Prediction of Car’s MPG using Regression

Ruize Ma

Data Science Institute, Brown University
https://github.com/RuizeMa666/Final_project_car-s_MPG.git

1. Introduction

The climate change problem is a serious problem that many people are worried about. How to reduce fossil fuel consumption and greenhouse gas (GHG) emissions has been a critical question for both researchers and policymakers. Light-duty cars account for almost 60% of the greenhouse gas emissions produced by the transportation sector, which has grown to be one of the main producers in the country (EPA, 2017). Upon reflection, it prompts consideration that the development of a predictive model for automobiles' fuel efficiency could potentially mitigate emissions. Such a model would enable manufacturers to refine their designs based on predictions rather than resorting to the construction of numerous physical prototypes for testing. This streamlined approach holds the potential to significantly reduce resource consumption and the associated environmental impact linked to prototype production, in addition to the substantial savings in both time and financial resources. Motivated by these considerations, I embarked on the endeavor of constructing predictive models for automobiles' Miles Per Gallon (MPG) using regression.

For this purpose, I selected a dataset from Kaggle known as "AUTO MPG." Originating from the CMU Statlib Library and curated by John Ross Quinlan, this dataset encompasses information on 405 cars. The processed dataset includes variables such as MPG, which quantifies the efficiency of a transporting vehicle in terms of energy production (Dua&Graff, 2019); the number of cylinders; displacement, denoting the distance covered by the pistons measured in cubic inches; horsepower, a metric for calculating the rate at which force is generated by a vehicle's engine; weight, measured in pounds; acceleration, expressed in miles per hour per second; and origin, categorized as 1, 2, or 3, signifying USA-origin, Europe-origin, and Asia/Elsewhere-origin, respectively.

The dataset exhibits six missing data points in horsepower and an additional six in MPG. As MPG serves as our target variable, the six points with missing values in this variable are excluded, while the missing points in horsepower are retained for subsequent processing.

2. EDA:

In our preliminary analysis, we examined the correlation matrix encompassing all numerical features. Features exhibiting a correlation coefficient exceeding 0.9 were identified as highly correlated. As illustrated in Fig 1, the correlation analysis revealed a notable correlation between displacement and both weight and horsepower. Consequently, a decision was made to exclude the displacement column from further consideration.

