16. Data Science – Machine Learning – Gradient Descent

Contents

1. Gradient Descent 2
2. How Gradient Descent works 2
3. Convergence 3
4. Process behind the gradient 3
5. Steps 5
6. Follow below images 7
7. Fixed steps 11
8. Learning rate: Reaching minimum error 12

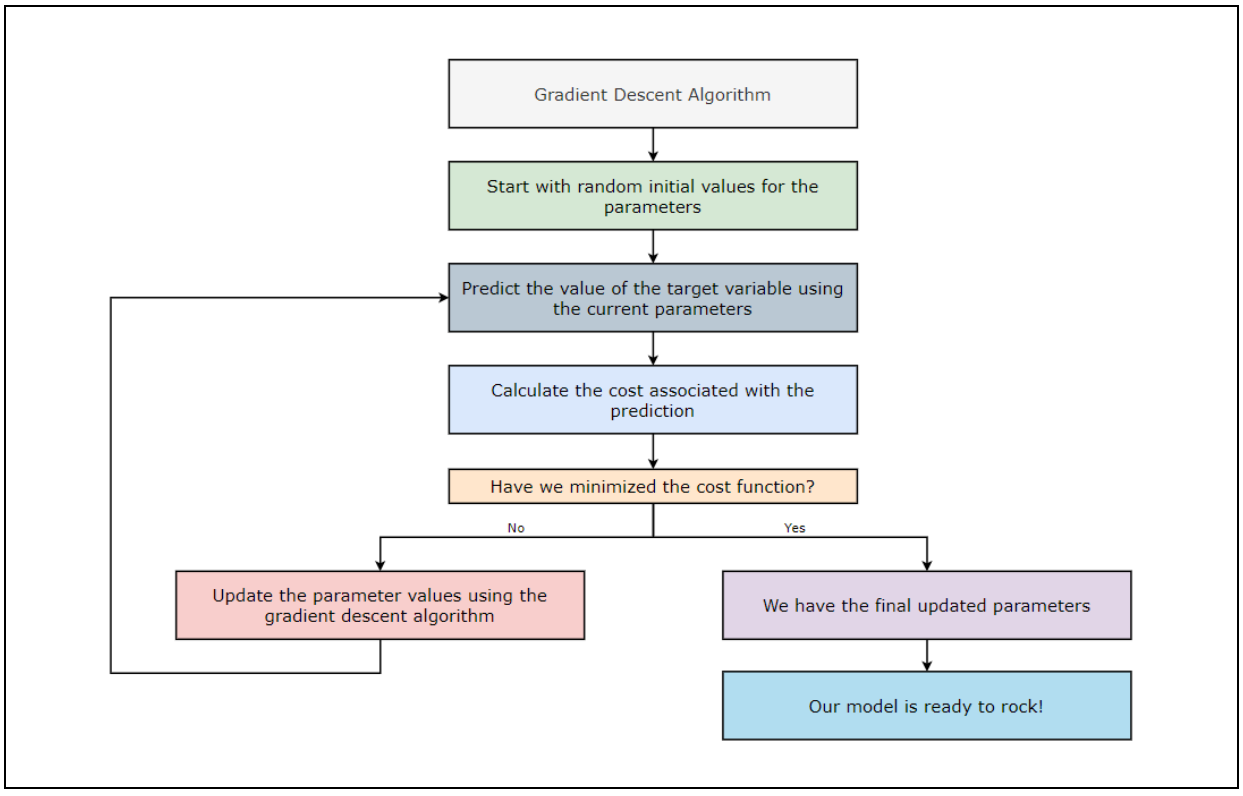
16. Data Science – Machine Learning – Gradient Descent

1. Gradient Descent

- ✓ Gradient descent is an optimization algorithm
- ✓ This algorithm find the values of parameters (coefficients) of a function (f) that minimizes a cost function (cost).

2. How Gradient Descent works

- ✓ 1. Start with random initial values for the parameters.
- ✓ 2. Predict the value of the target variable using the current parameters.
- ✓ 3. Calculate the cost associated with the prediction.
- ✓ 4. Is cost minimized?
  - If yes, then go to step no - **6**.
  - If no, then go to step no - **5**.
- ✓ **5**. Update the parameter values using the gradient descent algorithm and return to step no - **2**
- ✓ **6**. We have our final updated parameters.
- ✓ 7. Our model is ready.

