UNIT 1 → Introduction to Machine Learning

**Regression :** Linear regression (Continuous Variables)

**Variables** : Continuous, Categorical

**Classification** : (Categorical Variables)

Machine Learning methods are categorized into

1. **Supervised learning** : Dataset has features & targets (Used for predicting).  
   a. Linear Regression  
   b. Classification
2. **Unsupervised Learning** : The dataset has only features.  
   a. Used for finding groups/patterns/clusters.  
   b. Dimensionality(feature) Reduction.

### **Datasets:**

1. csv files - comma separated values (Easy to manipulate)
2. excel
3. json files

Datasets are publicly available in

1. kaggle
2. UCI repository
3. data.gov.in

Python has Machine Learning Module for different Algorithms like,

1. Linear Regression
2. Classification
3. Dimensionality Reduction
4. Clustering

### **ScKit-Learn(SKLearn)**

#### **Machine Learning Model Building** ::

1. Collect the dataset.
2. Load the dataset.
3. Data pre-processing :: handling missing values, removing duplicates, handling the outliers, rescale the features.
4. Build the model.
5. Evaluate the Model performance.
6. Repeat until satisfactory performance.

* Supervised : Features + Target
* If the target is continuous, the variable is called Regression. If the target is categorical it is called Classification.
* Unsupervised : Features Find some patterns/groups/Clusters.
  + **Clustering** : Data Points are grouped/ clustered.
  + **Dimensionality Reduction** : Each feature is one dimension. If the dataset has too many features, reduce the no. of features.
* Regression : The dataset has one or more Features + Target(What you want to predict).
* Target is a continuous variable - float/int.
* Features are also called as Independent variables Target/Label is also called as Dependent variable ie, it depends on future values.
* Regression is to establish a function between target & features.
* The function is called a regression function or model.
* target = f( f1, f2, f3,....)
* target = b1*f1 + b2*f2 +...
* f1, f2....are feature values available in the dataset.
* Target value is also available in the dataset.
* Our job is to find the value of feature coefficients b1, b2,..

**Machine Learning Modelling involves :**

1. Load the dataset
2. Preprocessing the dataset(Cleaning) & get it ready.
3. Split the dataset into train and test subsets.  
   3.1. Train subset will be used to train the model and test subset for testing the accuracy of the model.
4. Train the model using fit() function on the train subset.
5. Evaluate the model performance using a test subset.

**Performance metrics for Linear Regression :**

1. Mean Sq. Error
2. Absolute Error
3. Root Mean Sq. Error

**Normalization :**

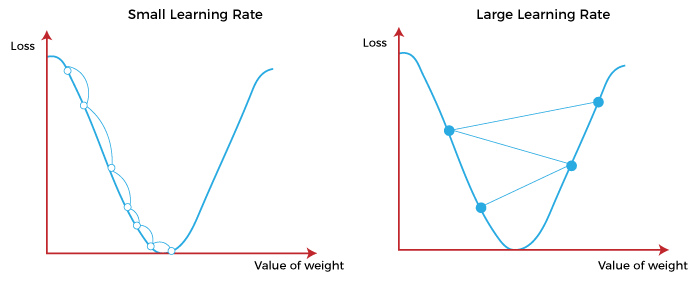
1. Range of values for different features vary greatly, they have to be brought down to similar ranges. Otherwise, the features with large ranges influence the model/function.
2. Dataset may contain missing values that have to be handled.
3. Dataset may also contain exceptions/extreme values called outliers.  
   Hence you have to perform data pre-processing before building a model/function.

**Regression function :**

* + w0, w2 - function parameters.
* x, y are available in the dataset.
* Our job is to find parameter values.
* **Simple Linear Regression** :
  + One feature (x1). Y (Y\_predicted) = w0 + w1\*x1
  + Regression function gives y\_pred and dataset contains y values (true value of y)
  + We have to find the parameters such that the diff. btw y\_pred and y values are minimized.This difference is called an Error.
  + If the dataset has m rows/observations/instances,
* We measure the performance of the model where it gives the minimum loss.

1. Mean Sq. Error (Standard)
2. Absolute Error

* Mean square. Error is our Loss function which is a sq. function. This is a Convex Function(It will have only one minimum).
* Non convex functions will have multiple minimums like sine,cosine graphs.
* **Gradient Descent Algorithm** is to find optimal values of the parameters where loss is minimum.



