Prediction of Car Price from Cardekho

Abhisek Chatterjee

14.10.2020

Contents

1 Introduction

While buying a first-hand car is easily possible for some people,many people do feel that they don't have the budget to go for a brand new car.However,there are also people who want to buy a new car by selling their old car and some people who just want to sell their old car in order to gather some money for important other purposes.In this situation,the appropriate determination of pricing any old car becomes absolutely necessary.Whether it be for a person who wants to buy an old car or a car-owner who wants to sell her/his car,the appropriate pricing according to depreciation based on the car age and other reasons,is a necessary task.Pricing also helps a company through which these types of transactions and trade takes place.A company like Cardekho needs appropriate insights from thier available information,which they can use in future.So there is a business problem that needs to be addressed.

2 Details about the Data

We can get the data provided by www.cardekho.com in Kaggle website.This data contains information about used cars listed on the cardekho website.Now,there are two datasets in the kaggle website regarding the car details.And here comes an underlying challenge that we may counter.The names of the two datasets are "car data.csv" and "CAR DETAILS FROM CAR DEKHO.csv".
There are 301 observations with 9 variables in the "car data.csv" dataset and 4340 observations with 8 variables in the "CAR DETAILS FROM CAR DEKHO.csv" dataset.Now,the 8 variables that are available in the larger dataset are also 8 of the 9 variables in the smaller dataset with some name and scale differences and the smaller dataset also contains an additional variable that isn't available in the larger dataset. The 8 common variables in the two given datasets are as follows:

Car Name : Name of the car.This is a character variable.
Year : Year in which the car was bought.This is an integer variable.
Selling Price : Price at which the car is being sold.This is a numeric variable.
Kms Driven : Number of Kilometres the car has been driven.This is a numeric variable.
Fuel Type : Fuel type of car (Petrol / Diesel / CNG / LPG / Electric).This is a categorical variable.
Seller Type : Tells if a Seller is Individual or a Dealer.This is a categorical variable.

Transmission : Gear transmission of the car (Automatic/Manual).This is a categorical variable.
Owner : Defines the number of owners the car has previously had.

Now,the variable “Selling Price” is in the scale of actual rupees in the larger dataset and it is in scale “lakhs of rupess” in the smaller dataset.Again,the categorical variable “Owner” has 3 distinct values excluding missing values in the smaller dataset,the variable has 5 distinct variables in the larger dataset. Important fact here is,the smaller dataset contains the variable “Present Price” that indicates the current ex-showroom price of the car in lakhs of rupess.This variable is missing in the larger dataset.

3 Problem

We need to address the business problem very exploratively.Let us divide our problem in some parts.
We have two datasets.Now the smaller one,that was uploaded to kaggle,more than 1 years ago has 301 observations with 9 variables.The second dataset was uploaded a few months ago,has over 4000 observations and 8 variables.The initial problem that we are here presented is that we have to predict selling price for any car,about which information is available on the other variables.Now,our primary problem is that only.But we will also look into some other things.
Firstly,we have two datasets.As we have more ways of information available for smaller number of observations in the smaller dataset,We will first try to build a model based on that dataset only.We will check how good our model performs from evaluation techniques and some validation.
Secondly,as we have more than 4000 observations available in the larger dataset,we will build another model combining the two datasets.We have no prior information whether the second dataset and the first dataset have some observations in common or not.As,we have one less variable for the larger dataset,we will se if we can come to any conclusion about the effect of that variable on our model.We will also in general evaluate how our model based on the merged dataset performs by using validation and some other techniques.

So,here comes our collection of goals for this purpose.
1)We are to build models based on the smaller dataset and evaluate the performance of the models.
2)We are to merge the two datasets suitably,discarding any duplicate observations if present,build models based on the merged dataset and evaluate the performance of the models.
3)We are to observe how the absence of the "Present Price" variable effects our models based on the merged dataset.

4 Code

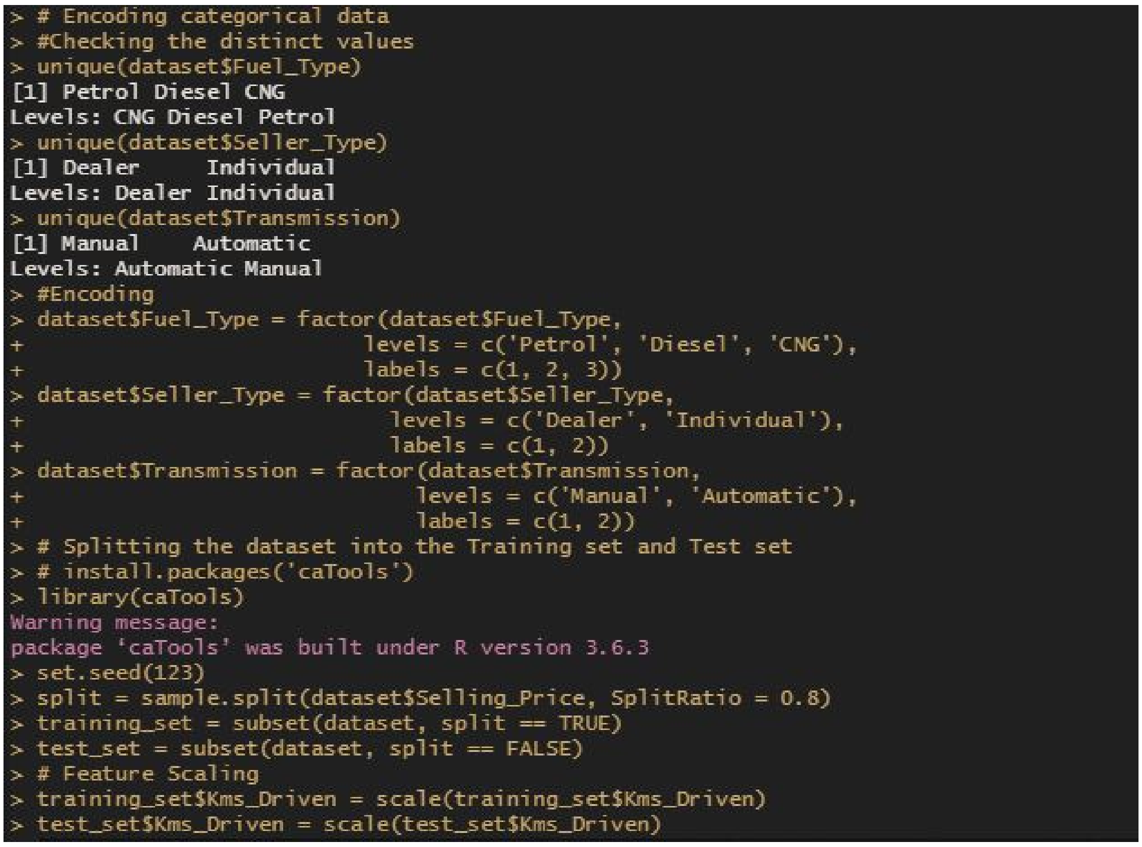
All the code related to this short project can be found in 4 R script files.The name of the files are
1: 'Cardekho1 LinReg.R'
2: 'Cardekho1 RandFor.R',
3: 'Cardekho2 LinReg.R'
4: 'Cardekho2 RandFor.R'

5 The Smaller Dataset

5.1 Exploratory Analysis

We first import the smaller dataset in the first script file using the 'read.csv' command.We observe that there are no missing values in the smaller dataset,so we don't have to deal with missing values here.Now,the variable 'Year' will not directly contribute to our model.So we create a new variable 'Age' by substracting the 'Year' column from the current year and thus get the age of the cars.As the name of the cars and the year in which they were brought will not contribute to our model,we drop those variables and get a dataset with 8 variables.

