CS 6375

ASSIGNMENT 1: Linear Regression using Gradient Descent

Names of students in your group:

1. Ameya Kulkarni (ANK190006)

2. Yamini Thota (YRT190003)

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.

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

- We haven’t used any resources directly.

Part 1:

Dataset used:

<https://archive.ics.uci.edu/ml/datasets/Metro+Interstate+Traffic+Volume>

Note: For the purpose of code execution we have already provided dataset file,you need not download it.

Plot for each input feature to the output:

Independent Features are holiday, temp,rain\_1h,snow\_1h, clouds\_all and weather\_main.

Dependent or target feature is traffic\_volume

1. Before feature scaling

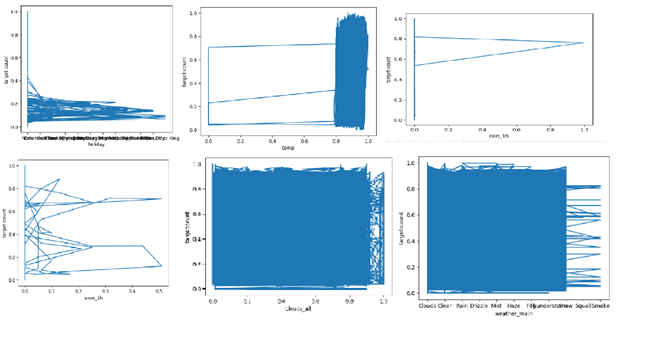


Fig1

2. After feature scaling

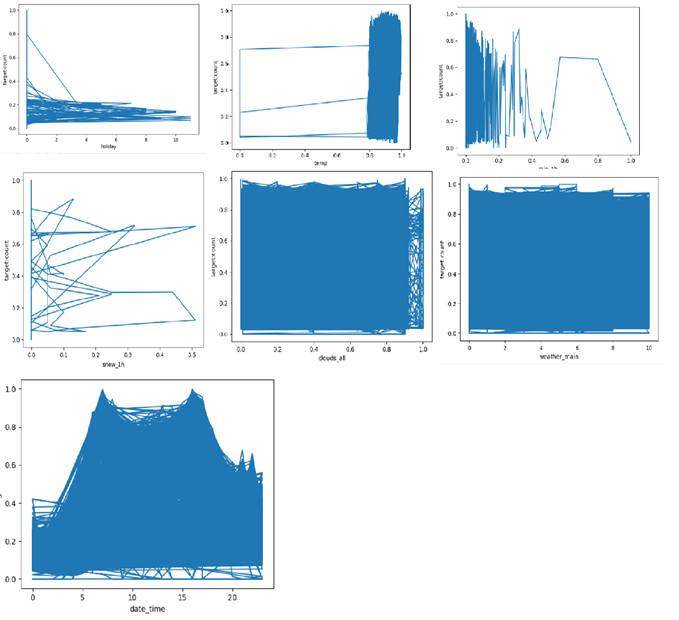


Fig 2

Pre-processing activities includes :

- We don’t have any rows with null values.

- We don’t have any redundant rows, verified this using “drop\_duplicates()”

- Removed **weather\_description** column as we already have **weather\_main**

- Weused **Min-Max Scaling** for scaling **temp, rain\_1h, clouds\_all** and **traffic\_volume**

**- Dates** like this ”10/2/2012 9:00:00 AM” to hour like “9” of the day because we generally need to find traffic volume based on hours.

- We catogrized **holiday** and **weather\_main** features

df['weather\_main'] = df['weather\_main'].map({

'Clouds': 0,

'Clear': 1,

'Drizzle': 2,

'Fog': 3,

'Haze': 4,

'Mist': 5,

'Rain': 6,

'Smoke': 7,

'Snow': 8,

'Squall': 9,

'Thunderstorm': 10,

})

df['holiday'] = df['holiday'].map({

'None': 0,

'Columbus Day': 1,

'Veterans Day': 2,

'Thanksgiving Day': 3,

'Christmas Day': 4,

'New Years Day': 5,

'Washingtons Birthday': 6,

'Memorial Day': 7,

'Independence Day': 8,

'State Fair': 9,

'Labor Day': 10,

'Martin Luther King Jr Day': 11

})

Observations:

How we got below observations:

- We ran our code with epoch range from 1000 to 1 reducing epoch by 20% every time.

