INTRODUCTION TO LINEAR REGRESSION

WWW.TANTROSTAV.CO.IN

Linear Regression

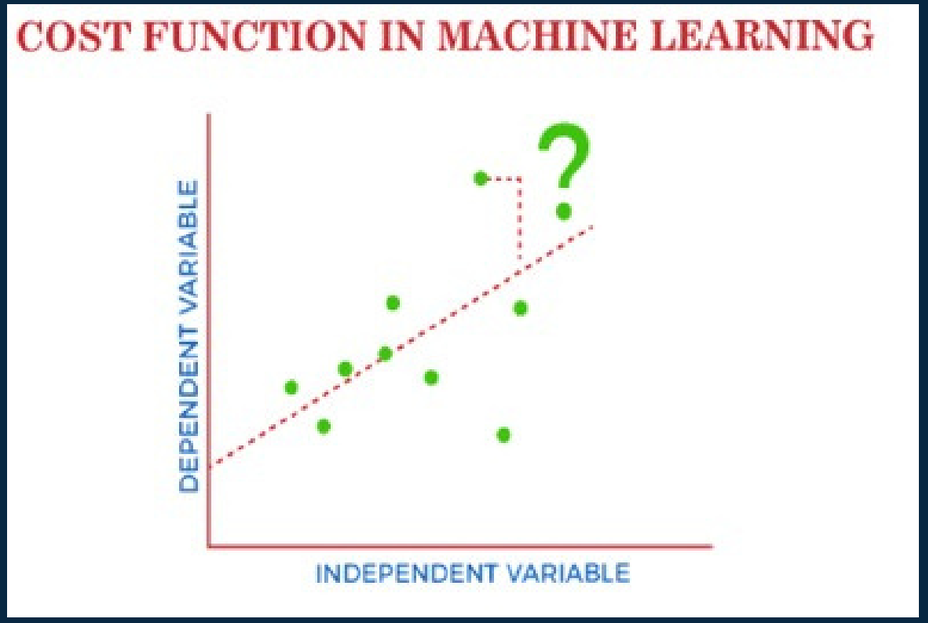
- Linear Regression is a supervised machine learning algorithm and performs a regression task
- In Regression there are infinitely many possible numbers that model could output.
- Linear regression is used in many different fields.
- **For example**, in finance, linear regression might be used to understand the relationship between acompany's stock price and its earnings, or to predict the future value of a currency based on its past performance.
- Linear Regression includes both input features and output targets. To train the model, we feed both input features and output targets to our learning Algorithm.
- Here we fit a straight line to our data.