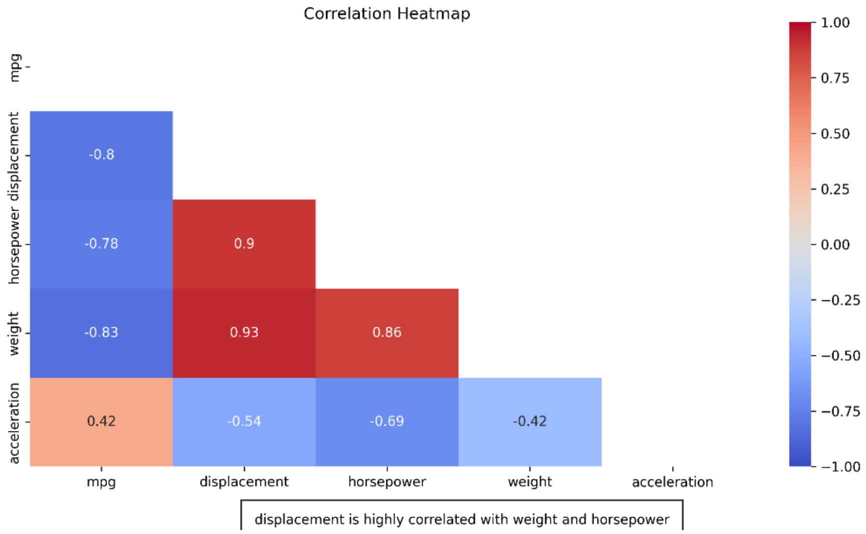
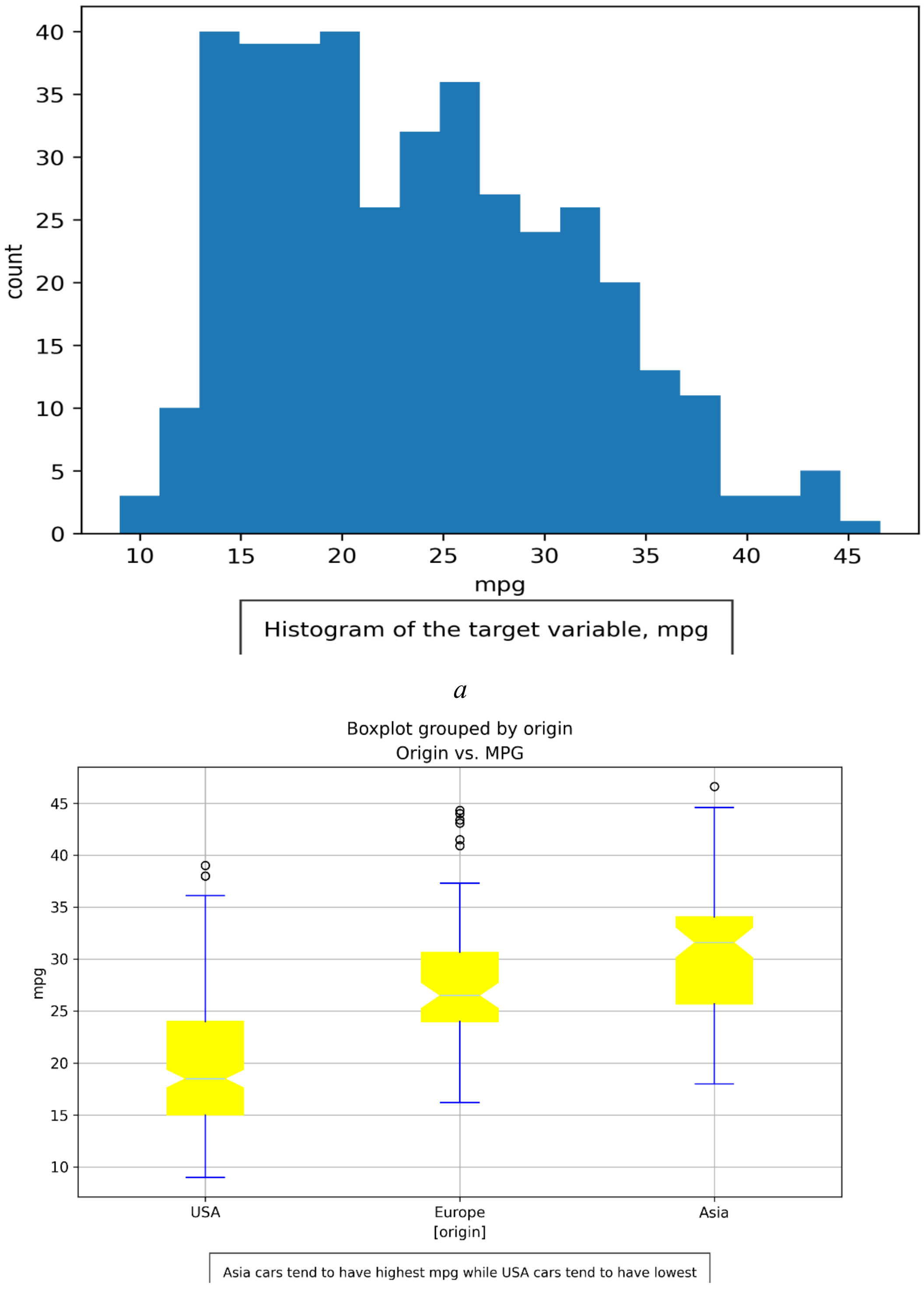


Figure 1. Correlation matrix of all numerical features

Following the initial analysis, we proceeded to generate several visualizations to gain a comprehensive understanding of the dataset. Fig 2a presents an overview of cars' miles per gallon (mpg), our designated target variable. Notably, the majority of data points fall within the range of 13 to 40 mpg. Recognizing the variations in fuel standards across different countries due to distinct fuel prices, cultural preferences, and living habits, our interest extended to exploring the fuel efficiency disparities among cars originating from different regions.

