Predicting Car Prices with a Multiple Regression Approach

Benjamin Gerber, SID: 305163009

March 22, 2021

Abstract

The goal of this project was to create a multiple linear regression (MLR) model that would try to accurately predict the price of different cars. The fitted values were based upon different numeric and categorical variables that were taken into consideration. After choosing different variables to create a regression model, the data was put into a Kaggle competition that would rank the different models based upon R^2 values produced by the model on the test data. The goal was to create the best model possible for predicting car prices, but while still maintaining a valid and simple model. This would mean making sure that the assumptions for a valid regression model would be met. Furthermore, overfitting the data with an extremely complex model would result in a deduction in points. So, the goal of the competition was to create the best model that did not overfit the data all while maintaining validity.

My Kaggle name for the competition was "Benjamin Gerber Lecture 2." While the R^2 of my training model was 81.9%, the R^2 of the testing data was 81.4%. This R^2 placed me in 86^{th} out of 99^{th} on the leaderboard for the Kaggle competition. While this sounds like a bad placement, my model displays great validity, and it is a very simple and efficient model; it only used six betas. Due to this, I believe that it is a fantastic model.

Introduction

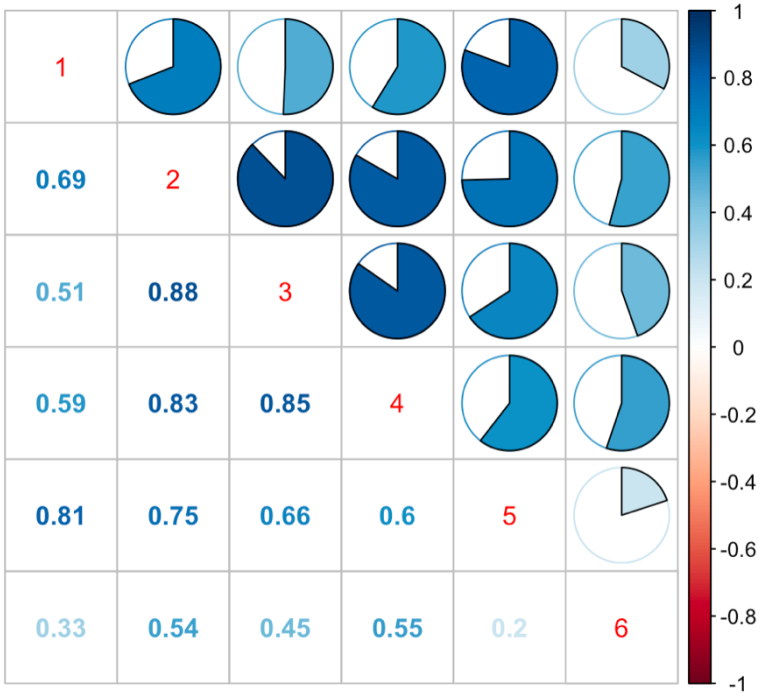
The goal of this Kaggle competition was to build a valid and regression model that would accurately predict the prices of cars based on the variables used in the model. The data came from a file called "carsTrain.csv" that had 23 different variables excluding the Price of the Car variable (PriceNew). They were as follows:

Manufacturer	Model	Type	MPG.Highway
AirBags	DriveTrain	Cylinders	EngineSize
Horsepower	RPM	Rev.per.mile	Man.trans.avail
Fuel.tank.capacity	Passengers	Length	Wheelbase
Width	Turn.circle	Rear.seat.room	Luggage.room
Weight	Origin	Make	

The variables highlighted in yellow were qualitative, while the rest were quantitative. The variables are very self-explanatory in their meaning. For example, the qualitative variable "Model" would be the model of the car in the training set. The variable "Horsepower" measured the horsepower of the car. In this training dataset, there were 1,500 observations that were used in creating the linear regression. I was able to choose from the 23 variables, or create new variables out of existing ones, to create the best linear regression possible.

