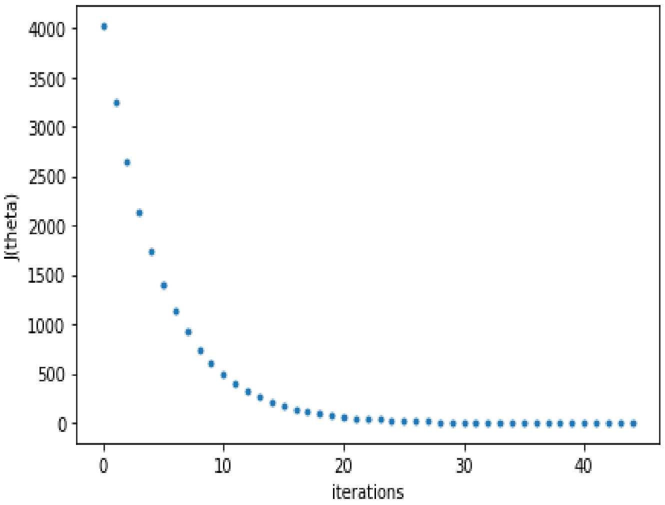
**NOTE:** To run the code for all questions : please go to respective question folders, there will be only one-main file containing implementation of sub sections - change the “input” and “output” path in the script - learned parameters will be printed on terminal and the plots will be saved with respective sub-question name in “output” directory. Below sections includes experimentation and theoritical answers asked per question.

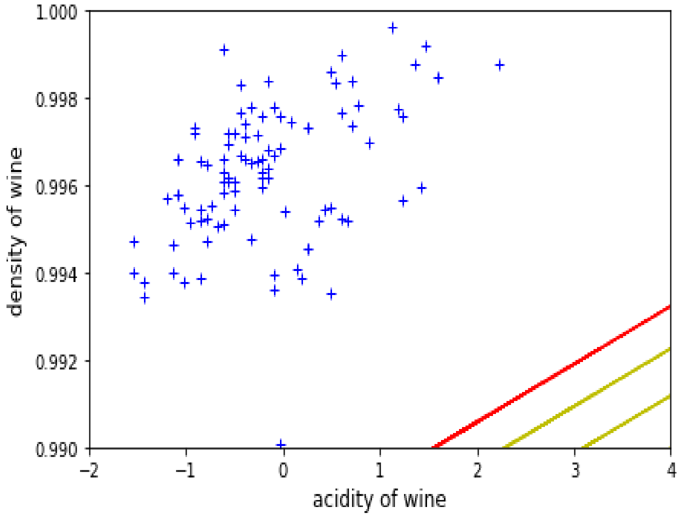
## QUESTION 1

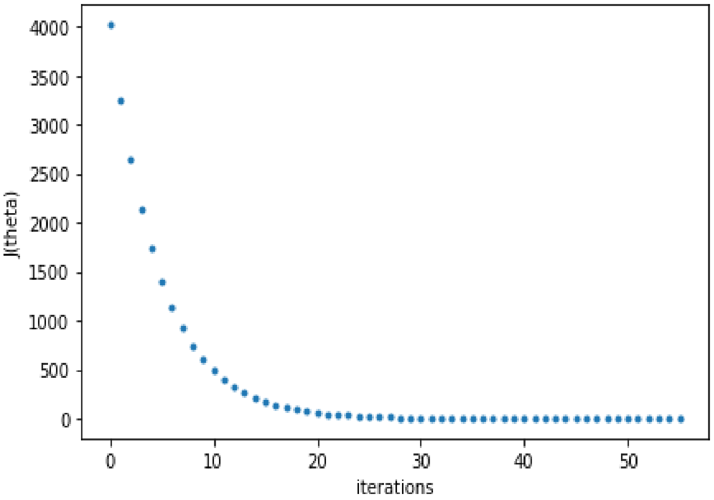
### A) Deciding stopping criteria and learning rate.

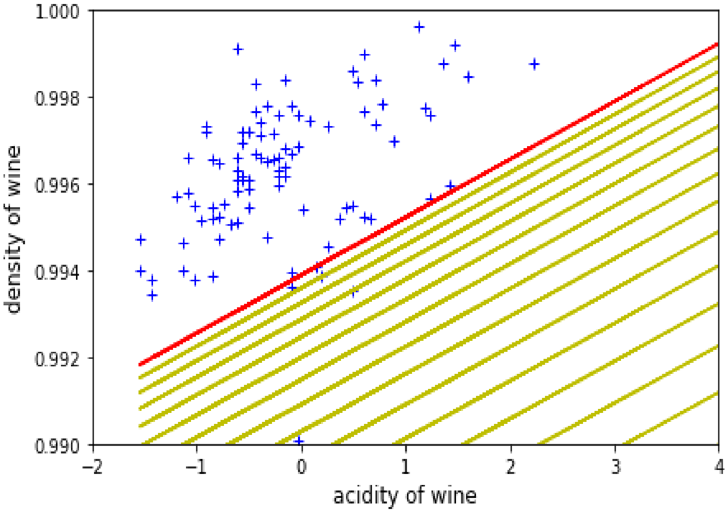
- Stopping criteria is defined by a threshold which is equal to the difference between loss of current iteration and previous iteration.
- Here I tried varying stopping threshold with fixed value of learning rate i.e. 0.01.
- Below mentioned is the table of learned parameters (rounded to 4 decimal)