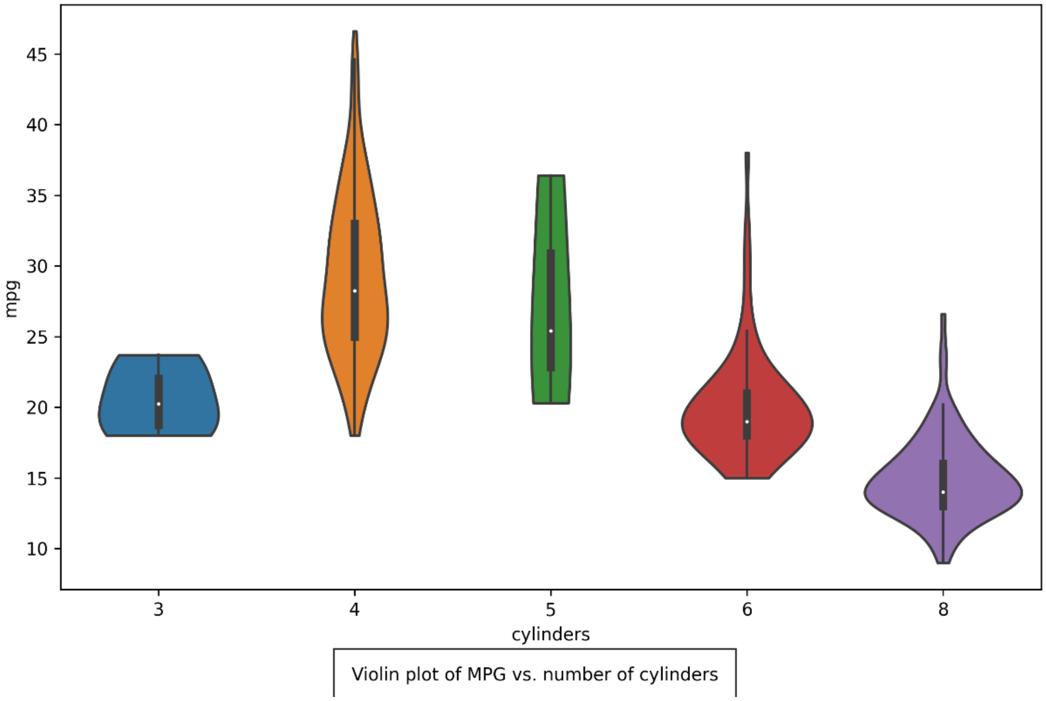
Figure 2b highlights such disparities, revealing that cars of Asian/Elsewhere origin tend to exhibit superior fuel efficiency, whereas those from the USA tend to have higher fuel consumption. Additionally, Figure 2c illustrates the relationship between cars' mpg and the number of cylinders in their engines. Contrary to conventional expectations, the figure indicates that, as the number of cylinders increases from 3 to 4, fuel efficiency improves; however, further increases in the number of cylinders lead to a decline in mpg.

In summary, armed with a comprehensive overview of the dataset, we have chosen to construct predictive models for cars' miles per gallon (MPG), utilizing the features of horsepower, weight, acceleration, origin, and cylinders. It is essential to note that in the preceding analysis, we identified six missing data points in the horsepower feature. In the subsequent sections, we will undertake feature processing and address the handling of missing values.


count
40
35
30
25
20
15
10
5
0
10
15
20
25
30
35
40
45
mpg
Histogram of the target variable, mpg
Boxplot grouped by origin
Origin vs. MPG
mpg
45
40
35
30
25
20
15
10
USA
Europe
Asia
[origin]
Asia cars tend to have highest mpg while USA cars tend to have lowest

a

b



c

Figure 2. EDA plots to get an overview of data.

3. Methods

3.1. Preprocess

Usually, we split the data before preprocessing it to prevent the problem of data leakage. However, since this dataset is independent and identically distributed (The information of one car will not influence the information of others), this problem can be avoided, wedecided to preprocess the data before we split the data. For the missing points in horsepower, since the utilization of sklearn’s SimpleImputer will affect the features’ variance badly, we decided to use sklearn's IterativeImputer using linear regression to predict the missing value. To make a successful prediction for the missing value, we need to preprocess the data before using the imputer. During preprocessing, we used StandardScaler for numerical features: horsepower, weight, and acceleration; used OneHotEncoder for categorical feature - origin; used OrdinalEncoder for ordinal feature – cylinders. After preprocessing and imputing, we have finally got 7 features to predict our target variable: num_horsepower, num_weight, num_acceleration, cat_origin_1, cat_origin_2, cat_origin_3, ord_cylinders, and there are no missing values anymore.

3.2. Splitting

Since my dataset is i.i.d and it is not a large dataset, I would like to have as much data as possible for training. Thus, the splitting method I chose is K-Fold Cross-Validation, which is a method for i.i.d data and can increase the data points trained while giving unbiased estimates and avoiding the problem of overfitting.