**Gradient Descent Algorithm :**

* Take the initial value of 'W'.
* Modify W value,
* Repeat Step2 until convergence (till it reaches almost zero).

**Convergence criteria :**

* When the derivative is 0, W does not get updated..ie, you reached the Loss function minimum.
* Repeat N no.of times.
* Alpha is called step/learning rate : typically 0.001.
* For large alpha values may lead to fluctuating around minimum but not reaching it.
* For small values of Alpha require more time for convergence.

**Formulae :**

* Objective is to minimize the L value.
* GD for Simple LR :
  + Take initial values for w1, w0
  + Repeat,
    - → Until Convergence.

**Different Types of Gradient Descent Algorithms :**

1. **Gradient Descent** : Entire dataset for updating parameters.
2. **Mini- batch Gradient Descent** : Batches/subsets of dataset.
3. **Stochastic Gradient Descent** : One observation/row of the dataset.

**Classification :**

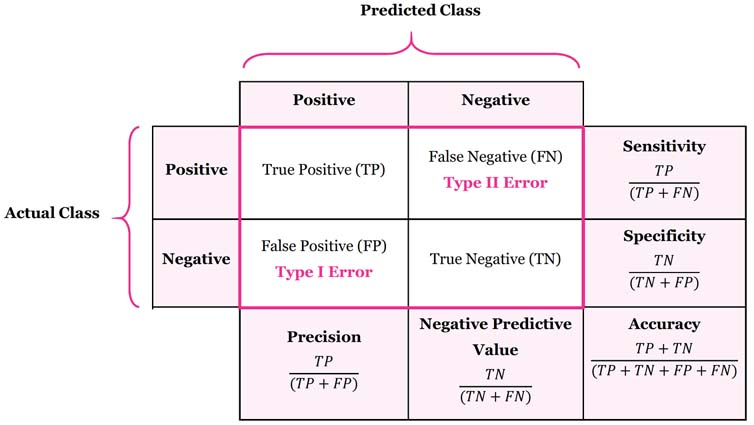
* Each unique value in the target is called class/category.
* If no. of classes is 2 that is **Binary classification**.
  + Eg: Spam/non-Spam, Male/Female
* If no. of classes is more than 2 that is **Multi class Classification.**
  + Eg: 1. emails-spam, company, personal
  + tumors - beginning, stage1, stage2

**Classification Algorithms :**

1. Logistic Regression
2. Decision Tree
3. SVM
4. KNN
5. Naive Bayes
6. Neural Networks

**Metrics used for Evaluation for Performance Evaluation ::**

* **Confusion Matrix :**
  + True Negatives
  + True Positives
  + False Positives
  + False Negatives.

  
  
It is N\*N matrix for N no. of classes.  
The main diagonal values are the ones that are correctly predicted & others are errors.

* **Accuracy score** : How many are correctly predicted out of M no. of observations.
* **Classification report** :
  + **Precision** : True positives : proportion of actual positives out of positives.  
    - 1. Eg : Total positive preds = 25; Actual positives = 20  
          Precision =20/25
  + **Recall** : Proportion of actual positives.
  + **F1 Score :**
* **Imbalanced Dataset** : More observations of one class and very few observations of the other class.
* Most of the medical datasets, products, and fraud cases are imbalanced.
* So, as a pre-processing step , an imbalanced dataset has to be handled.
* You should get observations/generate more observations, remove extra data observations.
* Up sampling and DownSampling.
* Regression problems can be converted to classification problems if we can convert continuous values into 0 or 1.
* This is done using the Sigmoid function on h(x)/ypred.

