MACHINE LEARNING ASSIGNMENT - 1 REPORT

Swati Sharma

Section - A

1. A.
   - No, if two variables are strongly correlated with a third variable doesn't imply that they will also display a high degree of correlation with each other.
   - This correlation is called 'spurious correlation' or 'confounding'. A confounding variable is one that makes two variables look in a way that they are connected to each other, i.e. it produces a spurious correlation between two variables.
   - e.g.: We're looking at a dataset which consists of student grades, number of study hours and the distance of the school from home. We find that both study hours and distance from school are strongly connected to better grades. But when we compare study hours and distance, we notice that students who live farther aren't necessarily studying more or less than those who live closer.
   - This shows that even though they both affect grades, study hours and distance aren't directly connected to each other.

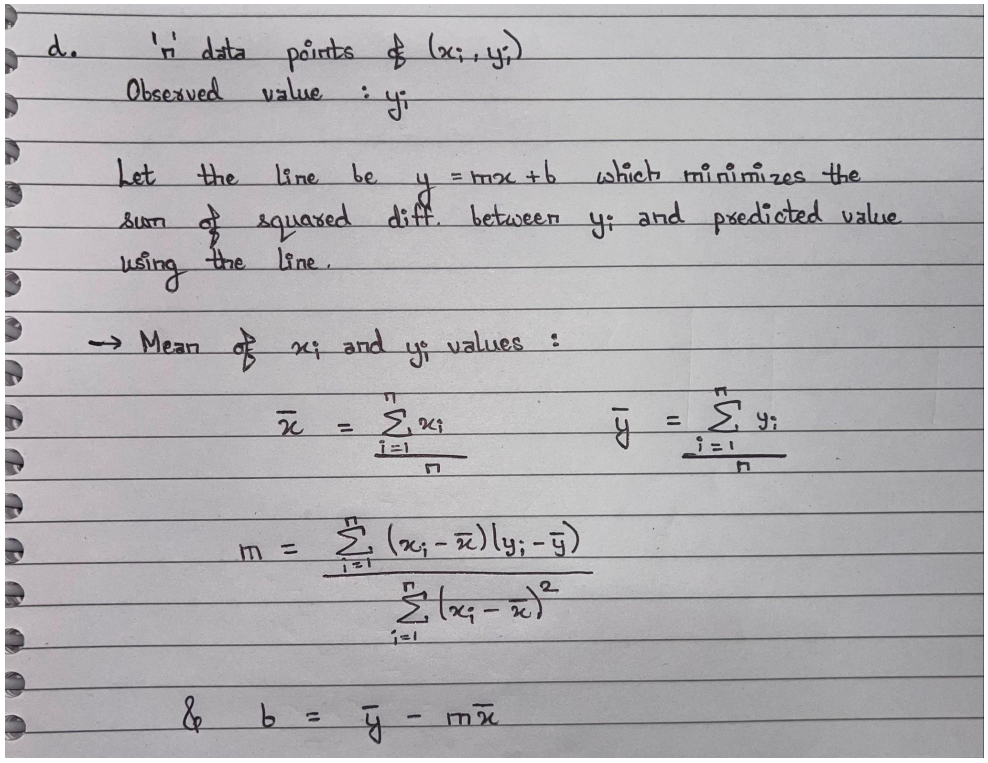
   B.
   - A function can be categorized as a logistic function if the curve of the function is a 'S-shaped' curve, i.e. it starts gently, steepens in the middle, and level off at the end.
     a. sinh(x) : not a logistic function
        : Its graph doesn't exhibit a S-shaped curve. It's a hyperbolic sine curve and rises more rapidly without leveling off, which is why it doesn't fit criteria for a logistic function
     b. cosh(x): not a logistic function
        : Its graph doesn't exhibit a S-shaped curve. It's a hyperbolic sine curve and rises more rapidly without leveling off, which is why it doesn't fit criteria for a logistic function
     c. tanh(x): logistic function
        : tanh(x) can be written as $(e^x - e^{-x})/(e^x + e^{-x})$ and sigmoid can be written as $1/(1+ e^x)$
        : Therefore, we can write $\tanh(x) = 2(\sigma(2x)) - 1$ and it can be said that tanh(x) can be made by rescaling the sigmoid function (logistic).
     d. signum(x): not a logistic function
        : Its graph doesn't exhibit a S-shaped curve and it doesn't produces continuous values between 0 and 1.

C.

- Leave-One-Out-Cross-Validation technique (LOOCV) can be used for validating very sparse datasets. It is a type of cross validation technique in which each sample is used as a validation set and other (n-1) samples are considered as the training set.
- ADVANTAGES:
    - It uses the full dataset available in each iteration which is beneficial specially for sparse datasets.
    - It has no randomization of using some samples in training and validation dataset and so there is less bias.
- Differences between K-fold cross validation and LOOCV:

K-Fold Cross Validation	LOOCV
Number of folds is equal to k which can range from 1 to n.	Number of folds is always equal to n, i.e. size of the dataset.
Can be less expensive for large datasets	Computationally expensive for large datasets
Provides some bias as each validation set contains (k-1)*n/k observations which is less than LOOCV but more than validation set method	Provides unbiased test estimates
Utilize a fraction of the dataset in each iteration	Maximizes data utilization as it uses all the data samples for training

D.



d. 'n' data points of (x_i, y_i)

Observed value : y_i

Let the line be $y = mx + b$ which minimizes the sum of squared diff. between y_i and predicted value using the line.

→ Mean of x_i and y_i values :

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} \qquad \bar{y} = \frac{\sum_{i=1}^{n} y_i}{n}$$

$$m = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

& $b = \bar{y} - m\bar{x}$

E. (a). α, β, σ

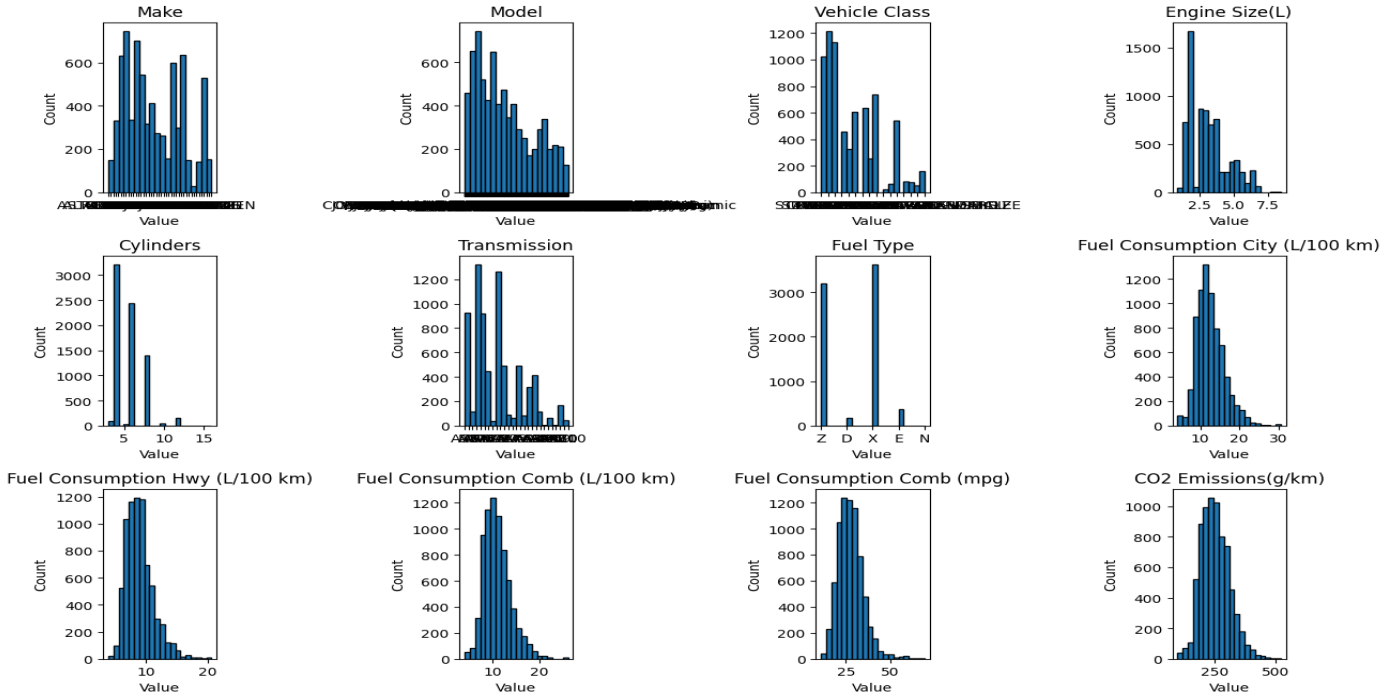
- α : y-intercept of the line
- β : slope of the line
- σ : standard deviation of error ε which measures the noise or randomness in the Y-values