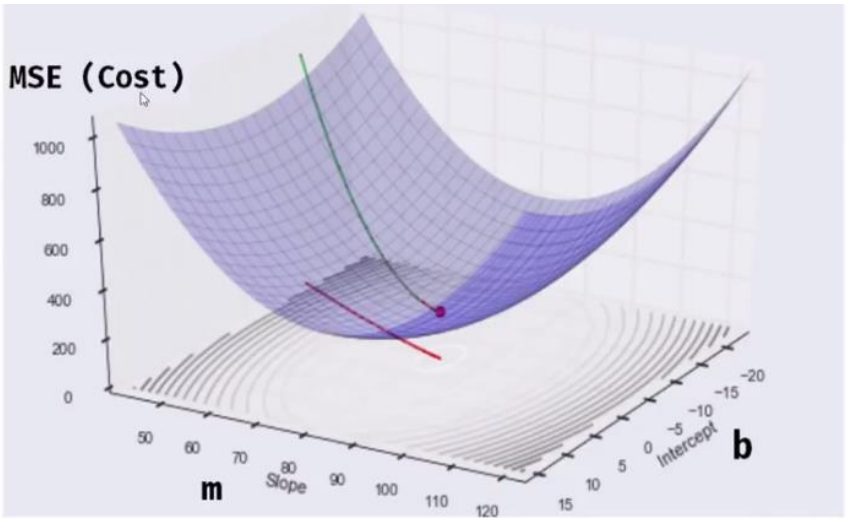


3. Convergence

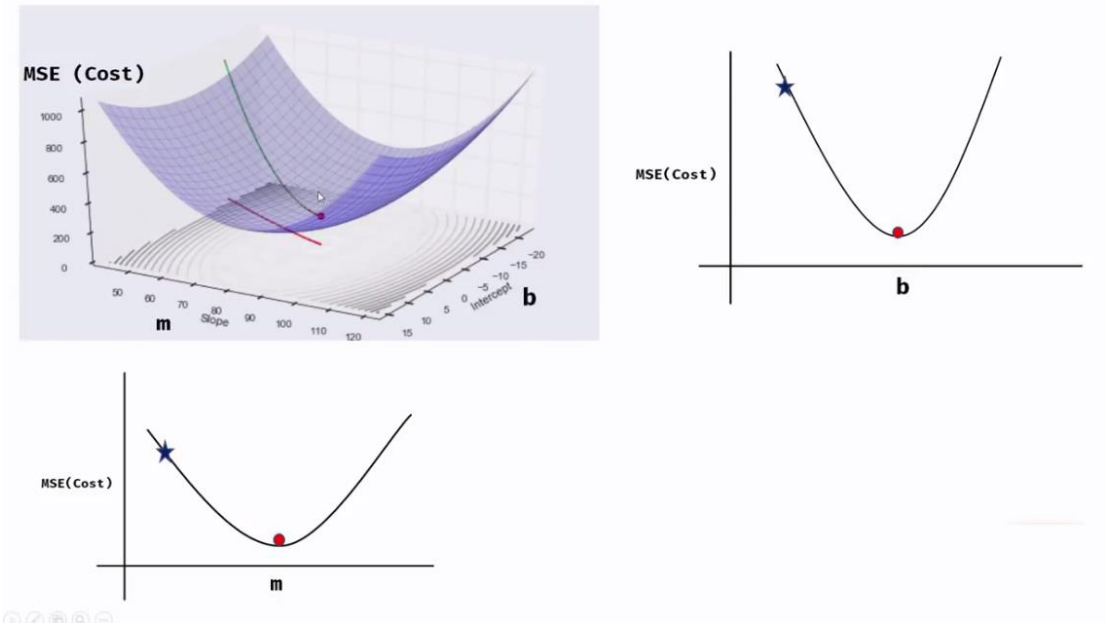
- ✓ Reaching a point in which gradient descent makes very small changes in your objective function is called convergence.
- ✓ It doesn't mean that, it has reached the optimal result but it is somewhat near to that.

4. Process behind the gradient

- ✓ Gradient descent is the algorithm that finds best fit line for given training data set.
- ✓ If we calculate for every value of **m** and **b** we will get MSE
- ✓ If we plot m and b against MSE then we will get below image

