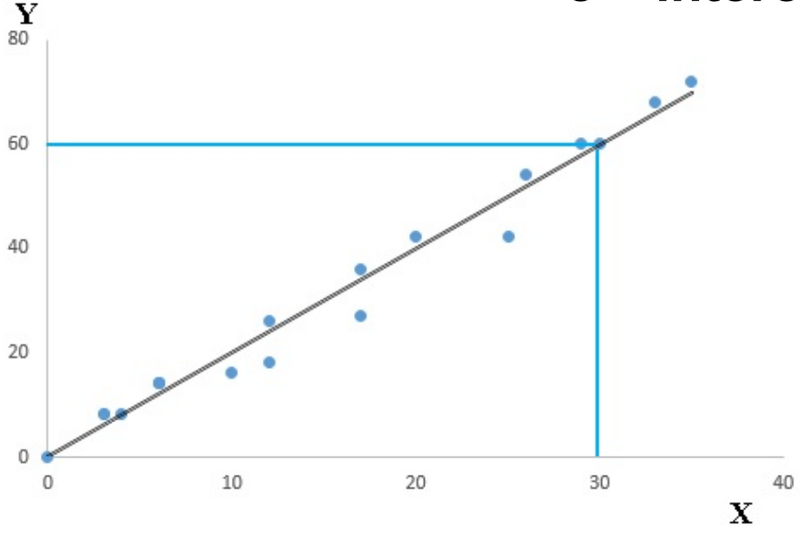
1.

What is Linear Regression?

- Linear Regression is used for predictive analysis. It is a technique which explains the degree of relationship between two or more variables (multiple regression, in that case) using a best fit line / plane.
- Simple Linear Regression is used when we have, one independent variable and one dependent variable.
- Regression technique tries to fit a single line through a scatter plot.

The simplest form of regression with one dependent and one independent variable is defined by the formula:

$$Y = mX + c$$; m = slope
c = intercept



2.

Hypothesis:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

Parameters:

$$\theta_0, \theta_1$$

Cost Function:

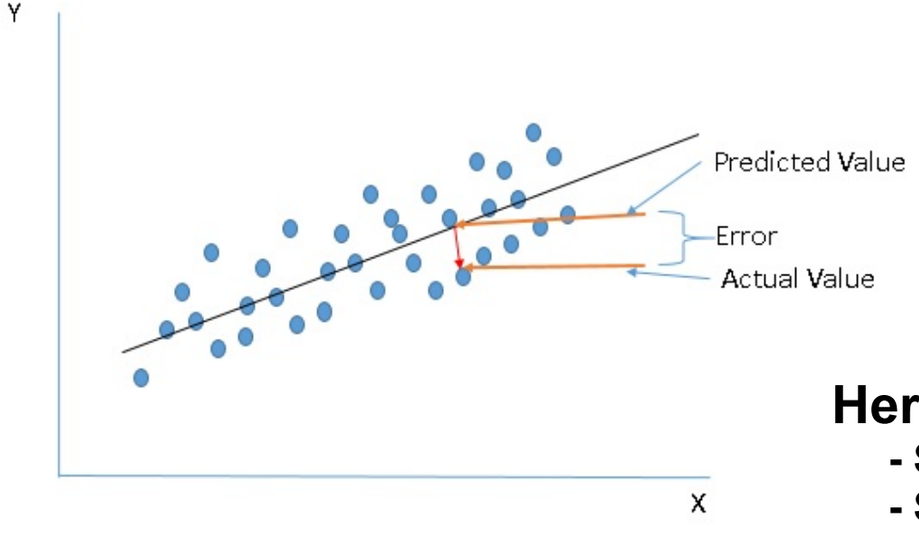
$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

3. **There can be multiple regression lines those can pass through the data points. So, how to choose the best fit line or value of co-efficients m and c.**

How to find the best regression line?

- We discussed above that regression line establishes a relationship between independent and dependent variable(s).
- A line which can explain the relationship better is said to be best fit line.
- The best fit line tends to return most accurate value of Y based on X i.e. causing a minimum difference between actual and predicted value of Y (lower prediction error).