F. (d). $Y = \alpha + \beta 1x + \beta 2x2 + \varepsilon$ $\beta 2 > 0$

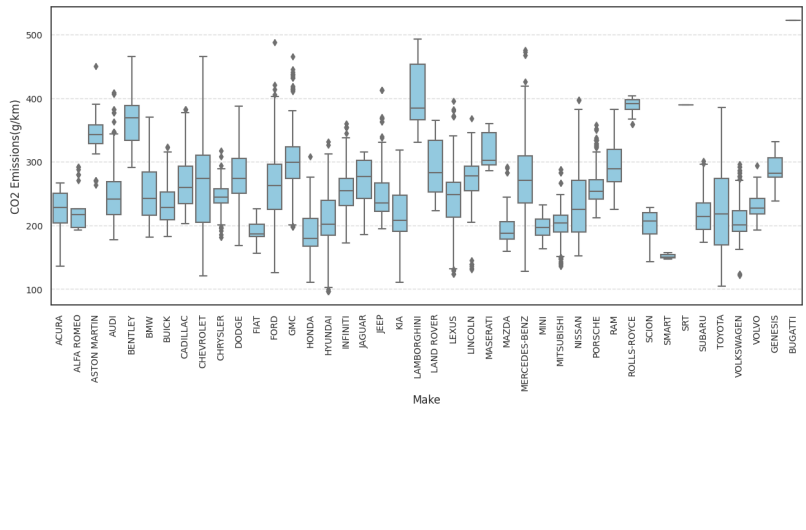
- Plotting the given points, we can see that the shape of the graph is an upward-facing U-curve so the target variable is quadratically dependent on the X-variable with value of Beta-2 > 0 as it is upward facing.

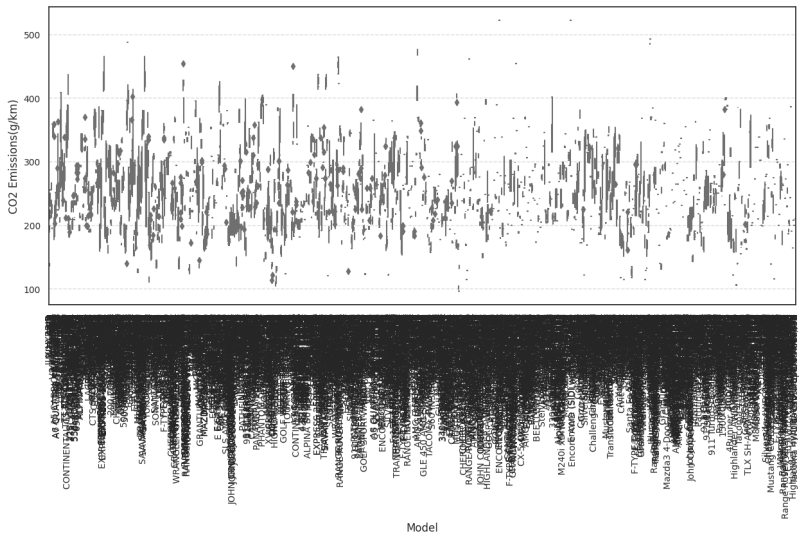
Section - C

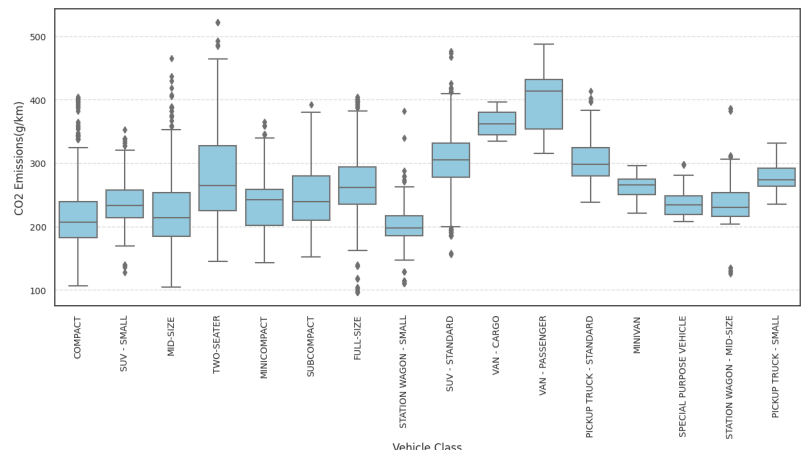
3.A. 1. Histogram:

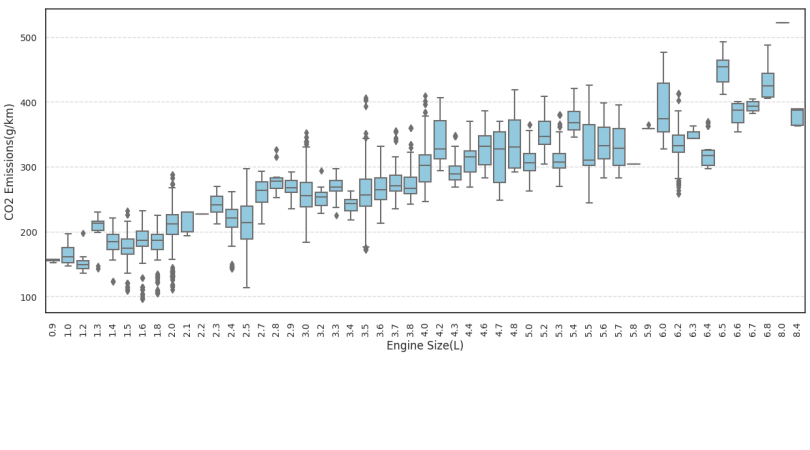

Make
Model
Vehicle Class
Engine Size(L)
Cylinders
Transmission
Fuel Type
Fuel Consumption City (L/100 km)
Fuel Consumption Hwy (L/100 km)
Fuel Consumption Comb (L/100 km)
Fuel Consumption Comb (mpg)
CO2 Emissions(g/km)
Count
Value

