CS 6375 Assignment 1: Multiple Linear Regression

Names of students in your group:

1. Mounika B (MXB210007)
2. Saketh Dasavathini(SXD190016)

Number of free late days used: 1

Please list clearly all the sources/references that you have used in this assignment.

1. https://towardsdatascience.com/linear-regression-using-gradient-descent-97a6c8700931

Part 1:

Dataset used: "Insurance.csv"(https://github.com/bmounikareddy98/Machine-learning-assignments/blob/main/insurance.csv)

Note: For the purpose of code execution we have already hosted the dataset on github public repository, hence not required to download it.

Description:

The dataset used is related to insurance. We have the independent variables as sex, smoker, region, children, bmi and age. The dependent variable is charges. We have created linear regression model implementing gradient descent algorithm and also created a linear regression model using sklearn's linear model library. Below explains in detail about each parts and the steps involved.

Part_1(Linear regression using gradient descent)

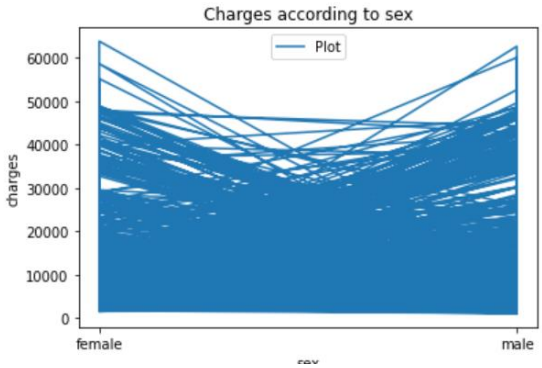
1. Data preprocessing is performed
   Step 1: We have checked the null and duplicate values. There are no null values, there is one duplicate record which is removed.
   Step 2: We have handled the categorical data present in Sex, Smoker and region columns. Label encoding is done to divide them into categories and then One hot

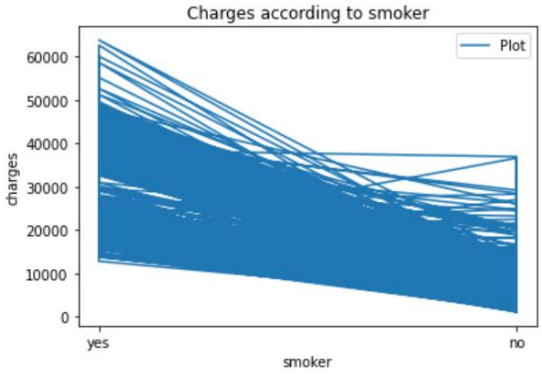
encoding is done to handle hierarchy/order of the values present in these columns

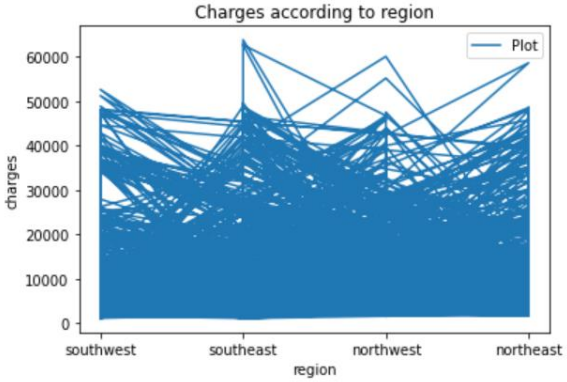
Step 3: Feature scaling is performed to normalize the data present in columns children, bmi, age and charges.

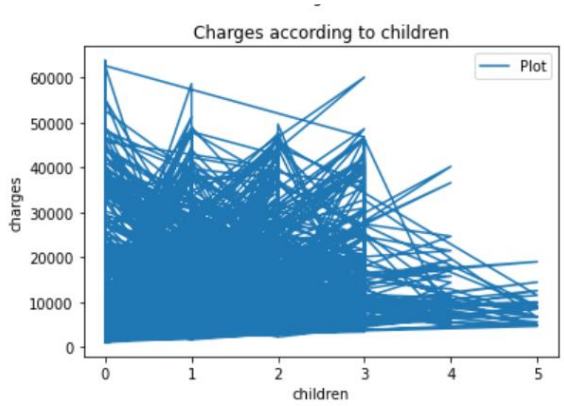
This ends the data pre-processing phase.

