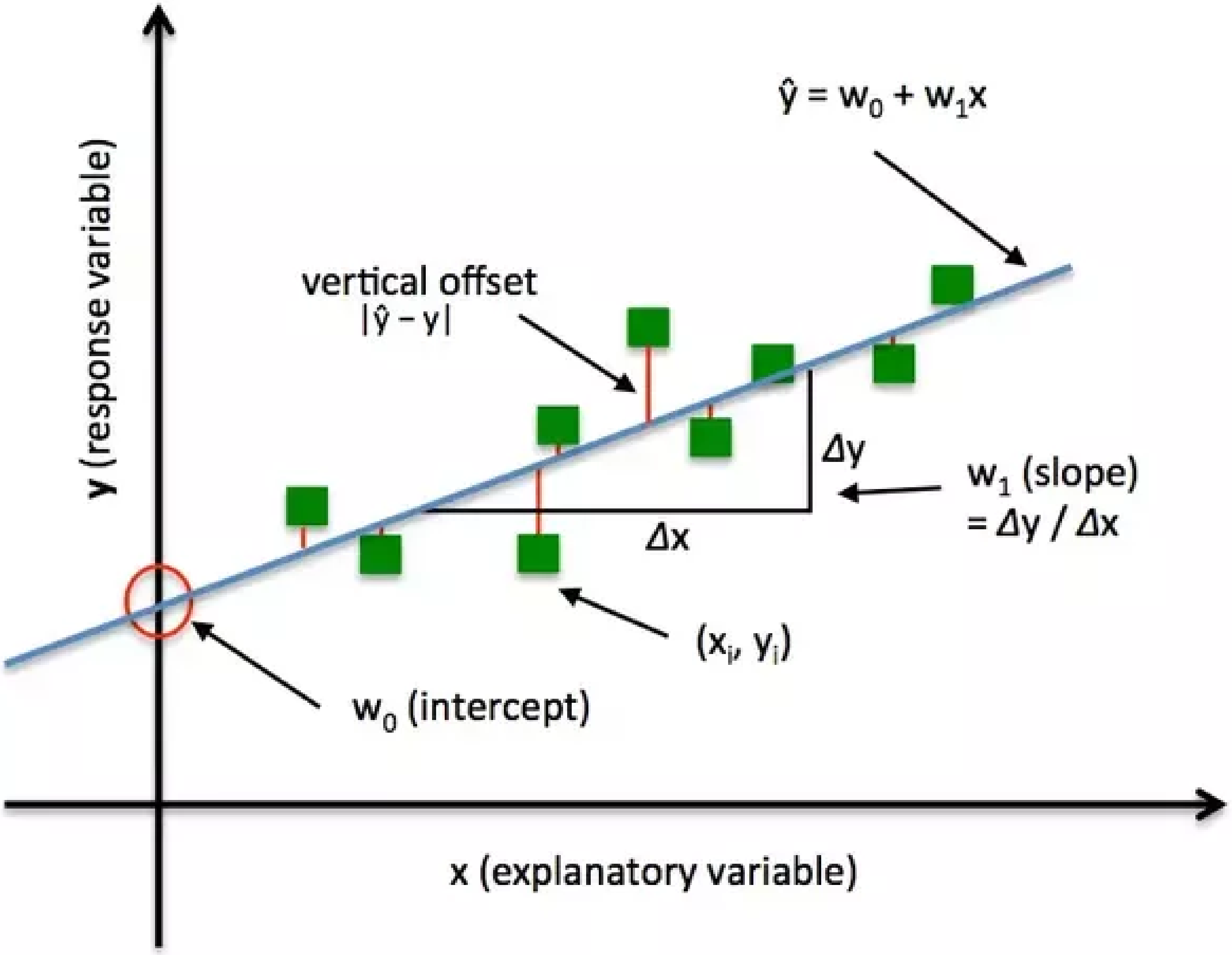
Linear regression



Assumptions of linear regression

Linear regression has the following **assumptions**, failing which the linear regression model does not hold true:

1. The dependent variable should be a linear combination of independent variables
2. No autocorrelation in error terms
3. Errors should have zero mean and be normally distributed
4. **No or little multi-collinearity**
5. Error terms should be homoscedastic

Optimization in Machine Learning

(A) Deterministic Problems

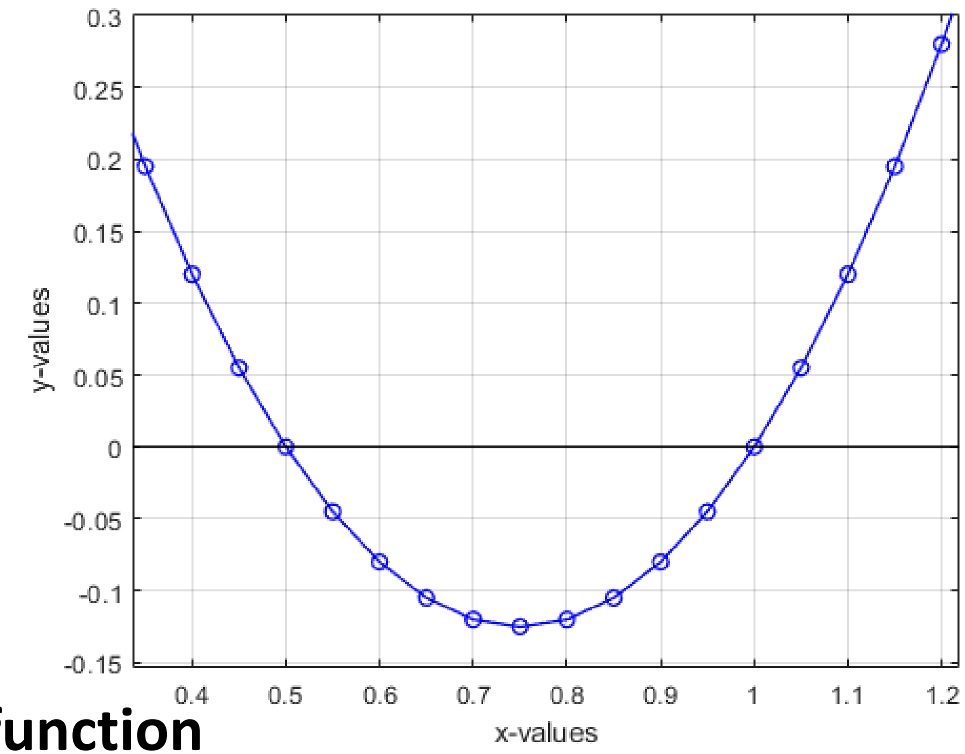
Q. Find the roots of the $f(x)$, where $f(x) = 2x^2 - 3x + 1$

$$Solution: x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = 0.5 \ and \ 1.0$$