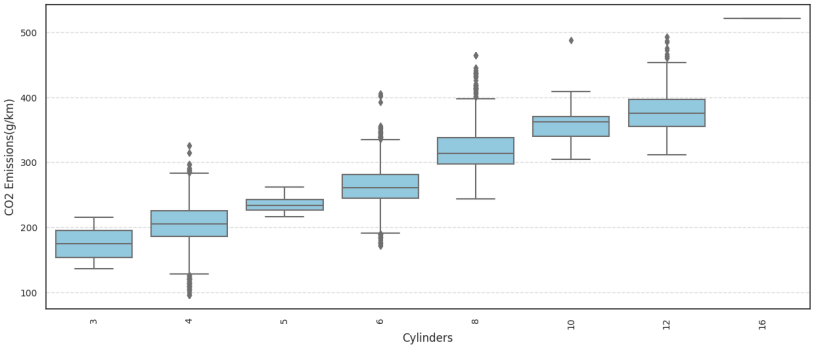
2. Box Plot:

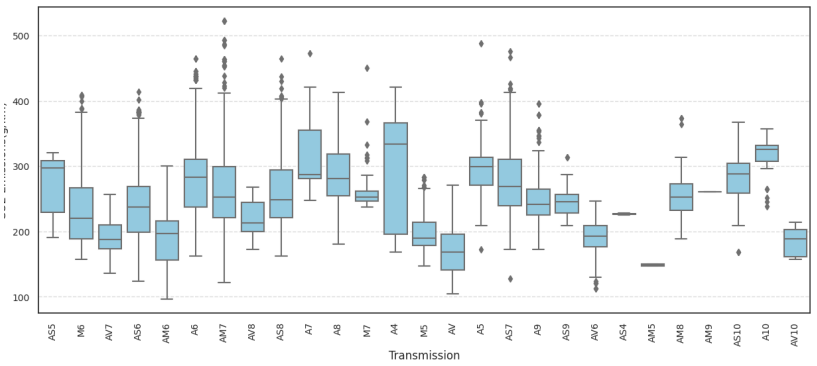

CO2 Emissions(g/km)
Make

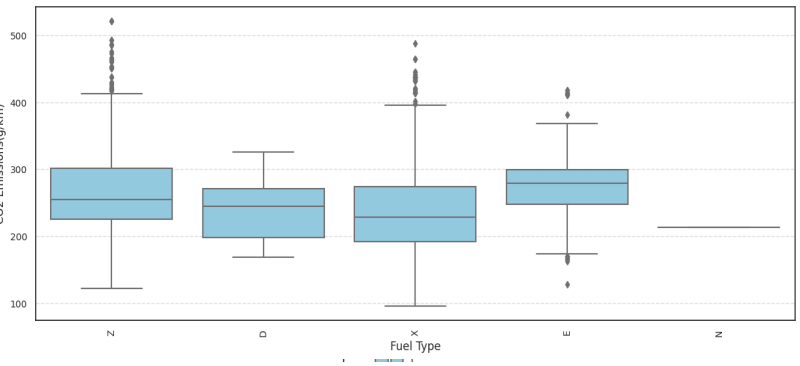

CO2 Emissions(g/km)
Model

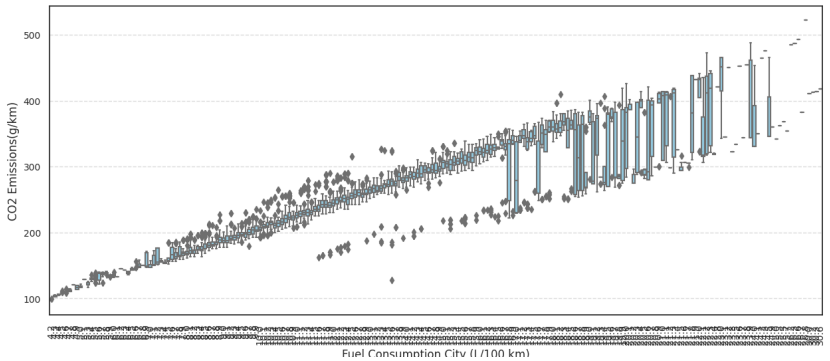

CO2 Emissions(g/km)
Vehicle Class

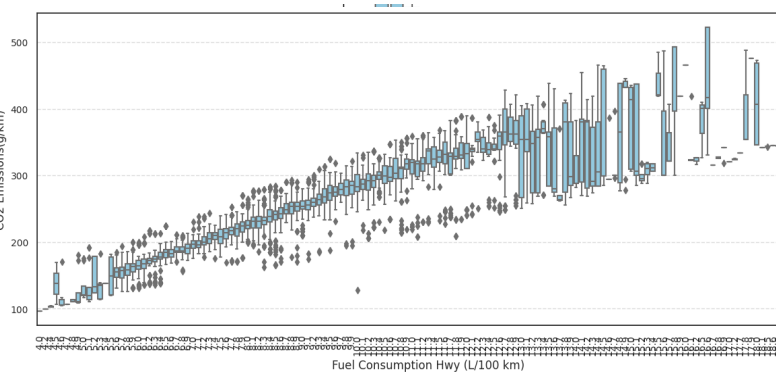

CO2 Emissions(g/km)
Engine Size(L)

