Fuel Efficiency Prediction:
Machine Learning Linear Regression

Jackson Thetford
COE 379L
Software Design for Responsible Intelligent Systems

Introduction

The purpose of this project is to introduce exploratory data analysis and various Python libraries, such as pandas, scikit-learn, seaborn and matplotlib. This project focuses on the processing and analysis of data in preparation to train a linear regression model focused on predicting the fuel efficiency of cars. The data is first processed using the pandas library. The processed data is then used to train a linear regression machine learning model using scikit-learn. This model is then tested, and relevant data is visualized to validate the model. The result is a machine learning model that is proficient in estimating the fuel efficiency of automobiles, which can offer insight into what characteristics of a car most affect the fuel efficiency.

Data Preparation

The dataset for this project is an automobiles dataset which consists of 398 entries with 9 variables associated with automobiles: the fuel efficiency (mpg), the number of cylinders, engine displacement, horsepower, weight, acceleration, model year, origin, and the car name. The dataset contains data types of float64 (3), int (4), and object (2). It should also be noted that there are 6 null values "?" in the horsepower column. To prepare this data for modeling the following steps were done:

1. **Dropping Columns:** The "car name" category had 305 unique entries and being a non-numeric value, had no potential to benefit our linear regression model. For this reason, the "car_name" variable was dropped from the data set.

2. **Type Conversion:** The "horsepower" attribute was classified as an object and had 6 missing values. To fix this, the `pd.to_numeric` function was called and the coerce option was used to convert the missing values to NaN values. This converted the valid horsepower entries to type float.

3. **Missing Values:** To address the missing values in the horsepower column, the statistical mean was used to fill the missing values.

4. **One-hot Encoding:** In the dataset, the "origin" attribute only had 3 unique values: 1, 2, and 3. This was transformed using one-hot encoding into two new binary columns "origin_2" and "origin_3" which allowed us to incorporate this categorical data into our linear regression model.

An analysis of the recently processed data was preformed and visualized to find correlations between columns, predict outliers in the dataset, and give insight into what variables may have a relationship with a vehicles fuel efficiency. Some noteworthy insights from this analysis are that the "horsepower" attribute has several outliers outside the range of 3 standard deviations from the mean; weight, horsepower, and displacement all have strong negative linear correlations with fuel efficiency; and the distribution of fuel efficiency is right skewed with a mean around 23.5 mpg (standard deviation of 7.8 mpg) and a mode around 20 mpg.

Linear Regression Model Fitting

To train the linear regression model the scikit-learn library was used. The objective of the model was to predict the fuel efficiency (mpg) of a vehicle given the other 8 variables in the dataset. The procedure used train the model was as follows:

1. **Data Splitting:** First, the target variable "mpg" column was isolated and saved as the independent variable "y". The other 8 independent variables were saved under the variable "X", including the added "origin_2" and "origin_3" and excluding the "car_name" variables. This data (the independent and dependent variables) was then split into training and testing sets using scikit-learn's `train_test_split` function, with 70% of the data allocated to the training set and the other 30% to the testing set. Random sampling was used to split the data into testing and training sets.

2. **Model Training:** The training data was then fed to a linear regression model using scikit-learn's `LinearRegression` class. The model was fit to the training data, where the model can be represented by the equation:

$$\begin{aligned} mpg \;=\; & (-0.396) * cylinders \;+\; (0.029) * displacement \;+\; (-0.021) * horsepower \\ & +\; (-0.007) * weight \;+\; (0.066) * acceleration \;+\; (0.838) * model\_year \\ & +\; (2.991) * origin\_2 \;+\; (2.378) * origin\_3 \;-\; 21.379795484727328 \end{aligned}$$

Model Analysis

To evaluate the model's performance, the R^2 score, or coefficient of determination, was used, a function in scikit-learn's "metrics" class. It was found that the model had a training R^2 value of 0.814 and a test R^2 of 0.843 (`random_state=1`). The R^2 score represents the variance in the dependent variable. As a R^2 score of 1 would mean our model is perfect, the high and consistent R^2 scores of both the training and testing data set indicate that our model is neither overfitting or underfitting and our model generalizes well to new data as it has found the underlying patterns in the independent variables without overfitting.

For our case, the test accuracy (the R^2 value) is more relevant and indicative of our model's performance. This is because the test data set is data the model hasn't seen before, and a high R^2 score on the test set tells us that the model can perform well, beyond just the training data which would be expected. While the training R^2 score tells us how well the model fits the training data, an overfit model could perform well on a training set but poorly on a testing set. The R^2 score of the testing data represents how our model will perform on the real-world data that we could give it related to the fuel efficiency of cars. The test accuracy is much more relevant to our problem statement as it gives us a better sense on how our model will perform in real world scenarios.

This fairly high R^2 value of 0.843 for our testing data gives us fairly high confidence in our model's ability to predict a vehicles fuel efficiency.

Resources

1. Data: https://raw.githubusercontent.com/joestubbs/coe379L-sp24/master/datasets/unit01/project1.data

2. Class Repo: https://coe-379l-sp24.readthedocs.io/en/latest/index.html

3. ChatGPT: https://chat.openai.com
   * Noted in the .ipynb when used. (Used for visualization formatting / generating some figures)