Car Manufacturer Suggested Retail Price Prediction Model

Capstone 2 Project

Lanxi Liu

1. Introduction

When it comes to car purchasing, people tend to have a picture in mind about what kind of car they are looking for. A big family would like to get a bigger car with more room like the SUV. People who commute a lot would prefer a car with higher MPG (miles per gallon) or an electric car. People who enjoy high performance and do not have a specific limit on budget would like to shop for luxury brands. However, not everyone is a fan of cars. Like myself, I don't know a whole a lot about the car market. I could not recognize all brands of cars. When I shop a car, I would like to know what kind of car I can afford and what kind features it could offer. Because the customers have needs and budgets, it would be beneficial if they can know the price range in advance.

On the other hand, the car manufacturer and car dealer need to be very sensitive about the market. When the manufacturer launches a new car, the MSRP (manufacturer suggested retail price) is crucial, which decides where this model sits in the market and who the competitors are. With the help of the prediction model, the manufacturer can set the retail price more accurately. Dealers control the final selling price. Setting a reasonable price would definitely increase the chance of selling a car.

2. Data Source

The data used for the prediction model is a Kaggle data set scraped from Edmunds and Twitter. See the following for the links:

- https://www.kaggle.com/CooperUnion/cardataset
- https://www.edmunds.com/

3. Data Wrangling

The data has total 11914 entries. Each row is corresponding to a single car model with details about the brand, model name, year, engine fuel type, engine horse power, number of engine cylinders, transmission type, driven wheels, number of doors, market category, vehicle size, vehicle style, highway MPG, city MPG, and popularity. Overall, the dataset doesn't have too much missing values.

Most of the missing values were located at the "Market Category" column. There isn't a clear pattern of why the information is missing after comparing the statistical summary of the data with and without market category. The data might be lost at the parsing process or that information is missing in the original source.

The second-most missing values occur at the "Engine HP" column. It is known that electric cars do not have engine, therefore, there is no information about the engine horsepower.

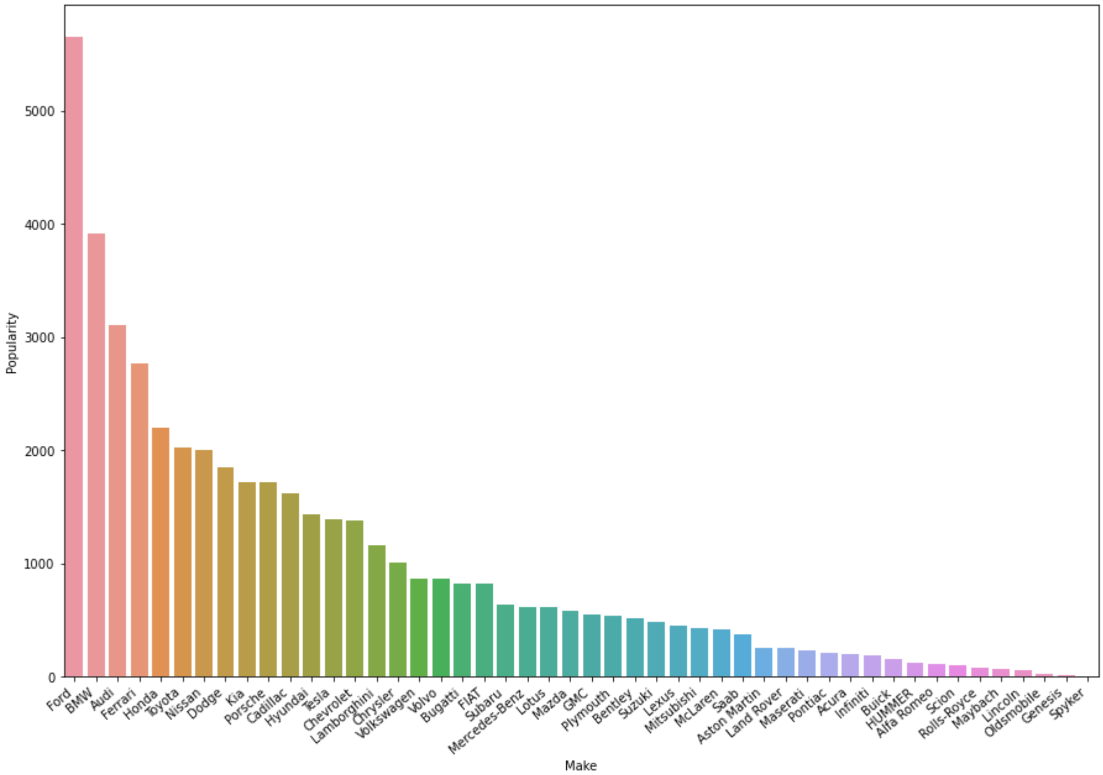
For all other missing values, I was able to fill them up with the information found on the internet. After dropping duplicate rows, there are 11914 entries left.