- For each epoch we changed the learning rate from 0.001 to 0.010 with a step of 0.001.

- We got total number of 270 observations

- On that we applied sorting to get a range of MSE on training data in which 0.193553234889187 is minimum MSE of training data.

- We considered 0.05 as a tolerating range to get a minimum epoch and maximum learning rate.

- The table below shows MSE values in range (0.19 to 0.25)

- We feel that highlighted rows can be the best values of epoch and learning rates.

Recording were taking with same starting weights

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Index | Epochs | Learning Rate | MSE Training Data | MSE Testing data |
| 199 | 12 | 0.009 | 0.24955098 | 0.237985157 |
| 198 | 12 | 0.008 | 0.25529565 | 0.243804941 |
| 189 | 16 | 0.009 | 0.23630345 | 0.224245793 |
| 188 | 16 | 0.008 | 0.24095059 | 0.229016238 |
| 187 | 16 | 0.007 | 0.24731688 | 0.23557902 |
| 186 | 16 | 0.006 | 0.25589799 | 0.244425476 |
| 179 | 20 | 0.009 | 0.23011224 | 0.217891765 |
| 178 | 20 | 0.008 | 0.23314507 | 0.22099012 |
| 177 | 20 | 0.007 | 0.23763966 | 0.225608025 |
| 176 | 20 | 0.006 | 0.24425232 | 0.23242054 |
| 180 | 20 | 0.01 | 0.24986636 | 0.239225368 |
| 175 | 20 | 0.005 | 0.25387417 | 0.242340392 |
| 169 | 26 | 0.009 | 0.22611866 | 0.213869919 |
| 168 | 26 | 0.008 | 0.22762091 | 0.21536825 |
| 170 | 26 | 0.01 | 0.22793378 | 0.215984987 |
| 167 | 26 | 0.007 | 0.23009179 | 0.217871243 |
| 166 | 26 | 0.006 | 0.23418695 | 0.222060797 |
| 165 | 26 | 0.005 | 0.24095276 | 0.229021169 |
| 164 | 26 | 0.004 | 0.25199203 | 0.24040105 |

