Geely Auto

Car Price Prediction

Christian Gonçalves

2023-01-8

Contents

1.Introduction

A Chinese car company aspires to enter the north-american market by installing a manufacturing unit there and producing the cars locally to give competition to their counterparts.

Based on various market surveys, the consulting firm has gathered a large data set of different types of cars across the America market.

The Original Dataset can be downloaded at: "https://www.kaggle.com/datasets/hellbuoy/car-price-prediction?resource=download".

After getting the dataset, many changes were made in order to optimize and access better data inputs to utilize in this project. By transforming the data these were the results of each step:

1. df - Original dataframe;
2. df1 - Dataframe after Exploratory and Descriptive Analysis (EDA);
3. df2 - Dataframe for the Correlation Graph;
4. df3 - Dataframe used for the Shiny app;
5. p1 - Predictive Dataframe using GBM_SM as the MLM;
6. p2 - Predictive Dataframe using GBM_BM as the MLM;

7. p3 - Predictive Dataframe using DRF_SM as the MLM;
8. p4 - Predictive Dataframe using DRF_BM as the MLM;

2.Problem Assignment

Specifically, they want to comprehend the factors that influence the US car market, which is vastly different compared to the Chinese market. These will be the objectives that Geely Auto wants to know:

- Which variables have the most impact in the car price prediction;
- How well do those variables describe the price of a car;

So, we are required to use MLM in order to predict the price of a car, with the available independent variables.

3.Success Criteria

As for success criteria, i have decided to make it so:

- The predictor model will be able to predict the car price within 1000$ mean range of error (over or bellow the real price)

4. Exploratory and Descriptive Analysis

4.1. Get libraries and dataset

4.2 Problem identification and solution step

We verified that:

1. Change the car name "alfa-romero" to "alfa" in order to ensure that we don't get a problem that i´ll state later in this report;
2. The column "Carname" is inefficient, so i´ll create to new columns in order to filter all the brands and models in different columns;
3. Some brands and models are misspelled, so they will be corrected Note: I changed the "alfa-romero" brand first because, if i did it in this step, the model of the car would be "romero", and not for example "giulia";
4. The brand subaru is missing 2 models, so we need to check that out;
5. The following columns are in categorical form, when they should be Integers: doornumber and cylindernumber;
6. It would be interesting to have a average MPG;
7. Get rid of Carid column, no need for unique cars in this dataset;
8. There is no currency stated in the price variable so its needing a refresh;

Let´s start by correcting the alfa Brand...

4.3. Problem Solving

###4.3.1. Orthographic correction of the Alfa-romero

```
df1 <- within (df, {
  CarName[CarName == 'alfa-romero giulia'] <- "alfa giulia"
  CarName[CarName == 'alfa-romero stelvio'] <- "alfa stelvio"
  CarName[CarName == 'alfa-romero Quadrifoglio'] <- "alfa quadrifoglio"
})
```

Next up we have to divide the CarName column into two separate columns...

###4.3.2. Separation of the CarName into two new columns

```
df1 <- df1 %>% separate(CarName, c('Brand', 'Model'))
```

With the creation of two new columns, we need to address the new missing values with the Brand "subaru", but firstly i will solve the following step that´s dedicated dedicated for orthographic errors inside the Brand column...

###4.3.3. Orthographic correction in Brand column

```
df1 <- within(df1,{
  Brand[Brand == 'vw'] <- "volkswagen"
  Brand[Brand == 'maxda'] <- "mazda"
  Brand[Brand == 'porcshce'] <- "porsche"
  Brand[Brand == 'vokswagen'] <- "volkswagen"
  Brand[Brand == 'toyouta'] <- "toyota"
  Brand[Brand == 'Nissan'] <- "nissan"
})
```

Now that the spellings are correct, it´s time to address the "subaru problem"...

###4.3.4 Addresing the missing Subaru models

```
df1$Model <- ifelse(df1$Brand == 'subaru' & is.na(df1$Model), 'impreza', df1$Model)
```

Next up, we have to change the variable types that we mentioned before...

