report

Part 1- Regression

Introduction

In this assignment, we had to create a regression model to predict fuel consumption of cars of different types from the provided features. In the training dataset, we are provided with the following information: Year, Make, Model, Vehicle class, Engine Size, Number of Cylinders, Transmission type, Fuel Type, Fuel Consumption, CO emissions.

In order to complete this task, i made a multiple linear regression model that uses a few of these features, trained on the given fuel_train.csv dataset to predict Fuel consumption information.

Methodology

In order to build and train this model, I built a model to optimize the linear regression parameters to minimize the sum of the square of error terms. If we are using p features from the dataset which is n datapoints in size, the matrix $X \in \mathbb{R}^{n \times p+1}$ is our feature matrix that has n rows and $p+1$ columns. The vector $\beta \in \mathbb{R}^{p+1}$ contains all the parameters that we need to set in the model. $Y, \epsilon \in \mathbb{R}^n$ are the vectors of the target variable (fuel consumption) and the residuals (difference between predicted and actual fuel consumption). The model is given as

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_p x_{pi} + \epsilon_i$$

Representing the model using matrix notation,

$$Y = X\beta + \epsilon$$

In order to find the optimal value of β, we need to minimize the sum of square residuals

$$\min_{\beta} (Y - X\beta)^2$$

Taking the derivative with respect to β, we get

$$\frac{\partial}{\partial \beta}((Y - X\beta)^T (Y - X\beta)) = -2X(Y - X\beta)$$

To minimize, we set this value to zero,

$$-2X(Y - X\beta) = 0$$

$$-2X^TY + 2X^TX\beta = 0$$

$$X^TX\beta = X^TY$$

Therefore, the optimal values of β that minimize the sum of square residuals is given by