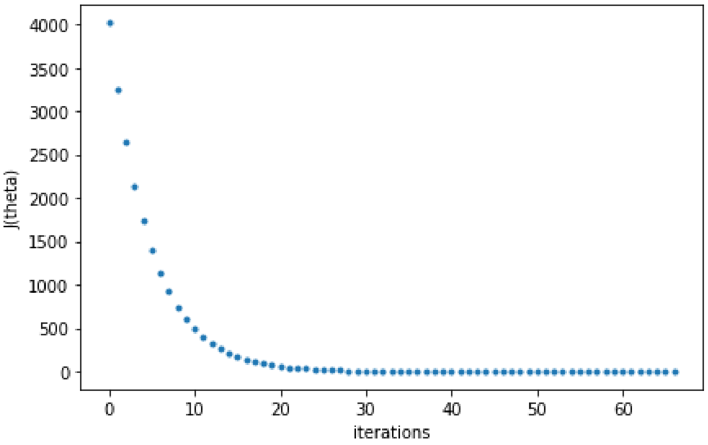
| **Stopping_threshold** | **Theta0** | **Theta1** | **Final cost/MSE** | **Total iterations** |
|---|---|---|---|---|
| 0.1 | 0.9879 | 0.0013 | 0.3903 | 45 |
| 0.01 | 0.9939 | 0.0013 | 0.0492 | 56 |
| 0.001 | 0.9958 | 0.0013 | 0.0156 | 67 |
| 0.0001 | 0.9964 | 0.0013 | 0.0123 | 78 |
| 1.00E-05 | 0.9965 | 0.0013 | 0.012 | 89 |

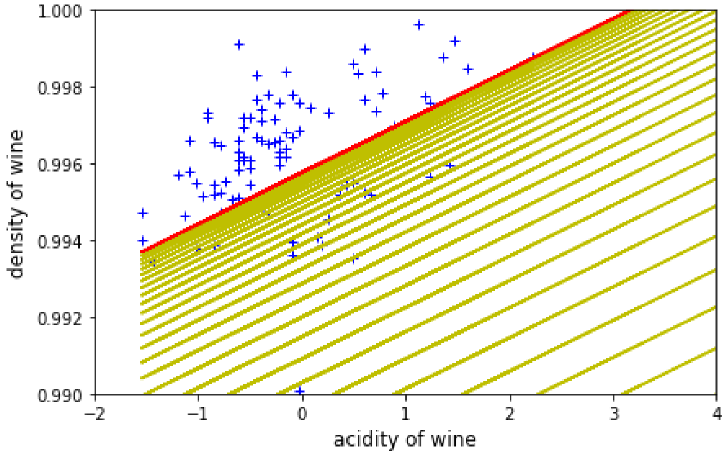

4000
3500
3000
2500
2000
1500
1000
500
0
J(theta)
0
10
20
30
40
iterations

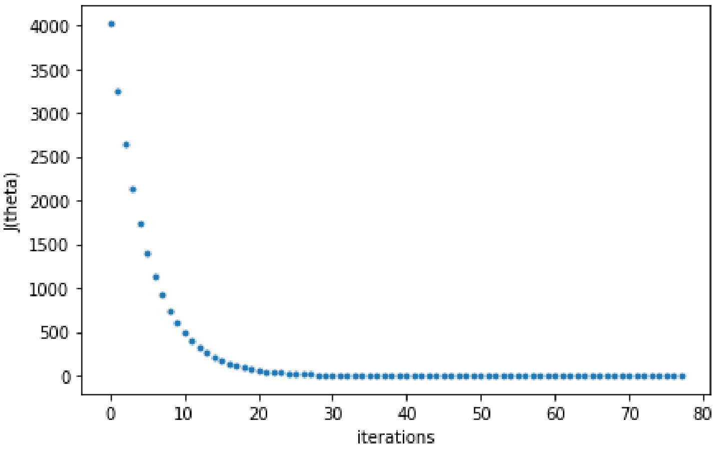

1.000
0.998
0.996
0.994
0.992
0.990
density of wine
−2
−1
0
1
2
3
4
acidity of wine

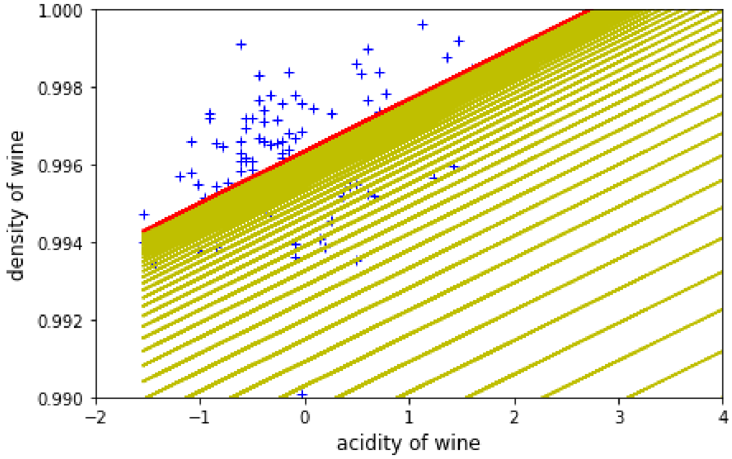

4000
3500
3000
2500
2000
1500
1000
500
0
J(theta)
0
10
20
30
40
50
iterations

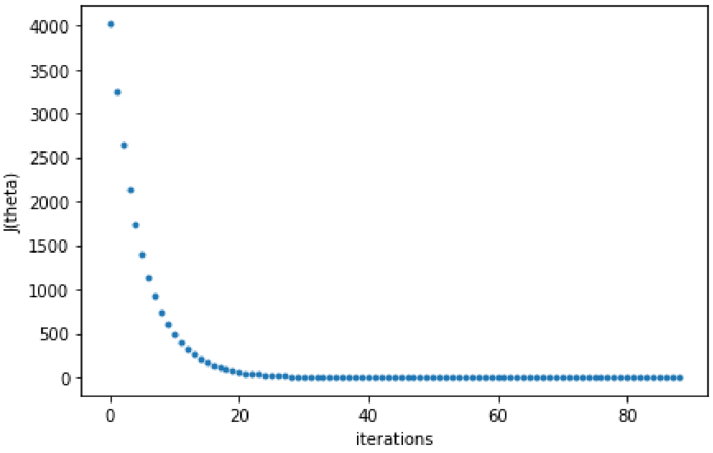

1.000
0.998
0.996
0.994
0.992
0.990
density of wine
−2
−1
0
1
2
3
4
acidity of wine