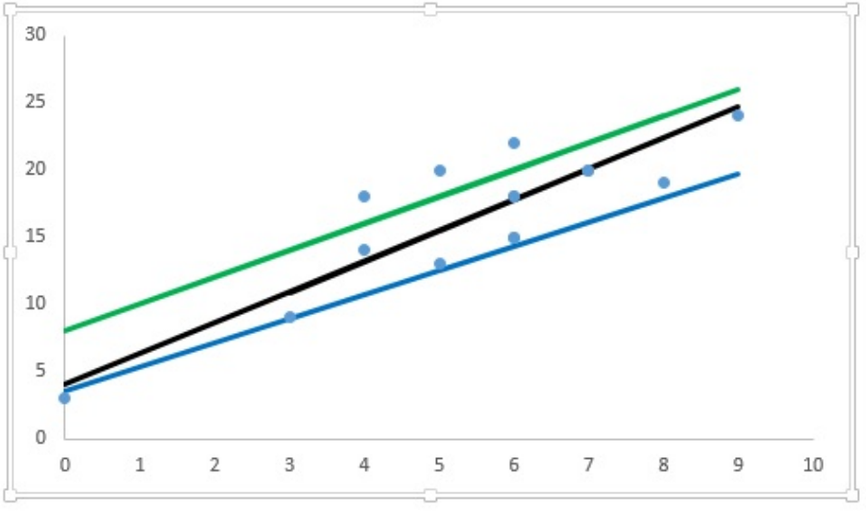


Here are some methods which check for error:

- Sum of all errors (∑error)
- Sum of absolute value of all errors (∑|error|)
- Sum of square of all errors (∑error^2)

4.

(y=2.3x+4, y=1.8x+3.5 and y=2x+8)



		Predicted Value			Error			|Error|			Error^2		
X	Y	Y= 2.3x+4	Y=1.8x+3.5	Y=2x+8	Y= 2.3x+4	Y=1.8x+3.5	Y=2x+8	Y= 2.3x+4	Y=1.8x+3.5	Y=2x+8	Y= 2.3x+4	Y=1.8x+3.5	Y=2x+8
8	19	22.4	17.9	24	-3.4	1.1	-5	3.4	1.1	5	11.56	1.21	25
0	3	4	3.5	8	-1	-0.5	-5	1	0.5	5	1	0.25	25
6	15	17.8	14.3	20	-2.8	0.7	-5	2.8	0.7	5	7.84	0.49	25
3	9	10.9	8.9	14	-1.9	0.1	-5	1.9	0.1	5	3.61	0.01	25
6	15	17.8	14.3	20	-2.8	0.7	-5	2.8	0.7	5	7.84	0.49	25
5	13	15.5	12.5	18	-2.5	0.5	-5	2.5	0.5	5	6.25	0.25	25
9	24	24.7	19.7	26	-0.7	4.3	-2	0.7	4.3	2	0.49	18.49	4
7	20	20.1	16.1	22	-0.1	3.9	-2	0.1	3.9	2	0.01	15.21	4
4	14	13.2	10.7	16	0.8	3.3	-2	0.8	3.3	2	0.64	10.89	4
6	18	17.8	14.3	20	0.2	3.7	-2	0.2	3.7	2	0.04	13.69	4
7	20	20.1	16.1	22	-0.1	3.9	-2	0.1	3.9	2	0.01	15.21	4
6	18	17.8	14.3	20	0.2	3.7	-2	0.2	3.7	2	0.04	13.69	4
4	18	13.2	10.7	16	4.8	7.3	2	4.8	7.3	2	23.04	53.29	4
6	22	17.8	14.3	20	4.2	7.7	2	4.2	7.7	2	17.64	59.29	4
5	20	15.5	12.5	18	4.5	7.5	2	4.5	7.5	2	20.25	56.25	4
			Sum		-0.6	47.9	-36	30	48.9	48	100.26	258.71	186

5.

From the previous table we can say:

- Sum of all errors (∑error): Using this method leads to cancellation of positive and negative errors, which certainly isn't our *motive*. Hence, it is not the right method.

- The other two methods perform well but, if you notice, **∑error^2, we penalize the error value much more compared to ∑|error|**. You can see that two equations has almost similar value for ∑|error| whereas in case of ∑error^2 there is significant difference

