Artificial Intelligence

Unit-6 (Machine Learning)

Artificial Intelligence 01CE0702

**Marwadi University**

Department of Computer Engineering

Shilpa Singhal

- Machine Learning
- Types of Machine Learning
- Supervised Learning
- Unsupervised Learning
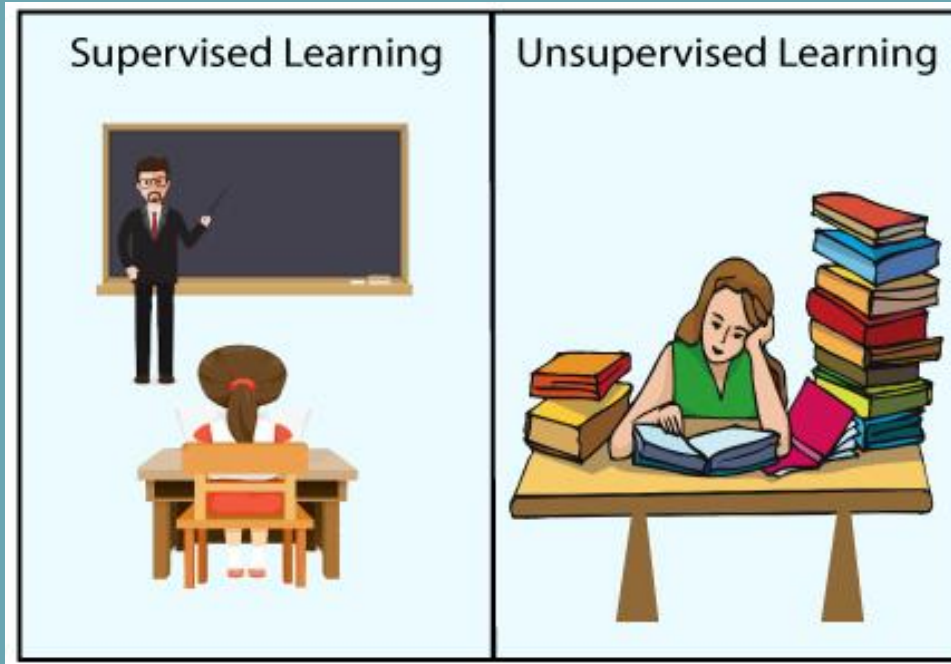- Types of Supervised Learning
- Types of Unsupervised Learning

Machine learning is a subset of AI, which enables the machine to automatically learn from data, improve performance from past experiences, and make predictions.

- Machine learning contains a set of algorithms that work on a huge amount of data.

- Data is fed to these algorithms to train them, and on the basis of training, they build the model & perform a specific task.
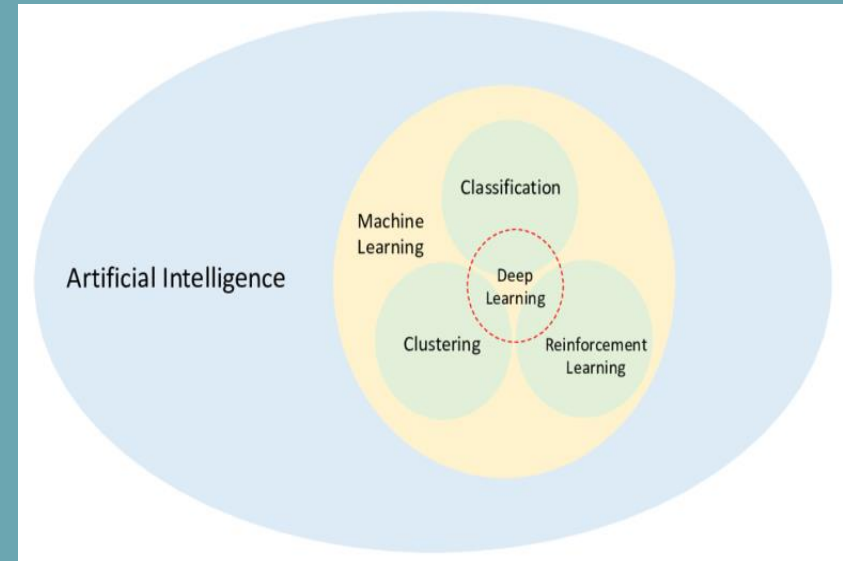
Types of Machine Learning

- Supervised and Unsupervised learning are the two techniques of machine learning. But both the techniques are used in different scenarios and with different datasets.

- Supervised learning is a machine learning method in which models are trained using labeled data.
- In supervised learning, models need to find the mapping function to map the input variable (X) with the output variable (Y).
  - $Y = f(X)$
- Supervised learning needs supervision to train the model, which is similar to as a student learns things in the presence of a teacher. Supervised learning can be used for two types of problems:
  - Classification
  - Regression.

Labeled Data

- Supervised learning is a process of providing input data as well as correct output data to the machine learning model.
- The aim of a supervised learning algorithm is to **find a mapping function to map the input variable(x) with the output variable(y)**.

	Outlook	Temperature	Humidity	Windy	Play Golf
0	Rainy	Hot	High	False	No
1	Rainy	Hot	High	True	No
2	Overcast	Hot	High	False	Yes
3	Sunny	Mild	High	False	Yes
4	Sunny	Cool	Normal	False	Yes
5	Sunny	Cool	Normal	True	No
6	Overcast	Cool	Normal	True	Yes
7	Rainy	Mild	High	False	No
8	Rainy	Cool	Normal	False	Yes
9	Sunny	Mild	Normal	False	Yes
10	Rainy	Mild	Normal	True	Yes
11	Overcast	Mild	High	True	Yes
12	Overcast	Hot	Normal	False	Yes
13	Sunny	Mild	High	True	No

Example:

- ○ Suppose we have an image of different types of fruits.
- ○ The task of our supervised learning model is to identify the fruits and classify them accordingly.
- ○ So to identify the image in supervised learning, we will give the input data as well as output for that, which means we will train the model by the shape, size, color, and taste of each fruit.
- ○ Once the training is completed, we will test the model by giving the new set of fruit.
- ○ The model will identify the fruit and predict the output using a suitable algorithm.

Example:

- Suppose we have a dataset of different types of shapes which includes square, rectangle, triangle, and Polygon. Now the first step is that we need to train the model for each shape.
  - If the given shape has four sides, and all the sides are equal, then it will be labelled as a Square
  - If the given shape has three sides, then it will be labelled as a triangle.
  - If the given shape has six equal sides then it will be labelled as hexagon.
- Now, after training, we test our model using the test set, and the task of the model is to identify the shape.
- The machine is already trained on all types of shapes, and when it finds a new shape, it classifies the shape on the bases of a number of sides, and predicts the output.

Supervised Learning

- In supervised learning, models are trained using labelled dataset, where the model learns about each type of data.
- Once the training process is completed, the model is tested on the basis of test data (a subset of the training set), and then it predicts the output.
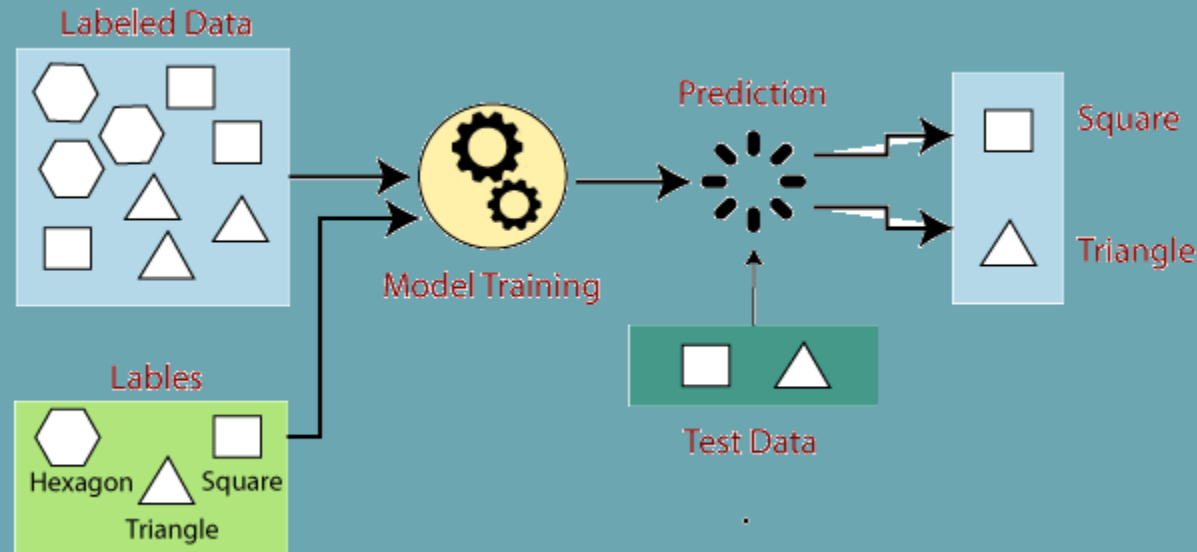
Steps of Supervised Learning

- First Determine the type of training dataset
- Collect/Gather the labelled training data.
- Split the training dataset into training dataset, test dataset, and validation dataset.
- Determine the input features of the training dataset, which should have enough knowledge so that the model can accurately predict the output.
- Determine the suitable algorithm for the model, such as support vector machine, decision tree, etc.
- Execute the algorithm on the training dataset. Sometimes we need validation sets as the control parameters, which are the subset of training datasets.
- Evaluate the accuracy of the model by providing the test set. If the model predicts the correct output, which means our model is accurate.