Methodology

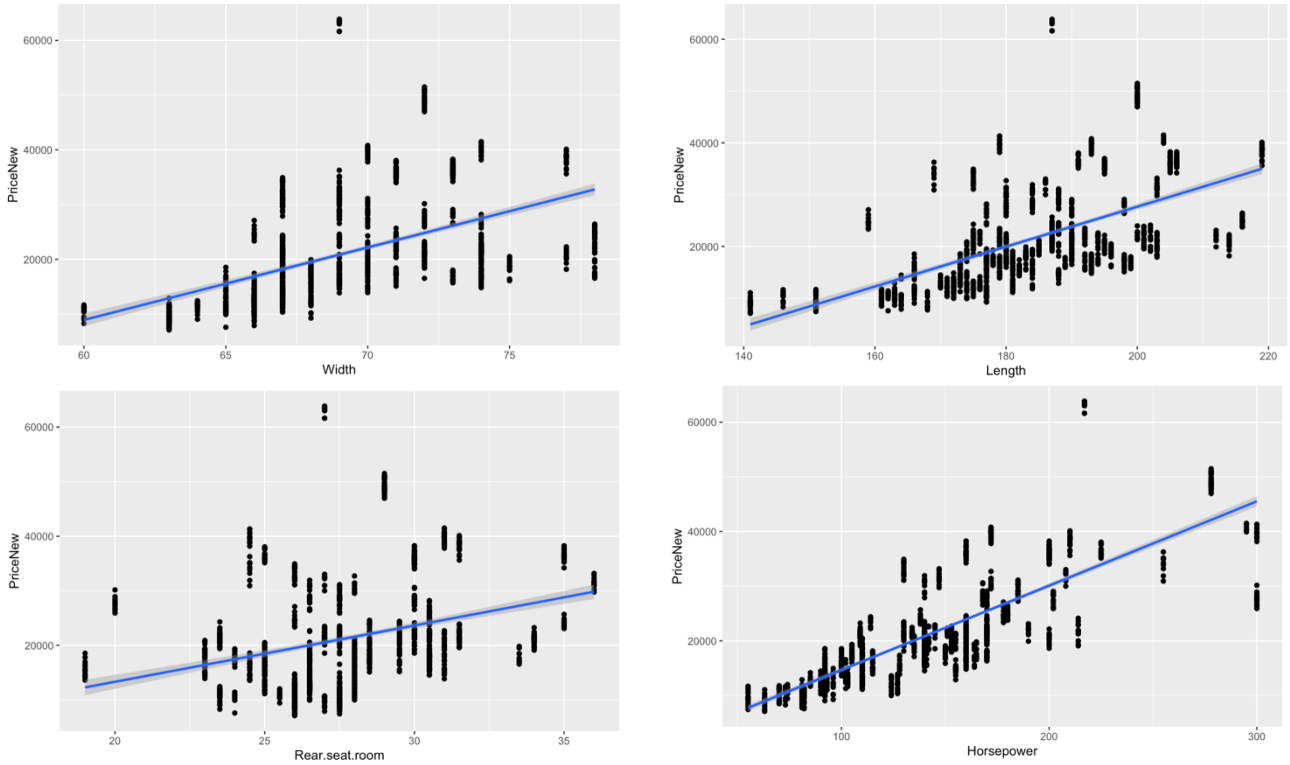
The first step in creating the best possible regression would be to choose the predictors that would be used in the regression. To do this, I looked at correlation plots between certain predictors and the data for the price of the car (PriceNew). So, I created a correlation plot of some of the variables I initially thought would be good predictors for PriceNew to see which of the variables could be my best predictors for PriceNew. Here is the graph:

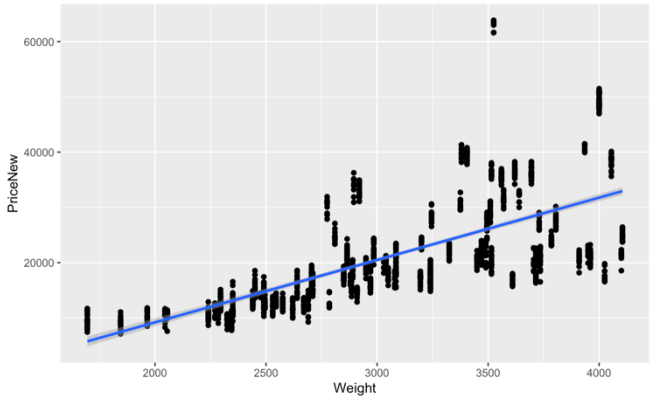


1. PriceNew
2. Width
3. Weight
4. Length
5. Horsepower
6. Rear.seat.room

The correlation plot shows that the variable most closely associated with PriceNew is Horsepower. So, I decided to use that as a predictor in my model. Next, I had to decide what I wanted the next variable to be for the regression. I decided upon Width because it had the next highest correlation with PriceNew. I was a little hesitant about its Correlation with Horsepower at .75, but I thought that it was worth it to add the predictor. I would check the VIF later to make sure that multicollinearity was not an issue. Although the variable with the

next highest correlation to PriceNew was Length, I chose not to use this as a predictor in my regression. I chose not to use Length because of its correlation with Width (0.83). This might have led to issues of multicollinearity, which would invalidate the model. Furthermore, Length and Width essentially measure the same thing; they are measure the dimensions of the car; it is clear why their correlation would be so high. Instead, I decided to use Weight as a predictor for the regression model because of its decent correlation with PriceNew. Although it has a high correlation with width (0.88), the practicality of a car's weight predicting its price makes sense. If a car weighs more, it takes more material to manufacture, and so the manufacturer would sell the car for more money. Lastly, I decided not to use Rear.seat.room as a predictor in my regression because of its low correlation with PriceNew. Below are graphs showing the relationship between the five variables listed above and Price New, which also helped me in choosing which variables to choose for my regression. A higher slope shows a greater relationship between the variable and PriceNew.





Clearly, from the graphs above, it can be seen that Rear.seat.room has the worst relationship with PriceNew. I also decided to add another variable in the regression, but this time a categorical variable. At first I tried adding Cylinders to the regression, but there were too many categories involved (see to the right). It would make the model too complex in my opinion. Also, when I ran a regression with the predictor Cylinders, I found that some of the betas that were created were not statistically significant. Another variable that seemed similar, but had less betas, was AirBags. I decided to use this categorical variable and see if it would do a better job than Cylinders.

