

ML
Assignment - 1

Ojasva Singh
2020318

Attempting **Sections A and C**

Section A

1.

- a.** No, if two variables exhibit a strong correlation with a third variable, they don't need to display a high degree of correlation.

Let's understand this with the help of an example,

We can assume that there is a very famous restaurant chain, so our first variable is the number of chefs required, and the second variable is the number of tables and chairs at the restaurants, i.e., an increase in the number of tables and chairs does not result in increase in number of chefs required.

Now, we can see that these two variables are not correlated. But both of these variables are correlated to the number of restaurants, which is the third variable. As the number of restaurants increases, there must be an increase in the number of chefs required and tables and chairs.

At the same time, there could be a restaurant with many tables and chairs but fewer chefs, depending on the location and crowd. The vice versa is also possible.

Therefore, the relationship between the first and second variables is not guaranteed and may be indirect or coincidental.

- b.** Logistic regression is a type of regression used for a binary response variable or a classification-type problem. It uses a function to map real numbers to $\{0,1\}$.

The defining criteria for a mathematical function to be categorised as a logistic regression are:

1. It has a sigmoid curve.
2. It approaches limiting values.
3. The curve should be symmetrical around a particular point of the curve.

Let's look at the below functions and list them as logistic or not.

1. $\sinh(x)$: We can see that it is not a sigmoid function, and it does not approach any limiting values. Hence, it is not a logistic function.
2. $\cosh(x)$: Similar to hyperbolic sine, this function does not represent the sigmoid function and does not approach any limiting value; hence, it is not a logistic function.
3. $\tanh(x)$: This function is a logistic function since it represents a sigmoid function and is also bounded between -1 and 1.
4. $\text{signum}(x)$: This function does not have a smooth curve and, therefore, is not a logistic function.

- c. We can use the Leave One Out Cross Validation technique, which is beneficial for very sparse datasets; this technique maximises the utilisation of the small dataset by creating a separate fold for each individual data point.

In this technique, since the dataset is small, each data point acts as a validation set while the rest are used for training. Also, because the dataset is small, the iterations are less, and the model's performance is enhanced since it is getting trained using the validation sets for n number of iterations where n is the size of the dataset.

In K-Fold cross-validation, we make K folds and in each iteration, one fold act as the validation set and the $k-1$ fold act as the training set.

Whereas in LOOCV, each data point acts as a validation set while the others act as the training set; this helps in using the small data to its fullest extent.

LOOCV can be more costly in computation terms, but it also performs well.

Therefore, the validation technique to be used ultimately depends on the size of the data set.

d.

(d)

TF: Coefficients of the least square regression line for a set of n data points (x_i, y_i) in slope-intercept form.

→ slope intercept form = $y = mx + b$

where m is the slope
 b is the y-intercept

→ Equation for regression is

$$y = \beta_0 + \beta_1 x + \epsilon$$

where ϵ is the error.

→ Assuming $\epsilon \sim N(0, \sigma^2)$

$y_i \sim \text{normal}$ as a linear combination of normal is normal.

$$\epsilon_i = y_i - (\beta_0 + \beta_1 x_i)$$

⇒

$$y_i \sim N(\beta_0 + \beta_1 x_i, \sigma^2)$$

Minimizing total square loss

$$\min \sum_{i=1}^n e_i = \min \sum_{i=1}^n (y_i - \beta_0 + \beta_1 x_i)^2$$

since $e_i \sim N(0, \sigma^2)$

$$E(y_i | x_i) = \beta_0 + \beta_1 x_i$$

$$\sum e_i^2 = \sum (y_i - \beta_0 - \beta_1 x_i)^2$$

$$\frac{d}{d\beta_0} \sum (y_i - \beta_0 - \beta_1 x_i)^2 = 0$$

$$\sum y_i - n\beta_0 - \beta_1 \sum x_i = 0 \quad \text{--- (1)}$$

$$\frac{d}{d\beta_1} \sum (y_i - \beta_0 - \beta_1 x_i)^2 = 0$$

$$\sum (y_i - \beta_0 - \beta_1 x_i) x_i = 0$$

$$\sum y_i x_i - \beta_0 \sum x_i - \beta_1 \sum x_i^2 = 0 \quad \text{--- (2)}$$

$$(\sum y_i - n\beta_0 - \beta_1 \sum x_i) \sum x_i$$

$$(\sum y_i x_i - \beta_0 \sum x_i - \beta_1 \sum x_i^2) n$$

$$\sum x_i \sum y_i - n \sum x_i y_i = \beta_1 [(\sum x_i)^2 - n \sum x_i^2]$$

$$\Rightarrow \hat{\beta}_1 = \frac{\sum x_i \sum y_i - n \sum x_i y_i}{(\sum x_i)^2 - n \sum x_i^2}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

- e. The correct answer is (a) α , β , σ .

The other options include epsilon, which is the just the noise that follows the normal distribution with mean 0 and standard deviation σ . Apart from this a,b,s are not used in the equation, instead greek letters are used.

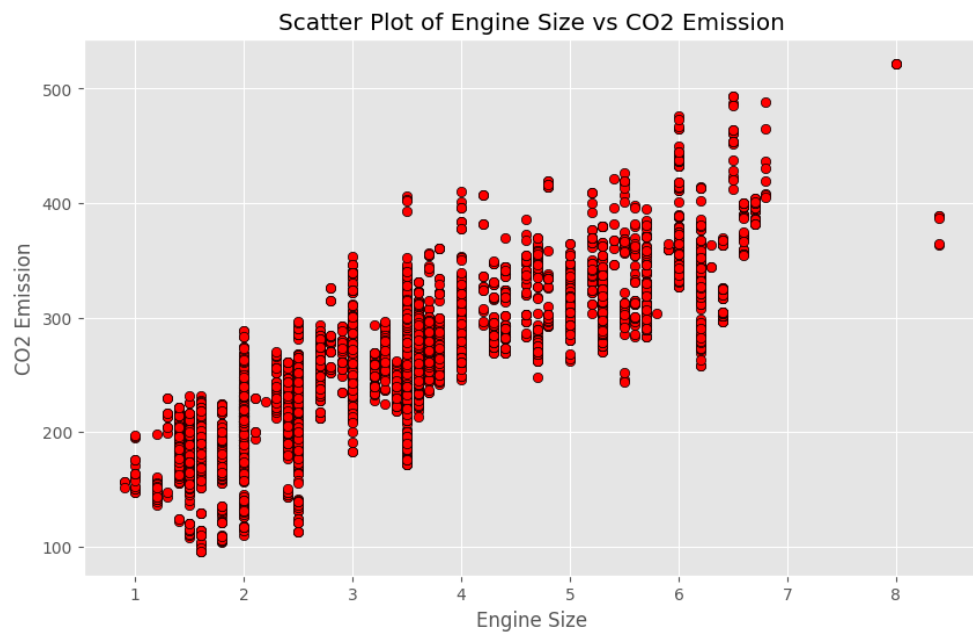
- f. The correct answer is (c) $Y = \alpha + \beta_1 X + \beta_2 X^2 + \epsilon$