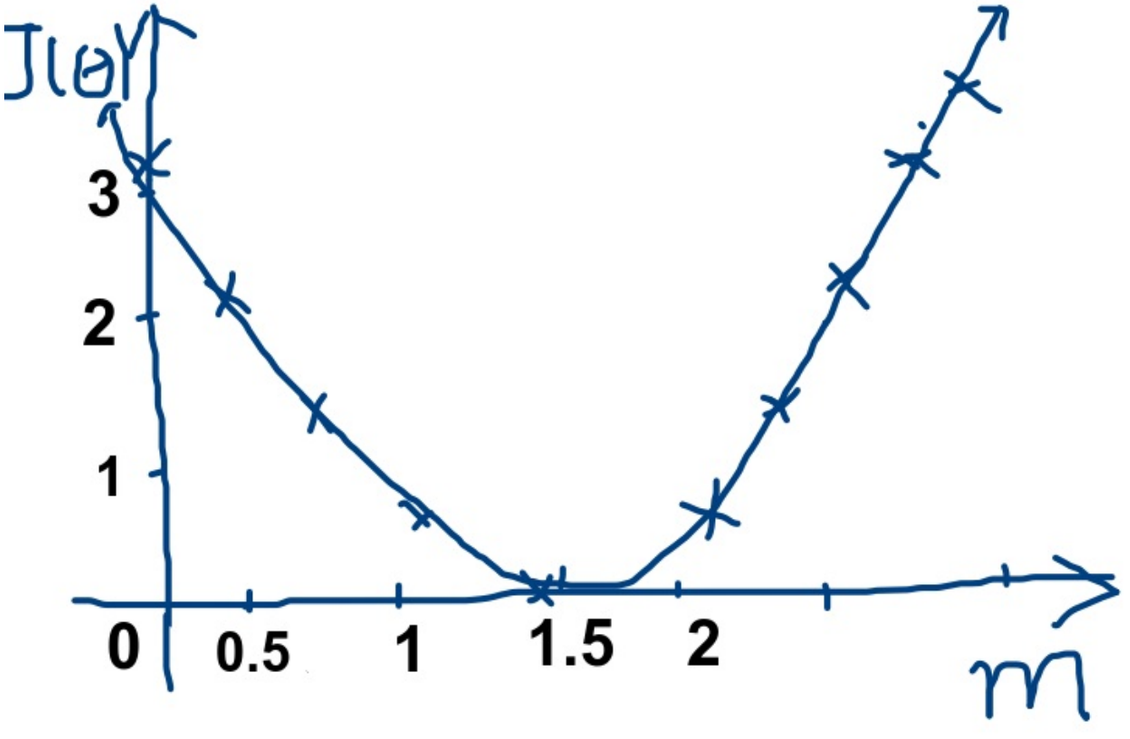
Therefore, we can say that these coefficients m and c are derived based on minimizing the sum of squared difference of distance between data points and regression line

There are two common algorithms to find the right coefficients for minimum sum of squared errors

1. Ordinary Least Sqaure (OLS, used in python library sklearn)

6.

$$J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$



Minimize $J(\theta_0, \theta_1)$
θ_0, θ_1

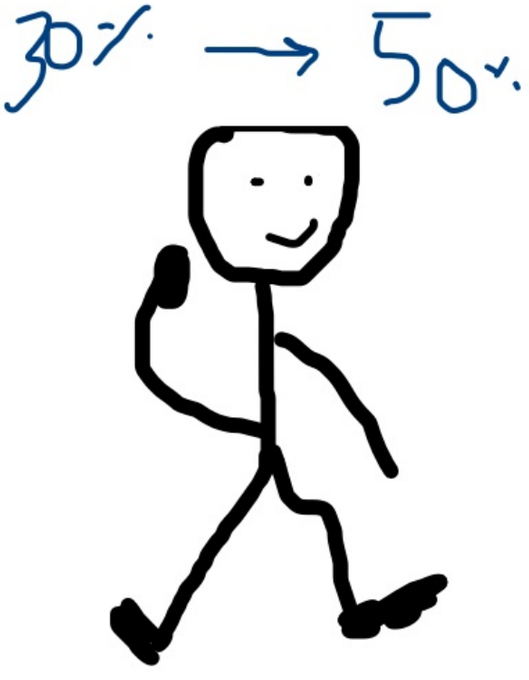
Cost fuction

Gradient descent

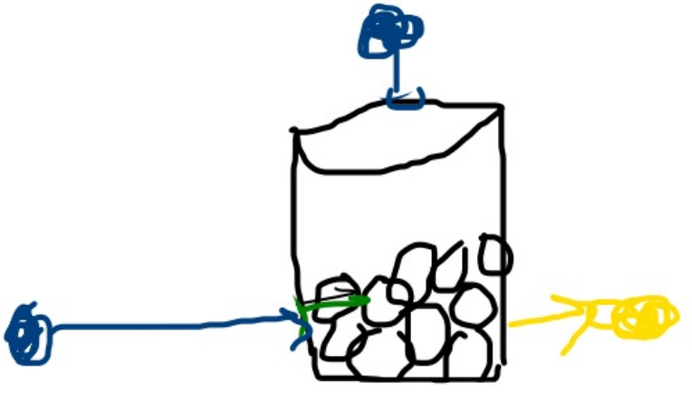
- optimization technique
- you can start with any point in this curve
- minimize the prediction with interation
- get the best (optimizes) perameters for the model
- untill model converges to minimal cost

7.

Gradient descent



45%



what you do?

1. random intialization
2. calculate the cost(error)
3. update the strength accordingly

8.