###4.3.5. Changing the data types of the variables "cylindernumber" and "doornumber"

```
df1$cylindernumber <- as.numeric(ifelse(df1$cylindernumber == 'two', 2,
                                        ifelse(df1$cylindernumber == 'three', 3,
                                               ifelse(df1$cylindernumber == 'four', 4,
                                                      ifelse(df1$cylindernumber == 'five', 5,
                                                             ifelse(df1$cylindernumber == 'six', 6,
                                                                    ifelse(df1$cylindernumber == 'eight
                                                                           ifelse(df1$cylindernumber ==

df1$doornumber <- as.numeric(ifelse(df1$doornumber == 'two', 2,
                                    ifelse(df1$doornumber == 'four', 4, 4)))
```

Now that´s complete we can now create a new column based on the overall mpg of the cars...

###4.3.6. Creating a new column meanmpg

```
df1$meanmpg <- rowMeans(df1[,c('citympg', 'highwaympg')], na.rm=TRUE)
```

Almost forgot, we don´t need this column for the purpose of the prediction:

###4.3.7. Discarding the column Car_ID

```
df1$car_ID = NULL
```

Also we can add the currency:

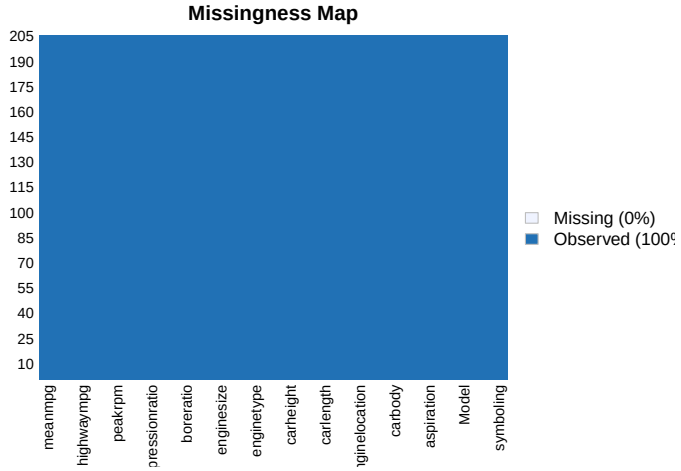
###4.3.8. Adding a currency to the price column

```
currency(df1$price, digits = 0L)
```