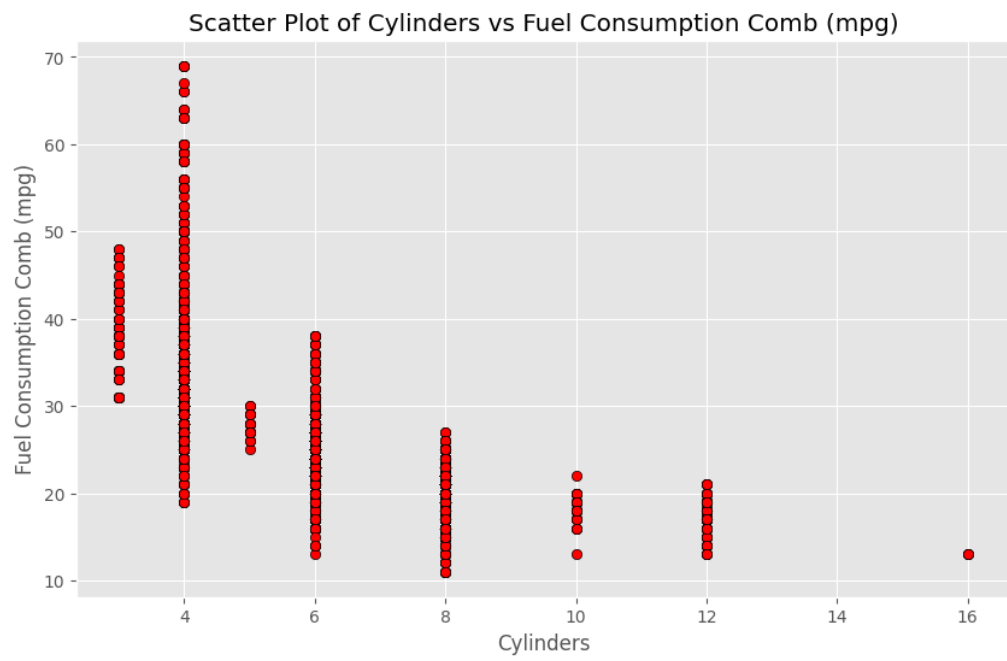
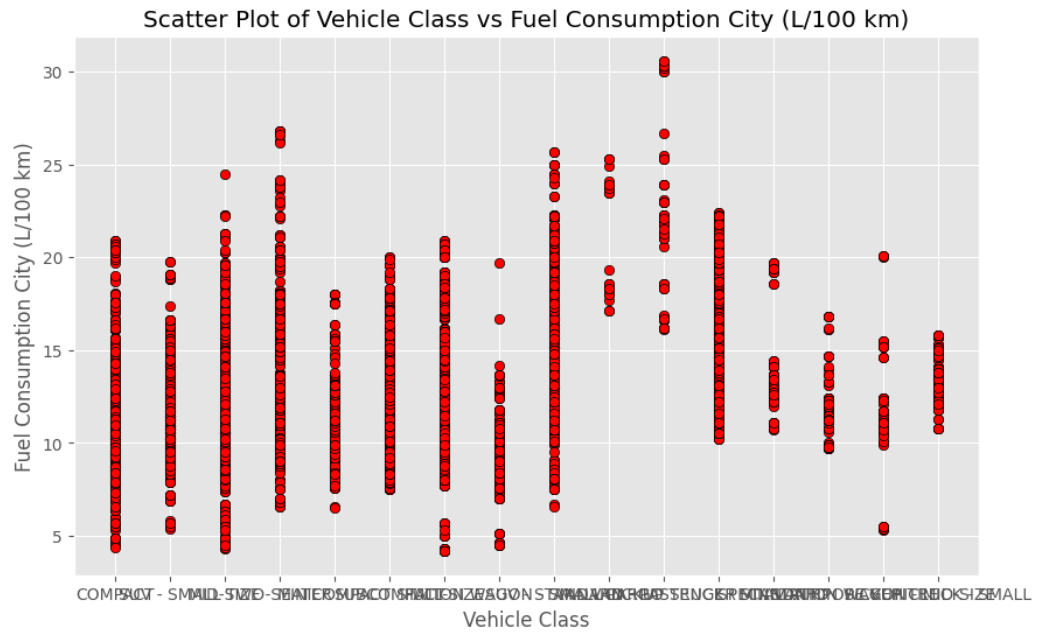
Out of all the options, only option (c) shows a U-shaped curve since Y decreases till a certain increase in X, and after that, Y increases as X increases. This curve best fits the data provided to us.