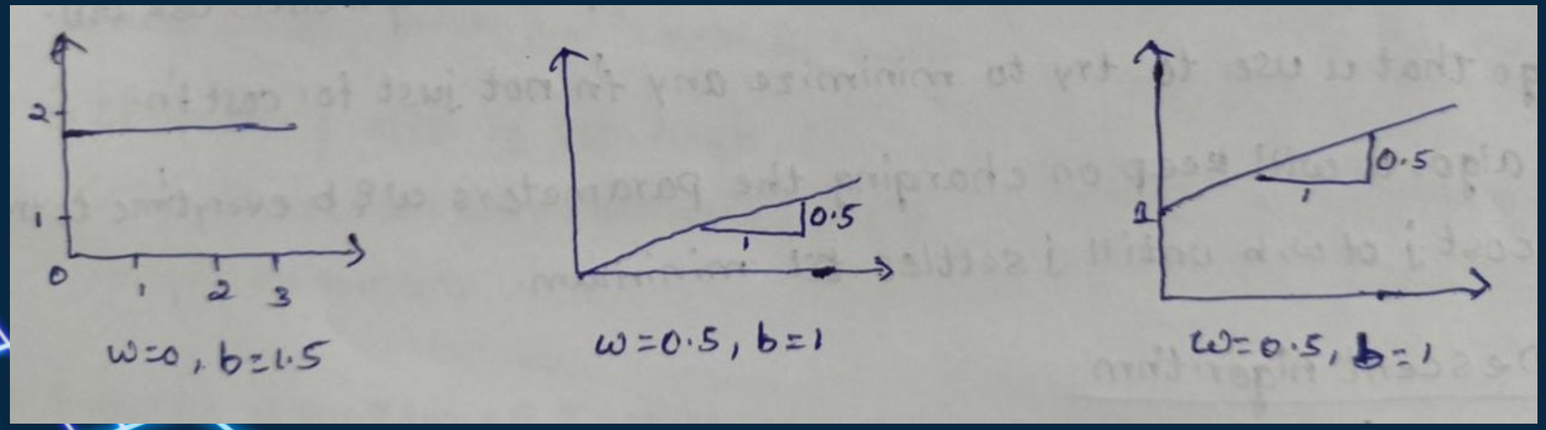
Cost Function

- In order to implement Linear Regression we need to define Cost Function which tells us how well the model is doing.

- For Example, the model we use to fit in the training set is f(x)=w*x+b where w,b are parameters.

- Depending on the values of w,b we get different f(x) which generates a different line.

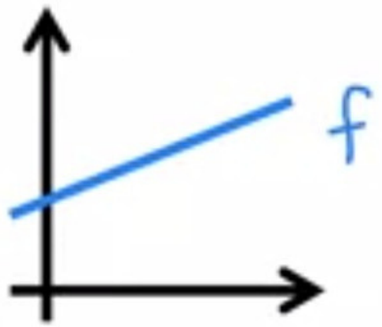




How do you find the values of w and b so that we get different f(x) which generates a line on the graph?

model:

$$f_{w,b}(x) = wx + b$$

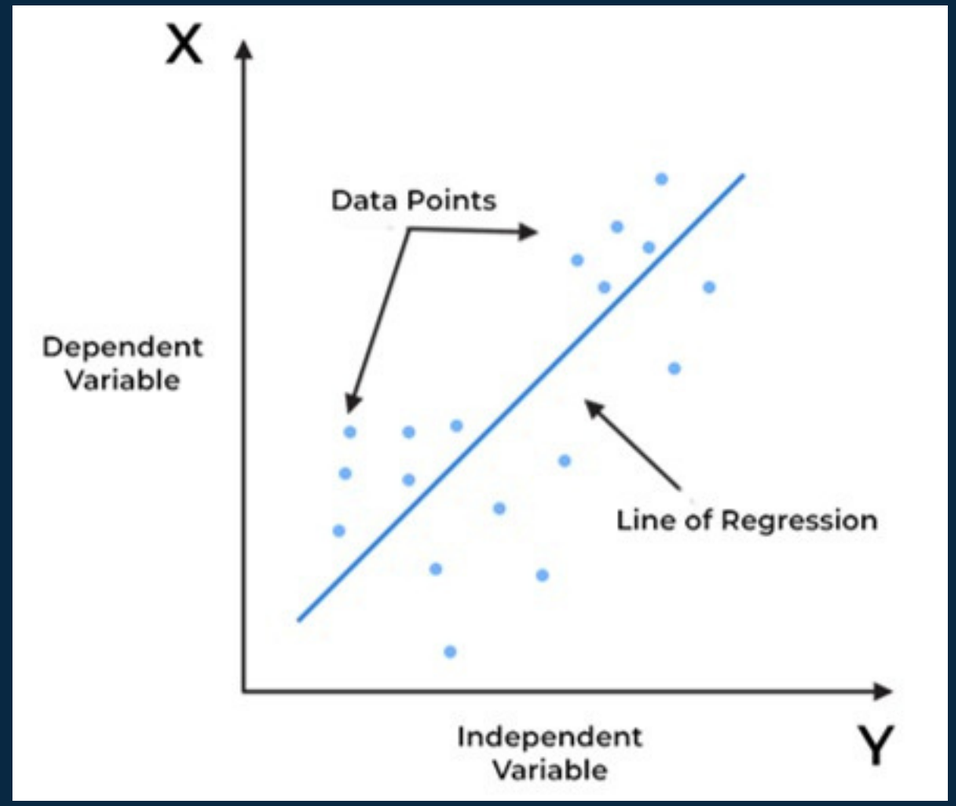


parameters:

w, b

cost function:

$$J(w,b) = \frac{1}{2m}\sum_{i=1}^{m}(f_{w,b}(x^{(i)}) - y^{(i)})^2$$



GOAL

Minimize cost Function (i.e.) J(w,b)