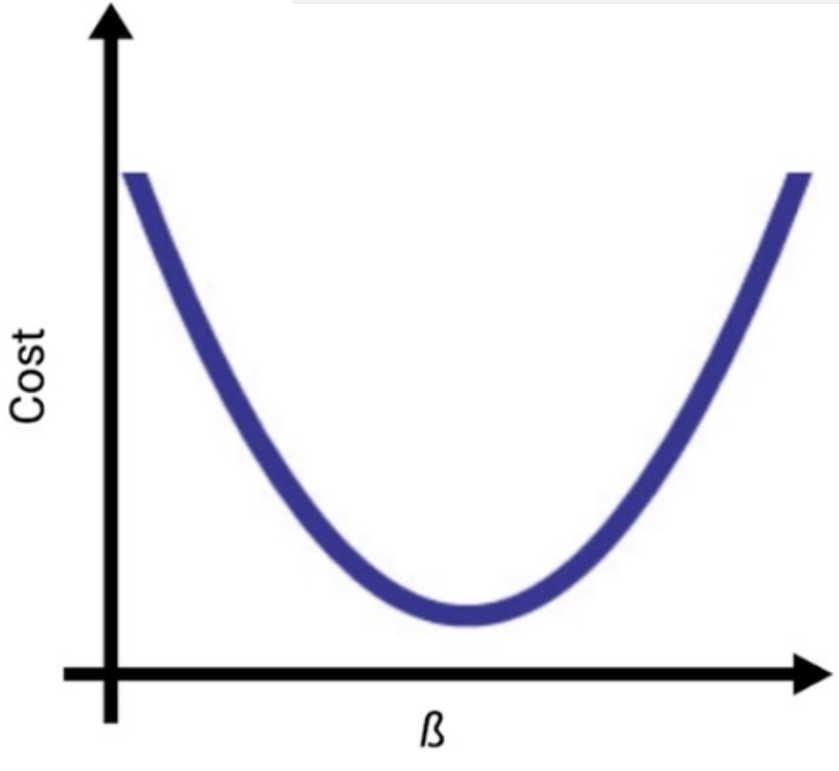
Have some function $J(\theta_0, \theta_1)$

Want $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

Outline:

- Start with some θ_0, θ_1
- Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$

  until we hopefully end up at a minimum



Reapeat untill converge

{

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \quad (\text{for } j = 0 \text{ and } j = 1)$$

}

$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$

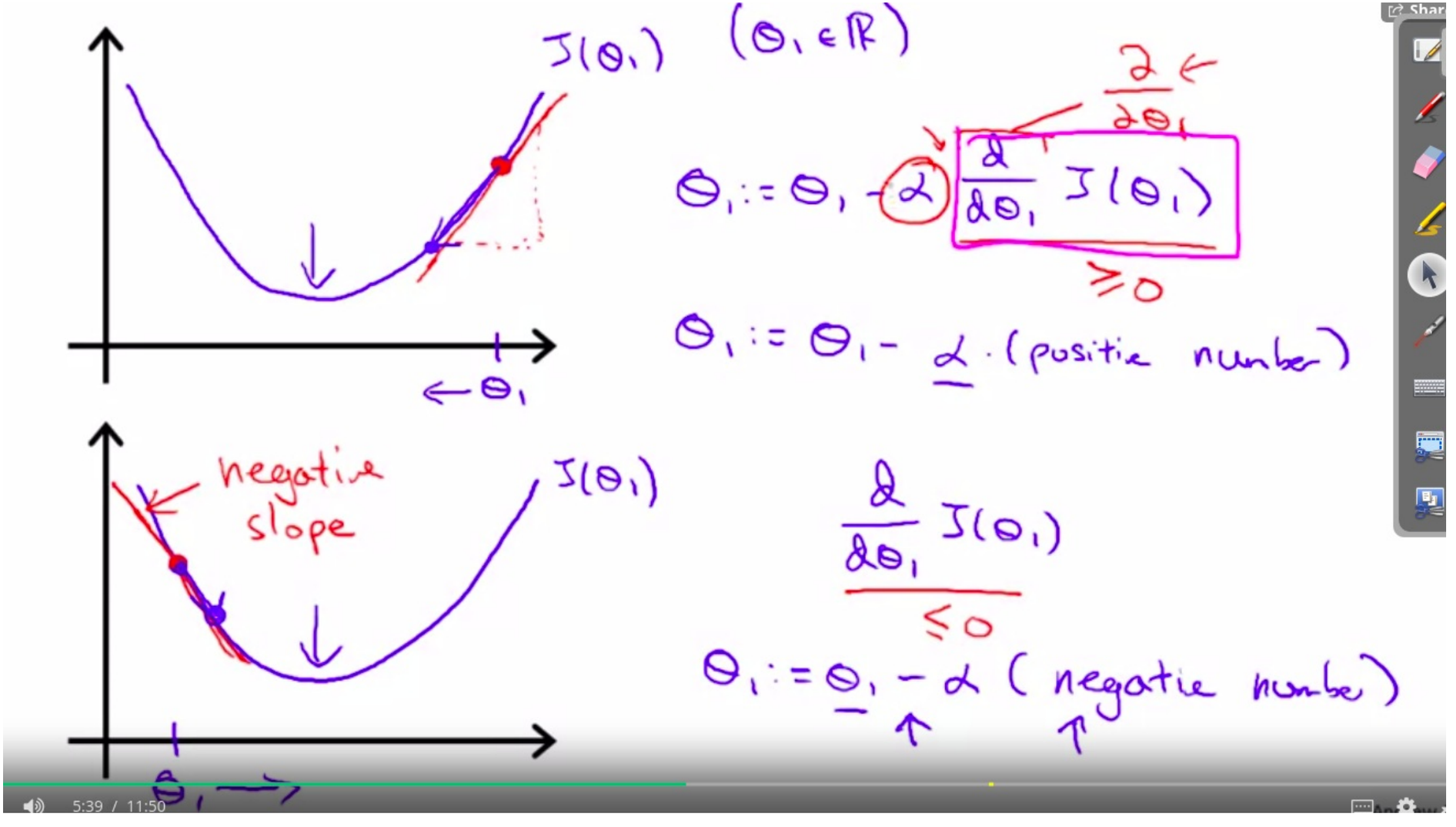
$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$

