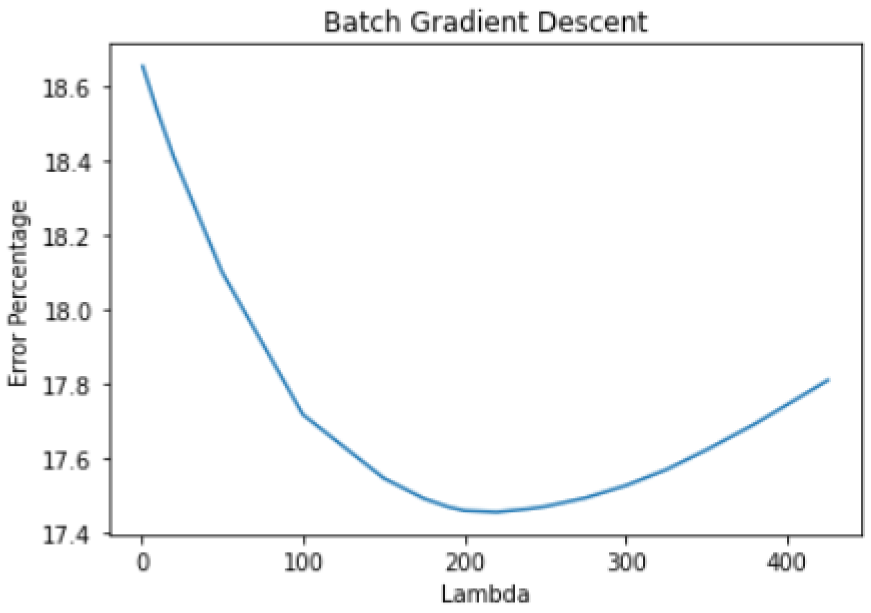
ANIRUDH ARORA
IIT2019003

Q6 (c)
Batch Gradient Descent
Simple Gradient Descent Algorithm is applied on 70% of the data used for training purposes and the error is calculated with the value of alpha = 0.01 and number of iterations = 10000. Feature scaling is applied and Net error is calculated using the mean squared error formula. Error without Regularization came out to be `18.66649001%`

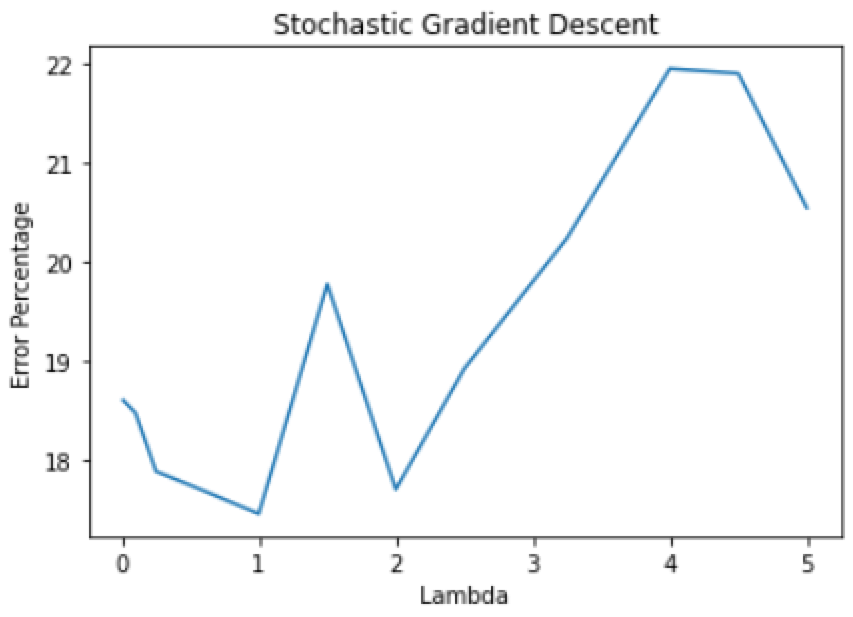
Regularization Algorithm was then Applied for different values of lambda and analysis was done using graph plotting. The error percentage is plotted against the different values of lambda. It can be observed that, with the increase in the value of lambda, the error percentage first decreases up to the minimum value and then again starts to rise.



