ANALYSIS ON THE FACTORS PREDICTING THE USED-CAR PRICES: MILEAGE, YEAR, HORSEPOWER, DAYS ON THE MARKET

MEICHAN HUANG

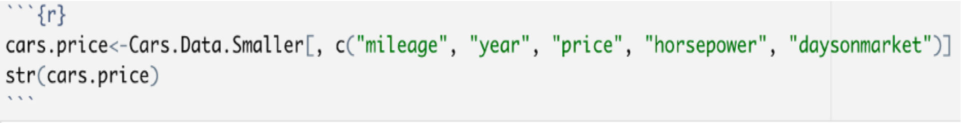
The rationale behind this analysis is to identify a model that would appropriately and reasonably price a used car within the dealership's inventory, so that it reflects the current market with used cars around the states, maintains profit margin for the company, and minimizes the loss of car lot real-estate by fast sales. Several candidate factors were proposed, including the cars' mileage, year, cars' horsepower, days on market. These factors were chosen based on two business decision-making aspects: (1) customers' needs (mileage, year, a car's horsepower) and (2) the flow of the inventory (days on the market. The following business question is asked:

Business Question:

How do the number of mileages, years, horsepower, days on market predict the price of the used cars, so that the dealer can adjust the trajectory of predicted price for the used cars?

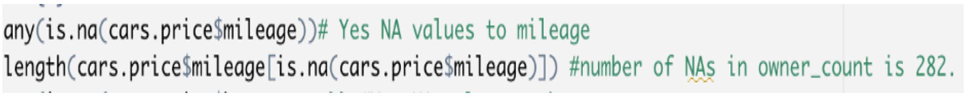
Data Cleansing, Transformation, Architecture

To answer my business question, I first extracted the subset of data named **cars.price** that contains only the attributes that I need for the analysis and conducted a brief inspection, using the following syntax (more details see Appendix – B)



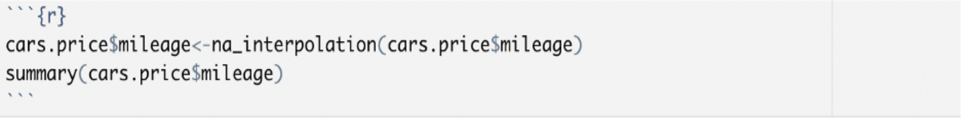
~~~
```{r}
cars.price<-Cars.Data.Smaller[, c("mileage", "year", "price", "horsepower", "daysonmarket")]
str(cars.price)
```
~~~

A brief inspection of the data showed that there were NA values in the dataset. The next step is to clean up the empty values amongst the attributes. To determine the best way to remove the NAs in the data, I have first used the following syntax, for instance:



```
any(is.na(cars.price$mileage))# Yes NA values to mileage
length(cars.price$mileage[is.na(cars.price$mileage)]) #number of NAs in owner_count is 282.
```

Within the dataset, two out of five attributes, mileage and horsepower, contain 282 and 209 out of 5315 observations respectively. The optimal way to remove the NA is to use the **na_interpolation** to replace the NAs with a mean value of the attribute, for instance:

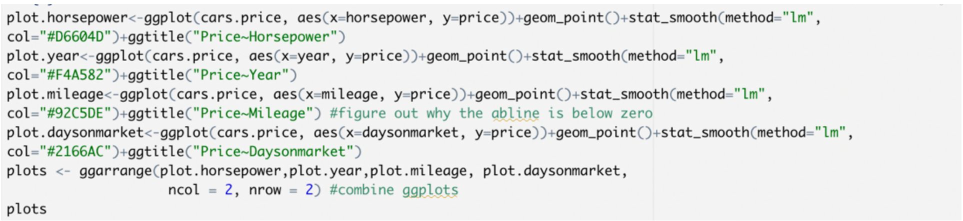


~~~
```{r}
cars.price$mileage<-na_interpolation(cars.price$mileage)
summary(cars.price$mileage)
```
~~~

