UNIT – I

FUNDAMENTALS OF DEEP LEARNING

ARTIFICIAL INTELLIGENCE:

Simulation of human brain, character, behaviours and Everything was Captured by the Machines, here the Machines are Computer systems**.**

History of Machine learning:

Machine learning was first conceived from the mathematical modeling of neural networks. A paper by logician Walter Pitts and neuroscientist Warren McCulloch, published in 1943, attempted to mathematically map out thought processes and decision making in human cognition.

To Understand the past history of Machine Learning we must know about its Life Cycle

The ML lifecycle is the cyclic iterative process with instructions, and best practices to use across defined phases while developing an ML workload. The ML lifecycle adds clarity and structure for making a machine learning project successful.

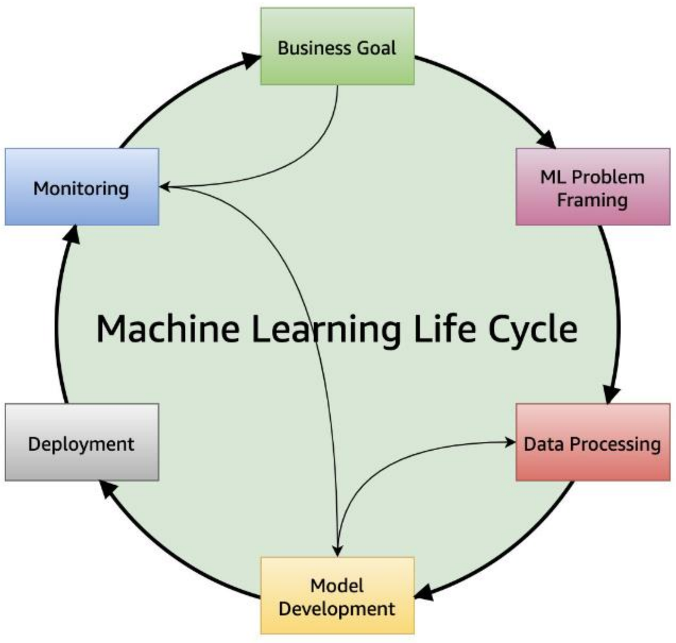


Fig. Life cycle of Machine Learning

In the1952 Arthur Samuel, who was the pioneer of machine learning, created a program that helped an IBM computer to play a checkers game.

It performed better more it played. In 1959, the term "Machine Learning" was first coined by Arthur Samuel.

The term "machine learning" was coined by Arthur Samuel, a computer scientist at IBM and a pioneer in AI and computer gaming.

Samuel designed a computer program for playing checkers. The more the program played, the more it learned from experience, using algorithms to make predictions.

PROBABILISTIC MODELING

Machine learning algorithms today rely heavily on probabilistic models, which take into consideration the uncertainty inherent in real-world data. These models make predictions based on probability distributions, rather than absolute values, allowing for a more nuanced and accurate understanding of complex systems. One common approach is Bayesian inference, where prior knowledge is combined with observed data to make predictions. Another approach is maximum likelihood estimation, which seeks to find the model that best fits observational data.

What are Probabilistic Models?

Probabilistic models are an essential component of machine learning, which aims to learn patterns from data and make predictions on new, unseen data. They are statistical models that capture the inherent uncertainty in data and incorporate it into their predictions. Probabilistic models are used in various applications such as image and speech recognition, natural language processing, and recommendation systems. In recent years, significant progress has been made in developing probabilistic models that can handle large datasets efficiently.

Categories Of Probabilistic Models

These models can be classified into the following categories:

- Generative models
- Discriminative models.
- Graphical models

Generative models:

Generative models aim to model the joint distribution of the input and output variables. These models generate new data based on the probability distribution of the original dataset. Generative models are powerful because they can generate new data that resembles the training data. They can be used for tasks such as image and speech synthesis, language translation, and text generation.

Discriminative models

The discriminative model aims to model the conditional distribution of the output variable given the input variable. They learn a decision boundary that separates the different classes of the output variable. Discriminative models are useful when the focus is on making accurate predictions rather than generating new data. They can be used for tasks such as image recognition, speech recognition, and sentiment analysis.

Graphical models

These models use graphical representations to show the conditional dependence between variables. They are commonly used for tasks such as image recognition, natural language processing, and causal inference.