It was found that there is an outlier for highway MPG. An entry for Audi A6 2017 model was incorrect. The highway MPG was recorded as 354 mpg instead of 34 mpg as what it actually is.

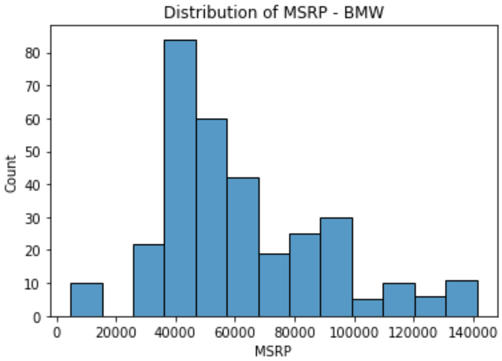
More detailed clean process could be found in the IPython notebook.

4. Exploratory Data Analysis (EDA)

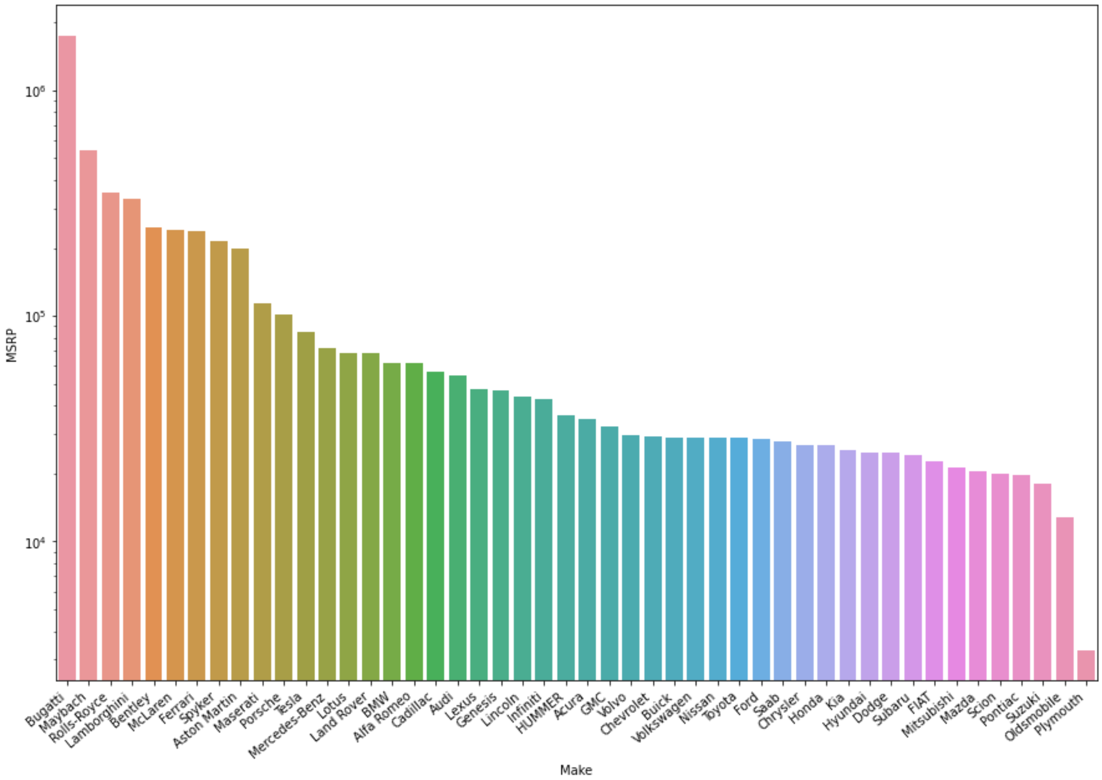
There are many categorical data in the dataset. In order to study the relationship between them and the retail price. The data was grouped by the categories. Then the average of the retail price was computed for comparison.