## **Distance based methods & K Nearest Neighbors ::**

* ML Algorithms require measures for finding out the closeness/similarity/distance between the data points.
* The metrics used to measure the closeness is called **Distance Measure or Similarity Measure**.
* Distance measures for numeric are :

1. Euclidean distance
2. Hamming distance
3. Manhattan distance
4. Minkowski distance

* ML Algorithms like KNN, K Means use distance measures.
  1. **K Nearest Neighbour (KNN) Algorithm** :
* The class of the new datapoint is predicted as the majority class.
* Assume K value(odd number).

| **Merits** | **Demerits** |
| --- | --- |
| 1. Simple & easy to implement algorithms. 2. Lazy algorithm. Find neighbors only when you want to predict the class of the new datapoint. 3. Non parametric algorithm - does not assume anything about data distribution. | 1. Does not work for high dimensional datasets. 2. Does not work for strings, categorical features. |

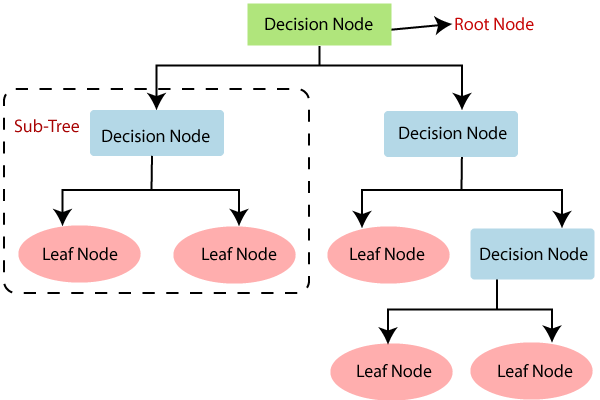
* **Application** :
  + Document similarity (eg - News classification )
  + Pattern recognition
  + Recommendor systems
  + Dimensionality reduction

**Classification Algorithms :**

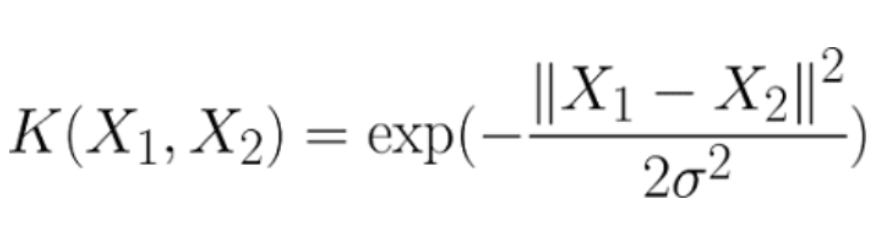
1. Logistic regression
2. KNN
3. Decision tree
4. SVM
5. Naive Bayes

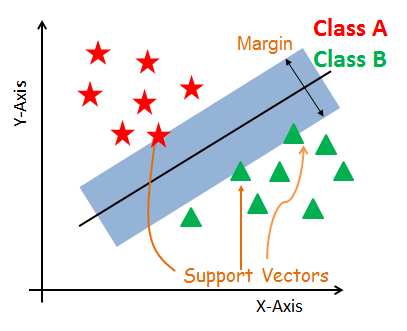
**Decision Tree :**

* Tree contains nodes and edges.
* The tree is used for making the decisions such as -- if an instance or row or example belongs to a particular class.
* Nodes represent the features and Edges are used for traversing.
* Leaf node contains Target.
* Intermediate nodes contain the features. Edge - Branching to the next level node based on the value of the feature present in the node.



* Training phase involves building a decision tree.
* Make Prediction for the new examples by traversing the tree starting from root till you reach the leaf node.
* **Features of a decision tree :**
  1. No need of Normalization or Feature scaling.
  2. No need of transformations ie, no need to convert categorical features into numericals.
  3. Easy to interpret/predict by just traversing the tree.
  4. Easy to visualize.
  5. No assumptions about data distribution.
  6. Not suitable for biased datasets.
  7. Sensitive to noise.

**Support Vector Machines SVM :** 

****

* **Decision boundary** :
  + The line separating the classes.
  + SVM Maximizes the margin from support vectors to decision boundary so as to increase the model performance & reduce the errors.
* Here the data points are **linearly separable** ie, you are able to separate the classes using a line.
* Few data points are **not linearly separable**, i.e., the decision boundary is not linear but circular.
* So to handle such non linearly separable data points, we will transform the data points from non-linear to linearly separable space using **Kernels**.
* **Kernels** :
  + These are the functions that are applied on the data to make it into linearly separable space.
  + It transforms low dimensional input into high dimensional data by generating more features from few features.
  + **Kernel Functions are** :