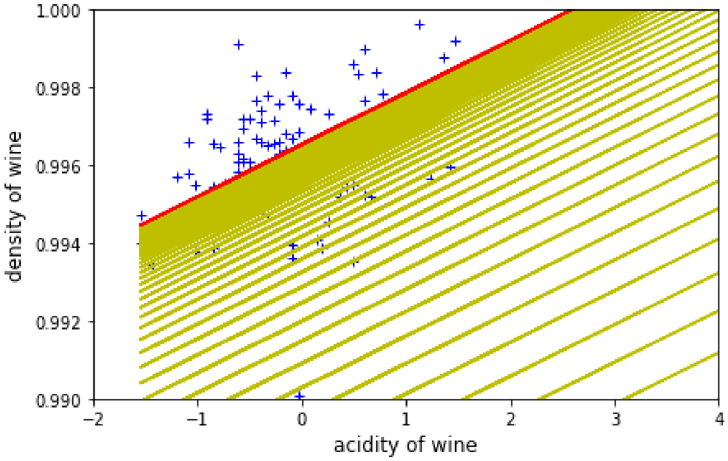






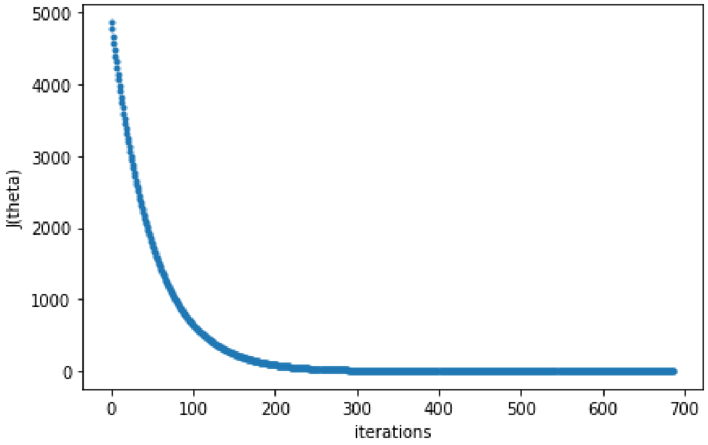


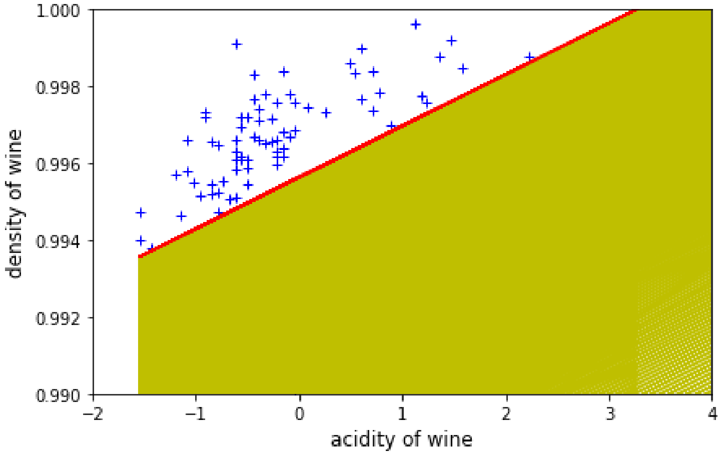


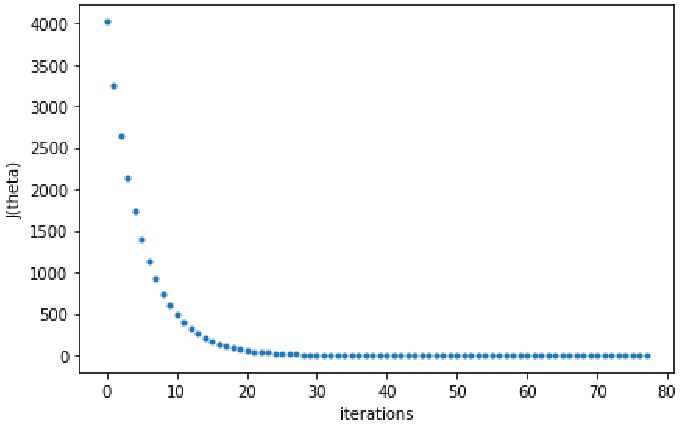


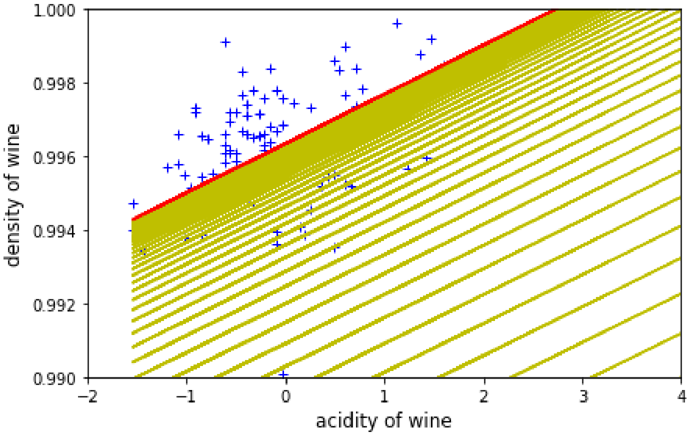
- From this the stopping threshold of 0.0001 seems the best - as there is not much diff between 0.0001 and 0.00001 in MSE while other bigger values do not fit well (can be observed visually better since data values are small)
- Now to identify appropriate learning late - stopping criteria was fixed.

