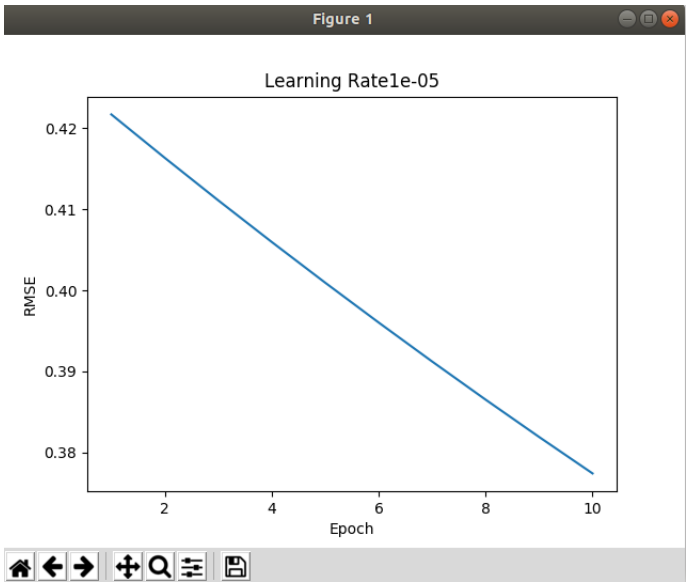
Q1- Gradient Descent Linear Regression

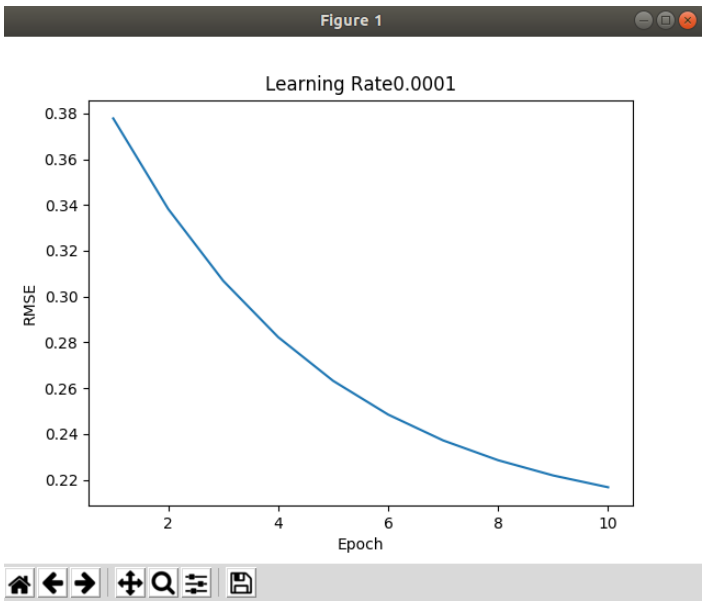
Gradient Descent was performed on Boston Dataset in the SciKitLearn python package.

Training Data was used to calculate the coefficients.

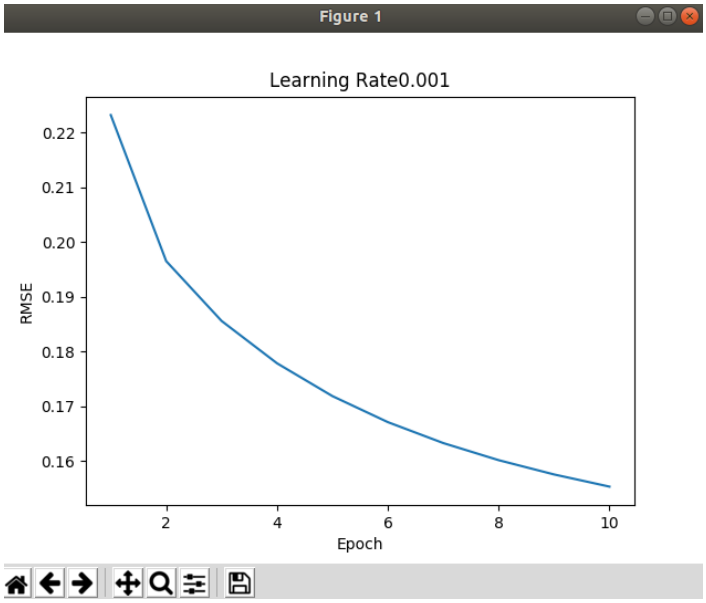
Test Data set was used to calculate the RMSEs [Commented block is present in the .py file to calculate RMSEs based on Training Dataset itself].