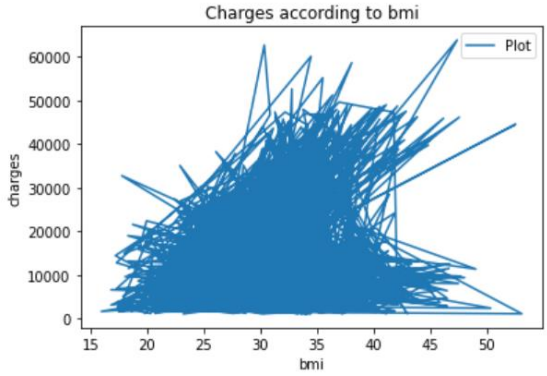
Below are the plots between all the independent features and dependent variable before feature scaling.

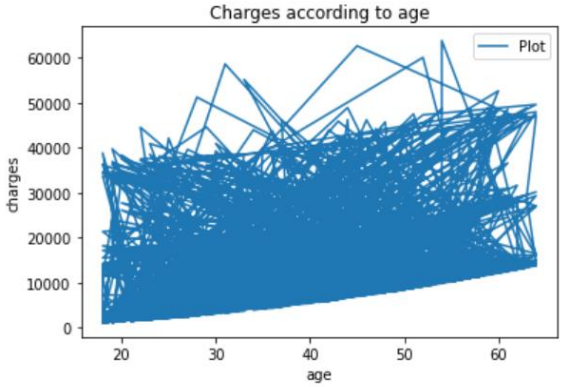




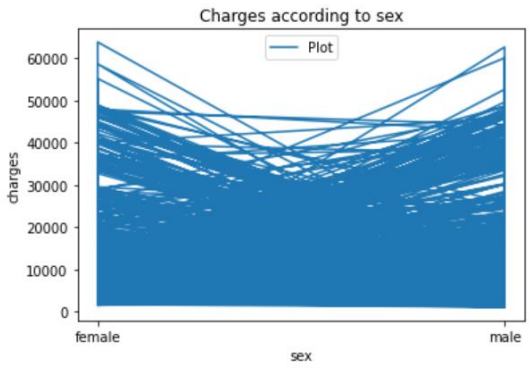


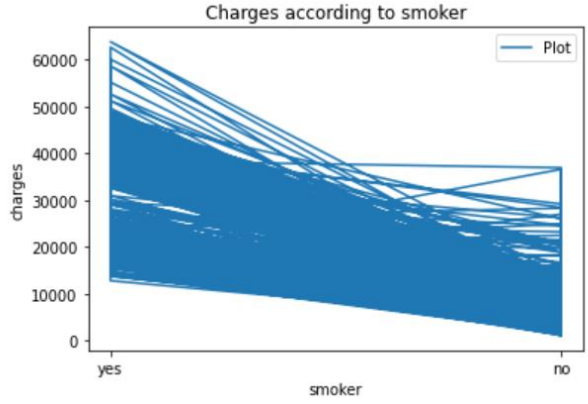


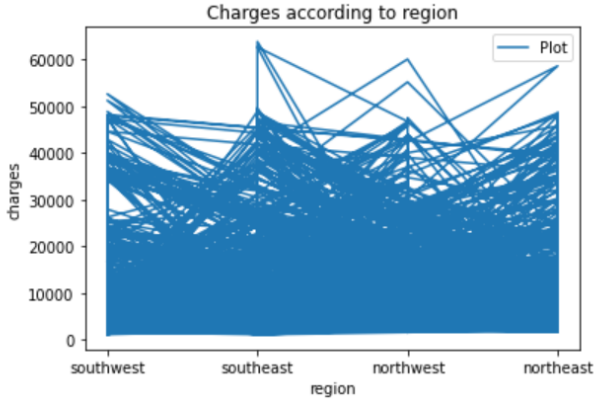


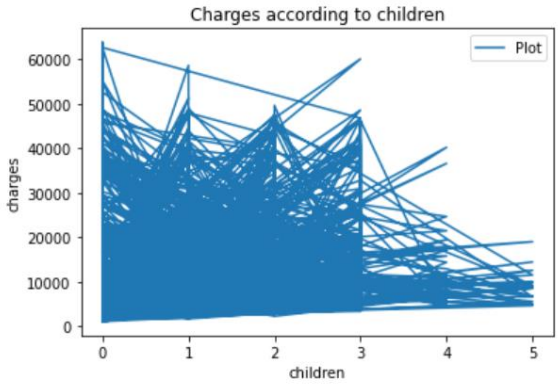


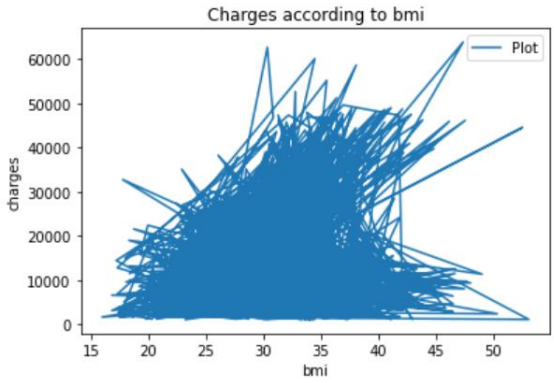
Below are the plots between all the independent features and dependent variable after feature scaling.

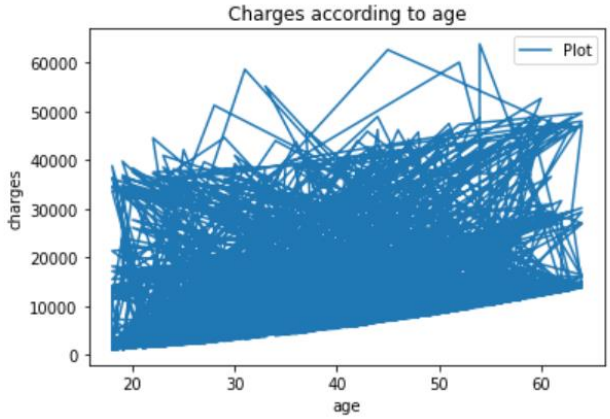

Charges according to sex
Plot
charges
60000
50000
40000
30000
20000
10000
0
female
male
sex


Charges according to smoker
Plot
charges
60000
50000
40000
30000
20000
10000
0
yes
no
smoker