Naive Bayes Algorithm in Probabilistic Models

The Naive Bayes algorithm is a widely used approach in probabilistic models, demonstrating remarkable efficiency and effectiveness in solving classification problems. By leveraging the power of the Bayes theorem and making simplifying assumptions about feature independence, the algorithm calculates the probability of the target class given the feature set. This method has found diverse applications across various industries, ranging from spam filtering to medical diagnosis. Despite its simplicity, the Naive Bayes algorithm has proven to be highly robust, providing rapid results in a multitude of real-world problems.

Naive Bayes is a probabilistic algorithm that is used for classification problems. It is based on the Bayes theorem of probability and assumes that the features are conditionally independent of each other given the class. The Naive Bayes Algorithm is used to calculate the probability of a given sample belonging to a particular class. This is done by calculating the posterior probability of each class given the sample and then selecting the class with the highest posterior probability as the predicted class.

The algorithm works as follows:

1. Collect a labeled dataset of samples, where each sample has a set of features and a class label.
2. For each feature in the dataset, calculate the conditional probability of the feature given the class.
3. This is done by counting the number of times the feature occurs in samples of the class and dividing by the total number of samples in the class.
4. Calculate the prior probability of each class by counting the number of samples in each class and dividing by the total number of samples in the dataset.
5. Given a new sample with a set of features, calculate the posterior probability of each class using the Bayes theorem and the conditional probabilities and prior probabilities calculated in steps 2 and 3.
6. Select the class with the highest posterior probability as the predicted class for the new sample.

Probabilistic Models in Deep Learning

Deep learning, a subset of machine learning, also relies on probabilistic models. Probabilistic models are used to optimize complex models with many parameters, such as neural networks. By incorporating uncertainty into the model training process, deep learning algorithms can provide higher accuracy and generalization capabilities. One popular technique is variational inference, which allows for efficient estimation of posterior distributions.

Importance of Probabilistic Models

- Probabilistic models play a crucial role in the field of machine learning, providing a framework for understanding the underlying patterns and complexities in massive datasets.
- Probabilistic models provide a natural way to reason about the likelihood of different outcomes and can help us understand the underlying structure of the data.
- Probabilistic models help enable researchers and practitioners to make informed decisions when faced with uncertainty.
- Probabilistic models allow us to perform Bayesian inference, which is a powerful method for updating our beliefs about a hypothesis based on new data. This can be particularly useful in situations where we need to make decisions under uncertainty.

Advantages Of Probabilistic Models

- Probabilistic models are an increasingly popular method in many fields, including artificial intelligence, finance, and healthcare.
- The main advantage of these models is their ability to take into account uncertainty and variability in data. This allows for more accurate predictions and decision-making, particularly in complex and unpredictable situations.
- Probabilistic models can also provide insights into how different factors influence outcomes and can help identify patterns and relationships within data.

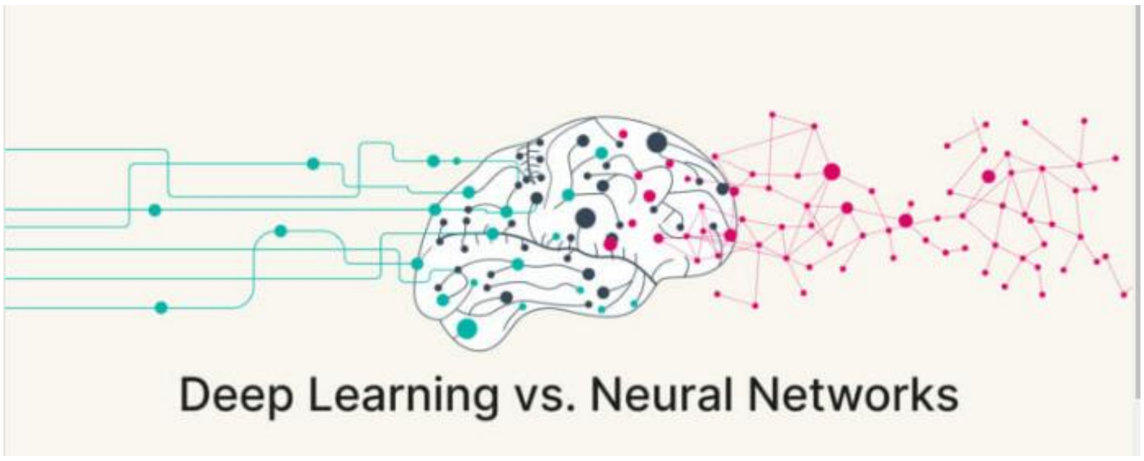
Disadvantages Of Probabilistic Models
There are also some disadvantages to using probabilistic models.

