Jason Kim, jk46965

COE379L Project 1 Report

Data Preparation/Preprocessing:
A crucial aspect of working with datasets and machine learning models is to preprocess the data. This preparation entails addressing missing values, converting any data types to numeric, having column headers for easy access/indexing, checking for consistency in number of rows, etc.

Specifically for this project, we began by using the *info()* function to get a comprehensive overview of the dataset, detailing the columns, their non-null count, and datatype.

The first thought process was to check for any columns with non-numeric data types and determine whether we want to convert it to numeric for later use. From block [4] in our jupyter notebook, we saw that there were two columns with object type - *horsepower* and *car_name*. For *car_name*, we left alone, as it was best represented by string type and presumably wouldn't be needed for later use in training our ML model. At first, I wondered whether we should hot-encode *car_name*, but I discovered that there were an extremely high number of distinct names and concluded it wouldn't serve any relevance in our study.

However, we aimed to convert *horsepower* to a numerical data type (float) for later use in training our ML model. Notably, when we tried converting *horsepower* to float, we received an error, saying that python could not convert String '?' to float. This suggested there were some rows with a String '?', so we had to replace these with *NaN* to allow for the type conversion.

Once we converted *horsepower* to float, we then had to address the presence of *NaN* values in the column by replacing them with the mean value of all the other *horsepower* values.

Insights from Data Preparation:
I've learned that data preparation is probably *the* most important step to take. Without clean data, it's likely we will run into issues when performing analysis and trying to create models.

An important skill to have when preparing data is knowing the right and useful functions to help understand the ins and outs of the dataset we are working with. For instance, being able to identify where and whether there are non-numeric or null values present.

It's also helpful to have statistical knowledge. As I've taken many statistics and data science courses, I understood why and what to replace null values with. Having an understanding of what can skew the data and cause inconsistencies is crucial in preparing data, as we need accuracy and consistency.

Furthermore, referring back to the prior section where I mentioned that I ran into an error when trying to convert *horsepower* column to float, it's important to note that in the future and for best practice, I should have first checked for any non-numeric values before trying to convert data type - and further extract what these non-numeric values were. As in other cases, there might've been rows with something other than a '?', leading me to continually running into *cannot convert to float type* errors until I've addressed all of the rows with non-numeric *horsepower* values.

Additionally, I was reflecting on the approach I took in replacing the rows with '?'. I first replaced them with *NaN* in order to convert them to float. However, in hindsight, this method doesn’t seem the most efficient, as later I had to account for these *NaN* values. Instead, I wonder and assume that there is a more efficient solution - specifically, to simply replace the ‘?’ with the mean *horsepower* value, skipping the step of addressing *NaN* values.

Procedure to Train the Model:
After preprocessing our data, we took the following steps in training our model:

1. Determine our response variable, which was given as fuel efficiency, *mpg*.
2. Visualize and analyze bivariate plots to identify correlations between variables, specifically possible predictors for *mpg*.
    a. Possible predictors: *'cylinders', 'displacement', 'horsepower', 'weight'*
    b. Heatmap of correlation matrix and pairplot extremely helpful.
3. Split the dataset into training and test sets.
    a. Our *X* consists of the four identified possible predictors from step 2.
    b. Our *y* is *mpg*.
    c. We used *train_test_split* from *sklearn.model_selection*.
4. Fit linear regression model using the training dataset.

Model Performance in Predicting Fuel Efficiency:
In block [23] in our notebook, we see that the accuracy of our model on the training and testing dataset is:

```
Test Score: 0.7490524242533151
Train Score: 0.6868525566338963
```

These scores indicate that our trained model explains approximately 74.9% of the variance in the test set for *mpg* and 68.7% for the training set. It’s important to note that because we worked with a linear regression model, it isn’t semantically correct to consider these scores to be metrics of “*accuracy* of our model in *predicting* fuel efficiency.” Rather, it’s more of a measure of how well the model fits the data.

Confidence in Model:
I would say I’m relatively confident in our model, as per the ~75% score on the test data, and ~69% on the training data.

How I used ChatGPT for Help:
ChatGPT helped me in debugging the *cannot convert ‘?’ to float* error and showing me how to properly use the *replace()* function. Furthermore, GPT taught me about *pairplot()*, a helpful function that allowed for a comprehensive analysis of the variables within our dataset and how they're correlated by visualization.