IA1 - Group 26

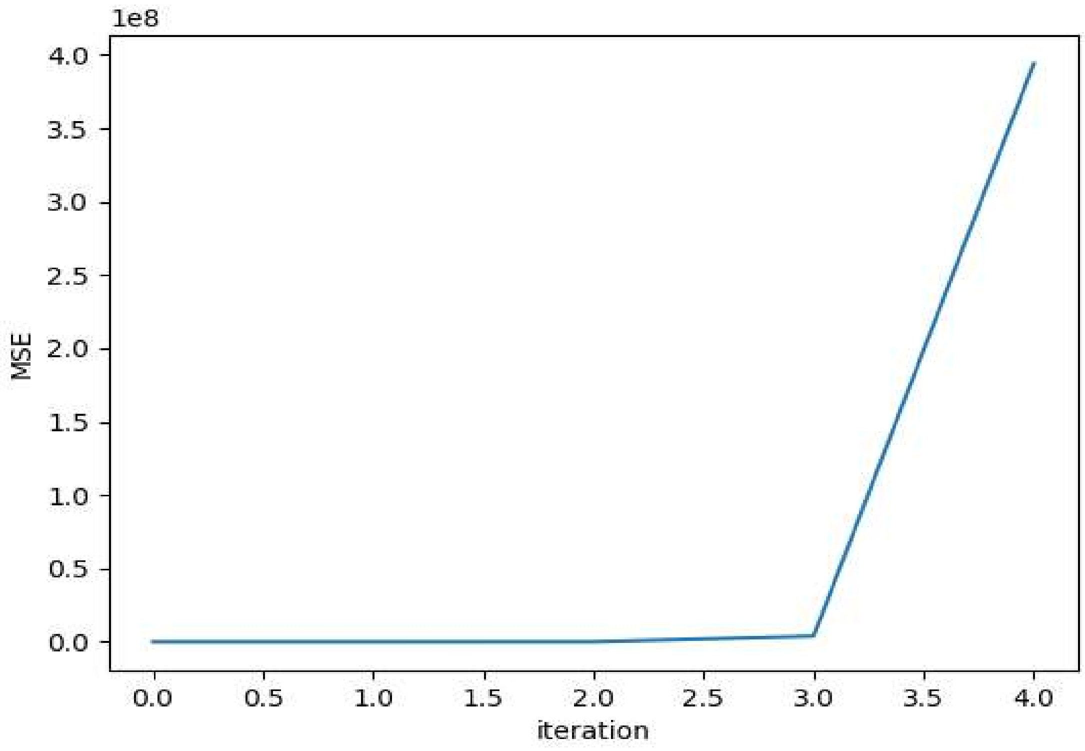
Cheng Zhen
Bharath Padmaraju
Bharghav Srikhakollu

We tested all of our code in **flip1 server**

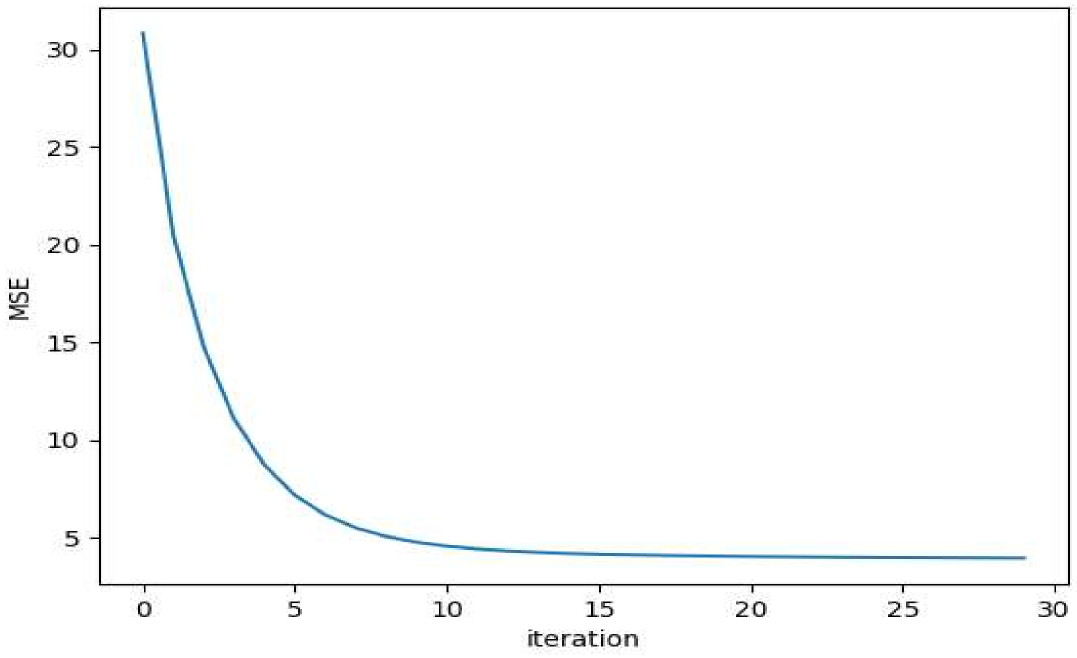
Part - 1:

a) Plot the MSE as a function of iterations for each learning rate.

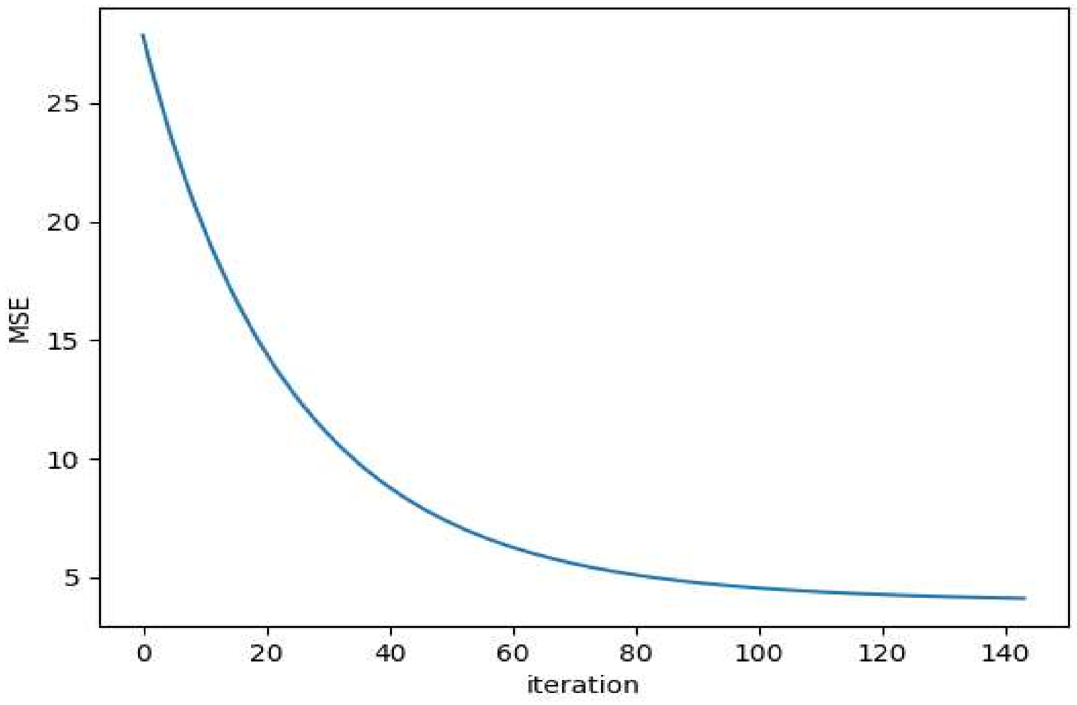
1) lr = 10^0



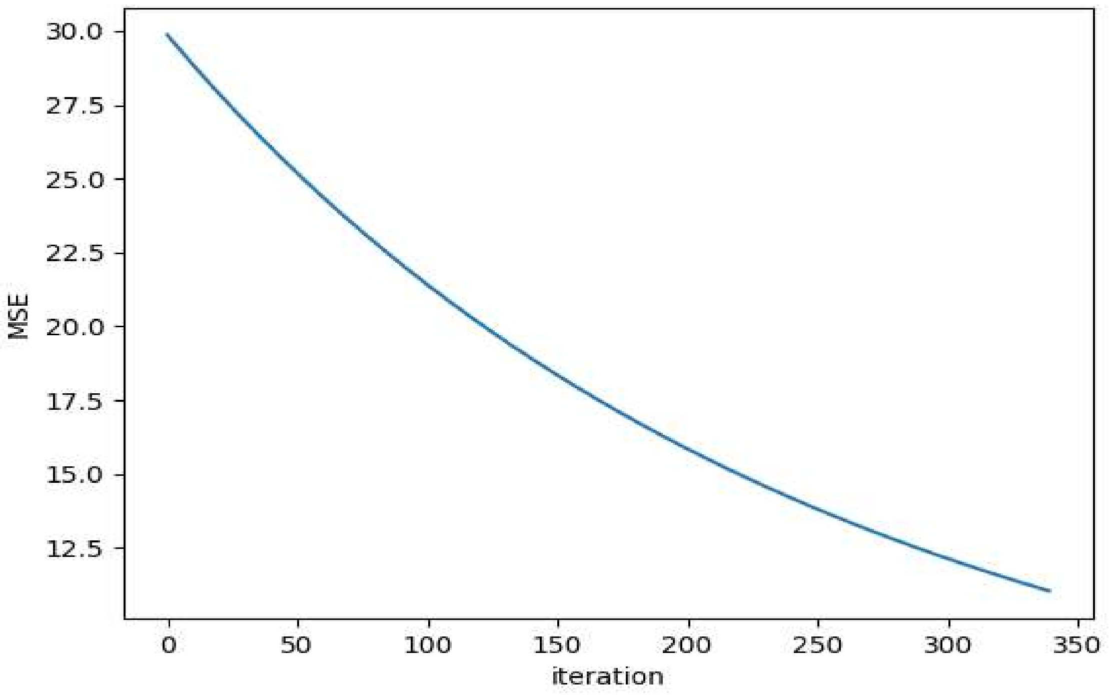
2) lr = 10^-1



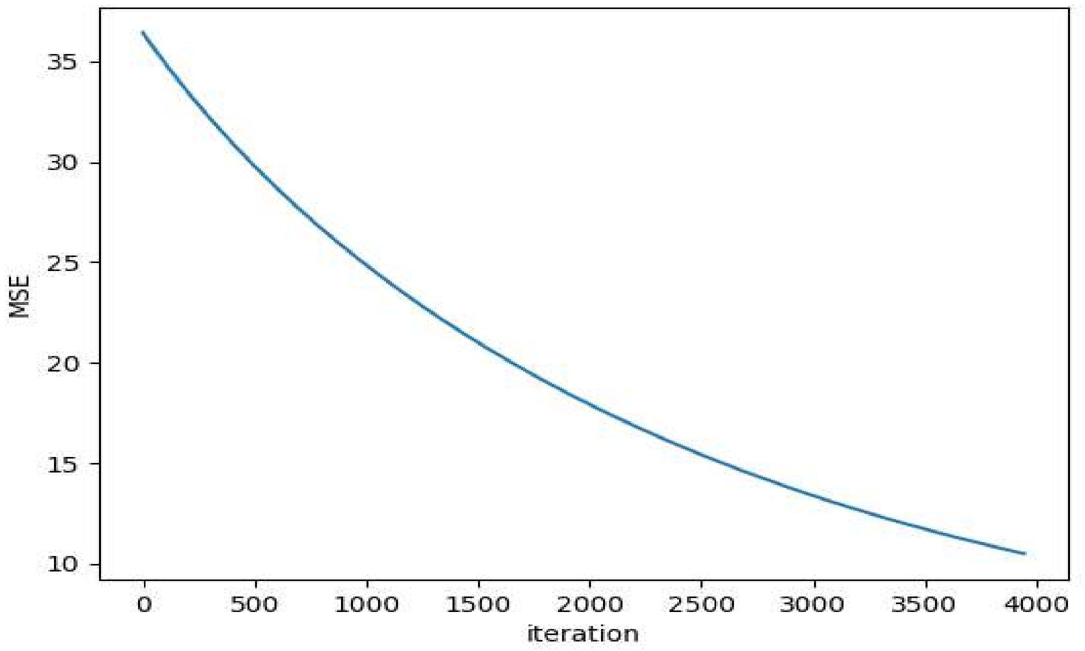
3) lr = 10^-2



4) lr = 10^-3



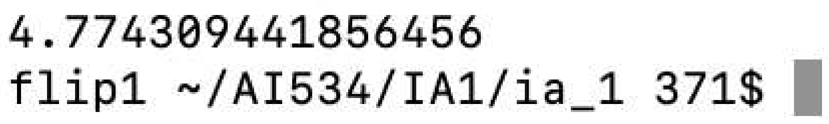
5) lr = 10^-4



Question: Which learning rate or learning rates did you observe to be good for this particular dataset? What learning rates (if any) make gradient descent diverge?
Ans: lr = 10^-1 or 10^-2 seem good enough for this training set.