Unsupervised Learning

- Unsupervised learning is another machine learning method in which patterns inferred from the unlabeled input data.
- The goal of unsupervised learning is to find the structure and patterns from the input data.
- Unsupervised learning does not need any supervision. Instead, it finds patterns from the data by its own.
- It can be compared to learning which takes place in the human brain while learning new things.
- It can be defined as:

  Unsupervised learning is a type of machine learning in which models are trained using unlabeled dataset and are allowed to act on that data without any supervision.
- Unsupervised learning can be used for two types of problems:
 - Clustering
  - Association.

Example

- Example: Suppose the unsupervised learning algorithm is given an input dataset containing images of different types of cats and dogs.

- The algorithm is never trained upon the given dataset, which means it does not have any idea about the features of the dataset.

- The task of the unsupervised learning algorithm is to identify the image features on their own.

- Unsupervised learning algorithm will perform this task by clustering the image dataset into the groups according to similarities between images.

Unsupervised Learning

- Working of unsupervised learning can be understood by the below diagram:
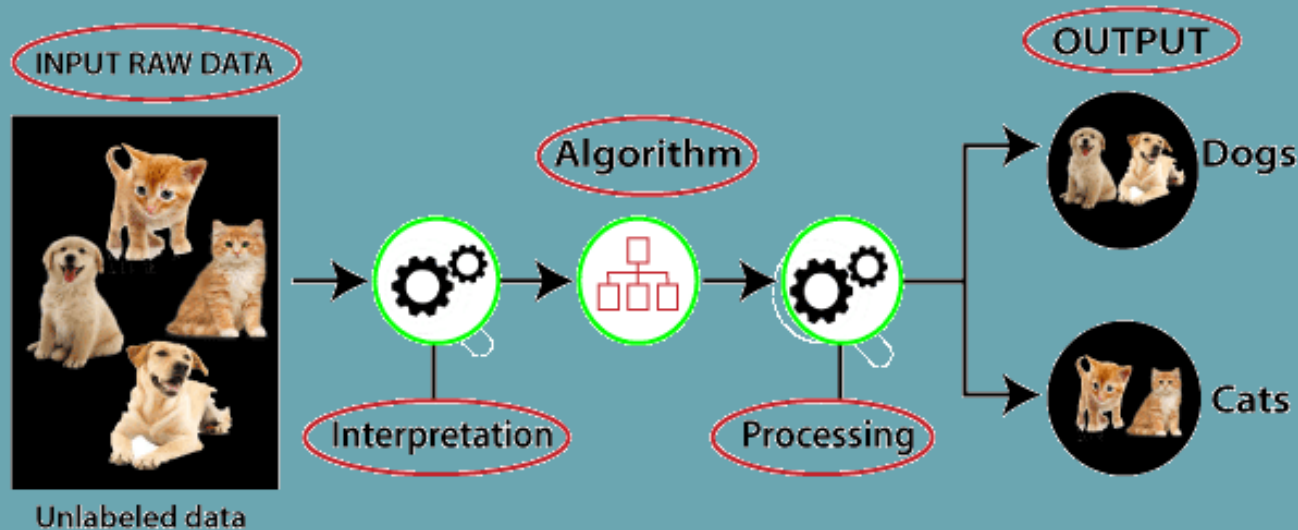
Importance of Unsupervised Learning

Below are some main reasons which describe the importance of Unsupervised Learning:

- Unsupervised learning is helpful for finding useful insights from the data.
- Unsupervised learning is much similar as a human learns to think by their own experiences, which makes it closer to the real AI.
- Unsupervised learning works on unlabeled and uncategorized data which make unsupervised learning more important.
- In real-world, we do not always have input data with the corresponding output so to solve such cases, we need unsupervised learning.

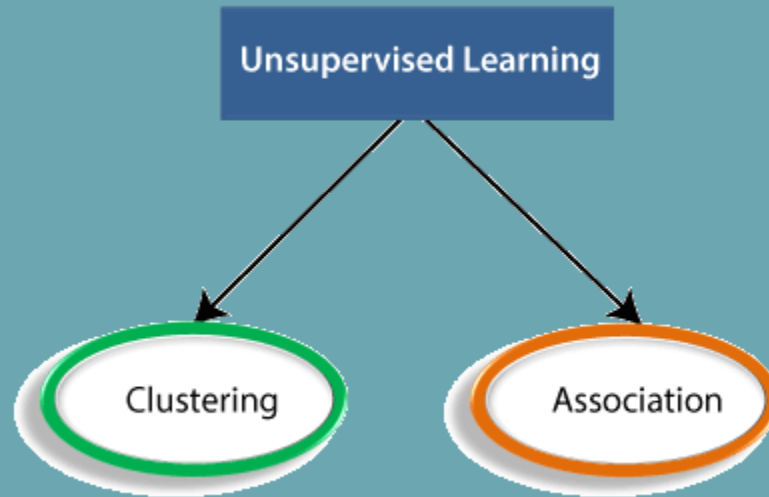| Supervised Learning	Unsupervised Learning
• Supervised learning algorithms are trained using labeled data.	• Unsupervised learning algorithms are trained using unlabeled data.
• Supervised learning model takes direct feedback to check if it is predicting correct output or not.	• Unsupervised learning model does not take any feedback.
• Supervised learning model predicts the output.	• Unsupervised learning model finds the hidden patterns in data.
• In supervised learning, input data is provided to the model along with the output.	• In unsupervised learning, only input data is provided to the model.
• The goal of supervised learning is to train the model so that it can predict the output when it is given new data.	• The goal of unsupervised learning is to find the hidden patterns and useful insights from the unknown dataset.
• Supervised learning needs supervision to train the model.	• Unsupervised learning does not need any supervision to train the model.
• Supervised learning can be categorized in **Classification** and **Regression** problems.	• Unsupervised Learning can be classified in **Clustering** and **Associations** problems.
• Supervised learning can be used for those cases where we know the input as well as corresponding outputs.	• Unsupervised learning can be used for those cases where we have only input data and no corresponding output data.
• Supervised learning model produces an accurate result.	• Unsupervised learning model may give less accurate result as compared to supervised learning.
• Supervised learning is not close to true Artificial intelligence as in this, we first train the model for each data, and then only it can predict the correct output.	• Unsupervised learning is more close to the true Artificial Intelligence as it learns similarly as a child learns daily routine things by his experiences.
• It includes various algorithms such as Linear Regression, Logistic Regression, Support Vector Machine, Multi-class Classification, Decision tree, Bayesian Logic, etc.	• It includes various algorithms such as Clustering, KNN, and Apriori algorithm

- Types of supervised Machine learning Algorithms
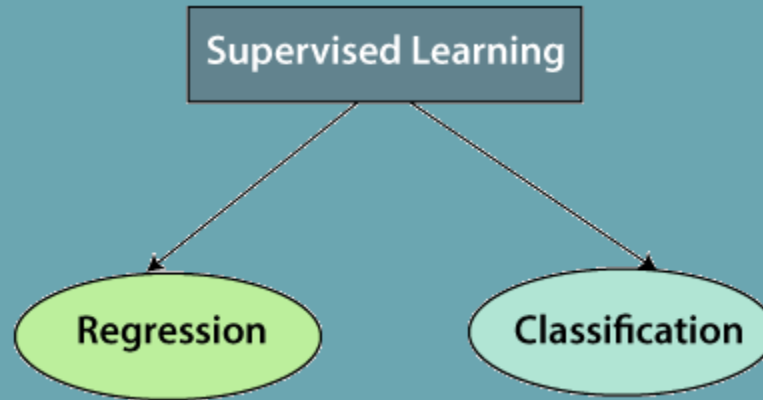
- Regression
  - Regression algorithms are used if there is a relationship between the input variable and the output variable. It is used for the prediction of continuous variables, such as Weather forecasting, Market Trends, etc.
- Below are some popular Regression algorithms which come under supervised learning:
  - Linear Regression
  - Regression Trees
  - Non-Linear Regression
  - Bayesian Linear Regression
  - Polynomial Regression

- Classification algorithms are used when the output variable is categorical, which means there are two classes such as Yes-No, Male-Female, True-false, etc.