(B) Convex Optimization Problems

Q. $\min f(x)$, where $f(x) = 2x^2 - 3x + 1$

Q. $\underset{x}{\operatorname{argmin}} \ f(x)$, where $f(x) = 2x^2 - 3x + 1$



f(x) is called as objective function, **cost function / Loss function**

Optimization in Machine Learning

(A) Deterministic Problems: Q. Find the roots of the $f(x)$, where $f(x) = 2x^2 - 3x + 1$

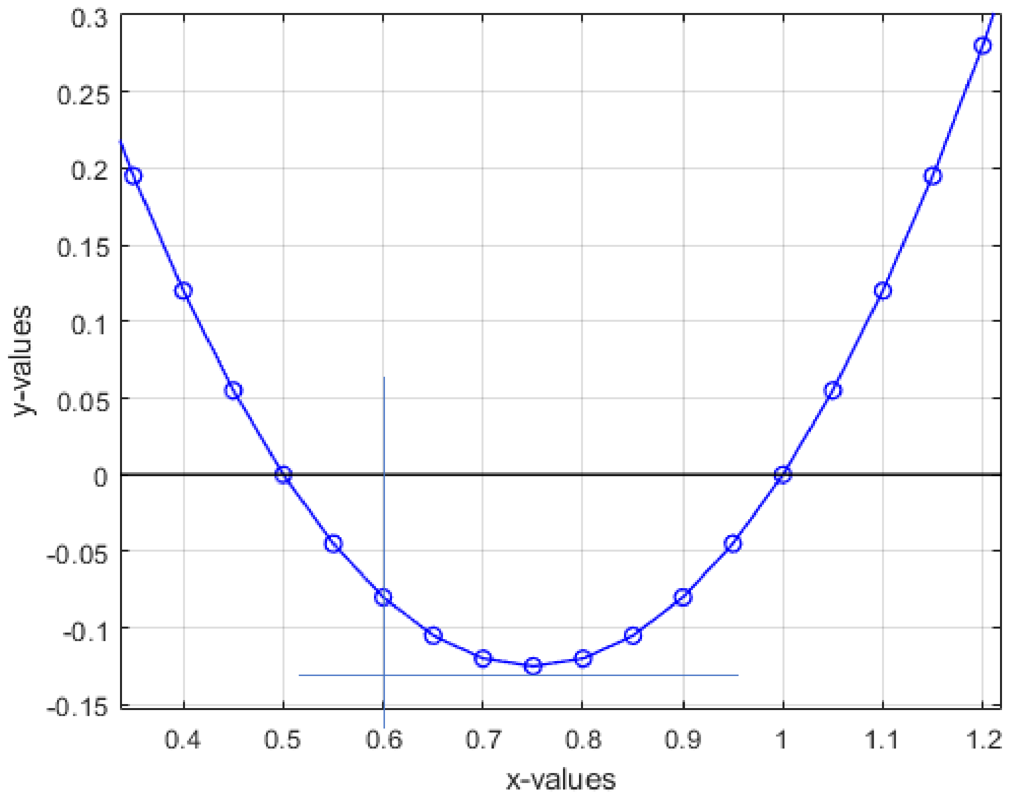
$$Solution: x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = 0.5 \; and \; 1.0$$

(B) Optimization Problems:
minimization/maximization of an objective function, subject to some constraints.

$$\min f(x), \text{ where } f(x) = 2x^2 - 3x + 1$$

$$\underset{x}{\operatorname{argmin}} \; f(x), \text{ where } f(x) = 2x^2 - 3x + 1$$

$$\underset{x}{\operatorname{argmin}} \; f(x), \text{ where } f(x) = 20x^2 - 30x + 10$$



Both lead to the same “optimal values” of x

Optimization in Machine Learning

Types of Optimization Problems, in general:

(A) Unconstrained Optimization

$$\min f(x), \text{ where } f(x) = 2x^2 - 3x + 1$$

$$\underset{x}{\operatorname{argmin}}\ f(x), \text{ where } f(x) = 2x^2 - 3x + 1$$

(B) Constrained Optimization:

a. Linear or non-linear constraints

b. Equality or inequality constraints

$$\min\ (x_1 - 2)^2 + (x_2 - 1)^2 \quad \text{subject to} \begin{cases} x_1^2 - x_2 & \leq 0, \\ x_1 + x_2 & \leq 2. \end{cases}$$

Possible Loss (Cost) functions in ML (Regression)