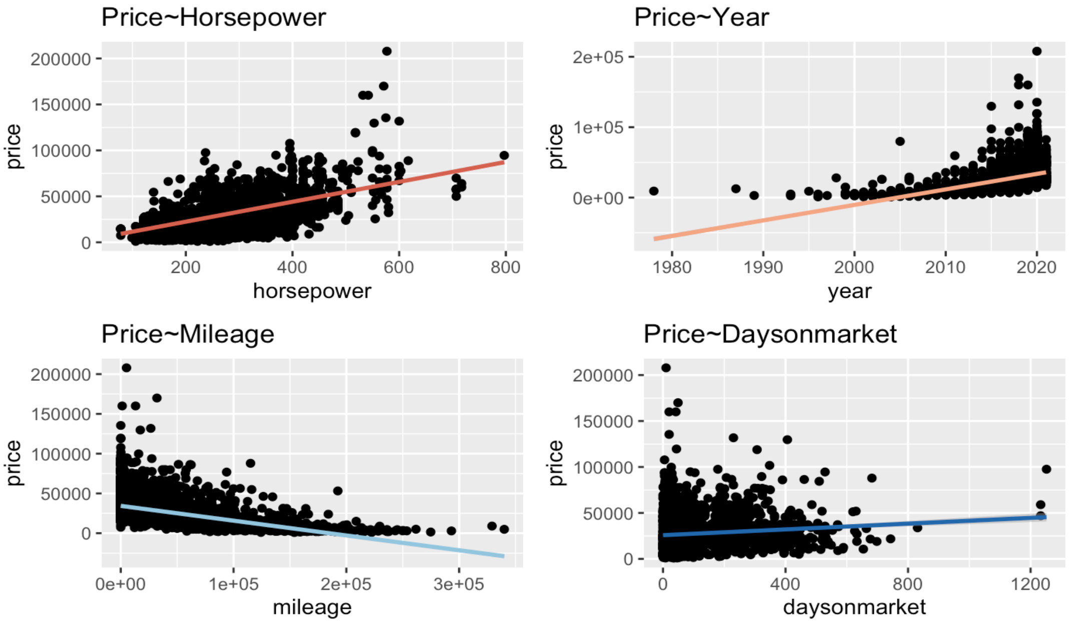
Then the dataset is now prepared for visualization and further analysis to understand the relationship between the used car prices and their mileage, model year, horsepower, and days on market.

Visualization

First, to understand the correlation between each independent variable and the dependent variable, I started with exploring the relationship between individual independent variable (mileage, year, horsepower, and days on market) and price using **ggplot()+geom_point()**. The syntax is as follows (more details in Appendix – B):



```
plot.horsepower<-ggplot(cars.price, aes(x=horsepower, y=price))+geom_point()+stat_smooth(method="lm",
col="#D6604D")+ggtitle("Price~Horsepower")
plot.year<-ggplot(cars.price, aes(x=year, y=price))+geom_point()+stat_smooth(method="lm",
col="#F4A582")+ggtitle("Price~Year")
plot.mileage<-ggplot(cars.price, aes(x=mileage, y=price))+geom_point()+stat_smooth(method="lm",
col="#92C5DE")+ggtitle("Price~Mileage") #figure out why the abline is below zero
plot.daysonmarket<-ggplot(cars.price, aes(x=daysonmarket, y=price))+geom_point()+stat_smooth(method="lm",
col="#2166AC")+ggtitle("Price~Daysonmarket")
plots <- ggarrange(plot.horsepower,plot.year,plot.mileage, plot.daysonmarket,
                   ncol = 2, nrow = 2) #combine ggplots
plots
```



The scatterplots show that horsepower, year, and days on the market have a positive correlation with the used car price. It is obvious that the higher the horsepower and the newer the manufacture year, the higher the used car price. On the other hand, mileage has a negative impact on the car price.

Interestingly, there is a slight upward slop in the best fit line for price ~ days on market, which means the longer the car is on the market, the higher the price. Possibly explanations are: (1) the cars were not priced right, possibly too high for their actual worth, or (2) the car could be a luxurious car for most customers so that it has a narrower clientele.