- List of few Classification Algorithms:
  - Random Forest
  - Decision Trees
  - Logistic Regression
  - Support vector Machines

- The unsupervised learning algorithm can be further categorized into two types of problems:

Unsupervised Learning

- Clustering:
  - Clustering is a method of grouping the objects into clusters such that objects with most similarities remains into a group and has less or no similarities with the objects of another group.
  - Cluster analysis finds the commonalities between the data objects and categorizes them as per the presence and absence of those commonalities.
- Association:
  - An association rule is an unsupervised learning method which is used for finding the relationships between variables in the large database.
  - It determines the set of items that occurs together in the dataset. Association rule makes marketing strategy more effective.
  - Such as people who buy X item (suppose a bread) are also tend to purchase Y (Butter/Jam) item. A typical example of Association rule is Market Basket Analysis.

- K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions).
- KNN has been used in statistical estimation and pattern recognition already in the beginning of 1970's as a non-parametric technique.
- Algorithm
  - A case is classified by a majority vote of its neighbors, with the case being assigned to the class most common amongst its K nearest neighbors measured by a distance function.
  - If K = 1, then the case is simply assigned to the class of its nearest neighbor.

K Nearest Neighbors(KNN) - Classification

- It should also be noted that all three distance measures are only valid for continuous variables.
- In the instance of categorical variables the Hamming distance must be used.

Distance functions

Euclidean
$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

Manhattan
$$\sum_{i=1}^{k}|x_i - y_i|$$

Minkowski
$$\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$$

Hamming Distance

$$D_H = \sum_{i=1}^{k}|x_i - y_i|$$

$$x = y \Rightarrow D = 0$$
$$x \neq y \Rightarrow D = 1$$

X	Y	Distance
Male	Male	0
Male	Female	1

K Nearest Neighbors(KNN) - Classification

- Choosing the optimal value for K is best done by first inspecting the data. In general, a large K value is more precise as it reduces the overall noise but there is no guarantee.
- Cross-validation is another way to retrospectively determine a good K value by using an independent dataset to validate the K value.
- Historically, the optimal K for most datasets has been between 3-10.
- That produces much better results than 1NN.

K Nearest Neighbors(KNN) - Classification

- We can now use the training set to classify an unknown case (Age=48 and Loan=$142,000) using Euclidean distance.
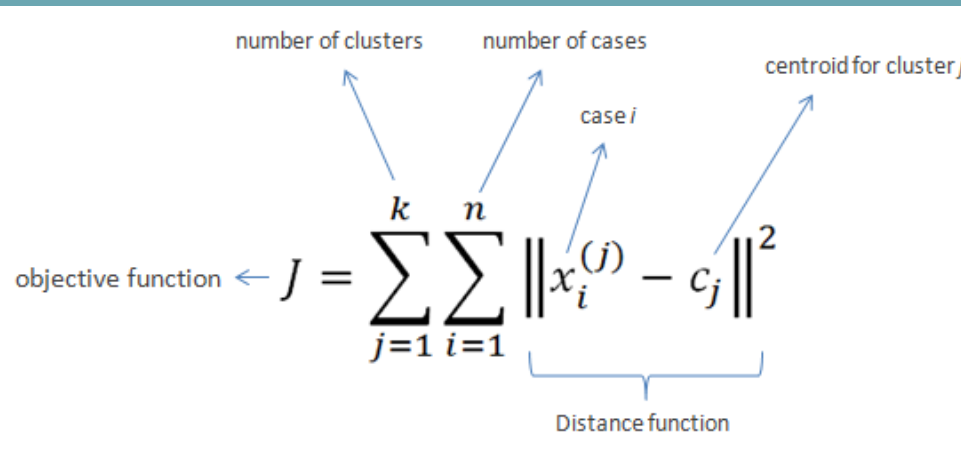- If K=1 then the nearest neighbor is the last case in the training set with Default=Y.

Age	Loan	Default	Distance	
25	$40,000	N	102000	
35	$60,000	N	82000	
45	$80,000	N	62000	
20	$20,000	N	122000	
35	$120,000	N	22000	2
52	$18,000	N	124000	
23	$95,000	Y	47000	
40	$62,000	Y	80000	
60	$100,000	Y	42000	3
48	$220,000	Y	78000	
33	$150,000	Y	8000	1
48	**$142,000**	**?**		

Euclidean Distance

$$D = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

K Means - Clustering

- K-Means clustering intends to partition n objects into k clusters in which each object belongs to the cluster with the nearest mean.
- This method produces exactly k different clusters of greatest possible distinction. The best number of clusters k leading to the greatest separation (distance) is not known as a priori and must be computed from the data.
- The objective of K-Means clustering is to minimize total intra-cluster variance, or, the squared error function:

number of clusters   number of cases   centroid for cluster j

case i

objective function $\leftarrow$ $$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

Distance function

K Means - Clustering

Algorithm

1. Clusters the data into k groups where k is predefined.
2. Select k points at random as cluster centers.
3. Assign objects to their closest cluster center according to the *Euclidean distance* function.
4. Calculate the centroid or mean of all objects in each cluster.
5. Repeat steps 2, 3 and 4 until the same points are assigned to each cluster in consecutive rounds.
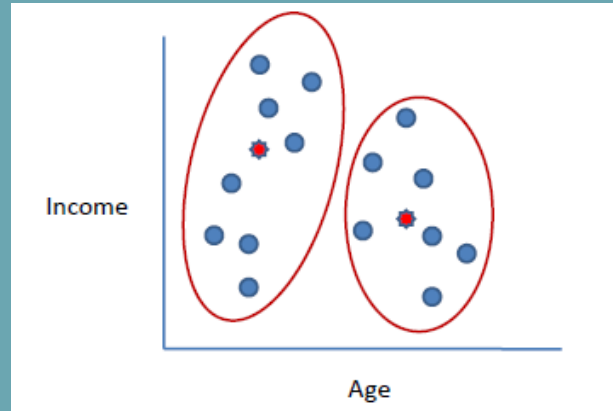
K Means - Clustering

Suppose we want to group the visitors to a website using just their age (one-dimensional space) as follows:

$$n = 19$$

15,15,16,19,19,20,20,21,22,28,35,40,41,42,43,44,60,61,65

Initial clusters (random centroid or average):

$$k = 2$$

$$c_1 = 16$$
$$c_2 = 22$$

$$Distance\ 1 = |x_i - c_1|$$

$$Distance\ 2 = |x_i - c_2|$$

K Means - Clustering

Iteration 1: $c_1 = 15.33$ $c_2 = 36.25$

x_i	c_1	c_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	16	22	1	7	1	
15	16	22	1	7	1	**15.33**
16	16	22	0	6	1	
19	16	22	9	3	2	
19	16	22	9	3	2	
20	16	22	16	2	2	
20	16	22	16	2	2	
21	16	22	25	1	2	
22	16	22	36	0	2	
28	16	22	12	6	2	
35	16	22	19	13	2	
40	16	22	24	18	2	**36.25**
41	16	22	25	19	2	
42	16	22	26	20	2	
43	16	22	27	21	2	
44	16	22	28	22	2	
60	16	22	44	38	2	
61	16	22	45	39	2	
65	16	22	49	43	2	

K Means - Clustering

Iteration 2: $c_1 = 18.56$ $c_2 = 45.90$

x_i	c_1	c_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	15.33	36.25	0.33	21.25	1	
15	15.33	36.25	0.33	21.25	1	
16	15.33	36.25	0.67	20.25	1	
19	15.33	36.25	3.67	17.25	1	
19	15.33	36.25	3.67	17.25	1	**18.56**
20	15.33	36.25	4.67	16.25	1	
20	15.33	36.25	4.67	16.25	1	
21	15.33	36.25	5.67	15.25	1	
22	15.33	36.25	6.67	14.25	1	
28	15.33	36.25	12.67	8.25	2	
35	15.33	36.25	19.67	1.25	2	
40	15.33	36.25	24.67	3.75	2	
41	15.33	36.25	25.67	4.75	2	
42	15.33	36.25	26.67	5.75	2	
43	15.33	36.25	27.67	6.75	2	**45.9**
44	15.33	36.25	28.67	7.75	2	
60	15.33	36.25	44.67	23.75	2	
61	15.33	36.25	45.67	24.75	2	
65	15.33	36.25	49.67	28.75	2	

K Means - Clustering

Iteration 3: $c_1 = 19.50$ $c_2 = 47.89$

x_i	c_1	c_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	18.56	45.9	3.56	30.9	1	
15	18.56	45.9	3.56	30.9	1	
16	18.56	45.9	2.56	29.9	1	
19	18.56	45.9	0.44	26.9	1	
19	18.56	45.9	0.44	26.9	1	**19.50**
20	18.56	45.9	1.44	25.9	1	
20	18.56	45.9	1.44	25.9	1	
21	18.56	45.9	2.44	24.9	1	
22	18.56	45.9	3.44	23.9	1	
28	18.56	45.9	9.44	17.9	1	
35	18.56	45.9	16.44	10.9	2	
40	18.56	45.9	21.44	5.9	2	
41	18.56	45.9	22.44	4.9	2	
42	18.56	45.9	23.44	3.9	2	
43	18.56	45.9	24.44	2.9	2	**47.89**
44	18.56	45.9	25.44	1.9	2	
60	18.56	45.9	41.44	14.1	2	
61	18.56	45.9	42.44	15.1	2	
65	18.56	45.9	46.44	19.1	2	

Biological Neuron

- A neuron or nerve cell is an electrically excitable cell that communicates with other cells via specialized connections called synapses.
- A typical neuron consists of a cell body (soma), dendrites, and a single axon. The soma is usually compact. The axon and dendrites are filaments that extrude from it.
- Dendrites typically branch profusely and extend a few hundred micrometers from the soma.