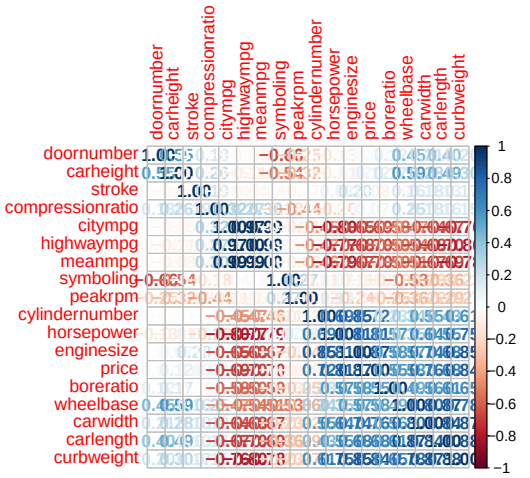
```
##   [1] $13,495 $16,500 $16,500 $13,950 $17,450 $15,250 $17,710 $18,920 $23,875
##  [10] $17,859 $16,430 $16,925 $20,970 $21,105 $24,565 $30,760 $41,315 $36,880
##  [19] $5,151  $6,295  $6,575  $5,572  $6,377  $7,957  $6,229  $6,692  $7,609
##  [28] $8,558  $8,921  $12,964 $6,479  $6,855  $5,399  $6,529  $7,129  $7,295
##  [37] $7,295  $7,895  $9,095  $8,845  $10,295 $12,945 $10,345 $6,785  $8,916
##  [46] $8,916  $11,048 $32,250 $35,550 $36,000 $5,195  $6,095  $6,795  $6,695
##  [55] $7,395  $10,945 $11,845 $13,645 $15,645 $8,845  $8,495  $10,595 $10,245
##  [64] $10,795 $11,245 $18,280 $18,344 $25,552 $28,248 $28,176 $31,600 $34,184
##  [73] $35,056 $40,960 $45,400 $16,503 $5,389  $6,189  $6,669  $7,689  $9,959
##  [82] $8,499  $12,629 $14,869 $14,489 $6,989  $8,189  $9,279  $9,279  $5,499
##  [91] $7,099  $6,649  $6,849  $7,349  $7,299  $7,799  $7,499  $7,999  $8,249
## [100] $8,949  $9,549  $13,499 $14,399 $13,499 $17,199 $19,699 $18,399 $11,900
## [109] $13,200 $12,440 $13,860 $15,580 $16,900 $16,695 $17,075 $16,630 $17,950
## [118] $18,150 $5,572  $7,957  $6,229  $6,692  $7,609  $8,921  $12,764 $22,018
## [127] $32,528 $34,028 $37,028 $31,401 $9,295  $9,895  $11,850 $12,170 $15,040
## [136] $15,510 $18,150 $18,620 $5,118  $7,053  $7,603  $7,126  $7,775  $9,960
## [145] $9,233  $11,259 $7,463  $10,198 $8,013  $11,694 $5,348  $6,338  $6,488
## [154] $6,918  $7,898  $8,778  $6,938  $7,198  $7,898  $7,788  $7,738  $8,358
## [163] $9,258  $8,058  $8,238  $9,298  $9,538  $8,449  $9,639  $9,989  $11,199
## [172] $11,549 $17,669 $8,948  $10,698 $9,988  $10,898 $11,248 $16,558 $15,998
## [181] $15,690 $15,750 $7,775  $7,975  $7,995  $8,195  $8,495  $9,495  $9,995
## [190] $11,595 $9,980  $13,295 $13,845 $12,290 $12,940 $13,415 $15,985 $16,515
## [199] $18,420 $18,950 $16,845 $19,045 $21,485 $22,470 $22,625
```

```
##  num [1:205] 13495 16500 16500 13950 17450 ...
```

Time to check for missing values in our dataframe...



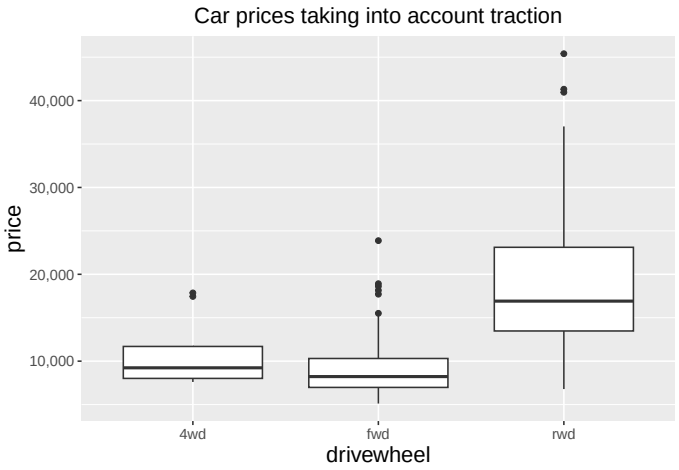
With no missing values, the next step will be checking the variable correlation by executing a corrplot...