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

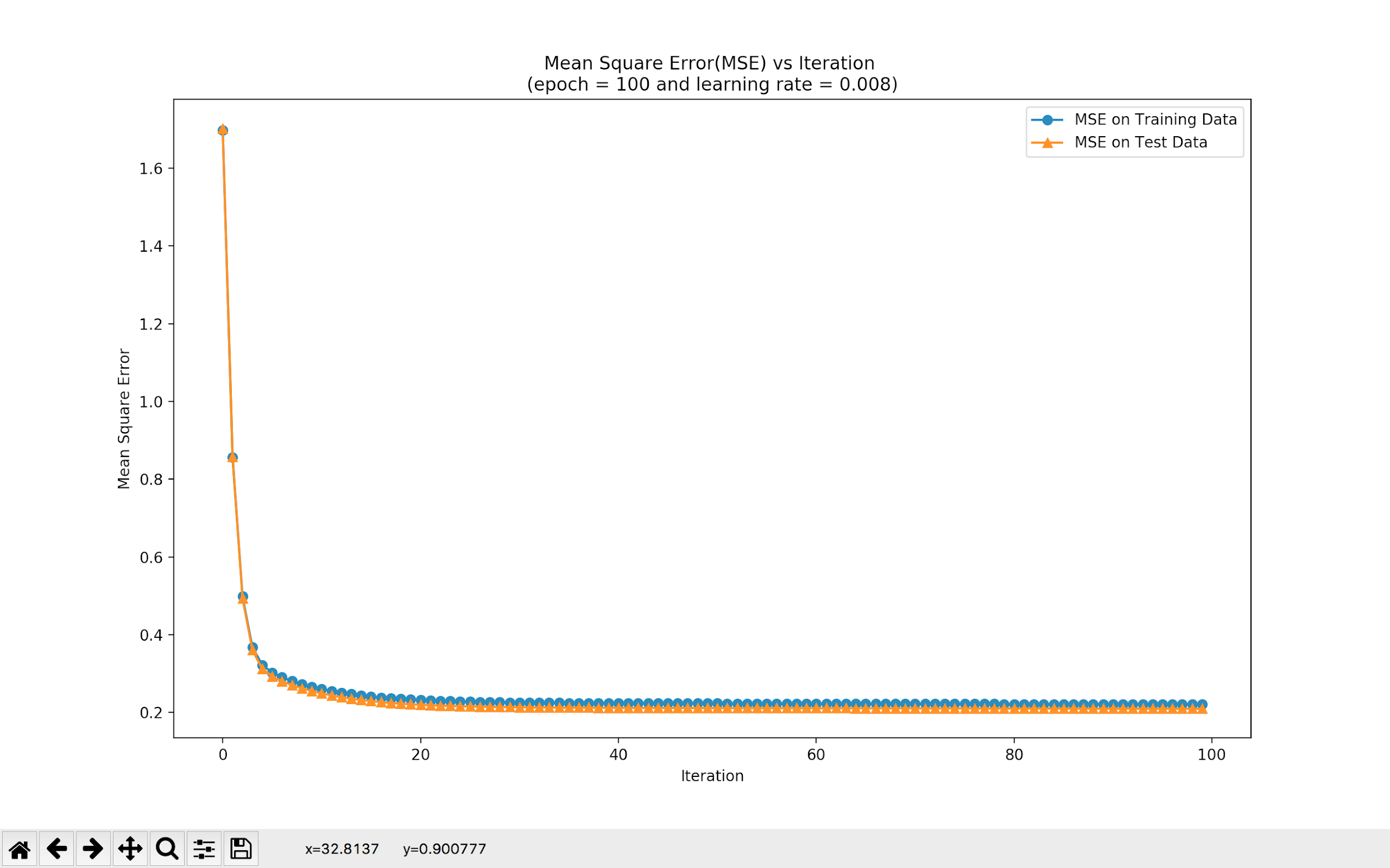


Fig 3

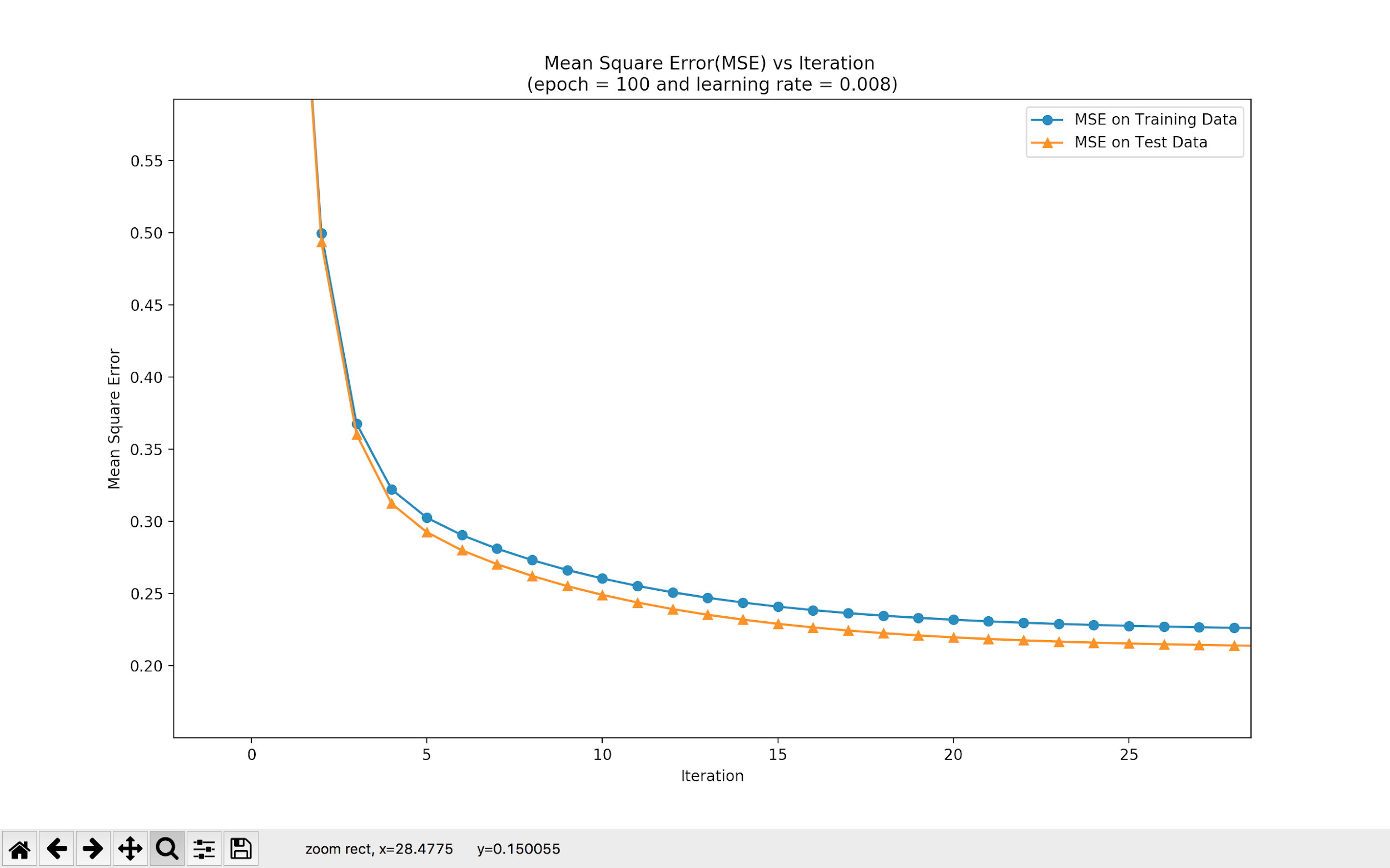


Fig 4

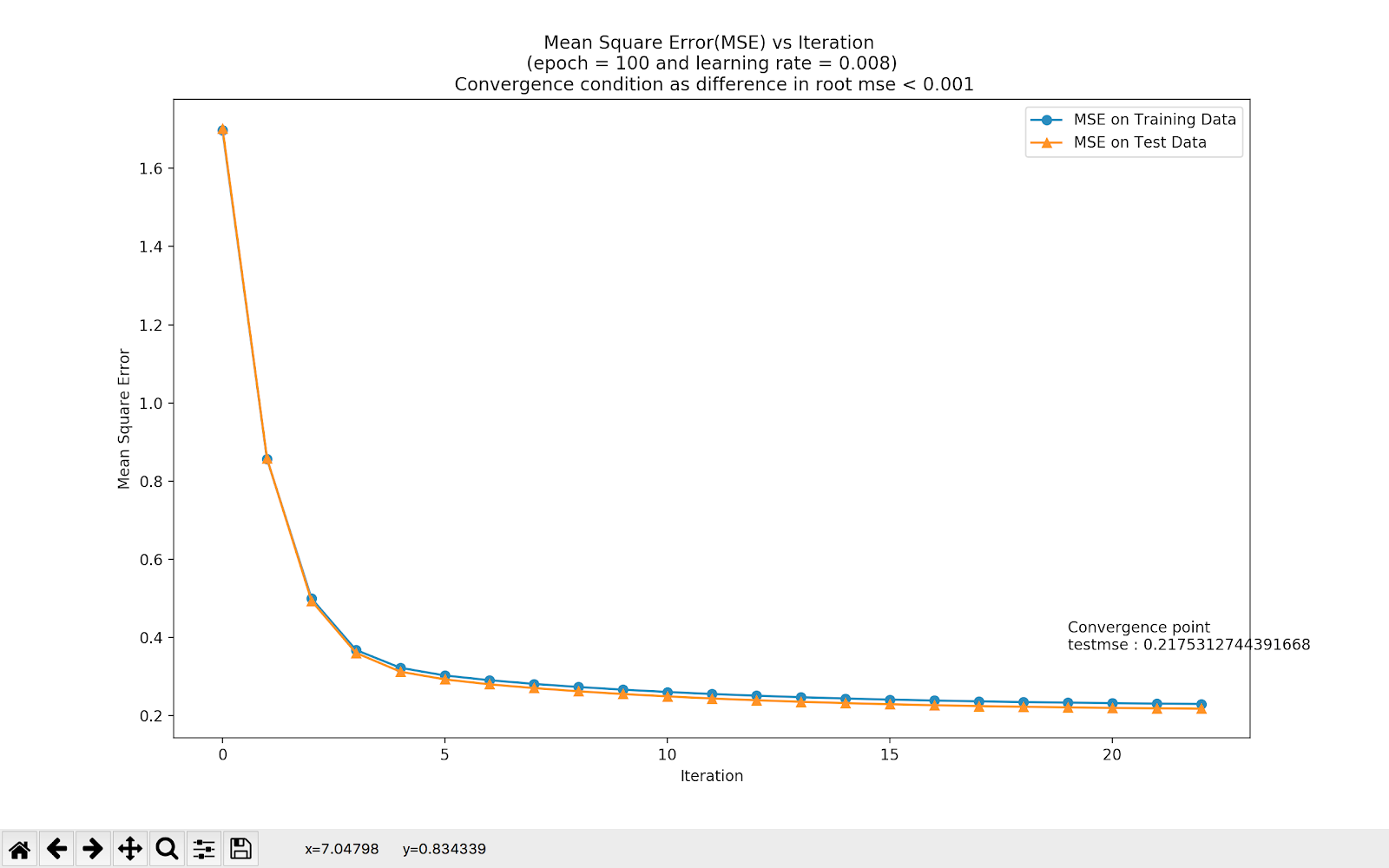


Fig 5

How Satisfied?

* We are satisfied with our values. We trained our model with different epochs and learning rates and recorded observations. From these