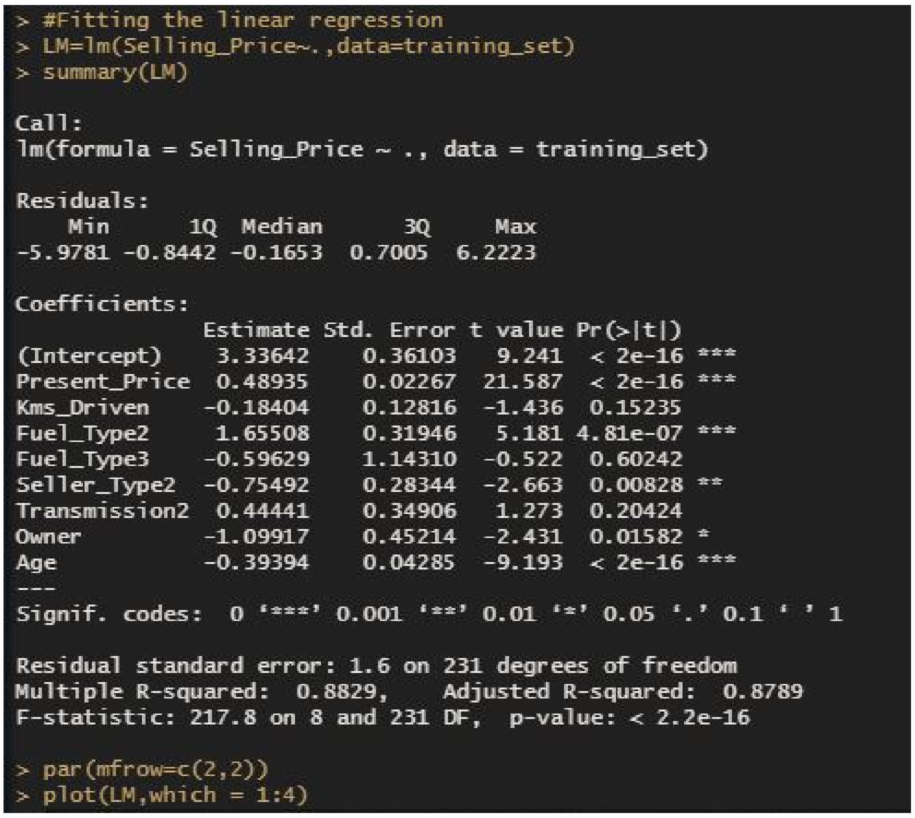
Now,we have some categorical variables in our dataset and in order to use them properly,we need to encode them.We first check the unique values of the variables Fuel Type,Seller Type and Transmission and then properly factorize them with apprpriate labels.We also split the dataset into a training and a test set,where we allocate 80 percent of the total observations randomly to the training set and the rest to the test set.We also observe that the Kms Driven variable is in a different scale and can cause problems in model building if not dealt properly.So we standardize the Kms Driven variable to bring it to an appropriate scale.The output of this section is given in the picture below.



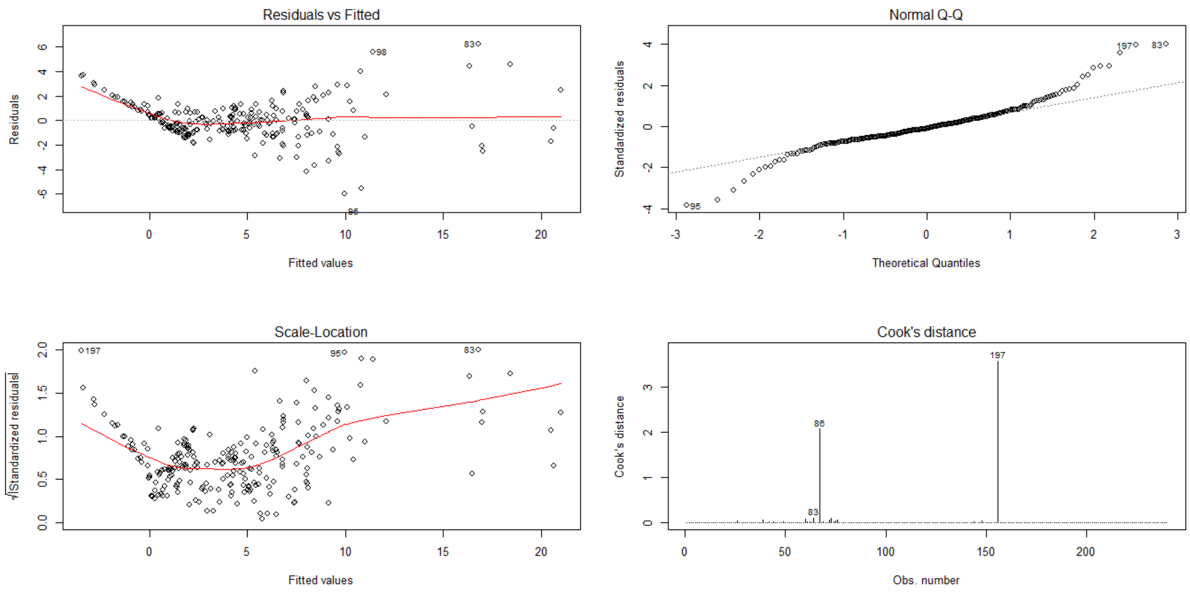
```
> # Encoding categorical data
> #Checking the distinct values
> unique(dataset$Fuel_Type)
[1] Petrol Diesel CNG
Levels: CNG Diesel Petrol
> unique(dataset$Seller_Type)
[1] Dealer     Individual
Levels: Dealer Individual
> unique(dataset$Transmission)
[1] Manual    Automatic
Levels: Automatic Manual
> #Encoding
> dataset$Fuel_Type = factor(dataset$Fuel_Type,
+                            levels = c('Petrol', 'Diesel', 'CNG'),
+                            labels = c(1, 2, 3))
> dataset$Seller_Type = factor(dataset$Seller_Type,
+                              levels = c('Dealer', 'Individual'),
+                              labels = c(1, 2))
> dataset$Transmission = factor(dataset$Transmission,
+                               levels = c('Manual', 'Automatic'),
+                               labels = c(1, 2))
> # Splitting the dataset into the Training set and Test set
> # install.packages('caTools')
> library(caTools)
Warning message:
package ‘caTools’ was built under R version 3.6.3
> set.seed(123)
> split = sample.split(dataset$Selling_Price, SplitRatio = 0.8)
> training_set = subset(dataset, split == TRUE)
> test_set = subset(dataset, split == FALSE)
> # Feature Scaling
> training_set$Kms_Driven = scale(training_set$Kms_Driven)
> test_set$Kms_Driven = scale(test_set$Kms_Driven)
```

5.2 Model and Predictions

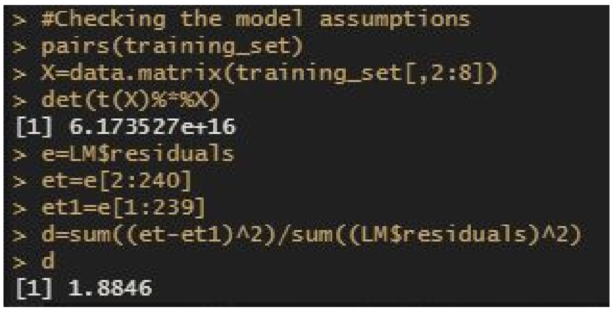
We first fit the linear model with Selling Price as our response variable and the other variables as our set of predictor variables.Then we see the summary of our model.We also plot our model for evaluation.We see that our model explains 88 percent of the total variation.We also see that the p-values for the variable Kms Driven,the 2nd dummy variables for Fuel Type and Transmission are less than 0.05.From the plots,we see that the Homoscedasticity assumption(i.e. the error variances are equal for all the observations) is not valid because the ‘Residuals vs Fit’ plot has some pattern,we have also two outliers because the corresponding Cook’s Distances are large.So we have to discard these outliers and deal with heteroscedasticity.
We also see some other assumptions.The code is shown in script 1.