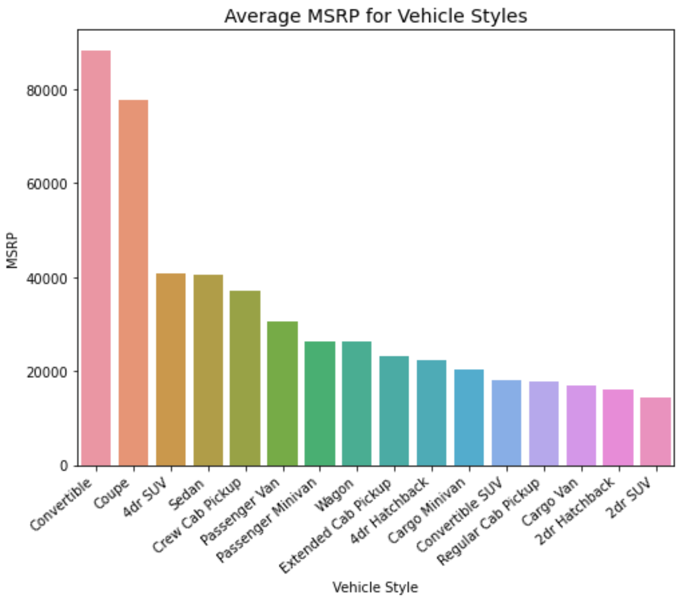
There are 48 car brands. Ford appears to be the brand with highest popularity, following by BMW and Audi. The popularity variable is directly associated with the brand. However, each brand typically has several models that cover a big range of prices. Such as the BMW brand showing below.



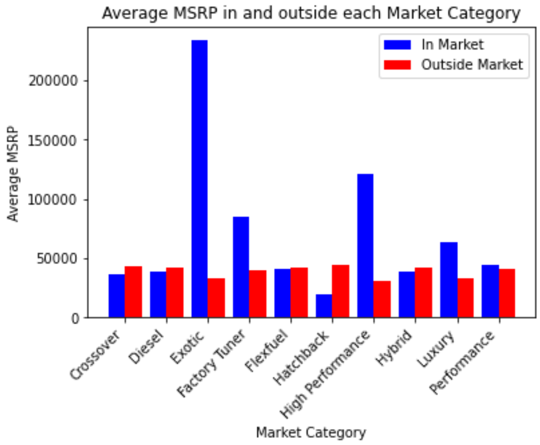
If plotting the average car retail price for each brand and rank it from highest to lowest, it shows that some brands were more focusing on luxury performance cars, which typically sold for more than 2x10^5 dollars. The y axis was taken log scale, since the distribution of the car price is highly skewed.



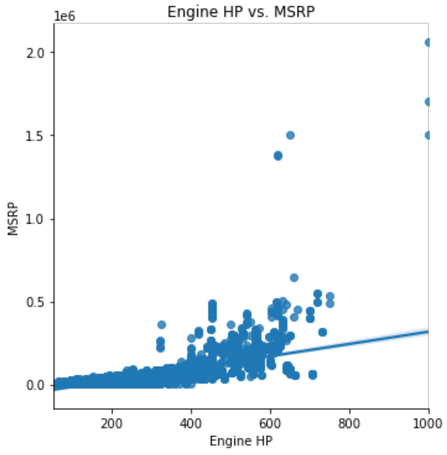
Looking at the vehicle style categories, it seems like some styles lead to a higher price. The top two styles are “Convertible” and “Coupe”. These all have very fancy looking and tend to show up on luxury models.