SECTION - C

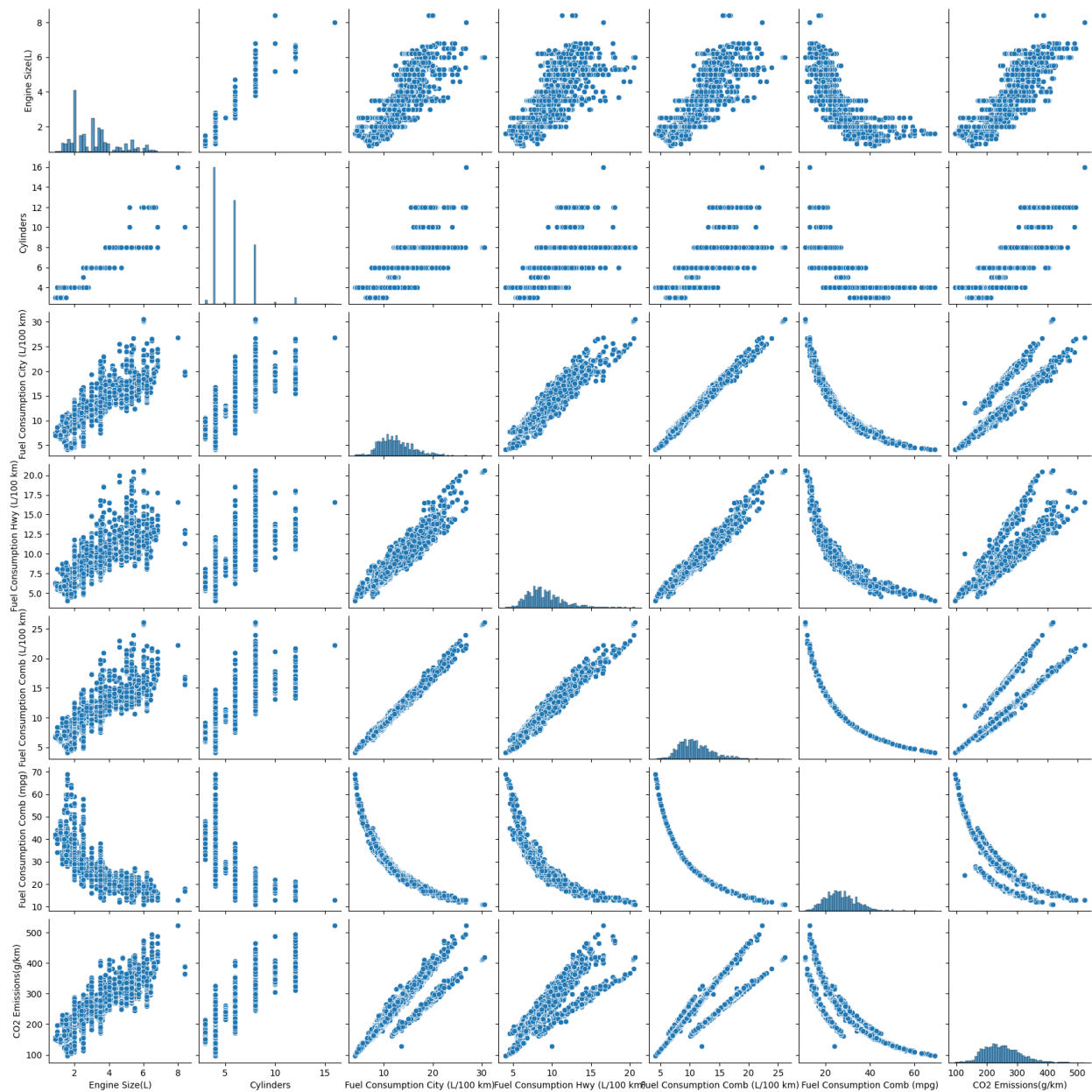
2.

- a. Scatter Plot

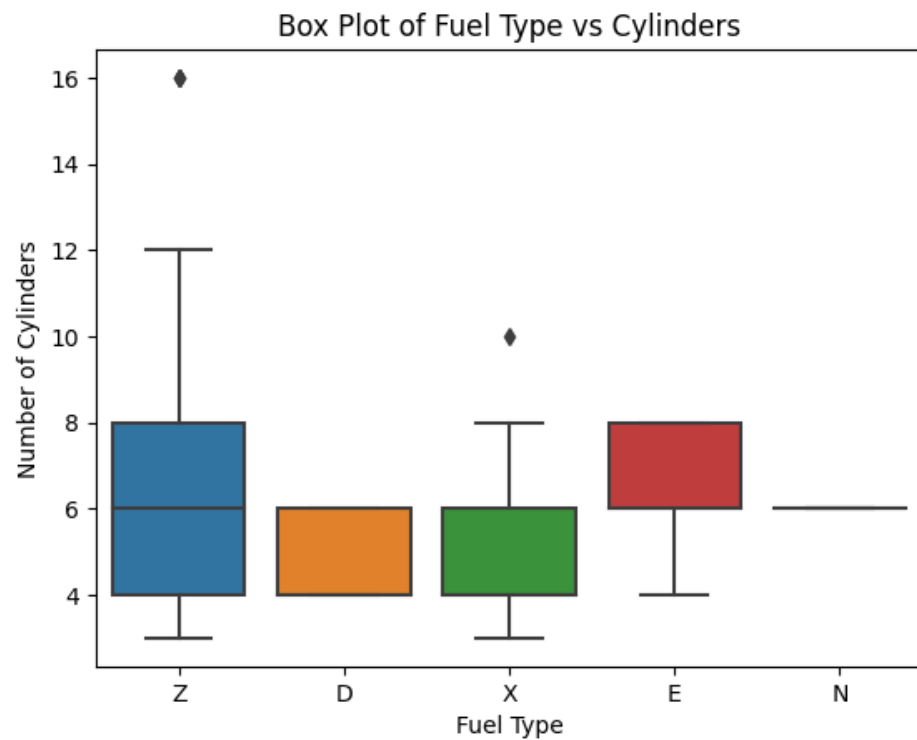
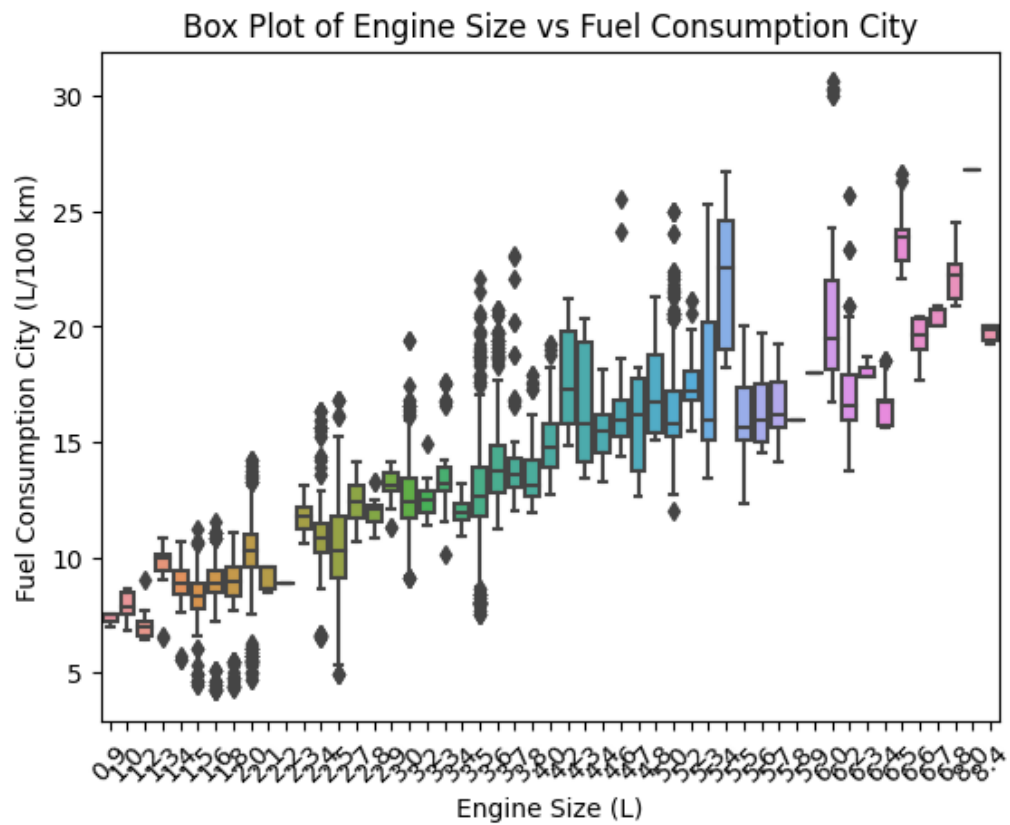




