CS 6375
ASSIGNMENT 1: Linear Regression using Gradient Descent

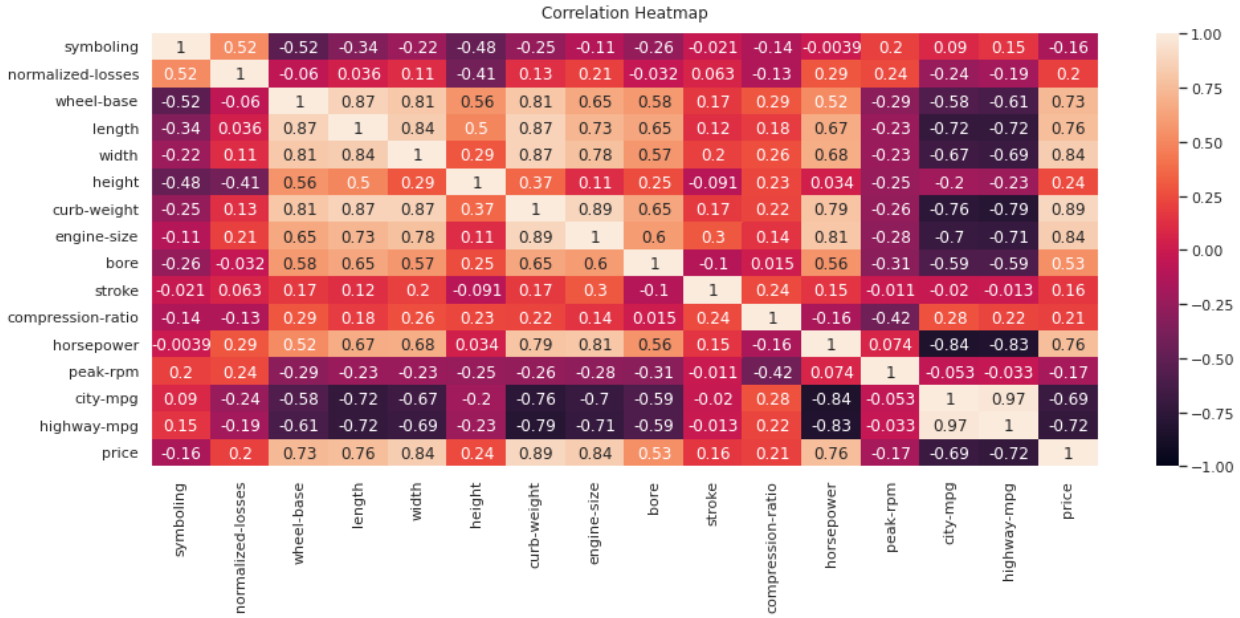
Names of students in your group:
Kunal Singh kds190006
Pranav More pxm210017

Number of free late days used:
Zero

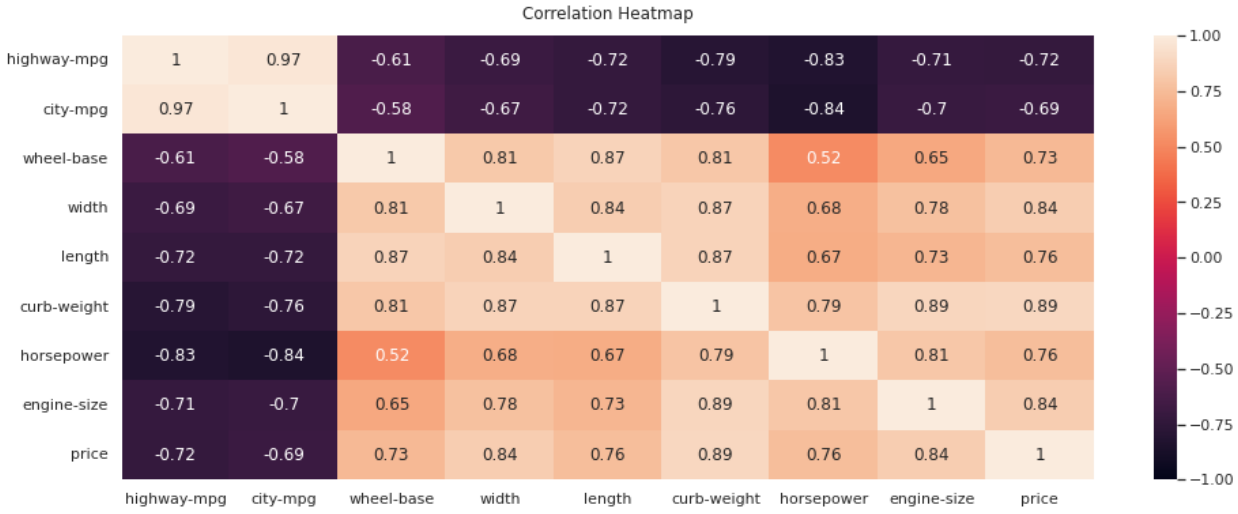
Part 1:

Here we have to develop a linear regression model using gradient descent. For this task we are going to use automobile data set (https://archive-beta.ics.uci.edu/ml/datasets/automobile). The goal is to predict the price of the automobile given required features.