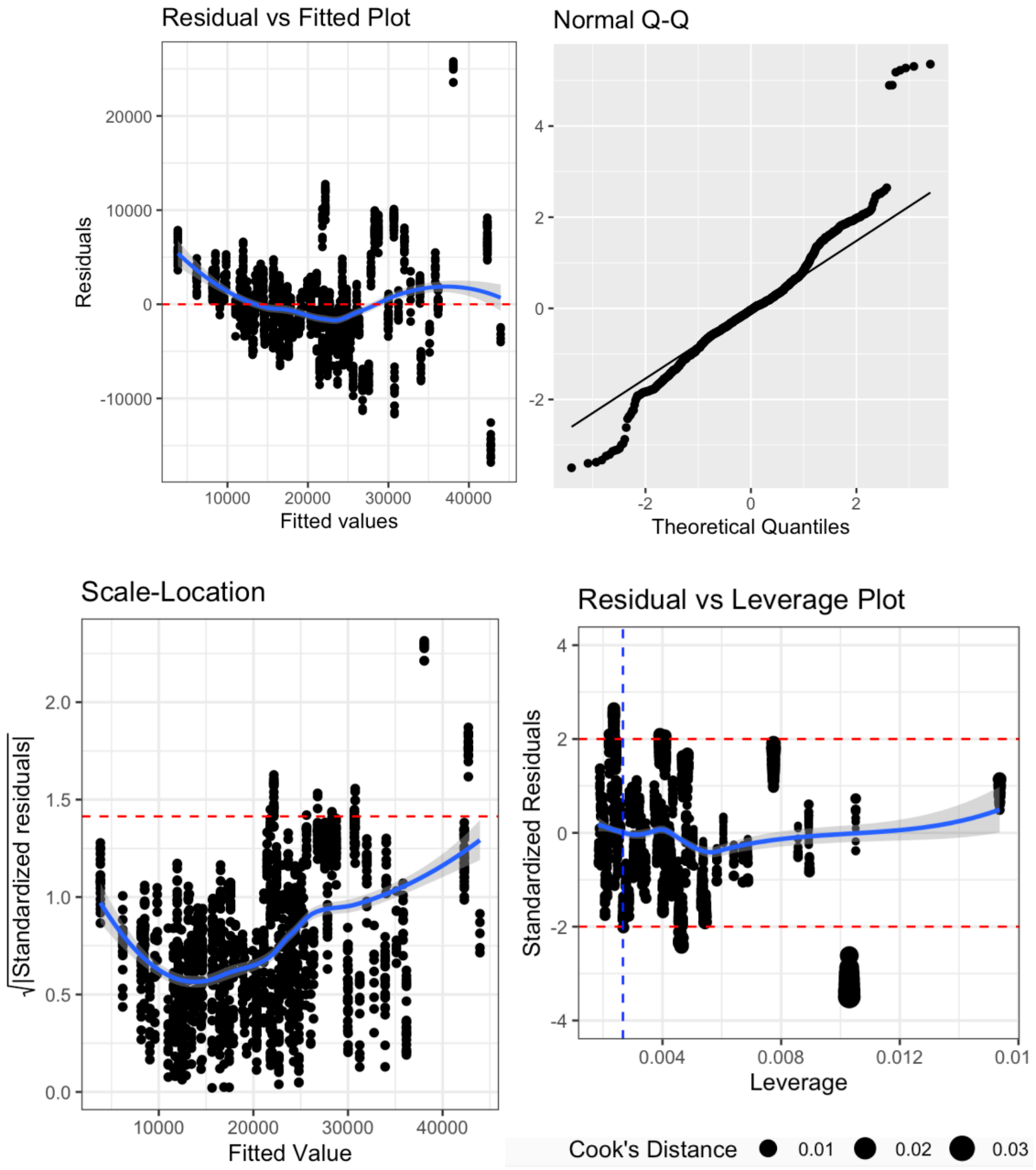
```
Call:
lm(formula = PriceNew ~ Width + Weight + Horsepower + Cylinders,
    data = carsTrain)

Residuals:
     Min       1Q   Median       3Q      Max
-14989.5  -2897.7   -198.6   2309.0  29198.0

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     74398.3422  4378.6862  16.991  < 2e-16 ***
Width           -1422.3987    78.5406 -18.110  < 2e-16 ***
Weight              9.5966     0.5924  16.199  < 2e-16 ***
Horsepower         94.3963     4.8779  19.352  < 2e-16 ***
Cylinders4        551.4953   790.9213   0.697 0.485735
Cylinders5       2356.4670  1218.8177   1.933 0.053376 .
Cylinders6       4087.6841  1066.6876   3.832 0.000132 ***
Cylinders8      10760.7364  1305.2050   8.244  3.6e-16 ***
Cylindersrotary  5859.5118  2010.9662   2.914 0.003624 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4982 on 1491 degrees of freedom
Multiple R-squared:  0.7422,    Adjusted R-squared:  0.7408
F-statistic: 536.5 on 8 and 1491 DF,  p-value: < 2.2e-16
```

The next step in creating the regression would be to test the predictors that I chose for the regression model. They were Weight, Width, Horsepower, and AirBags. Here is a picture of the regression output with no adjustments to the variables:

```
Call:
lm(formula = PriceNew ~ Width + Weight + Horsepower + AirBags,
    data = carsTrain)

Residuals:
     Min       1Q   Median       3Q      Max
-16805.2  -2598.0   -149.4   2313.0  25785.9

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)        70949.8690  3880.6370  18.283   <2e-16 ***
Width              -1326.0845    70.6649 -18.766   <2e-16 ***
Weight                 9.9813     0.4956  20.138   <2e-16 ***
Horsepower           107.9741     3.8301  28.191   <2e-16 ***
AirBagsDriver only -3119.9392   344.6144  -9.053   <2e-16 ***
AirBagsNone        -6434.7686   390.7257 -16.469   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4827 on 1494 degrees of freedom
Multiple R-squared:  0.7575,    Adjusted R-squared:  0.7566
F-statistic: 933.1 on 5 and 1494 DF,  p-value: < 2.2e-16
```

This model is already an improvement of the one shown above (that had Cylinders instead of AirBags) for a few reasons. First, it is simpler: the newer regression has only six betas while the other model had nine. Second, everything in the newer model is statistically significant, unlike in the regression with Cylinders. Finally, the newer model even has a higher value for R^2 (75.75% vs 74.22%). It is clear that the variable choice AirBags is better than Cylinders as a categorical predictor.