- One of the disadvantages is the potential for overfitting, where the model is too specific to the training data and doesn't perform well on new data.
- Not all data fits well into a probabilistic framework, which can limit the usefulness of these models in certain applications.
- Another challenge is that probabilistic models can be computationally intensive and require significant resources to develop and implement.

EARLY NEURAL NETWORKS:

A neural network is a method in artificial intelligence that teaches computers to process data in a way that is inspired by the human brain. It is a type of machine learning process, called deep learning, that uses interconnected nodes or neurons in a layered structure that resembles the human brain.

"Artificial Neural Network is biologically inspired by the neural network, which constitutes after the human brain. "



- ➔ Birds inspired us to fly, burdock plants inspired Velcro, and nature has inspired countless more inventions.
- ➔ It seems only logical, then, to look at the brain's architecture for inspiration on how to build an intelligent machine.
- ➔ This is the logic that sparked artificial neural networks (ANNs): an ANN is a Machine Learning model inspired by the networks of biological neurons found in our brains.
- ➔ ANNs are at the very core of Deep Learning. They are versatile, powerful, and scalable, making them ideal to tackle large and highly complex Machine Learning tasks .
- ➔ such as classifying billions of images (e.g., Google Images), powering speech recognition services (e.g., Apple's Siri), recommending the best videos to watch to hundreds of millions of users every day (e.g., YouTube), or learning to beat the world champion at the game of Go (DeepMind's AlphaGo).
- ➔ Keras is an open-source neural-network library written in Python.

- ➔ It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, Theano, or PlaidML. Keras is a high-level, deep learning API developed by Google for implementing neural networks.
- ➔ It is written in Python and is used to make the implementation of neural networks easy. It also supports multiple backend neural network computation.
- ➔ Keras is **a neural network Application Programming Interface (API) for Python that is tightly integrated with TensorFlow, which is used to build machine learning models**.
- ➔ Keras' models offer a simple, user-friendly way to define a neural network, which will then be built for you by TensorFlow.
- ➔ Keras is used for **creating deep models which can be productized on smartphones**. Keras is also used for distributed training of deep learning models.
- ➔ Keras is used by companies such as Netflix, Yelp, Uber, etc.

Biological Neurons:

- ➔ It's composed of a cell body containing the nucleus and most of the cell's complex components, many branching extensions called dendrites, plus one very long extension called the axon.
- ➔ The axon's length may be just a few times longer than the cell body, or up to tens of thousands of times longer. Near its extremity the axon splits off into many branches called telodendria.
- ➔ At the tip of these branches are minuscule structures called synaptic terminals (or simply synapses), which are connected to the dendrites or cell bodies of other neurons.
- ➔ Biological neurons produce short electrical impulses called action potentials (APs, or just signals) which travel along the axons and make the synapses release chemical signals called neurotransmitters.
- ➔ When a neuron receives a sufficient amount of these neurotransmitters within a few milliseconds, it fires its own electrical impulses (actually, it depends on the neurotransmitters, as some of them inhibit the neuron from firing).

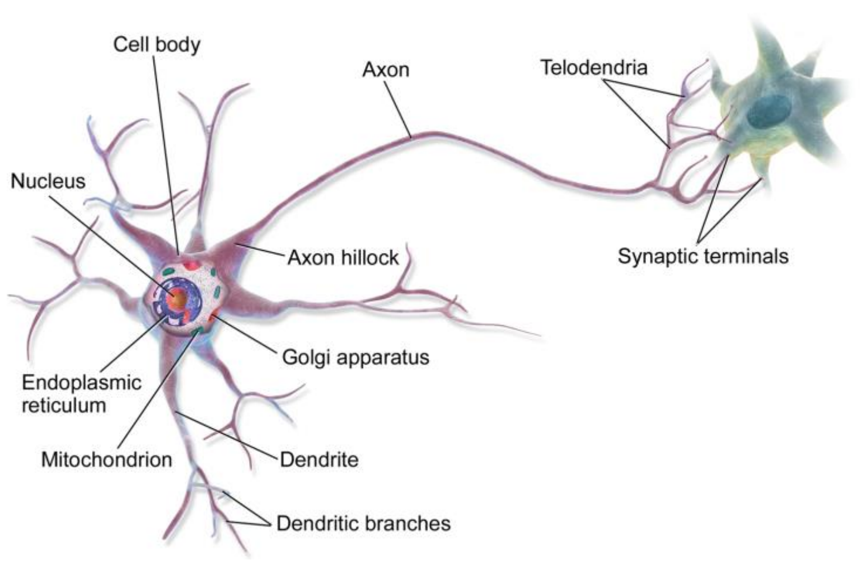


Figure. Biological neuron

- ➔ The architecture of biological neural networks (BNNs) is still the subject of active research, but some parts of the brain have been mapped.
- ➔ It seems that neurons are often organized in consecutive layers, especially in the cerebral cortex (i.e., the outer layer of your brain), as shown in Figure.