Stochastic Gradient Descent
Stochastic Gradient Descent Algorithm is applied on 70% of the data used for training purposes and the error is calculated with the value of alpha = 0.01 and number of iterations = 10000. Numpy library from python is used to generate random sequences while implementing the procedure. Feature scaling is applied and Net error is calculated using the mean squared error formula.
Error without Regularization came out to be `18.22820986%`

Regularization Algorithm was then Applied for different values of lambda and analysis was done using graph plotting. The error percentage is plotted against the different values of lambda. It can be observed that the error percentage does not follow an observable trend in the case of stochastic regression but the average value for net error first decreases then increases.

ANIRUDH ARORA
IIT2019003



Mini Batch Regression Algorithm

Mini Batch Gradient Descent Algorithm is applied on 70% of the data used for training purposes and the error is calculated with the value of alpha = 0.01 and number of iterations = 10000 and with varying batch sizes. A separate function is written in order to create batches of the required size. Feature scaling is applied and Net error is calculated using the mean squared error formula.

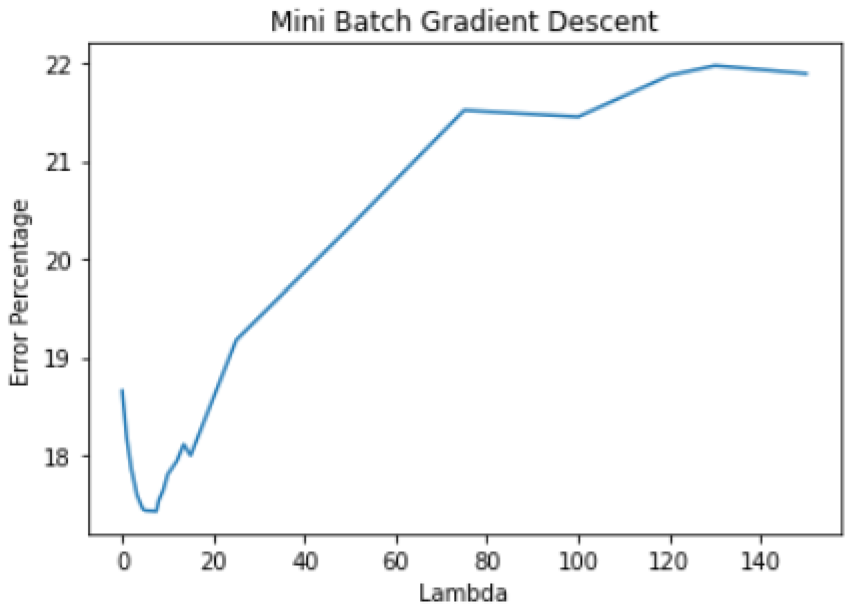
Errors for different batch sizes are as follows:

Batch Size	Error
10	18.5743133%
20	18.73536194%
30	18.67081279%
40	18.68320043%
50	18.75996553