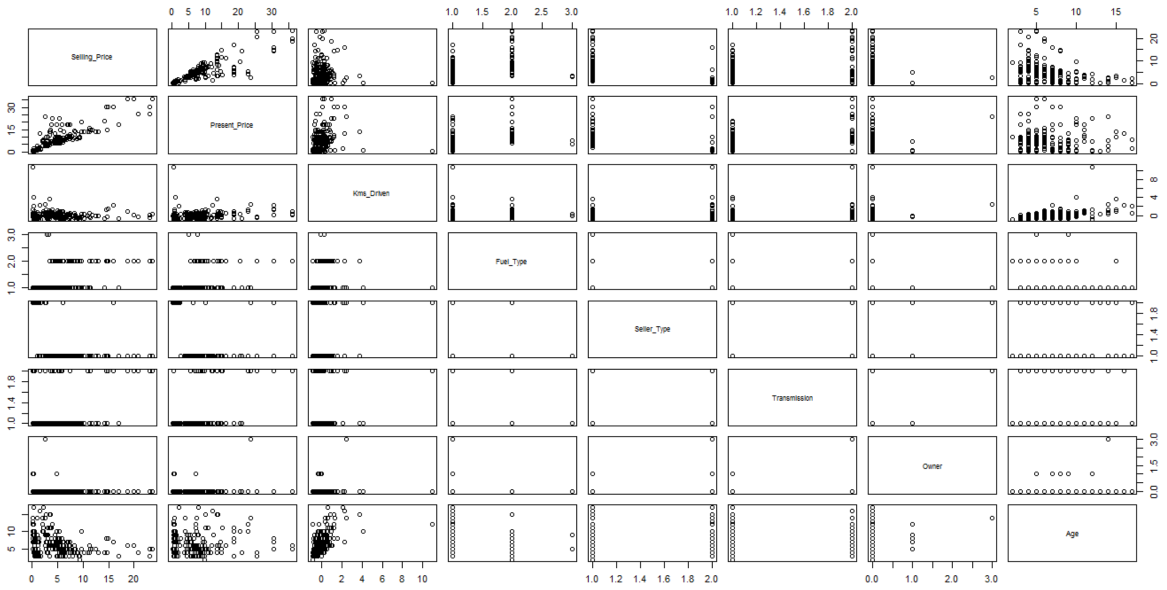
```
> #Fitting the linear regression
> LM=lm(Selling_Price~.,data=training_set)
> summary(LM)

Call:
lm(formula = Selling_Price ~ ., data = training_set)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.9781 -0.8442 -0.1653  0.7005  6.2223 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)    3.33642    0.36103   9.241  < 2e-16 ***
Present_Price  0.48935    0.02267  21.587  < 2e-16 ***
Kms_Driven    -0.18404    0.12816  -1.436  0.15235    
Fuel_Type2     1.65508    0.31946   5.181 4.81e-07 ***
Fuel_Type3    -0.59629    1.14310  -0.522  0.60242    
Seller_Type2  -0.75492    0.28344  -2.663  0.00828 ** 
Transmission2  0.44441    0.34906   1.273  0.20424    
Owner         -1.09917    0.45214  -2.431  0.01582 *  
Age           -0.39394    0.04285  -9.193  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.6 on 231 degrees of freedom
Multiple R-squared:  0.8829,	Adjusted R-squared:  0.8789 
F-statistic: 217.8 on 8 and 231 DF,  p-value: < 2.2e-16

> par(mfrow=c(2,2))
> plot(LM,which = 1:4)
```



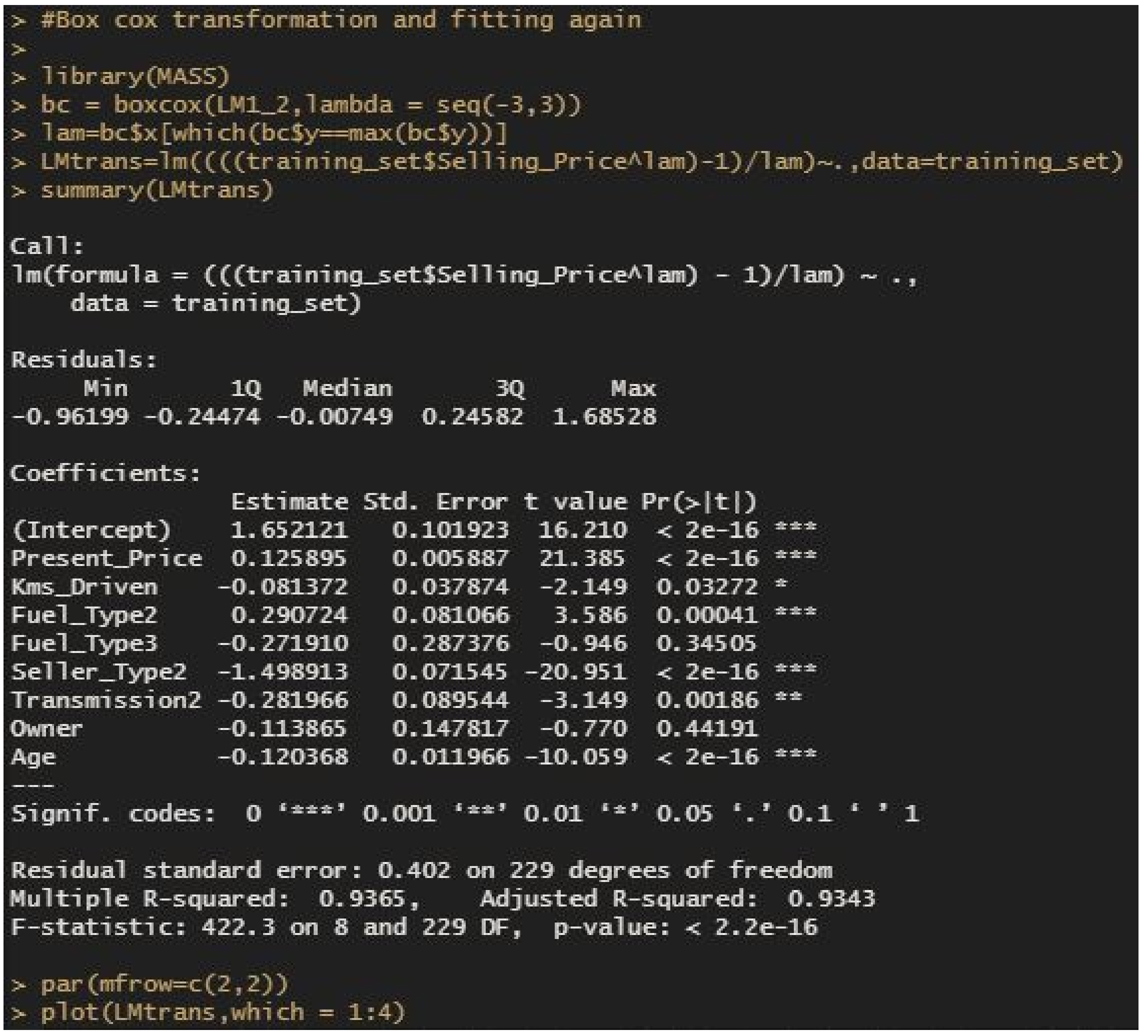
We see the scatter plot of different variables,calculate the Durbin Watson Statistic,determinant of the X^TX matrix,where X is our feature matrix..The output and the plots of this part are given below.



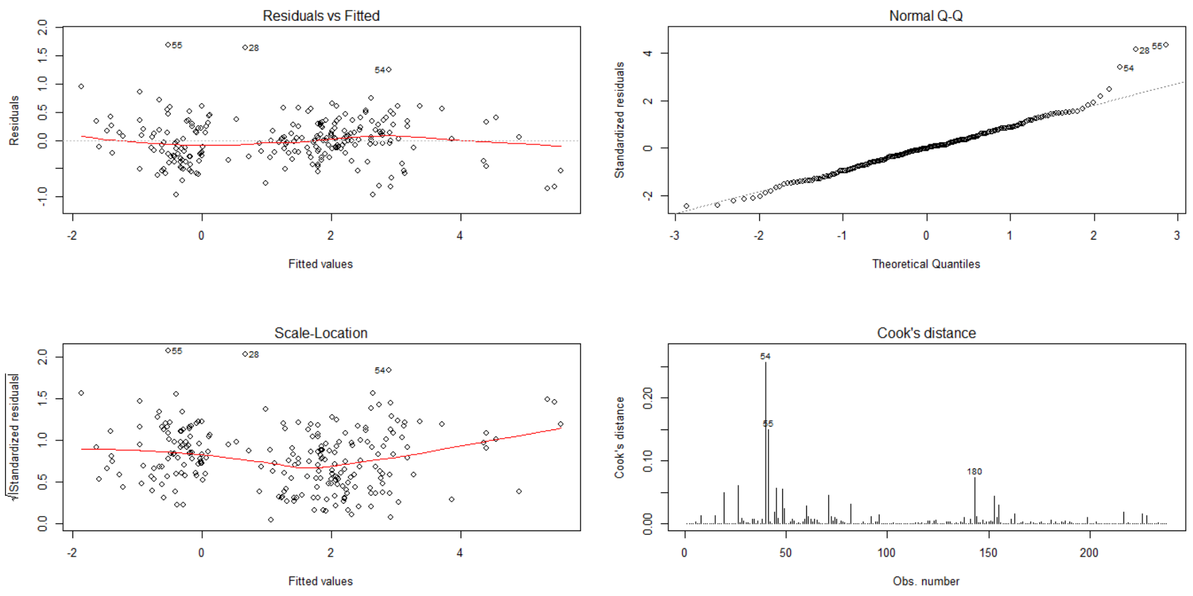
```
> #Checking the model assumptions
> pairs(training_set)
> X=data.matrix(training_set[,2:8])
> det(t(X)%*%X)
[1] 6.173527e+16
> e=LM$residuals
> et=e[2:240]
> et1=e[1:239]
> d=sum((et-et1)^2)/sum((LM$residuals)^2)
> d
[1] 1.8846
```



From the results,we see that the value of the Durbin Watson statistic is almost 2.So,autocorrelation is not present which is reasonable.Again,the value of the X^TX matrix is very large.Also,looking at the correlation plot,we also say that multicollinearity is also not present.
We check for outliers in the entire dataset and find some outliers in the test set too.So we discard them and again split the data.After this,for obtaining more linearity and to get rid of heteroscedasticity,we perform a box-cox transformation on our response variable by the help of the 'MASS' library and we get the value of the lambda that provides us the lowest residual sum of squares.We then fit our multiple linear regression model with the transformed response variable with the optimum lambda value and summarize the model.The output of the part is given below.