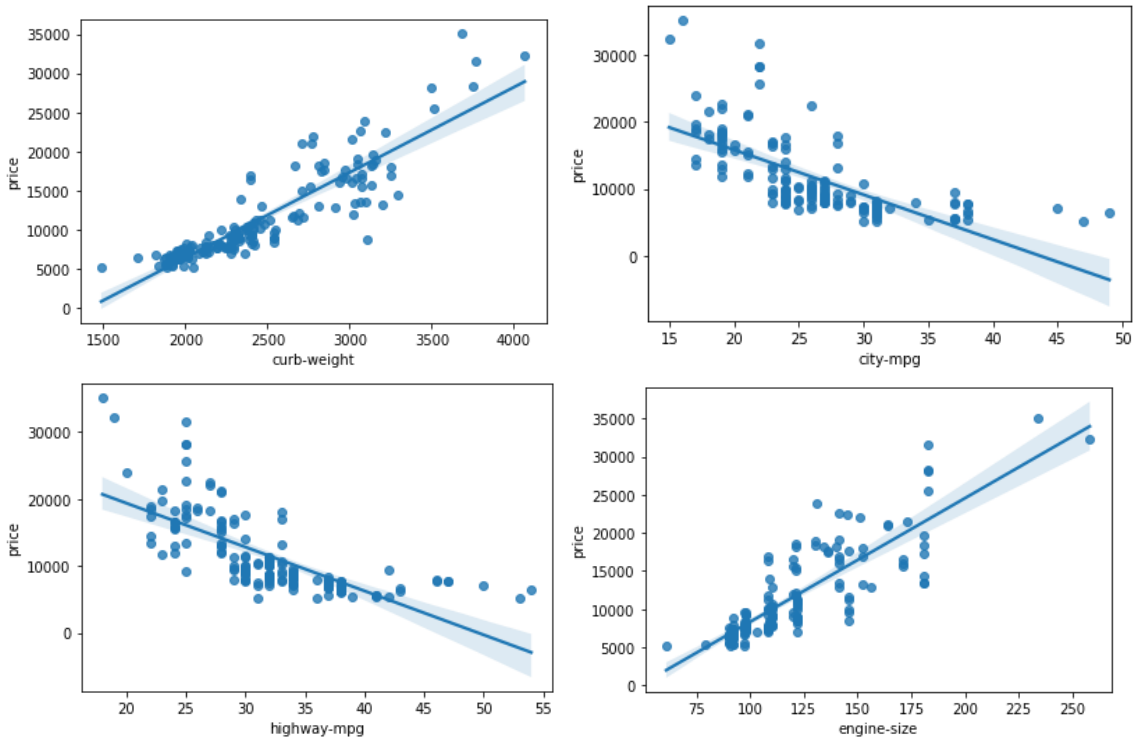
For this first we must analyze the data set and find the appropriate attributes for this task. Using correlation, we can see that,

