CS 6375
ASSIGNMENT 1

Names of students in your group:

Nitishree Supekar (NXS210021)
Devang Vamja(DSV200000)

Number of free late days used: 0

Note: You are allowed a **total** of 4 free late days for the **entire semester**. You can use at most 2 for each assignment. After that, there will be a penalty of 10% for each late day.

All the sources/references that we have used in this assignment:

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDRegressor.html

https://pandas.pydata.org/docs/reference/general_functions.html

Pre-Processing the data:

Input rows: 205
Input columns: 26

S. No.	Column Name
1	symboling
2	normalized-losses
3	make
4	fuel-type
5	aspiration
6	num-of-doors
7	body-style
8	drive-wheels
9	engine-location
10	wheel-base
11	length
12	width
13	height
14	curb
15	engine-type
16	num-of-cylinders
17	engine-size
18	fuel-system
19	bore
20	stroke
21	compression-ratio
22	horsepower
23	peak-rpm
24	city-mpg
25	highway
26	price

Target variable = Car price
Columns: 1

S. No.	**Column name**
1	price

In this data, it was found that the columns:

Normalized losses had around 42 missing values out of total 200 values which is almost 20% of the data. So, there were 2 options:

1) To remove this attribute

Or

2) To put the mean values for the normalized losses attribute

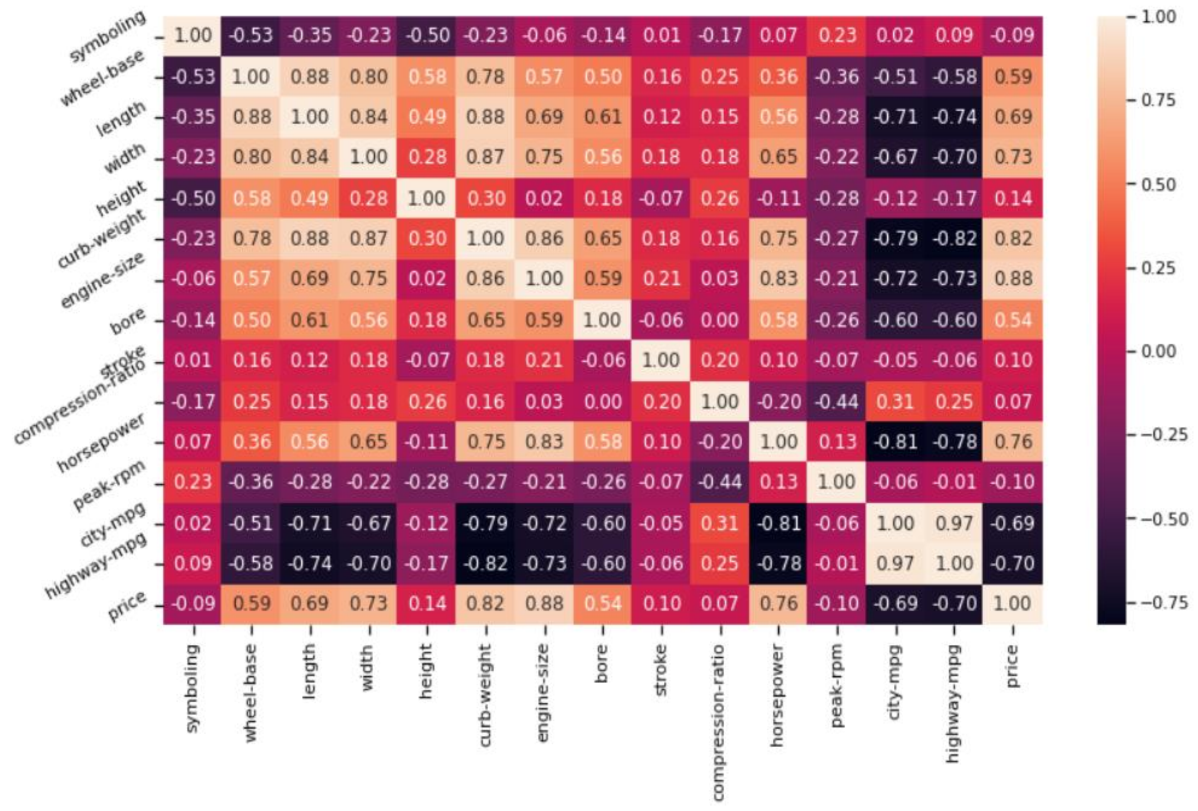
As, the missing values constitute almost 20% of the data, the normalized losses column was removed.