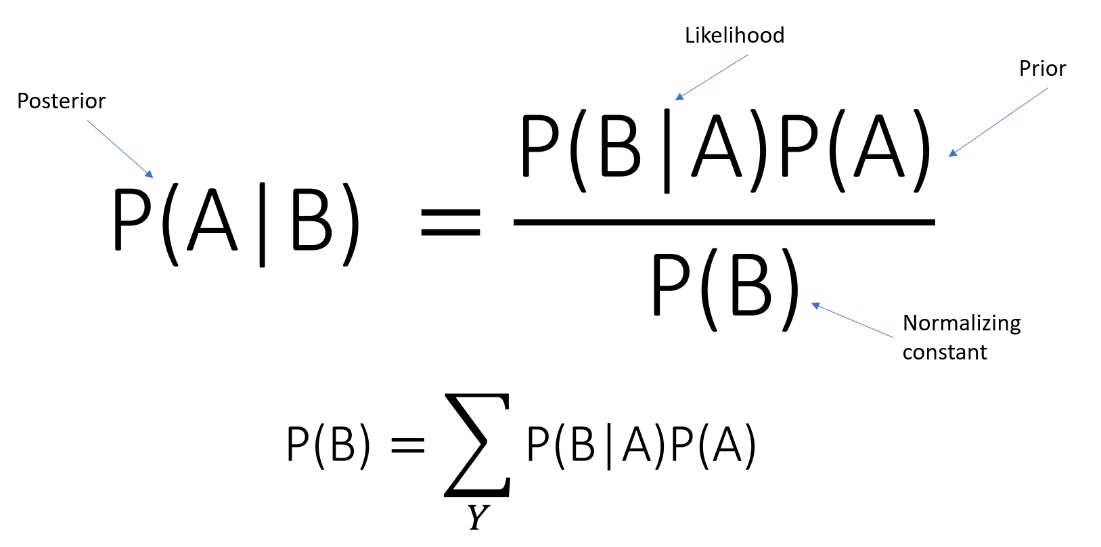
1. **Linear Kernel** -
2. **Polynomial Kernel** -

where d=degree of polynomial

1. **RBF Kernel** - Radial basis function.  
    Sigma value is between 0-1 & typically its 0.1.

| **Advantages** | **Disadvantages** |
| --- | --- |
| 1. SVM gives good accuracy and fast prediction. 2. Less memory. | 1. Not suitable for large datasets. 2. Time for training. 3. Sensitive to the kernel you have chosen. |

**Naive Bayes Algorithm :**

****

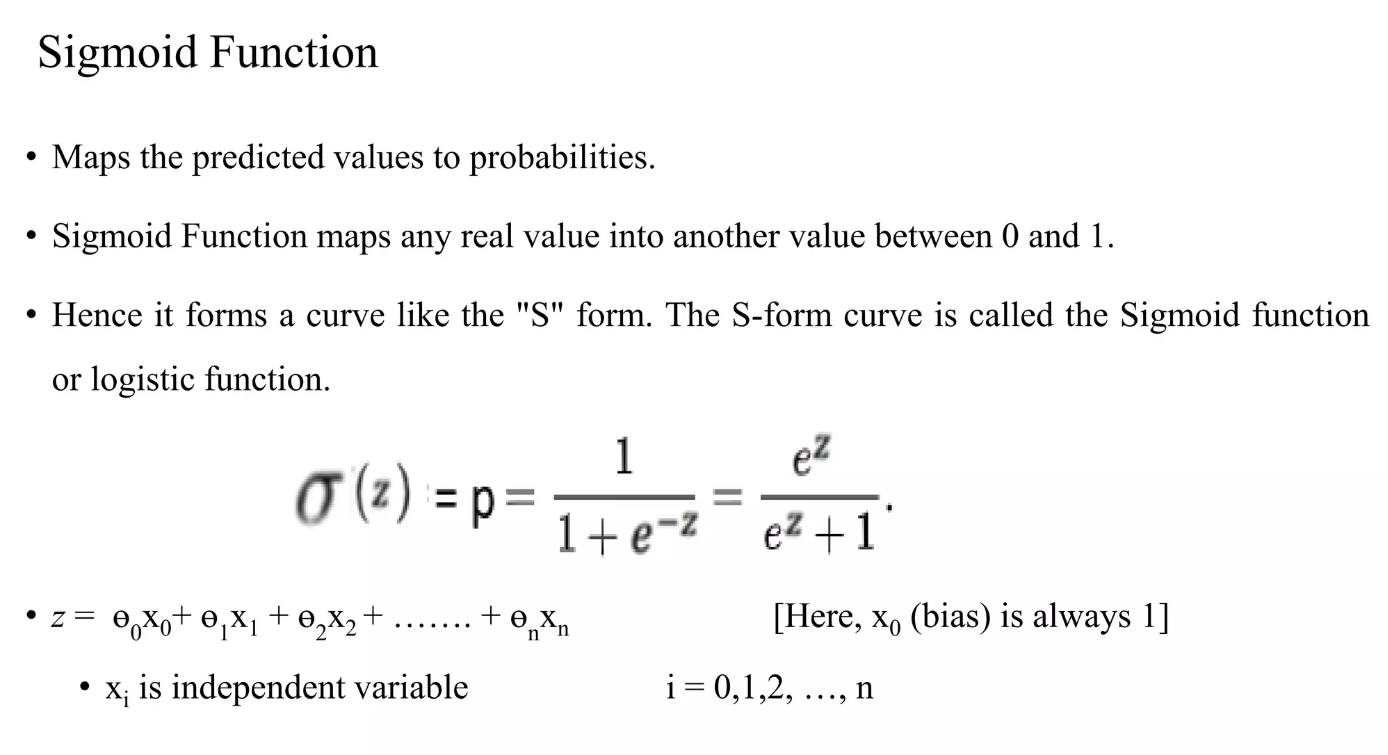
| **Advantages** | **Disadvantages** |
| --- | --- |
| 1. Simple and fast prediction because once the tables are formed it is only calculation. 2. Can also be used for multi class classification. 3. This method is good when features are independent of one another. | 1. In the real world, features are not independent of one another. |