$\theta_0 := \text{temp0}$

$\theta_1 := \text{temp1}$

; Alpha: it is step size by which param is updates

9.

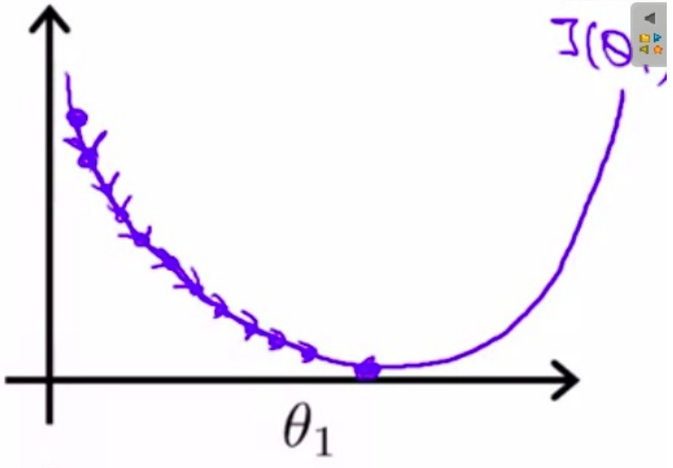

J(θ1)
(θ1 ∈ ℝ)
θ1
negative slope
J(θ1)
θ1 :=
α
(positive number)
≥ 0
≤ 0
(negative number)
5:39 / 11:50

10.

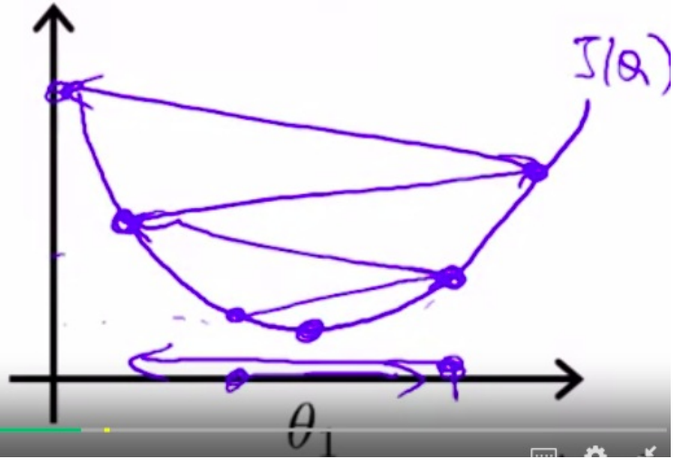
How Value of Alpha can affect!

$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

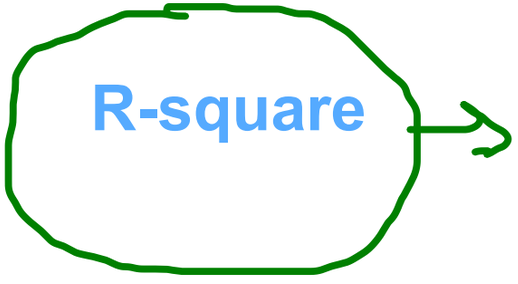
If α is too small, gradient descent can be slow.



If α is too large, gradient descent can overshoot the minimum. It may fail to converge, or even diverge.



11. What are the performance evaluation metrics in Regression?



"How much the change in output variable (y) is explained by the change in input variable(x).