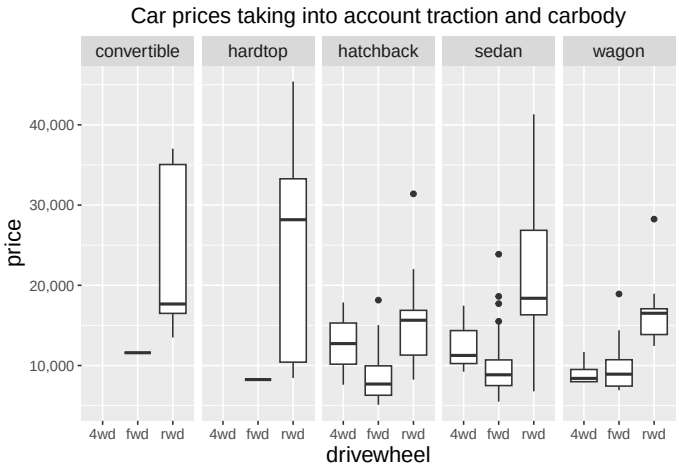
The price of the car has the highest positive correlation with the engine size, curbweight and horsepower, however it´s very important that we mention the wheelbase, carlength, carwidth and boreratio are also positively correlated just not as highly correlated as the first mentioned variables.

Let´s check on some of the most meaningful variables in a few plots...

4.4. Plotting

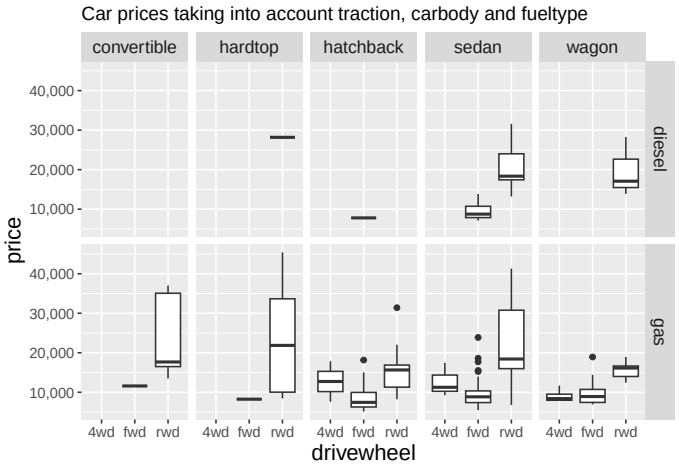


We can see that on average rwd cars are more expensive than 4wd cars and 4wd cars are slightly more expensive than fwd cars. We can make this plot better by adding the carbody variable to it ....

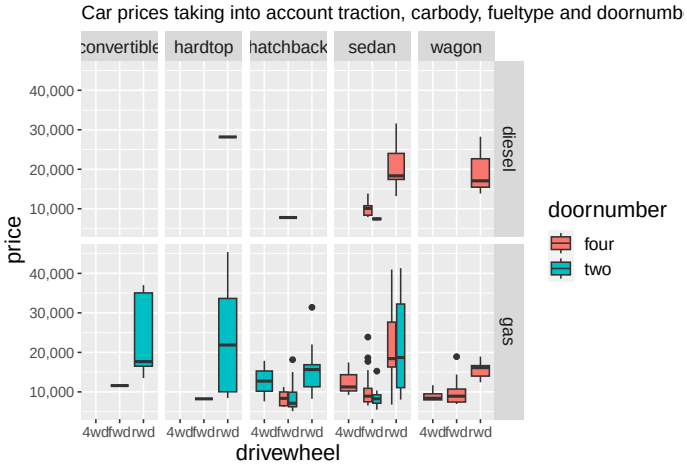


Here we can see that on average hard-top cars are the most expensive. Also, convertible and hardtop cars are mostly rwd.

We can also add the fueltype variable...