**Logistic Regression :**

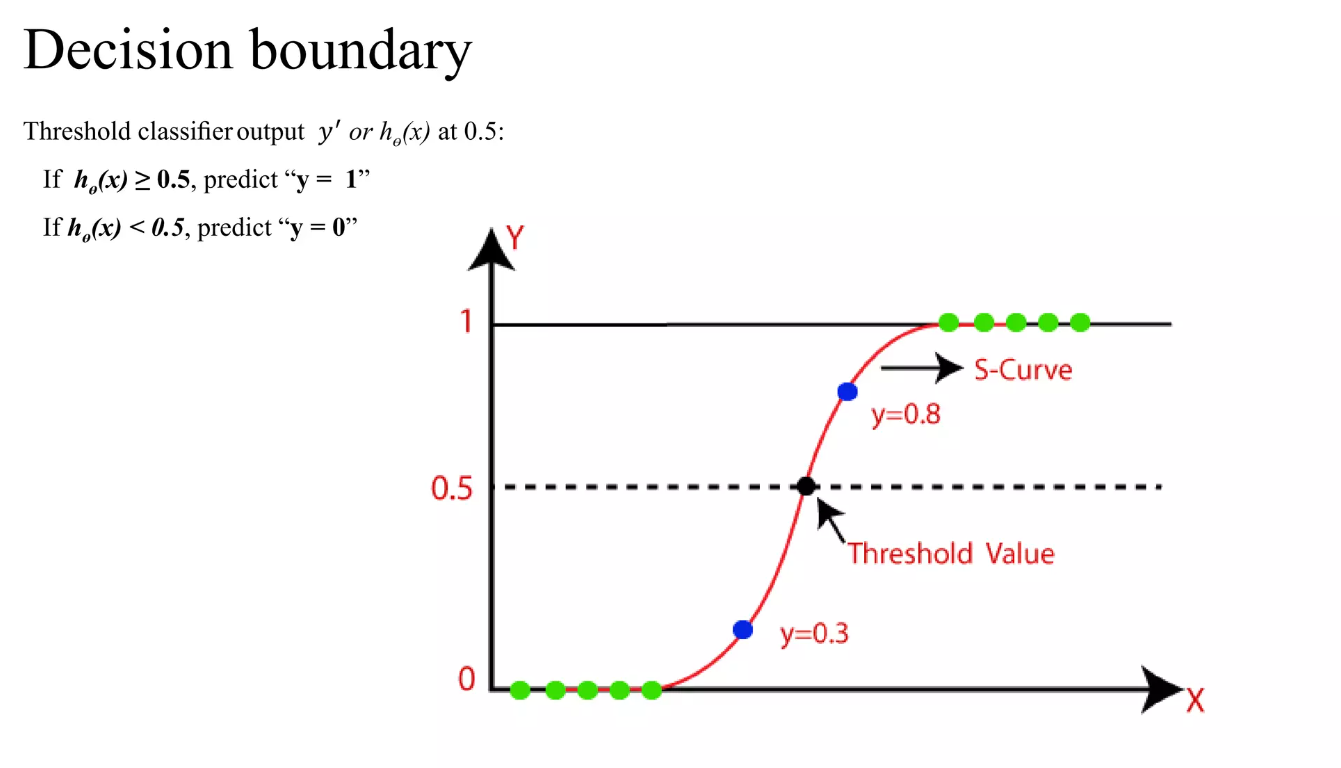
* It is a predictive analysis algorithm and based on the concept of probability.
* Measures the relationship between the dependent variable and the one or more independent variables (features), by estimating probabilities using logistic function.
* It used to assign observations to a discrete set of classes.
* **Examples :**
  + Email spam or not spam,
  + Online transactions Fraud or not Fraud,
  + Tumor Malignant or Benign.
* Logistic regression transforms its output using sigmoid (logistic) function to return a probability value.