3.3. Pipeline

For the machine learning pipeline, I have built a function called MLPipe_KFold_RMSE, it used GridSearchCV to do the splitting method mentioned before; give the best model based on the inputted machine learning algorithm by tuning the hyperparameters given. Since this is a regression problem, I chose neg_mean_squared_error as the scoring of GridSearchCV, it will choose the parameters that gave the least MSE as its best model. I have looped through 10 random states in the function to decrease the effect of uncertainty caused by splitting in this function. Thus, it will return the 10 best models and the corresponding 10 test scores. The evaluation metric I chose is RMSE. I will use this function to go through 7 machine-learning algorithms and compare the results based on the RMSE score. To avoid the problem caused by the non-deterministic ML algorithms, I have fixed a random seed of 10 while tuning the parameters.

3.4. ML algorithms

The first 6 algorithms used the features that have missing value imputed. I have chosen 3 linear models: The first two are Lasso and Ridge. I tuned the value of alpha (0.0001, 0.001, 0.01, 0.1, 1, 10, 100). The third is ElasticNet and I tuned the value of alpha (0.001, 0.01, 0.1, 1, 10, 100) and l1_ratio (0.2, 0.5, 0.8). I have also trained the Random Forest model tuning the value of max_depth (None, 1, 3, 10, 20) and max_features (None, 1, 3, 10, 20); SVR (Support Vector Regressor) tuning the value of gamma(1e-3, 1e-1, 1e1, 1e3, 1e5) and C(0.1, 1, 10); and KNeighborsRegressor tuning n_neighbors (1, 10, 30, 50, 70, 100) and weights ("uniform", "distance").

Besides these 6 models, I have also chosen XGB Regressor for this data. Since the XGB regressor can deal with features with missing values, I just transformed the data instead of handling the missing values. For this model, I tuned n_estimators (100, 200, 300), learning_rate (0.05, 0.1, 0.15), and max_depth (3, 4, 5).

4. Result

4.1. Overview of results

Following the acquisition of 10 test scores for each model, I computed the mean, standard deviation, and minimum of the test scores for each model. The baseline test scores, derived from the Root Mean Squared Error (RMSE) calculated based on the test sets and the mean value of the test sets, exhibit a mean of 7.8572 and a standard deviation of 0.421.

Upon analysis of Fig 3 and Table 1, it is evident that all seven models surpass the baseline model in terms of performance. Notably, the three linear models demonstrate comparable performance. Furthermore, the random forest model distinguishes itself with

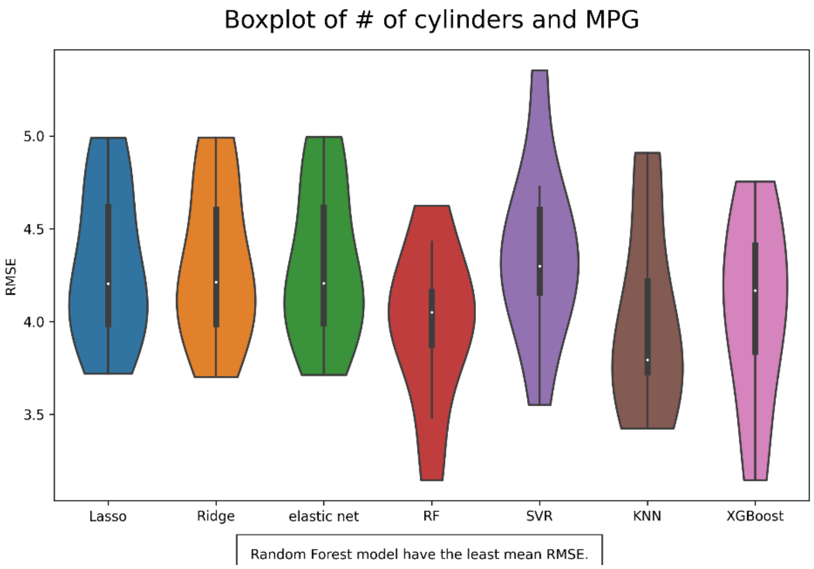
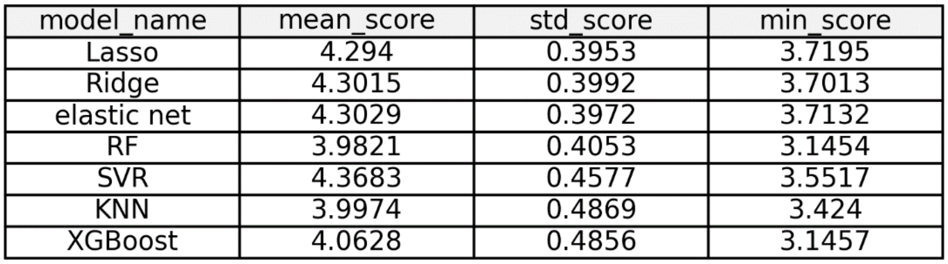


