Project 1 Report

By Steven Oh

Preparing the Data

To prepare the data, a quick analysis of the data was completed using functions like `data.info()`, `data.describe()`, and `data.shape`. With these functions, it was found that there were a total of 398 data points with 9 categories. To make sure that there were no duplicates, the `data[data.duplicated()]` function was run (results were 0 duplicates).

After a quick analysis of the data's structure and content was completed, the true preparations began. The first step was to convert all data types that were potentially useful into types that the algorithm would be able to use. For this data, it only included the horsepower column as it was an object type. However, while trying to convert it into an integer type, it was discovered that there were missing values that were replaced with question marks. As 0 horsepower is not a possible value, it was used to replace the question marks, allowing for the conversion into integer type successful.

To properly replace the values to help with the algorithm, a bivariate analysis plot comparing cylinder values with horsepower was used (line 12). The graph generally showed that the horsepower for each cylinder type was somewhat grouped, leading to the use of the cylinder's average horsepower as the replacement value (line 15). Since the missing values were replaced with 0 from the beginning, data2 was created to more accurately calculate the mean values. This was achieved by replacing the 0 values with NaN values so that they wouldn't be included in the calculations.

In addition to correcting missing values, three columns were dropped as they were deemed non-important or not correlated to the important information (origin, accuracy, and model_year columns). Utilizing the heatmap, the important values like cylinder, displacement, horsepower, and weight were all correlated with each other, while the three columns that were dropped did not correlate with them. Even though the fuel efficiency was reversely correlated with the important values, they were deemed important when utilizing prior knowledge of vehicles and how they work (essentially, the values that deal with the engine should be important for fuel efficiency).

Finally, one-hot encoding was used for the cylinders column to complete the preparations for the data.

Data Insight

There was much insight that was gained from preparing the data. As mentioned in the previous section, it was found that the missing values were stated as question marks in this data set. This would not have been easy to figure out if it wasn't for the preparation of the data. Lastly, it was quite useful to use bivariate and univariate plots to analyze the data and use the analysis to make better decisions.

Training the Model

The procedure used to train the model was quite similar to what was done during class. The X coordinates were set to everything in the prepared data besides mpg and car_name. The y coordinates was set to mpg, which is a numerical representation of fuel efficiency. Using `sklearn.model_selection`, the `train_test_split` function was used to split the data with a test size of 30 percent.

Then, the linear regression model was used from the sklearn library, fitting the training X and y coordinates to the model. Lastly, the score was used to determine the accuracy of the training and test data.

Results

The accuracy of the training data was 72.844% while the accuracy of the test data was 76.640%. The model predicts the fuel efficiency by the method known as linear regression. Essentially, the model creates the assumption that the relationship between the values in the independent and dependent variables is linear. This means that the model will draw a best-fit line through all the data, and use that line to predict the fuel efficiency.

Summary

Overall, I would say that there is a good amount of confidence in the model. For the accuracy to be over 70% for both test and training values seems reliable, especially for such a dataset.