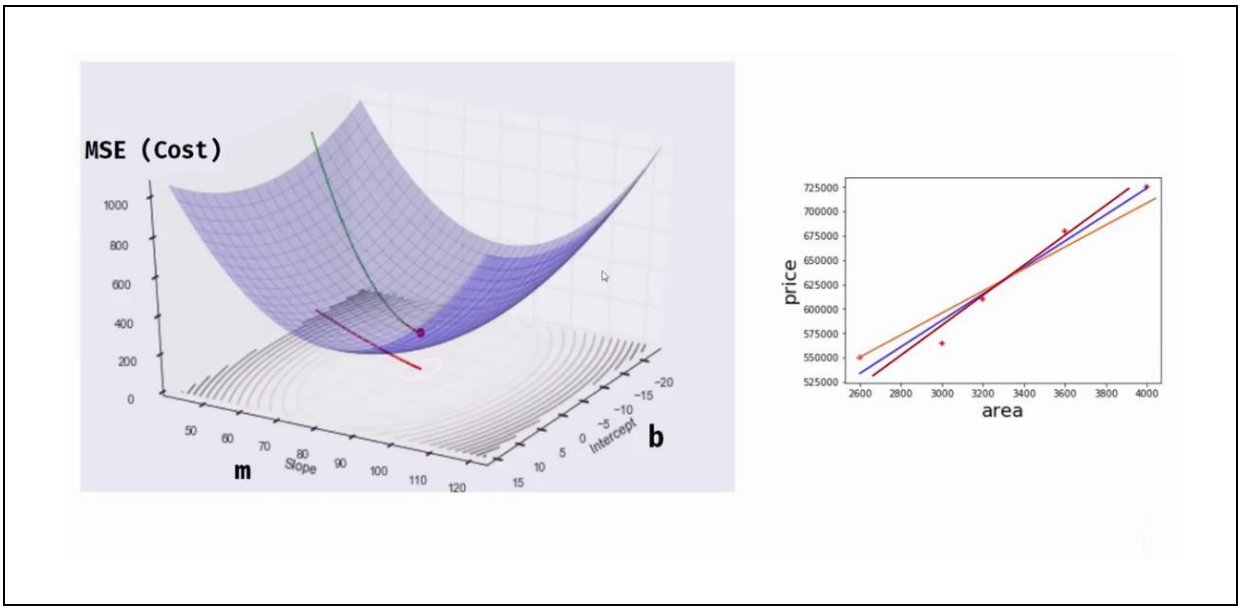


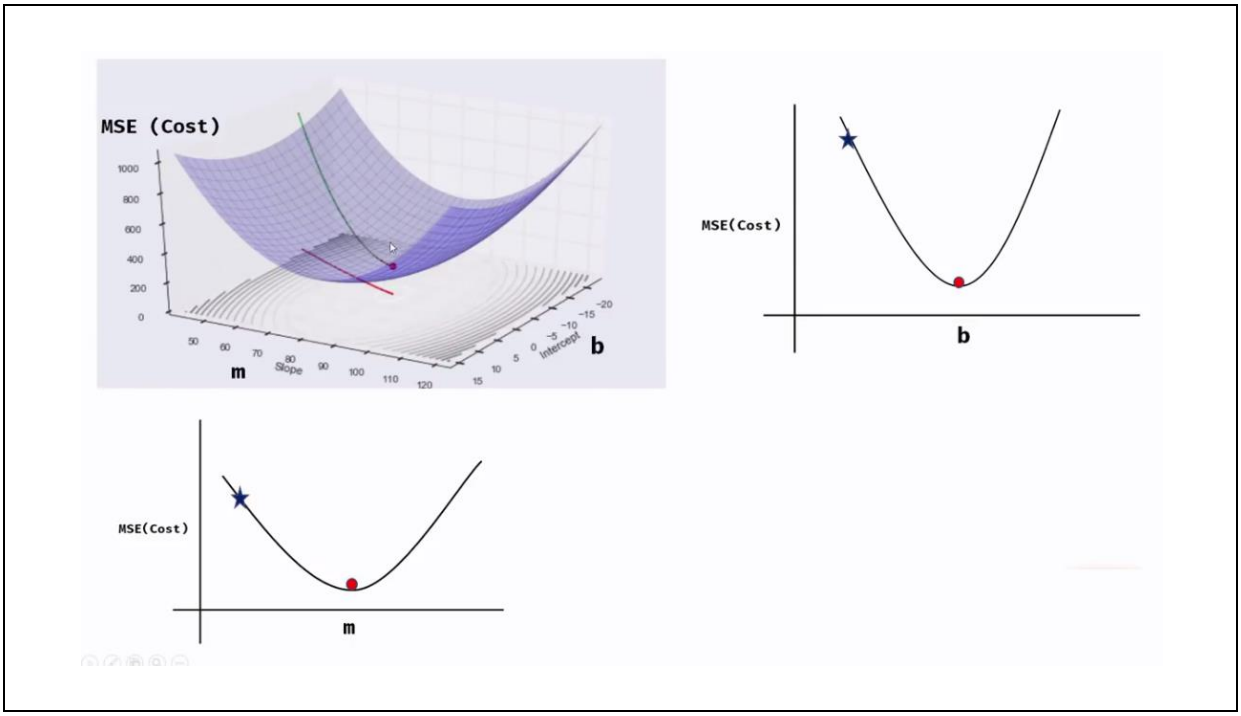
- ✓ Values
  - MSE values 0, 200, 400, 600, 800, 1000 etc
  - m values 50, 60, 70, 80, 90, 100, 110, 120 etc
  - b values 15, 10, 5, 0, -5, -10, -15 etc
- ✓ Here we will get an image which is like a bowl.
- ✓ We need to start some value for m and b, usually let's start from zero value