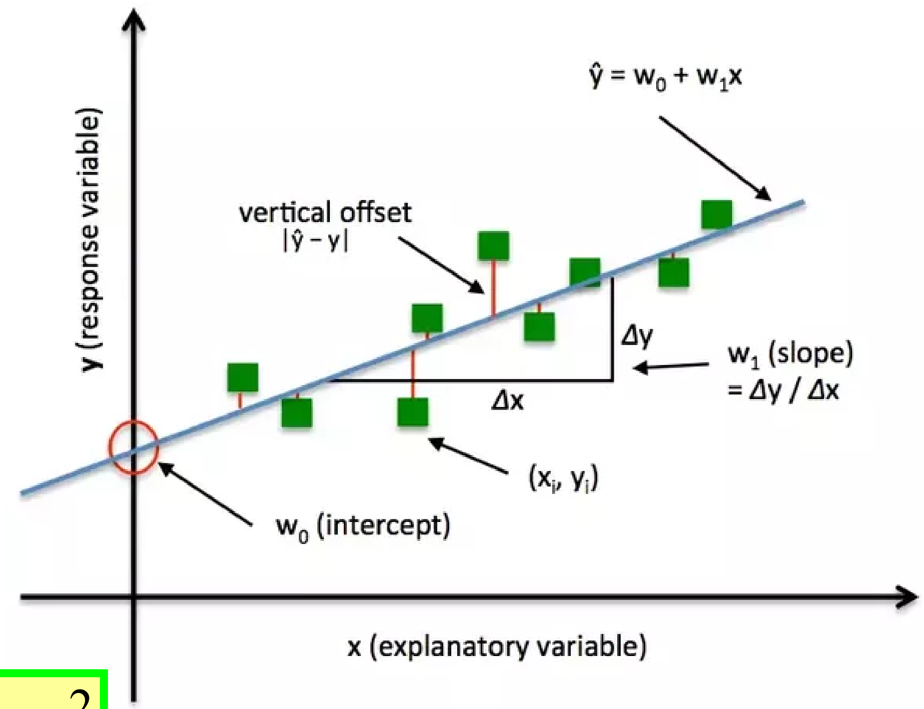
(1) Sum of errors (SE): $L = \sum_{i=1}^{N} \left(\hat{Y}_i - Y_i \right)$

(2) Sum of Absolute Errors (SAE): $L = \sum_{i=1}^{N} \left| \hat{Y}_i - Y_i \right|$

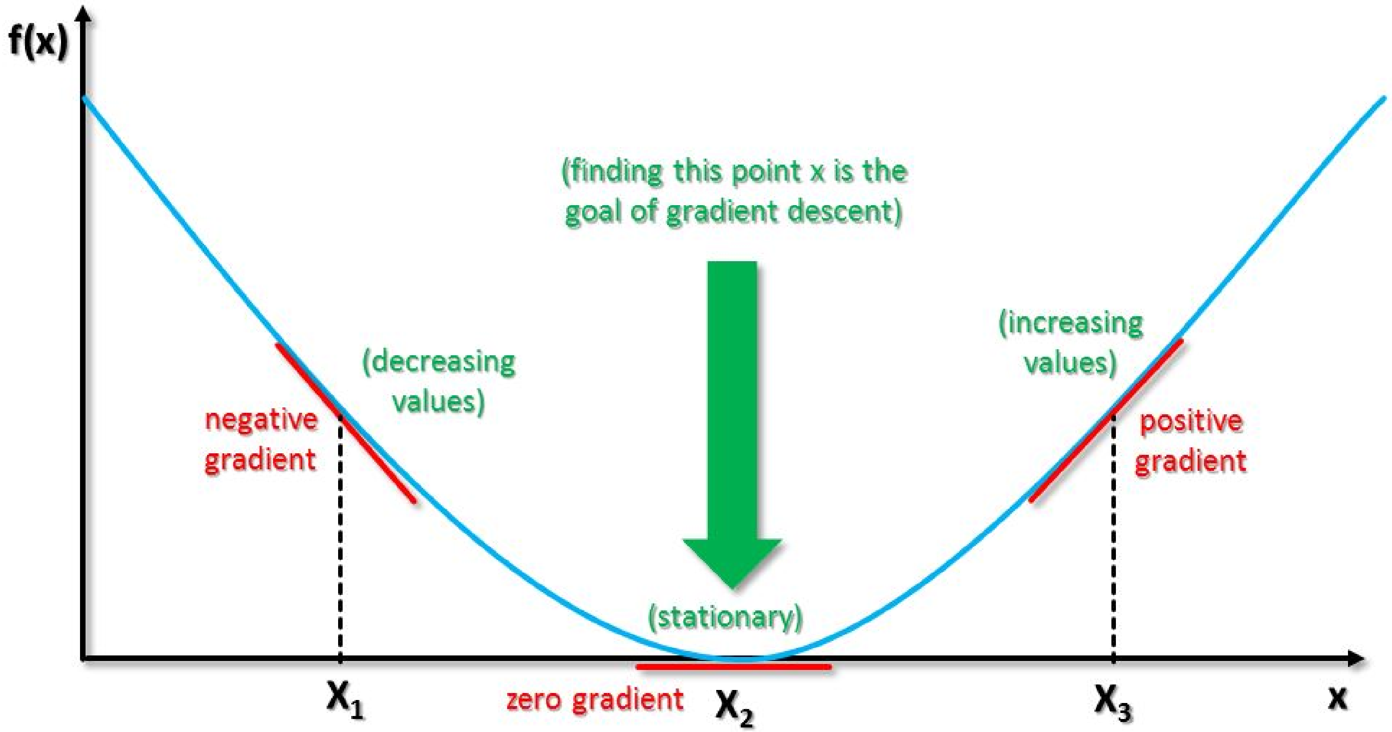
(3) Sum of Squares of Errors (SSE): $L = \sum_{i=1}^{N} \left(\hat{Y}_i - Y_i \right)^2$