Simple Neural Network
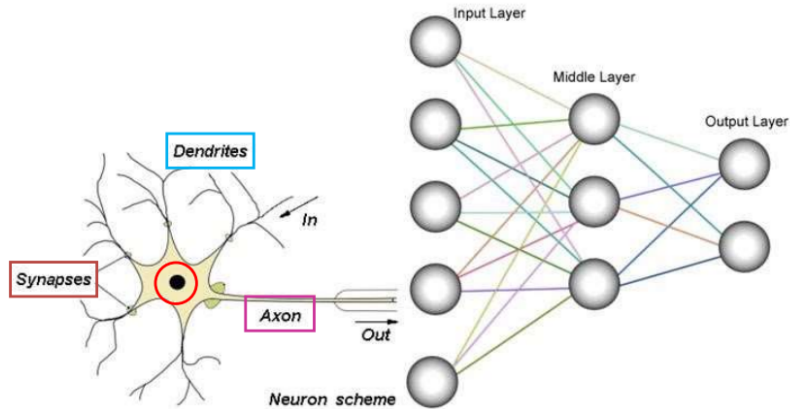
- To understand how an artificial neuron works, we should know how the biological neuron works.

- Dendrites : These receive information or signals from other neurons that get connected to it.

- Cell Body : Information processing happens in a cell body. These take in all the information coming from the different dendrites and process that information.

- Axon : It sends the output signal to another neuron for the flow of information. Here, each of the flanges connects to the dendrite or the hairs on the next one.

- Synapses are elementary structural and functional units that mediate the interactions between neurons.

Simple Neural Network
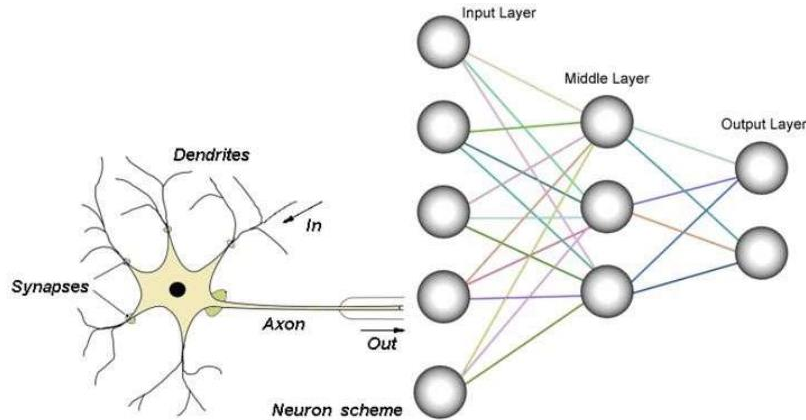
- The network starts with an input layer that receives input in the form of data.

- The lines connected to the hidden layers are called weights, and they add up on the hidden layers.

- Each dot in the hidden layer processes the inputs, and it puts an output into the next hidden layer and lastly, into the output layer.

- A neural network is a system of hardware or software patterned after the operation of neurons in the human brain.

- Neural networks, also called artificial neural networks, are ways of achieving deep learning.

Artificial Neural Network

- Artificial neural networks (ANNs), usually simply called neural networks (NNs), are computing systems inspired by the biological neural networks that constitute animal brains.
- An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain.
- Each connection, like the synapses in a biological brain, can transmit a signal to other neurons.
- An artificial neuron receives a signal then processes it and can signal neurons connected to it.
- The "signal" at a connection is a real number, and the output of each neuron is computed by some non-linear function of the sum of its inputs.
- The connections are called edges.

Artificial Neural Network

- Neurons and edges typically have a weight that adjusts as learning proceeds.
- The weight increases or decreases the strength of the signal at a connection.
- Neurons may have a threshold such that a signal is sent only if the aggregate signal crosses that threshold.
- Typically, neurons are aggregated into layers. Different layers may perform different transformations on their inputs.
- Signals travel from the first layer (the input layer), to the last layer (the output layer), possibly after traversing the layers multiple times.

Artificial Neural Network
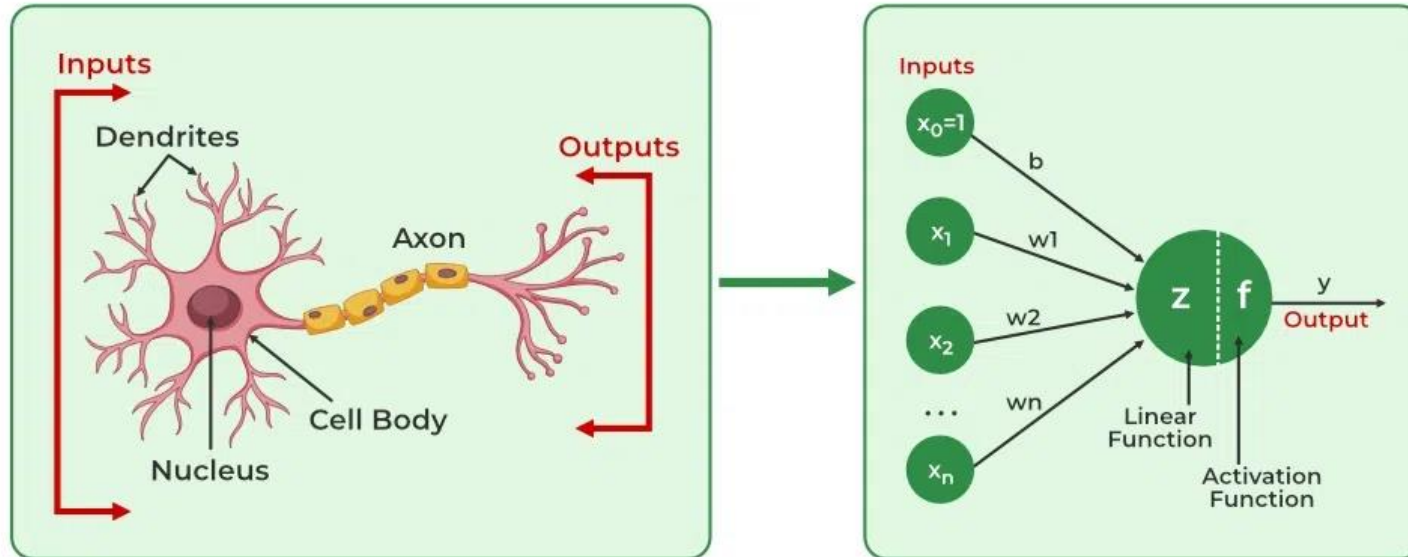
Transfer(Activation) Function

- The transfer function translates the input signals to output signals.
- It is similar in behavior to the biological neuron which transmits the signal only when the total input signal meets the firing threshold.
- Some of transfer functions are commonly used