lr = 10^0 diverged because the training step is too large.

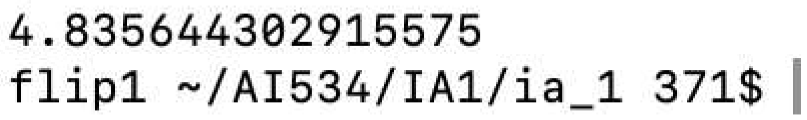
b) Question: Which learning rate leads to the best validation MSE? Between different convergent learning rates, how should we choose one if the validation MSE is nearly identical?
Ans: lr = 10^-1 leads to the best validation MSE.
For nearly identical validation MSEs, we should choose the large learning rate so that we can save a lot of computational time.

1) **lr = 10^-1: MSE for validation data:**



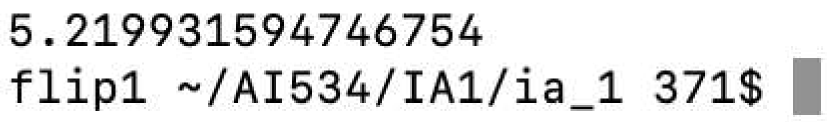
```
4.774309441856456
flip1 ~/AI534/IA1/ia_1 371$
```

2) **lr = 10^-2: MSE for validation data:**



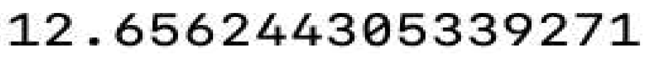
```
4.835644302915575
flip1 ~/AI534/IA1/ia_1 371$
```

3) **lr = 10^-3: MSE for validation data:**



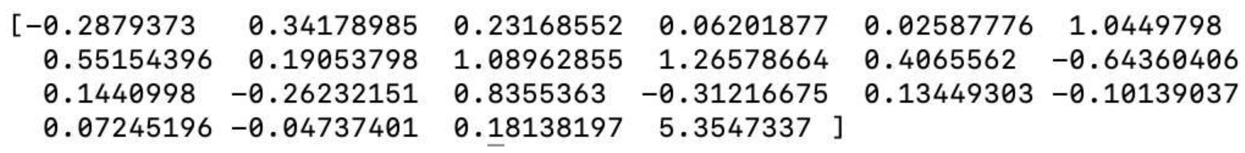
```
5.219931594746754
flip1 ~/AI534/IA1/ia_1 371$
```

4) **lr = 10^-4: MSE for validation data:**



```
12.656244305339271
```

As lr = 10^-1 leads to the smallest validation MSE, below are the weights learned from it.