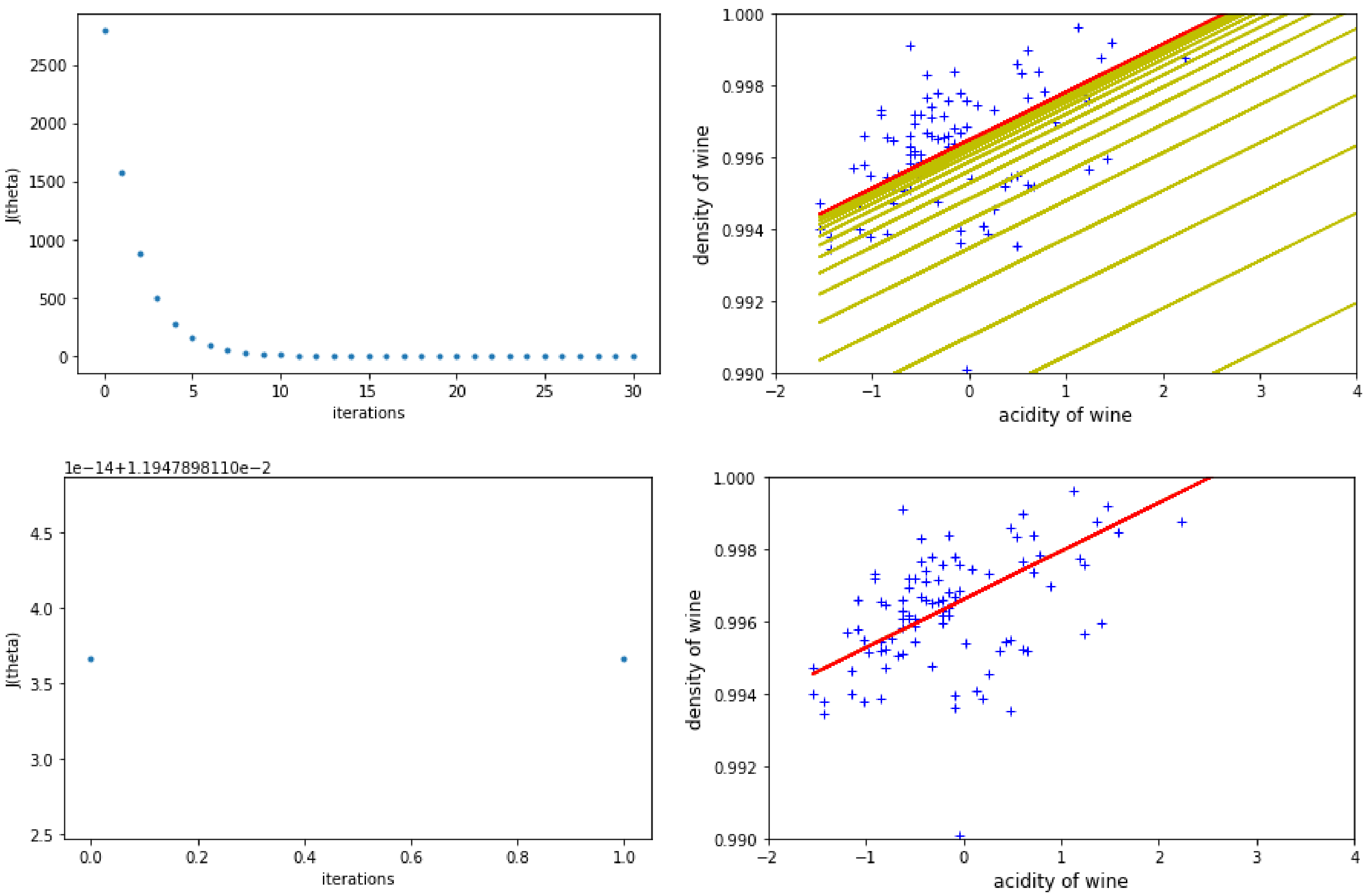
| **Learning_rate** | **Theta0** | **Theta1** | **Final cost/MSE** | **Total iterations** |
|---|---|---|---|---|
| 0.001 | 0.9956 | 0.0013 | 0.0168 | 688 |
| 0.01 | 0.9964 | 0.0013 | 0.0123 | 78 |
| 0.025 | 0.9965 | 0.0013 | 0.012 | 31 |
| 0.1 | 0.9966 | 0.0013 | 0.0119 | 2 |