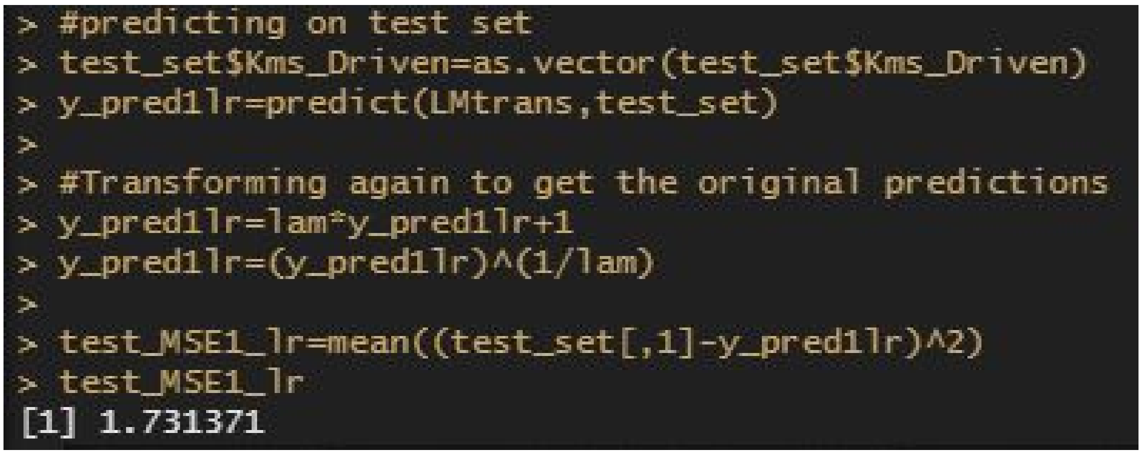


```
> #Box cox transformation and fitting again
>
> library(MASS)
> bc = boxcox(LM1_2,lambda = seq(-3,3))
> lam=bc$x[which(bc$y==max(bc$y))]
> LMtrans=lm((((training_set$Selling_Price^lam)-1)/lam)~.,data=training_set)
> summary(LMtrans)

Call:
lm(formula = (((training_set$Selling_Price^lam) - 1)/lam) ~ .,
    data = training_set)

Residuals:
     Min       1Q   Median       3Q      Max
-0.96199 -0.24474 -0.00749  0.24582  1.68528

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.652121   0.101923  16.210  < 2e-16 ***
Present_Price  0.125895   0.005887  21.385  < 2e-16 ***
Kms_Driven    -0.081372   0.037874  -2.149  0.03272 *
Fuel_Type2     0.290724   0.081066   3.586  0.00041 ***
Fuel_Type3    -0.271910   0.287376  -0.946  0.34505
Seller_Type2  -1.498913   0.071545 -20.951  < 2e-16 ***
Transmission2 -0.281966   0.089544  -3.149  0.00186 **
Owner         -0.113865   0.147817  -0.770  0.44191
Age           -0.120368   0.011966 -10.059  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.402 on 229 degrees of freedom
Multiple R-squared:  0.9365,    Adjusted R-squared:  0.9343
F-statistic: 422.3 on 8 and 229 DF,  p-value: < 2.2e-16

> par(mfrow=c(2,2))
> plot(LMtrans,which = 1:4)
```



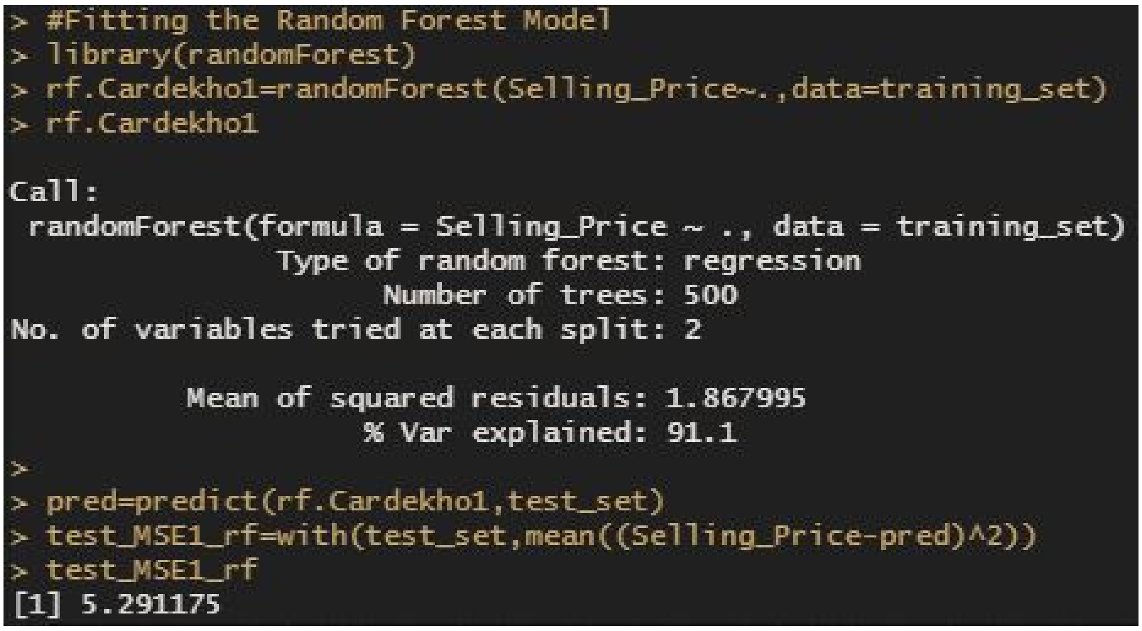
From the results,we see that heteroscedasticity is reduced to a great extent.We can now explain almost 94 percent of the variation in the dataset by the model.However the regression coefficients for the 3rd dummy variable of the Fuel Type and the Owner variable are insignificant.We now predict the selling price for the test set and calculate the mean squared error.The mean squred error is almost 1.7.

We observe that our predictions are fairly accurate.



```
> #predicting on test set
> test_set$Kms_Driven=as.vector(test_set$Kms_Driven)
> y_pred1lr=predict(LMtrans,test_set)
>
> #Transforming again to get the original predictions
> y_pred1lr=lam*y_pred1lr+1
> y_pred1lr=(y_pred1lr)^(1/lam)
>
> test_MSE1_lr=mean((test_set[,1]-y_pred1lr)^2)
> test_MSE1_lr
[1] 1.731371
```

From the similar data preprocessing and removing outliers,we also fit a Random Forest Model to our dataset with 500 trees.The code is shown in script 2.The results are given below.