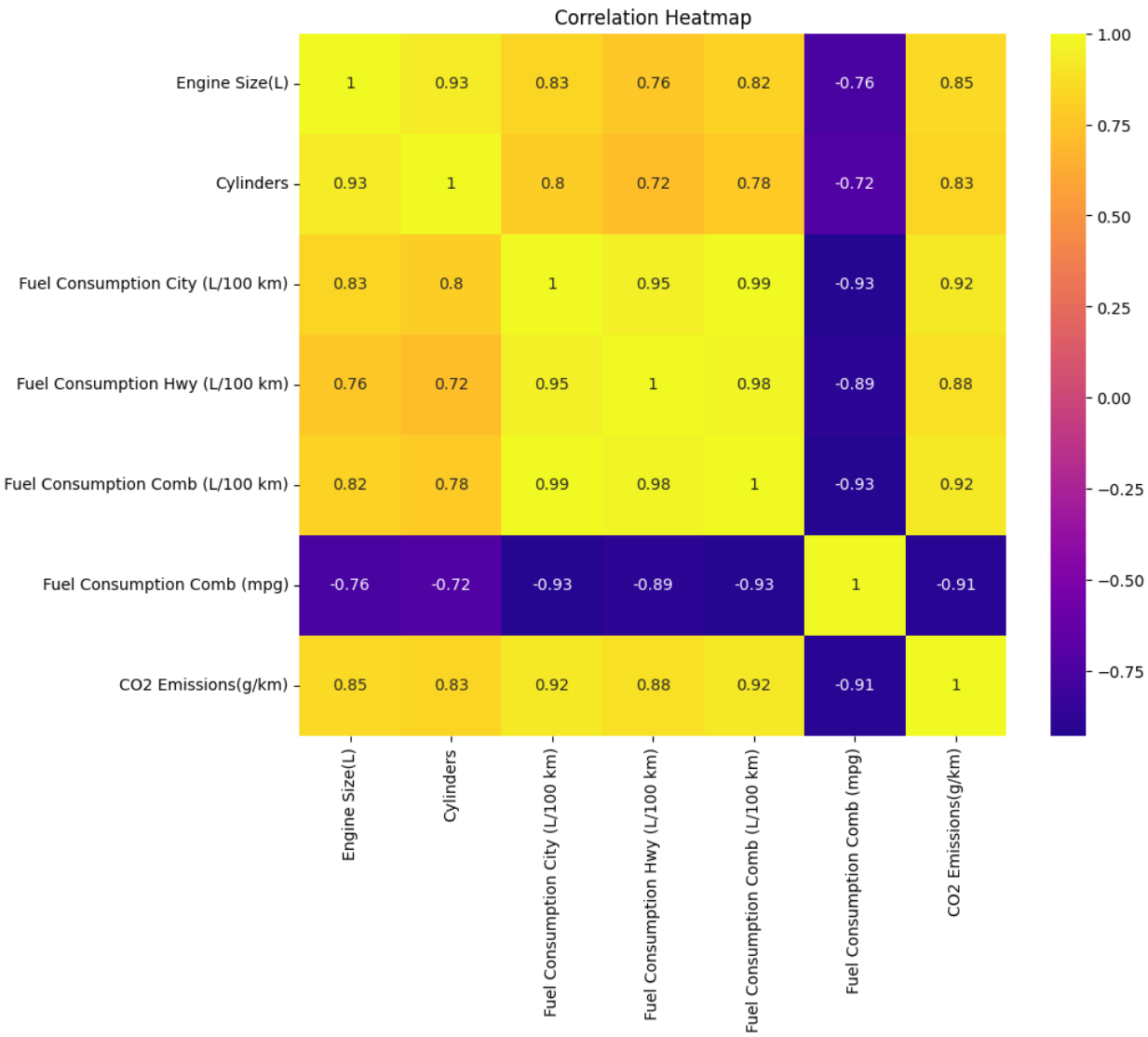
Pair Plot