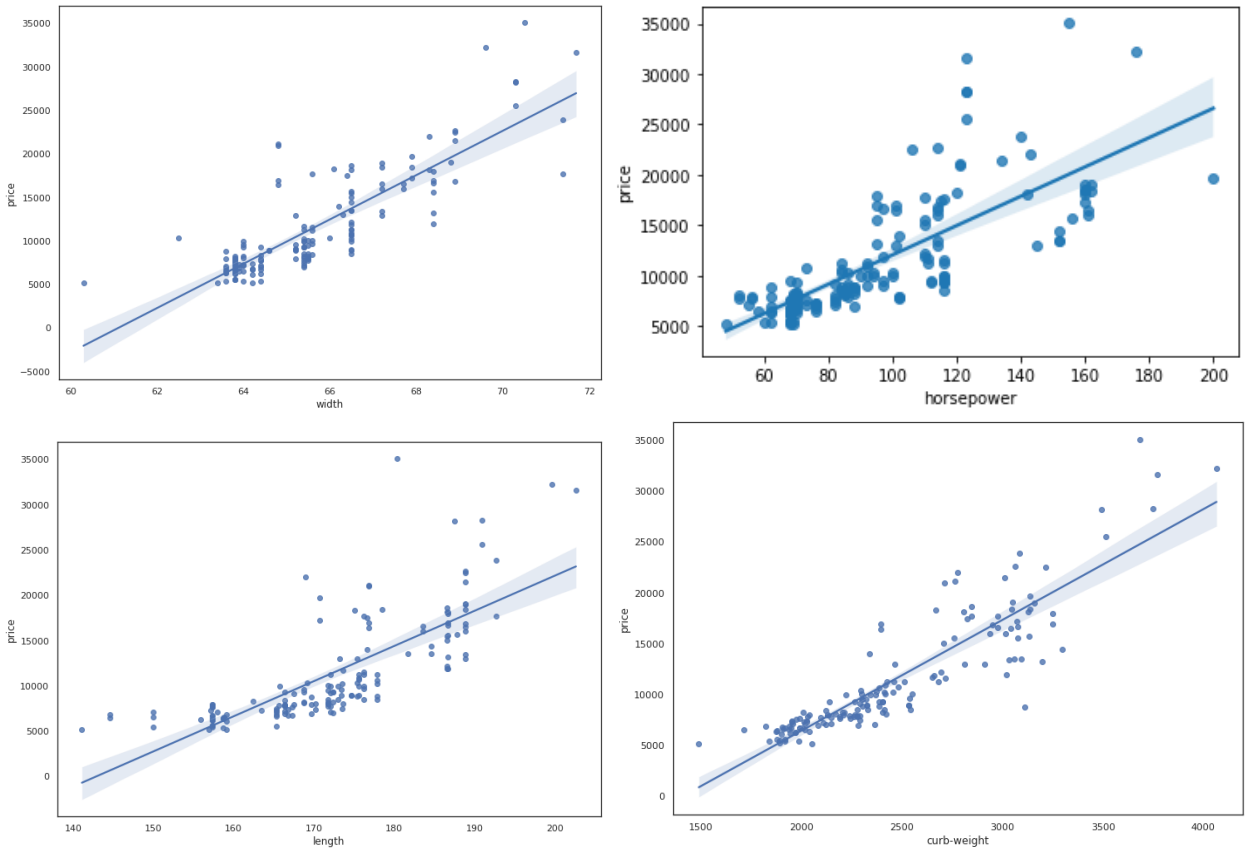


We can see that price has high negative correlation with highway-mpg and city-mpg. Which makes sense since as miles per gallon decrease chances as the vehicle is old or not efficient and hence has low price.
Using correlation matrix, we can also see that price has high positive correlation with horsepower, engine-size, curb-weight, length and wheel-base.



To verify our assumptions let make individual plot w.r.t price,





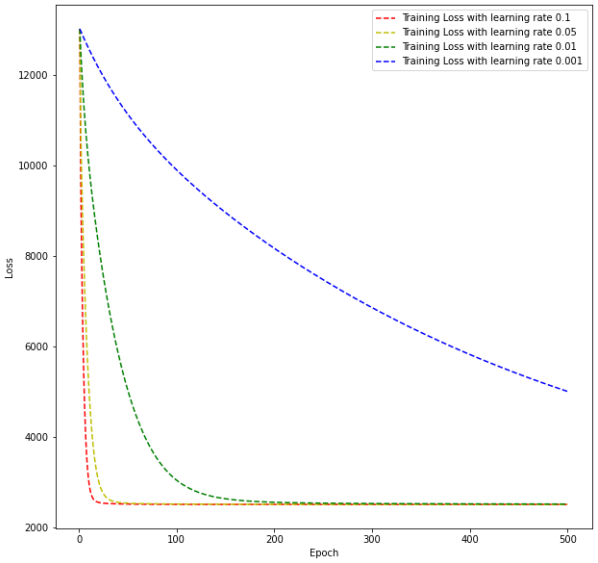
So this verifies our assumption about the selected features. The next step is to process the data to make it fit for our model. We do this by removing all the Null values from our dataset and also removing all the unwanted features from our dataset.

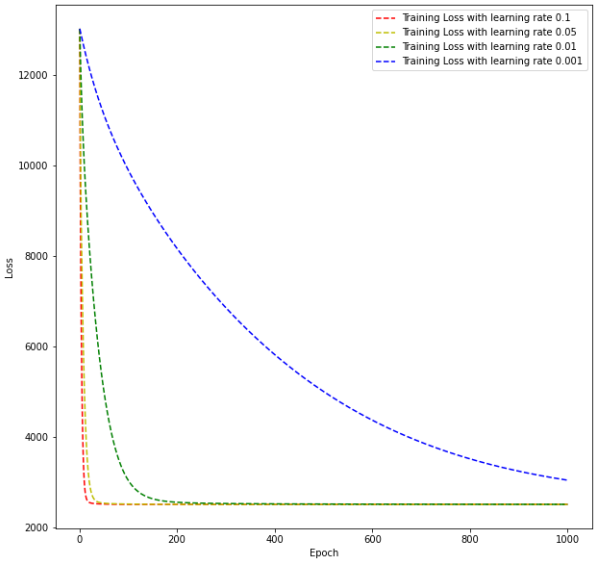
For this we have created a Data class with all the relevant function to process data, decide features and target, and split dataset into training and testing set with a split ration of 8:2 because this split gave us more optimized answers.