Figure 3. Plot of the structure of the seven models



model_name	mean_score	std_score	min_score
Lasso	4.294	0.3953	3.7195
Ridge	4.3015	0.3992	3.7013
elastic net	4.3029	0.3972	3.7132
RF	3.9821	0.4053	3.1454
SVR	4.3683	0.4577	3.5517
KNN	3.9974	0.4869	3.424
XGBoost	4.0628	0.4856	3.1457

Table 1. Summary of the performance of the seven models

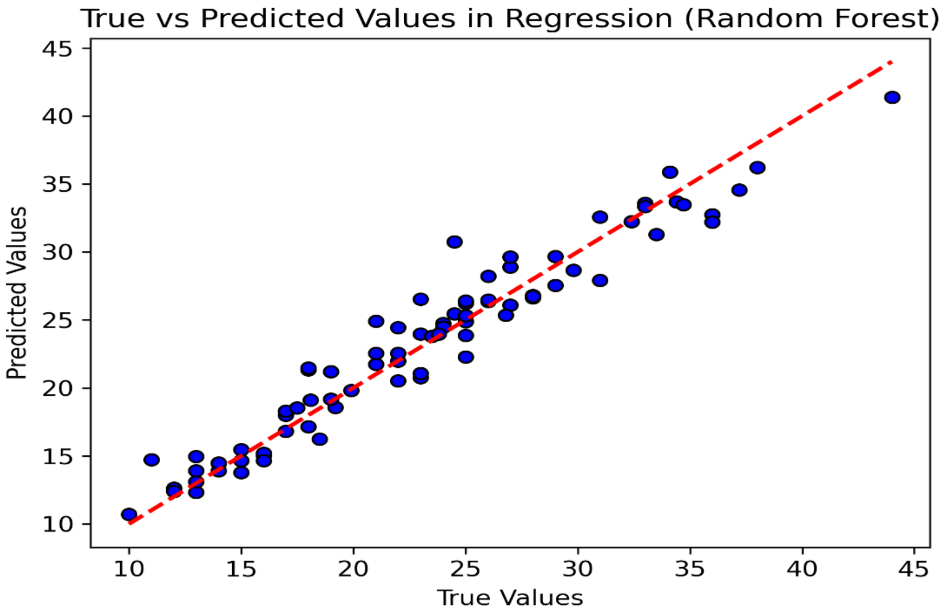


Figure 4. Scatterplot of True value and Predicted value(by random forest)

the lowest mean test score of 3.9821, exhibiting a performance that is 9 standard deviations superior to the mean baseline test score. Given its acceptable standard deviation of 0.4053, we have chosen the random forest model as the most optimal.

Inspection of the plot in Fig 4 underscores the excellent fit of the random forest model to the data. The chosen model yields an RMSE of 3.4823 and an R^2 of 0.7745.