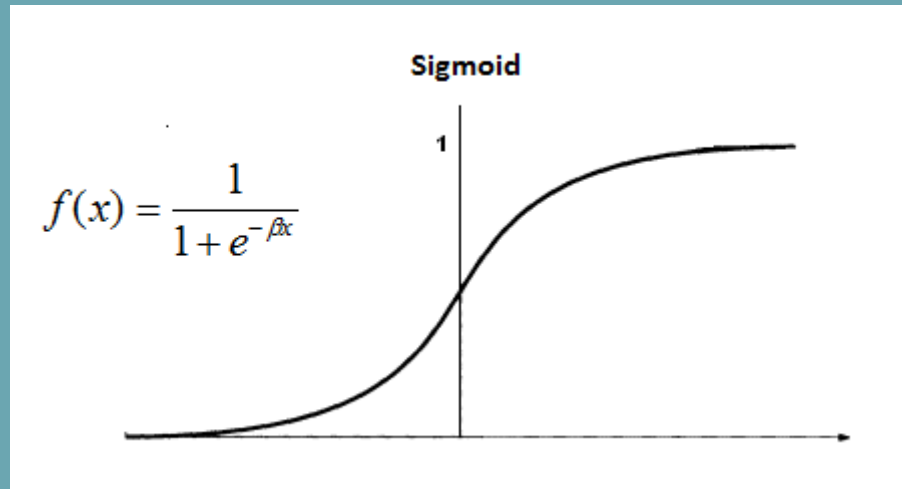   Unit step (threshold), sigmoid, piecewise linear, and Gaussian.
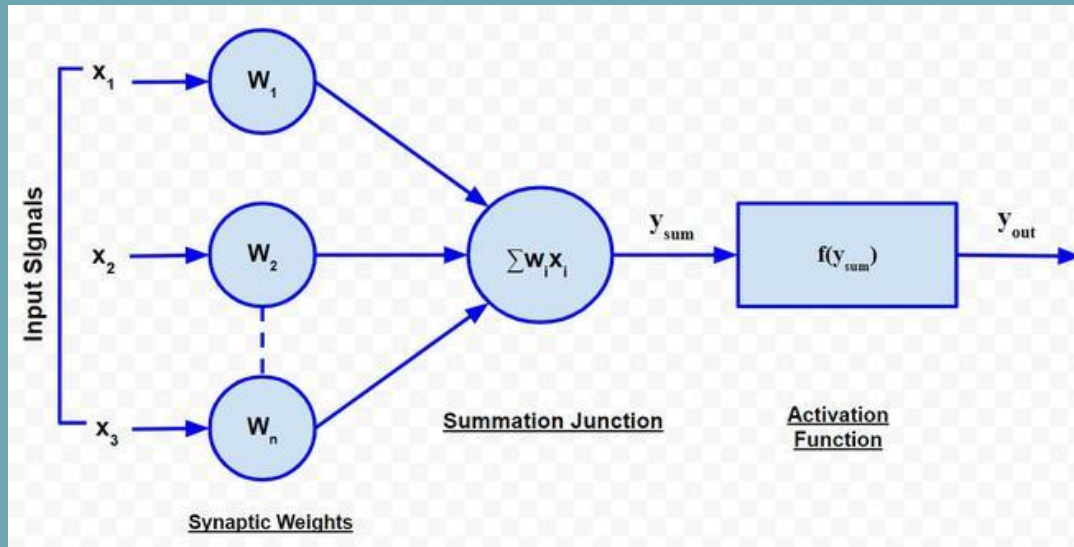
Threshold/step Function

- It is a commonly used activation function. As depicted in the diagram, **it gives 1** as output of the **input is either 0 or positive**. If the input is negative, it gives 0 as output. Expressing it mathematically,
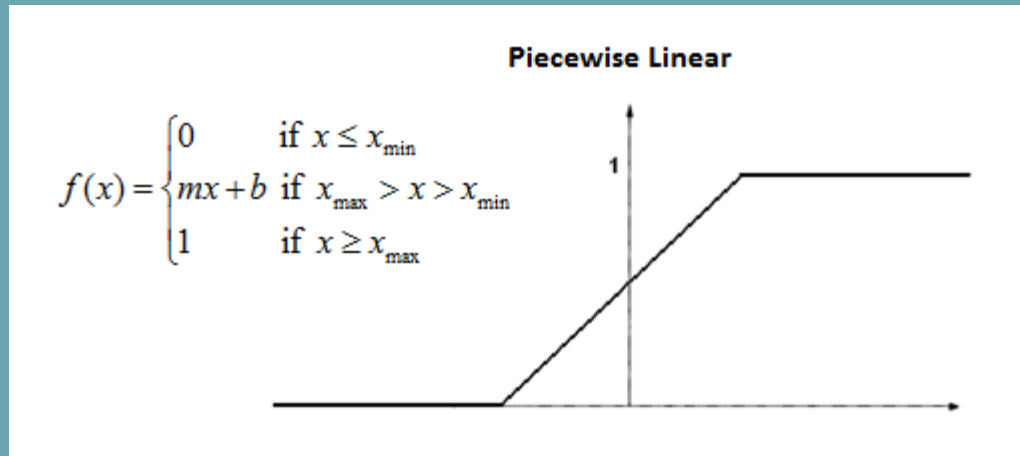
Unit step (threshold)

$$f(x) = \begin{cases} 0 \text{ if } 0 > x \\ 1 \text{ if } x \geq 0 \end{cases}$$
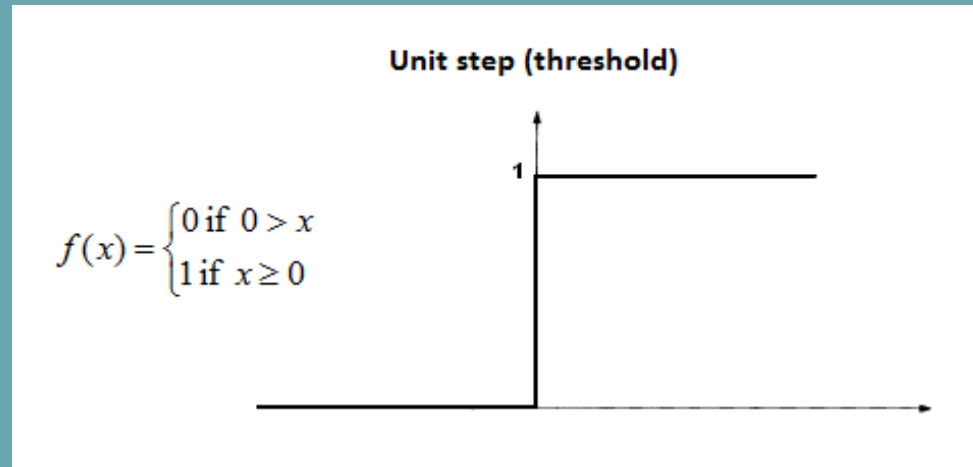
Sigmoid Function

The sigmoid function consists of 2 functions, **logistic and tangential**. The values of logistic function range from 0 and 1 and -1 to +1 for tangential function.

$$f(x) = \frac{1}{1 + e^{-\beta x}}$$

Sigmoid

The output is proportional to the total weighted output.

Piecewise Linear

$$f(x) = \begin{cases} 0 & \text{if } x \le x_{min} \\ mx + b & \text{if } x_{max} > x > x_{min} \\ 1 & \text{if } x \ge x_{max} \end{cases}$$

Gaussian Function

○ Gaussian functions are bell-shaped curves that are continuous.

○ The node output (high/low) is interpreted in terms of class membership (1/0), depending on how close the net input is to a chosen value of average.

Gaussian

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$
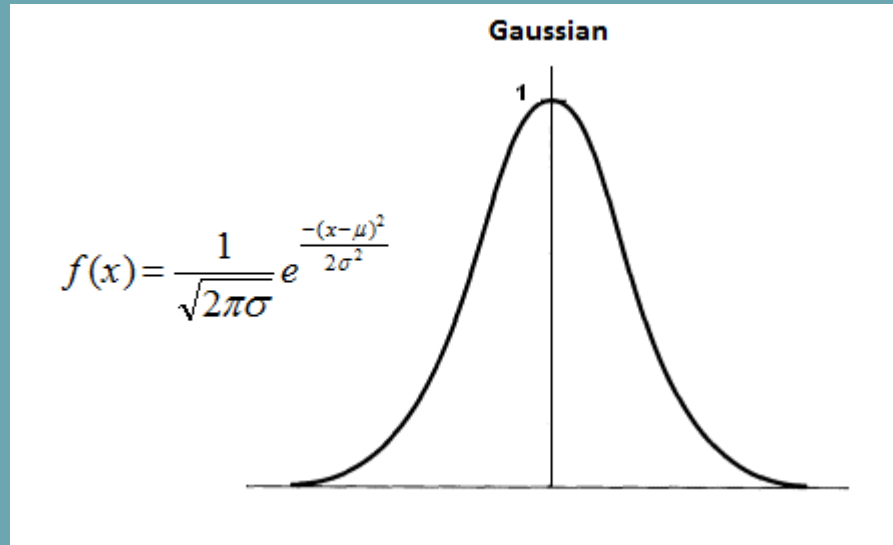
Linear Function