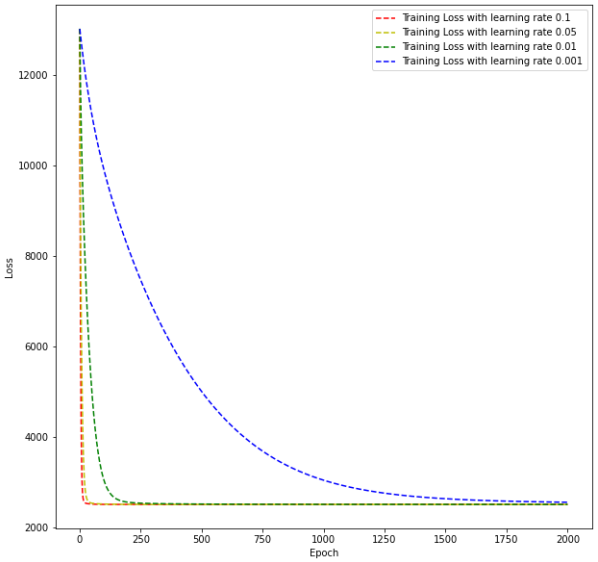
Once our dataset is ready for training we can proceed to the next step which is creating our model. We are going to perform multivariant linear regression with Gradient descent on our dataset. This algorithm uses certain hyper parameter's which we are going to optimize to get the best possible model.

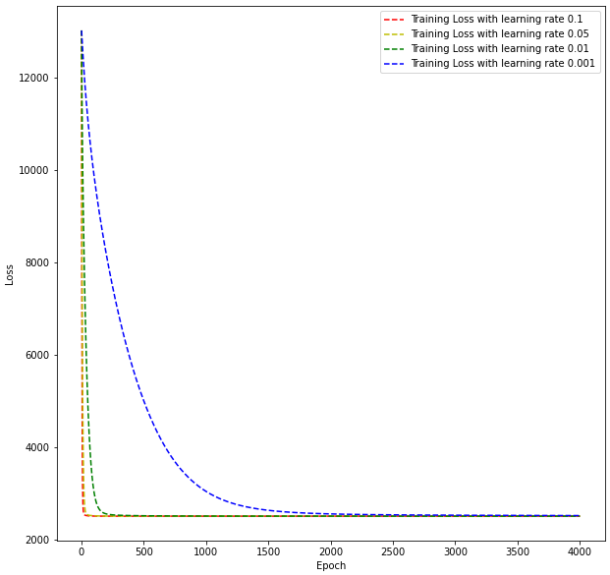
We can test the performance of our model and compare them with other model using measures like RMSE(root mean squared error) and R^2.