MSE (Cost)
1000
800
600
400
200
0
50
60
70
80
90
100
110
120
m
Slope
15
10
5
0
-5
-10
-15
-20
Intercept
b
MSE(Cost)
b
MSE(Cost)
m

5. Steps

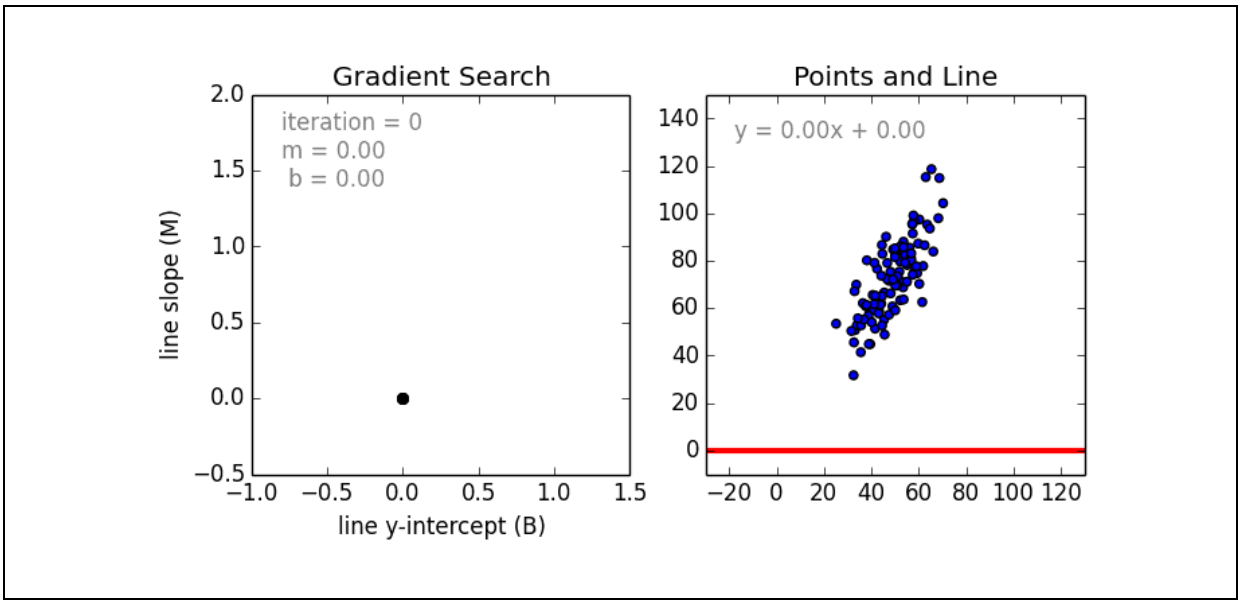
- ✓ Initially lets starts the point with m = 0 and b = 0
- ✓ Assuming that according to the diagram mse value is 1000 at m, b equals to zero
- ✓ Now just tune the values of m and b with some amount
- ✓ Now check mse value and the error will get reduce
- ✓ So, we need to keep on doing this process until to reach error level to minima(minimum)
- ✓ Once it reach the minima then we got our answer
- ✓ Here we need to use that m and b values, in the prediction function