By adding this variable we can see that on average there are more gas cars than diesel based cars. Finally we add the last variable: doornumber

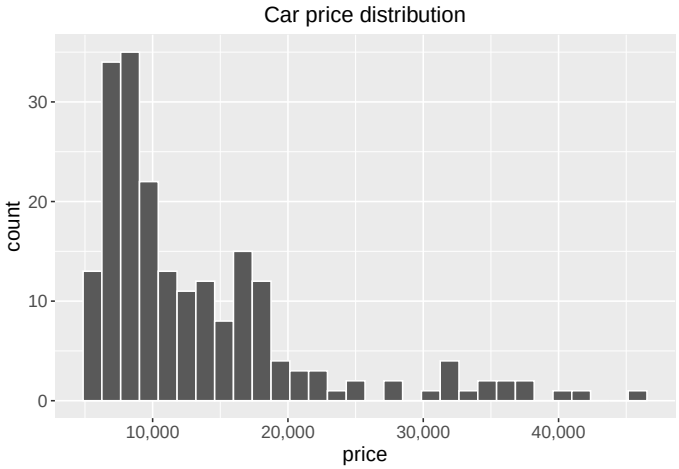


By adding this variable it´s noticeable that hardtop and convertible cars have two doors.

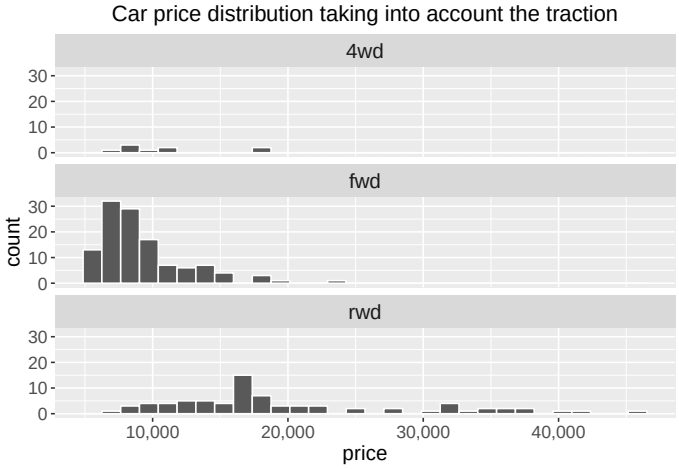
So, what information do we take from all this ?

1. Most of the cars are rwd;
2. On average, rwd based cars are more expensive than the other types and are distributed in all the categories ;
3. Most of the cars are gas powered, and there are just a few diesel cars
4. In one hand, cabriolet type cars and hardtops usually have 2 doors, and on the other hand, wagons and sedans usually have 4 doors (Mostly due to be resorted has family cars and are more practical).

Moving on to our response variable (price), it will be of interest to see how price is distributed across our dataset.

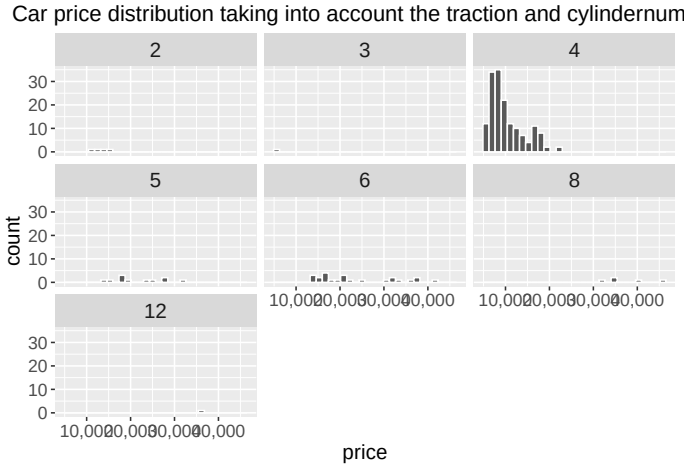


From this plot, we can observe that most of our cars from the dataset are priced below 20,000$ Let´s add the drivewheel variable and see what happens next...



From this addition, now we can see, again, that rwd cars are all over the price range and fwd have the cheapest cars from our dataset.

Now we add the cylindernumber variable to this exact plot and we see...