Like a linear regression, a linear activation function transforms the weighted sum inputs of the neuron to an output using a linear function.
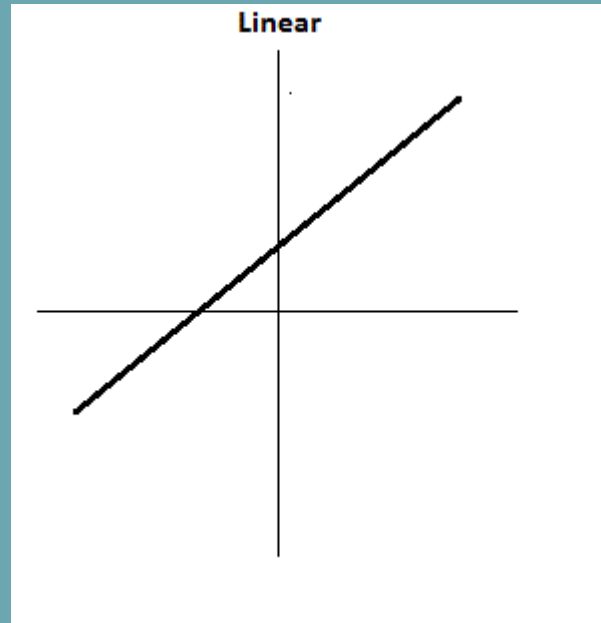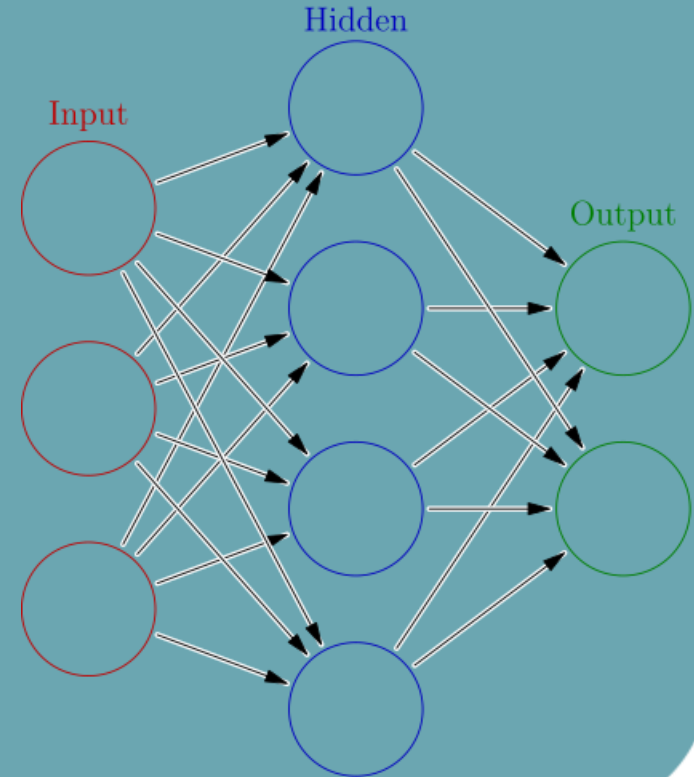
Components of ANN

- **Neurons**
  - ANNs are composed of artificial neurons which are conceptually derived from biological neurons.
  - Each artificial neuron has inputs and produces a single output which can be sent to multiple other neurons.
  - The inputs can be the feature values of a sample of external data, such as images or documents, or they can be the outputs of other neurons.
  - The outputs of the final output neurons of the neural net accomplish the task, such as recognizing an object in an image.
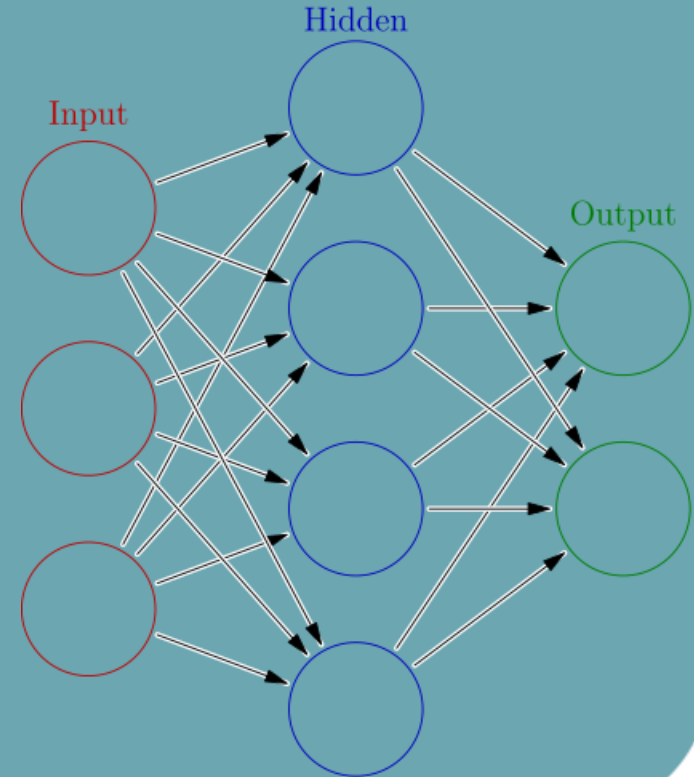
○ To find the output of the neuron, first we take the weighted sum of all the inputs, weighted by the weights of the connections from the inputs to the neuron.

○ We add a bias term to this sum. This weighted sum is sometimes called the activation. This weighted sum is then passed through a (usually nonlinear) activation/transformation function to produce the output.

○ The initial inputs are external data, such as images and documents. The ultimate outputs accomplish the task, such as recognizing an object in an image.

- Connections and weights
  - The network consists of connections, each connection providing the output of one neuron as an input to another neuron.
  - Each connection is assigned a weight that represents its relative importance.
 - A given neuron can have multiple input and output connections.
- Propagation function
  - The propagation function computes the input to a neuron from the outputs of its predecessor neurons and their connections as a weighted sum.
  - A bias term can be added to the result of the propagation.
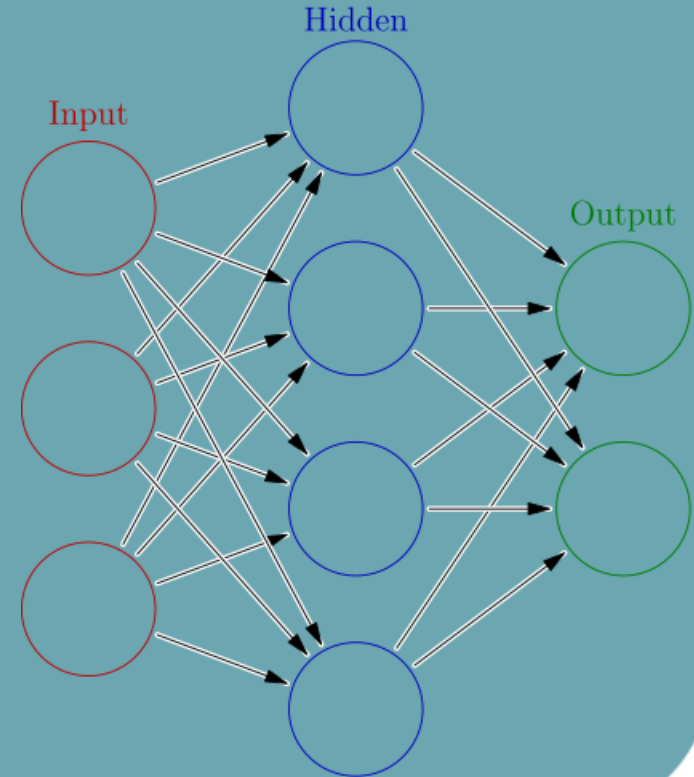
- Organization
  - The neurons are typically organized into multiple layers, especially in deep learning.
  - Neurons of one layer connect only to neurons of the immediately preceding and immediately following layers.
  - The layer that receives external data is the input layer.
  - The layer that produces the ultimate result is the output layer.
  - In between them are zero or more hidden layers.
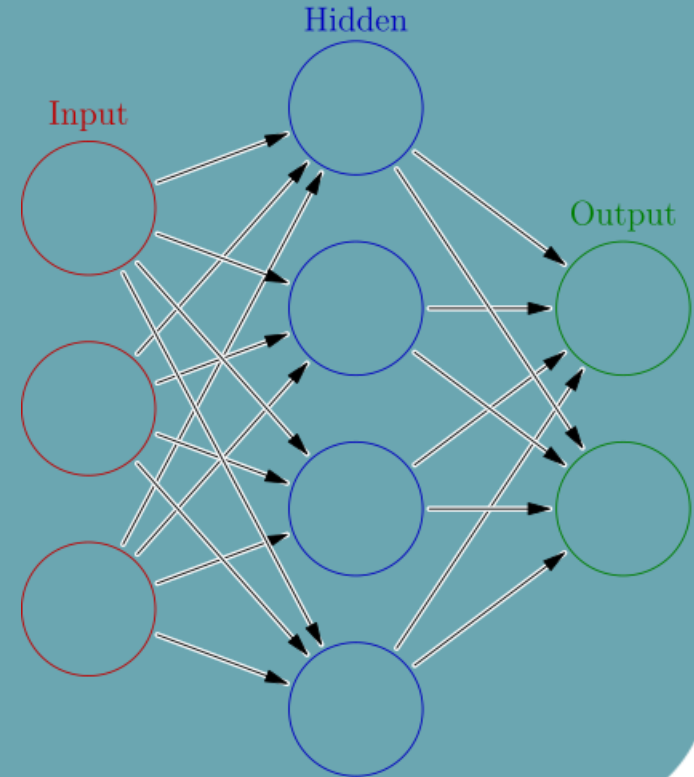
Hyper Parameter Tuning

- A hyperparameter is a constant parameter whose value is set before the learning process begins.
- The values of parameters are derived via learning.
- Examples of hyperparameters include learning rate, the number of hidden layers and batch size.
- The values of some hyperparameters can be dependent on those of other hyperparameters.
- For example, the size of some layers can depend on the overall number of layers.

