HOST : Deepesh Bhatore
Github :
github.com/deepeshdm

# Introduction to Linear Regression

- what is regression ?
- what is a scatterplot ?
- what is correlation ?
- linear regression
- cost function & gradient descent
- linear regression in python

NOTE : Implementing these Algorithms is dead easy, but understanding them is a bit challenging.

## Regression Problems - Linear Regression

A regression problem is when you have one or multiple independent variables and you try to predict one or more dependent variable. The dependent value is a continuous or real value.

Examples of Regression problems :
1) Predicting a person's age
2) Predicting stock prices
3) Predicting price of something

Target Variable or Dependent Variable

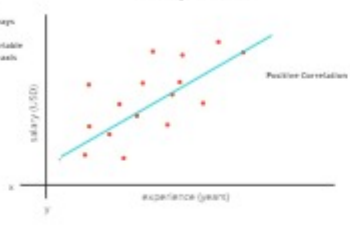
Independent Variables

| | Age | Experience(yr) | Job | Salary |
|---|---|---|---|---|
| 1] | 24 | 3.5 | Software engineer | 51500 |
| 2] | 28 | 5.6 | Data scientist | 86700 |
| 3] | 35 | 10.1 | ML engineer | 125600 |

NOTE : In regression,Target Variable is a Continuous value

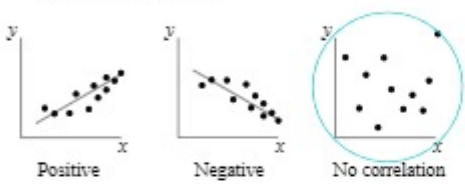
## Scatterplot in 2D

Independent Variable is always on X-axis and Dependent Variable is always on Y-axis




## Correlation

We plot scatterplots to find the correlation between variables. Correlation is the measure of how the values of variables change in relation to each other




NOTE : For regression problems , it is good if the independent & Dependent Variable is "correlated".

NOTE : It is compulsory that the independent and Dependent Variables are Correlated when using Linear Regression.

## Linear Regression

It is the simplest Machine Learning Algorithm for solving Regression problems

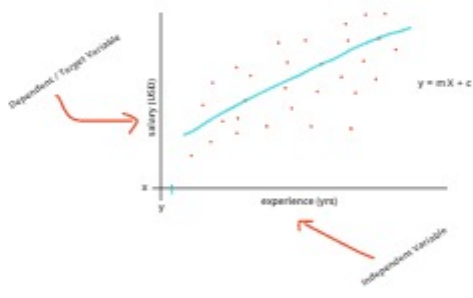
- Simple Linear Regression (1 Independent & 1 Dependent Variable)
- Multiple Linear Regression (Multiple Independent & Dependent Variables)

## Simple Linear Regression

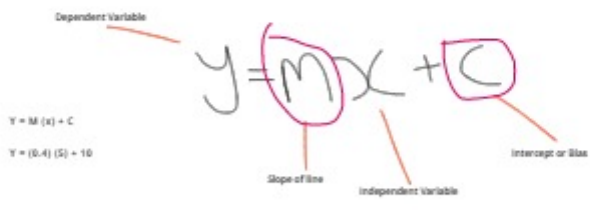
(Data of some employees)

| | Age | Experience(yr) | Job | Salary |
|---|---|---|---|---|
| 1] | 24 | 3.5 | Software engineer | 51500 |
| 2] | 28 | 5.6 | Data scientist | 86700 |
| 3] | 35 | 10.1 | ML engineer | 125600 |

predicted value
41500




### Equation of "Line of Best Fit"




NOTE : Our learning parameters are "M" and "C"

## COST FUNCTION or LOSS FUNCTION

It's a method of evaluating how well your algorithm models your dataset. If your model is predicting all values correctly then the Loss will be a lower number, whereas if your model is predicting values incorrectly , the Loss will be higher . As you change pieces of your algorithm to try and improve your model, your loss function will tell you if you're getting anywhere.

NOTE : We have different Cost/Loss functions depending on your problem.

- Binary Cross-Entropy Loss
- Hinge Loss
- Mean Square Error
- Huber Loss

NOTE : Cost functions output a single number representing how well our model is predicting values. Lower cost function is always prefferable.
THE GOAL OF TRAINING ANY MODEL IS TO LOWER THE LOSS FUNCTION VALUE.

### For Regression Problems the most common LOSS function is "MSE"

Mean Squared Error

$$mse = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_{predicted})^2$$

### EXAMPLE of MSE

| | Y1 True Values | Y Predicted Values |
|---|---|---|
| 1] | 3.5 | 1.5 |
| 2] | 4.2 | 3.5 |
| 3] | 1.8 | 2.5 |
| 4] | 0.5 | 0.9 |

$$\text{MSE LOSS} = \frac{(3.5-1.5)^2 + (4.2-3.5)^2 + (1.8-2.5)^2 + (0.5-0.9)^2}{4}$$

$$\text{MSE LOSS} = \frac{4 + 0.49 + 0.49 + 0.16}{4}$$

MSE LOSS = 1.285

NOTE : We Optimize our Machine Learning Algorithm to Lower the Cost

Now we know what loss function is , i.e we know which "Line of best fit" or which set of parameter values for (m & b) would be good, but how do we choose the most optimal parameter values ?

For example, among a set of values for M & B, we can use the LOSS function to find which parameters are the best for our model. But how do we find these "set of values" for our parameters ?

This is where Gradient Descent Comes in...

## GRADIENT DESCENT

It is an optimization algorithm which is commonly-used to train machine learning models and neural networks. Training data helps these models learn over time, and the cost function within gradient descent specifically acts as a barometer, gauging its accuracy with each iteration of parameter updates.

In Simple words , "Gradient Descent" helps us find the most optimal parameter values which give the lowest LOSS function value.

### How Gradient Descent Works ?

Step 1 ] Set all the parameters to 0 at start
Step 2 ] Calculate the LOSS
Step 3] Update the parameter using "UPDATE FORMULA"
Step 4] Repeat Step 2 and 3 until you reach a Global Minimum Loss or Until your Loss is decreasing.


NOTE : Basically Rather than randomly choosing values for parameters ,we are giving a direction to our values using Gradient Descent.

### PARAMETER UPDATE FORMULA

$$w_{t+1} = w_t - \alpha \frac{\partial L}{\partial w_t}$$

Derivative of LOSS function w.r.t Parameter

New Value for Parameter
Old Value of Parameter
Learning Rate

Learning rate is a constant which decides how fast we want the algorithm to learn. A common value for learning rate is 0.01

Example , Lets say the current value of "m" is 0 and LR is 0.01 , and also the derivative of "m" is -24 then using the UPDATE FORMULA its value will be :

m = m - (0.01)*(-24)
m = 0 + 0.24
m = 0.24

**Basically, Its very simple , we have a LOSS FUNCTION and we try to reduce it by trying a large number of different parameter values.**

## Implementing Linear Regression in Python

NOTE : Dont try to understand or Implement these ML Algorithms from scratch,Not unless you really need to .