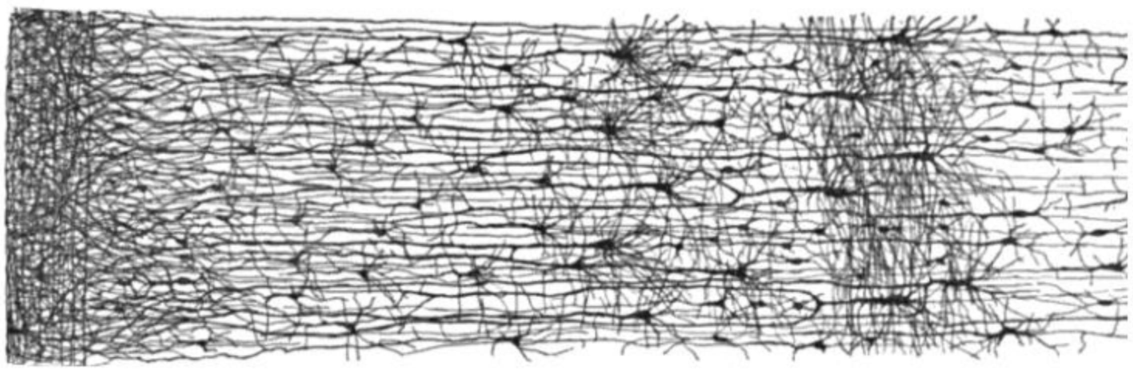


Figure. Multiple layers in a biological neural network (human cortex)

Logical Computations with Neurons:

- ➔ McCulloch and Pitts proposed a very simple model of the biological neuron, which later became known as an artificial neuron:
- ➔ It has one or more binary (on/off) inputs and one binary output. The artificial neuron activates its output when more than a certain number of its inputs are active.
- ➔ A simplified model it is possible to build a network of artificial neurons that computes any logical proposition you want.
- ➔ Let’s build a few ANNs that perform various logical computations (see in Figure), assuming that a neuron is activated when at least two of its inputs are active.

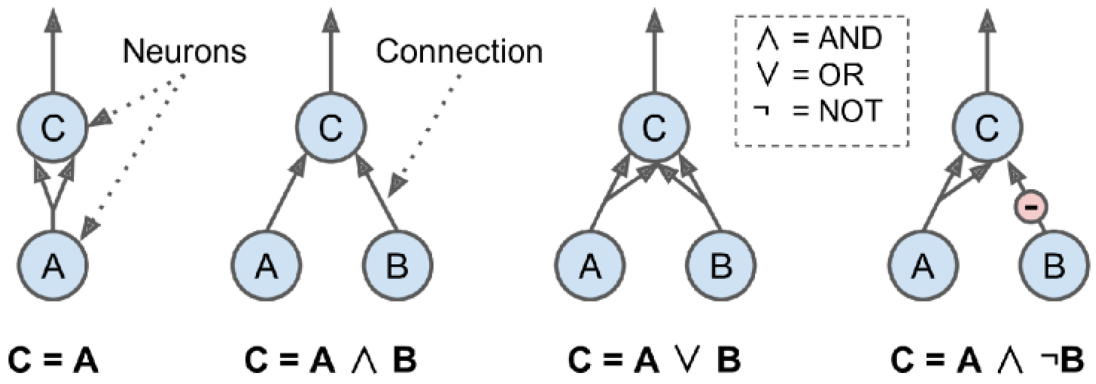


Figure. ANNs performing simple logical computations

The Perceptron:

- ➔ The Perceptron is one of the simplest ANN architectures, invented in 1957 by Frank Rosenblatt. It is based on a slightly different artificial neuron (shown in below Figure) called a threshold logic unit (TLU), or sometimes a linear threshold unit (LTU).