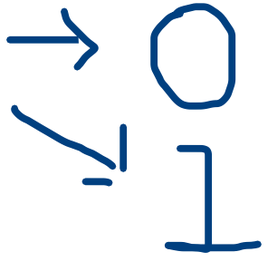
$$\text{R-Square} = 1 - \frac{\sum(Y\_actual - Y\_predicted)^2}{\sum(Y\_actual - Y\_mean)^2}$$

Sum Squared Regression Error

$$R^2 = 1 - \frac{SS_{Regression}}{SS_{Total}}$$

Sum Squared Total Error

Always between 0 to 1



indicates that the model explains NIL variability in the response data

indicates that the model explains full variability in the response data

Higher the R^2, more robust will be the model

12.

One disadvantage of R-squared is that it can only increase as predictors are added to the regression model. This increase is artificial when predictors are not actually improving the model's fit. To cure this, we use "Adjusted R-squared".

The adjusted R-squared increases only if the new term improves the model more than would be expected by chance.

Adjusted R-squared is nothing but the change of R-square that adjusts the number of terms in a model.

Formula:

$$R^2_{adjusted} = 1 - \frac{(1 - R^2)(N - 1)}{N - p - 1}$$

where

R^2 = sample R-square

p = Number of predictors

N = Total sample size.

Implementation

https://repl.it/@LakshayArora1/Linear-Regression

What is Multi-Variate Regression?

Once you have identified the level of significance between independent variables(IV) and dependent variables(DV), use these significant IVs to make more powerful and accurate predictions. This technique is known as "Multi-variate Regression".

In an multiple regression model, we try to predict

$$Y = a + b_1X_1 + b_2X_2 + \ldots\ldots + b_kX_k$$

; a = intercept
b = slope

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4$$

$$\rightarrow h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

For convenience of notation, define $x_0 = 1$. $(x_0^{(i)} = 1)$

$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1}$ $\quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}$

Repeat {

$\rightarrow \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \ldots, \theta_n)$

}

(simultaneously update for every $j = 0, \ldots, n$)

$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \cdots + \theta_n x_n$ ($x_0 = 1$)

$= \theta^T x.$

Cost function

$$J(\theta_0, \theta_1, \ldots, \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Gradient Descent

repeat until convergence: {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_0^{(i)}$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_1^{(i)}$$

$$\theta_2 := \theta_2 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_2^{(i)}$$

$\ldots$

}

New algorithm $(n \geq 1)$:

Repeat {

$\frac{\partial}{\partial \theta_j} J(\theta)$

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

(simultaneously update θ_j for $j = 0, \ldots, n$) }

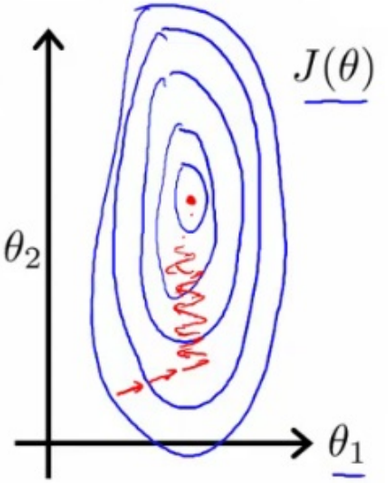
If we do not scale feature in multiple linear regression then, Gradient descent will not be able to converge

Let's say
if we have two features

E.g. x_1 = size (0-2000 feet^2)

x_2 = number of bedrooms (1-5)

then the grradient descent would have been like this



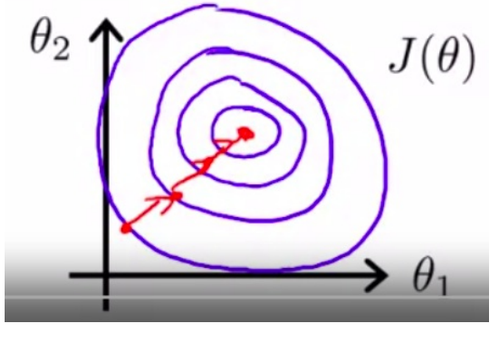
Which is difficult to converge

Feature scalling: Idea is to make sure features are on same scale

$$x_1 = \frac{\text{size (feet}^2\text{)}}{2000}$$

$$x_2 = \frac{\text{number of bedrooms}}{5}$$

$0 \le x_1 \le 1 \qquad 0 \le x_2 \le 1$



Which helps gradient descent to get converge easily and effientiently

Get every feature into approximately a $-1 \leq x_i \leq 1$ range.

Mean normalization

Replace x_i with $x_i - \mu_i$ to make features have approximately zero mean (Do not apply to $x_0 = 1$).

E.g. $x_1 = \frac{size - 1000}{2000}$

$x_2 = \frac{\#bedrooms - 2}{5}$

$-0.5 \leq x_1 \leq 0.5, -0.5 \leq x_2 \leq 0.5$

$x_1 = \frac{x_1 - \mu_1}{\text{max} - \text{min}}$

- For sufficiently small α, $J(\theta)$ should decrease on every iteration.
- But if α is too small, gradient descent can be slow to converge.