Gradient Descent

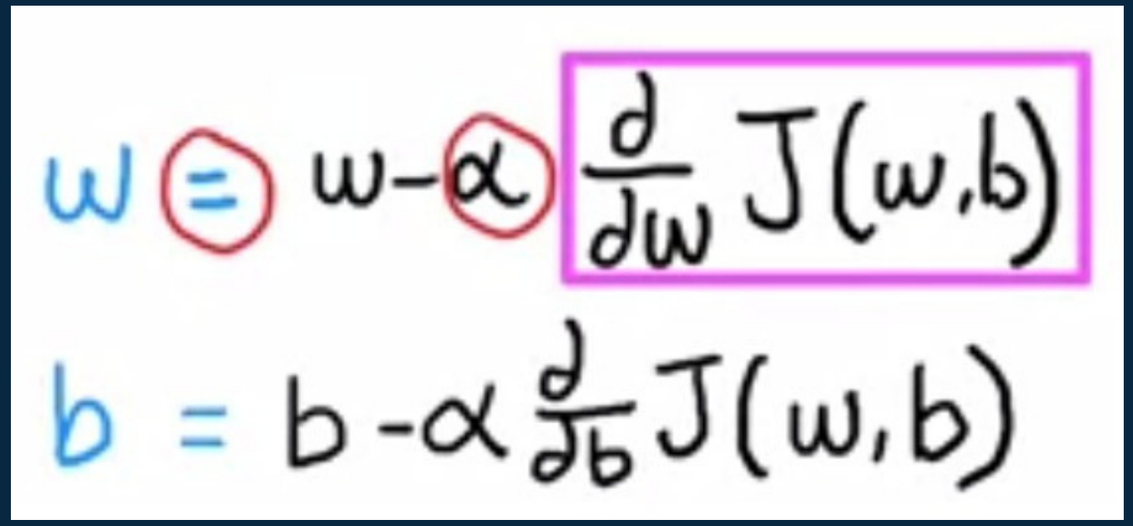
- Gradient Descent is an algorithm that we use to try to minimize any function not just for cost function.

- So by using Gradient Descent algorithm we keep on changing the parameters w,b everytime to try to reduce the cost J(w,b) until J settles at minimum.

Gradient Descent

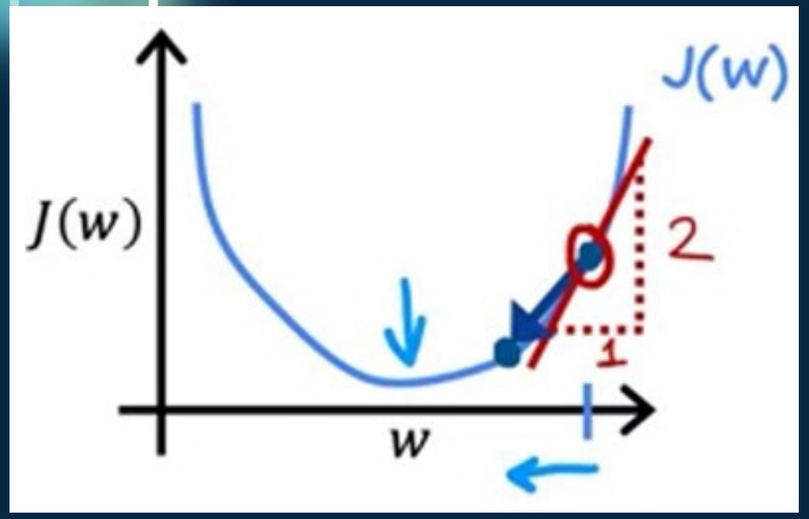
- Here alpha is learning rate which is small +ve number between 0 and 1

- If alpha is large a very aggressive gradient descent (i.e.) you are trying to take huge steps downhill.
- If alpha is small you are taking small steps downhill.



- For Gradient Descent Algorithm we are going to repeat the above 2 steps repeatedly until the algorithm converges.