Next, I took a look at the validity of the model. Below are the plots that are used in viewing the validity of a model:



The first plot shows that the residuals are not super random. The average of the residuals fluctuate as the fitted values increase. The normal Q-Q graph shows that the normality assumption is being broken. For the normality assumption to be sound, most of the points on the plot need to be on the Normal Q-Q line. The standardized residual graph shows that the constant variance assumption is definitely being broken. As the fitted values first begin to increase, the value of the standardized residuals decrease, but later increase as the fitted values increase.

Finally, the residual versus leverage plot shows that there is a group of bad leverage points towards the bottom right of the graph.

Next, I wanted to check that the multicollinearity assumption was not violated. The appropriate way to do this is by checking the VIF of each variable of the model. Here are the results:

VARIABLE	**VIF**
Width	4.518
Weight	5.659
Horsepower	2.476
Airbags	1.395

The VIF's look good except for weight, which has a VIF above five. Depending on what the threshold is for a good or bad VIF, the value for the VIF of Weight could be interpreted differently. In many cases, the threshold is ten, but we used five as our threshold for a good VIF. So, this slightly violates the multicollinearity assumption. But, I made the choice not to take the variable Weight out of the regression because it lowered the R^2 value vastly from 75.75% to 69.16%. From a practical standpoint, it seemed important to keep the variable Weight as a predictor, especially since it was statistically significant in the model.