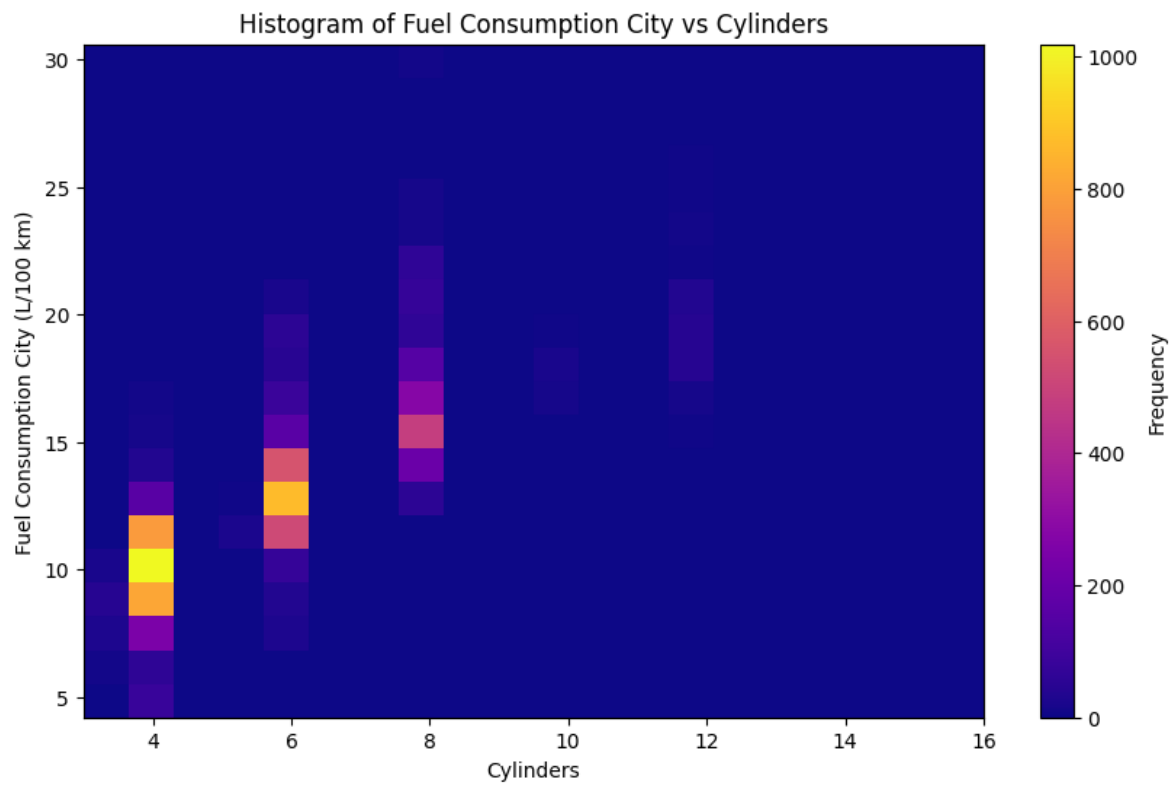
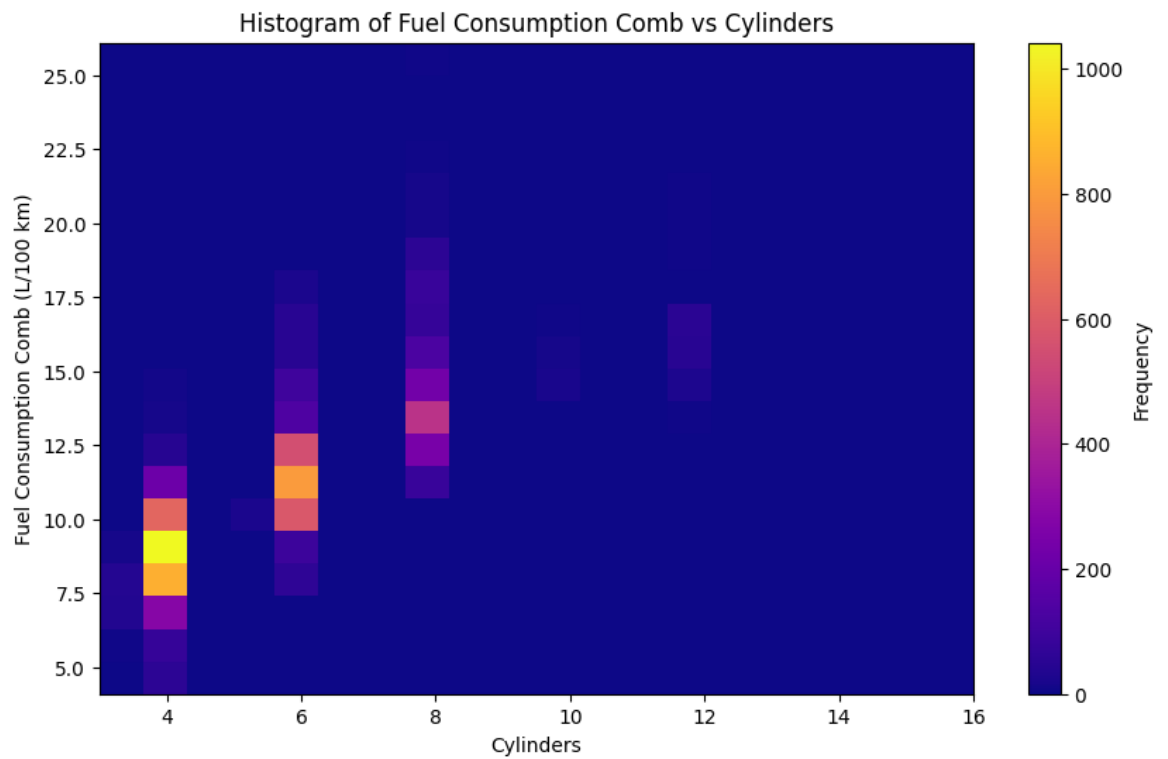
Box Plot

