Summary of Homework Assignment 2:

Jared Kelnhofer

2/12/2020

Data and goals:

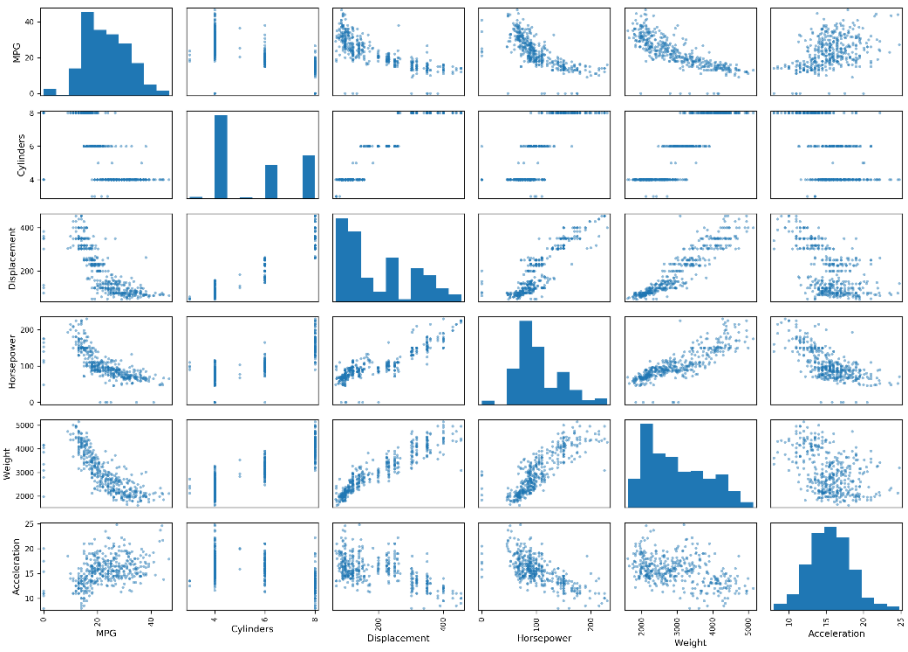
For this project, I chose a “Car information” dataset consisting of 400 instances that I found available online. It contains features such as car weight, car manufacturer, car acceleration, and car horsepower. I decided to attempt a prediction of car acceleration when given only a car weight.

Data preparation:

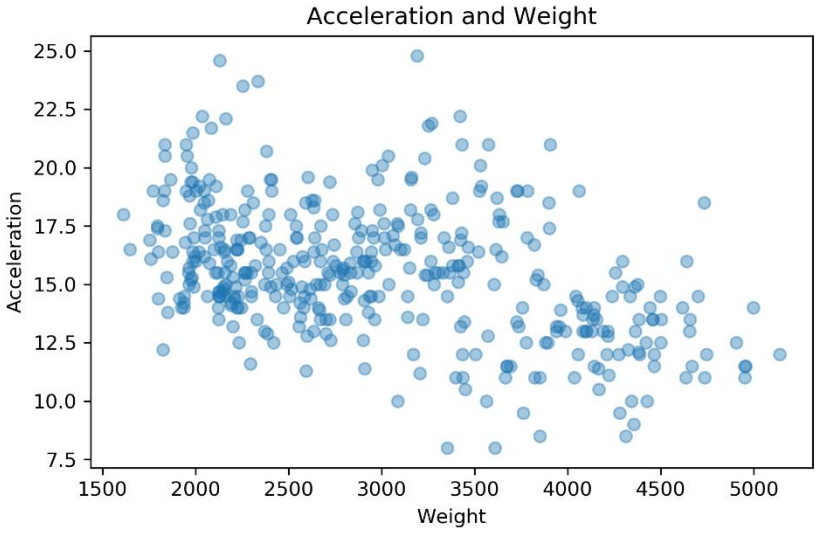
The row at index 1 of the dataframe was useless, so I needed to drop it. I wrote function to load the data and organized the project similarly to the structure of the chapter 2 California assignment. I also wrote a method to save figures to an “images” directory for future use. I also had to transform many columns from a simple object into either INT or FLOAT64 data types. I separated the data into testing and training sets.

Data exploration and visualization:

I ran a correlation algorithm on the data to see which features were correlated with acceleration. I ended up choosing a feature that was less obviously correlated (weight), so that I didn’t have an excuse to go with a simple linear model. I created a histogram of different correlations:

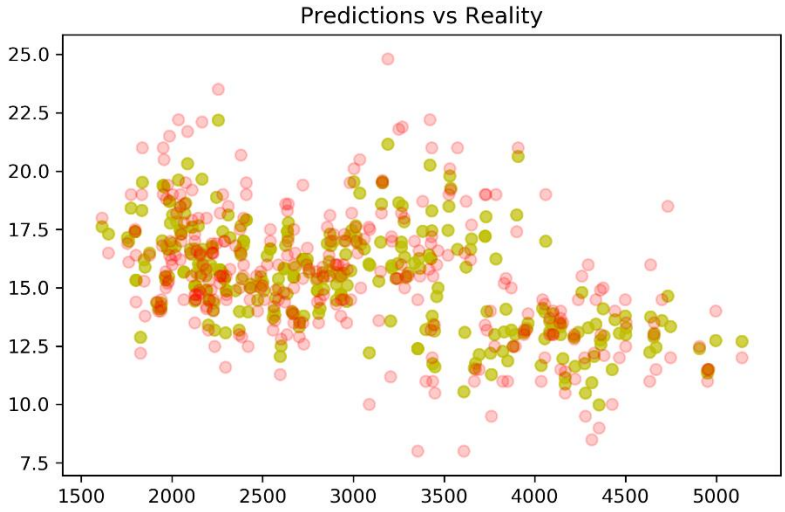


I also noticed that there did seem to be a somewhat linear relationship between weight and acceleration, but with a bit more complexity. I didn't notice any data "lines" that had to be removed, so I kept the data as I found it.



Model selection and training:

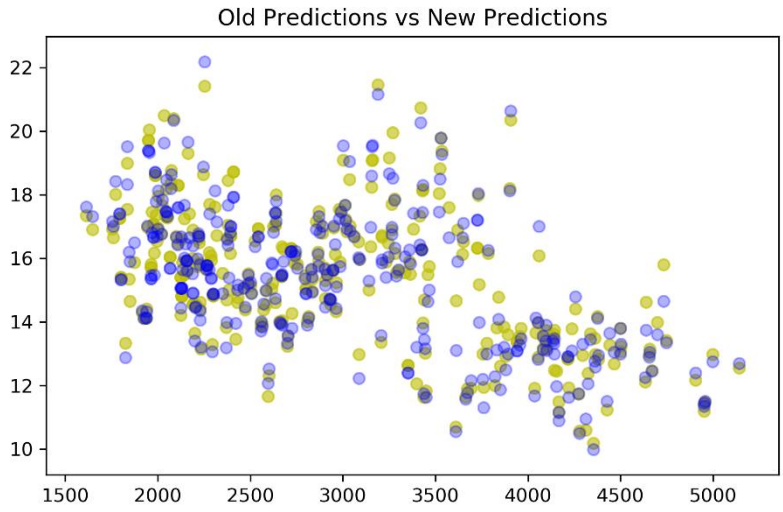
As the assignment requested, I went with a Random Forest model. I kept the random state set to 42 throughout the project, so that my reruns of cells would generate the same selections for the forest. When the forest was initially trained, it gave a decently close approximation of the acceleration of a car based on its weight. The MSE of this model was around 1.55, which I think is decent. In the below image, the guess is in yellow, and the real label is in red:



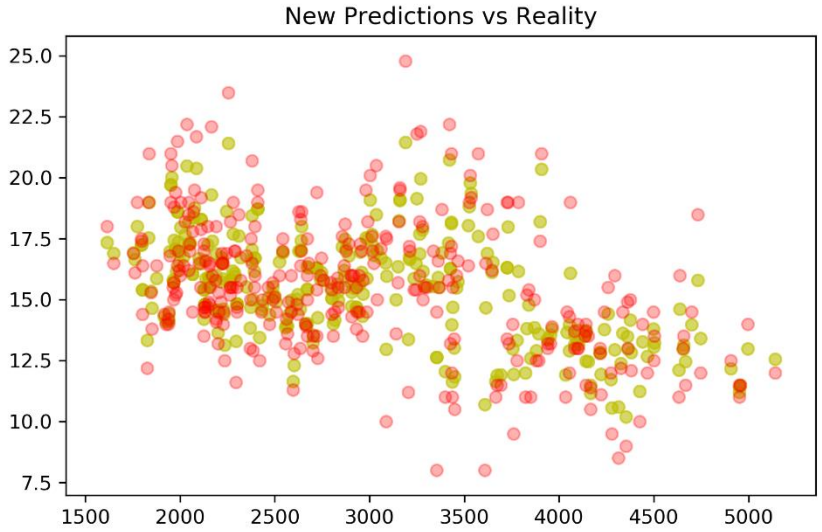
Model optimization and hyperparameter tuning:

I used a Grid Search Cross-Validation approach to hyperparameter tuning. The only hyperparameters that I changed were the “n_estimators” hyperparameter, which I gave four potential values to, and the Boolean “bootstrap” parameter, giving a total of 8 different configurations. Even with such a small amount of wiggle room, the model gained noticeable improvement, reaching a MSE of around 1.3:

In this image, the optimized guesses are in yellow, and the original guesses are in blue:



In this image, the optimized guesses are in yellow, and the real labels are in red:



Learning objectives and takeaway:

This biggest thing I learned from this project is the fact that machine learning has a huge amount of variety and change, and to know which tool to apply to which problem is very important. I also see that there is a lot I still need to learn, as I am unsure if my model would actually perform well in the real world.