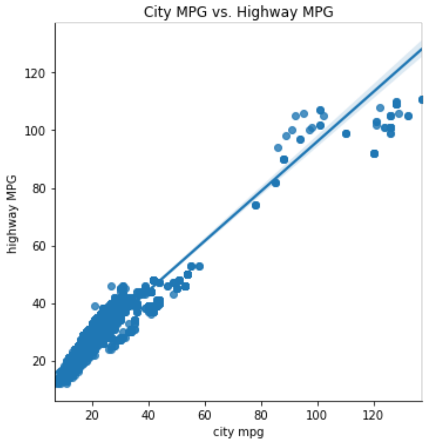


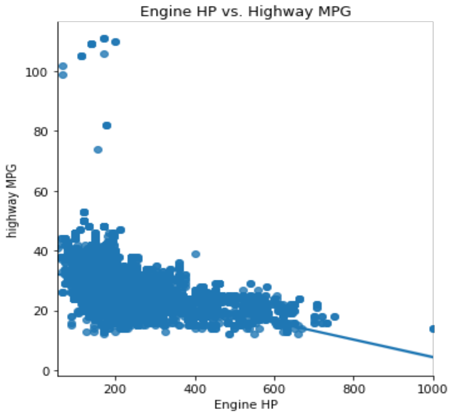
The market categories also have some relationship with the car retail price. From the following chart, it is obvious that the "Exotic", "High Performance", "Factory Tuner" and "Luxury" are the top four market categories that most affecting the car price. Whoever falls into those market categories tend to have higher price than those cars outside those market categories.



Moving on to the numerical features, there are some correlation relationships found within the dataset. The engine horsepower is positively correlated with MSRP. The highway MPG is positively correlated with city MPG but negatively correlated with Engine horsepower.







For all other categorical or numerical features, they might be still related to the car price. However, the relationship is more subtle and required the machine learning model to discover.

More details about the procedure of exploratory data analysis could be found in the IPython notebook.

5. Modeling

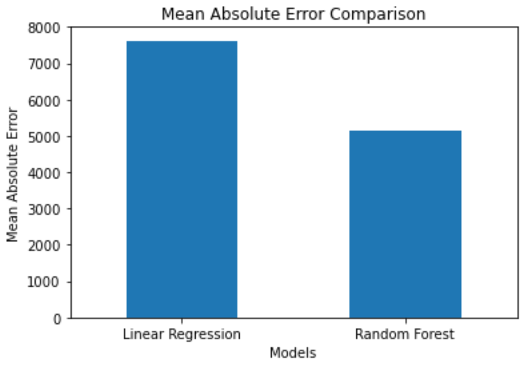
While MSRP is the target variable of the dataset, all other features were used to train the machine learning models. The models are trained using 70% of the data and remaining 30% is used to evaluate the performance of the model. Since we are predicting price based on historical data, we need supervised machine learning model such as linear regression model and random forest regression model.

After cleaning the data, there are only a few missing data for the "Engine horsepower". In order to perform the fitting on the model, those missing data were filled by the median. For some algorism, it is necessary to scale the values for all features. The "pipeline" class in the scikit learn library is a very useful tool, which combines the modeling steps to just one pipeline and makes it easier for tuning the hyperparameters. The imputer step and scaling step were added to the pipeline. 5-fold cross validation was performed with grid search method in scikit learn for both models. The cross validation help me to know the average performance of the model and the range of variability.