Horsepower and Price were other attributes which have almost 4 to 5 null values, so those null values were not removed but they were replaced by the mean value.

Bore, Stroke were some attributes where we can't take the average and thus these values were directly converted to their numeric values. For missing values, the NA was assigned.

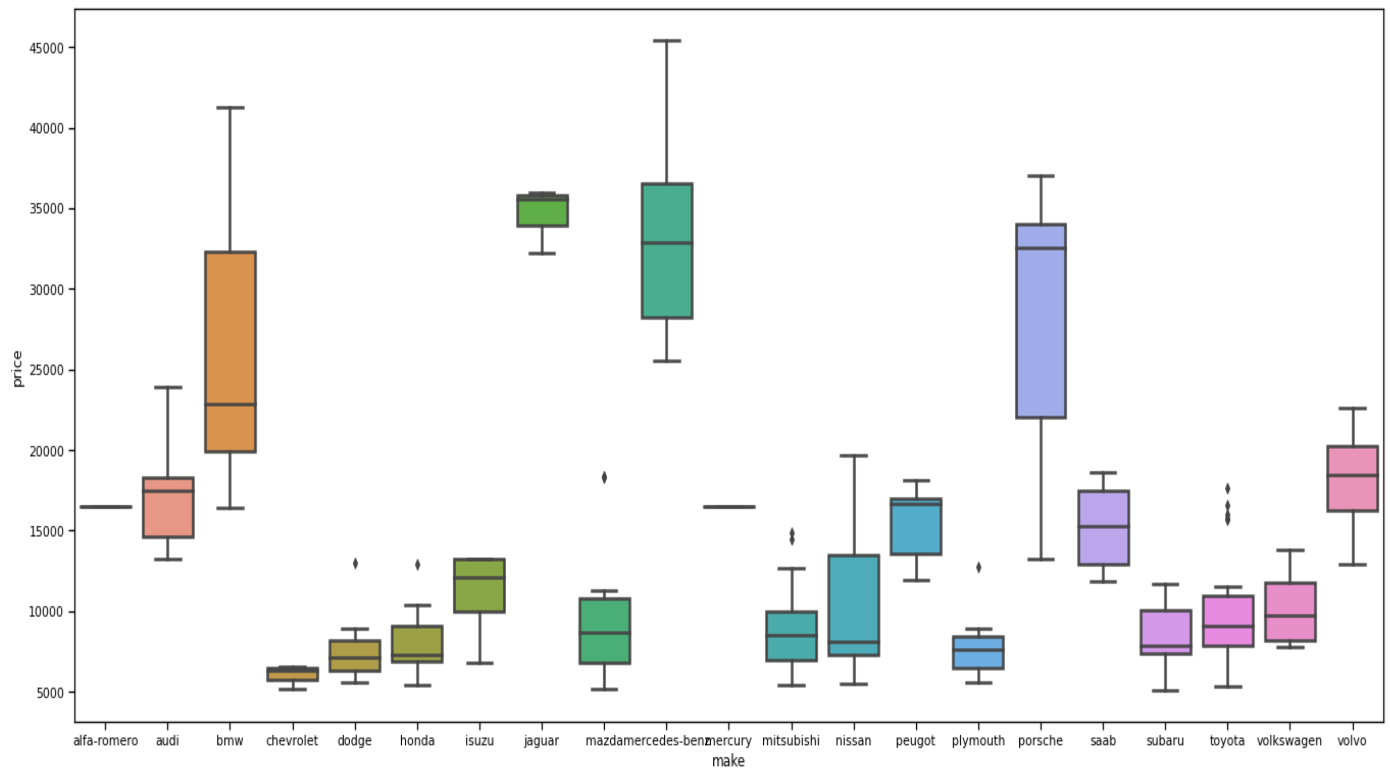
Attribute like num-of-doors has some values which had to be replaced with NA.