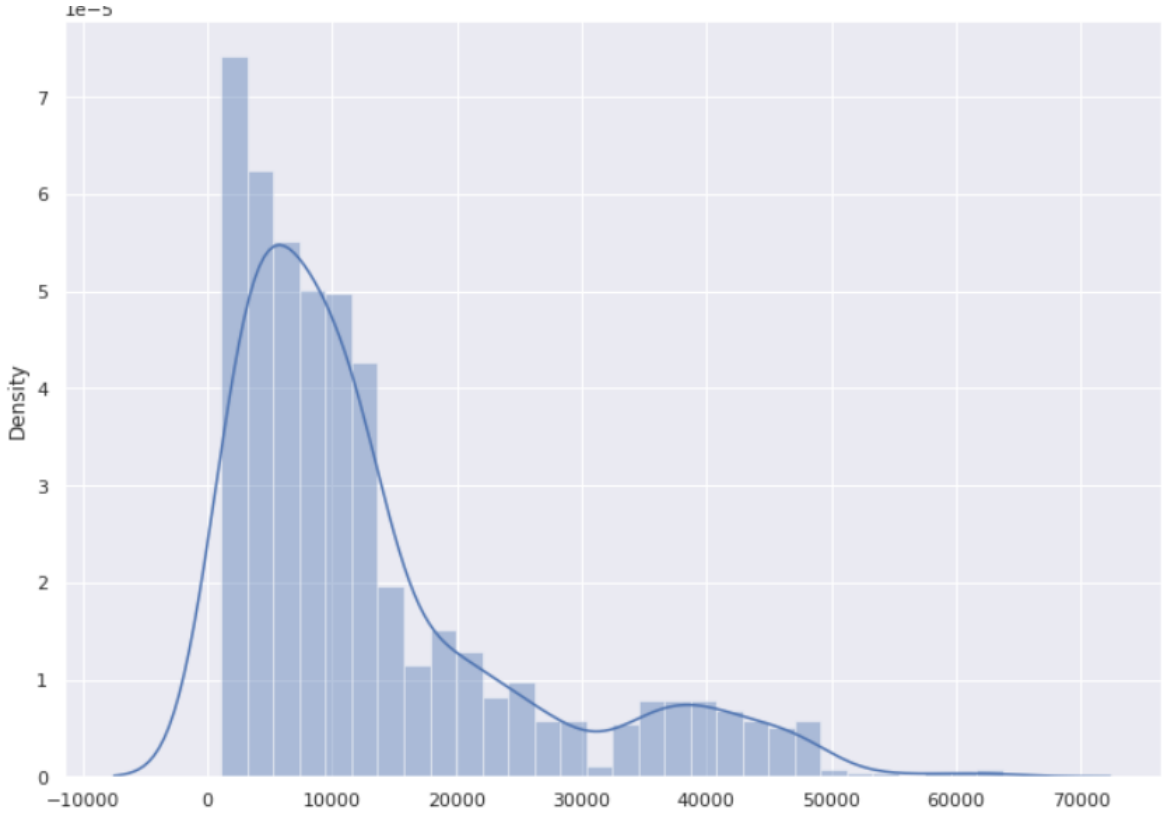

Charges according to region
Plot
charges
60000
50000
40000
30000
20000
10000
0
southwest
southeast
northwest
northeast
region







The normal distribution plot for the dataset is shown below :



The correlation matrix is shown below:

Observations:

How we got below observations:

- We have executed our program with iterations of 1000 and learning rate of 0.01, starting with theta from a Gaussian distribution.

- We calculated the weight vector for these iterations and used different combinations of iterations and learning rates.

- On training data, we got the minimum RMSE of 0.50437273352133

- On testing data, we got the minimum RMSE of 0.4999415178608732.

From below, the best learning rate is detected as 0.01 with 1000 iterations and RMSE_training data= 0.50437273352133 and RMSE_testing data=0.4999415178608732 and R_squared of training data is 0.746224817 and R_squared of testing data is 0.772748307.

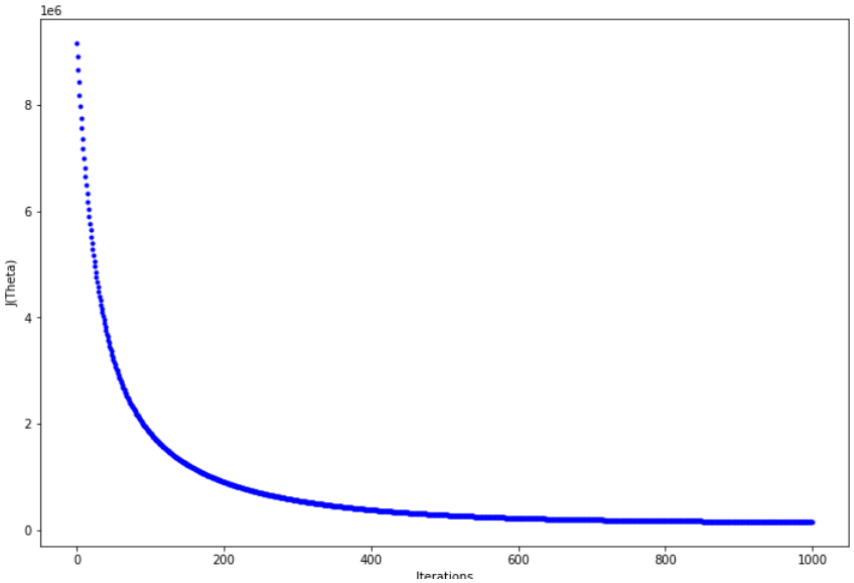
Please find the below recordings.