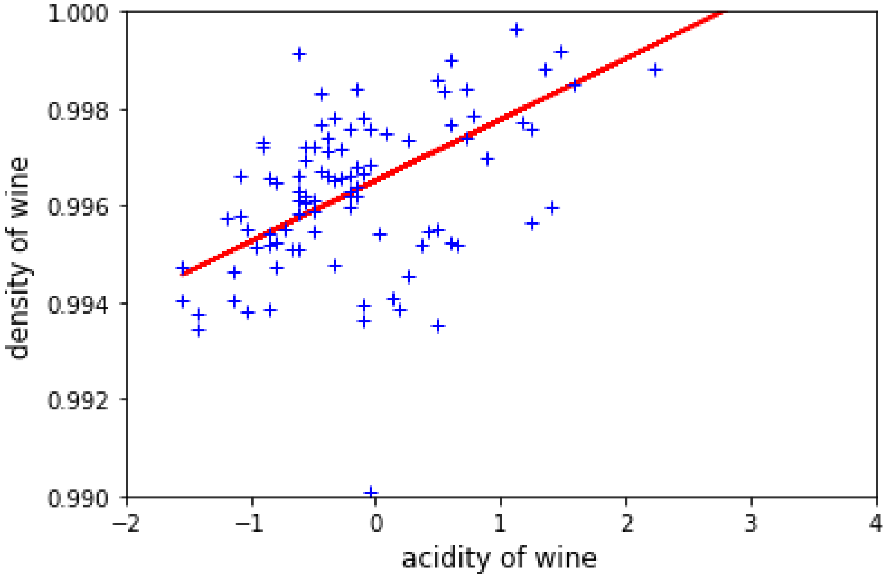


- Best learning rate : 0.025 (diff between 0.025 and 0.01 is very less in terms of MSE and fit, but is comparatively bigger in terms of #iterations)

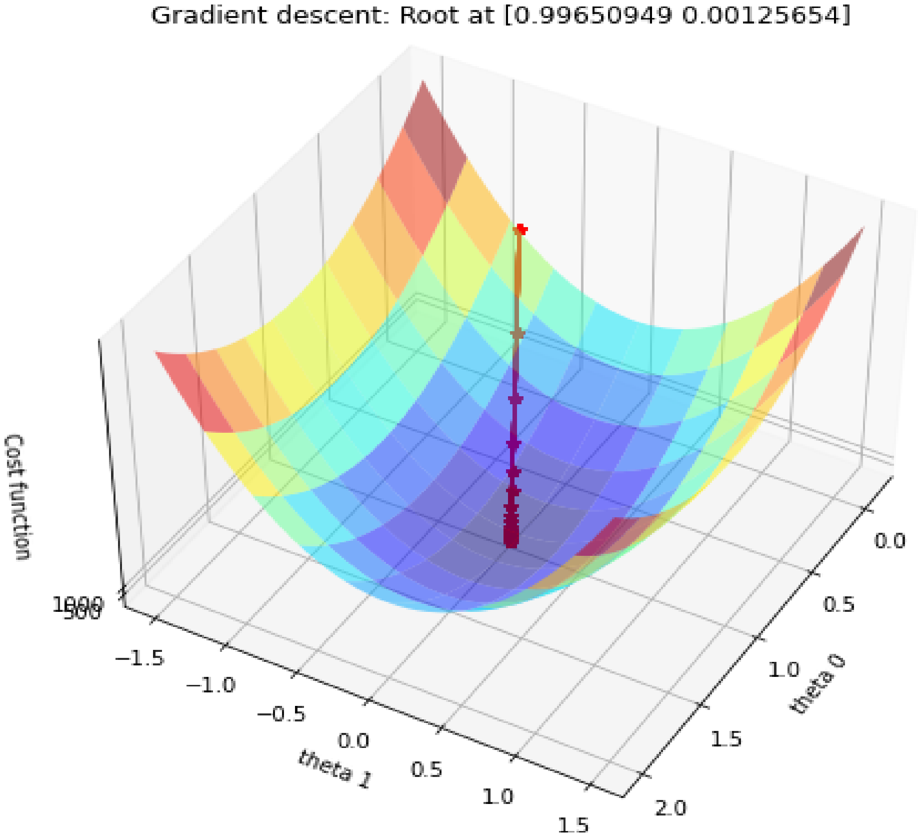
B) Learned parameters:

   a) Theta0 : 0.99650949
   b) Theta1: 0.99650949

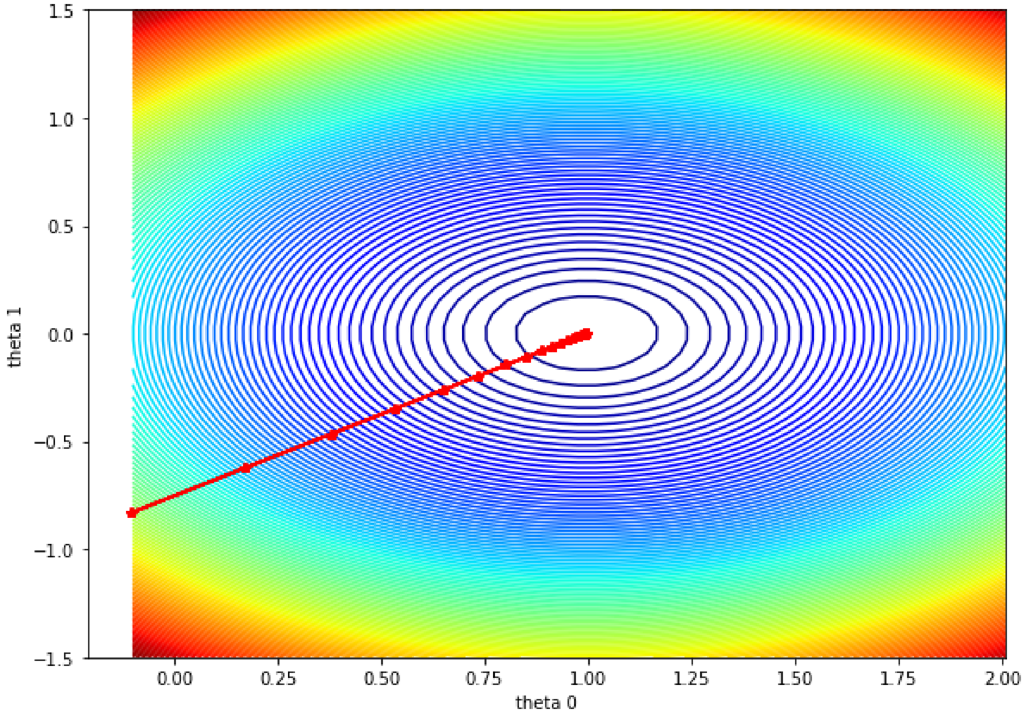
c) Final cost/MSE: 0.0120440596116729



C) 3D Mesh

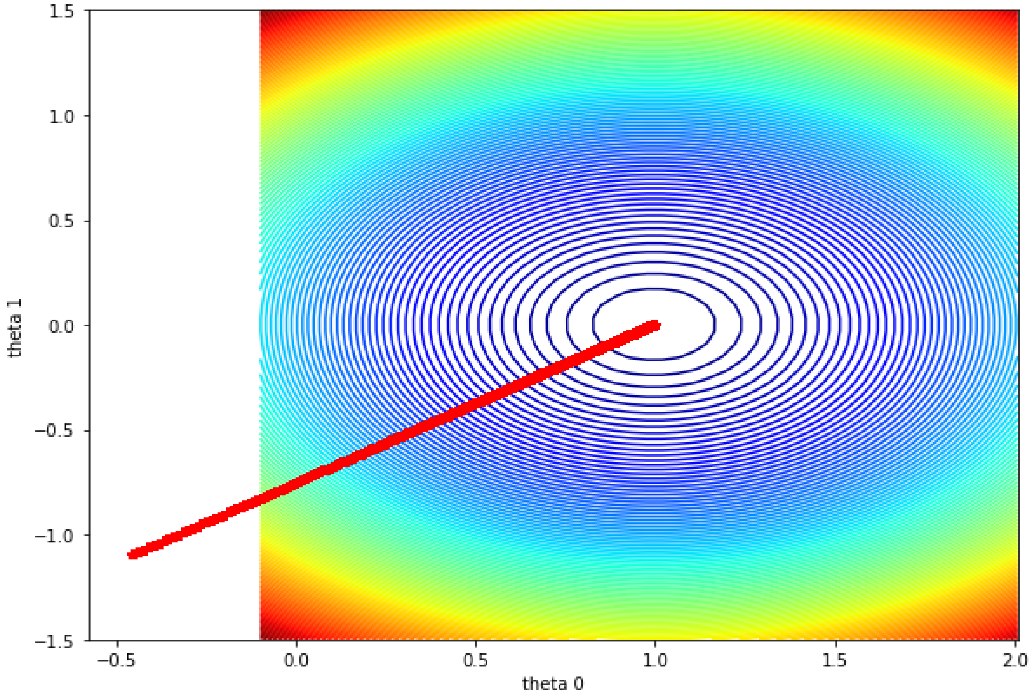


D) Contours - 0.025

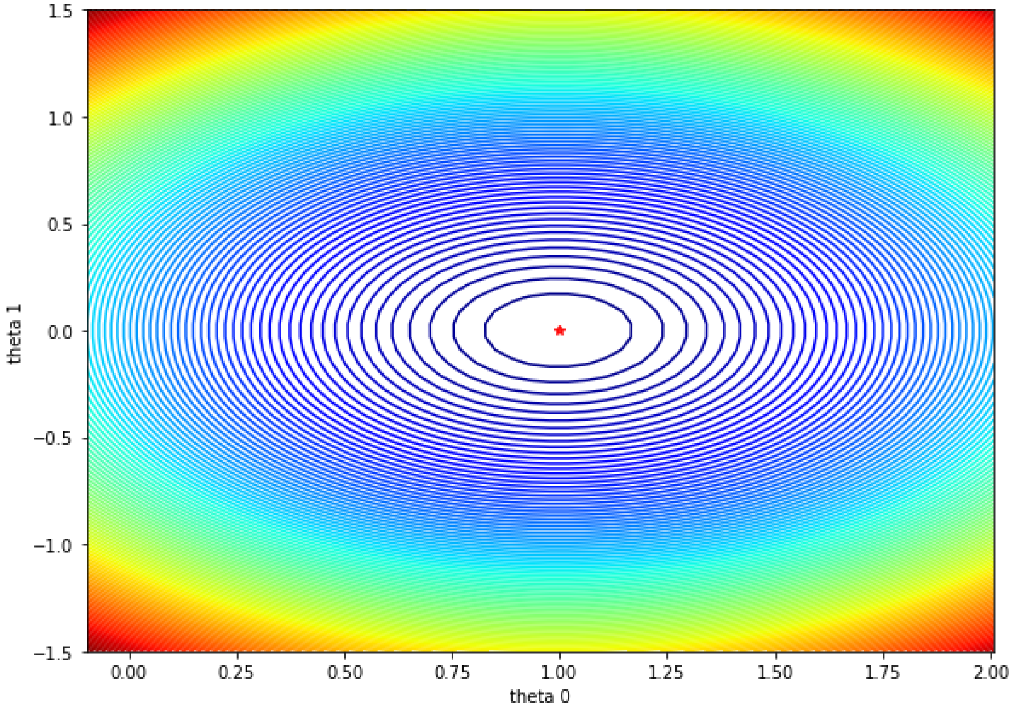


E) Contours

a) 0.001

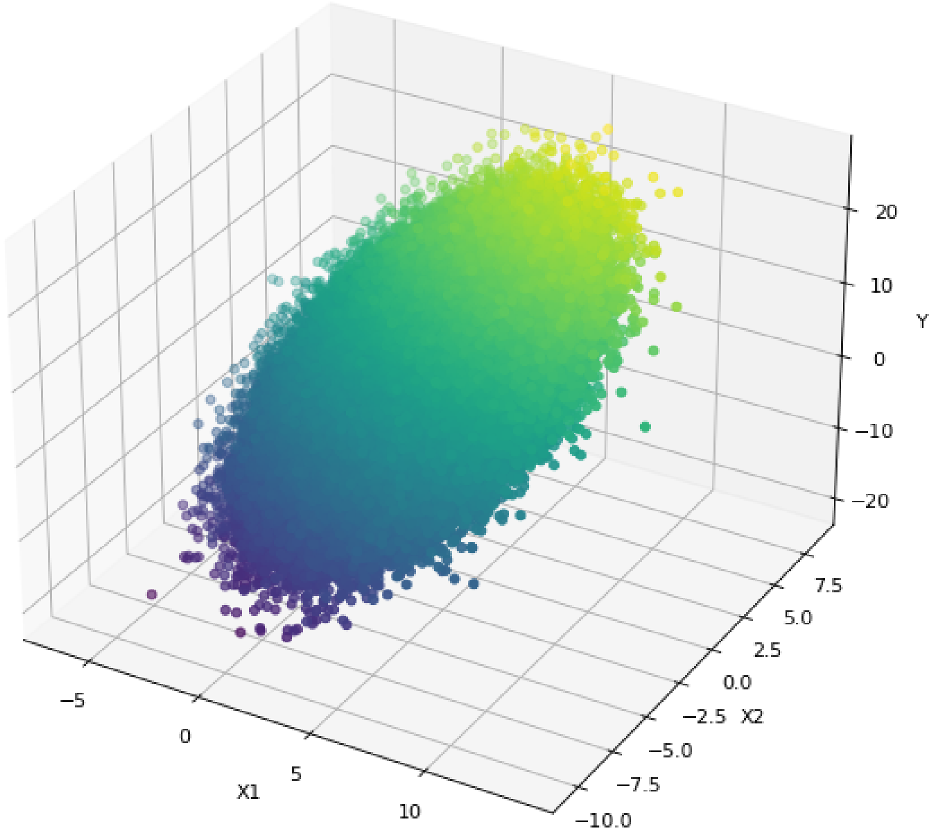


b) 0.1


1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
theta 1
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
theta 0

# **QUESTION 2**

1. Sampling Data Points



2. Convergence Criteria
    a. Here in min-batch gradient descent (stochastic gradient descent, when batch size=1) for different batch size the stopping criteria will be different.

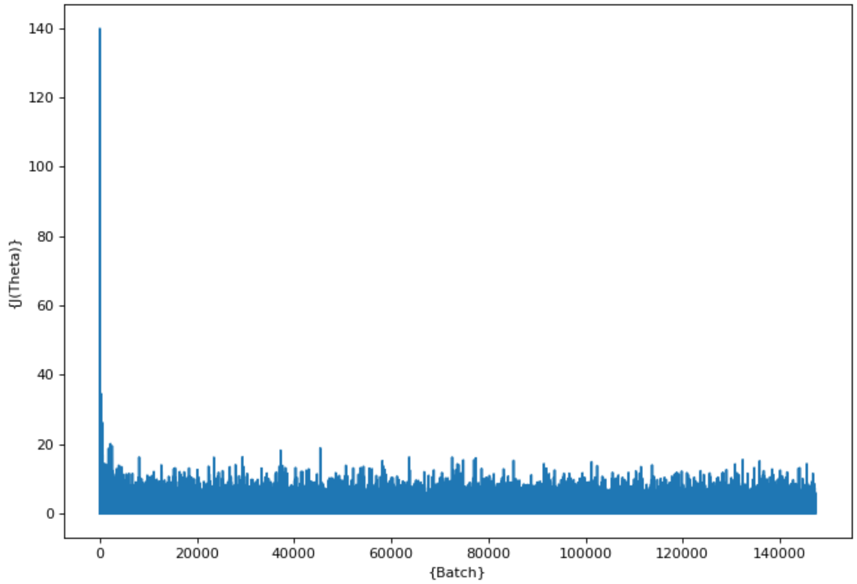
b. As in Batch Gradient Descent, we computed cost diff of whole batch between two iterations, here we compute difference of cost between current batch and average of previous “k” batches.

c. Choosing k depends on batch size.

    i. For batch size = 1, I experimented over different k and stopping criteria.

    ii. Observation : relatively smaller values of k(eg: k=100) converges faster and better than larger values(eg k=1000)

    iii. Also the value of cost falls to smaller range in initial iterations and remains the same further as well, which implies that the model fits with only few number of data points ( but this might not be the case in non-normal distributions).



    iv. So model fits best after around 1437 mini-batches i.e. (143700 data points).

