Constantinos Skevofilax
cgs2528

COE 379L Cars Data Project 1

Data Preparation

To prepare the data, I read in the csv file using the built-in pandas function and looked at the data's size and shape. After using **cars.info()**, I looked at the count for each of the columns to see if they matched with the row count produced from **cars.shape**. When looking at the **cars.info()** output, I noticed that the data type of the column **'horsepower'** was of type object even though there were values of int. Looking further into this, I found that there were values that had a '?' instead for some specific rows. From this, I replaced the '?' with NaN and converted the column to type float. The reason why is because as one of the methods we learned in class, we can replace these missing values with that of the median value of the column. As such, I replaced the NaN's with the median value of the horsepower column. After this, I decided to split the **'origin'** into separate columns for each of the 3 world regions to do one-hot encoding. To be more specific, because the **'origin'** column had values between 1-3 it "means" that these ints are values, however, they are not values but rather categories.

Data Insights

After analyzing the data, I noticed that there was a high standard deviation for the **horsepower** and **displacement** columns. I also noticed that there was a relatively even standard deviation distribution for the **model_year** column. To get a better picture, I created histograms and boxplots to visualize the horsepower distribution because from the data it seems that there are outliers in the data. After visualizing it via a boxplot, you can see that 75% of the data falls within 75-125 horsepower, and 25% of the data falls out of the normal distribution bounds. Similarly, I wanted to analyze how a car's MPG would be affected by its horsepower, as well as

its weight vs acceleration. I created bivariate plots for this analysis and saw a negative exponential plot for MPG vs Horsepower and a horizontally linear plot for Weight vs Acceleration. I also wanted to see the model year by the number of cylinders because there would be innovation/evolution over time with the number of cylinders used in a car, which was noted in a histogram I created. Finally, wanting to analyze all the important columns such as 'mpg', 'cylinders', 'displacement', 'horsepower', 'weight', 'acceleration', I created a heat map to see if there was any correlation between each of these variables. I noted that displacement and the number of cylinders are closely related to each other, as well as cylinders and weight, and horsepower and displacement,

Training Model Procedure

To train the model, I imported the sklearn linear regression models. From there, I split the dataset into X and y, where X was made up of all the columns except **car_name** (because it is a string and non-numeric) and **mpg** (because **mpg** would be used as a y to identify what the fuel efficiency is). I set y to the values of cars[‘mpg’]. From there I used the sklearn train_test_splot function to split the X and y into training and test data, which I then loaded the training data into sklearn’s linear regression model.

Model Performance

I believe that the model performed well on prediction the fuel efficiency of the car. Comparing the standard deviation between the cars data set mpg (7.815984) and the model’s std (2.088389), you can see that the model was able to reduce the discrepancy between the changing mpg values and tie them more closely together.

Model Confidence

I’m confident that the model is more than 80% accurate, as the test score for the model’s performance was ~82.01% while the model’s training score was ~82.23%. They’re both very close to each other so I think that the model predicts relatively well. It will need more data to confirm it’s accuracy though.