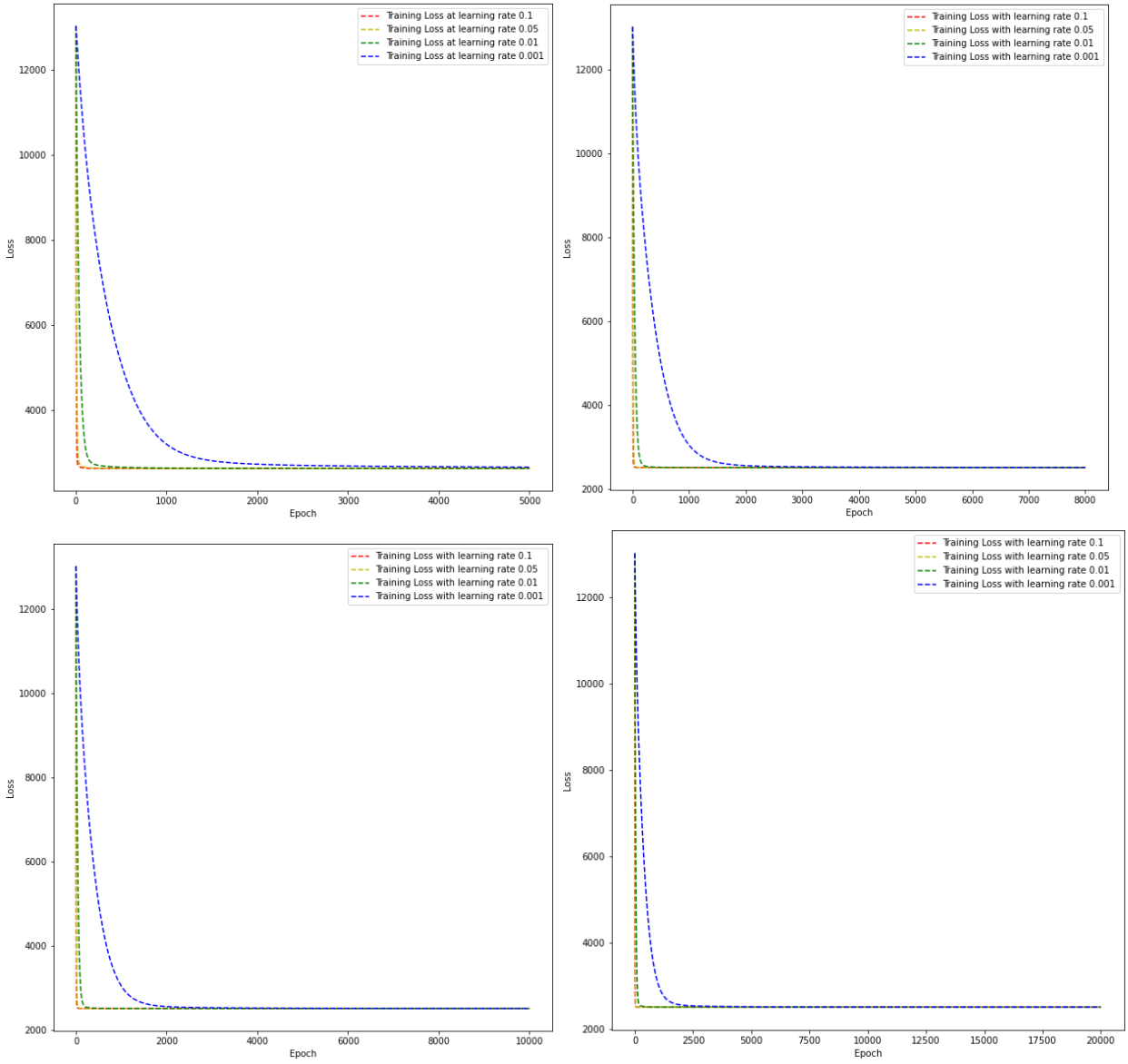
These Hyperparameters are Learning rate and iteration for which our model should learn on also known as Epochs. Using our dataset and model we have tested several combinations of these hyperparameter. Below are graphs of loss per iteration and number of iteration for different learning rates.


Training Loss with learning rate 0.1
Training Loss with learning rate 0.05
Training Loss with learning rate 0.01
Training Loss with learning rate 0.001
Loss
Epoch
12000
10000
8000
6000
4000
2000
0
100
200
300
400
500


Training Loss with learning rate 0.1
Training Loss with learning rate 0.05
Training Loss with learning rate 0.01
Training Loss with learning rate 0.001
Loss
Epoch
12000
10000
8000
6000
4000
2000
0
200
400
600
800
1000


Training Loss with learning rate 0.1
Training Loss with learning rate 0.05
Training Loss with learning rate 0.01
Training Loss with learning rate 0.001
Loss
Epoch
12000
10000
8000
6000
4000
2000
0
250
500
750
1000
1250
1500
1750
2000


Training Loss with learning rate 0.1
Training Loss with learning rate 0.05
Training Loss with learning rate 0.01
Training Loss with learning rate 0.001
Loss
Epoch
12000
10000
8000
6000
4000
2000
0
500
1000
1500
2000
2500
3000
3500
4000



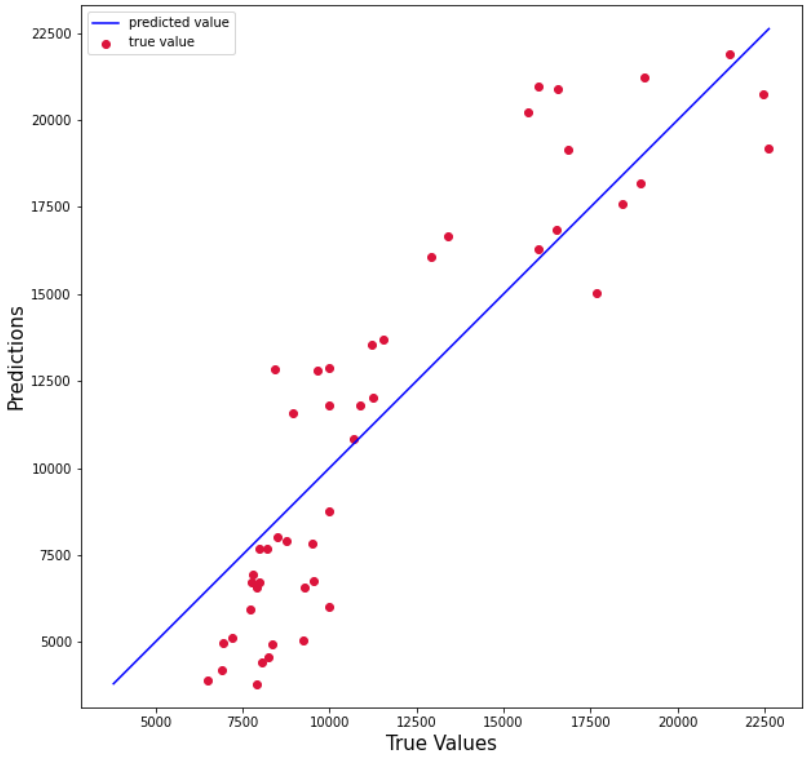
We can see that as number of iteration increase the loss decreases and if epochs are big enough they converge. For smaller learning rate like 0.001 the decrease in loss is gradual, but for larger learning rate the decrease in loss is steep.