- ➔ The inputs and output are numbers (instead of binary on/off values), and each input connection is associated with a weight.
- ➔ The TLU computes a weighted sum of its inputs (z = w x + w x + ⋯ + w x = x w), then applies a step function to that sum and outputs the result: h (x) = step(z), where z = x w.

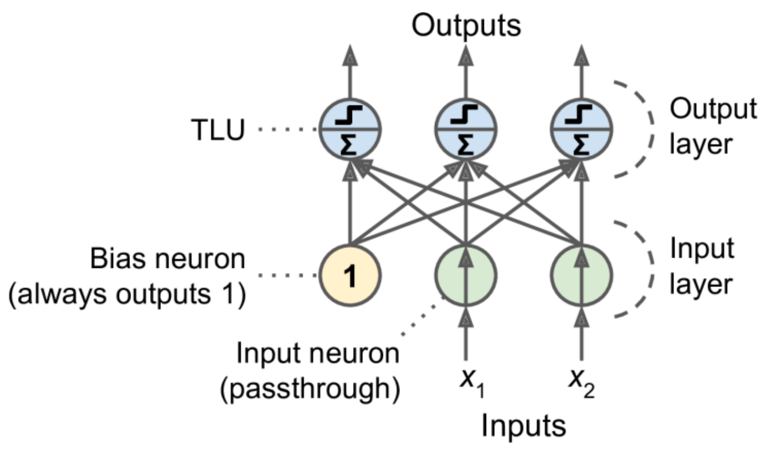


Figure. Architecture of a Perceptron with two input neurons, one bias neuron, and three output neurons.

KERNAL METHODS:

Kernel method in machine learning is defined as the class of algorithms for pattern analysis, which is used to study and find the general types of relations (such as correlation, classification, ranking, clusters, principle components, etc) in datasets by transforming raw representation of the data explicitly into feature vector representation using a user-specified feature map so that the high dimensional implicit feature space of these data can be operated with computing the coordinates of the data in that particular space.

The algorithm used for pattern analysis. In general pattern, analysis is done to find relations in datasets. These relations can be clustering, classification, principal components, correlation, etc. Most of these algorithms that solve these

tasks of analyzing the pattern, Need the data in raw representative, to be explicitly transformed into a feature vector representation. This transformation can be done via a user-specified feature map. So, it can be taken that only the user-specified kernel is required by the kernel method.

The terminology Kernal Method comes from the fact that they use kernel function, which allows them to perform the operation in high-dimensional, implicit feature space without the need of computing the coordinates of the data in that space. Instead, they simply compute the inner product between the images of all pairs of data in feature space.

These kinds of operations are computationally cheaper most of the time compared to the explicit computation of the coordinates. This technique is termed as 'kernel trick'. Any linear model can be converted into a non-linear model by applying the kernel trick to the model.

Kernel Method available in machine learning is principal components analysis (PCA), spectral clustering, support vector machines (SVM), canonical correlation analysis, kernel perceptron, Gaussian processes, ridge regression, linear adaptive filters, and many others. Let's have a high-level understanding of a few of these kernel methods.

Top 7 Methods of Kernel in Machine Learning

Here are the Methods of Kernel in Machine Learning mention below:

1. Principle Component Analysis

2. Support Vector Machine

3. Gaussian Process

4. Canonical Correlation Analysis

5. Spectral Clustering

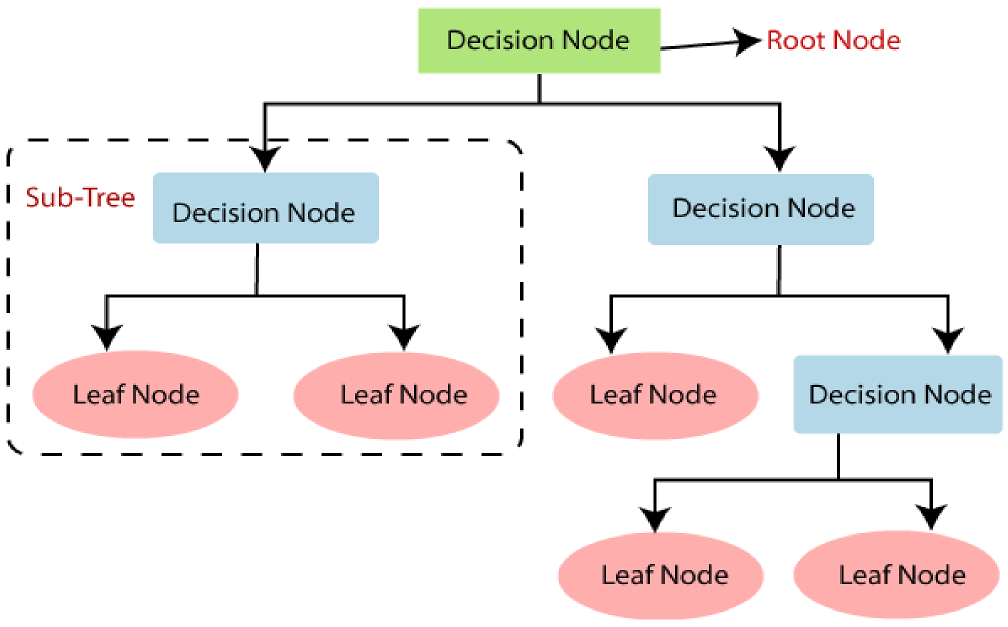
6. Adaptive Filter

7. Kernel Perceptron

DECISION TREES:

- Decision Tree is a **Supervised learning technique** that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where **internal nodes represent the features of a dataset, branches represent the decision rules** and **each leaf node represents the outcome.**
- In a Decision tree, there are two nodes, which are the **Decision Node** and **Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.
- ***It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.***
- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
- In order to build a tree, we use the **CART algorithm,** which stands for **Classification and Regression Tree algorithm.**
- A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.
- Below diagram explains the general structure of a decision tree:

Note: A decision tree can contain categorical data (YES/NO) as well as numeric data.



Why use Decision Trees?

There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision tree:

- Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
- The logic behind the decision tree can be easily understood because it shows a tree-like structure.

Decision Tree Terminologies

- **Root Node:** Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.
- **Leaf Node:** Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
- **Splitting:** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
- **Branch/Sub Tree:** A tree formed by splitting the tree.
- **Pruning:** Pruning is the process of removing the unwanted branches from the tree.
- **Parent/Child node:** The root node of the tree is called the parent node, and other nodes are called the child nodes.

How does the Decision Tree algorithm Work?

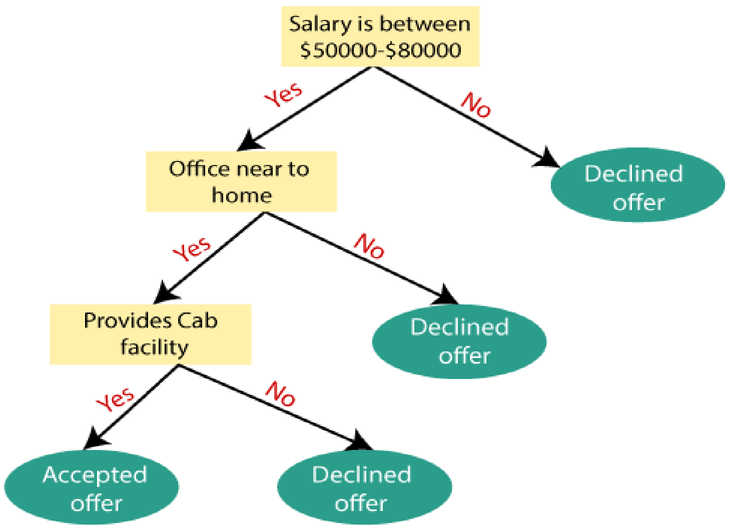
In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

- **Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.
- **Step-2:** Find the best attribute in the dataset using **Attribute Selection Measure (ASM).**
- **Step-3:** Divide the S into subsets that contains possible values for the best attributes.

- **Step-4:** Generate the decision tree node, which contains the best attribute.
- **Step-5:** Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

Example: Suppose there is a candidate who has a job offer and wants to decide whether he should accept the offer or Not. So, to solve this problem, the decision tree starts with the root node (Salary attribute by ASM). The root node splits further into the next decision node (distance from the office) and one leaf node based on the corresponding labels. The next decision node further gets split into one decision node (Cab facility) and one leaf node. Finally, the decision node splits into two leaf nodes (Accepted offers and Declined offer). Consider the below diagram:



Attribute Selection Measures

While implementing a Decision tree, the main issue arises that how to select the best attribute for the root node and for sub-nodes. So, to solve such problems there is a technique which is called as **Attribute selection measure or ASM.** By this measurement, we can easily select the best attribute for the nodes of the tree. There are two popular techniques for ASM, which are:

- **Information Gain**
- **Gini Index**

1. Information Gain:

- Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute.
- It calculates how much information a feature provides us about a class.
- According to the value of information gain, we split the node and build the decision tree.
- A decision tree algorithm always tries to maximize the value of information gain, and a node/attribute having the highest information gain is split first. It can be calculated using the below formula:

1. Information Gain= Entropy(S)- [(Weighted Avg) *Entropy(each feature)

Entropy: Entropy is a metric to measure the impurity in a given attribute. It specifies randomness in data. Entropy can be calculated as:

Entropy(s)= -P(yes)log2 P(yes)- P(no) log2 P(no)
Where,

- **S= Total number of samples**
- **P(yes)= probability of yes**
- **P(no)= probability of no**

2. Gini Index:

- Gini index is a measure of impurity or purity used while creating a decision tree in the CART(Classification and Regression Tree) algorithm.
- An attribute with the low Gini index should be preferred as compared to the high Gini index.
- It only creates binary splits, and the CART algorithm uses the Gini index to create binary splits.
- Gini index can be calculated using the below formula:

Gini Index= $1- \sum_j P_j^2$

Pruning: Getting an Optimal Decision tree

Pruning is a process of deleting the unnecessary nodes from a tree in order to get the optimal decision tree.

A too-large tree increases the risk of overfitting, and a small tree may not capture all the important features of the dataset. Therefore, a technique that decreases the size of the learning tree without reducing accuracy is known as Pruning. There are mainly two types of tree **pruning** technology used:

- **Cost Complexity Pruning**
- **Reduced Error Pruning.**

Advantages of the Decision Tree

- It is simple to understand as it follows the same process which a human follow while making any decision in real-life.
- It can be very useful for solving decision-related problems.
- It helps to think about all the possible outcomes for a problem.
- There is less requirement of data cleaning compared to other algorithms.

Disadvantages of the Decision Tree

- The decision tree contains lots of layers, which makes it complex.
- It may have an overfitting issue, which can be resolved using the **Random Forest algorithm.**
- For more class labels, the computational complexity of the decision tree may increase.

RANDOM FOREST ALGORITHM:

 Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML.
 It is based on the concept of **ensemble learning,** which is a process of *combining multiple classifiers to solve a complex problem and to improve the performance of the model.*

 As the name suggests, ***"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."***
 Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.
The below diagram explains the working of the Random Forest algorithm:
Note: To better understand the Random Forest Algorithm, you should have knowledge of the Decision Tree Algorithm.
Assumptions for Random Forest
Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random Forest classifier:
o There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.

o The predictions from each tree must have very low correlations.

Why use Random Forest:
Below are some points that explain why we should use the Random Forest algorithm:
o It takes less training time as compared to other algorithms.

o It predicts output with high accuracy, even for the large dataset it runs efficiently.

o It can also maintain accuracy when a large proportion of data is missing.

Steps for Random Forest algorithm work:
Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.
The Working process can be explained in the below steps and diagram:
Step-1: Select random K data points from the training set.
Step-2: Build the decision trees associated with the selected data points (Subsets).
Step-3: Choose the number N for decision trees that you want to build.
Step-4: Repeat Step 1 & 2.
Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.
The working of the algorithm can be better understood by the below example:
Example: Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random Forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision. Consider the below image:

Advantages of Random Forest

 Random Forest is capable of performing both Classification and Regression tasks.

 It is capable of handling large datasets with high dimensionality.

 It enhances the accuracy of the model and prevents the overfitting issue.

 Can perform both Regression and classification tasks.

 Produces good predictions that can be understood easily.

 Can handle large data sets efficiently.

 Provides a higher level of accuracy in predicting outcomes over the decision algorithm.

Disadvantages of Random Forest

o Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

o Less intuitive when we have an extensive collection of decision trees.
o It Consumes more time compared to the decision tree algorithm.
o While using a Random Forest Algorithm, more resources are required for computation. Extremely complex and requires more computational resources.

Applications of Random Forest

1. Banking: It predicts a loan applicant's solvency. This helps lending institutions make a good decision on whether to give the customer loan or not. They are also being used to detect fraudsters.
2. Health Care: Health professionals use random forest systems to diagnose patients. Patients are diagnosed by assessing their previous medical history. Past medical records are reviewed to establish the proper dosage for the patients.

3. Stock Market: Financial analysts use it to identify potential markets for stocks. It also enables them to remember the behaviour of stocks.