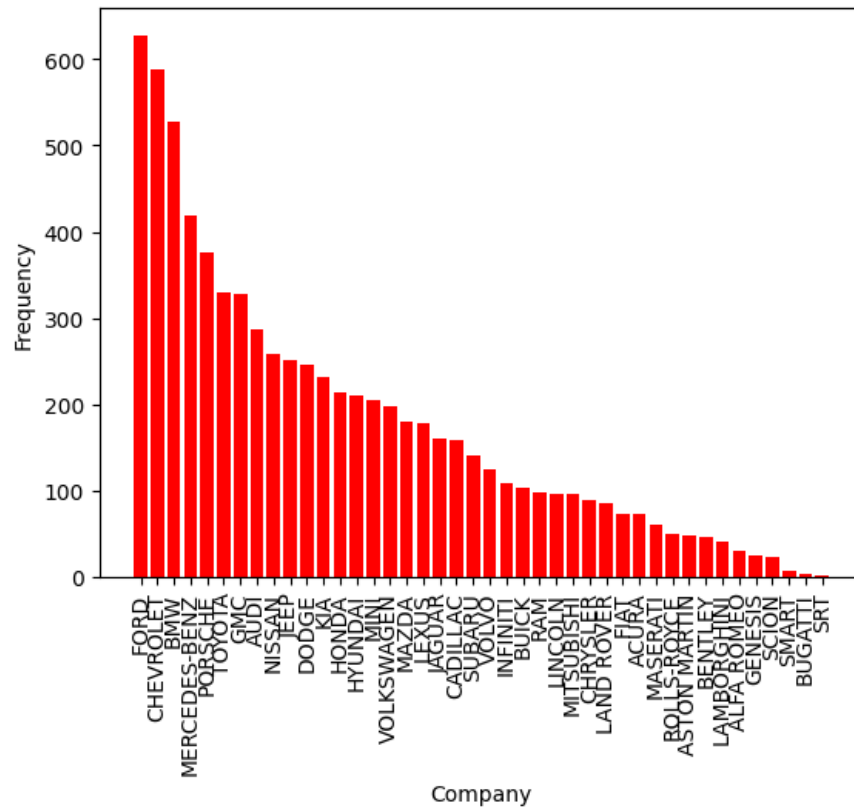


Correlation Heatmap



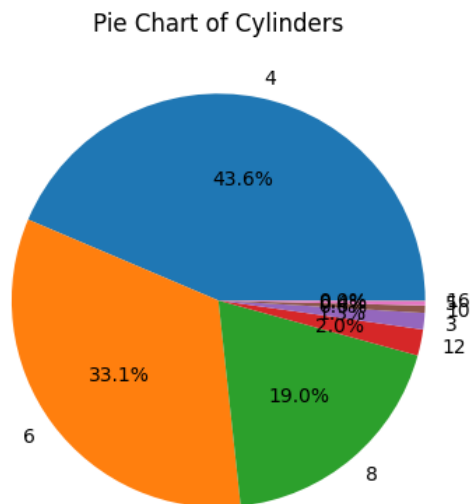
Histograms

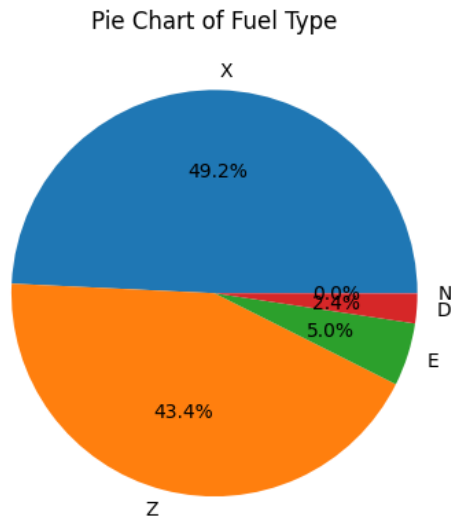




Frequency vs. Company Bar Plot

Pie Chart

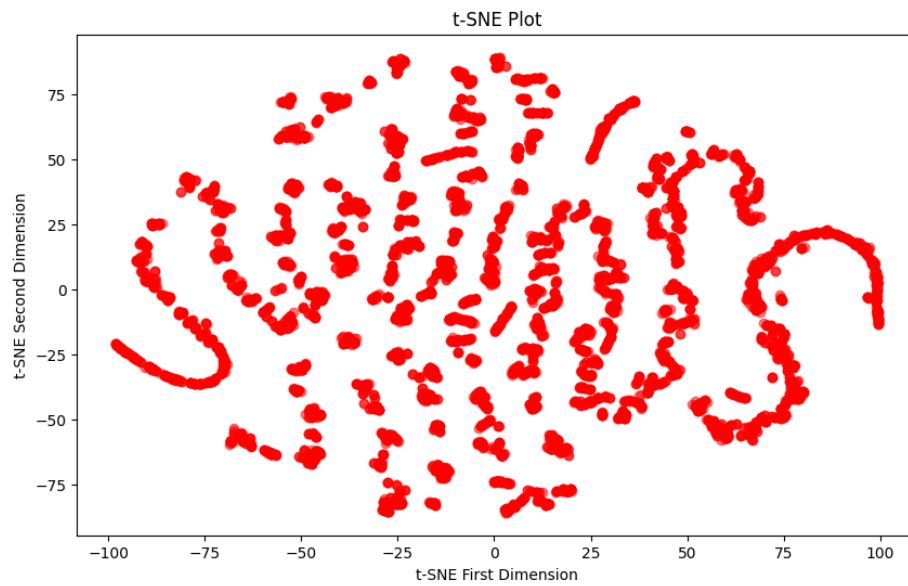




Insights from the plots:

- X is the most used type of fuel, with a percentage of 49.2%.
- As the engine size increases, the amount of CO2 emissions also increases.
- As the number of cylinders increases, the fuel consumption comb (mpg) decreases.
- In the data provided, FORD has the maximum number of vehicles exceeding 600.
- There is a high correlation between Cylinders and CO2 emissions.

b.



Comments on the plot

- i. t-SNE is a dimension reduction technique and is helpful for visualising large datasets.
- ii. From the plot, we can see that there is weak separability since there is overlapping and mixing of data points.

c.

Used label-based encoding for the categorical features, which basically converts the categorical data to numerical form by assigning a unique integer value to them.