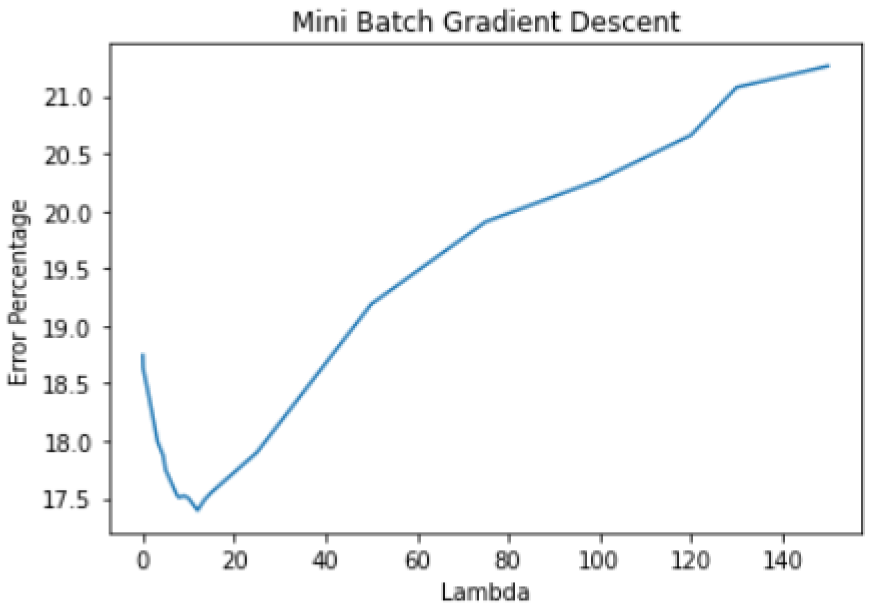
Regularization Algorithm was then Applied for different values of lambda and analysis was done using graph plotting. The error percentage is plotted against the different values of lambda. It can be observed that the error percentage first decreases and then increases with a rising value of lambda.