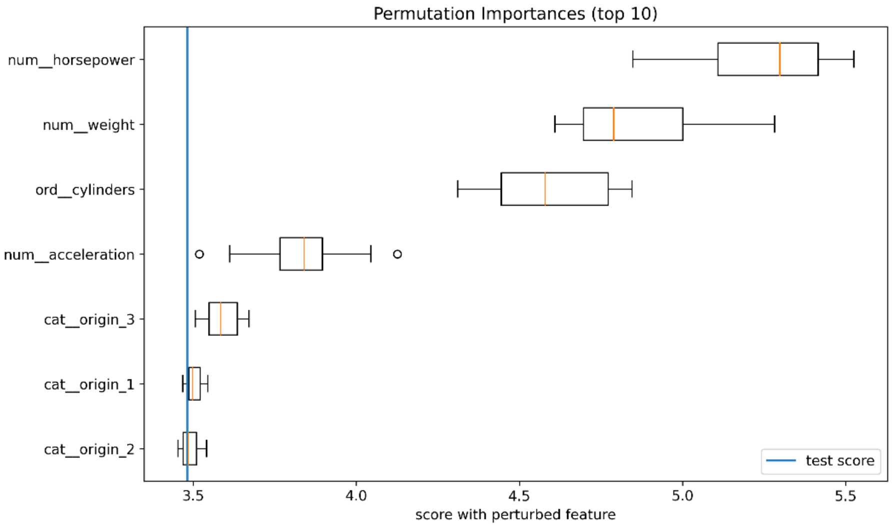
4.2. Interpretability

Our initial focus centers on the global feature importance of the model, employing three distinct methods for comprehensive insight. The first method entails permutation feature importance (depicted in Fig 5a), involving the random shuffling of a single feature in the test set. Subsequently, the RMSE is recalculated with the shuffled data, allowing for a comparison with the original RMSE. A larger difference signifies a more substantial importance for the feature.

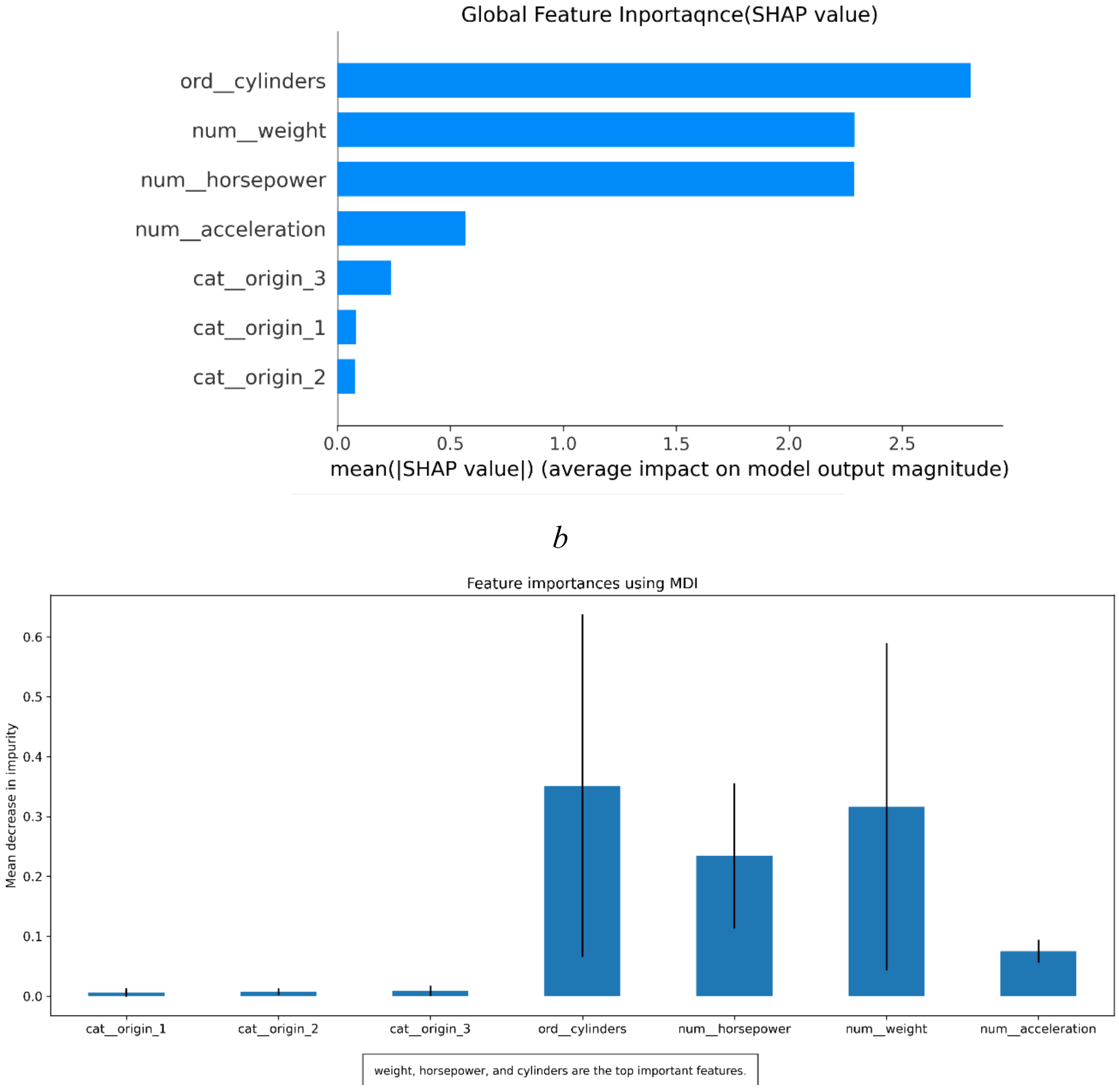
The second method involves the calculation of the mean absolute SHAP value across the entire test set, as illustrated in Fig 5b. Features with higher values in this context are deemed more influential.