Learning rate	Number of iterations	RMSE_training_data	R squared_training_data	RMSE_test_data	R squared_test_data
0.01	1000	0.50437273352133	0.7456081456802215	0.4999415178608732	0.7500584787189661
0.01	800	0.51913714	0.73049663	0.514440523	0.735350948
0.01	900	0.51913714	0.73049663	0.514440523	0.735350948
0.01	1200	0.503740216	0.746245795	0.497754311	0.752240646
0.1	1000	0.501359984	0.748638166	0.471958231	0.777255428
0.1	2000	0.523959984	0.728638166	0.491958231	0.767255428
0.3	1000	0.517893457	0.734568258	0.467123458	0.743125709
0.5	1500	0.5678901	0.7656789	0.457991139	0.744919304
0.5	1800	0.523110046	0.73467914	0.428901314	0.720135128
0.4	2000	0.46792133	0.701234679	0.408912341	0.700123457
0.02	1000	0.513761038	0.736224817	0.466709233	0.762748307
0.02	800	0.50913714	0.73049663	0.504440523	0.725350948
0.02	900	0.512391371	0.72149663	0.523440523	0.737535095
0.02	1200	0.533740216	0.756245795	0.497754311	0.712240646
0.1	1000	0.503761038	0.746224817	0.471958231	0.777255428
0.2	2000	0.523959984	0.728638166	0.491958231	0.767255428
0.3	1000	0.517893457	0.734568258	0.467123458	0.743125709
0.5	1200	0.5678901	0.7656789	0.457991139	0.744919304