ANIRUDH ARORA
IIT2019003

Batch Size = 10

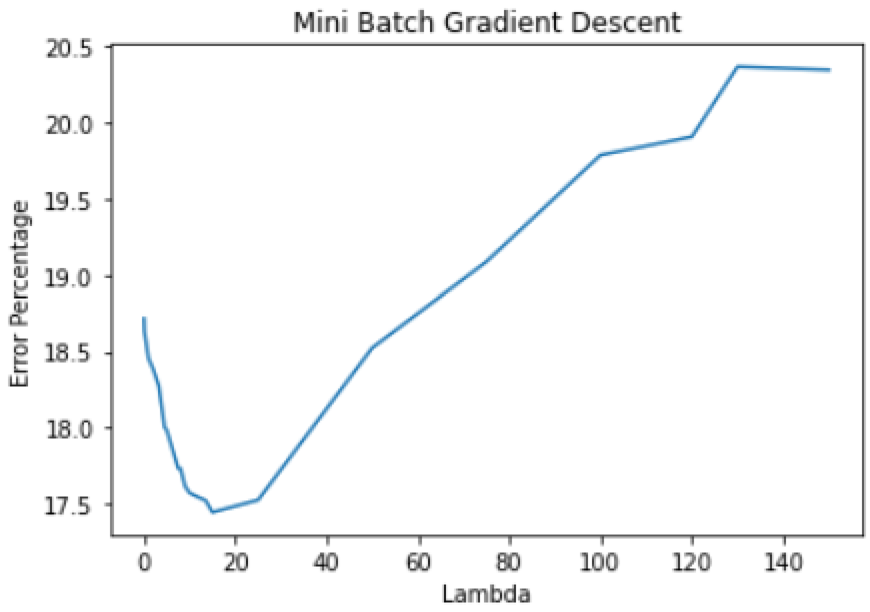


Batch Size = 20

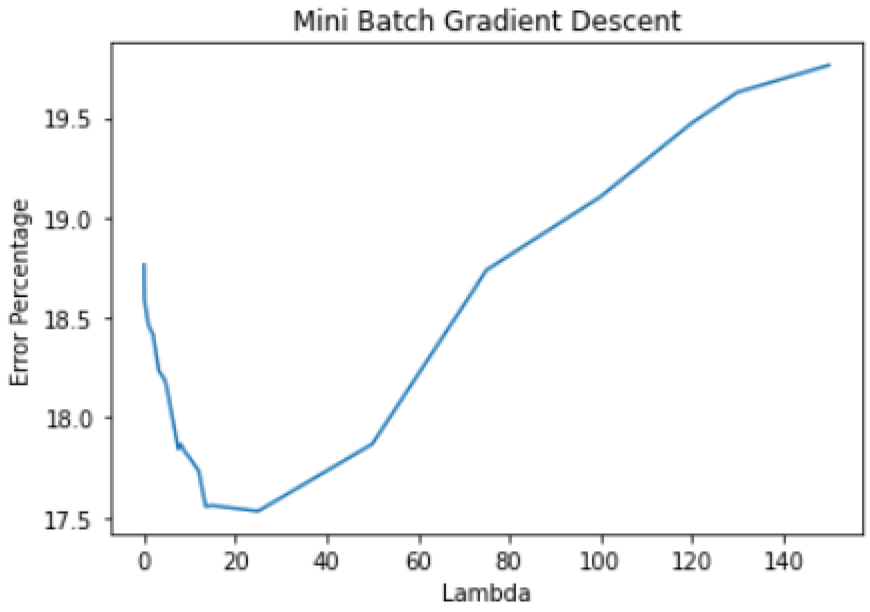


ANIRUDH ARORA
IIT2019003

Batch size = 30

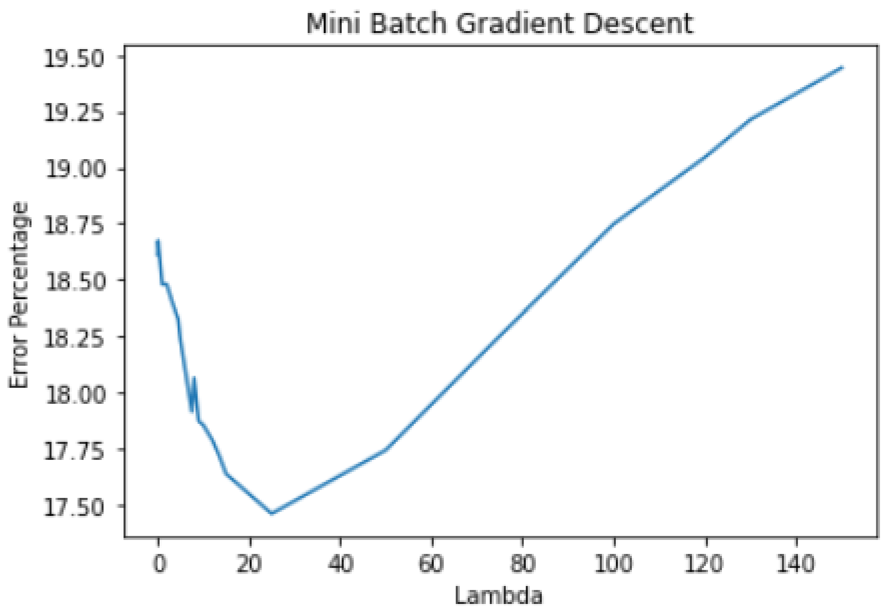


Batch size = 40



ANIRUDH ARORA
IIT2019003

Batch size = 50



Q 6 (d)
Locally weighted Regression

First of all, reading data from the data file given is done. First of all, the values of alpha and lambda are taken to be alpha = 0.01 and lambda = -49. Taking these values the coefficients for the batch gradient with scaling and regularization is done. The values for the coefficients comes out to be:

```
[5037.585668619078, 11147.667574879839, 10378.580439168689, 22647.298983883848]
```

The mean absolute error is calculated, it comes out to be `19.92701396456417%`

Again using the stochastic gradient descent with scaling and regularization is calculated, the coefficients come out to be:

```
[68977.37183533033, 153.6672368058788, 622.1158811423422, 207.34938918615728]
```

The mean absolute error is calculated, it comes out to be `22.392602067246287%`

Again using the mini-batch gradient descent with batch size = 20 along with scaling and regularization is calculated, the coefficients come out to be:

```
[888.9201243627547, 5168.211726125255, 17701.360814619125, 15202.387873756412]
```

The mean absolute error is calculated, it comes out to be `19.550681895981263%`

A separate function for Locally weighted Regression is written which is commonly called for each of the cases discussed above. Separate functions for each calculation of slope, error calculation, kernel formation, and slope calculation are written so as to increase the code reusability