observations we checked for minimum MSE on training data and picked best values for epoch and learning rate.
* Fig 3 shows output for 100 epochs with learning rate of 0.008. From the graph it is clear that the model is converging somewhere near iteration number 20-25. Rest of the iterations are not required. Fig 4 is a zoomed version of fig 3
* To find this convergence point we added a condition to keep difference in error more than 0.001. Training will stop if the difference in error is less than 0.001. Fig 5 shows that training stopped at convergence point.

Part 2:

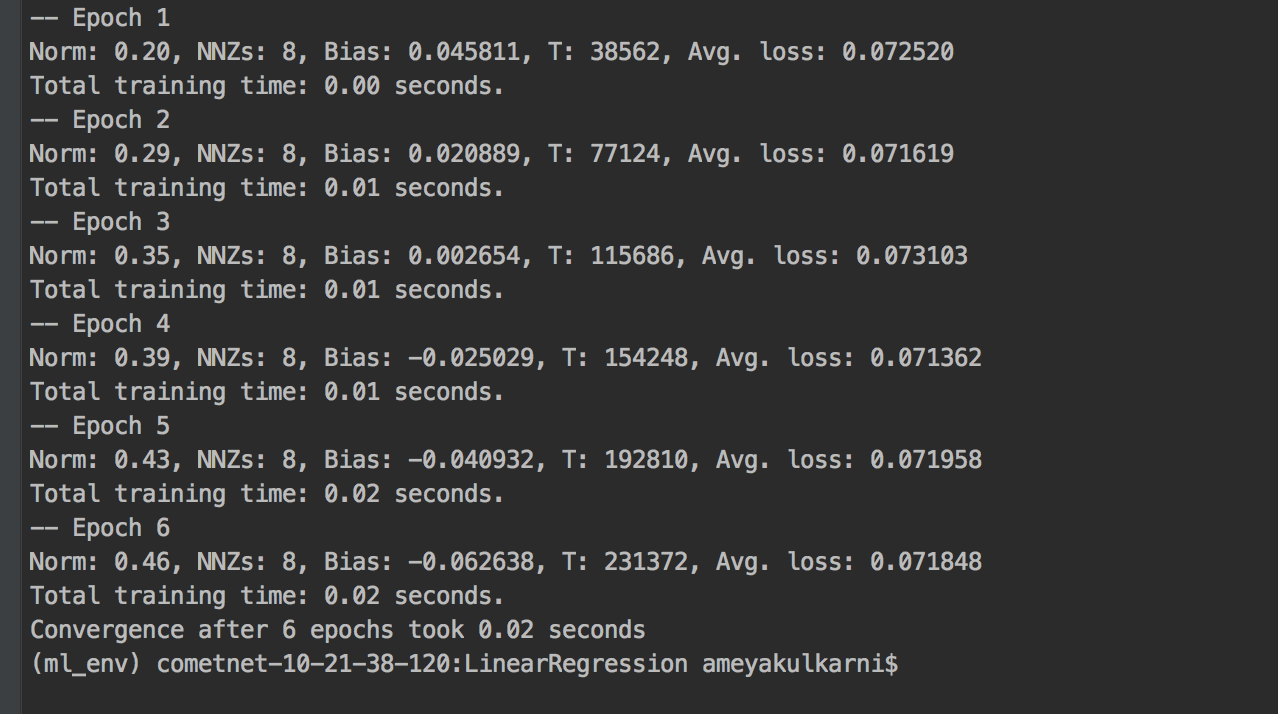
* We used the SGD linear regression package.
* We applied same logic as part 1 to get the observations.