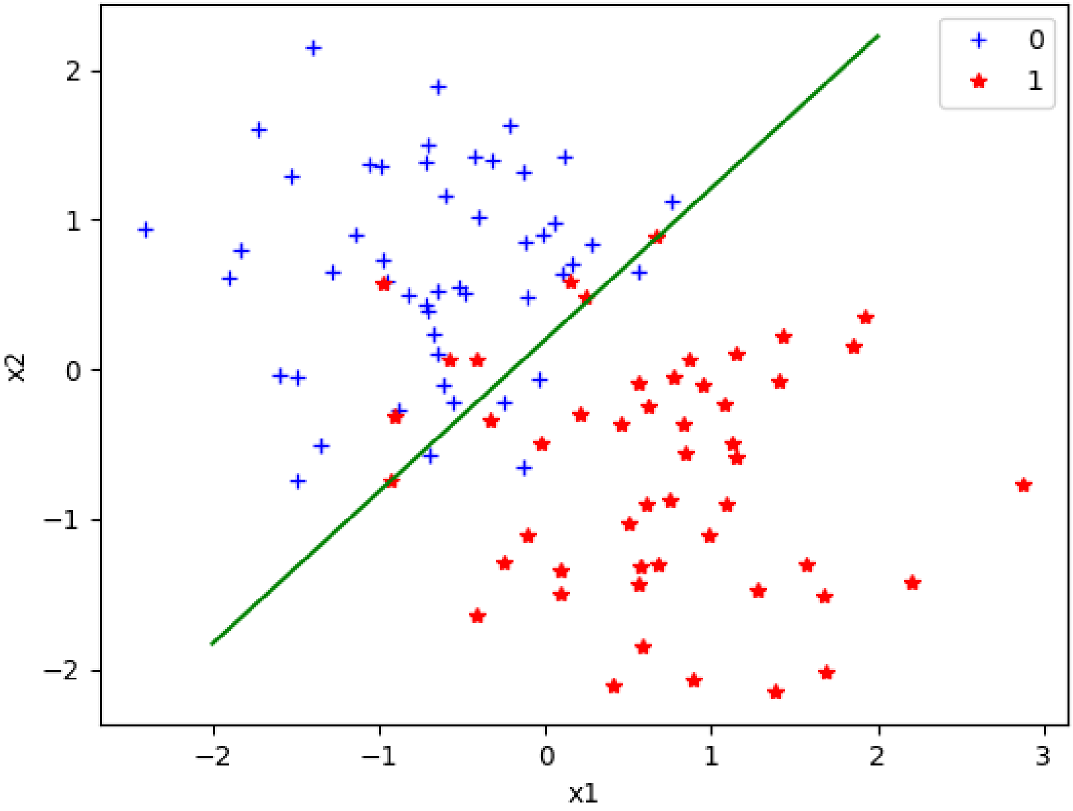
d. Other convergence method can be to compute difference of cost between two batches of size k i.e. (N-2*k:N-k) and (N-k:N) - this worked well for higher batch sizes.

e. Experimented on various iterations for batch sizes to converge to decide on single value.

f. Observations : Higher batch sizes take longer to converge on same criteria. The learning for batch size 1 is faster because of the underlying data distribution which is normal so model fits after seeing about 100000 samples and then converges.

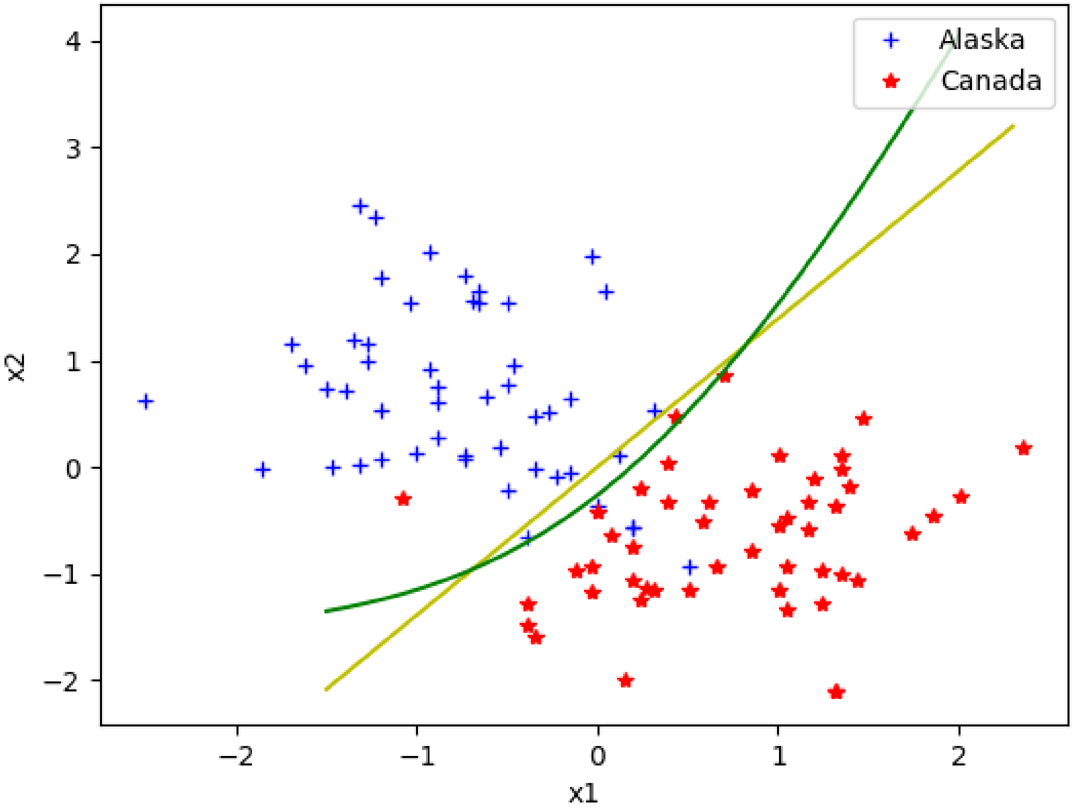
## QUESTION 3

- Newtons method was applied directly.
- Convergence was observed to be much faster.



# QUESTION 4

- All the plots and parameter details are mentioned with code.
- Both linear and quadratic boundary was plotted as below



- It can be observed that the quadratic boundary fits better than the linear boundary.