To check this, I ran a partial F-Test on the model with Weight versus the model without Weight as a predictor. Here are the results:

```
Analysis of Variance Table

Model 1: PriceNew ~ Width + Horsepower + AirBags
Model 2: PriceNew ~ Width + Weight + Horsepower + AirBags
  Res.Df        RSS Df  Sum of Sq      F    Pr(>F)
1   1495 4.4267e+10
2   1494 3.4816e+10  1 9450929439 405.55 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Since the P-value for this partial F-Test is so small, we should keep the full model. This partial F-Test is saying that dropping the variable Weight is significant. Consequently, I decided to keep the variable in the model.

To help with the validity of the model, specifically the normality assumption, I decided to use the Box-Cox Transformation to improve the model. The Box-Cox Transformation helps to normalize the densities of the variables in the regression. When I ran the function for the transformation, the results were as follows:

Variable	Estimated Power	Rounded Power
PriceNew	-0.5533	-0.50
Width	-2.2653	-2.00
Weight	0.2490	0.33
Horsepower	-0.3174	-0.33

Consequently, I decided that I should raise the specified variables to the rounded power (for simpler computation) to help validate my model. After toying around with the variables

though, I found that taking the predictor width to the -1 power was better for the regression because it increased the R^2 value more. You cannot run a Box-Cox transformation on a categorical variable, so I left the predictor AirBags as is. After these transformations, I was left with my final model.

Results

Based on the methodology above, here is the final model that I decided upon for my regression:

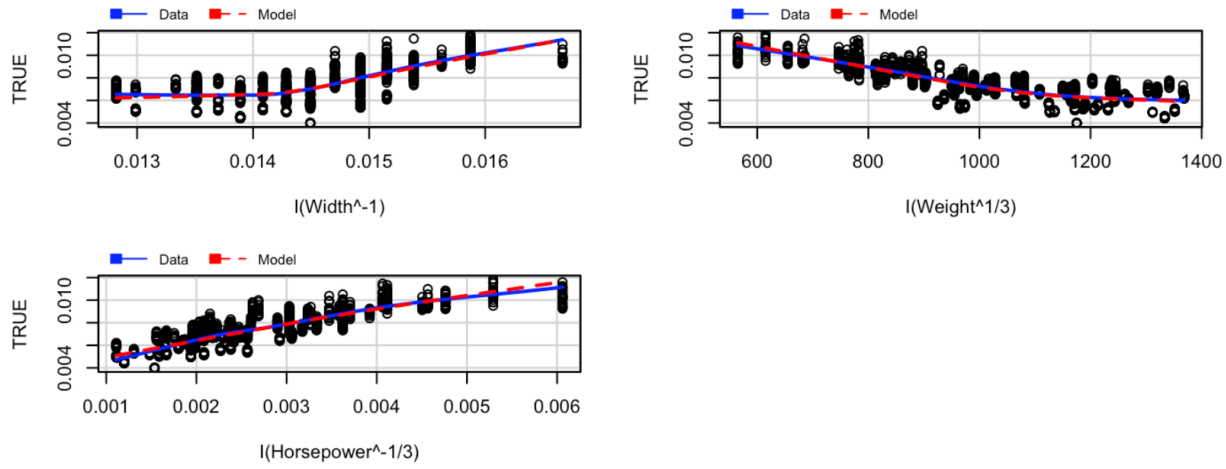
$$PriceNew = (2.051e-02) + (-7.225e-01)(Width^{-1}) + (-4.990e-06)\left(Weight^{1/3}\right) + (7.194e-01)\left(Horsepower^{-1/3}\right) + (2.712e-04)(AirBagsDriver\ only) + (9.790e-04)(AirBagsNone)$$

Discussion

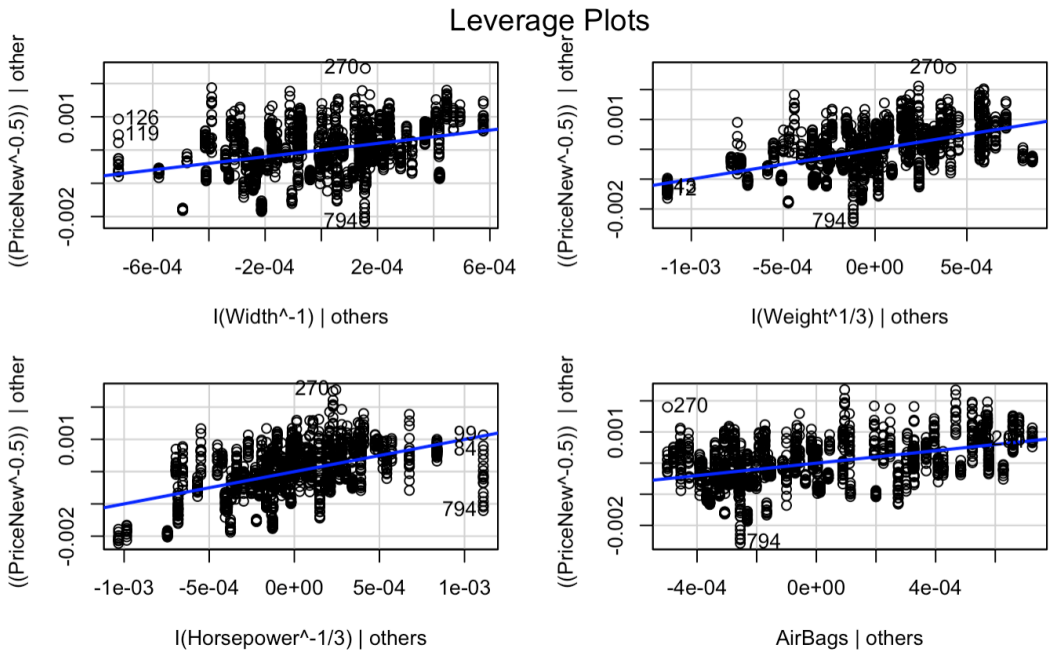
Below is the summary output of the regression in R:

```
Call:
lm(formula = ((PriceNew^-0.5)) ~ I(Width^-1) + +I(Weight^1/3) +
    I(Horsepower^-1/3) + AirBags, data = carsTrain)

Residuals:
       Min         1Q     Median         3Q        Max
-2.320e-03 -4.321e-04 -1.527e-05  4.534e-04  2.296e-03