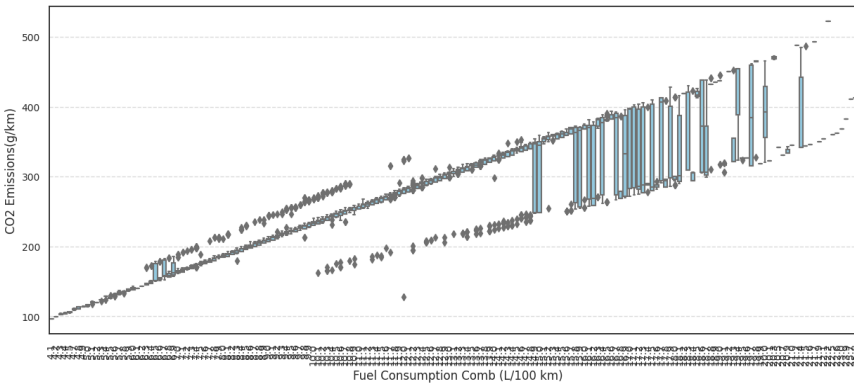




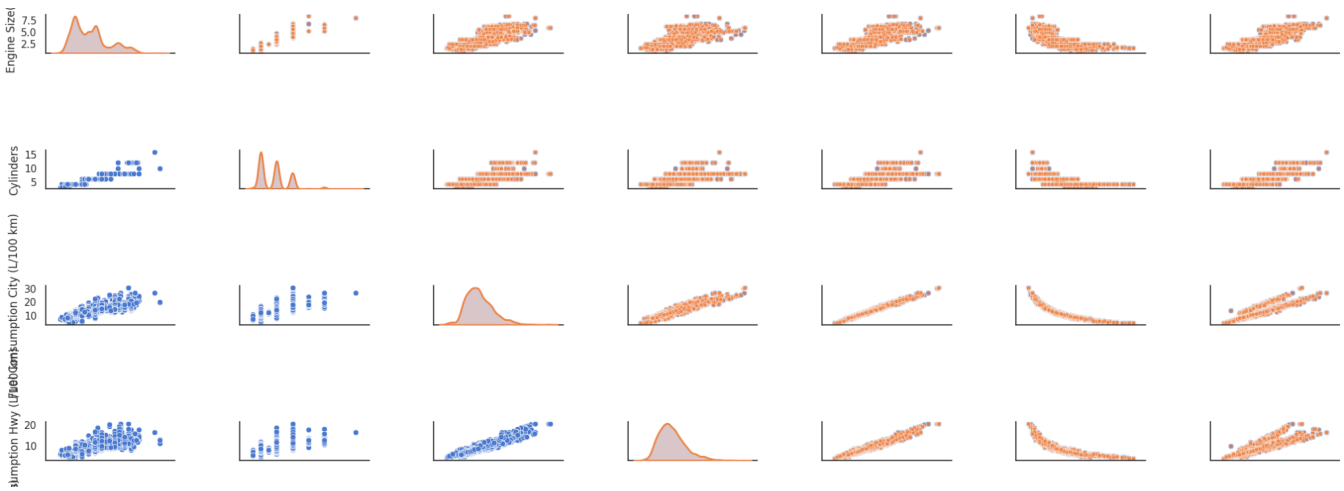


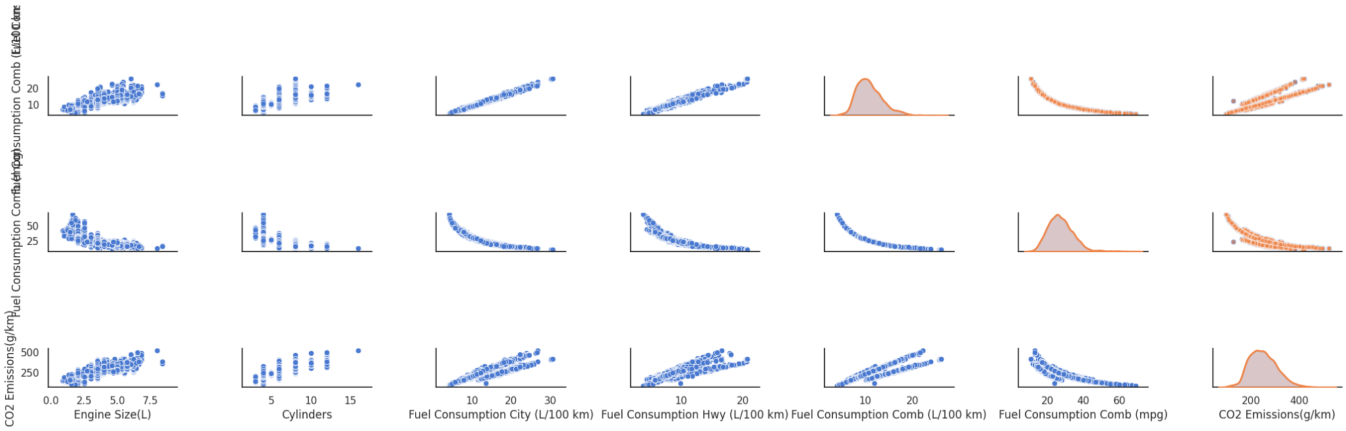




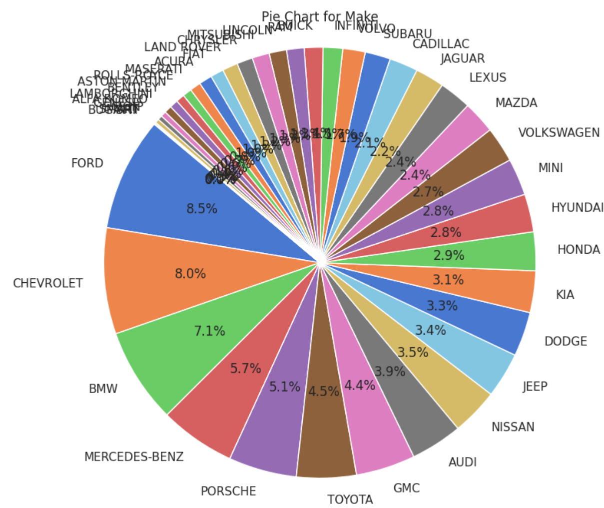


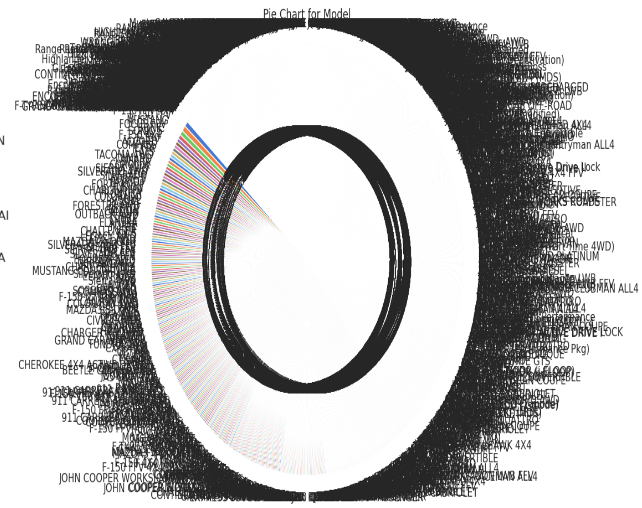
3. Scatter Plot:

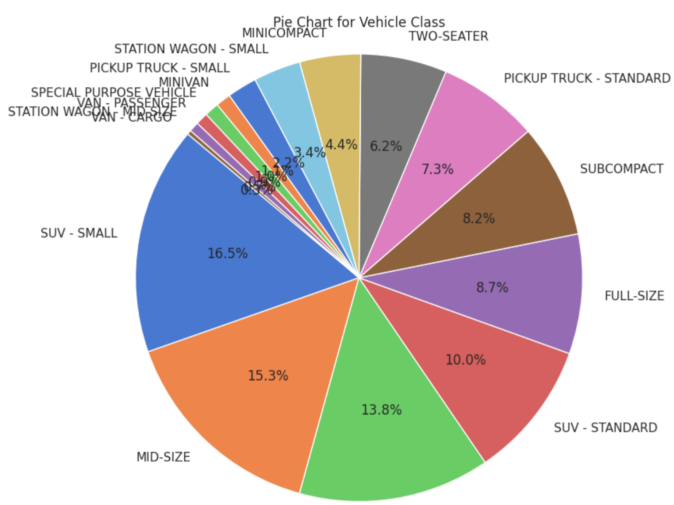


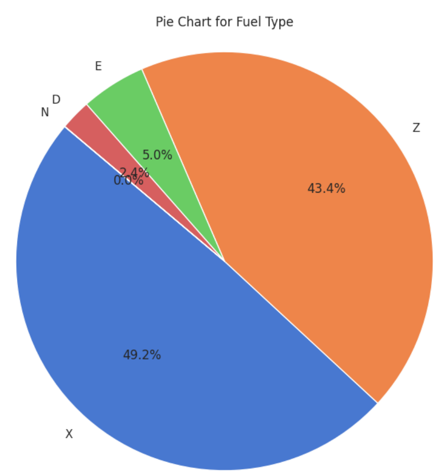

Fuel Consumption Comb (L/100 km)
Fuel Consumption Comb (mpg)
CO2 Emissions(g/km)
Engine Size(L)
Cylinders
Fuel Consumption City (L/100 km)
Fuel Consumption Hwy (L/100 km)
Fuel Consumption Comb (L/100 km)
Fuel Consumption Comb (mpg)
CO2 Emissions(g/km)