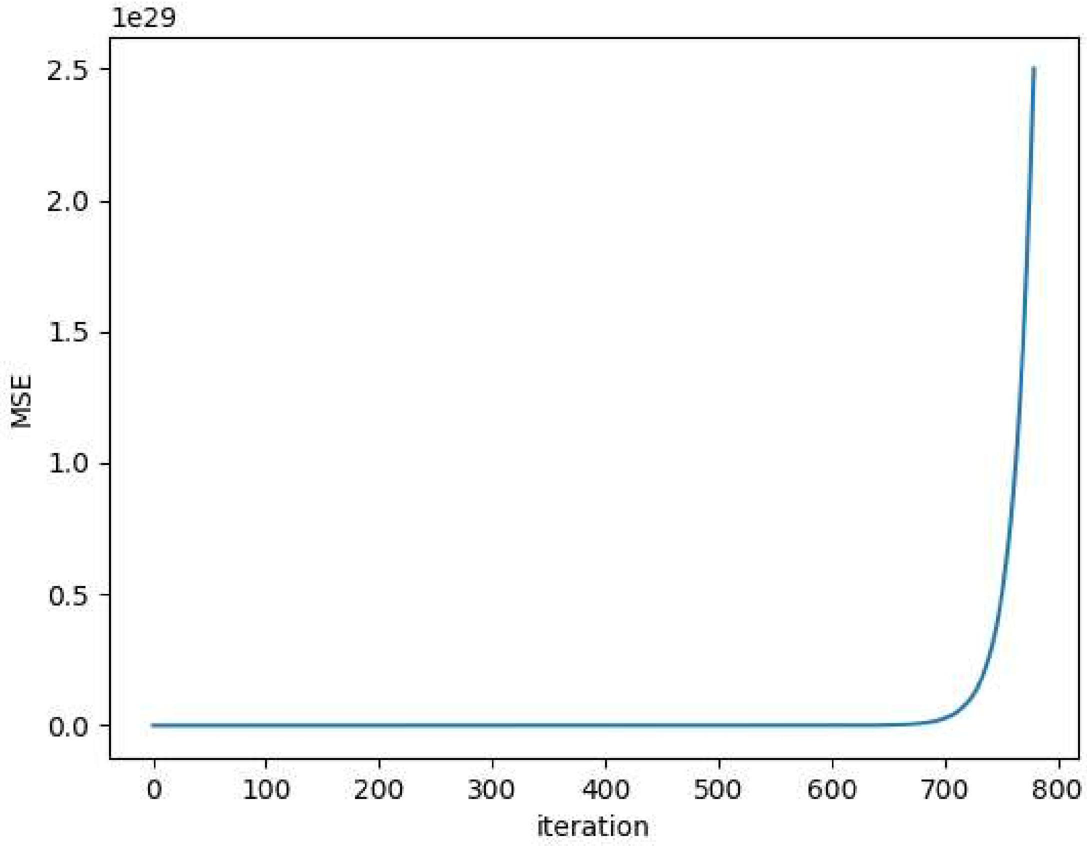
```
[-0.2879373   0.34178985  0.23168552  0.06201877  0.02587776  1.0449798
  0.55154396  0.19053798  1.08962855  1.26578664  0.4065562  -0.64360406
  0.1440998  -0.26232151  0.8355363  -0.31216675  0.13449303 -0.10139037
  0.07245196 -0.04737401  0.18138197  5.3547337 ]
```

c) Question: learned feature weights are often used to understand the importance of the features. What features are the most important in deciding the house prices according to the learned weights?
Ans: From our training result, sqft_above is the most important feature.This makes sense in reality. Grading is the second most important factor.

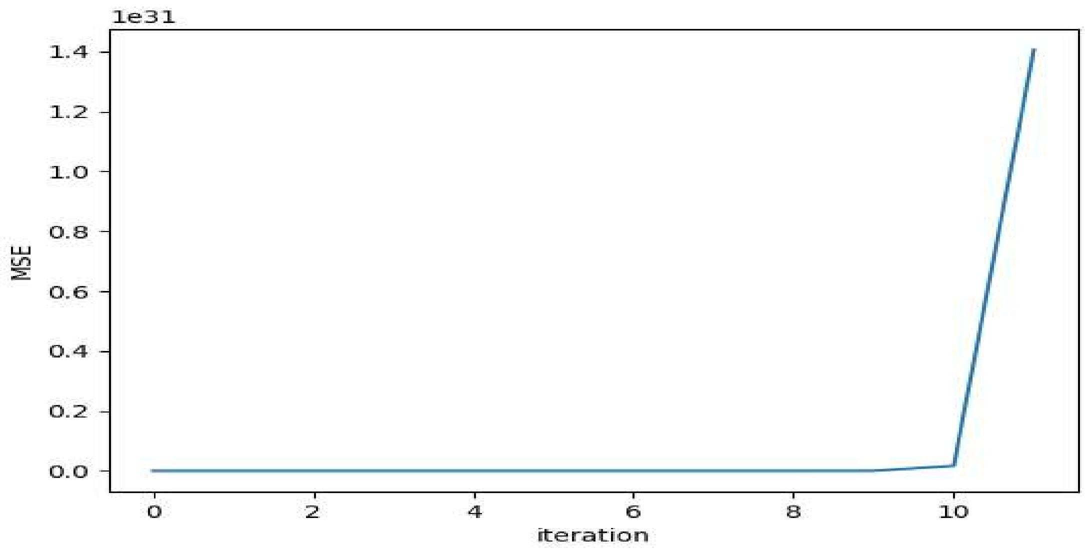
Part - 2:

a) Training with non-normalized data
    i) a list all the learning rates that you experimented with for this part and whether the learning is converging or diverging.

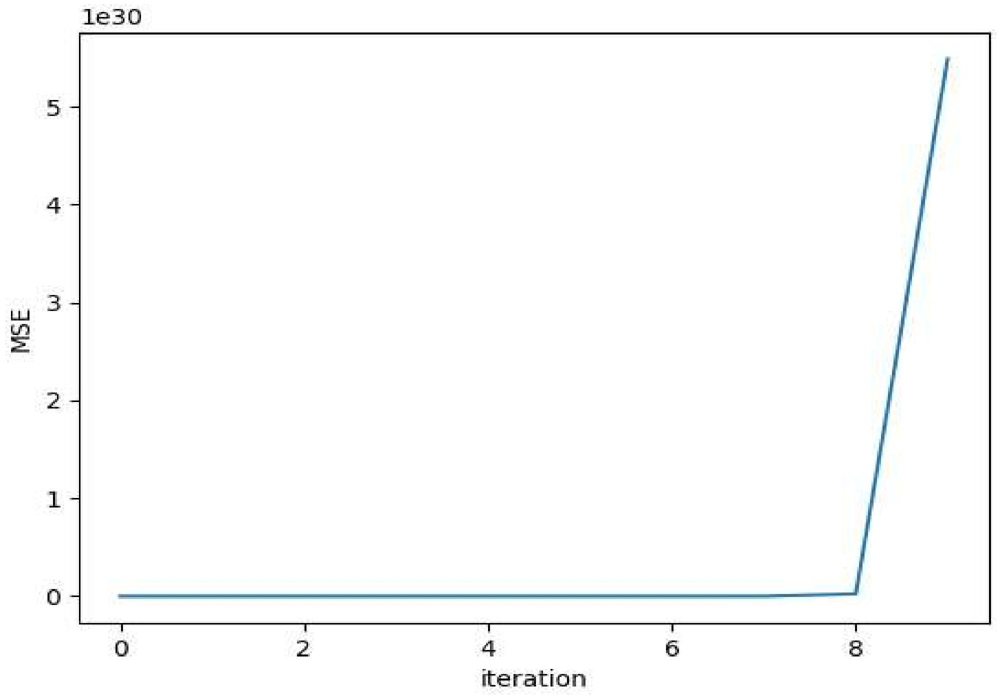
1) lr = 10^-10: diverged



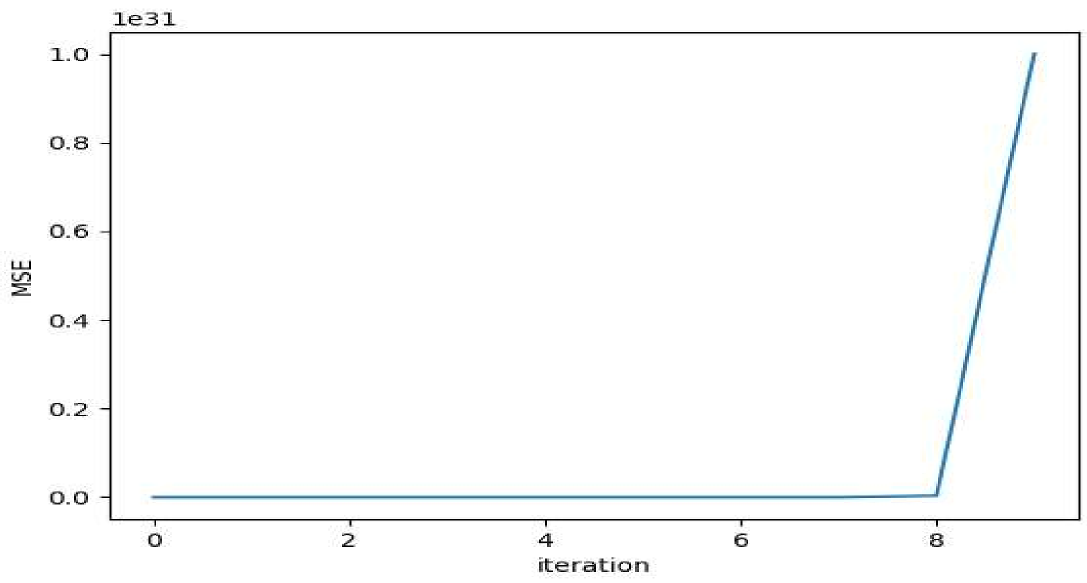
2) **lr = 5*10^-10: diverged**



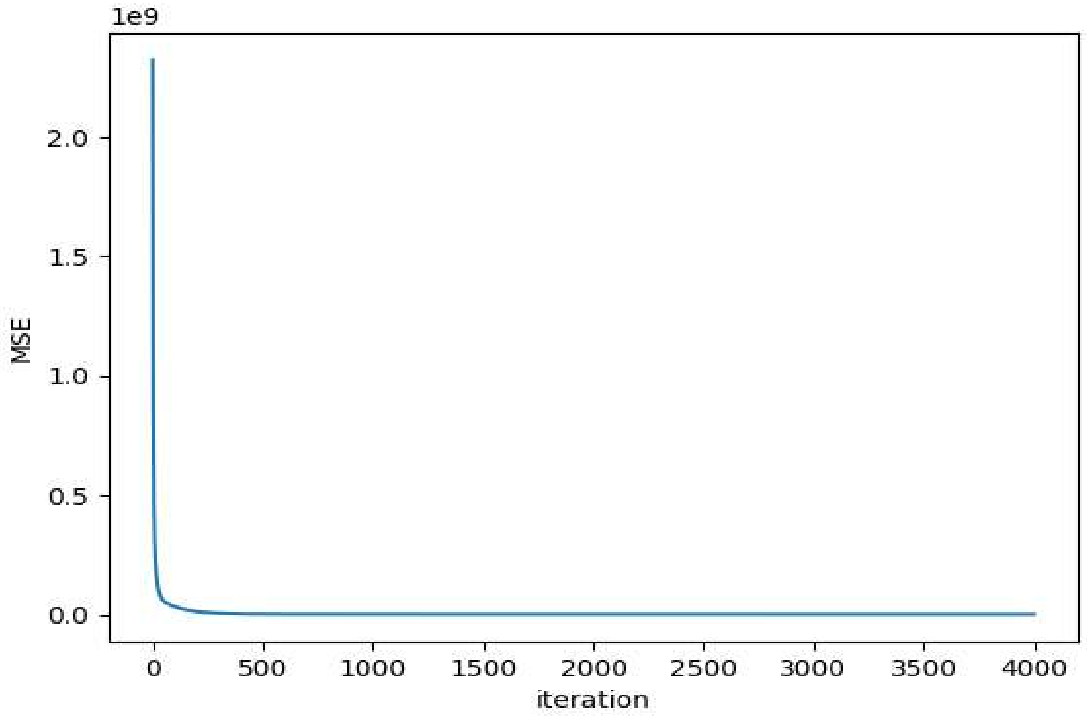
3) lr = 8*10^-10:diverged



4) lr = 9*10^-10: diverged

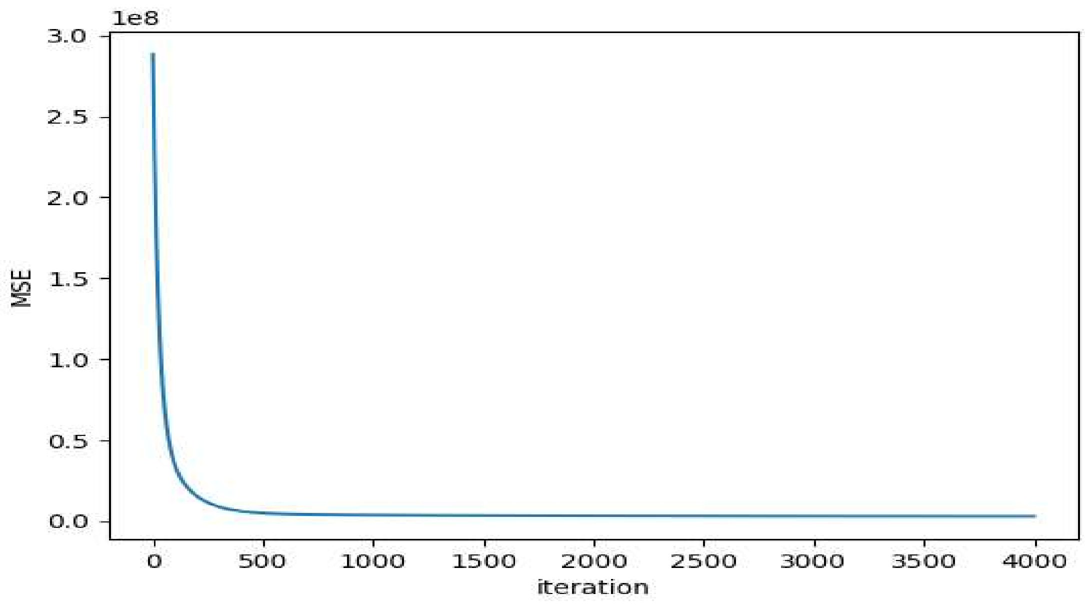


5) lr = 10^-11: converged



MSE for validation data: 1221061.1811492385

6) lr = 10^-12: converged



MSE for validation data: 3119670.059574447

ii) the best learning rate that you have found and its resulting MSE on the training and validation data respectively
Ans: lr = 10^-11, MSE for validation data: 1221061.1811492385

iii) the learned feature weights using the best learning rate (excluding w0)

[0.43847204 0.69690245 0.45346466 -0.00145701 0.19588303 0.69186168 0.7505955 0.06775366 0.96351262 0.45538154 -0.00379558 0.35977379 0.69084888 -0.05507143 0.61634378 0.495859 0.60183697 -0.00576556 0.34218132 0.20758527 0.9069893 0.77576905]