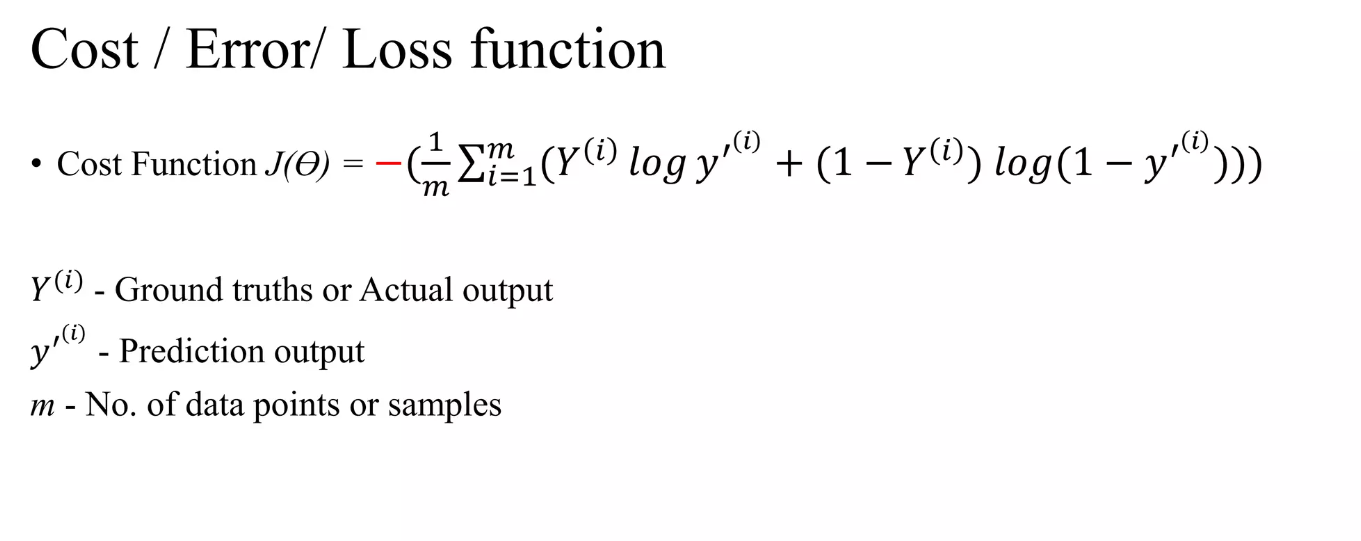
Logistic regression limits the hypothesis function between 0 to 1 i.e. h(x) / y' € [0,1]

* **Types of classification using Logistic Regression :**
  + Binary classification(e.g. Tumor Malignant or Benign)
  + Multi-Class classification(eg. Cats, dogs or Sheep's)
* **Assumptions for Logistic Regression:**
  + The dependent variable must be categorical in nature.
  + The independent variable should not have multi-collinearity i.e. independent variables must be independent of each other.
* **Sigmoid Function :** 
  + Maps the predicted values to probabilities.
  + Sigmoid Function maps any real value into another value between 0 and 1.
  + Hence it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or logistic function.