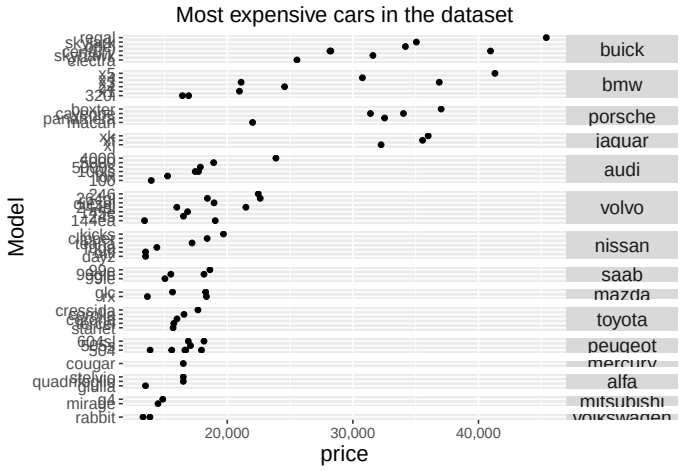
That most of our cars have 4 cylinders, and are also the cheapest cars in the dataset.

Until now we analysed most of our numerical type data but, what about the categorical data? Let's switch to categorical columns and let´s check how distributed our car brands are across drivewheel and doornumber...

```
## # A tibble: 22 x 3
##    Brand    mean    sd
##    <chr>   <dbl> <dbl>
##  1 jaguar  34600  2048
##  2 buick   33647  6790
##  3 porsche 31400  5654
##  4 bmw     26119  9264
##  5 volvo   18063  3315
##  6 audi    17859  3152
##  7 mercury 16503    NA
##  8 alfa    15498  1735
##  9 peugeot 15489  2247
## 10 saab    15223  2861
## # ... with 12 more rows
```

It´s clear that Jaguar has the most expensive cars on average, although it´s safe to say that BMW has such a high variance in prices. That means there is a BMW in all price ranges. Quick mention to Mercury that appears to have a NA value! That´s because there is only one car with the brand Mercury.

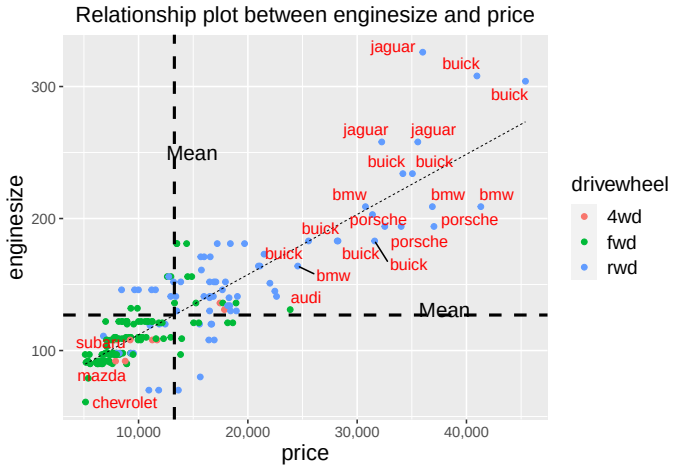
Let´s see in detail how these cars are distributed...



With this plot we can see that Porsche, Jaguar and Buick are considered as luxury cars. Also it´s noticeable that the variance of prices that we came to conclusion on the last tibble, is again true.

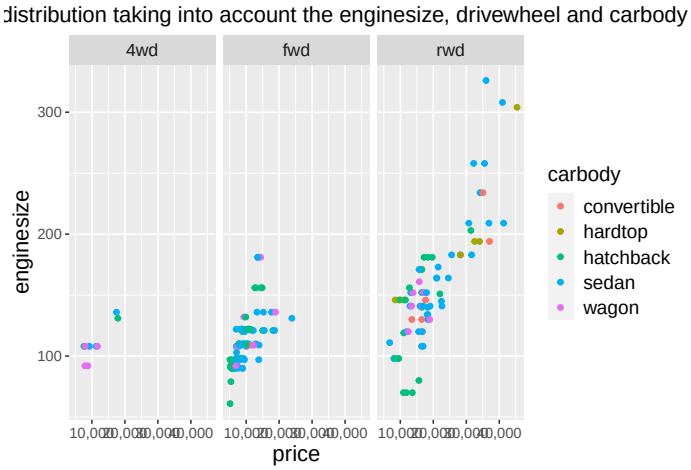
Next up, just like in the corrplot that i made before, we have seen that the enginesize variable is also highly correlated to the price variable. Let´s plot it ...

```
## `geom_smooth()` using formula = 'y ~ x'
```