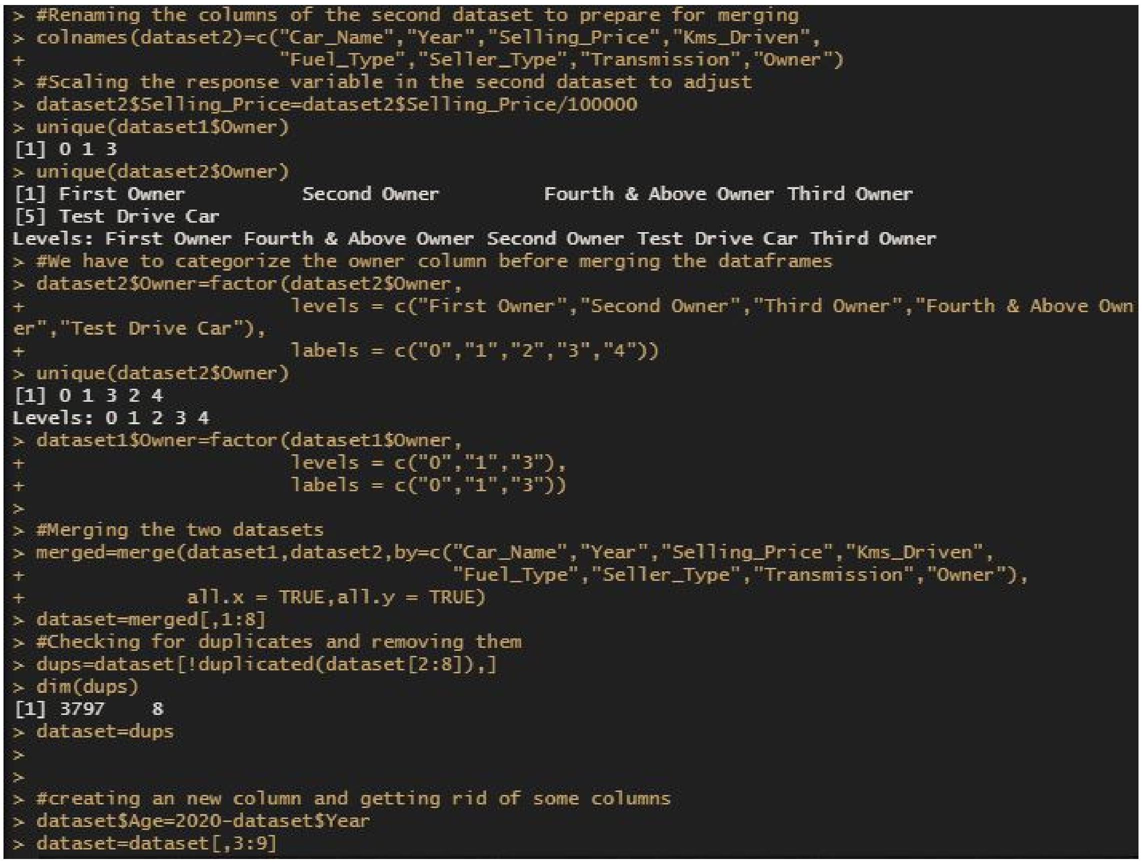
```
> #Fitting the Random Forest Model
> library(randomForest)
> rf.Cardekho1=randomForest(Selling_Price~.,data=training_set)
> rf.Cardekho1

Call:
 randomForest(formula = Selling_Price ~ ., data = training_set)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 2

          Mean of squared residuals: 1.867995
                    % Var explained: 91.1
>
> pred=predict(rf.Cardekho1,test_set)
> test_MSE1_rf=with(test_set,mean((Selling_Price-pred)^2))
> test_MSE1_rf
[1] 5.291175
```

We observe that the variance explained by our Random Forest Model is 91 percent.We also see that the mean squared error is almost 5.So our Multiple Linear Regression model is better than the Random Forest model and we can use it to predict car selling prices with great accuracy given the other information is available.

6 The Combined Dataset

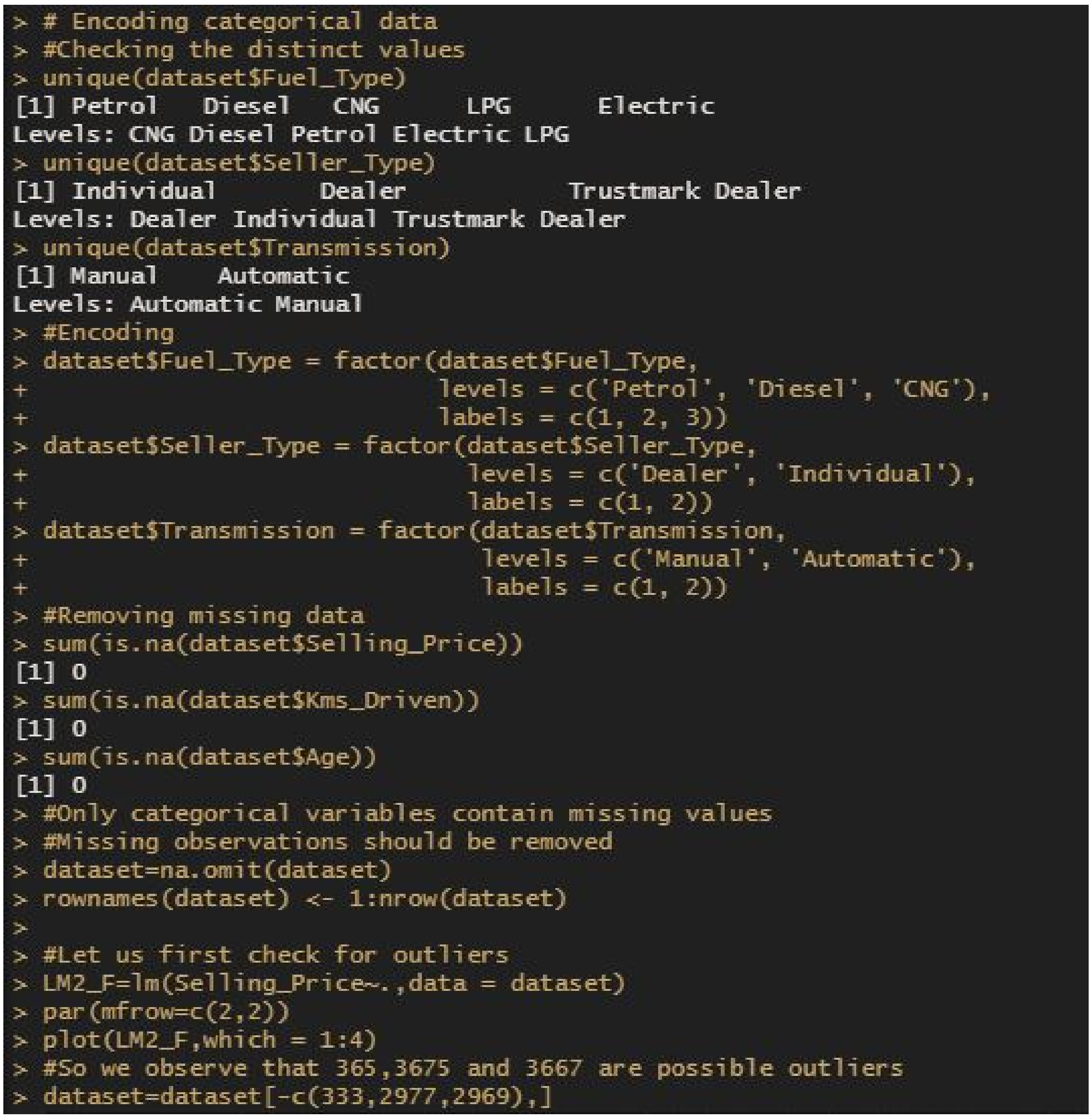
6.1 Exploratory Analysis

Our next aim is to combine the larger dataset with the smaller dataset and look if we can fit a good model with that.We import the two datasets and observe that for the same columns in the two datasets,the names are quite different.So we rename the columns of the larger dataset before merging.Now,as we have stated previously,the values of the categorical variable Owner are in different forms in the two datasets.So,we first check the unique values of that variable in both the datasets and then factorize them appropriately,so that they act similar in the merged dataset.We then merge the datasets by the similar columns in both the datasets.But we don't have the Present Price variable in the larger dataset,so we drop that column after merging.As we didn't know whether the two datasets overlapped,we check for duplicates in the larger dataset and remove the duplicates,we also create the Age variable as we've done before and get rid of the Car Name and the Year column,as they won't contribute to our model.The output of this part is given below.