Learning

- ○ Learning is the adaptation of the network to better handle a task by considering sample observations.
- ○ Learning involves adjusting the weights (and optional thresholds) of the network to improve the accuracy of the result.
- ○ This is done by minimizing the observed errors.
- ○ Learning is complete when examining additional observations does not usefully reduce the error rate.
- ○ Even after learning, the error rate typically does not reach 0. If after learning, the error rate is too high, the network typically must be redesigned.
- ○ Practically this is done by defining a cost function that is evaluated periodically during learning. As long as its output continues to decline, learning continues.

Hyper Parameter Tuning

- Learning rate
  - The learning rate defines the size of the corrective steps that the model takes to adjust for errors in each observation.
  - A high learning rate shortens the training time, but with lower ultimate accuracy, while a lower learning rate takes longer, but with the potential for greater accuracy.
  - Optimizations methods are primarily aimed at speeding up error minimization
  - In order to avoid oscillation inside the network such as alternating connection weights, and to improve the rate of convergence, refinements use an adaptive learning rate that increases or decreases as appropriate.
  - The concept of momentum allows the balance between the gradient and the previous change to be weighted such that the weight adjustment depends to some degree on the previous change.
- Cost function
  - loss function or cost function (sometimes also called an error function) is a function that maps an event or values of one or more variables onto a real number intuitively representing some "cost" associated with the event.
  - An optimization problem seeks to minimize a loss function.

Performance Assessment of ML Models

- After implementing a machine learning algorithm the next step is to find out how effective is the model based on metric and data sets.
- Classification/Regression models are evaluated based on different performance metrics.
- Different performance metrics are used to evaluate different Machine Learning Algorithms.
- Performance Assessment of Classification Model
  - Confusion matrix
  - precision
  - Recall
  - F1 score
  - Accuracy
  - ROC Curve –AUCMSE
- Performance Assessment of Regression Model
  - MAE
  - Cross-Entropy Loss

Confusion Matrix

○ Confusion Matrix as the name suggests gives us a matrix as output as N X N matrix , where N is the number of classes being predicted.

○ Confusion matrix, also known as an error matrix.

Confusion Matrix

- The Confusion matrix in itself is not a performance measure as such, but almost all of the performance metrics are based on confusion matrix and the numbers inside it.
  - True Positives: The cases in which we predicted YES and the actual output was also YES.
  - True Negatives: The cases in which we predicted NO and the actual output was NO.
  - False Positives: The cases in which we predicted YES and the actual output was NO.
  - False Negatives: The cases in which we predicted NO and the actual output was YES.

$$\textbf{Precision} = \frac{TP}{TP + FP}$$

$$\textbf{Recall} = \frac{TP}{TP + FN}$$

F1 Score: F1 score combines precision and recall relative to a specific positive class. It conveys the balance between the precision and the recall and there is an uneven class distribution. F1 score reaches its best value at 1 and worst at 0.

$$\textbf{F - measure} = \frac{2*Recall*Precision}{Recall + Precision}$$

- Accuracy:
  - we use Accuracy in classification problems and it is the most common evaluation metric.
  - Accuracy is defined as the ratio of the number of correct predictions made by the model over all kinds of predictions made.

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- Error Rate:
  - 1 - Accuracy

- AUC-ROC Curve:
  - When we need to check or visualize the performance of the multi-class classification problem, we use the AUC(Area Under The Curve) ROC (Receiver Operating Characteristics) curve.
  - AUC-ROC Curve is plotted with TPR against the FPR where TPR is on y-axis and FPR is on the x-axis.
  - _TPR/Recall/Sensitivity_ = TP/TP +FN
  - _Specificity_ = TN/TN+FP
  - _FPR_ = 1-Specificity= FP/TN+FP

- MSE / Quadratic loss / L2 loss:
  - Mean Squared Error, or MSE loss is the default loss to use for regression problems.

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (f_i - y_i)^2$$

where N is the number of data points, f_i the value returned by the model and y_i the actual value for data point i.

Say, y_i = [5,10,15,20] and f_i = [4.8,10.6,14.3,20.1]

MSE = 1/4 * (|5-4.8|^2+|10-10.6|^2+|15-14.3|^2+|20-20.1|^2) = 0.225

- Root Mean Squared Error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{j=1}^{n} (y_j - \hat{y}_j)^2}$$

  - where y_i is the actual expected output and $\hat{y}_i$ is the model's prediction
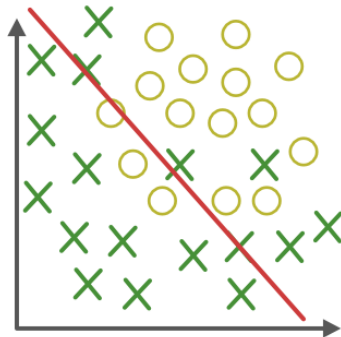 - RMSE = (0.225)^0.5 = 0.474
- Mean Absolute Error (MAE)

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i|$$

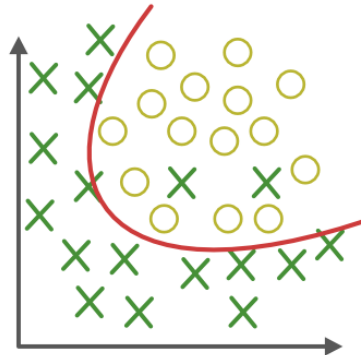Say, y_i = [5,10,15,20] and $\hat{y}_i$ = [4.8,10.6,14.3,20.1]

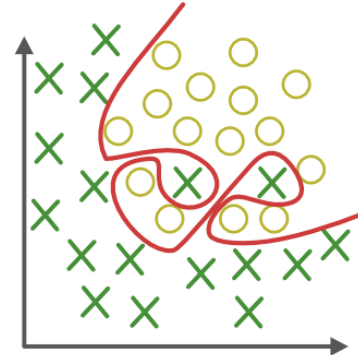Thus, MAE = 1/4 * (|5-4.8|+|10-10.6|+|15-14.3|+|20-20.1|) = 0.4

Model Fitting

Under-fitting (too simple to explain the variance) | Appropirate-fitting | Over-fitting (forcefitting--too good to be true)

Underfitting

○ A statistical model or a machine learning algorithm is said to have underfitting when it cannot capture the underlying trend of the data.

○ Underfitting destroys the accuracy of our machine learning model. Its occurrence simply means that our model or the algorithm does not fit the data well enough.

○ It usually happens when we have less data to build an accurate model and also when we try to build a linear model with fewer non-linear data.

○ Underfitting can be avoided by using more data and also reducing the features by feature selection.

Underfitting
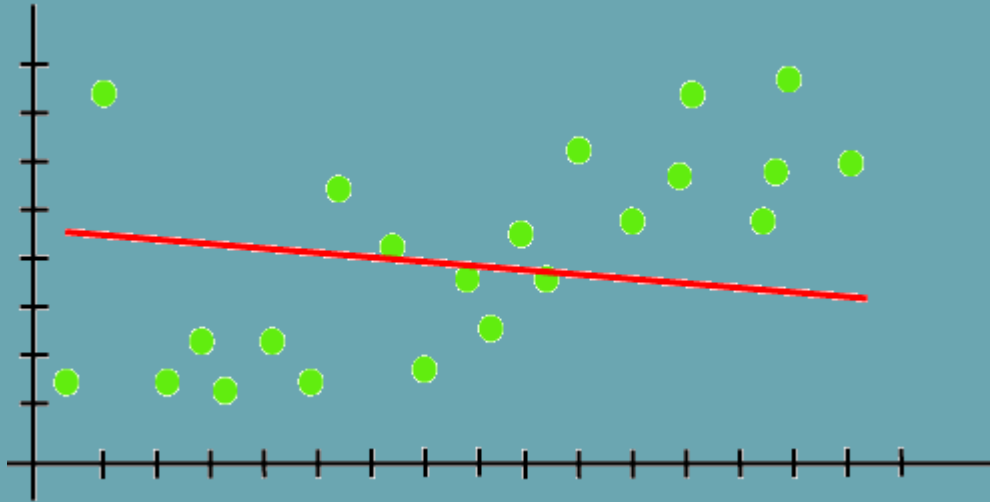
As we can see from the above diagram, the model is unable to capture the data points present in the plot.