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)         2.051e-02  8.724e-04  23.516  < 2e-16 ***
I(Width^-1)        -7.225e-01  4.801e-02 -15.048  < 2e-16 ***
I(Weight^1/3)      -4.990e-06  2.153e-07 -23.174  < 2e-16 ***
I(Horsepower^-1/3)  7.194e-01  3.311e-02  21.727  < 2e-16 ***
AirBagsDriver only  2.712e-04  4.739e-05   5.723 1.26e-08 ***
AirBagsNone         9.790e-04  5.489e-05  17.836  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0006608 on 1494 degrees of freedom
Multiple R-squared:  0.819,     Adjusted R-squared:  0.8184
F-statistic:  1352 on 5 and 1494 DF,  p-value: < 2.2e-16
```

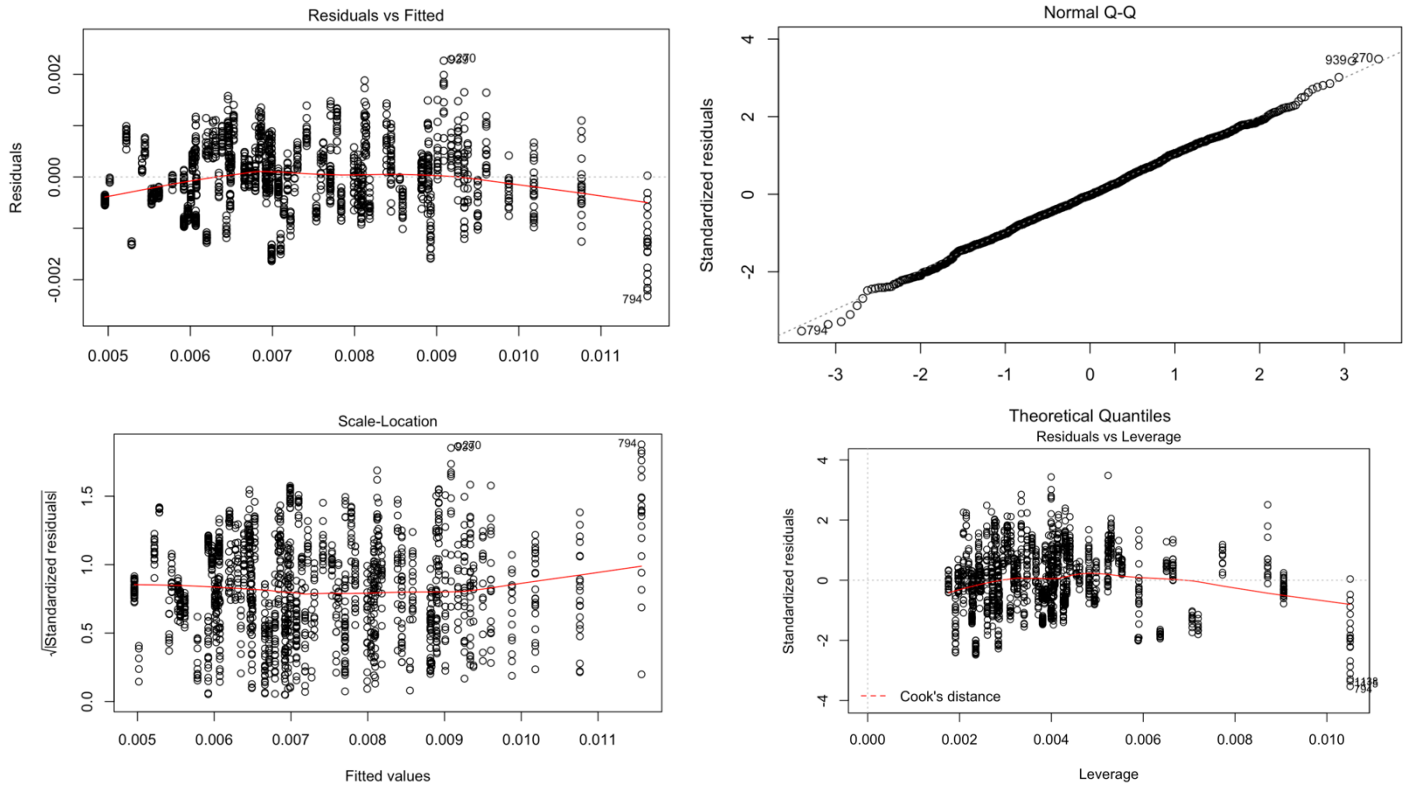
As shown, the R^2 value improved drastically, increasing from 75.75% to 81.9% without adding any predictors. Only the Box-Cox transformations helped the R^2 value for the regression to increase. All of the betas in the regression are significant as well which is a good sign. This may be because the chosen predictors did not have a very linear relationship with PriceNew so the new model fits the variables better. Now, the powers that each of the predictors were taken to may fit the data better. Here are Marginal Model Plots that show how well the new regression captures the numeric predictor relationships with PriceNew:



As shown, the model seems to capture each variable's relationship with PriceNew well. Furthermore, each of the variables help us to predict PriceNew on its own. This can be seen through leverage plots. These plots show the effect of each predictor of a regression on the predicted variable, given the other predictors. In the leverage plots for this regression, each of the variables has a decent positive slope with Horsepower having the steepest slope. Although the slope on the variable AirBags is not very steep, the regression summary shows that the predictor is significant in helping to predict the PriceNew variable. The leverage plots for this regression are shown below:



The regression was improved from an R^2 point of view, which is great, but the issue that I tried to fix with the Box-Cox transformation was the validity of the model. Here are the validity plots of the new model:



The greatest improvement in validity can clearly be seen in the Normal Q-Q plot. The normality assumption is clearly met now; it really looks great. The second noticeable improvement is in the top left graph, the residual vs fitted values plot. This plot shows that the residuals look much more random than before, as well as the average line staying close to zero. Furthermore, another plot that shows improvement in the model is the Scale-Location plot. This plot shows that the final model has a much more constant variance than before. The constant variance assumption is definitely improved upon. The final bottom right plot shows that most of the datapoints land in a large cluster to the left, but there are definitely still a few points to the far right that could be bad leverage points.