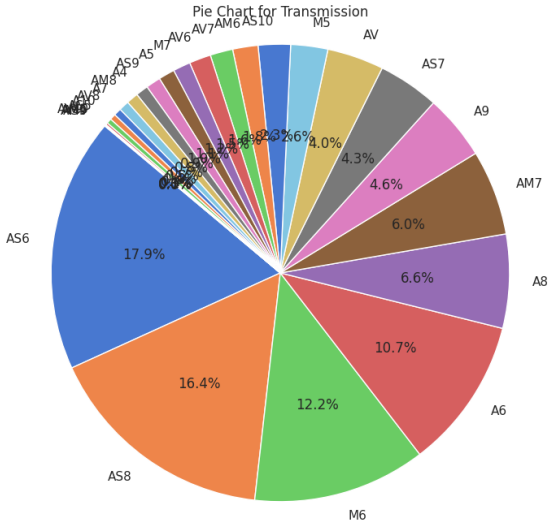
4. Pie Chart


Pie Chart for Make
FORD 8.5%
CHEVROLET 8.0%
BMW 7.1%
MERCEDES-BENZ 5.7%
PORSCHE 5.1%
TOYOTA 4.5%
GMC 4.4%
AUDI 3.9%
NISSAN 3.5%
JEEP 3.4%
DODGE 3.3%
KIA 3.1%
HONDA 2.9%
HYUNDAI 2.8%
MINI 2.8%
VOLKSWAGEN 2.7%
MAZDA 2.4%
LEXUS 2.4%
JAGUAR 2.2%
CADILLAC
SUBARU
VOLVO
INFINITI
BUICK
RAM
LINCOLN
MITSUBISHI
CHRYSLER
LAND ROVER
FIAT
ACURA
MASERATI
ROLLS-ROYCE
ASTON MARTIN
BENTLEY
LAMBORGHINI
ALFA ROMEO

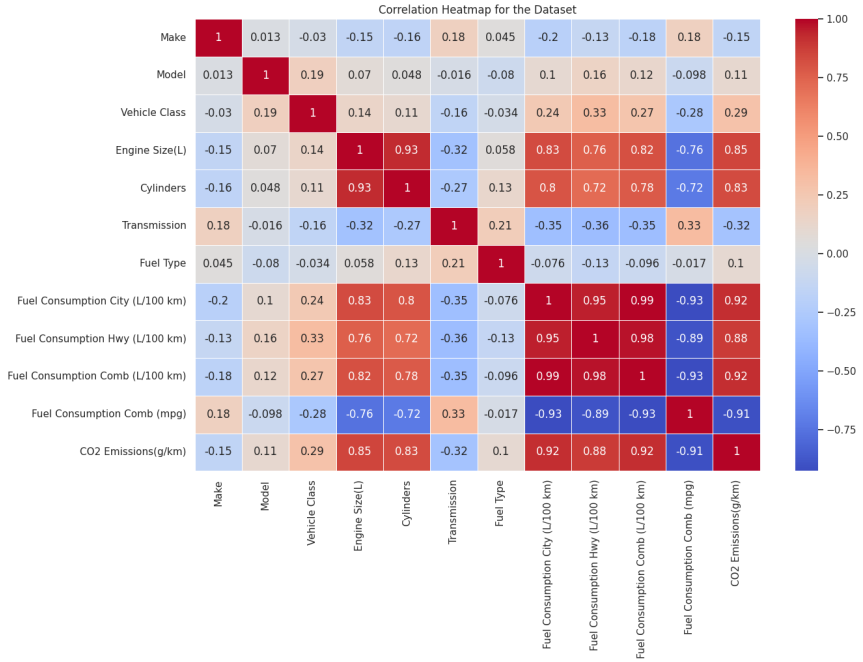

Pie Chart for Model


Pie Chart for Vehicle Class
SUV - SMALL 16.5%
MID-SIZE 15.3%
COMPACT 13.8%
SUV - STANDARD 10.0%
FULL-SIZE 8.7%
SUBCOMPACT 8.2%
PICKUP TRUCK - STANDARD 7.3%
TWO-SEATER 6.2%
MINICOMPACT 4.4%
STATION WAGON - SMALL 3.4%
PICKUP TRUCK - SMALL 2.2%
MINIVAN
SPECIAL PURPOSE VEHICLE
VAN - PASSENGER
STATION WAGON - MID-SIZE
VAN - CARGO

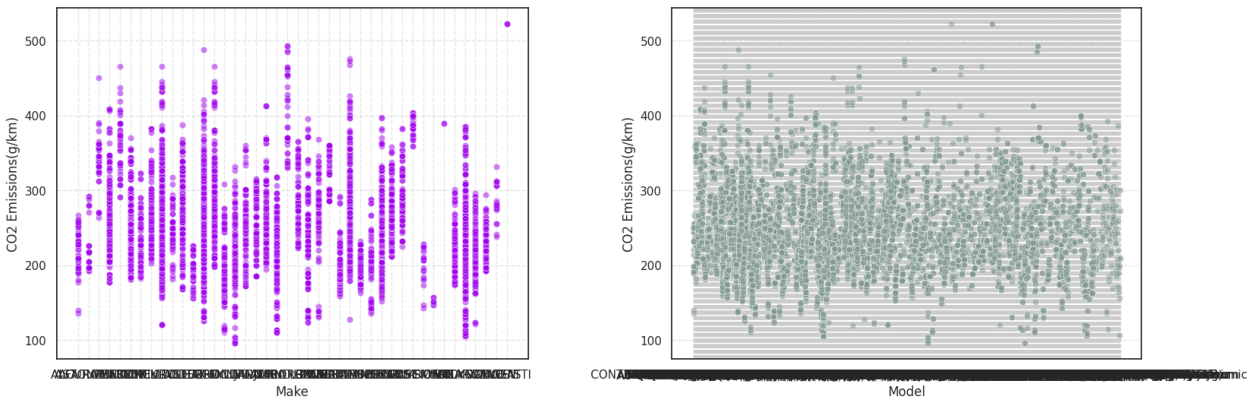

Pie Chart for Fuel Type
X 49.2%
Z 43.4%
E 5.0%
D 2.4%
N 0.0%

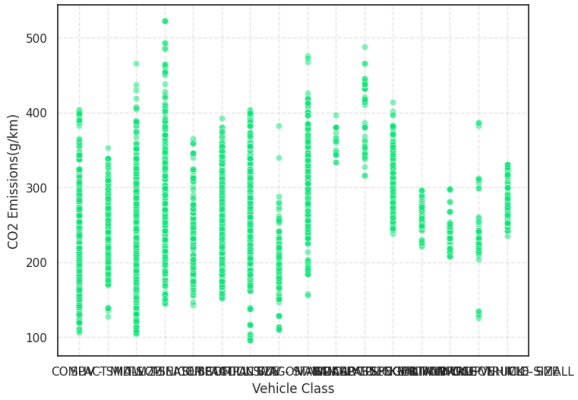


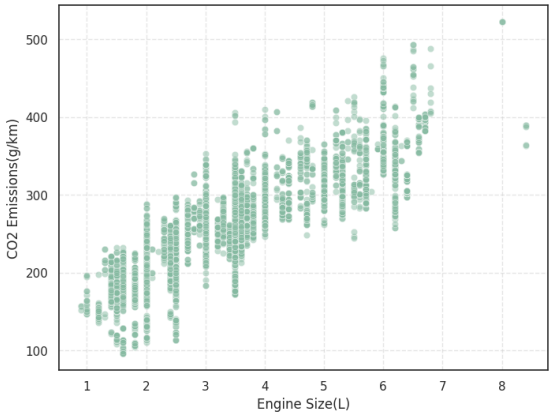
5. Correlation Heatmap:

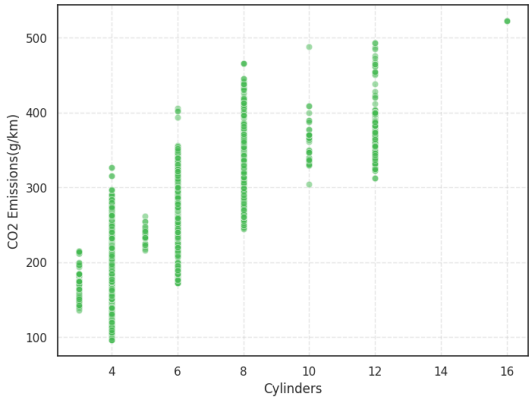


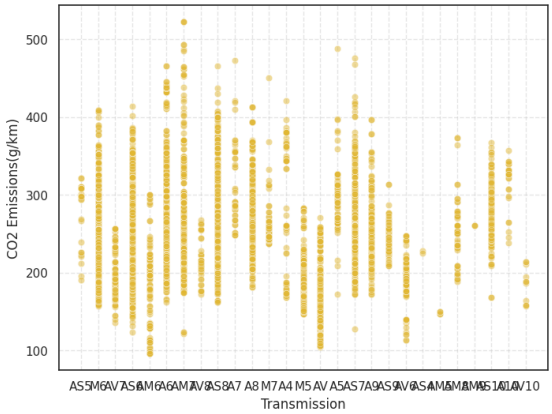
6. Scatter Plots:

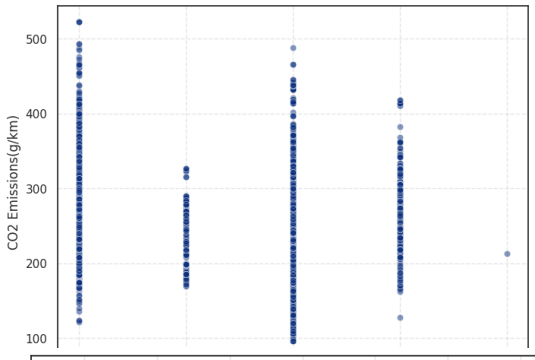


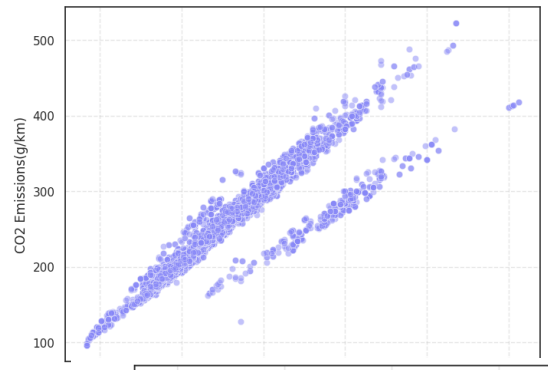

CO2 Emissions(g/km)
Vehicle Class

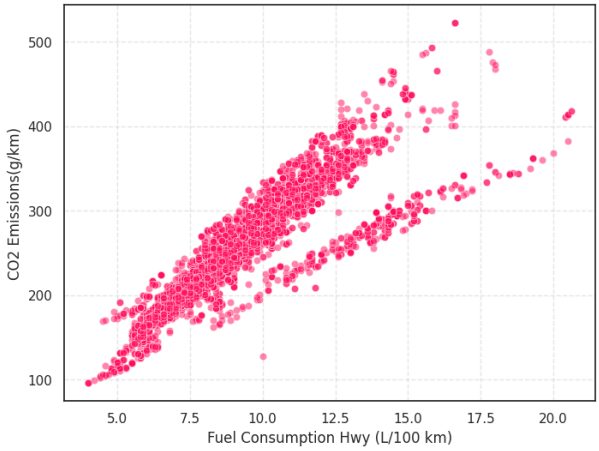

CO2 Emissions(g/km)
Engine Size(L)

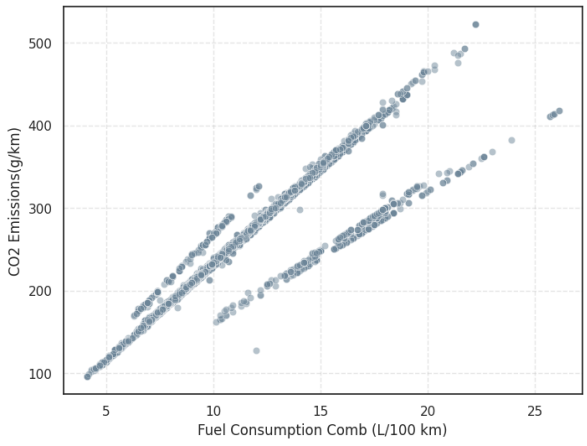

CO2 Emissions(g/km)
Cylinders

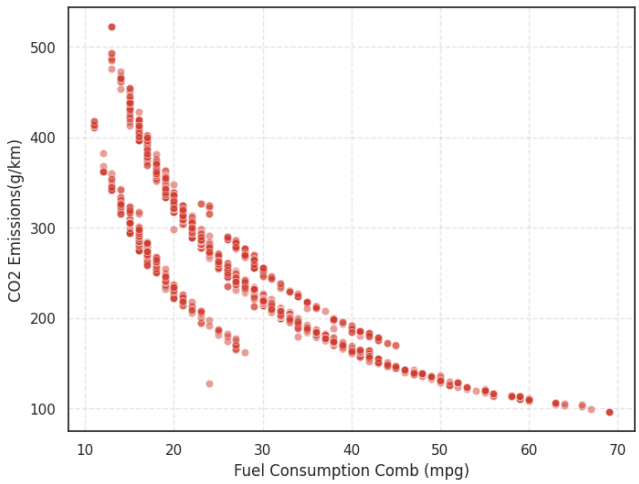

CO2 Emissions(g/km)
Transmission


CO2 Emissions(g/km)


CO2 Emissions(g/km)


CO2 Emissions(g/km)
Fuel Consumption Hwy (L/100 km)


CO2 Emissions(g/km)
Fuel Consumption Comb (L/100 km)



Insights of the dataset:

a. A linear distribution can be seen from the box plot of Fuel Consumption City (L/100km), Fuel Consumption Hwy (L/100km), Fuel Consumption Comb (L/100km) which suggests that there can be a linear relationship between these features and CO2 emissions.
b. Logarithmic distribution can be seen from the box plot of Fuel Consumption Comb (mpg) which suggests that there can be a logarithmic relationship between these features and CO2 emissions.
c. From the correlation heatmap it can be concluded that there's a strong positive relationship between Fuel Consumption City (L/100km) and Fuel Consumption Hwy (L/100km); Fuel Consumption City (L/100km) and Fuel Consumption Comb (L/100km); Fuel Consumption Hwy (L/100km) and Fuel Consumption Comb (L/100km) and Fuel Consumption City (L/100km) and Fuel Consumption Comb (L/100km) as there values are closer to 0.99.
d. A linear relationship between Fuel Consumption City (L/100km), Fuel Consumption Hwy (L/100km), Fuel Consumption Comb (L/100km) and CO2 emission can be seen again in the scatter plots.
e. Scatter plot of Make, Vehicle Class, Cylinders, Transmission, etc. consists of vertical lines suggesting that each category or discrete value of the feature has a constant value for the target variable.
f. Scatter plot of the 2-D t-SNE representation suggests that data points with different features are closely clustered together which suggests that there isn't strong separability based on the original features in lower dimensional space.