Following table shows R^2 value for various Epoch size:

Epoch vs Learning Rate	0.1	0.05	0.01	0.001
500	0.67832	0.67474	0.6560	-0.2664
1000	0.67911	0.67832	0.66515	0.5317
2000	0.67911	0.67911	0.6729	0.6576
4000	0.67911	0.67911	0.67752	0.65479
8000	0.67911	0.67911	0.67904	0.66192
10000	0.67911	0.67911	0.67911	0.66511
20000	0.67911	0.67911	0.67911	0.6729

Our best model give us an R^2 score of 0.67911 on our test dataset.

Below is the graph of Expected test values and predicted test line.



Are you satisfied that you have found the best solution? Explain.

For our dataset this model provides a moderate fit since our R^2 is between 0.5 and 0.8. This is a good indication that our model is not overfitting as we can see in the above graph that model is trying to find the best possible linear relationship between true values and predicted values. Also, our model gives a relatively low RMSE score of 2700, meaning that predictions are not far off from the true values. So, I believe this is a good solution which can be further improved using advance techniques like Stochastic Gradient Descent, batch gradient descent etc.

Part 2:

In second part of this assignment, we will be using a pre-existing library to perform our task of gradient descent. The goals and expected results are similar to the previous part. The data used for this part will be similar to the one used in part I.

Hence, we will use similar assumptions about our dataset and select the same features as before for automobile price prediction task.

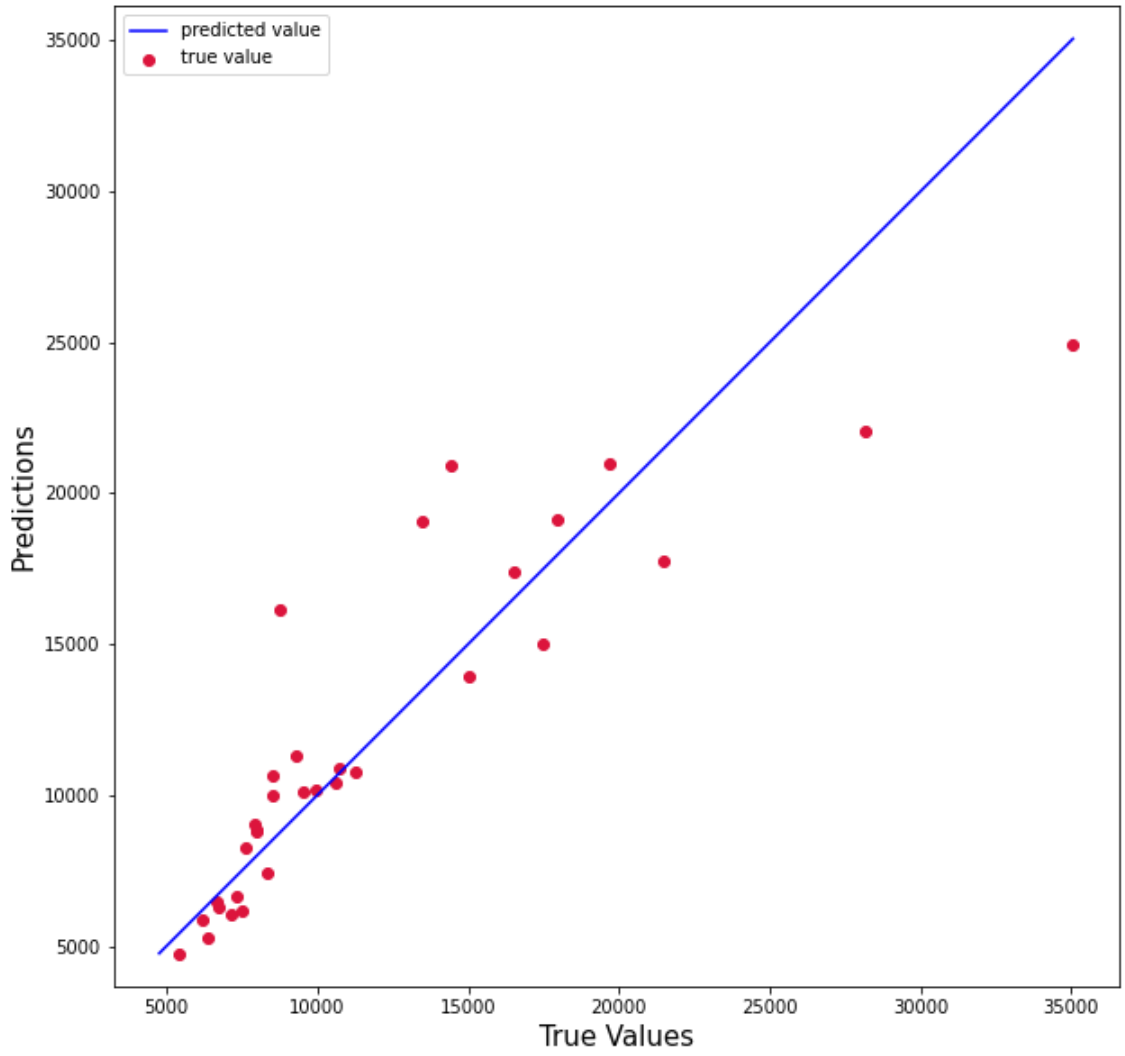
We have used sklearn library to implement linear regression using Stochastic Gradient Descent (SGD). First, we have preprocessed the dataset by removing null values, converting categorical variables to numerical variables. After me observed correlation of all the attributes with the target and only selected the best attributes which showed high correlation. We had to scale the data as the attributes had different ranges. After scaling we divided the data into training and test data in the ration 8:2. This split was decided because it was the most optimized split.
The following table shows r2 score for various epochs:

Epoch vs Learning Rate	0.1	0.05	0.01	0.001
500	0.80903	0.80828	0.81604	0.81726
1000	0.80127	0.81326	0.81630	0.81889
2000	0.81162	0.81455	0.81540	0.81859
4000	0.80752	0.81027	0.81607	0.81525
8000	0.80236	0.80432	0.81519	0.81710
10000	0.81087	0.81401	0.81525	0.81413
20000	0.80563	0.80491	0.81695	0.81810

Our best model give us an R^2 score of 0. 81889 on our test dataset.

Are you satisfied that the package has found the best solution? How can you check. Explain.

Yes, we are satisfied that the package has found the best solution which can be checked by comparing the r2 score that we got from the first part which was 0.67911 and the best r2 score for second part is 0.81889 on the test dataset. This r2 score of 0.81889 also shows a high correlation and the rmse valse is around 3000 which shows that the predicted values are not far from the actual values. We can confirm this by observing the plot of expected value and predicted value which we have done below. So, we satisfied that the package has found the best solution.



Output Logs:

Part 1:

```
EPOCH:  500
_____________________________________________

Training for learning rate 0.1.
Testing rmse:  2548.890240443067
Testing r2:  0.6783253707053885
_____________________________________________

Training for learning rate 0.05.
Testing rmse:  2563.027277042548
Testing r2:  0.6747472353899935
_____________________________________________

Training for learning rate 0.01.
Testing rmse:  2635.7768962110176
Testing r2:  0.6560210756637546
_____________________________________________

Training for learning rate 0.001.
Testing rmse:  5057.392517636174
Testing r2:  -0.2663911407615871


EPOCH:  1000
_____________________________________________

Training for learning rate 0.1.
Testing rmse:  2545.76128658372
Testing r2:  0.6791146454074592
_____________________________________________

Training for learning rate 0.05.
Testing rmse:  2548.8936398615356
Testing r2:  0.6783245126791091
_____________________________________________

Training for learning rate 0.01.
Testing rmse:  2600.9462083289786
Testing r2:  0.6650520831597533
_____________________________________________

Training for learning rate 0.001.
Testing rmse:  3075.0946366826834
Testing r2:  0.5317998893920046
```

EPOCH: 2000

Training for learning rate 0.1.
Testing rmse: 2545.766292653083
Testing r2: 0.679113383407027

Training for learning rate 0.05.
Testing rmse: 2545.7694367125664
Testing r2: 0.6791125908069867

Training for learning rate 0.01.
Testing rmse: 2570.4182621289733
Testing r2: 0.6728686721263244

Training for learning rate 0.001.
Testing rmse: 2629.173566865472
Testing r2: 0.6577424359862082

EPOCH: 4000

Training for learning rate 0.1.
Testing rmse: 2545.7726804592835
Testing r2: 0.6791117730753193

Training for learning rate 0.05.
Testing rmse: 2545.7662915447277
Testing r2: 0.6791133836864369

Training for learning rate 0.01.
Testing rmse: 2552.027331465911
Testing r2: 0.677533070139902

Training for learning rate 0.001.
Testing rmse: 2640.546924204814
Testing r2: 0.6547749354549961