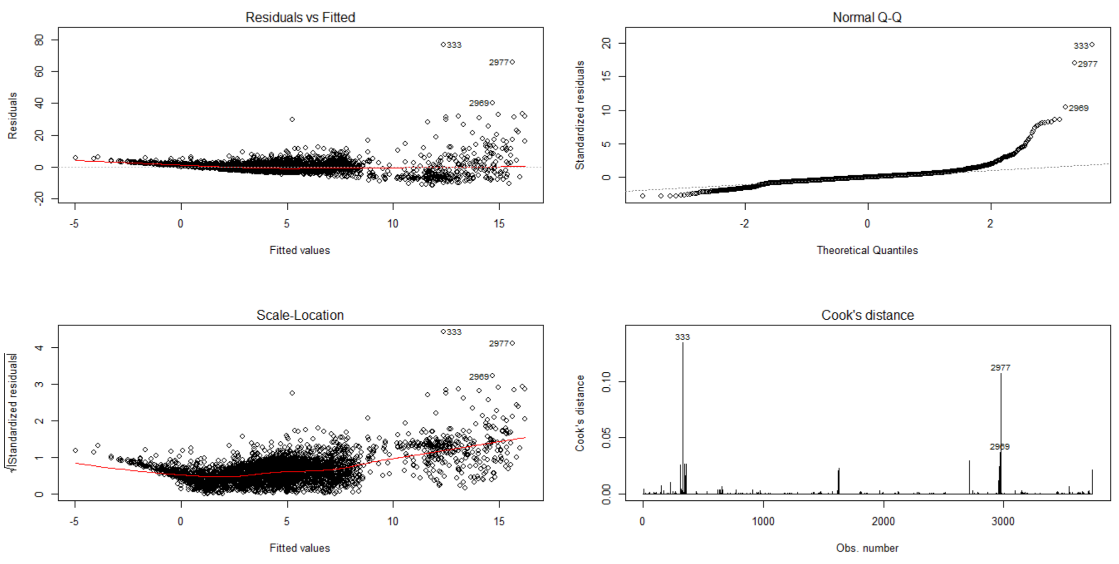


```
> #Renaming the columns of the second dataset to prepare for merging
> colnames(dataset2)=c("Car_Name","Year","Selling_Price","Kms_Driven",
+                      "Fuel_Type","Seller_Type","Transmission","Owner")
> #Scaling the response variable in the second dataset to adjust
> dataset2$Selling_Price=dataset2$Selling_Price/100000
> unique(dataset1$Owner)
[1] 0 1 3
> unique(dataset2$Owner)
[1] First Owner          Second Owner         Fourth & Above Owner Third Owner
[5] Test Drive Car
Levels: First Owner Fourth & Above Owner Second Owner Test Drive Car Third Owner
> #We have to categorize the owner column before merging the dataframes
> dataset2$Owner=factor(dataset2$Owner,
+                       levels = c("First Owner","Second Owner","Third Owner","Fourth & Above Own
er","Test Drive Car"),
+                       labels = c("0","1","2","3","4"))
> unique(dataset2$Owner)
[1] 0 1 3 2 4
Levels: 0 1 2 3 4
> dataset1$Owner=factor(dataset1$Owner,
+                       levels = c("0","1","3"),
+                       labels = c("0","1","3"))
>
> #Merging the two datasets
> merged=merge(dataset1,dataset2,by=c("Car_Name","Year","Selling_Price","Kms_Driven",
+                                     "Fuel_Type","Seller_Type","Transmission","Owner"),
+              all.x = TRUE,all.y = TRUE)
> dataset=merged[,1:8]
> #Checking for duplicates and removing them
> dups=dataset[!duplicated(dataset[2:8]),]
> dim(dups)
[1] 3797    8
> dataset=dups
>
>
> #creating an new column and getting rid of some columns
> dataset$Age=2020-dataset$Year
> dataset=dataset[,3:9]
```

We now encode the three categorical variables ‘Fuel Type’,‘Seller Type’ and ‘Transmission’ in our dataset to use them properly in our model.We also check for any missing values that are available in our dataset and find out that only the categorical variables have missing values,so we decide to remove the missing observations.For checking the possible outliers that are present in the entire dataset,we first fit a linear regression model in the whole dataset using Selling Price as the response variable and the other variables as the set of predictor variables and plot them.We find 3 possible outliers by using the Cook’s Distance and remove the observations from our dataset.The Output and the plots are given below.

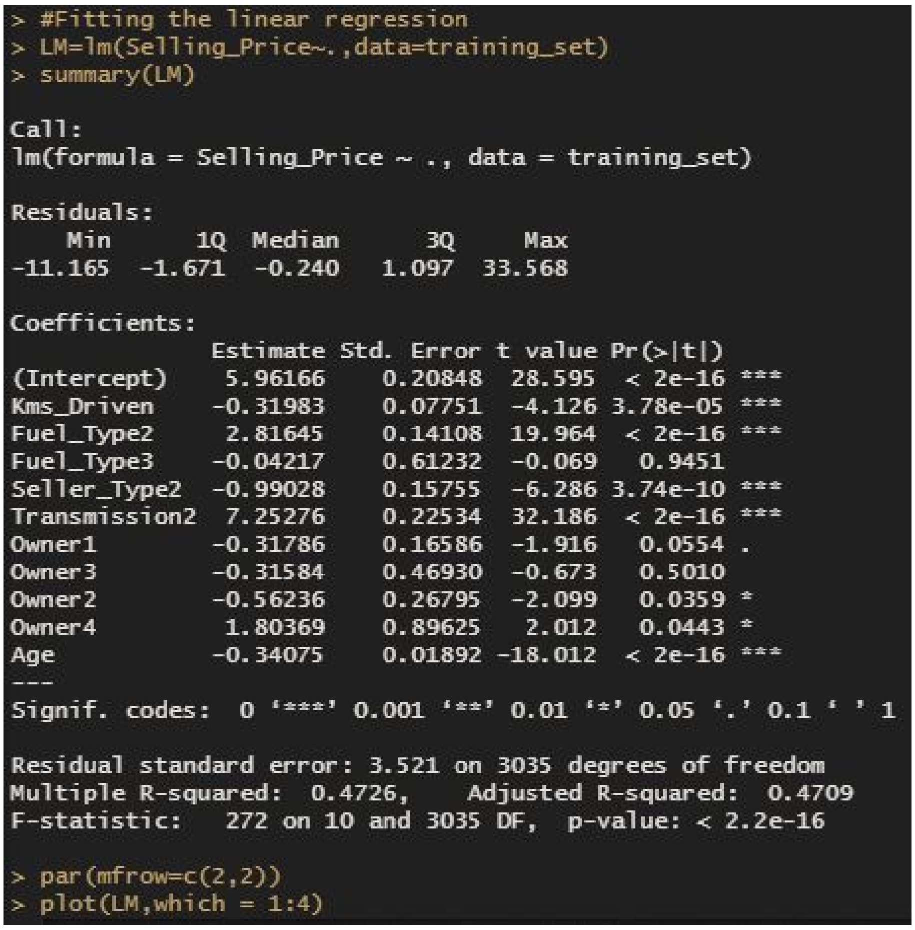


```
> # Encoding categorical data
> #Checking the distinct values
> unique(dataset$Fuel_Type)
[1] Petrol   Diesel   CNG      LPG      Electric
Levels: CNG Diesel Petrol Electric LPG
> unique(dataset$Seller_Type)
[1] Individual       Dealer           Trustmark Dealer
Levels: Dealer Individual Trustmark Dealer
> unique(dataset$Transmission)
[1] Manual    Automatic
Levels: Automatic Manual
> #Encoding
> dataset$Fuel_Type = factor(dataset$Fuel_Type,
+                            levels = c('Petrol', 'Diesel', 'CNG'),
+                            labels = c(1, 2, 3))
> dataset$Seller_Type = factor(dataset$Seller_Type,
+                              levels = c('Dealer', 'Individual'),
+                              labels = c(1, 2))
> dataset$Transmission = factor(dataset$Transmission,
+                               levels = c('Manual', 'Automatic'),
+                               labels = c(1, 2))
> #Removing missing data
> sum(is.na(dataset$Selling_Price))
[1] 0
> sum(is.na(dataset$Kms_Driven))
[1] 0
> sum(is.na(dataset$Age))
[1] 0
> #Only categorical variables contain missing values
> #Missing observations should be removed
> dataset=na.omit(dataset)
> rownames(dataset) <- 1:nrow(dataset)
>
> #Let us first check for outliers
> LM2_F=lm(Selling_Price~.,data = dataset)
> par(mfrow=c(2,2))
> plot(LM2_F,which = 1:4)
> #So we observe that 365,3675 and 3667 are possible outliers
> dataset=dataset[-c(333,2977,2969),]
```