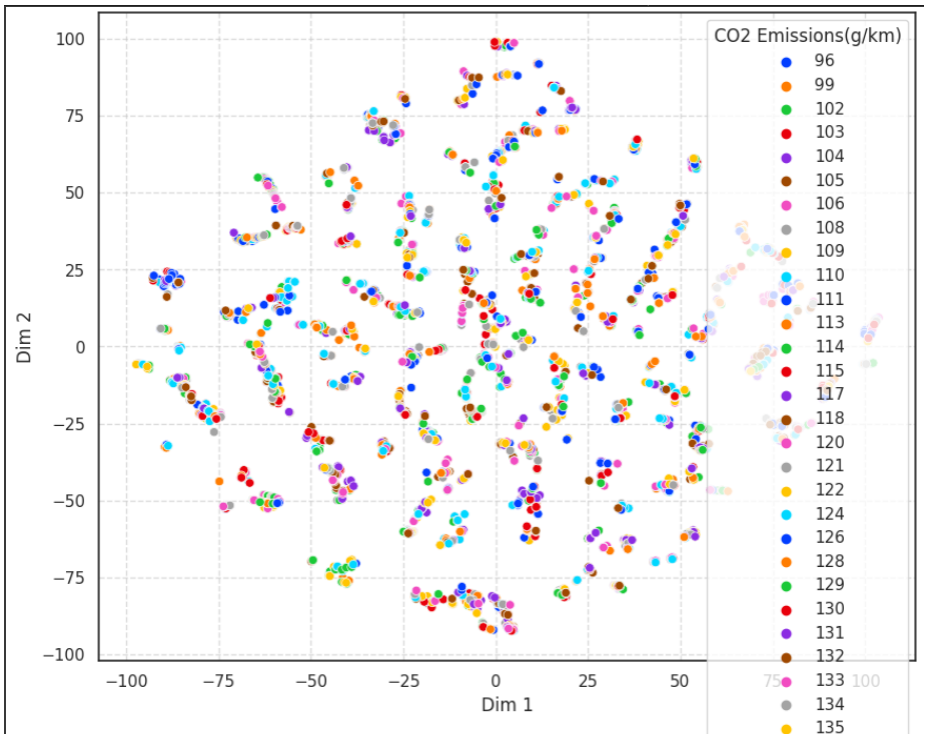
Section - C

3.B.

1. t-SNE was implemented using the following code:

```
tsne = TSNE(n_components=2, random_state=42)
X_2d = tsne.fit_transform(X)
```

2. It can be seen from the graph generated that data points with different features are closely clustered together which suggests that there isn't strong separability based on the original features in lower dimensional space.
3. t-SNE isn't able to effectively preserve the original feature relationships.



Section - C

3.C. Pre-processing steps done:

1. Check for any missing values
2. Removal of duplicates
3. Normalization of data
4. Label encoding using LabelEncoder

1. Data Splitting
2. Linear Regression model

Results:

Metric	Training Data	Testing Data
MSE	274.18083247175986	307.86968906440933
RMSE	16.55840670088037	17.5462158046802
R2 Score	0.9139939145584578	0.9071910099024745
Adjusted R2 Score	0.9138155459800534	0.9064164280427834
MAE	10.73776441898862	11.003910664669611

Section - C

3.D. Results after applying PCA on the original dataset with label encoding:

	Components	MSE Train	MSE Test	RMSE Train	RMSE Test	R2 Train	\
0	4.0	451.101198	460.765023	21.239143	21.465438	0.868043	
1	6.0	372.518647	380.730634	19.300742	19.512320	0.891030	
2	8.0	288.444771	298.343949	16.983662	17.272636	0.915624	
3	10.0	286.256673	295.978894	16.919122	17.204037	0.916264	

	R2 Test	Adjusted R2 Train	Adjusted R2 Test	MAE Train	MAE Test
0	0.866042	0.867954	0.865678	13.634166	13.801328
1	0.889311	0.890919	0.888859	11.038369	11.162797
2	0.913263	0.915509	0.912790	11.017948	11.324676
3	0.913950	0.916122	0.913363	10.960490	11.212906

Analysis:

1. Increasing the number of components is leading to lower MSE, RMSE and MAE on both training and testing datasets. However, after a certain number of components, the improvement is marginal.
2. The model will become more complex as we will start increasing the number of components. So, we will have to take this as an input in making the final decision.
3. MSE, RMSE and MAE values are almost the same for both testing and training dataset.

Section - C

3.E. Results after applying One-Hot Encoding and Linear Regression on the original dataset:

Metric	Training Data	Testing Data
MSE	8.534818957381319	6.221142204925232e+19
RMSE	2.921441246607797	7887421761.846663
R2 Score	0.9975033811381525	-1.8086674674817508e+16
Adjusted R2 Score	0.9960756978134824	3.9667060653834536e+16
MAE	1.8703274168351574	1551013140.415223

Analysis:

1. It appears that LabelEncoder provides a more effective encoding strategy for the dataset as compared to previous models with One-Hot Encoding as there is a significant difference in the MSE, RMSE and R2 scores for both the methods.

Section - C

3.F. Results after applying PCA on the original dataset with One-Hot encoding:

	Components	MSE Train	MSE Test	RMSE Train	RMSE Test	R2 Train
0	4.0	335.100744	340.675555	18.305757	18.457398	0.901976
1	5.0	332.296609	337.894311	18.229005	18.381902	0.902796
2	6.0	325.924084	331.999090	18.053368	18.220842	0.904660
3	8.0	309.941251	322.081968	17.605148	17.946642	0.909335
4	10.0	308.584407	320.986894	17.566571	17.916107	0.909732

	R2 Test	Adjusted R2 Train	Adjusted R2 Test	MAE Train	MAE Test
0	0.900956	0.901909	0.900687	11.950279	11.870733
1	0.901764	0.902714	0.901430	11.906866	11.833885
2	0.903478	0.904563	0.903084	11.970890	11.856524
3	0.906361	0.909213	0.905851	11.531490	11.734342
4	0.906680	0.909579	0.906043	11.500865	11.704957

Analysis:

1. Increasing the number of components is leading to lower MSE, RMSE and MAE on both training and testing datasets. However, after a certain number of components, the improvement is marginal.
2. The model will become more complex as we will start increasing the number of components. So, we will have to take this as an input in making the final decision.

Section - C

3.G. Results after applying L1 and L2 regularization while training the linear model:

```
Training Data:
MSE: 285.8391473834993
RMSE: 16.90677814911816
R2 Score: 0.9163858764461709
MAE: 10.96561560552965

Testing Data:
MSE: 296.03340291570106
RMSE: 17.205621259219356
R2 Score: 0.9139344564865189
MAE: 11.196485397785933
```

```
Training Data:
MSE: 285.75176878184664
RMSE: 16.90419382229885
R2 Score: 0.9164114365741709
MAE: 10.962126267804814

Testing Data:
MSE: 295.80267386628753
RMSE: 17.198914903745745
R2 Score: 0.9140015361499845
MAE: 11.190160930794667
```

Analysis:

1. There is no significant difference in the results using L1 and L2 regularization so the choice can be based upon other factors such as features, etc.