Below are the errors observed for train and test data by using linear regression:

```
[45] ✓ 0.0s
... Train MSE: 282.8600239315867, Test MSE: 308.31245189424703

Train RMSE: 16.818442969894292, Test RMSE: 17.55882831780774

Train R2 Score: 0.9179997219430924, Test R2 Score: 0.9069404719936918
Adjusted R2 Score (Train): 0.91738402019659, Adjusted R2 Score (Test): 0.9062417315103679

MAE on train: 10.957467304326746, MAE on test: 11.397225891250429
```

- d. First performed label-encoding on the categorical data and then split the dataset for training and testing. Then, PCA reduced the number of features of the dataset and trained the model with the reduced features. Did this 4 times for a different number of features. Below are the errors and other data obtained from running the model for a different number of features.

```
CO2 Emissions.csv  2020318_Ojasva.ipynb  2020318_Ojasva.ipynb (output) ×
1  Number of Components: 4
2  Train MSE: 448.1072794008468, Test MSE: 471.59020119297276
3
4  Train RMSE: 21.168544574458746, Test RMSE: 21.716127674909558
5
6  Train R2 Score: 0.8677405357865231, Test R2 Score: 0.8676095806911696
7  Adjusted R2 Score (Train): 0.8673811350685516, Adjusted R2 Score (Test): 0.8672498241169608
8
9  MAE on train: 13.626655831914766, MAE on test: 14.066003002061906
10 -----
11 Number of Components: 6
12 Train MSE: 365.0860581107382, Test MSE: 392.8045769487023
13
14 Train RMSE: 19.10722528549706, Test RMSE: 19.819298094249007
15
16 Train R2 Score: 0.8922443605421928, Test R2 Score: 0.889727219698132
17 Adjusted R2 Score (Train): 0.8918045416056304, Adjusted R2 Score (Test): 0.889277126717308
18
19 MAE on train: 11.015015809975628, MAE on test: 11.614035080429435
```

```
CO2 Emissions.csv  2020318_Ojasva.ipynb  2020318_Ojasva.ipynb (output) X
20 -----
21 Number of Components: 8
22 Train MSE: 284.908275722602, Test MSE: 312.74329069715566
23
24 Train RMSE: 16.8792261588795, Test RMSE: 17.684549490930088
25
26 Train R2 Score: 0.9159089404942488, Test R2 Score: 0.912202977740673
27 Adjusted R2 Score (Train): 0.9154506785895853, Adjusted R2 Score (Test): 0.9117245198537013
28
29 MAE on train: 11.05593979153963, MAE on test: 11.53406278378542
30 -----
31 Number of Components: 10
32 Train MSE: 281.4585845735203, Test MSE: 313.39908394152314
33
34 Train RMSE: 16.776727469131764, Test RMSE: 17.70308119908857
35
36 Train R2 Score: 0.9169271214612922, Test R2 Score: 0.9120188756486829
37 Adjusted R2 Score (Train): 0.9163604578969081, Adjusted R2 Score (Test): 0.911418731553517
38
39 MAE on train: 10.966764583768496, MAE on test: 11.530103015411122
40 -----
```

From the results obtained, we can conclude that:

1. As the number of components increases, the R2 score improves until one point, and afterwards, even after increasing the number of components, does not bring a significant change in the scores.
2. We can see that from 8 components onwards, the scores don't improve much, but the complexity increases.

e.

Instead of label encoding, I used one hot encoding here and obtained the following results.

```
CO2 Emissions.csv  2020318_Ojasva.ipynb X
2020318_Ojasva.ipynb > for item in [4, 6, 8, 10]:
+ Code + Markdown | Run All Restart Clear All Outputs | Variables Outline ...
    print("MAE on train:", maetrain, end=", ")
    print("MAE on test:", maetest)
[46] ✓ 0.0s
... Train MSE: 8.87739790104233, Test MSE: 3.9443117359873314e+20
    Train RMSE: 2.9794962495432564, Test RMSE: 19860291377.488224
    Train R2 Score: 0.9973743944873489, Test R2 Score: -1.1024495107522789e+17
    Adjusted R2 Score (Train): 1.005758385938593, Adjusted R2 Score (Test): 2.4178536075339728e+17
    MAE on train: 1.8966879615447936, MAE on test: 3510939044.114403
```

In part C, we observed that the model trained on the training data performed almost similarly, and the difference between the performance of the training and testing data did not vary significantly. However, we can see in this part that the difference between the training and testing data is enormous for all the metrics; also, the errors for the training data were reduced,

and the performance was also better. All of this might be due to overfitting. This implies that this encoding increased the computational complexity, as seen in the dataset.

The dataset now possesses unique values as new columns encoded with 1 or 0 to show the inclusivity of the value in the data.

All this leads to terrible results for the testing data.

f.

Similar to part d, we applied PCA here, but the encoding used here is one-hot encoding which is different from the one used in the c part. The results obtained are as follows for a different number of components.

```
CO2 Emissions.csv  2020318_Ojasva.ipynb  2020318_Ojasva.ipynb (output) X
1  Number of Components: 4
2  Train MSE: 330.17230244070015, Test MSE: 345.99722728050995
3
4  Train RMSE: 18.17064397429822, Test RMSE: 18.601000706427328
5
6  Train R2 Score: 0.9019919956718352, Test R2 Score: 0.9049071340461069
7  Adjusted R2 Score (Train): 0.9017256695731173, Adjusted R2 Score (Test): 0.9046487295190583
8
9  MAE on train: 11.45184502747027, MAE on test: 12.017376758032064
10 -----
11 Number of Components: 6
12 Train MSE: 321.4190262231965, Test MSE: 335.41139801167134
13
14 Train RMSE: 17.92816293498016, Test RMSE: 18.31424030670318
15
16 Train R2 Score: 0.9045903091192959, Test R2 Score: 0.9078165124003339
17 Adjusted R2 Score (Train): 0.9042008818095788, Adjusted R2 Score (Test): 0.9074402532672741
18
19 MAE on train: 11.28790604461448, MAE on test: 11.768132667988159
20
```

```
CO2 Emissions.csv  2020318_Ojasva.ipynb  2020318_Ojasva.ipynb (output) X
20 -----
21 Number of Components: 8
22 Train MSE: 320.87603176176947, Test MSE: 335.2140978412685
23
24 Train RMSE: 17.913012916920746, Test RMSE: 18.30885299086943
25
26 Train R2 Score: 0.9047514910328978, Test R2 Score: 0.9078707378020929
27 Adjusted R2 Score (Train): 0.9042324255889354, Adjusted R2 Score (Test): 0.9073686709781261
28
29 MAE on train: 11.246322430672498, MAE on test: 11.712801814760423
30 -----
31 Number of Components: 10
32 Train MSE: 320.7050997883268, Test MSE: 334.9834077704262
33
34 Train RMSE: 17.908241113753377, Test RMSE: 18.3025519469397
35
36 Train R2 Score: 0.9048022303028763, Test R2 Score: 0.9079341399864281
37 Adjusted R2 Score (Train): 0.9041528594318182, Adjusted R2 Score (Test): 0.9073061327557762
38
39 MAE on train: 11.255195396199627, MAE on test: 11.72804737363519
40 -----
```

```
CO2 Emissions.csv  2020318_Ojasva.ipynb  2020318_Ojasva.ipynb (output) X
40 -----
41 Number of Components: 12
42 Train MSE: 302.01275430443764, Test MSE: 372.0827503370913
43
44 Train RMSE: 17.378514156982398, Test RMSE: 19.28944660525779
45
46 Train R2 Score: 0.9110350837808205, Test R2 Score: 0.8946245641560436
47 Adjusted R2 Score (Train): 0.9103058631560732, Adjusted R2 Score (Test): 0.8937608310753554
48
49 MAE on train: 11.159273941156542, MAE on test: 12.292700180796718
50 -----
```