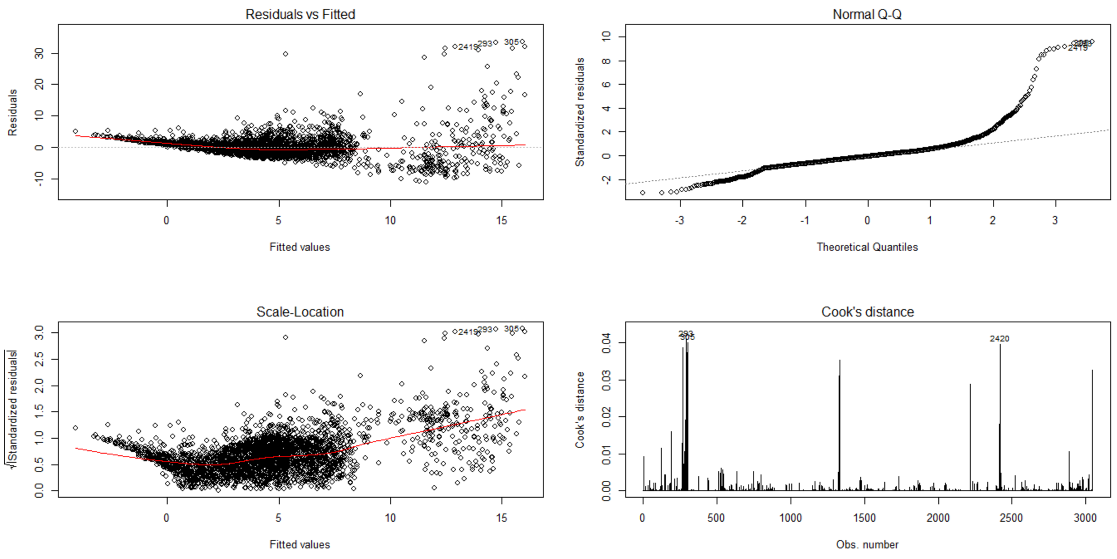


6.2 Model and Predictions

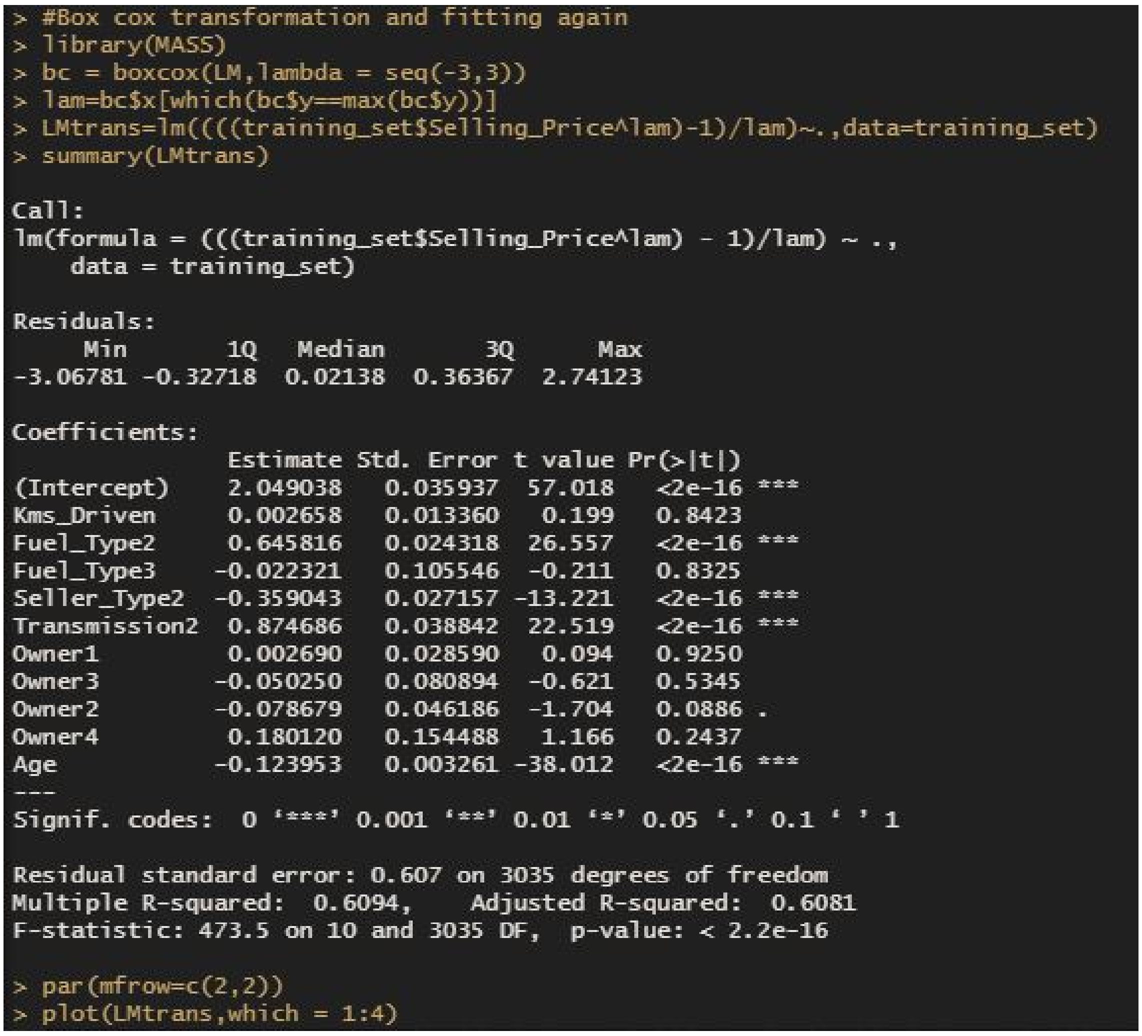
We now split our dataset into a training set and a test set and then standardize the variable ‘Kms Driven’,because the variable is in a different scale.Then we fit a multiple linear regression model using the training set and plot the model to observe some assumptions.The Output and the plots are below.



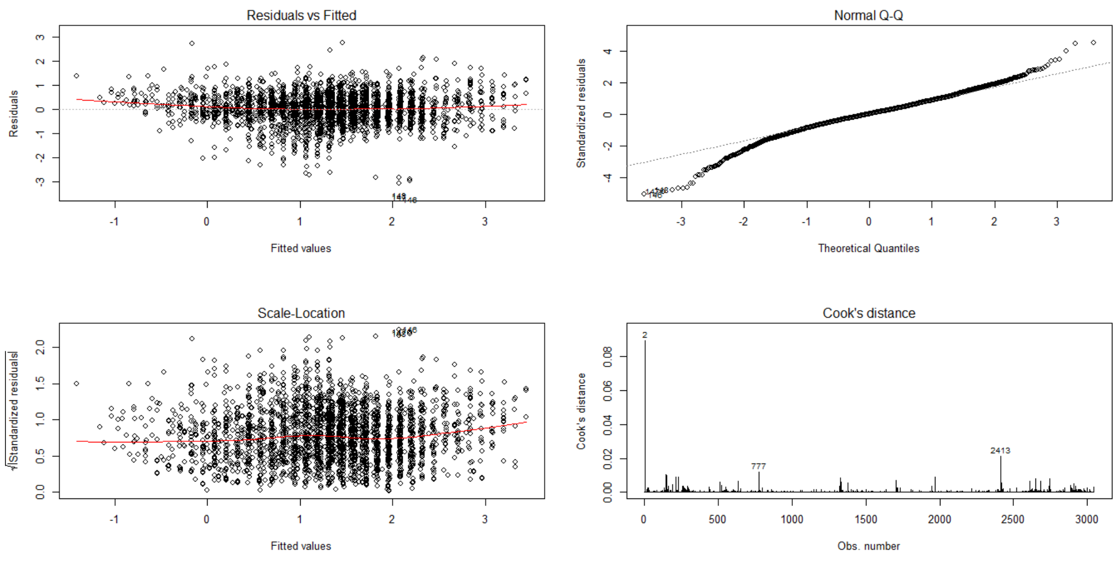
```
> #Fitting the linear regression
> LM=lm(Selling_Price~.,data=training_set)
> summary(LM)

Call:
lm(formula = Selling_Price ~ ., data = training_set)

Residuals:
    Min      1Q  Median      3Q     Max 
-11.165  -1.671  -0.240   1.097  33.568 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)    5.96166    0.20848  28.595  < 2e-16 ***
Kms_Driven    -0.31983    0.07751  -4.126 3.78e-05 ***
Fuel_Type2     2.81645    0.14108  19.964  < 2e-16 ***
Fuel_Type3    -0.04217    0.61232  -0.069   0.9451    
Seller_Type2  -0.99028    0.15755  -6.286 3.74e-10 ***
Transmission2  7.25276    0.22534  32.186  < 2e-16 ***
Owner1        -0.31786    0.16586  -1.916   0.0554 .  
Owner3        -0.31584    0.46930  -0.673   0.5010    
Owner2        -0.56236    0.26795  -2.099   0.0359 *  
Owner4         1.80369    0.89625   2.012   0.0443 *  
Age           -0.34075    0.01892 -18.012  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.521 on 3035 degrees of freedom
Multiple R-squared:  0.4726,	Adjusted R-squared:  0.4709 
F-statistic:   272 on 10 and 3035 DF,  p-value: < 2.2e-16

> par(mfrow=c(2,2))
> plot(LM,which = 1:4)
```