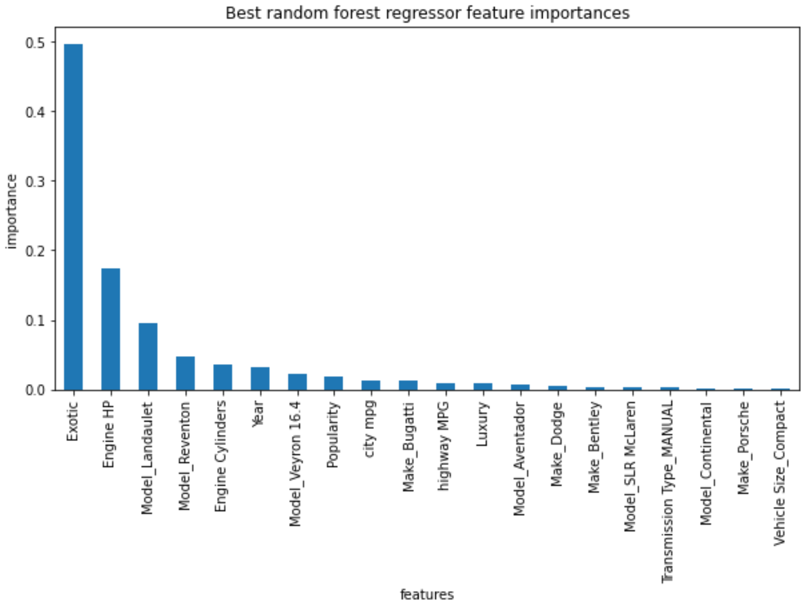
For the linear regression model, a feature selection function was also added to the pipeline to prevent overfitting the training data. Here I used "SelectKBest" function using f_regression as the scoring method.

For the random forest model, a limit on the depth of the tree was defined to prevent overfitting. The other parameters assigned to the models is the number of estimators.

For both models, I used the same evaluation metrics. First is the accuracy. Since this project is about a price prediction model, the more accurate the better. One other metric is the R^2 score (the coefficient of determination). It measures the proportion of variance in the predicted car prices. Last metric is the Mean Absolute Error, which is the most intuitive metric. The error represents the how much off the predicted value is from the truth.

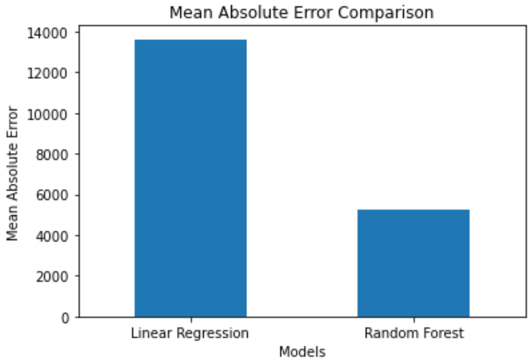


After tuning all parameters and fitting the training dataset, it was obvious that random forest regression outperforms the linear regression model. It not only has lower mean absolute error, also higher R^2 score and higher accuracy.



The feature that affects the performance the most is the market category -exotic. If you look back to the EDA process, it already shows that cars in this particular market tends to have very high price. The second dominating feature is engine horsepower. We already knew that price is highly correlated with the engine horsepower. No doubt that it is a very important feature to the model. One problem I noticed is that the prediction model considers the brand and car model as import features. However, it might be more useful if the model could accurately predict the price without knowing the brand or model name beforehand. Then, people who are looking for a specific combination of features can get an idea how much the car worth and then compare to the price in the market. So, I run both linear regression model and random forest model again.

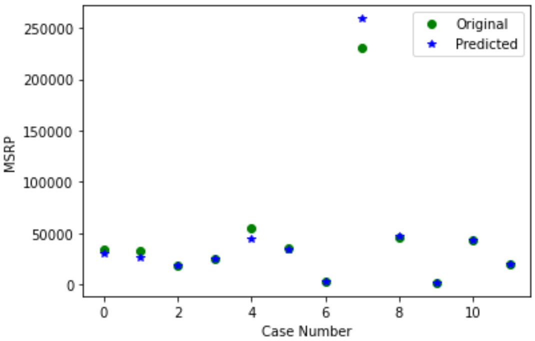
It was interesting that the mean absolute error went down a little bit for the random forest model. However, the error for linear regression almost doubled. Therefore, random forest model is still the best choice no matter including the brand and model information or not. The good news is that the best parameters for the recommendation systems after hyper tuning are the same. It means I can adopt one model to predict both cases.