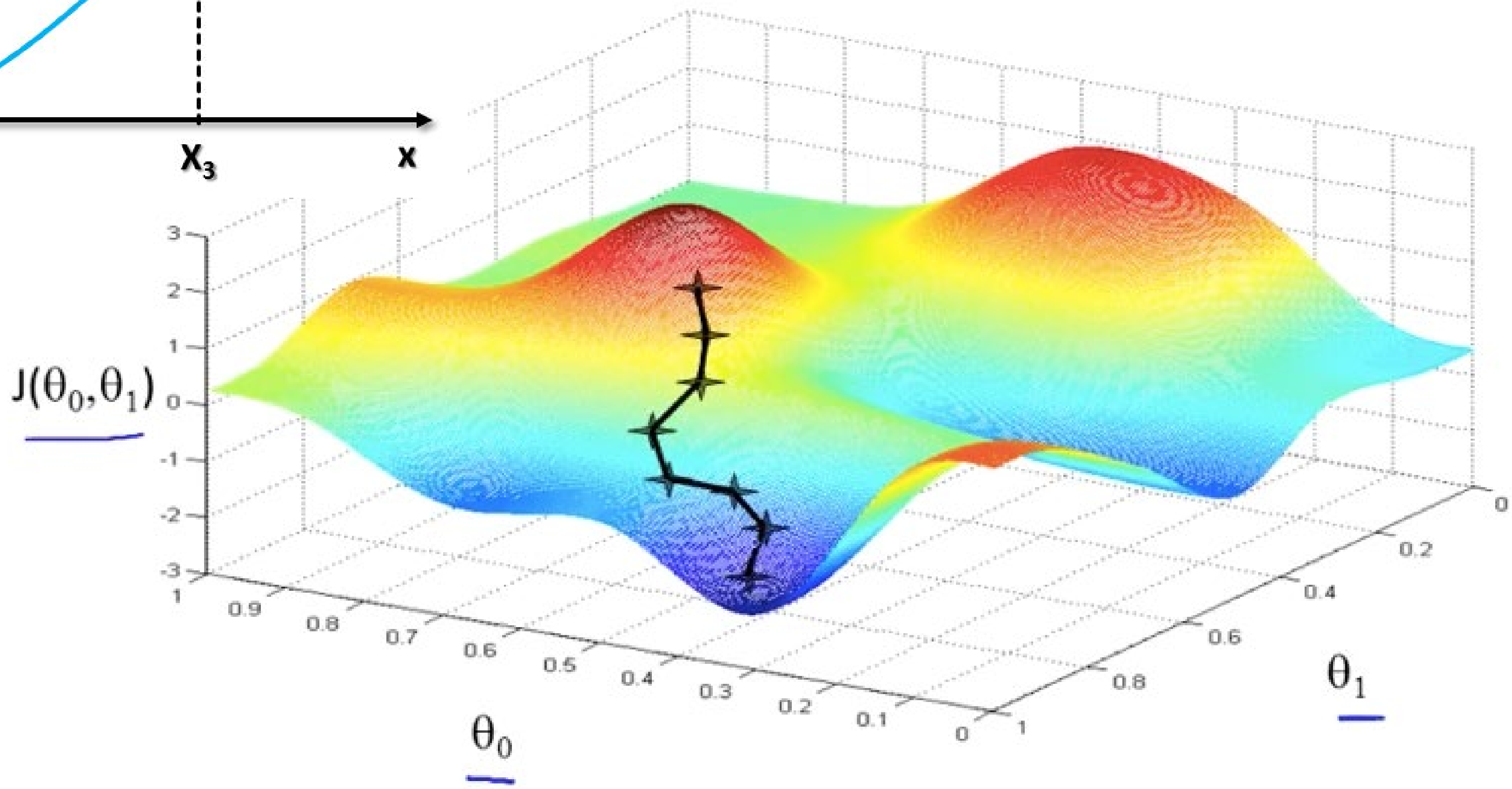
(4) Mean of Squares of Errors (MSE): $L = \frac{1}{N} \sum_{i=1}^{N} \left(\hat{Y}_i - Y_i \right)^2$

(5) Root Mean of Squares of Errors (RMSE): $L = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(\hat{Y}_i - Y_i \right)^2}$



X1	Y
1	4.8
3	11.4
5	17.5
..	...


f(x)
(finding this point x is the goal of gradient descent)
(decreasing values)
negative gradient
(increasing values)
positive gradient
(stationary)
X_1
zero gradient
X_2
X_3
x


$J(\theta_0,\theta_1)$
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ_0
0
0.2
0.4
0.6
0.8
1
θ_1

Linear Regression Problem Formulation:

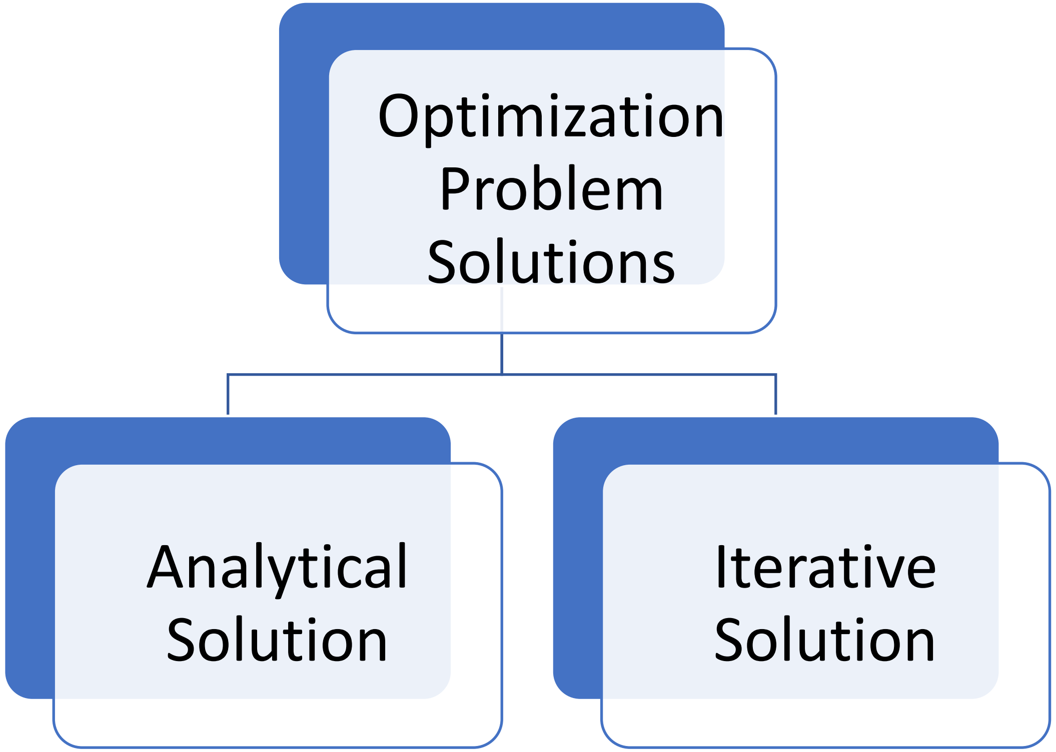
$$Objective\ Fn: MSE:\ L = \frac{1}{N}\sum_{i=1}^{N}(\hat{Y}_i - Y_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(error)^2$$

$$\boxed{\arg\min_{w_j} L(w_j \mid X, Y)} \quad where,\ \ \hat{Y} = w_0 + w_1 X_1$$

X1	Y
1	4.8
3	11.4
5	17.5
..	…

Says, Find the **optimal weights (w_j)** for which the **MSE** Loss function has **min value**, for a **GIVEN X,Y data.**

Optimization Problem Solution Methods:



- **Theoretical Solution**, which gives the “exact” solution to the problem, provided the optim. problem has a “closed-form solution”

-**Approximate Solution** to the optim. problem, based on some iterative algorithm
- Can solve all types of optim. problems.