Limitations and Conclusion

There were many limitations in the process of trying to create this regression. Most noticeably, the biggest challenge was to create a great regression while still maintaining validity and keeping the model as simple as possible. Once I had the predictors that I decided upon for my final model (Weight, Width, Horsepower, AirBags) I tested adding many others. But I kept running into issues. Sometimes the new predictor would mess up the validity of the model, other times the model would have betas that were not significant. And most importantly to me, I did not want to sacrifice validity in my model when the R^2 value for the model would only go up by 0.5% or so. Furthermore, it was not worth it to lose the simplicity of my model. It is extremely important to try and keep the model as simple as possible, otherwise it becomes difficult to tell what is actually predicting the predicted variable.

The model that I created does a great job predicting the variable PriceNew. The R^2 on the training data was 81.9%. When the model was used on the Test data, though, the R^2 dropped to

81.4%, which is not less by a large amount. Even though I ranked 86th out of 99th on the Kaggle leaderboard, there are many reasons why my model is great. It is a very valid model, which was shown in the Discussion section with the plots of the final model. This is extremely important, because if the model lacks validity, then does it even matter if the R^2 score is high? So, I sacrificed a little bit of R^2 to keep my model valid. Even more, my model was very simple, with only six betas. The model used only two numeric predictors and one categorical predictor (with three different categories). Overall, the final model I created does a incredible job of predicting the variable PriceNew while maintaining validity and staying very simple.

Requirements

1. Name: Benjamin Gerber

   SID: 305163009

   Kaggle Nickname: Benjamin Gerber Lecture 2
2. Kaggle Rank: 86th out of 99th
3. Kaggle R^2 Score: 81.4%
4. Number of Predictors 2 Numeric, 1 Categorical (3 categories)
5. Total Number of Betas: 6
6. BIC Score of Model: -21928.49

References

"STAT101A US Car Prices." *Kaggle*, www.kaggle.com/c/stat101a-us-car-prices/leaderboard.

RStudio Team (2020). RStudio: Integrated Development for R. RStudio, PBC, Boston, MA URL http://www.rstudio.com/.