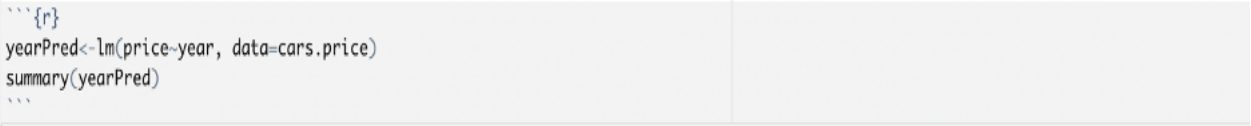
Nevertheless, with visualization, it is still hard to determine each individual factor's actual impact on car prices; hence, further correlation studies need to be run.

Analysis

In this section, I detail the codes and analysis of my business question. For the analysis, I first conducted separate simple linear regressions for each independent variable against the dependent variable price. Based on the preliminary results, I further narrowed the scope of the research by eliminating the less impactful independent variable(s) from the model. Then, I trained the model using multiple linear regression to see how accurate the model predicts the used car price. Lastly, I created a new test set to see how the dealership could price the used cars based on the input given.

1. Results for price ~ mileage, year, horsepower, days on market using lm ()

First, I ran simple linear regressions on price ~ mileage, year, horsepower, days on market. An example of the code is as followed (more codes, see Appendix – B):



~~~
```{r}
yearPred<-lm(price~year, data=cars.price)
summary(yearPred)
```
~~~

Overall, the results showed that horsepower, year, and mileage all have relatively high impact on the used car price. More specifically, horsepower has the most impact on the price (R2= 0.33, F(1, 5313) = 2638, p < 0.001), which means horsepower along counts for 33% variance in the car price. Every degree of change in horsepower will increase the price by 108.35 dollars, based on the coefficient of the model. The model year itself counts for 30% of variance in the car price (R2= 0.30, F(1, 5313) = 2294, p < 0.001). Similarly, mileage counts for 29% of variance in the car price (R2= 0.29, F(1, 5313) = 2190, p < 0.001).

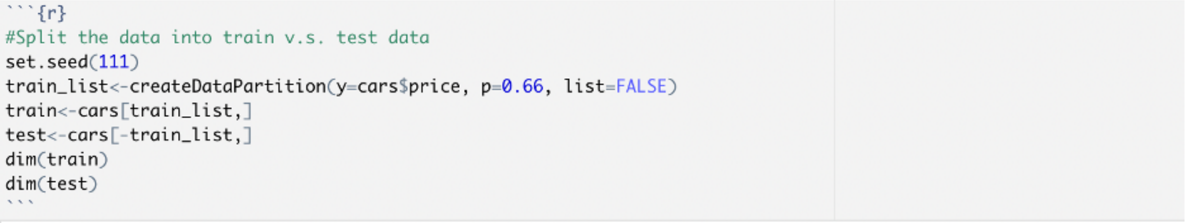
The only exception is the attribute "days on market", which only counts for 1% of the variance in the used car price (R2= 0.01, F(1, 5313) = 58.51, p < 0.001). Although the result is significant, the factor does not seem to have a significant impact on the used car's price.

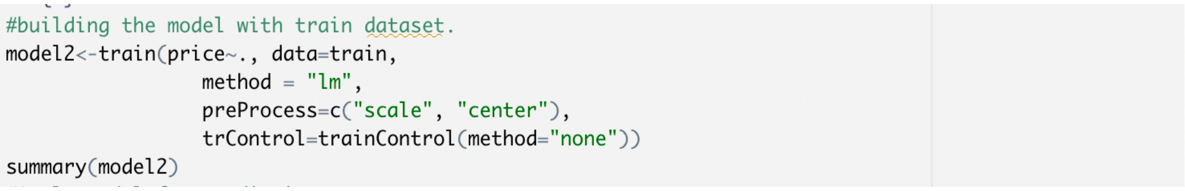
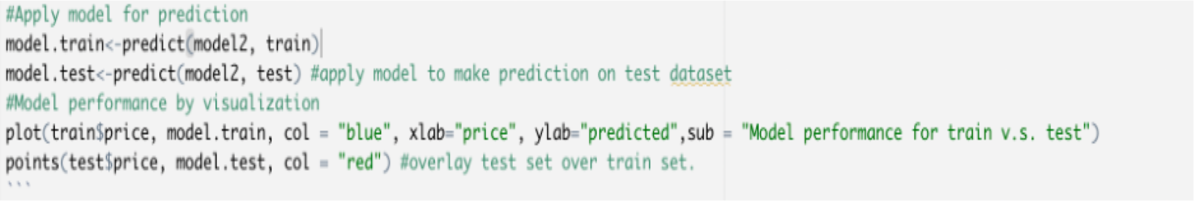
Based on the results, only horsepower, year, and mileage are included for further analysis. The factor, days on market, was excluded from the final model.