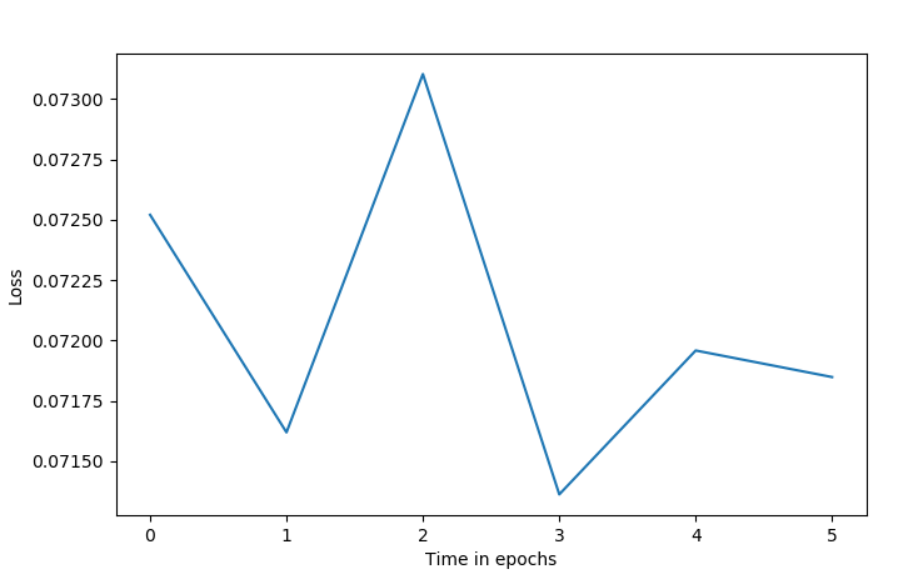
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Index | Epochs | Learning Rate | MSE Training Data | MSE Testing data |
| 131 | 53 | 0.001 | 0.254060491 | 0.258383816 |
| 1 | 1000 | 0.001 | 0.254148093 | 0.272112047 |
| 261 | 1 | 0.001 | 0.254416362 | 0.255711369 |
| 222 | 5 | 0.002 | 0.25441734 | 0.26322152 |
| 105 | 105 | 0.005 | 0.254566409 | 0.305966067 |
| 181 | 16 | 0.001 | 0.255023464 | 0.265665126 |
| 103 | 105 | 0.003 | 0.255304489 | 0.292813149 |
| 81 | 166 | 0.001 | 0.256135017 | 0.255118766 |
| 91 | 132 | 0.001 | 0.256341374 | 0.295609517 |
| 211 | 7 | 0.001 | 0.256353388 | 0.254889851 |
| 163 | 26 | 0.003 | 0.256754949 | 0.25831221 |
| 172 | 20 | 0.002 | 0.256846168 | 0.265325789 |
| 144 | 42 | 0.004 | 0.256873059 | 0.289084395 |
| 36 | 512 | 0.006 | 0.257148897 | 0.265194518 |
| 124 | 67 | 0.004 | 0.257982637 | 0.387657682 |
| 21 | 640 | 0.001 | 0.258289628 | 0.254007627 |
| 82 | 166 | 0.002 | 0.259437483 | 0.300478048 |
| 204 | 9 | 0.004 | 0.259596137 | 0.269058609 |

- We feel that highlighted rows can be the best values of epoch and learning rates.

- We used number of epochs = “9” & learning rate = “0.004” and got below values.

Plot of Average Loss and number of iterations in scikit learn with SGDClassifier





How Satisfied?

We are happy that we got similar values in comparison with part1.

We have a concern with SDG Classifier as fit function is not returning any list to give MSE at each iteration which can be used later to plot graph.

We had to use custom code to plot graph for iterations and average loss.