<u>Heat Map of the data:</u>



	symboling	wheel-base	length	width	height	curb-weight	engine-size	bore	stroke	compression-ratio	horsepower	peak-rpm	city-mpg	highway-mpg	price
symboling	1.00	-0.53	-0.35	-0.23	-0.50	-0.23	-0.06	-0.14	0.01	-0.17	0.07	0.23	0.02	0.09	-0.09
wheel-base	-0.53	1.00	0.88	0.80	0.58	0.78	0.57	0.50	0.16	0.25	0.36	-0.36	-0.51	-0.58	0.59
length	-0.35	0.88	1.00	0.84	0.49	0.88	0.69	0.61	0.12	0.15	0.56	-0.28	-0.71	-0.74	0.69
width	-0.23	0.80	0.84	1.00	0.28	0.87	0.75	0.56	0.18	0.18	0.65	-0.22	-0.67	-0.70	0.73
height	-0.50	0.58	0.49	0.28	1.00	0.30	0.02	0.18	-0.07	0.26	-0.11	-0.28	-0.12	-0.17	0.14
curb-weight	-0.23	0.78	0.88	0.87	0.30	1.00	0.86	0.65	0.18	0.16	0.75	-0.27	-0.79	-0.82	0.82
engine-size	-0.06	0.57	0.69	0.75	0.02	0.86	1.00	0.59	0.21	0.03	0.83	-0.21	-0.72	-0.73	0.88
bore	-0.14	0.50	0.61	0.56	0.18	0.65	0.59	1.00	-0.06	0.00	0.58	-0.26	-0.60	-0.60	0.54
stroke	0.01	0.16	0.12	0.18	-0.07	0.18	0.21	-0.06	1.00	0.20	0.10	-0.07	-0.05	-0.06	0.10
compression-ratio	-0.17	0.25	0.15	0.18	0.26	0.16	0.03	0.00	0.20	1.00	-0.20	-0.44	0.31	0.25	0.07
horsepower	0.07	0.36	0.56	0.65	-0.11	0.75	0.83	0.58	0.10	-0.20	1.00	0.13	-0.81	-0.78	0.76
peak-rpm	0.23	-0.36	-0.28	-0.22	-0.28	-0.27	-0.21	-0.26	-0.07	-0.44	0.13	1.00	-0.06	-0.01	-0.10
city-mpg	0.02	-0.51	-0.71	-0.67	-0.12	-0.79	-0.72	-0.60	-0.05	0.31	-0.81	-0.06	1.00	0.97	-0.69
highway-mpg	0.09	-0.58	-0.74	-0.70	-0.17	-0.82	-0.73	-0.60	-0.06	0.25	-0.78	-0.01	0.97	1.00	-0.70
price	-0.09	0.59	0.69	0.73	0.14	0.82	0.88	0.54	0.10	0.07	0.76	-0.10	-0.69	-0.70	1.00

Box Plot for the data:



For this data, the test size chosen is 0.20 and the training size chosen is 0.80.
The Gradient Descent can be calculated as:

$$\Theta_j := \Theta_j - \frac{\alpha}{m} \sum_{i=1}^{m} [(h_\Theta(x_i) - y)x_i]$$

Part1:

About the data:

Input columns:

S. No.	Column Name	No. of null values after cleaning the data
1	symboling	0
2	normalized-losses	0
3	make	0
4	fuel-type	0
5	aspiration	0
6	num-of-doors	0
7	body-style	0
8	drive-wheels	0
9	engine-location	0
10	wheel-base	0
11	length	0
12	width	0
13	height	0
14	curb	0
15	engine-type	0
16	num-of-cylinders	0
17	engine-size	0
18	fuel-system	0
19	bore	0
20	stroke	0
21	compression-ratio	0
22	horsepower	0
23	peak-rpm	0
24	city-mpg	0
25	highway	0

Target variable = Car price
Columns: 1

S. No.	Column name	No. of null values after cleaning the data
1	price	0

Correlation between variables:

```
Index                  Price
symboling           -0.085552
make                -0.163905
fuel-type           -0.115455
aspiration           0.183760
num-of-doors        -0.046338
body-style          -0.071060
drive-wheels         0.586461
engine-location      0.330850
wheel-base           0.587962
length               0.688535
width                0.733179
height               0.136955
curb-weight          0.821705
engine-type          0.082498
num-of-cylinders     0.010061
engine-size          0.877588
fuel-system          0.502175
bore                 0.535607
stroke               0.096915
compression-ratio    0.074465
horsepower           0.760029
peak-rpm            -0.103084
city-mpg            -0.686792
highway-mpg         -0.704826
price                1.000000
```

From the above data, it can be observed that the engine-size varies proportionally with the target variable price.
And, the variable highway-mpg varies inversely with the target variable price.

