Predicting Fuel Efficiency (MPG): Machine Learning Linear Regression Model

Jakob Long
COE 379L

Software Design for Responsible Intelligent Systems

Introduction

The intention behind this project was to introduce exploratory analysis, and get our fingertips wet with different Python libraries, with these libraries being Seaborn, Matplotlib, Pandas, Numpy, & most importantly Scikit-Learn.

With the prior mentioned libraries, I was able to first tidy the data which was primarily done using Pandas, and Numpy. Additionally, with Seabron I construct visual representations of then data to aid in demonstrating correlations between the different variables within the data. Not only were visual representations constructed, but a machine model of linear regression to predict the fuel efficiency of the vehicles within the dataset. Two different models were constructed, with one removing the Car_Make variable that is present within the attached notebook. These models were trained & tested for validation and to compare to see which model had performed better. Which could potentially offer insight on which characteristics of cars have the largest effect on motor vehicle fuel efficiency.

Data Preparation

The dataset used for this project was an automobile dataset that had 398 observations, with 9 unique columns: fuel efficiency (mpg), the number of cylinders, engine displacement, horsepower, weight, acceleration, model year, origin, and the car name. The dataset held two float64 data types, four int data types, and three object data types. With null or bad values being held within the horsepower column that were corrected.

Thus, to prepare the data I took the following steps:

1. **Re-mapping Columns:** To begin I re-organized the origin (cars nation origin) column, to list the actual country the car was from. I then also kept only the first word within each cell of the car_name column due to that first word being the make of the car.

2. **Dropping Columns:** After re-mapping the prior mentioned variables, the original columns (car_name & origin) were dropped, as they were now considered duplicate variables.

3. **Type Conversion:** Horsepower was an object variable with null or bad "?' values that didn't allow for proper data manipulation. Thus, the nulls and bad values were replaced by the median value of Horsepower. Additionally, I converted the Country & Car_make variable to categorical variables to then be encoded later.

4. **Value Tidying:** The Car_Make column had a lot of duplicate names, but it was filled with typos. So with the use of str.replace(), I was able to correct these duplicates.

5. **One-Hot Encoding:** The categorical variables mentioned above (Car_Make & Country), were then One-Hot Encoded to Boolean variables to allow for easier manipulation & incorporate these variables into our machine model!

Linear Regression Model Fitting & Learning

To begin I used the Scikit learn library to develop this linear regression model. The objective of this model was to predict the fuel efficiency, using the other variables that have been tidied within our dataframe. The procedure is the following:

For Both Models:

1. **Data Splitting:**
   First the target variable "mpg", was isolated and put into variable Y, with the rest of the columns going into variable X. The data was then split using scikit learns 'train_test_split' function, with 70% of the data allocated to training & the remining 30% for testing.
2. **Model Training:**
   The training data was fed into the linear regression model within scikit-learns 'LinearRegression' function. The model was then fit to the training data, of which the model represented can be found in the conjoining notebook.

Model Analysis

To evaluate the models performance, the R^2 , or coefficient of determination was used. This was found using scikit-learns metrics. For the first model, the model with the car_make variables included we found the testing model R^2 was: 0.8225, while the training model R^2 was: 0.8401. Since an R^2 value of 1 indicates a perfect fit, we can justify that our first model performed quite well, especially since its greater than the universal guideline of being greater than 0.7, for indication that the model is performing well. Additionally, we can assume that the model would perform well outside of the training data, due to the testing model having a very similar R^2 value

to that of our training one. Thusly, our high R^2 value gives us high confidence that our first mode l can predict a vehicles fuel efficiency.

The same testing was done using another dataset, but not including the car_make variables, just to see the differences between the two models. Surprisingly, there wasn't much difference. The testing R^2 value was: 0.8433, while the training R^2 value was: 0.8141. These values are extremely close to the initial model, indicating that the make of a car make not be all that important for fuel efficiency. But similar to the first model, we have a high confidence in this model to predict a vehicles fuel efficiency as well.

Resources

1. Data: https://raw.githubusercontent.com/joestubbs/coe379Lsp24/master/datasets/unit01/project1.data
2. Linear Regression Help: https://www.freecodecamp.org/news/how-to-build-and-train-linear-and-logistic-regression-ml-models-in-python/
3. Visualization Aid: https://seaborn.pydata.org