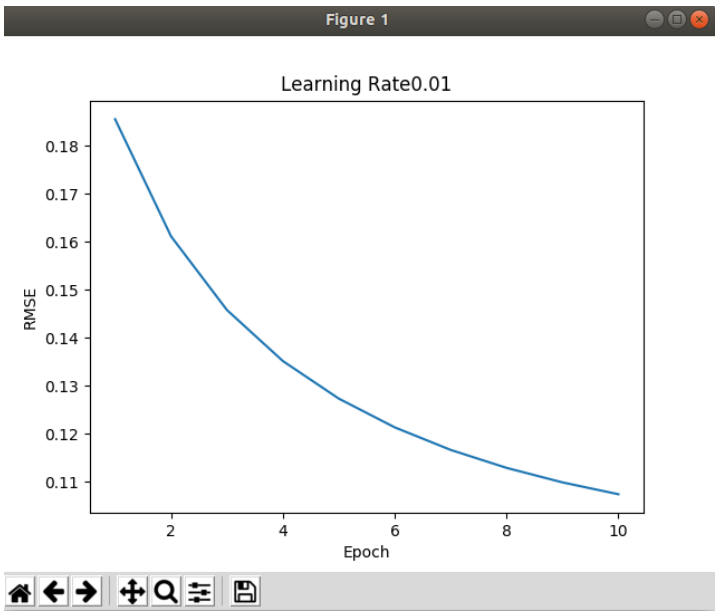

```
RMS Errors FOR LEARNING RATE 1e-05 are [0.4217033608739931, 0.4163467819875731,
0.4111037491907191, 0.4059723773652679, 0.40095080552637985, 0.3960371962468596,
 0.39122973510063386, 0.38652663012574623, 0.3819261113072146, 0.377426430080065
2]
```



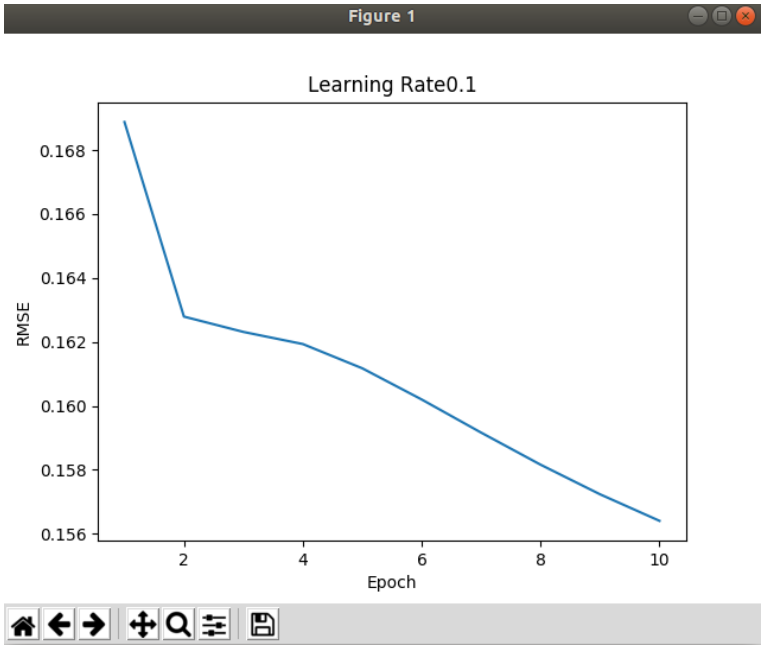

```
RMS Errors FOR LEARNING RATE 0.0001 are [0.3778047859545601, 0.3382402082509205,
 0.3068737823382115, 0.28227370619902653, 0.2631692009058361, 0.2484481012592850
1, 0.2371588283925924, 0.22850923265549525, 0.22185835927843847, 0.2167011919773
7356]
```