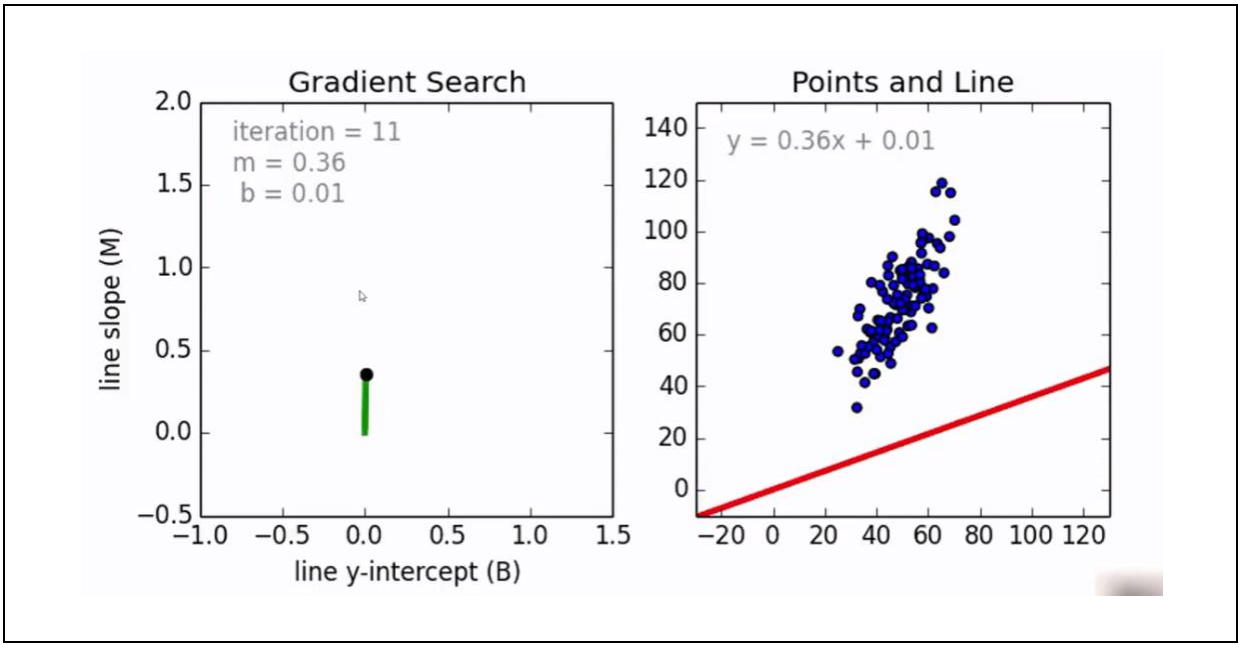


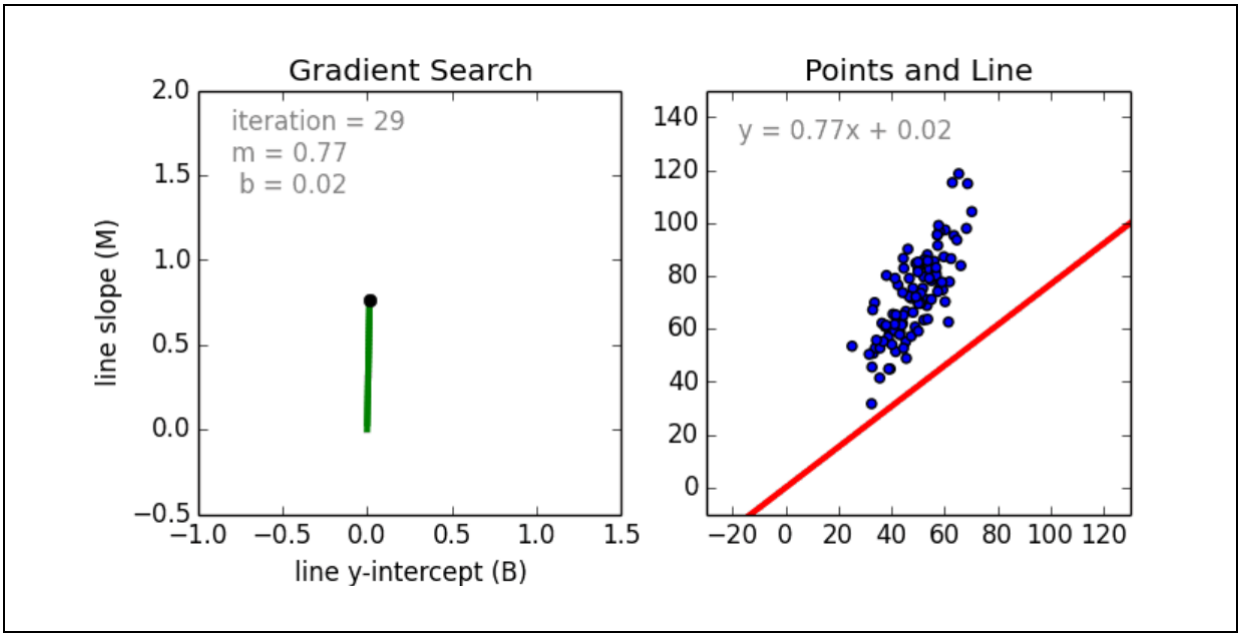

MSE (Cost)
1000
800
600
400
200
0
50
60
70
80
90
100
110
120
m
Slope
15
10
5
0
-5
-10
-15
-20
Intercept
b
MSE(Cost)
b
MSE(Cost)
m