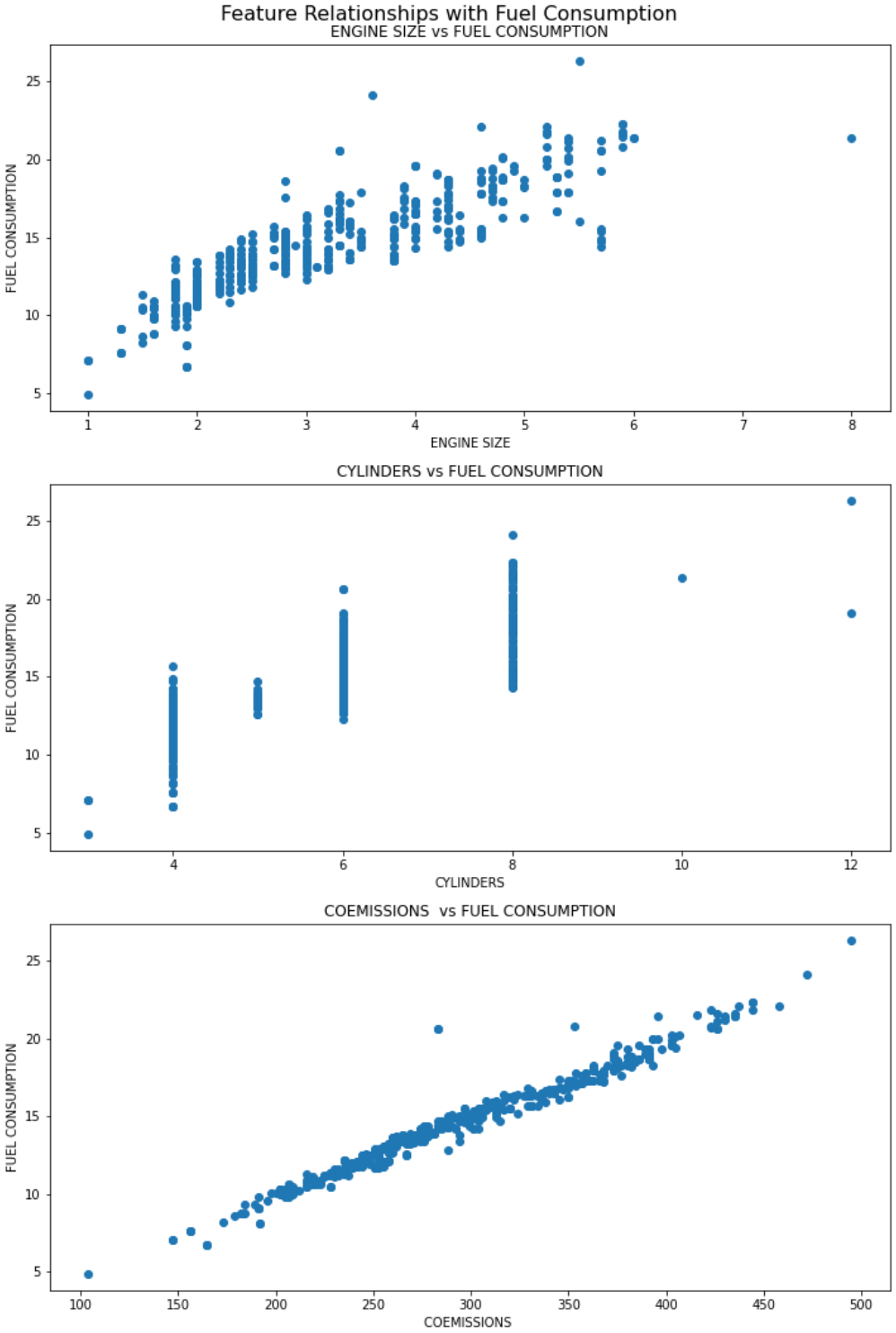
$$\beta^* = (X^TX)^{-1}X^TY$$

This can be executed fairly easily using linear algebra operations available in numpy.

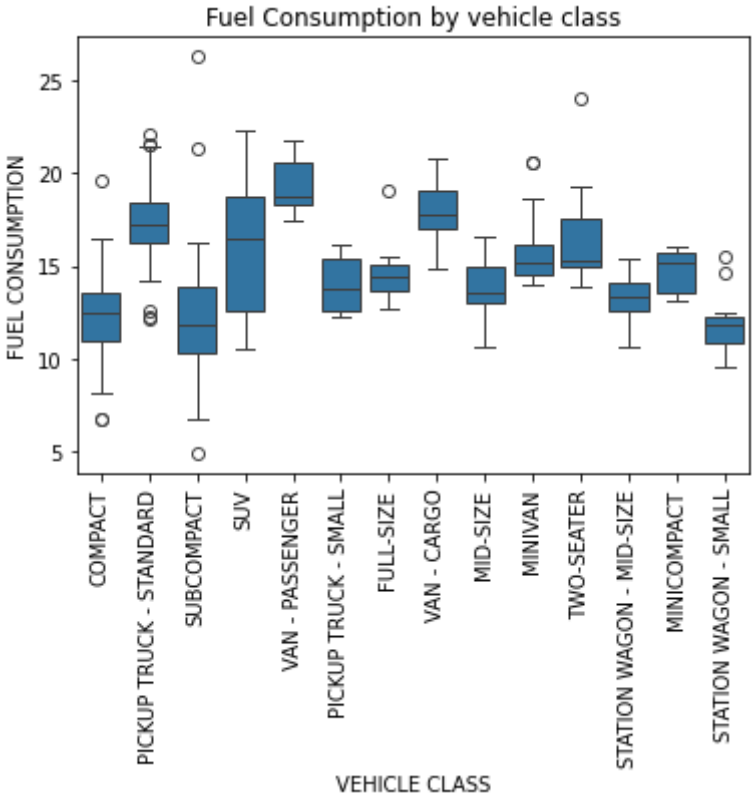
Experimentation

Pre-model observations

Before building out the model, using libraries like matplotlib and seaborn, I tried to get some understanding of the dataset. Plotting scatterplots between the individual numerical features and fuel consumption,



CO emissions appears to be the closest related to fuel consumption. Further, looking at the correlation matrix, we find that Engine size, cylinders , fuel consumption and co emission are all highly correlated with each other. Using more than one of these variables may result in multicolinearity issues.



This boxplot shows the distribution of fuel consumption according to emissions. This could be a promising variable for prediction, but we need to convert it from categorical to many binary variables.

Model 1

We use the simple model with CO emissions as a first step. The regression equation is

$$y_i = \beta_0 + \beta_1 \text{CO emissions}_i + \epsilon_i$$

Model 2

To ideally improve the simple model, we can try using vehicle class to give the model more features. We first combine a few of the classes, in order to reduce the total number of binary variables necessary. Compact and Subcompact are combined into Small Car, Passenger and cargo vans into Vans, Small and mid-size station wagons into Station wagon, Standard and small pickup trucks into pickup trucks, minicompact and two seater into very small car. We then use these new categories to create 9 total variables in our regression equation

$$y_i = \beta_0 + \beta_1 \text{CO emissions}_i + \beta_2 \text{VC Small car}_i + \cdots + \beta_9 \text{VC van} + \epsilon_i$$

Model 3

To test if this model can be improved further, we could drop some of the dummy variables that not statistically significant (their coefficient β is not statistically different from zero). This is for two classes - minivans and very small cars. Therefore our simpler model becomes

$$y_i = \beta_0 + \beta_1 \text{CO emissions}_i + \beta_2 \text{VC Small car}_i + \cdots + \beta_7 \text{VC van} + \epsilon_i$$

Results

Metric	Model 1	Model 2	Model 3
MSE	0.3853	**0.2824**	0.2861
RMSE	0.6207	**0.5314**	0.5349
R^2	0.9614	**0.9717**	0.9714

Based on the training data, Model 2 achieves the best performance according to MSE, RMSE and R^2. Even when adjusting the R-squared for the additional loss of degrees in freedom in model 2 due to a larger number of features using Adjusted R-squared, Model 2 still shows the best in-sample performance.