Questions:

i) Why do you think the learning rate needs to be much smaller for the non-normalized data?
Ans: Because the raw data can be very large/very small such that the program is more likely to skip the local minimum for a specific learning rate than the case with normalized data.

ii) Compare between using the normalized and the non-normalized versions of the data, which one is easier to train and why?
Ans: Normalized data is easier to train because it trains with larger learning rate and thus saving a lot of time.

iii) Compare the learned weights to those of 1(c), what key difference do you observe? How do you think this impacts the validity of using weights as the measure of feature importance?
Ans: Grade becomes the feature with the largest weight. This is because grades are important and come with some small numbers. When the program learns, it has to assign a larger linear coefficient (i.e., weight) so grade can be considered together with other features having a large number from raw data (e.g., sqft_above).

b) Redundancy in features
We used lr=1e-2. The MSE of validation data is 0.4346134817527694. The learned features are:

```
[-0.01447845  0.07229107 -0.26896385 -0.0014246   0.04824445  0.04614385
  0.04419286 -0.02212163  0.13038155  0.21629339  0.14530849  0.33635968
  0.46928024  0.05671406  0.00999747  0.07483471 -0.00810853 -0.05428892
 -0.06136607 -0.76585511  0.33671231]
```

Questions:
How does this new model compare to the one in part 1 (c)?
Ans: The validation MSE is smaller than part 1 (c) , which means a better model.

What do you observe when comparing the weight for sqrt living in both versions?
Ans: The weight decreased

Consider the situation in general when two features x1 and x2 are redundant, what do you expect to happen to the weights (w1and w2) when learning with
Ans: both features, in comparison with w1 which is learned with just x1? How does this phenomenon influence the interpretation of feature importance?
When the redundancy exists, we expect a smaller x1 than the case without the redundancy, because x2 shares weight with x1 for representing a similar feature.