Overfitting

- Overfitting occurs when our **machine learning model** tries to cover all the data points or more than the required data points present in the given dataset. Because of this, the model starts caching noise and inaccurate values present in the dataset, and all these factors reduce the efficiency and accuracy of the model. The overfitted model has **low bias and high variance.**
- The chances of occurrence of overfitting increase as much we provide training to our model. It means the more we train our model, the more chances of occurring the **overfitted model.**

- Overfitting is the main problem that occurs in **supervised learning.**

Overfitting
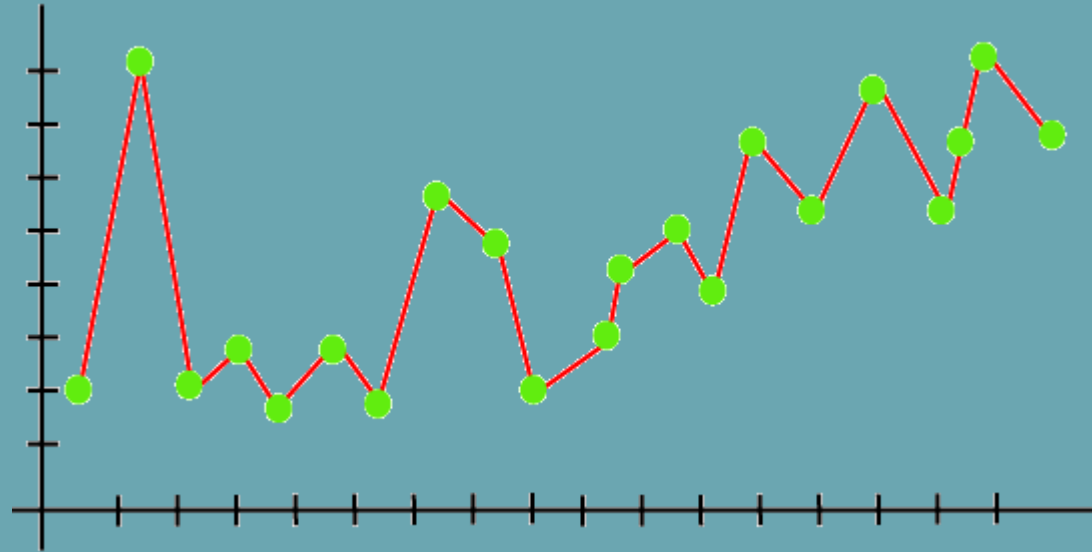
As we can see from the above graph, the model tries to cover all the data points present in the scatter plot. It may look efficient, but in reality, it is not so. Because the goal of the regression model to find the best fit line, but here we have not got any best fit, so, it will generate the prediction errors.

Feature Selection

- Feature selection, one of the main components of feature engineering, is the process of selecting the most important features to input in machine learning algorithms.

- Feature selection techniques are employed to reduce the number of input variables by eliminating redundant or irrelevant features and narrowing down the set of features to those most relevant to the machine learning model.

Feature Selection

The main benefits of performing feature selection in advance, rather than letting the machine learning model figure out which features are most important, include:

- Simpler models: simple models are easy to explain - a model that is too complex and unexplainable is not valuable
- Shorter training times: a more precise subset of features decreases the amount of time needed to train a model
- Variance reduction: increase the precision of the estimates that can be obtained for a given simulation
- Avoid the curse of high dimensionality: dimensionally cursed phenomena states that, as dimensionality and the number of features increases, the volume of space increases so fast that the available data become limited - PCA feature selection may be used to reduce dimensionality

Feature Selection -Filter methods:

- These methods are generally used while doing the pre-processing step. These methods select features from the dataset irrespective of the use of any machine learning algorithm.
- In terms of computation, they are very fast and inexpensive and are very good for removing duplicated, correlated, redundant features but these methods do not remove multicollinearity.
- Selection of feature is evaluated individually which can sometimes help when features are in isolation (don't have a dependency on other features) but will lag when a combination of features can lead to increase in the overall performance of the model.

Set of all features → Selecting the best subset → Learning algorithm → Performance

Filter Method (Techniques)

Information Gain – It is defined as the amount of information provided by the feature for identifying the target value and measures reduction in the entropy values. Information gain of each attribute is calculated considering the target values for feature selection.

Chi-square test — Chi-square method (X2) is generally used to test the relationship between categorical variables. It compares the observed values from different attributes of the dataset to its expected value.

$$X^2 = \sum \frac{(Observed\ value - Expected\ value)^2}{Expected\ value}$$

Fisher's Score – Fisher's Score selects each feature independently according to their scores under Fisher criterion leading to a suboptimal set of features. The larger the Fisher's score is, the better is the selected feature.

Correlation Coefficient – Pearson's Correlation Coefficient is a measure of quantifying the association between the two continuous variables and the direction of the relationship with its values ranging from -1 to 1.

Feature Selection –Wrapper Methods

- Wrapper methods, also referred to as greedy algorithms train the algorithm by using a subset of features in an iterative manner.
- Based on the conclusions made from training in prior to the model, addition and removal of features takes place.
- Stopping criteria for selecting the best subset are usually pre-defined by the person training the model such as when the performance of the model decreases or a specific number of features has been achieved.
- The main advantage of wrapper methods over the filter methods is that they provide an optimal set of features for training the model, thus resulting in better accuracy than the filter methods but are computationally more expensive.

Set of all features → Consider subset of all features → Learning algorithm → Performance

Selecting the best subset
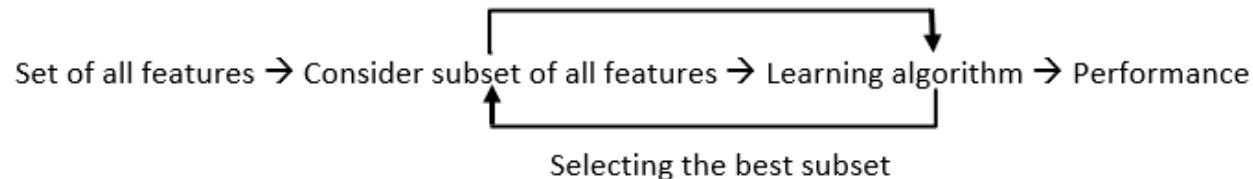
Wrapper Methods(Techniques)

- **Forward selection –** This method is an iterative approach where we initially start with an empty set of features and keep adding a feature which best improves our model after each iteration. The stopping criterion is till the addition of a new variable does not improve the performance of the model.

- **Backward elimination –** This method is also an iterative approach where we initially start with all features and after each iteration, we remove the least significant feature. The stopping criterion is till no improvement in the performance of the model is observed after the feature is removed.

- **Bi-directional elimination –** This method uses both forward selection and backward elimination technique simultaneously to reach one unique solution.

Feature Selection –Embedded methods

In embedded methods, the feature selection algorithm is blended as part of the learning algorithm, thus having its own built-in feature selection methods. Embedded methods encounter the drawbacks of filter and wrapper methods and merge their advantages. These methods are faster like those of filter methods and more accurate than the filter methods and take into consideration a combination of features as well.

Set of all features → Consider subset of all features → Learning algorithm + Performance

Selecting the best subset
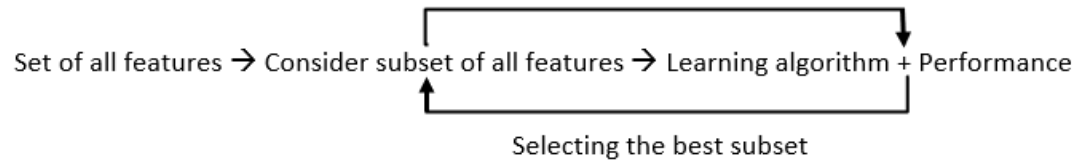
Embedded methods-Techniques

- **Regularization –** This method adds a penalty to different parameters of the machine learning model to avoid over-fitting of the model. This approach of feature selection uses Lasso (L1 regularization) and Elastic nets (L1 and L2 regularization). The penalty is applied over the coefficients, thus bringing down some coefficients to zero. The features having zero coefficient can be removed from the dataset.
- **Tree-based methods –** These methods such as Random Forest, Gradient Boosting provides us feature importance as a way to select features as well. Feature importance tells us which features are more important in making an impact on the target feature.

Regularization

○ Regularization is a technique used to reduce the errors by fitting the function appropriately on the given training set and avoid overfitting.

○ The commonly used regularization techniques are :

- L1 regularization
- L2 regularization
- Dropout regularization

- L1 and L2 are the most common types of regularization. These update the general cost function by adding another term known as the regularization term.
- In L1, we have:

$$Cost\ function\ =\ Loss\ +\ \frac{\lambda}{2m}\ *\ \sum \|w\|$$

- In L2, we have:

$$Cost\ function\ =\ Loss\ +\ \frac{\lambda}{2m}\ *\ \sum \|w\|^2$$

Regularization-Dropout

○ This is the one of the most interesting types of regularization techniques.

○ It also produces very good results and is consequently the most frequently used regularization technique in the field of deep learning.

○ At every iteration, it randomly selects some nodes and removes them along with all of their incoming and outgoing connections

Feature Engineering

- Feature engineering is the process of taking raw data and transforming it into features that can be used in machine learning algorithms.
- Features are the specific units of measurement that algorithms evaluate for correlations.
- Steps:
  - Data preparation
  - Exploratory data analysis
  - Establish a benchmark and choose features
  - Avoid bias in feature engineering
- (Home Work): Refer youtube