Section - C

3.H. Results after applying SGD Regressor to perform the linear regression on the preprocessed dataset:

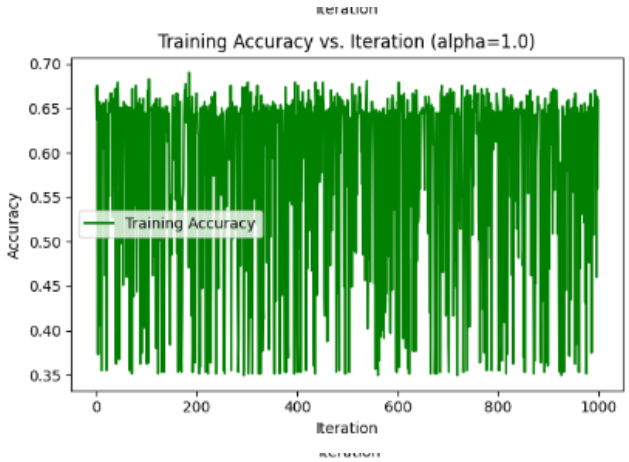
```
Training Data:
MSE: 3.246659288905697e+29
RMSE: 569794637470878.4
R2 Score: -9.497179564264925e+25
MAE: 495164485125612.56

Testing Data:
MSE: 3.252318851217157e+29
RMSE: 570291052991115.9
R2 Score: -9.45543938123946e+25
MAE: 496581529004706.94
```

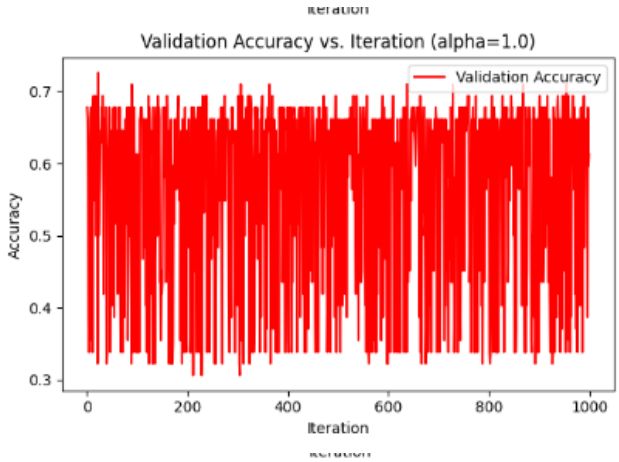
Analysis:

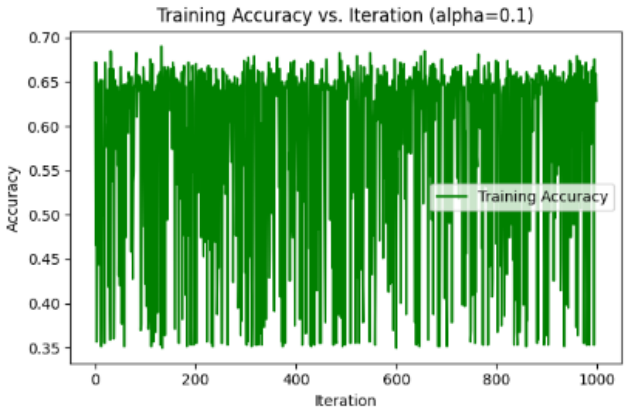
1. The values of MSE and RMSE are very large indicating a poor fit model.
2. The MAE values are extremely large indicating that the model's values are very different from the true values

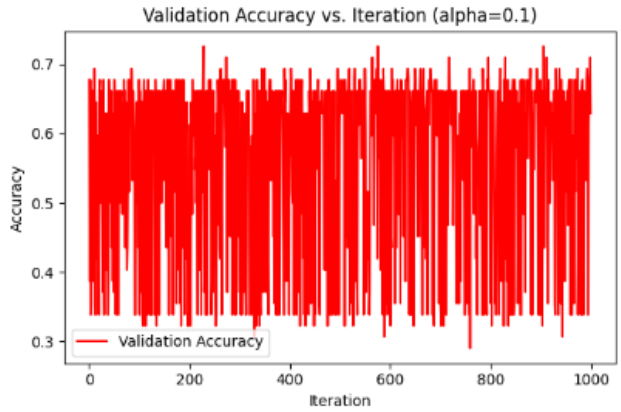
Section - B

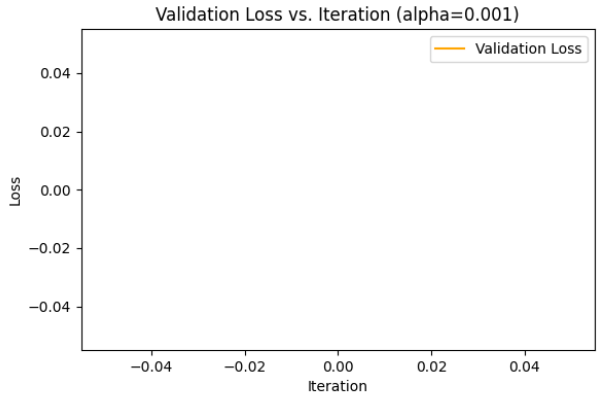
2.A. Results after applying SGD Regressor to perform the linear regression on the preprocessed dataset:

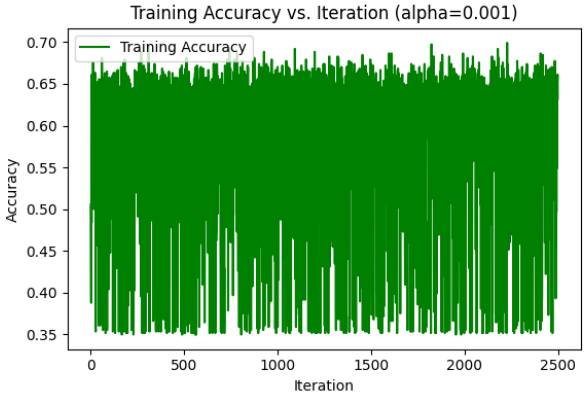


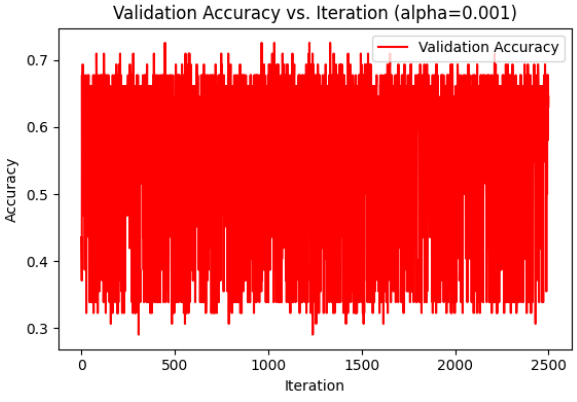












Alpha	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Confusion Matrix**
1	0.616883	0.545455	0.4	0.427184	[[73 26] [33 22]]
0.1	0.681818	0.5625	0.490909	0.524272	[[78 21] [28 27]]
0.01	0.707792	0.571429	0.727273	0.64	[[69 30] [15 40]]
0.001	0.675325	0.6	0.0545455	0.418605	[[86 13] [37 18]]

1. Accuracy: The accuracy values range from approximately 0.616 to 0.707. The highest accuracy is achieved at Alpha = 0.01. Model's performance is highly sensitive to the alpha values so it should be taken into consideration.
2. Precision: Precision values range from 0.545 to 0.6. An Alpha value of 0.001 yields the highest precision.
3. Recall: Recall values vary from 0.0545 to 0.727. Alpha = 0.01 results in the highest recall.
4. F1 Score: The F1 score, which balances precision and recall, ranges from 0.418 to 0.64. An Alpha value of 0.01 yields the highest F1 score.

- There needs to be a tradeoff between Precision, Recall and F1 score as when one of the values is getting improved the other is fluctuating.