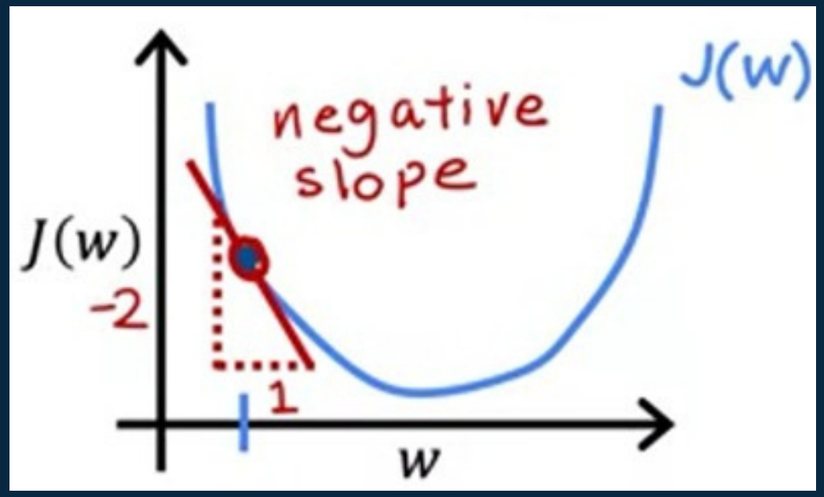
- Here we need to simultaneously update w,b.



•Here let’s initialize Gradient Descent with some starting value of w.
•Here Gradient Descent will update w to

$$w - \alpha \frac{\partial}{\partial w}(J(w,b))$$

•Here alpha and slope both are +ve.
•So we move to left side by decreasing value of w.



- Here let’s initialize Gradient Descent with some starting value of w.
- Here Gradient Descent will update w to

$$w - \alpha \frac{\partial}{\partial w}(J(w,b))$$

- Here alpha is +ve and slope is –ve.
- This step of Gradient Descent causes w to increase which means we are moving towards right of graph.

Evaluation metrics

- The sum of squares due to error (SSE)

$$Err(i) = Y_i - f(x(i))$$

SSE(sum of squared errrors) = $\sum_{i=1}^{n} Err(i)^2$

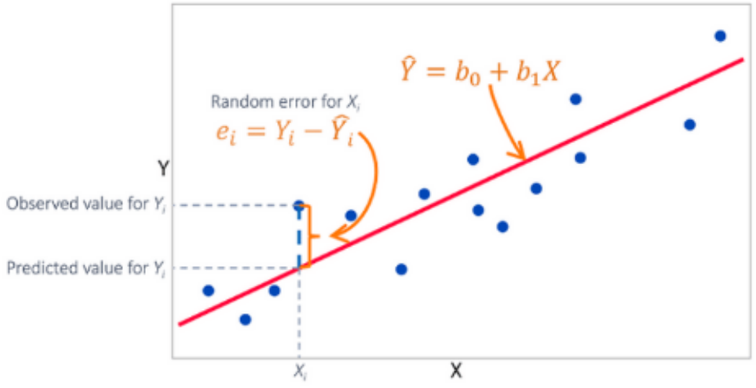
- R-square

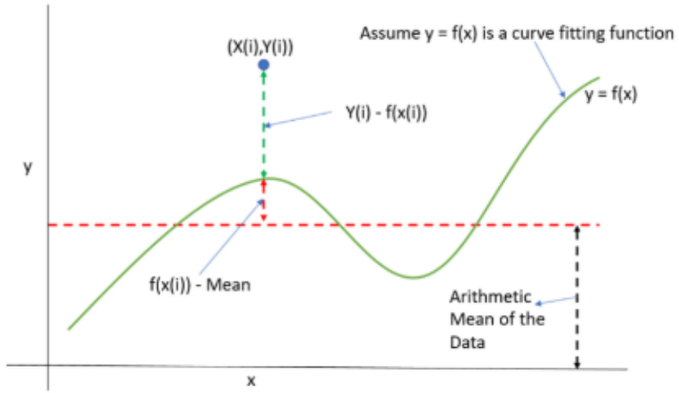
$$R^2 = \frac{SSR}{SST}$$

$$\text{SSR} = \sum_{i=1}^{n} (f(x(i)) - Mean)^2$$

$$\text{SSE} = \sum_{i=1}^{n} (Y_i - f(x(i)))^2$$

SST = SSR + SSE





Evaluation metrics

- Adjusted R-square

$$R^2_{adj} = 1 - \frac{(1 - R^2)(n - 1)}{n - k - 1}$$

- Root mean squared error (RMSE)

$$RMSE = \sqrt{\left(\frac{SSE}{n}\right)}$$