The plot represents the relationship between the price and enginesize variables. It´s clear that the 3rd quadrant is occupied mostly with fwd cars that represent the cheapest cars and the smallest engines. About the rwd cars they are all over the 4 quadrants, which means that rwd cars can be cheap or expensive and even have the small or big engines. One last note regarding this relationship, it´s clear that this is a linear relationship, which means the bigger the engine the more expensive will be, and vice-versa.

Just one more plot to see what happens with the addition of carbody too...



Now we can see that the cheapest car is a hatchback and the most expensive one is a hardtop.

Now that we studied the relationship between variables, let´s get to the MLM (Machine Learning Models) implementation to predict the price of a car.

5. Machine Learning Models

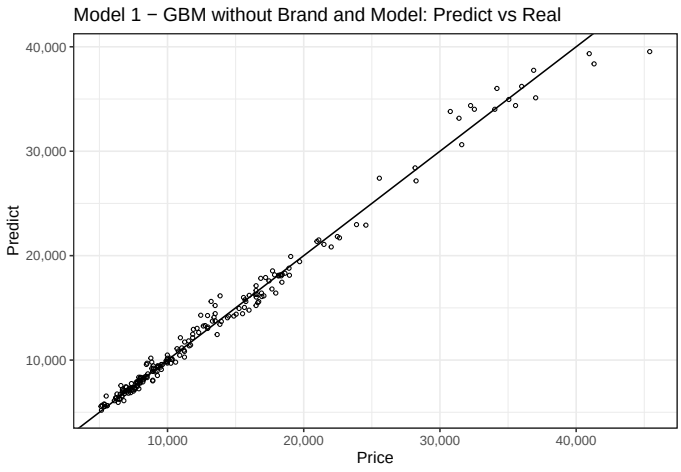
In order to implement MLM´s, i used the H2O platform, details about the platform later on the report. Since we are facing a regression type problem it´s better to implement a supervised type of machine learning model, so i tested two of them called Gradient Boosting Machine (or GBM) and Distributed Random Forest (or DRF).

With these i intend to predict a price of a car, by choosing the most accurate model between the two above.

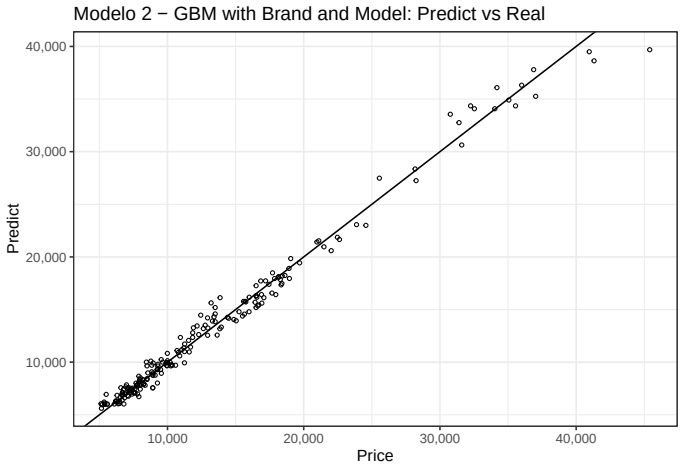
Quick note: For the sake of getting the best of the MLM´s iv´e made two of each model. What differentiates them is one of them has all the variables (p2 and p4) and the other does not have the Brand nor Model (p1 and p3). The reason behind this is to prove that the MLM will be "bias" towards the Brand or Model variable because of its significance towards this prediction. The main point of this prediction is not only to get the best accuracy of the prediction but as well get a trustworthy result at the end.