EPOCH: 8000

Training for learning rate 0.1.
Testing rmse: 2545.772686793889
Testing r2: 0.679111771478397

```
Training for learning rate 0.05.
Testing rmse:  2545.7726804056433
Testing r2:  0.6791117730888419
_____________________________________________

Training for learning rate 0.01.
Testing rmse:  2546.0511708718655
Testing r2:  0.6790415632067623
_____________________________________________

Training for learning rate 0.001.
Testing rmse:  2612.8479992520647
Testing r2:  0.6619796620358493


EPOCH:  10000
_____________________________________________

Training for learning rate 0.1.
Testing rmse:  2545.772686793891
Testing r2:  0.6791117714783965
_____________________________________________

Training for learning rate 0.05.
Testing rmse:  2545.7726865993604
Testing r2:  0.6791117715274367
_____________________________________________

Training for learning rate 0.01.
Testing rmse:  2545.7677819706046
Testing r2:  0.6791130079584229
_____________________________________________

Training for learning rate 0.001.
Testing rmse:  2600.743483953862
Testing r2:  0.6651042945132131


EPOCH:  20000
_____________________________________________

Training for learning rate 0.1.
Testing rmse:  2545.772686793891
Testing r2:  0.6791117714783966
_____________________________________________

Training for learning rate 0.05.
Testing rmse:  2545.772686793887
Testing r2:  0.6791117714783976
_____________________________________________

Training for learning rate 0.01.
Testing rmse:  2545.7662314208897
Testing r2:  0.6791133988433151
_____________________________________________
```

```
Training for learning rate 0.001.
Testing rmse:  2570.3582490380745
Testing r2:  0.6728839474091239
```

Part 2:

```
EPOCH:  500
_____________________________________________

Training for learning rate 0.1.
r2:  0.7798546634430198
RMSE:  3119.6016981457783
_____________________________________________

Training for learning rate 0.05.
r2:  0.777766620127446
RMSE:  3134.3612443963325
_____________________________________________

Training for learning rate 0.01.
r2:  0.7825703832538257
RMSE:  3100.300235122869
_____________________________________________

Training for learning rate 0.001.
r2:  0.7773577663222107
RMSE:  3137.2431396730017


EPOCH:  1000
_____________________________________________

Training for learning rate 0.1.
r2:  0.7815169396450484
RMSE:  3107.8016168337303
_____________________________________________

Training for learning rate 0.05.
r2:  0.775038321368614
RMSE:  3153.542403747788
_____________________________________________

Training for learning rate 0.01.
r2:  0.7760922152966587
RMSE:  3146.1469201839004
_____________________________________________

Training for learning rate 0.001.
r2:  0.7775947663039148
RMSE:  3135.572916264405
```

EPOCH: 2000

Training for learning rate 0.1.
r2: 0.7762887486465506
RMSE: 3144.7658638354046

Training for learning rate 0.05.
r2: 0.776925566227521
RMSE: 3140.2867207593786

Training for learning rate 0.01.
r2: 0.7746035150457413
RMSE: 3156.588518689673

Training for learning rate 0.001.
r2: 0.7806580043822723
RMSE: 3113.904565835855

EPOCH: 4000

Training for learning rate 0.1.
r2: 0.7765458610689738
RMSE: 3142.9581969927203

Training for learning rate 0.05.
r2: 0.7800808103310588
RMSE: 3117.9989626520783

Training for learning rate 0.01.
r2: 0.7813709517115764
RMSE: 3108.8397425600106

Training for learning rate 0.001.
r2: 0.7801445465192022
RMSE: 3117.547106291435

EPOCH: 8000

Training for learning rate 0.1.
r2: 0.7830155252847918
RMSE: 3097.124998368825

Training for learning rate 0.05.
r2: 0.7769533110138303
RMSE: 3140.0914287848677

Training for learning rate 0.01.
r2: 0.7817316277553273
RMSE: 3106.274331284741

Training for learning rate 0.001.
r2: 0.7780347919055526
RMSE: 3132.469537226993

EPOCH: 10000

Training for learning rate 0.1.
r2: 0.773423451809959
RMSE: 3164.840890633887

Training for learning rate 0.05.
r2: 0.7780355862221944
RMSE: 3132.463932351052

Training for learning rate 0.01.
r2: 0.7789652348236575
RMSE: 3125.8972349523588

Training for learning rate 0.001.
r2: 0.7769667102742326
RMSE: 3139.9971087686154

EPOCH: 20000

Training for learning rate 0.1.
r2: 0.7746968101268591
RMSE: 3155.935170741067

Training for learning rate 0.05.
r2: 0.7768458069701649
RMSE: 3140.8480682615736

Training for learning rate 0.01.
r2: 0.7845068263224582
RMSE: 3086.463615562767

```
_________________________________________________

Training for learning rate 0.001.
r2:  0.7798136949853484
RMSE:  3119.8919594036806
```