From the results obtained, as the number of components increases, the performance also increases, but it does not vary much. This might be due to other factors, such as the data itself. Also, we can see that the errors decrease as we increase the number of components. In the previous part, the same encoding was used, but due to the very high complexity of the dataset, we observed overfitting and terrible testing results.

g.

L1 Regularization - Lasso

```
CO2 Emissions.csv  2020318_Ojasva.ipynb X
2020318_Ojasva.ipynb > for item in [4, 6, 8, 10]:
+ Code + Markdown | Run All Restart Clear All Outputs | Variables Outline ...
print(MAE on test, mae_test_lasso)

[51] ✓ 0.2s
... Train MSE: 290.07444337007627, Test MSE: 279.83667114257395

Train RMSE: 17.031571958280196, Test RMSE: 16.728319435692693

Train R2 Score: 0.9160639984038067, Test R2 Score: 0.9149178289724959
Adjusted R2 Score (Train): 0.9154337622143472, Adjusted R2 Score (Test): 0.9142789867326989

MAE on train: 11.140935954864489, MAE on test: 10.828263957660283
```

L2 Regularization - Ridge

```
CO2 Emissions.csv  2020318_Ojasva.ipynb X
2020318_Ojasva.ipynb > for item in [4, 6, 8, 10]:
+ Code + Markdown | Run All Restart Clear All Outputs | Variables Outline ...
print(MAE on test, mae_test_lasso)

[52] ✓ 0.0s
... Train MSE: 289.9385768939773, Test MSE: 280.1394248898702

Train RMSE: 17.027582825932086, Test RMSE: 16.737366127616085

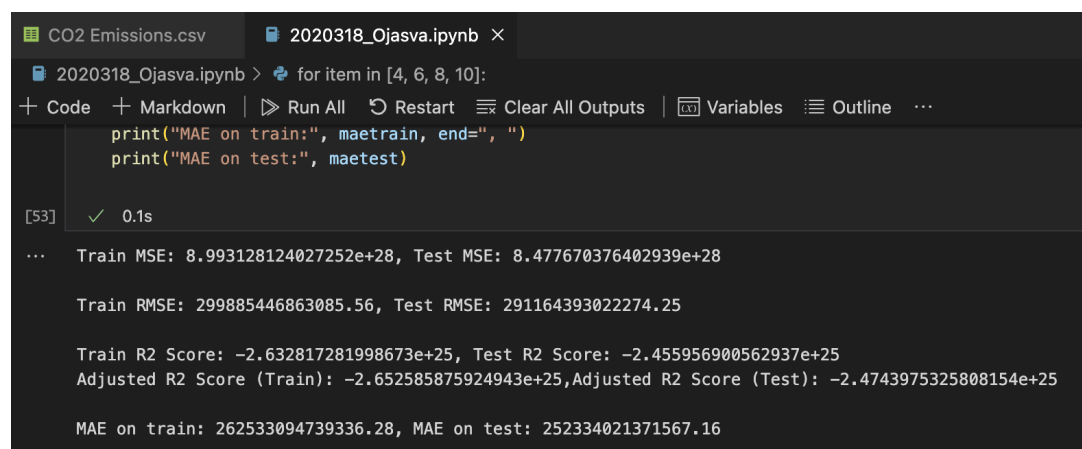
Train R2 Score: 0.9161033127557442, Test R2 Score: 0.9148257790420793
Adjusted R2 Score (Train): 0.9154733717593709, Adjusted R2 Score (Test): 0.9141862456423953

MAE on train: 11.137804637535634, MAE on test: 10.836968787115586
```


From performing L1 and L2 regularisations and by looking at the results, we can conclude that even though the performance is very minutely better for L2, L2 is performing better than L1. Ridge regularisation is performing better because, for large datasets, it does not act sensitively to the noise in the data, which helps in keeping the model stable. At the same time, Lasso regularisation works better in cases where only a few features are relevant to the model.

h.

Used the preprocessed dataset of part c, where label-based encoding was used. Then the SGDRegressor model to fit the data and evaluate the error, r2 and more metrics, shown below.



```
CO2 Emissions.csv 2020318_Ojasva.ipynb X
2020318_Ojasva.ipynb > for item in [4, 6, 8, 10]:
+ Code + Markdown | Run All Restart Clear All Outputs Variables Outline ...
print("MAE on train:", maetrain, end=", ")
print("MAE on test:", maetest)

[53] ✓ 0.1s

... Train MSE: 8.993128124027252e+28, Test MSE: 8.477670376402939e+28

Train RMSE: 299885446863085.56, Test RMSE: 291164393022274.25

Train R2 Score: -2.632817281998673e+25, Test R2 Score: -2.455956900562937e+25
Adjusted R2 Score (Train): -2.652585875924943e+25, Adjusted R2 Score (Test): -2.4743975325808154e+25

MAE on train: 262533094739336.28, MAE on test: 252334021371567.16
```

From the results obtained and compared to those obtained from the C part, we can conclude that the results obtained for both the training and testing data are terrible for the SGDRegressor.

This is due to the fact that no standardisation was done since the dataset contains a lot of different scales and the dataset is huge. Also, the high complexity adds to observing terrible results for the errors and r-square.