Let´s start of with the MLM GBM...

```
p1 <- read.csv("GBM_SM.csv")
difp1 <- abs(p1$price - p1$predict)
mdp1 <- mean(difp1)
```



```
p2 <- read.csv("GBM_M.csv")
difp2 <- abs(p2$price - p2$predict)
mdp2 <- mean(difp2)
```

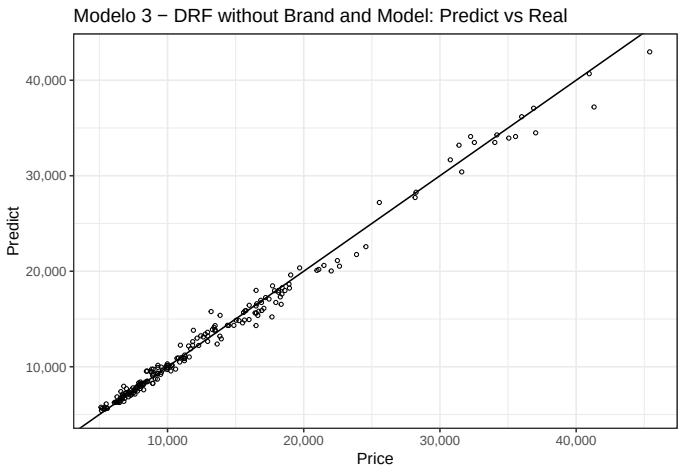


To explain the prediction process, The p1 and p2 variables is the prediction made by the h20 localhost api, the config for this result is basically 5 nfolds and 3 trees per column. After the prediction, i downloaded the prediction and called it p1 and p2. Now for the fun part, in R i created a variable called difp1 which is the price in the dataset minus the prediction made, and then i calculated the mean of all those results to determine the error margin in price of the MLM.

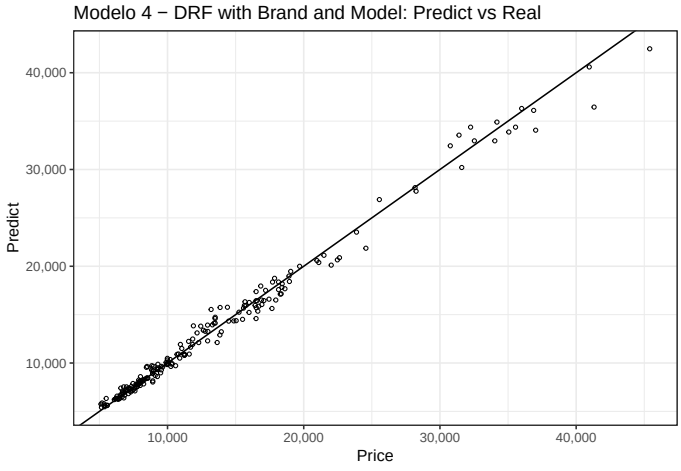
With the results, i came to the conclusion that the best model out of these two is the one without the Brand and Model variables (p1), because the p1 got a median error of 547.82 dollars and the p2 got a 646.07 dollars. So overall the p1 is the best model for this cause.

Let´s go and test the DRF MLM now...

```
p3 <- read.csv("DRF_SM.csv")
difp3 <- abs(p3$price - p3$predict)
mdp3 <- mean(difp3)
```




```
p4 <- read.csv("DRF_M.csv")
difp4 <- abs(p4$price - p4$predict)
mdp4 <- mean(difp4)
```



Going in depth in the drf´s, the conclusion is that the model´s are more precise without the Brand and Model included, by looking at the mdp3 and mdp4, averaging a mean error of 550.20$ and 574.59$.

6. Conclusion

I came to the conclusion that Geely Auto should use the GBM Machine Learning Model but the one without the Brand and Model variables aka p1 because it passed the success criteria and i believe its a ready for production MLM.

For the future, to improve even more the MLM, i believe by adding a Year of Production of the cars column would be a great addition overall as well a Mileage of the cars.