Analytical Solution for Unconstrained Optimization:

FOC: Necessary Condition: States that the first (odd) derivative (gradient) of the objective function must vanish at the optimal points (points of maxima/minima)

$$e.g.\ f(x) = 2x^2 - 3x + 1 \qquad \frac{df}{dx} = 4x - 3 = 0 \qquad \Rightarrow x^* = 0.75$$

SOC: Sufficiency Condition: States that the second (even) derivative (gradient) of the objective function **evaluated at the optimal points**, must be:

- **Positive >> for minima**
- Negative >> for maxima

$$\left|\frac{d^2 f}{dx^2}\right|_{x^*=0.75} = 4 \ \ (positive)$$

which means minima occurs at $x^* = 0.75$

c) worst-case scenario: d2f/dx2 evaluated at some optimal x is ZERO...,
THEN, that optimal point is neither a point of maxima/minima. It could just a point of inflexion or a saddle point.

if you want to maximize a fcn:
1) minimize the negative of the obj function
2) minimize the inverse of the obj fcn. >> used on SVM

Linear Regression – Analytical Solution:

$$MSE : L = \frac{1}{N}\sum\left(\hat{Y} - Y\right)^2 \qquad where, \quad \hat{Y} = w_0 + w_1 X_1$$

X1	Y
1	4.8
3	11.4
5	17.5

$$\boxed{\underset{w_j}{\arg\min}\; L(w_j \mid X, Y)}$$

Find the **optimal weights (wj)** for which the MSE Loss function has min value, for **GIVEN X,Y data**.

STEP – 1: Get the Gradients:

$$\frac{\partial L}{\partial w_j} = \frac{\partial L}{\partial \hat{Y}} \times \frac{\partial \hat{Y}}{\partial w_j}$$

$$\frac{\partial L}{\partial w_j} = \frac{2}{N}\sum\left(\hat{Y} - Y\right) \times \frac{\partial \hat{Y}}{\partial w_j}$$

FINAL GRADIENTS

$$\boxed{\begin{aligned} \frac{\partial L}{\partial w_0} &= \frac{2}{N}\sum\left(\hat{Y} - Y\right) \times 1 \\ \frac{\partial L}{\partial w_1} &= \frac{2}{N}\sum\left(\hat{Y} - Y\right) \times X_1 \end{aligned}}$$

Linear Regression – Analytical Solution:

STEP – 2: Equate the Gradients to zero and solve:

$$\frac{2}{N}\sum\left(\hat{Y}-Y\right)\times 1 \quad = 0 \qquad -- \quad (1)$$

$$\frac{2}{N}\sum\left(\hat{Y}-Y\right)\times X_1 \quad = 0 \qquad -- \quad (2)$$

FINAL SOLUTION ⟵ **Ordinary Least Squares (OLS) !**

$$\boxed{w_1^* = \frac{\left[\overline{XY} - \overline{X}\,\overline{Y}\right]}{\left[\left(\overline{X^2}\right) - \left(\overline{X}\right)^2\right]}}$$

$$\hat{Y} = w_0 + w_1 X_1$$

$$\overline{Y} = w_0 + w_1 \overline{X_1}$$

Intercept / Bias Term:

$$\boxed{w_0^* = \overline{Y} - \left(w_1^* \cdot \overline{X}\right)}$$

Linear Regression – Analytical Solution:

STEP – 3: Calculate the optimal weights

X	Y
1	4.8
3	11.4
5	17.5

$$w_1^* = \frac{\left[\overline{XY} - \overline{X}\,\overline{Y}\right]}{\left[\left(\overline{X^2}\right) - \left(\overline{X}\right)^2\right]}$$

$$w_0^* = \overline{Y} - \left(w_1^* \cdot \overline{X}\right)$$

Substitute:

$$w_1^* = \frac{\left[547.87 - (3 \times 11.67)\right]}{\left[(11.67) - (3)^2\right]}$$

	X	Y	X^2	XY
	1	4.8	1	1x4.8
	3	11.4	9	3x11.4
	5	17.5	25	5x17.5
Mean	9/3 = 3	35/3 =11.67	35/3 = 11.67	(1643.6)/3 = 547.87

Linear Regression – Analytical Solution:

$$w_1^* = \frac{\left[\overline{XY} - \overline{X}\,\overline{Y}\right]}{\left[\left(\overline{X^2}\right) - \left(\overline{X}\right)^2\right]}$$

is also equivalent to:

$$w_1{}^* = \frac{\sum_i \left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{\sum_i \left(X_i - \overline{X}\right)^2} \qquad = \frac{\mathrm{cov}(X, Y)}{\mathrm{var}(X)}$$

OLS in Vector Form:

$$L = \frac{1}{N}\sum\left(\hat{Y}_i - Y_i\right)^2 = \frac{1}{N}\left[\left(\hat{Y} - Y\right)^T\left(\hat{Y} - Y\right)\right]$$