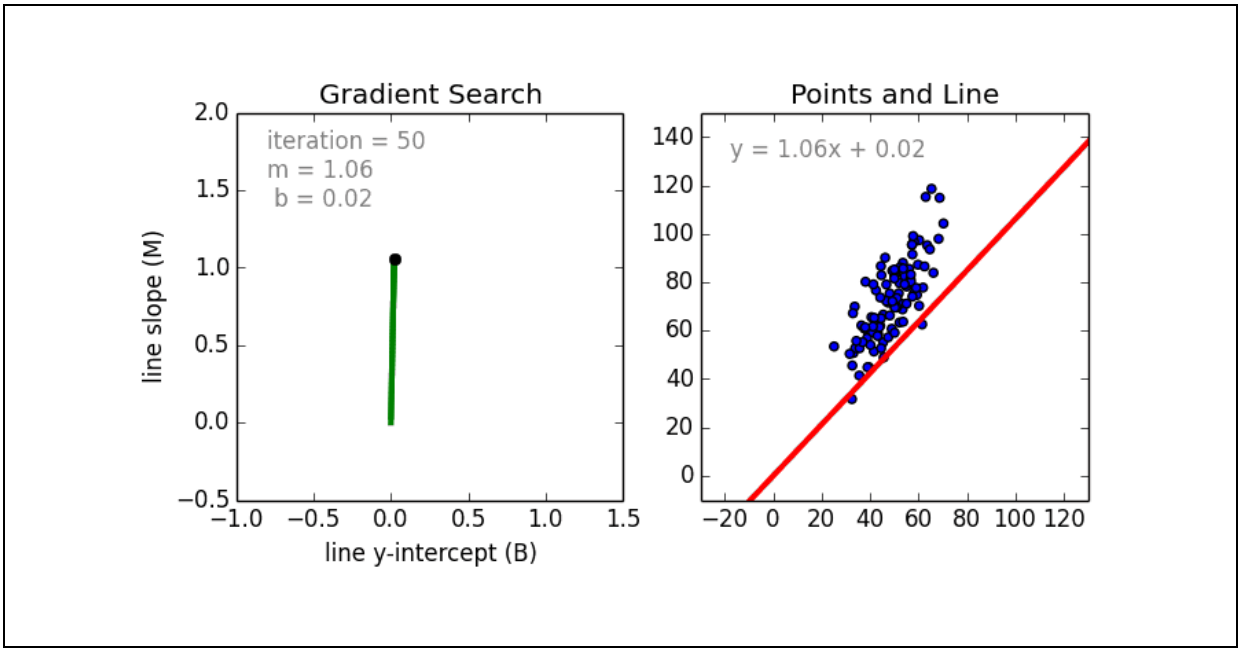
6. Follow below images

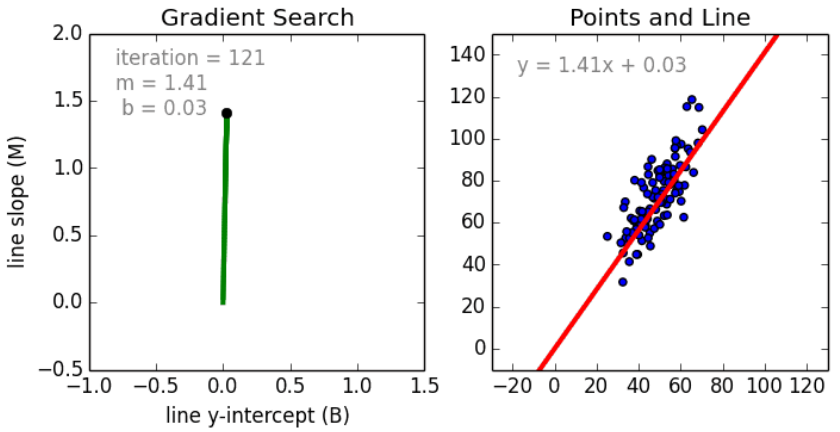
✓ Just try to understand below images for better understanding