Normal Equation

- Normal equation gives much better way to obtain optimal perameter value for some regression

- Method to solve for theta analytically

$\theta \in \mathbb{R}^{n+1}$ $\quad J(\theta_0, \theta_1, \ldots, \theta_m) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

$\frac{\partial}{\partial \theta_j} J(\theta) = \cdots = 0$ (for every j)

Solve for $\theta_0, \theta_1, \ldots, \theta_n$

Examples: $m = 4$.

x_0	Size (feet²) x_1	Number of bedrooms x_2	Number of floors x_3	Age of home (years) x_4	Price ($1000) y
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix} \quad m \times (n+1)$$

$$y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix} \quad m\text{-dimensional vector}$$

$$\theta = (X^T X)^{-1} X^T y$$

Andrew Ng

$$\theta = (X^T X)^{-1} X^T y$$

Feature scalling does not required in this method

When to use which one?

m **training examples,** n **features.**

Gradient Descent

- Need to choose α.
- Needs many iterations.
- Works well even when n is large.

Normal Equation

- No need to choose α.
- Don't need to iterate.
- Need to compute $(X^T X)^{-1}$
- Slow if n is very large.

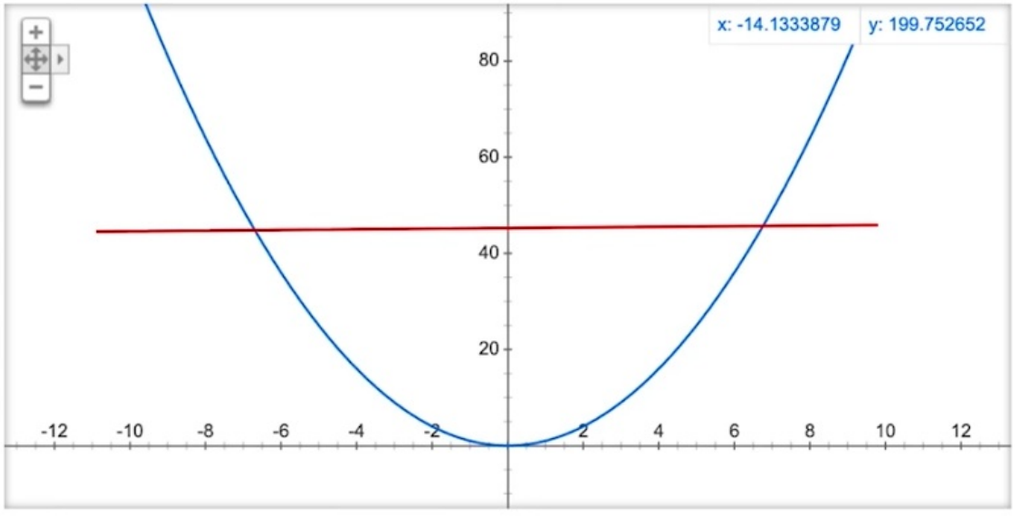
Gradient Descent	Normal Equation
Need to choose alpha	No need to choose alpha
Needs many iterations	No need to iterate
O (kn^2)	O (n^3), need to calculate inverse of $X^T X$
Works well when n is large	Slow if n is very large

Assumptions we make before applying Linear Regresion

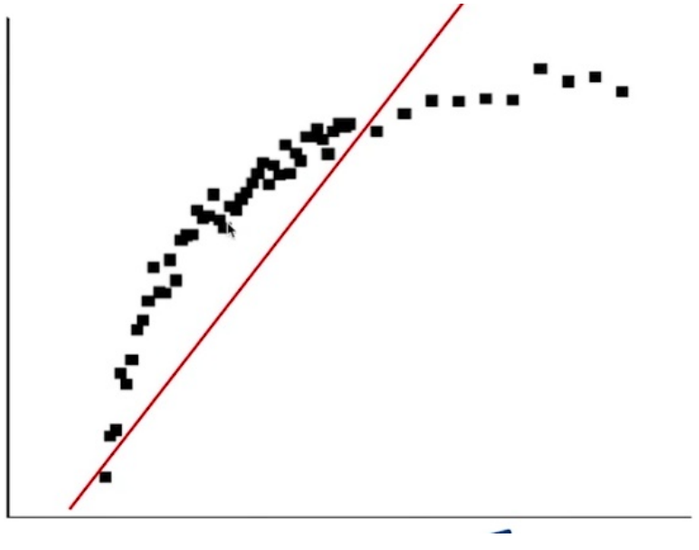
1. Linear Relationship
2. No Correlation of Error Term
3. Constant Variance of Error Terms
4. No Correlation among independant Variables
5. Errors Normally distributed

1. Linear Relationship

- there has to be linear relationship between dependant and independant variable
- if not then we may try to transform or not use LR in that case