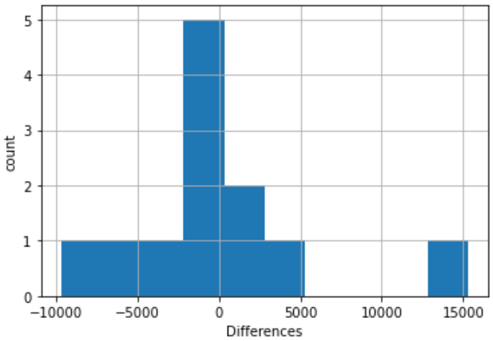
More details about the modeling procedure could be found in [this IPython notebook.]

6. Prediction

To use the best model, one could choose to include specific brand and model or not. Here I used two recent car models, one from Audi and one from Toyota. Noted that the original dataset only has data up to 2017. The gap in between might cause some error to the final prediction. This could be solved if we can feed model with some newer data. Other than those two models I added. I also randomly chose 10 entries from the original dataset to see the individual performance.

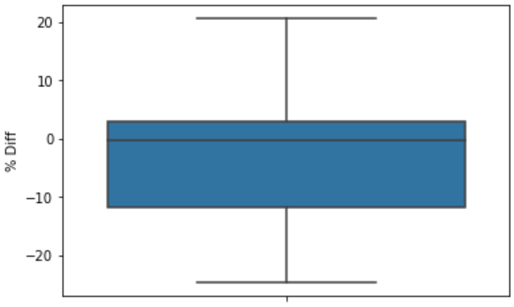


Price Comparison with Brands and Models

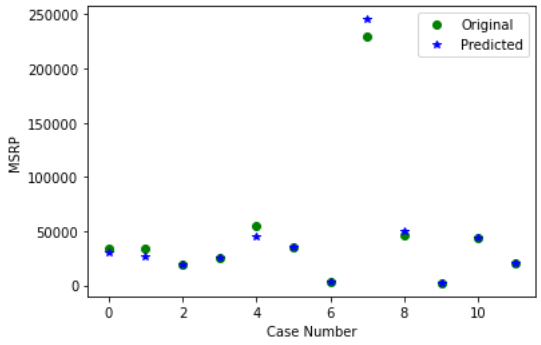


Price in Difference

The last two cases on the scatter plot above represent those two recent car models I added for prediction. The model performs fairly good on these two cases. As you can see, the error for the other ten cases randomly picked from the original dataset various a lot. Since the car price varies a lot, it makes more sense to use percentage difference as the error metric. For a car worth more than 50,000, a few thousand differences would not be too worse.

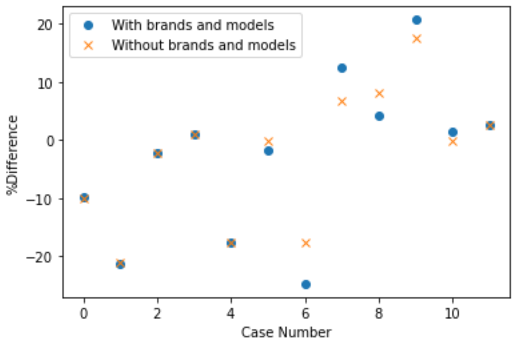


As shown above, the percentage difference varies from -24.7% to 21.2%. 75% of the cases has percentage difference within 12%. It doesn’t look too bad since my initial target is to have 80% accuracy for the prediction model.



Price Comparison without Brands and Models

After removing the information about brands and models, the prediction became more accurate. It might imply that the brand adds value to the car. There might be some other factors affecting the price that are not included in our model such as material, quality, brand reputation, and customer services.



More information about the prediction procedure could be found in the IPython notebook.

7. Conclusions

The model is decent enough to predict the car price just looking at the accuracy of the random forest model. However, the price difference is still an issue because it varies a lot. Although having prediction error is acceptable, costumers might not be satisfied when it comes to the situation with limited budget.

There are still rooms to improve this model.

1. We could use more recent data to train the model. Car price for very old cars do not contribute a lot to the future cars. It won’t be reasonable to have a 2000 dollars new car on 2021.
2. It looks like one vehicle model could have a price range because the features vary within the model itself. However, the data didn’t reflect the difference very well. We may consider including more features to target on those differences.
3. The model can only predict when all features are filled up right now. It might be useful if the model could run with data with missing values and give a range to the users instead of a exact price.