacity+Model.Yea
summary(svm1)
```

```
##
## Call:
## svm(formula = Price ~ Registered + Transmission + Engine.Type + Body.Type +
##     Local + Mileage + Engine.Capacity + Model.Year, data = train,
##     kernel = "linear", cost = 10, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  linear
##        cost:  10
##       gamma:  0.07142857
##     epsilon:  0.1
##
##
## Number of Support Vectors:  10590
```

```r
pred <- predict(svm1, newdata=test)
```

```r
cor_svm1 <- cor(pred, test$Price)
print(paste('correlation:', cor(pred, test$Price)))
```

SVM got a lower correlation than decision tree and linear regression. The RMSE for SVM was also higher from decision tree and linear regression. I decided not to do hyper parameter tuning for SVM because it took alot of time and was unable to find the optimal paramters, as it gave a warning message "WARNING: reaching max number of iterations".

```
## [1] "correlation: 0.740897161311206"
```

```r
rmse_svm1 <- sqrt(mean((pred - test$Price)^2))
print(paste('rmse:', rmse_svm1))
```

```
## [1] "rmse: 2420531.09540852"
```

Results Analysis

Correlation for these algortihms:

Decision Tree: 0.90

Pruned Decision Tree: 0.85

Linear regression: 0.77

Support vector machine: 0.74

RMSE for these algortihms:

Decision Tree: 1,429,424 (PKR Rupees) or $6,000

Pruned Decision Tree: 1,714,840 (PKR Rupees) or $9,600

Linear regression: 2,051,290 (PKR Rupees) or $12,000

Support vector machine: 2,420,531 (PKR Rupees) or $12,400

Summary:

Decision tree performed much more efficiently than SVM and linear regression. Our decision tree was off by only about 1.4 million (Rupees) or $6,000 compared to the other algorithms which were off by more than 2 million (Rupees) or $12,000. Linear regrssion works much better when our data is linear, however decision tree work better with more qualatitive and factor values and with more complex data. Pruning the decision tree also did not help improve perfromance of our data. In our case, our data was much more complex because our variables were not much correlated with the price, which is why decision tree performed better. On the other hand, Support vector machine took alot of time to compile and failed to give efficent results since it is performing dot product of training examples and our data had already been scaled. Furthermore, this script can defientely be useful in the new data because it was able to learn different variations in prices of car based on car's attributes, as we got a correlation of 0.90 and was off by $6,000.