TRANSFORM



Not Linear relationship

Linear

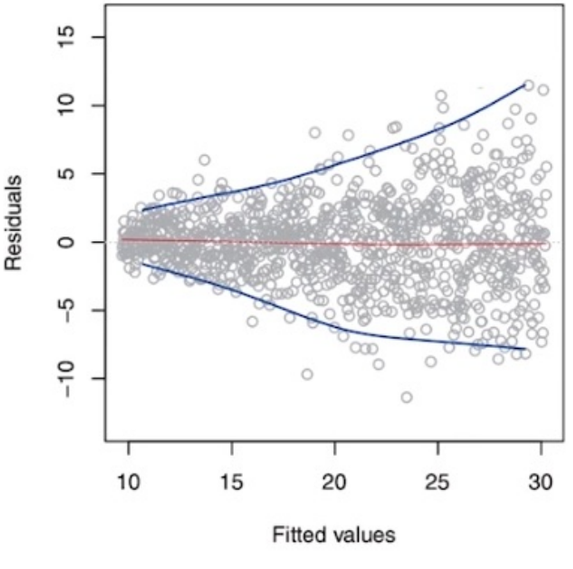
-> Transform using: log(x), sqrt of x, x^2

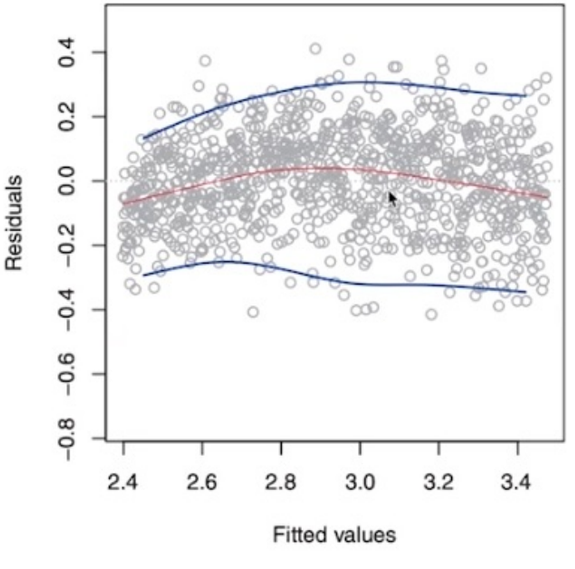
2. No Correlation of Error Term

- When we plot the residuals on a chart, there should not be underline term in it.
- that means,
my previous value of residual should not help me predict the fururе (next) value of residual

3. Constant Variance of Error Terms

Not correct





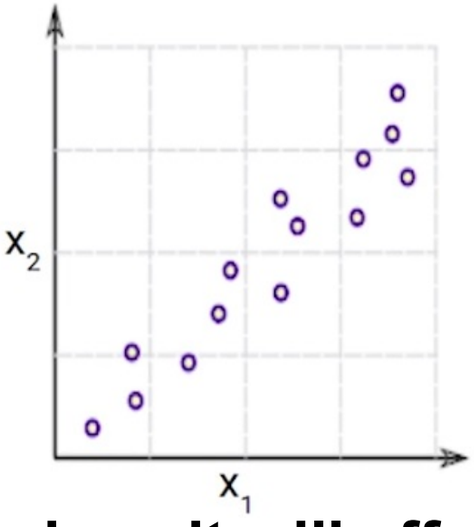
correct

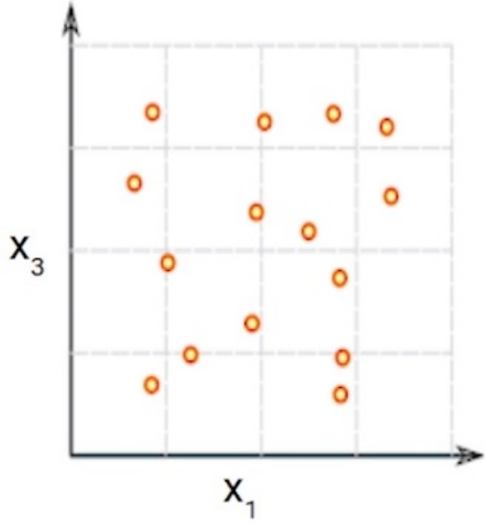
- if we plot the residuals and we observ there is some trend in variance in it, then that will violte our assumtion... to avoid it we need to transform using any of the technique

4. Multi - collinearity

- it essentially means there is some colinearity between dependant variables

$$Y = \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + b$$





Here we can observ that there is high corelation between x1 and x2

how it will affect?

- every time when i try to figure-out relation between y and x1, it will also lead to change in x2 as well

- it will still give me regression analysis but the model would become voletile (Unpredictable)

- To overcome this problem, we can elemenet any one variable...

$$Y = \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + b$$

Read about

5. Errors Normally distributed