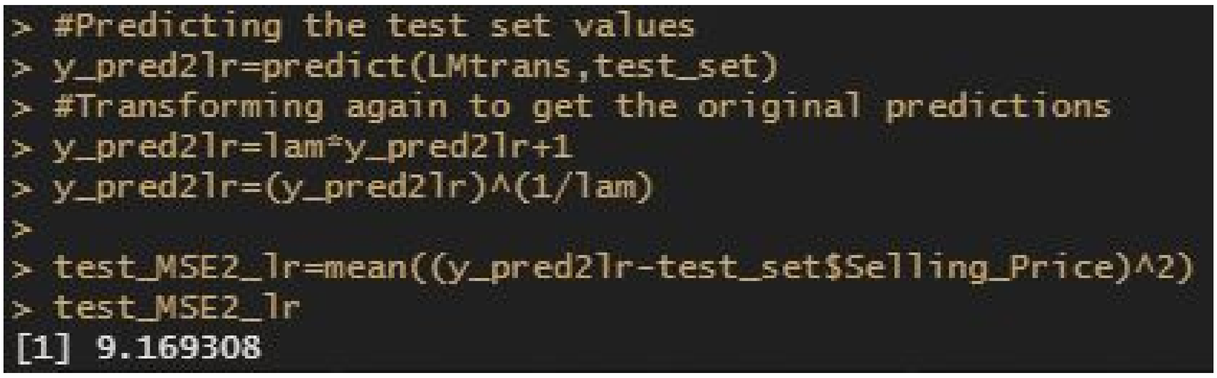
Now for checking further model assumptions,we use the same techniques as before.From the plots,we see that the Homoscedasticity assumption(i.e. the error variances are equal for all the observations) is not valid because the 'Residuals vs Fit' plot has some pattern,we have also some influential observations because the corresponding Cook's Distances are large.Though,whether we treat them as outliers or not,are subjective. Again,the value of the X^TX matrix is very large.Also,looking at the correlation plot,we also say that multicollinearity is also not present. So,as before,to get rid of heteroscedasticity,we perform a Box-Cox transformation on our response variable by the help of the 'MASS' library and we get the value of the lambda that provides us the lowest residual sum of squares.We then fit our multiple linear regression model with the transformed response variable with the optimum lambda value and summarize the model.
We observe from the plots,that homoscedasticity,normality and linearity is valid to a great extent and our model explains almost 61 percent of the variation that is present.We also observe that many of the regression coefficients become insiginificant after this transformation. The output of the part is given below.The code is shown in script 3.



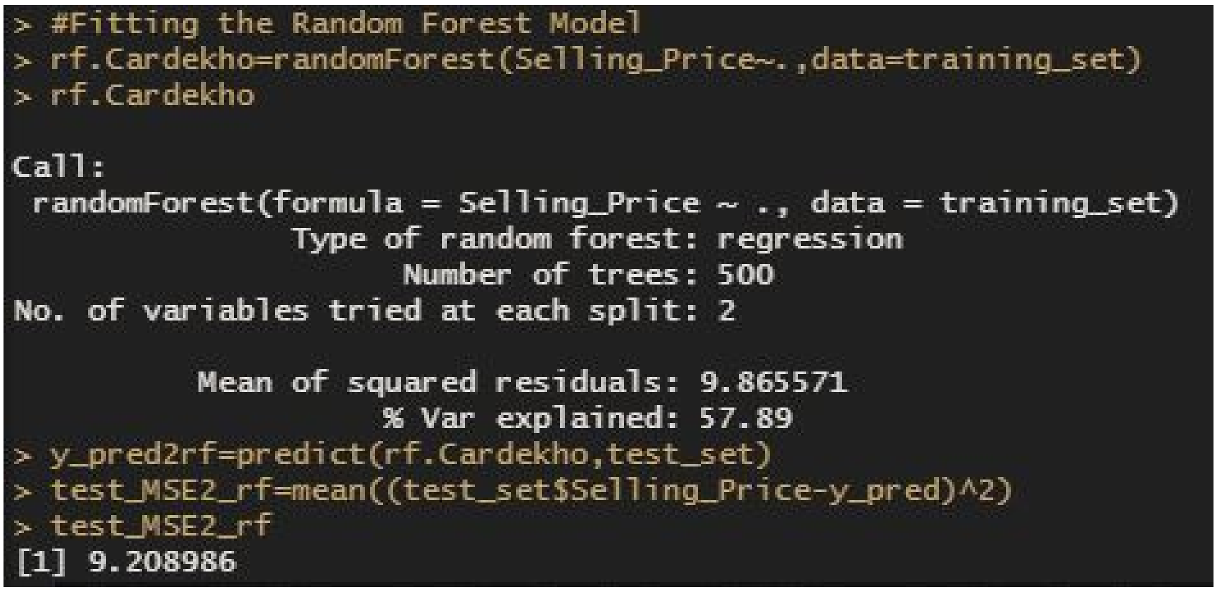
```
> #Box cox transformation and fitting again
> library(MASS)
> bc = boxcox(LM,lambda = seq(-3,3))
> lam=bc$x[which(bc$y==max(bc$y))]
> LMtrans=lm((((training_set$Selling_Price^lam)-1)/lam)~.,data=training_set)
> summary(LMtrans)

Call:
lm(formula = (((training_set$Selling_Price^lam) - 1)/lam) ~ .,
    data = training_set)

Residuals:
     Min       1Q   Median       3Q      Max
-3.06781 -0.32718  0.02138  0.36367  2.74123

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    2.049038   0.035937  57.018   <2e-16 ***
Kms_Driven     0.002658   0.013360   0.199   0.8423
Fuel_Type2     0.645816   0.024318  26.557   <2e-16 ***
Fuel_Type3    -0.022321   0.105546  -0.211   0.8325
Seller_Type2  -0.359043   0.027157 -13.221   <2e-16 ***
Transmission2  0.874686   0.038842  22.519   <2e-16 ***
Owner1         0.002690   0.028590   0.094   0.9250
Owner3        -0.050250   0.080894  -0.621   0.5345
Owner2        -0.078679   0.046186  -1.704   0.0886 .
Owner4         0.180120   0.154488   1.166   0.2437
Age           -0.123953   0.003261 -38.012   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.607 on 3035 degrees of freedom
Multiple R-squared:  0.6094,    Adjusted R-squared:  0.6081
F-statistic: 473.5 on 10 and 3035 DF,  p-value: < 2.2e-16

> par(mfrow=c(2,2))
> plot(LMtrans,which = 1:4)
```



We now predict the selling price of the cars in the test set.The mse,we see is 9.16.But,observing the wide ranges of car we have in the merged dataset,that's not too bad.We also fit a Random Forest model to see if that improves the model performance.But the MSE comes out to be 9.2,so both of the models in this case are similarly accurate.The code is shown in script 4.The output and the plots are below.



```
> #Predicting the test set values
> y_pred2lr=predict(LMtrans,test_set)
> #Transforming again to get the original predictions
> y_pred2lr=lam*y_pred2lr+1
> y_pred2lr=(y_pred2lr)^(1/lam)
>
> test_MSE2_lr=mean((y_pred2lr-test_set$Selling_Price)^2)
> test_MSE2_lr
[1] 9.169308
```



```
> #Fitting the Random Forest Model
> rf.Cardekho=randomForest(Selling_Price~.,data=training_set)
> rf.Cardekho

Call:
 randomForest(formula = Selling_Price ~ ., data = training_set)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 2

          Mean of squared residuals: 9.865571
                    % Var explained: 57.89
> y_pred2rf=predict(rf.Cardekho,test_set)
> test_MSE2_rf=mean((test_set$Selling_Price-y_pred)^2)
> test_MSE2_rf
[1] 9.208986
```

7 Difference between the Two Scenarios

So we can use either of the models to predict the selling price in the combined dataset scenario. The advantage of the Linear Model is that it explains slightly more variance and the advantage of the Random Forest here is it doesn't require feature scaling and model assumptions.
However,one thing which is really important is,in spite of having 10 times the observation the smaller dataset has,any of the models based on the combined dataset only manages explain a maximum of 61 percent variation.On the other hand,the smaller dataset almost explains 93 percent of the variation.That just depicts how important the variable 'Present Price' is in car price prediction.
However,the combined dataset is based on many observations,and while the model based on the smaller dataset is preferrable in the cases where the present price is available,we can use the models based on the combined dataset where present price of the car is not available.

8 Conclusion

So we are now able to predict the selling price of any car,for which we know about the other information.In many cases,some cars are discontinued by the company,and the ex-showrrom price of that car is not avaialble.In these circumstances,we are left with two options.Either we find an equivalent car model and take that car model's ex-showroom price and use the prediction model based on the smaller dataset or we can simply use the model based combined dataset.In real life,it is reasonable to assume that in most of the cases,we will be able to apply the model based on the smaller dataset and can get excellent predictions.Thus our aim is fulfilled here,and the models we developed will be very beneficial to individuals and dealers.