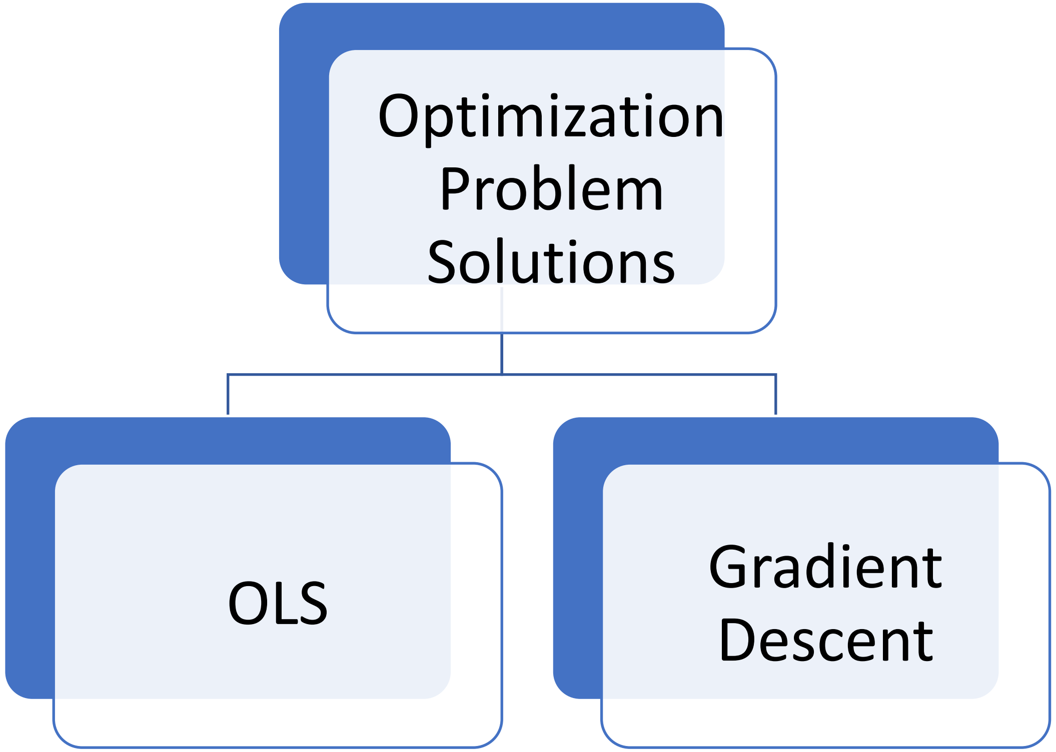
Loss Function in Vector Notation

$$W = (X^T X)^{-1} X^T Y$$

OLS Solution (Analytical Solution) !

$$\frac{\partial^2 L}{\partial W^2} = \frac{1}{N}\left[(X^T X)(2)\right] = +ve \ \ \text{(minima)}$$

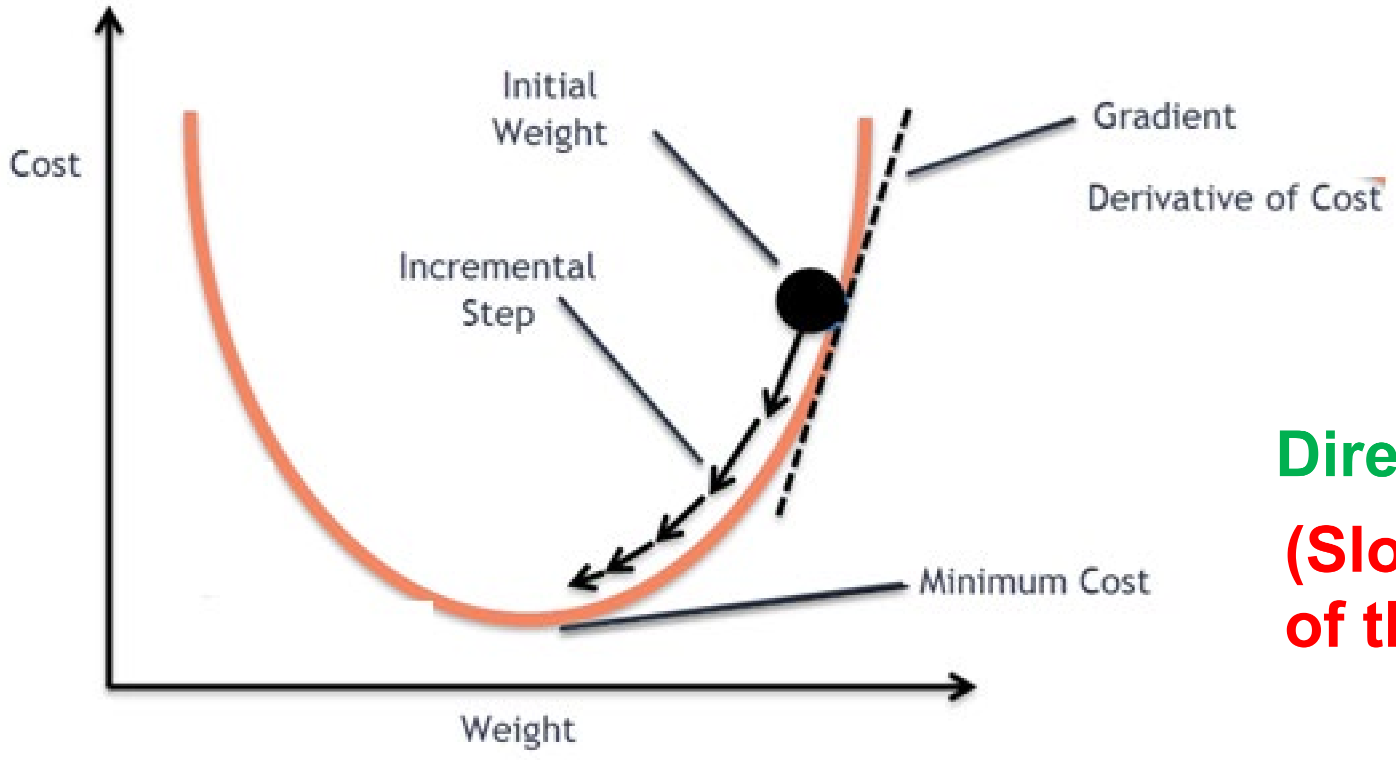
Optimization Problem Solution Methods:



- Analytical Solution, which gives the “exact” solution to the problem, provided the optim. problem has a “closed-form solution

-Approximate Solution (iterative) to the optim. problem, based on some iterative algorithm
- Can solve all types of optim. problems.