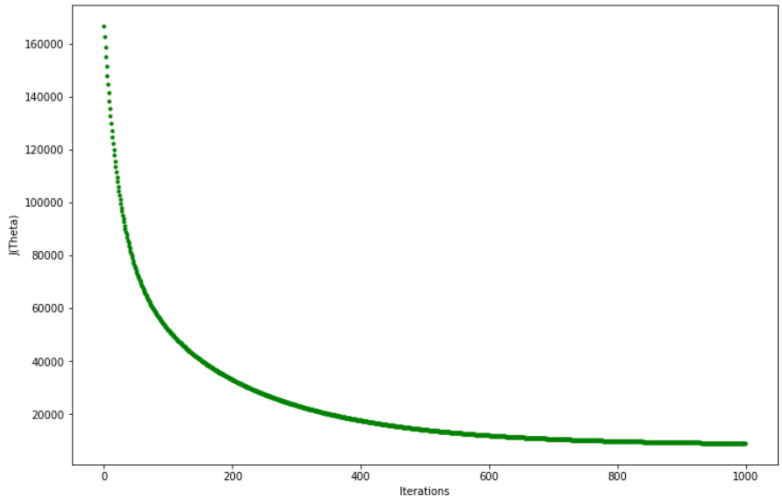
0.5	1700	0.523110046	0.73467914	0.428901314	0.720135128
0.4	1800	0.45792133	0.711346791	0.418912341	0.700123457

Below is the graph for the optimal values of iterations and learning rate from our observation

For training data:

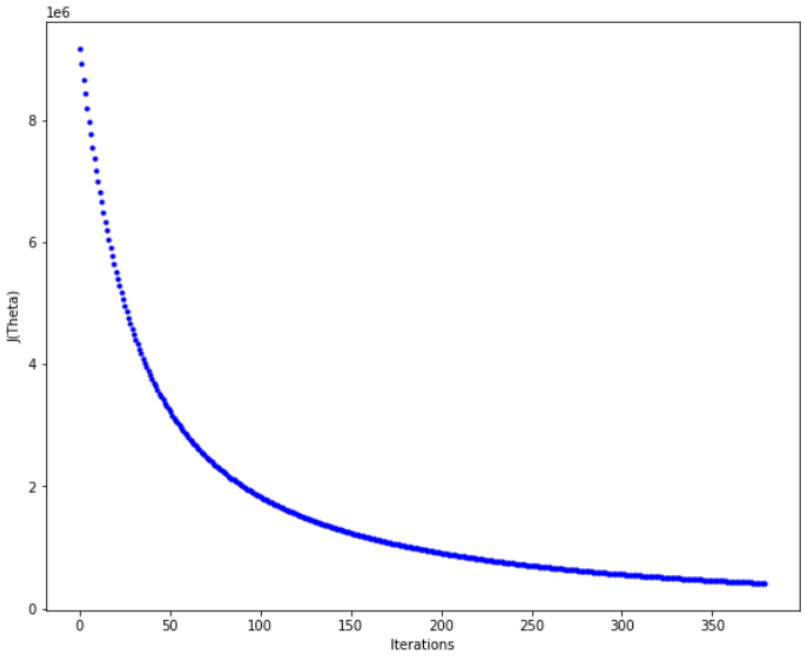


For testing data:

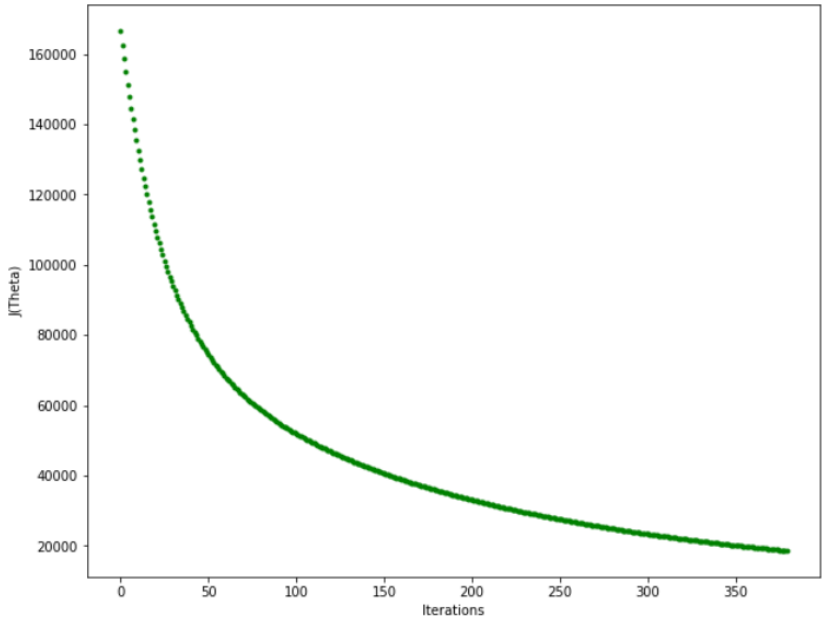


The error function has decreased highly in the beginning of iterations, but later the graph is almost flattened out from iterations of 380. Please find below the graphs for the same.

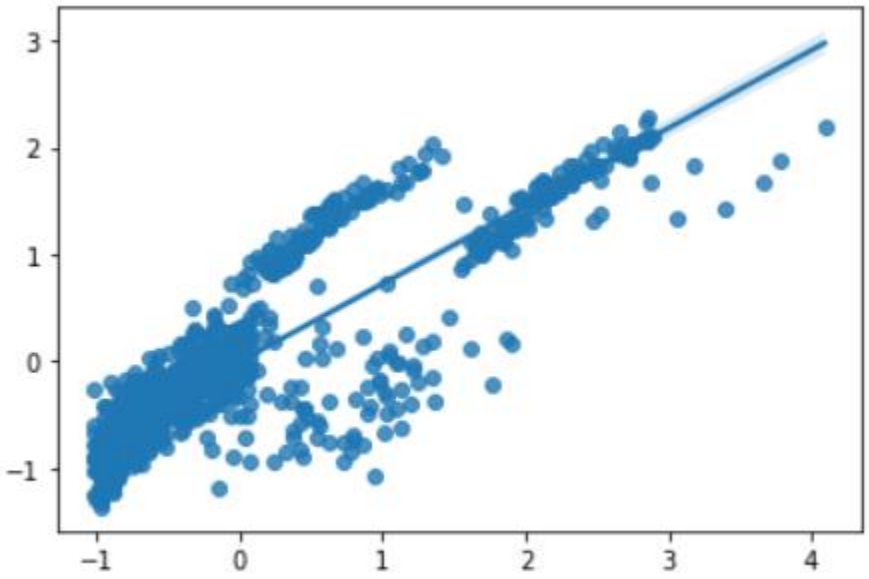
For training data:



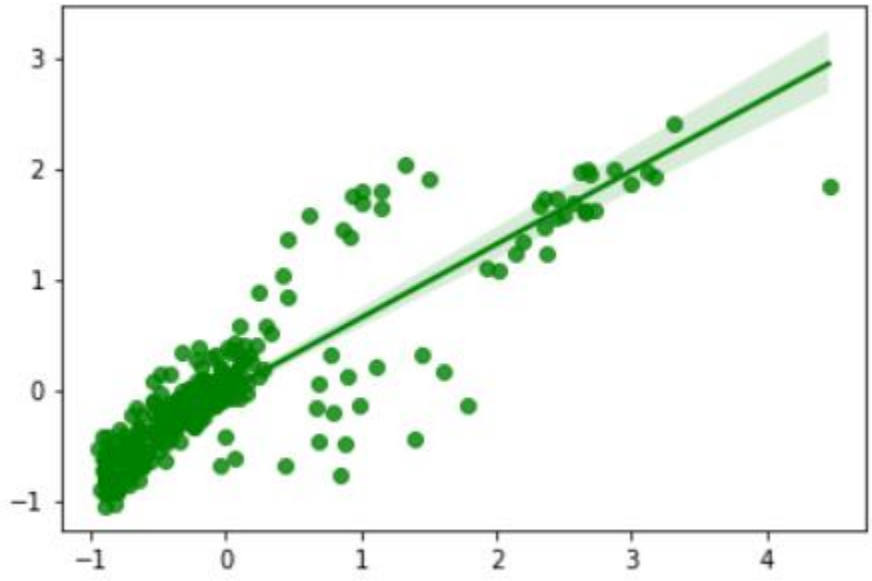
For testing data:



Below is the scatter plot for training data and regression line



The scatter plot for testing data and regression line



How Satisfied?

- We are satisfied with our values. The model is trained with different learning rates and iterations and the values are recorded. From these recordings we selected the best learning rate and epoch. We have found the hypothesis function using the weight vector and predicted the value of Y for each X and stored them. Later we found the rmse and r2 score between predicted values and actual values and got the accuracy as 74.5% on training set and accuracy on testing set as 75 %.

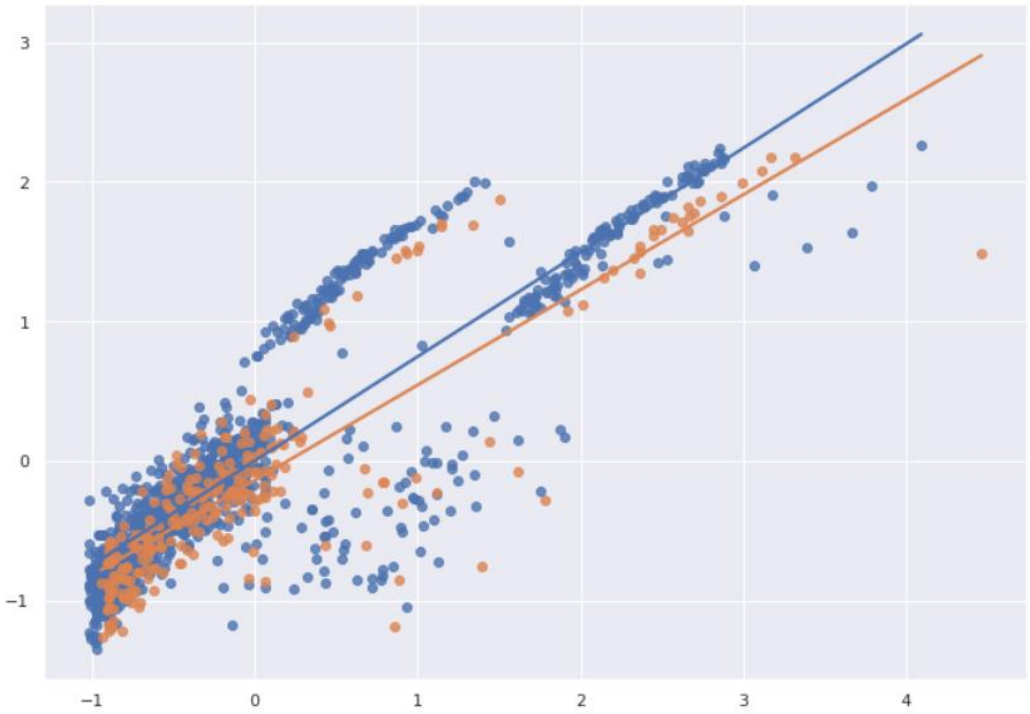
Part 2:

- We used Sklearn's Linear Regression class to build the multiple linear regression model.

- The data pre-processing steps are same as in part 1. We have handled null and duplicate values, categorical data and normalized the data as well.

- Later we imported Linear Regression class from sklearn's linear model library. The dataset is split into training and test sets of 80 % and 20 % and the model is fitted with training data. Later we used that model to predict on test data and compared the predicted values with actual values. The rmse of training data is 0.5013861298625156 and the R squared value is 0.7486119487814886. The rmse on test data is 0.5249847104636908 and R squared value is 0.7243910537793549.

The scattered plots of training and test data and the regressor



How Satisfied?

We are satisfied that we got similar values in comparison with part1. The accuracy on training dataset is 74.8% and the accuracy on testing dataset is 72.4%.