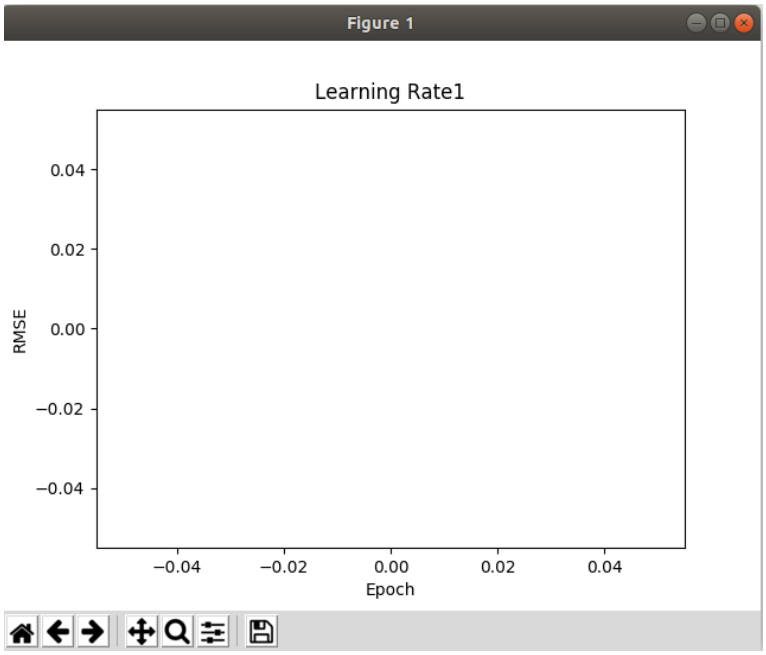

```
RMS Errors FOR LEARNING RATE 0.001 are [0.22314936256137788, 0.1965058907334597,
 0.1855924459445445, 0.17787111549091691, 0.17188231017483688, 0.167135828393067
65, 0.16331871553703778, 0.16019678589790268, 0.15759198438025396, 0.15537018308
300754]
```




```
RMS Errors FOR LEARNING RATE 0.01 are [0.18556262037111848, 0.16118141261616759,
 0.1458266315431111, 0.13520423301801807, 0.12738155866784104, 0.121394539649013
64, 0.11669988159396281, 0.11295672784193658, 0.10993370521631396, 0.10746535055
01876]
```




```
RMS Errors FOR LEARNING RATE 0.1 are [0.16887712347406142, 0.16279223125911602,
0.1623135272412272, 0.1619342995698029, 0.16117603747910833, 0.1602000154940262,
 0.15916448492785223, 0.15816154411595382, 0.1572361997374733, 0.156406489055064
```




```
2.06135110862e+307
-3.85164092118e+307
7.76127309772e+307
-1.74483864874e+308
Q1Gradient_Descent Complete.py:105: RuntimeWarning: overflow encountered in doub
le_scalars
  return b0 + sum(terms)
inf
Q1Gradient_Descent Complete.py:143: RuntimeWarning: invalid value encountered in
 double_scalars
  b[j] = b[j] - learning_rate * error * train[i][0][j]
nan
nan
nan
nan
nan
```

- From the graphs we saw that Root Mean Square Error values depend heavily on the Learning Rate we choose. We can note from the graphs that **as we decrease the learning rate**, the RMSE value **gets better much faster**.
- Also, Learning Rate=0.1 has the lowest RMSE values.
- Perhaps, a point to note is the fact that when we choose **Learning Rate =1.0**, the model **actually gets worse**, and results in huge error. This error when reaches e+308 and above, results in **INF values**, and hence the coefficient which are calculated using these errors result in **NAN values**. Therefore, there is no plot for Learning Rate = 1.0

```
Hope you liked the graphs. Now we can analyze and choose the best learning_rate!
!
WOAHHH it is 3:30 am, let's just call it a night!
Goodnight World!!!!
```