Gradient Descent Algorithm:



Gradient Descent Update Rule:

$$w_j^{k+1} = w_j^k - \Delta w_j$$

Direction of Update

(Slope / Gradient of the Loss Fn)

$$\frac{\partial L}{\partial w_j}$$

$$\frac{\partial L}{\partial w_0} = \sum \left(\hat{Y} - Y \right) \times 1$$

$$\frac{\partial L}{\partial w_1} = \sum \left(\hat{Y} - Y \right) \times X_1$$

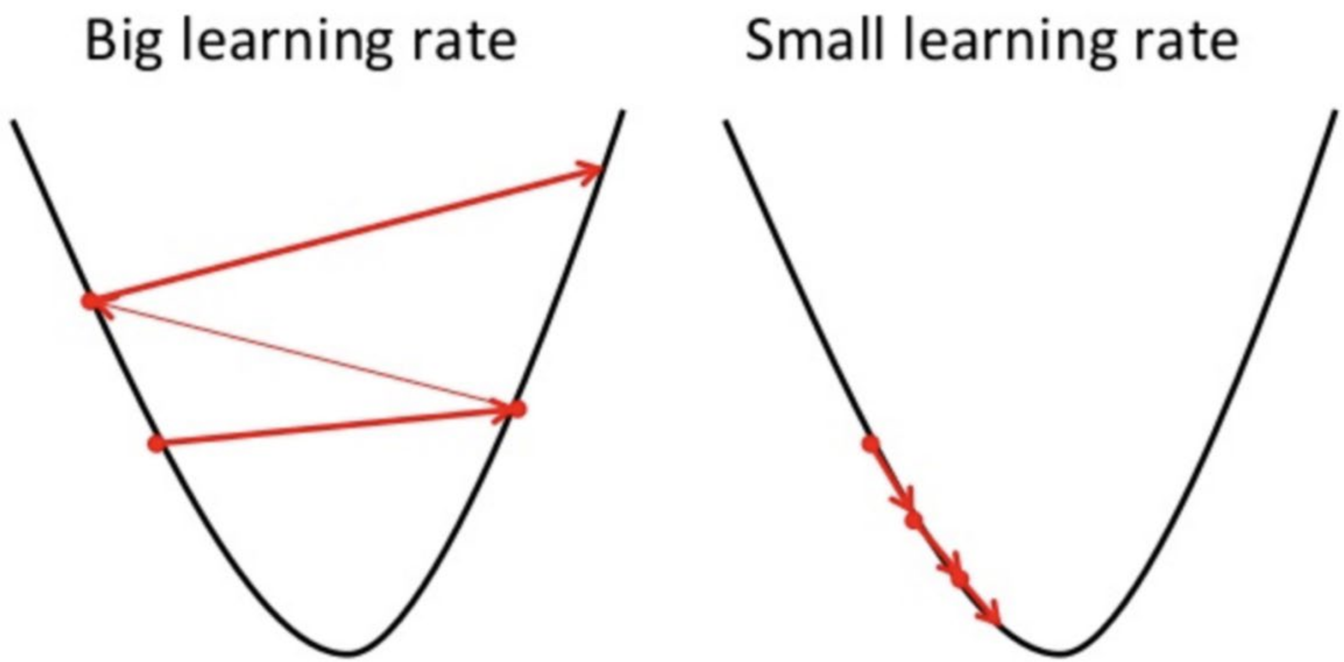
Amount of Update

Step Size: α or Learning Rate

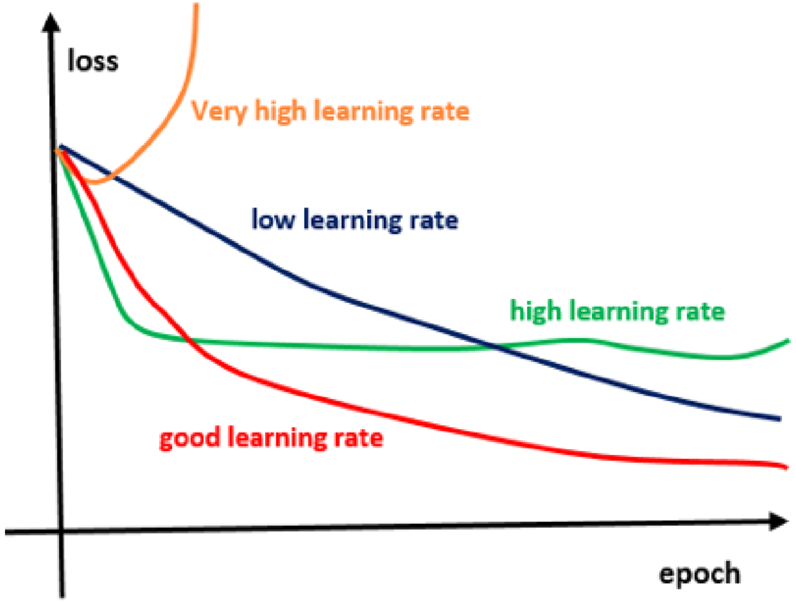
Final Gradient Descent Update Rule:

$$w_j^{k+1} = w_j^k - \left(\alpha \frac{\partial L}{\partial w_j} \right)$$

Effect of Learning Rate:



Loss minimization w.r.to Learning Rate



<u>**Alpha**</u> is a **Hyper-parameter** that <u>**YOU**</u> have to decide (based on your exp. & domain knowledge)… trail & error. HP are to be specified/decided before the start of the iterations start. [0.1, 0.02, 0.04, 0.05, 0.06, 0.08, 0.01]

<u>**Model coefficients/weights (W)**</u> >> **Model Parameters >> these are "learnt" by the optimizer algo from the DATA.** You don't specify this.