2. Results for train (price.~, model = "lm")

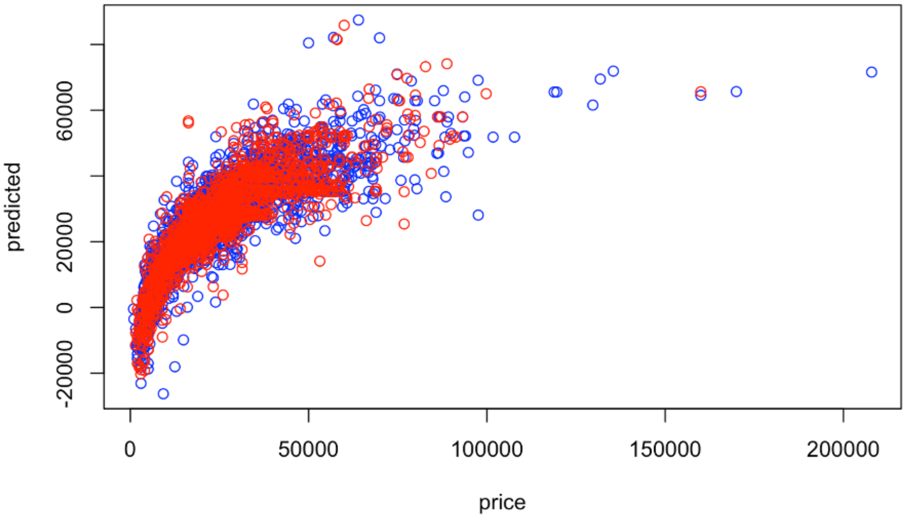
Second, based on the previous results, I adopted a machine learning model using three factors: horsepower, year, and mileage.

The first step in this process is to split the data into train v.s. test datasets using **createDataPartition().** The original dataset is divided into two subsets with 3511 and 1804 observations respectively.



~~~
```{r}
#Split the data into train v.s. test data
set.seed(111)
train_list<-createDataPartition(y=cars$price, p=0.66, list=FALSE)
train<-cars[train_list,]
test<-cars[-train_list,]
dim(train)
dim(test)
~~~
```

Next, I built the model by training the data with the linear regression (lm) method. The results showed that all three variables together explain 65.14% of the variance in the used car price (R^2= 0.65, *F*(3, 3507) =2187, p < 0.001). Although the model could be improved by adding other factors, it remains a fairly good model.



```
#building the model with train dataset.
model2<-train(price~., data=train,
                  method = "lm",
                  preProcess=c("scale", "center"),
                  trControl=trainControl(method="none"))
summary(model2)
```