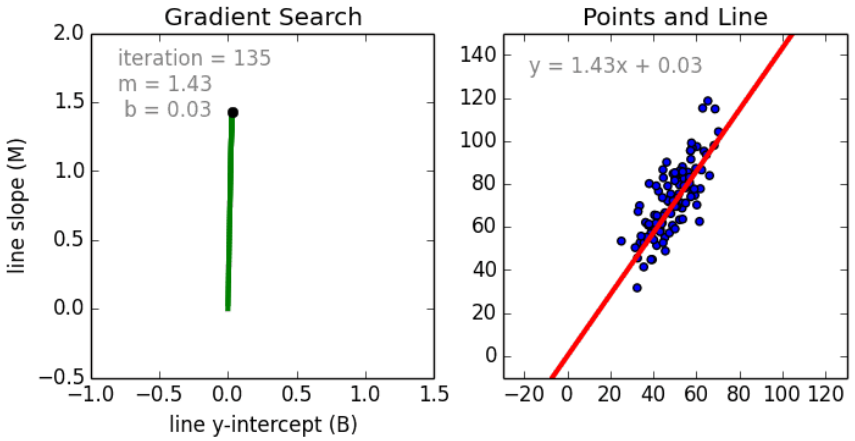


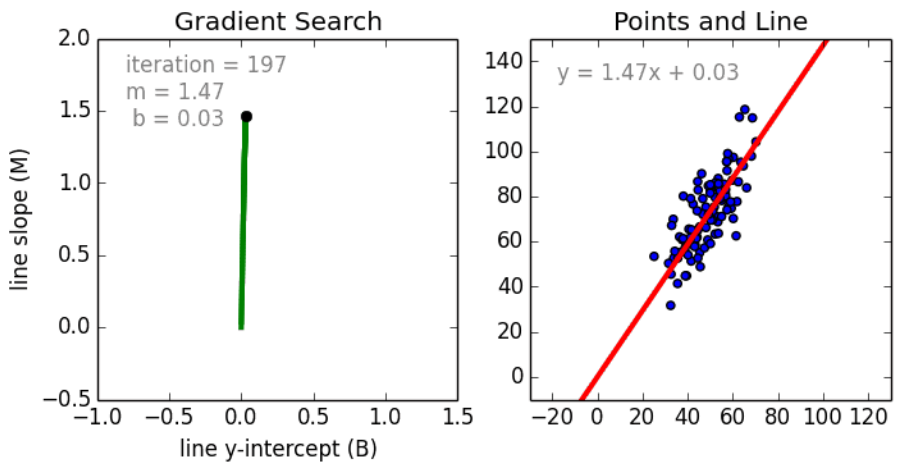


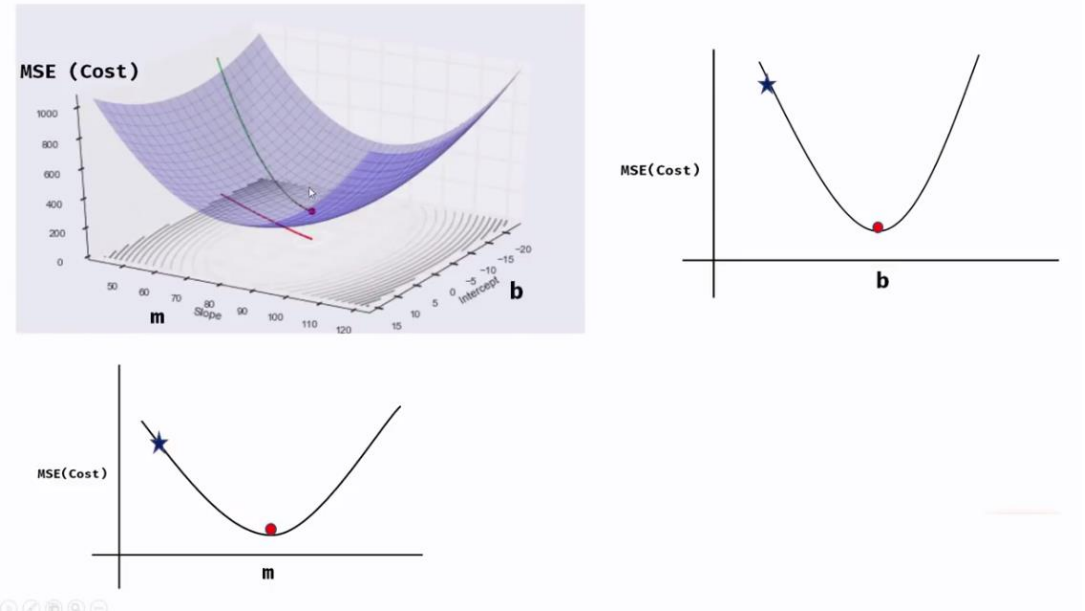

Gradient Search
iteration = 29
m = 0.77
b = 0.02
line slope (M)
line y-intercept (B)
Points and Line
y = 0.77x + 0.02


Gradient Search
iteration = 50
m = 1.06
b = 0.02
line slope (M)
line y-intercept (B)
Points and Line
y = 1.06x + 0.02


Gradient Search
iteration = 121
m = 1.41
b = 0.03
line slope (M)
line y-intercept (B)
Points and Line
y = 1.41x + 0.03