Regularization in Machine Learning

$$MSE: \; L = \frac{1}{N}\sum_{i=1}^{N}(\hat{Y}_i - Y_i)^2 \qquad p-Norm\;(L_p) = \left\|w_j\right\|_p = \left(\sum \left|w_j\right|^p\right)^{1/p}$$

$$Ridge: \; L = \{MSE\} + \lambda\left\|w_j\right\|_2^2 = \left\{\frac{1}{N}\sum_{i=1}^{N}(\hat{Y}_i - Y_i)^2\right\} + \lambda(w_1^2 + w_2^2 + \ldots + w_p^2)$$

$$LASSO: \; L = \{MSE\} + \lambda\left\|w_j\right\|_1 = \left\{\frac{1}{N}\sum_{i=1}^{N}(\hat{Y}_i - Y_i)^2\right\} + \lambda\left(\left|w_1\right| + \left|w_2\right| + \ldots + \left|w_p\right|\right)$$

$$ElasticNet: \; L = \{MSE\} + \lambda_1\left\|w_j\right\|_1 + \lambda_2\left\|w_j\right\|_2^2$$

where $j = 1, 2\ldots\ldots p$ number of features

L1 Regularization (also called as LASSO penalisation)

Involves penalising sum of absolute values (1-norms) of regression coefficients

$$LASSO:\ L = \{MSE\} + \lambda \left\| w_j \right\|_1 = \left\{ \frac{1}{N} \sum_{i=1}^{N} (\hat{Y}_i - Y_i)^2 \right\} + \lambda \left(|w_1| + |w_2| + \ldots + |w_p| \right)$$

- Here we are familiar with the First half of the Cost Function.
- By adding all weights to the cost function, which we want to minimize, we're adding further restrictions on these parameters
- **Typically intercepts are not penalised.**
- The lambda parameter in Lasso tunes the strength of the penalty, and should be determined via cross-validation.

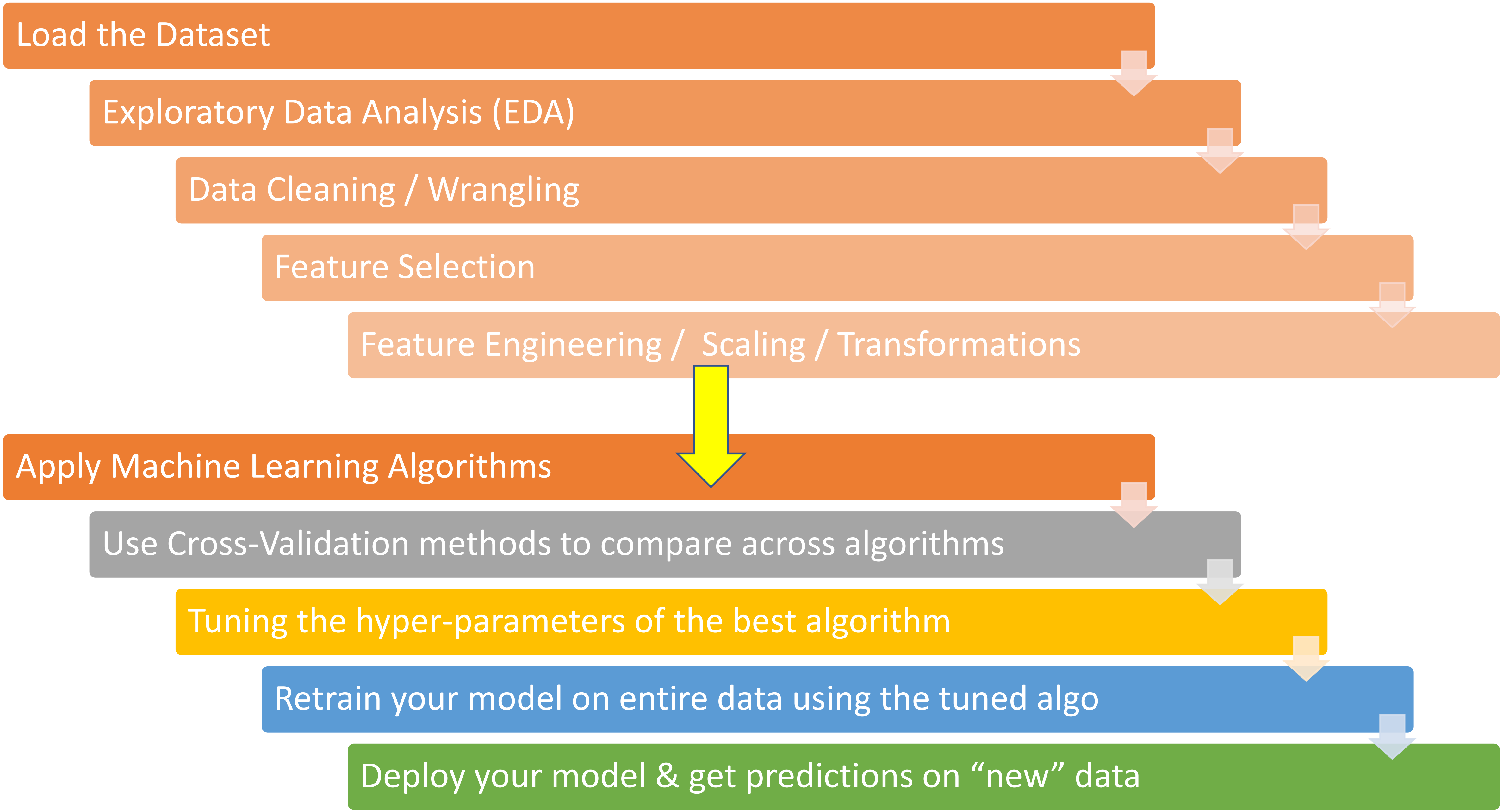
L2 Regularization (also called as Ridge Penalisation)

This proceeds by penalising the sum of squares (2-norms) of the model coefficients

$$Ridge:\ L = \left\{ \frac{1}{N} \sum_{i=1}^{N} (\hat{Y}_i - Y_i)^2 \right\} + \lambda \left\| w_j \right\|_2^2 = \left\{ \frac{1}{N} \sum_{i=1}^{N} (\hat{Y}_i - Y_i)^2 \right\} + \lambda (w_1^2 + w_2^2 \ldots + w_j^2)$$

- The L2 regularization will force the parameters to be relatively small, the bigger the penalization, the smaller (and the more robust to overfitting) the coefficients are.

- Here we are considering every feature, but we are penalizing the coefficients based on how significant the feature is.


Load the Dataset
Exploratory Data Analysis (EDA)
Data Cleaning / Wrangling
Feature Selection
Feature Engineering / Scaling / Transformations
Apply Machine Learning Algorithms
Use Cross-Validation methods to compare across algorithms
Tuning the hyper-parameters of the best algorithm
Retrain your model on entire data using the tuned algo
Deploy your model & get predictions on "new" data