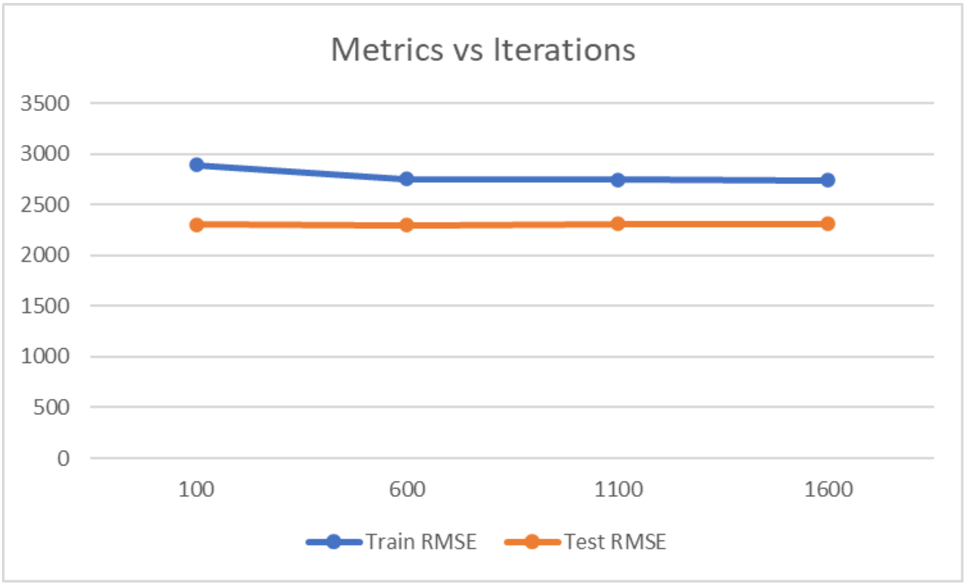
Model with all variables:

Train MSE	2782.558328957282
Train R2 score	0.8907422114518592
Test MSE	2289.075176093021
Test R2 score	0.8485296475351086

The R2 score increased with addition of variables to the model. So, we build the model with all variables.

Tuning the number of iterations with learning rate=0.1:

Iterations	100	600	1100	1600
Train MSE	2892.59847 03449717	2753.63909 7856751	2745.69592 5941894	2744.19984 48967583
Train R2 score	0.88193441 9764637	0.89300234 70267178	0.89361788 42926321	0.89373380 61528187
Test MSE	2307.27937 2470909	2303.41161 91621238	2309.90889 55316124	2313.12006 71907336
Test R2 score	0.84611088 87603909	0.84662639 30788132	0.84575992 51476096	0.84533078 64675345



We observe that there is not much difference in metrics after 500 iterations.
The model is reaching optimum point with 0.1 rate. As we decrease the rate it takes more time to reach optimum. This phenomenon could be observed in the last column with lowest learning rate.
Results:
The linear model with iterations=300 and learning rate=0.1 gives

Train MSE	2782.558328957282
Train R2 score	0.8907422114518592
Test MSE	2289.075176093021
Test R2 score	0.8485296475351086

The linear model is giving an R2 score of 0.890 and means square error of 2782.55 with training data. There is only a slight difference between metrics for training data and test data. This might be improved with some transformation of variables like algebraic, exponential, or logarithmic as most of the input variables are not linearly related to output (based on correlation values). The level of accuracy required is moderate based on the domain it is being used.
Hence, the model provides satisfactory results.

Part 2:

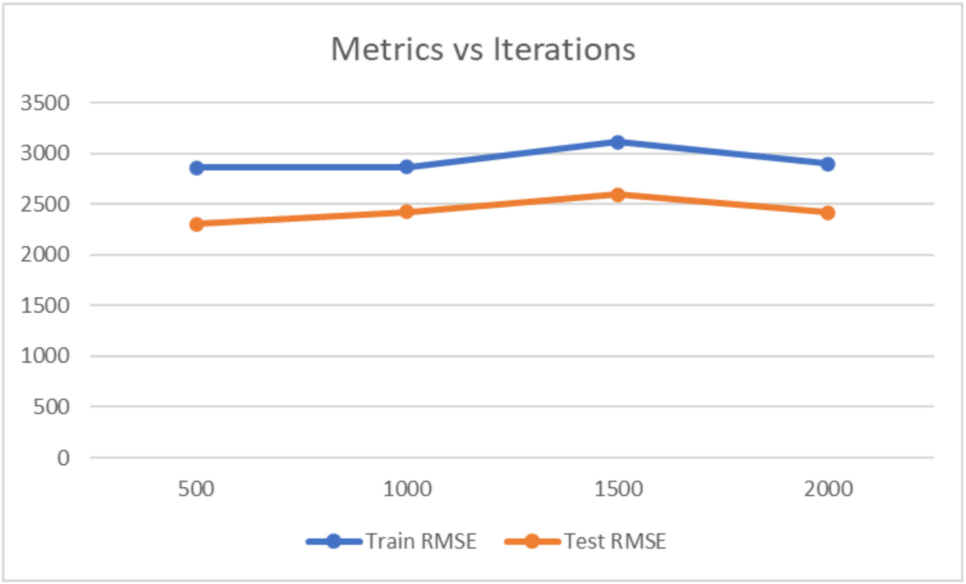
Model with all variables:
For Training:
Mean Absolute Error: 2001.46616499822
Mean Squared Error: 8162186.671586426
Root Mean Squared Error: 2856.954089863263
For Testing:
Mean Absolute Error: 1670.96876777198
Mean Squared Error: 5261450.300934324
Root Mean Squared Error: 2293.785147073353

Model with all variables:

Train MSE	2856.954089863263
Train R2 score	0.633733
Test MSE	2293.785147073353
Test R2 score	0.521216

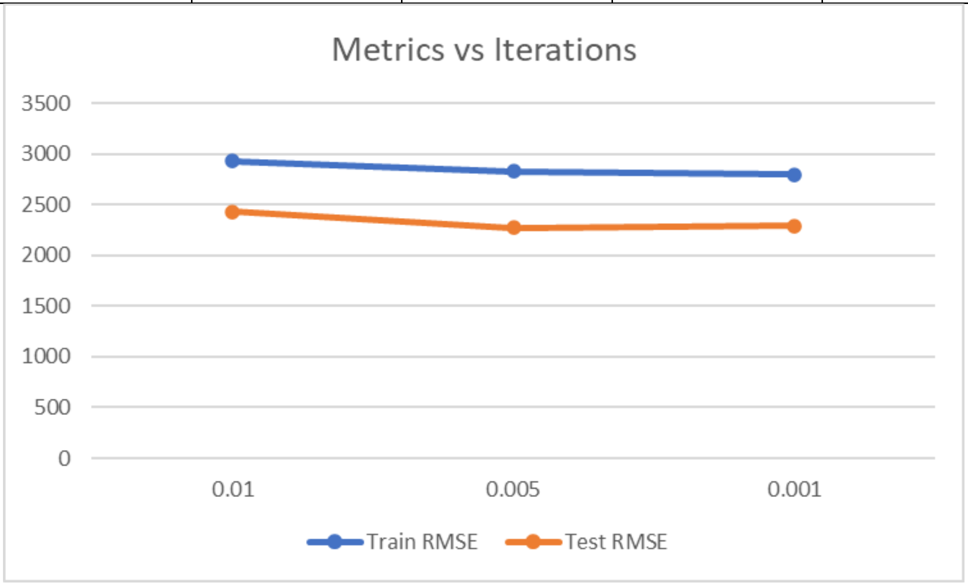
Tuning the number of iterations with learning rate=0.01:

Iterations	**500**	**1000**	**1500**	**2000**
Train RMSE	2862.9754	2869.075	3113.657	2897.286
Test RMSE	2303.754	2423.5483	2595.4164	2418.3513



Tuning the number of iterations with learning rate=500:

Learning Rate	**0.05**	**0.01**	**0.005**	**0.001**
Train MSE	187485678672 .14	2931.695	2832.6080	2798.0937
Test MSE	80500974105. 73	2432.212	2272.2667	2292.07337



The optimum learning rate from the plot is 0.01. There is not much difference in MSE after 0.01.
Results:
The linear model is giving a means square error of 2856.954089863263with training data. There is a slight difference between metrics for training data and test data. This might be improved with some transformation of variables like algebraic, exponential, or logarithmic as most of the input variables are not linearly related to output (based on correlation values). The level of accuracy required is moderate based on the domain it is being used. Hence, the model provides satisfactory results.

Observations of Part1 and Part2:

From both part1 and part2, it is evident that the part1 model is giving almost similar results compared to built-in module based on R2 score and MSE. Hence, we can say that the package (part1) is able to find the best solution.