Additionally, the third method involves an intrinsic capability within the random forest model known as Mean Decrease Impurity (depicted in Fig 5c). This method computes the Mean Squared Error (MSE) reduction at each decision node within a tree when data is split based on a particular feature. The reduction is weighted by the number of data points reaching the node, assigning greater importance to nodes with a higher volume of data. The blue bars represent the mean reduction, while the black vertical lines denote the confidence interval or the error bar for each feature.

Upon scrutiny of the three plots, a unanimous consensus emerges, identifying ord_cylinders, num_horsepower, and num_weight as the three most pivotal features in our model.



a



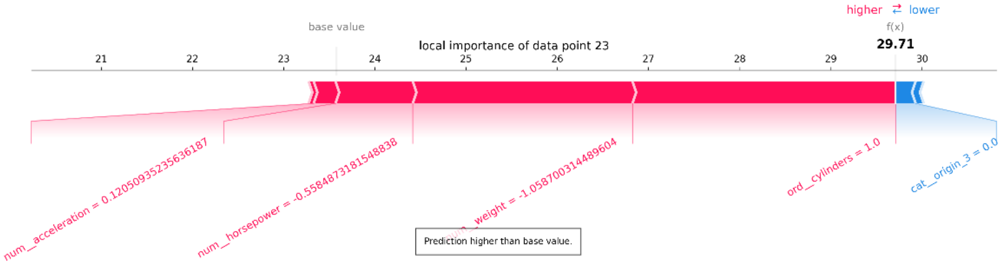
b

weight, horsepower, and cylinders are the top important features.

c

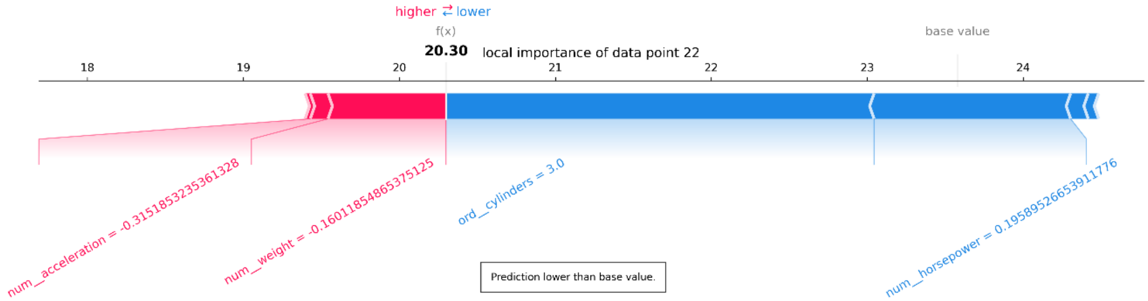
Figure 5. Plots for Global Feature Importance

For the local importance, we calculated the shap value two data points (Fig 6a and Fig 6b). We can see both points have reached their predictions because of the contribution mainly made by ord_cylinders, num_horsepower, and num_weight, too.



Prediction higher than base value.

a



b

Figure 6. Plots for Local Feature Importance

5. Outlook

To make more improvement, some more advanced approach like Locally Interpretable Model-agnostic Explanations may be able to further improve the model's interpretability. One weakness of my model is that the data is too small. I have created a brand column for this data but found that there are more than 50 brands in this data containing 406 data points. Thus, the brands can only provide little predict power. These problems can be resolved by collecting more data points. Moreover, using more complex algorithm such as algorithms in deep learning may help us built more accurate model.

Reference:

1. Fast facts on transportation greenhouse gas emissions | US EPA. (n.d.). https://www.epa.gov/greenvehicles/fast-facts-transportation-greenhouse-gas-emissions
2. D. Dua and C. Graff, "UCI Machine Learning Repository", 2019, [online] Available: http://archive.ics.uci.edu/ml.
3. J. Ross Quinlan, "Combining instance-based and model-based learning", Proceedings of the tenth international conference on machine learning, pp. 236-243, 1993.