To test whether this machine learning model is accurate, I applied the model for the prediction and plotted the model performance using **plot()** to plot the scattered plot for the train data performance with the model and overlaid the performance of the test model using **point().**



```
#Apply model for prediction
model.train<-predict(model2, train)
model.test<-predict(model2, test) #apply model to make prediction on test dataset
#Model performance by visualization
plot(train$price, model.train, col = "blue", xlab="price", ylab="predicted",sub = "Model performance for train v.s. test")
points(test$price, model.test, col = "red") #overlay test set over train set.
```
```

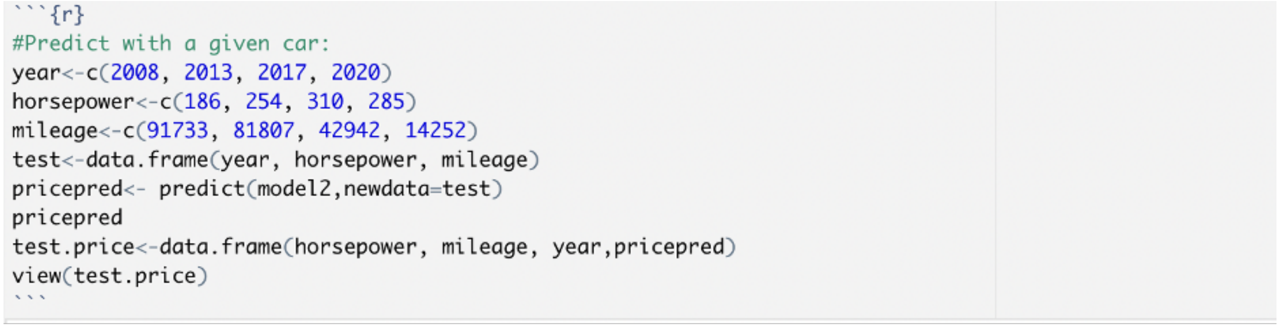
The results of the performance plots (see blows) showed that relatively high performance accuracy for the model with both train and test sets. The computation of the Pearson's correlation coefficient for both outputs are very similar to each other as well ($R^2{}_{\text{Train}}$ = 0.65, $R^2{}_{\text{test}}$ = 0.66).



Model performance for train v.s. test

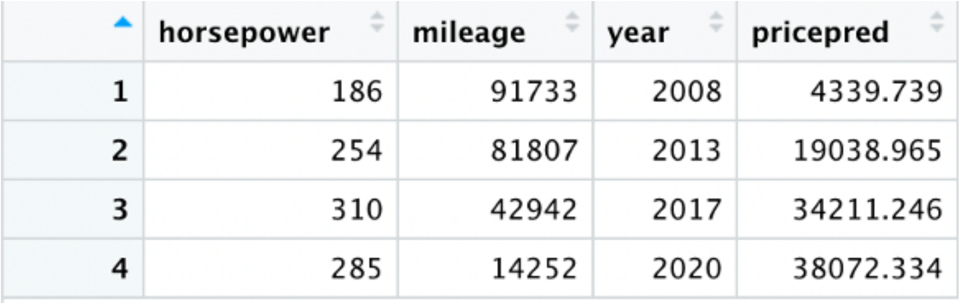
### (3) Predicting car prices based on the new car inventory using predict()

In this step, I create a new data frame with the indices of four incoming cars from the dealership. Using the following code, I was able to obtain the approximate price for the four cars based on the model. I first started with creating the data frame using **data.frame()**. From there, I used the following code to run the prediction based on the model generated in the previous step:



~~~
```{r}
#Predict with a given car:
year<-c(2008, 2013, 2017, 2020)
horsepower<-c(186, 254, 310, 285)
mileage<-c(91733, 81807, 42942, 14252)
test<-data.frame(year, horsepower, mileage)
pricepred<- predict(model2,newdata=test)
pricepred
test.price<-data.frame(horsepower, mileage, year,pricepred)
view(test.price)
```
~~~

The results show that the lowest price is with the first car which has the lowest horsepower, longest mileage, and oldest manufacture year. Another interesting pricing is with the third v.s. fourth cars. Although the 3rd car has a significantly greater number of mileage and is 3 years older than the fourth, the pricing is quite close to the 4th one, by less than 4000-dollar differences, due to its higher horsepower.



	horsepower	mileage	year	pricepred
1	186	91733	2008	4339.739
2	254	81807	2013	19038.965
3	310	42942	2017	34211.246
4	285	14252	2020	38072.334

Interpretation

Based on the results, we know that the factor, days on the market, may not be an important factor that drives car prices. Hence, as a business, we should consider not adjusting price based on how long it has sat on the lot. However, other business decisions with cars with longer days on the market should be considered as well. As other team members rightly point out, having a car that does not sell but occupies the lot would have an impact on restocking cars that are more popular and sell faster.

Three major factors, namely horsepower, year, and mileage are the three major predictors of a used car's price, which count for more than 65% of the pricing factors. Therefore, when pricing a car at the dealership, these factors should be considered.

Nevertheless, we need to proceed with caution with this model. Further analysis is needed to understand additional factors that would drive the used car prices to improve the performance of the current model we built. For instance, factors such as car make and model (Toyota Tuscon v.s. Ford F150, interior features (leather/fabric, sound systems, etc.), and car types (sedan, crossover, etc.) may have a significant impact on car prices, but it is out of the scope for this research. Therefore, future fine tuning of the model will be needed.