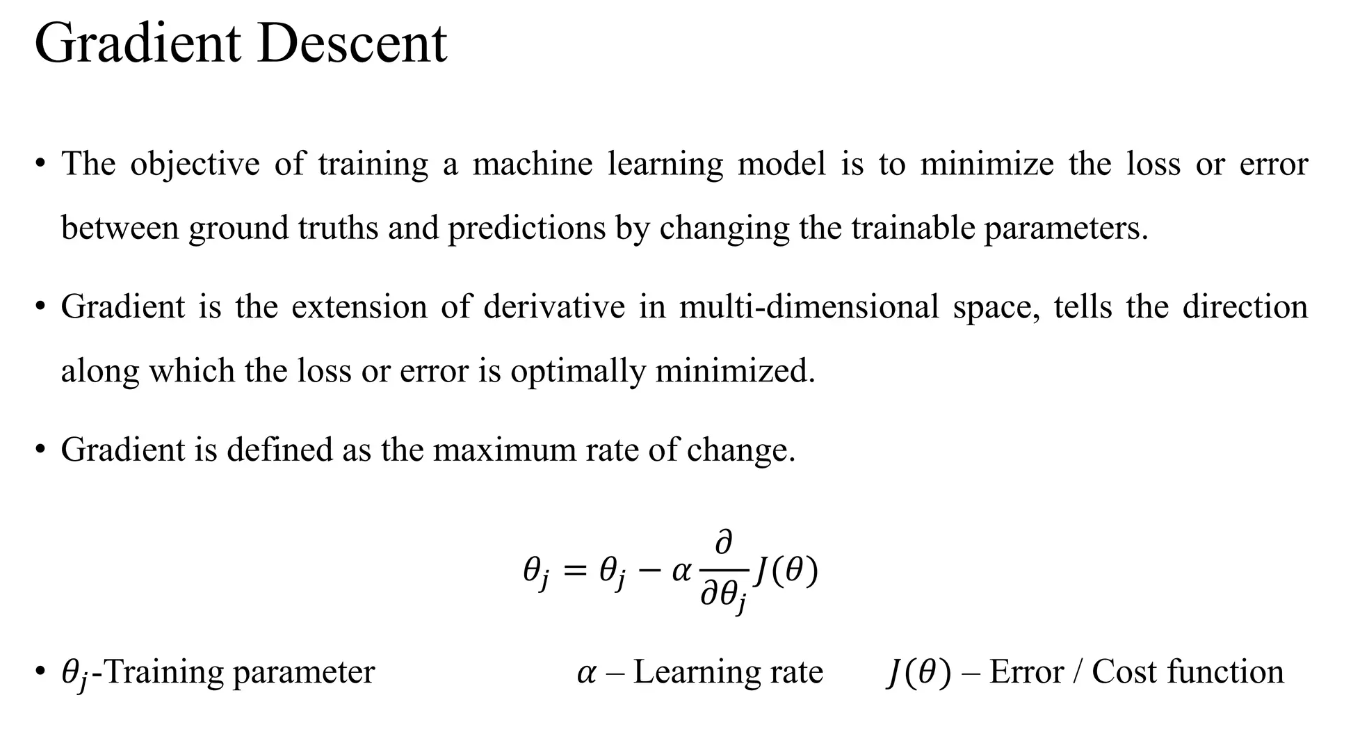


* **Decision boundary :** 
  + Threshold classifier output or at 0.5:
  + If > 0.5, predict "y = 1"
  + If < 0.5, predict "y = 0"





* **Gradient Descent :** 
  + The objective of training a machine learning model is to minimize the loss or error between ground truths and predictions by changing the trainable parameters.
  + Gradient is the extension of derivatives in multi-dimensional space, which tells the direction along which the loss or error is optimally minimized.
  + Gradient is defined as the maximum rate of change.



**Practical Steps in Logistic Regression :**

* To implement the Logistic Regression using Python,
  + Data Preprocessing step.
  + Fitting Logistic Regression to the Training set.
  + Predicting the test result.
  + Test accuracy of the result(Creation of Confusion matrix).
  + Visualizing the test set result.