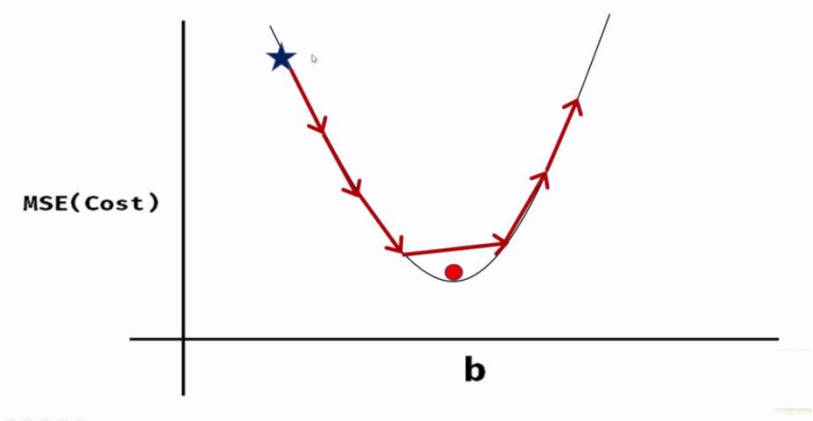

Gradient Search
iteration = 135
m = 1.43
b = 0.03
line slope (M)
line y-intercept (B)
Points and Line
y = 1.43x + 0.03


Gradient Search
iteration = 197
m = 1.47
b = 0.03
line slope (M)
line y-intercept (B)
Points and Line
y = 1.47x + 0.03


MSE (Cost)
m
b
MSE(Cost)
b
MSE(Cost)
m

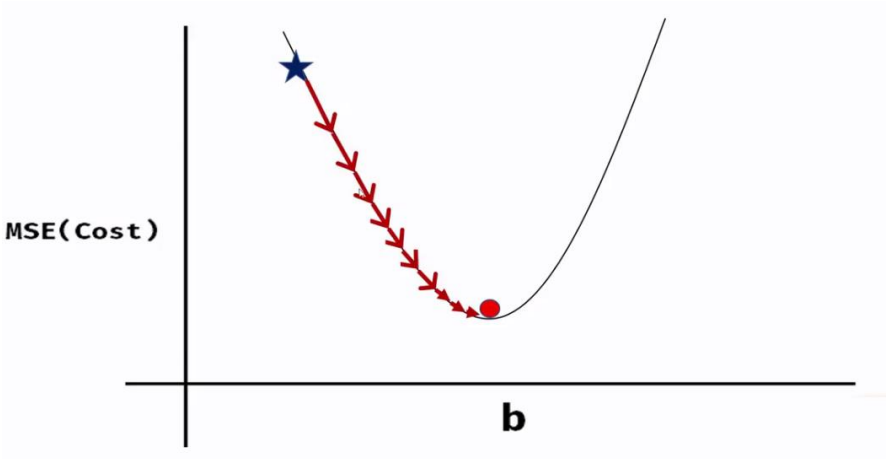
7. Fixed steps

- ✓ If we follow the fixed steps then we will get the below image
- ✓ By following this approach we may miss the global minima value.
- ✓ So, this approach is not going to work in well

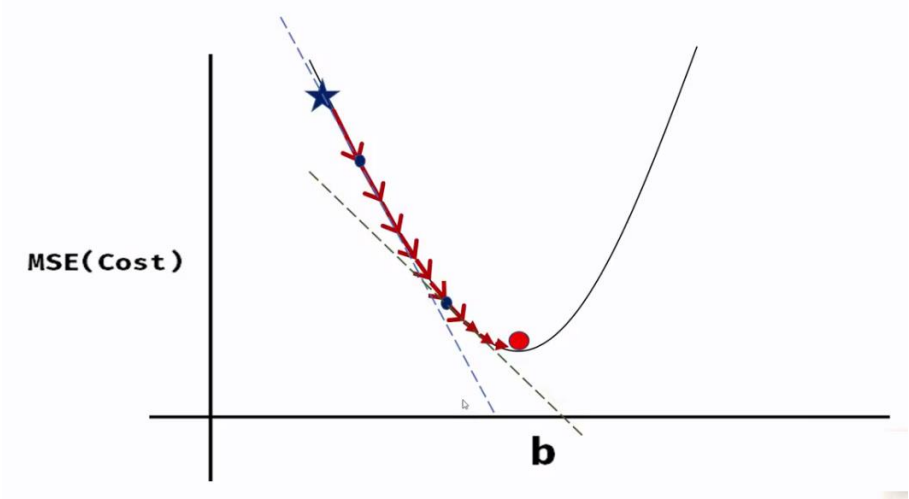


8. Learning rate: Reaching minimum error

- ✓ This approach seems to be good to catch minima value.
- ✓ The learning rate is a tuning parameter in an optimization algorithm.
- ✓ This determines the step size at each iteration while moving toward a minimum of a loss function.



- ✓ At each point if we calculate the slop then we can reach the minimum value



9. Small and large learning rates

- ✓ A small leaning rate may lead to the model to take some time to learn
- ✓ A large learning rate will make the model converge as our pointer will shoot and we'll not be able to get to minima.