4. E-Commerce: Through this system, e-commerce vendors can predict the preference of customers based on past consumption behaviour.

(or)
Applications of Random Forest
There are mainly four sectors where Random Forest mostly used:
1. **Banking:** Banking sector mostly uses this algorithm for the identification of loan risk.

2. **Medicine:** With the help of this algorithm, disease trends and risks of the disease can be identified.

3. **Land Use:** We can identify the areas of similar land use by this algorithm.

4. **Marketing:** Marketing trends can be identified using this algorithm

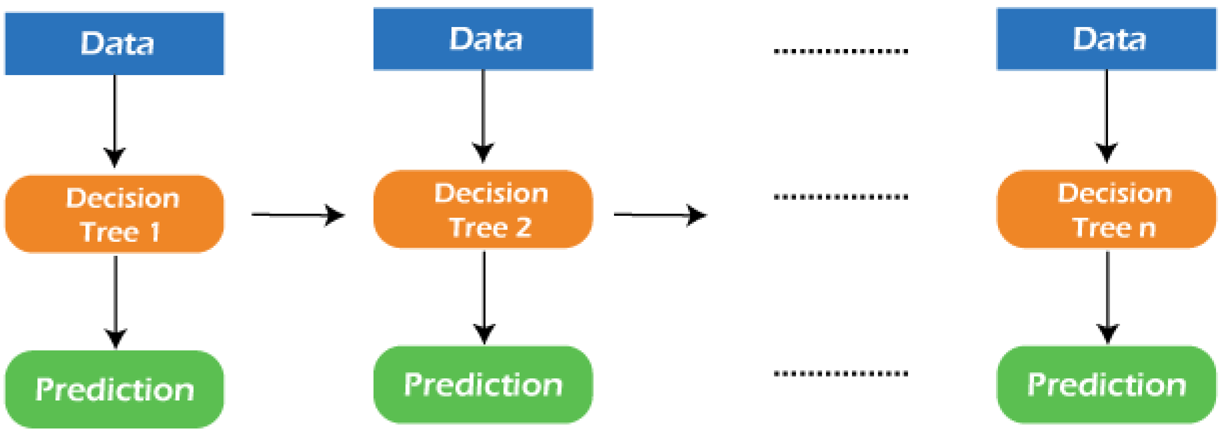
GBM in Machine Learning:

Gradient Boosting Machine (GBM) is one of the most popular forward learning ensemble methods in machine learning. It is a powerful technique for building predictive models for regression and classification tasks.

GBM helps us to get a predictive model in form of an ensemble of weak prediction models such as decision trees. Whenever a decision tree performs as a weak learner then the resulting algorithm is called gradient-boosted trees.

It enables us to combine the predictions from various learner models and build a final predictive model having the correct prediction.

But here one question may arise if we are applying the same algorithm then how multiple decision trees can give better predictions than a single decision tree? Moreover, how does each decision tree capture different information from the same data?



So, the answer to these questions is that a different subset of features is taken by the nodes of each decision tree to select the best split. It means, that each tree behaves differently, and hence captures different signals from the same data.

How do GBM works?

Generally, most supervised learning algorithms are based on a single predictive model such as linear regression, penalized regression model, decision trees, etc. But there are some supervised algorithms in ML that depend on a combination of various models together through the ensemble. In other words, when multiple base models contribute their predictions, an average of all predictions is adapted by boosting algorithms.

Gradient boosting machines consist 3 elements as follows:

- Loss function
- Weak learners
- Additive model

Let's understand these three elements in detail.

1. Loss function:

Although, there is a big family of Loss functions in machine learning that can be used depending on the type of tasks being solved. The use of the loss function is estimated by the demand of specific characteristics of the conditional distribution such as robustness. While using a loss function in our task, we must specify the loss function and the function to calculate the corresponding negative gradient. Once, we get these two functions, they can be implemented into gradient boosting machines easily. However, there are several loss functions have been already proposed for GBM algorithms.

Classification of loss function:

Based on the type of response variable y, loss function can be classified into different types as follows:

1. **Continuous response, $y \in R$:**
   - Gaussian L2 loss function
   - Laplace L1 loss function
   - Huber loss function, δ specified
   - Quantile loss function, α specified
2. **Categorical response, $y \in \{0, 1\}$:**
   - Binomial loss function
   - Adaboost loss function
3. **Other families of response variables:**
   - Loss functions for survival models
   - Loss functions count data
   - Custom loss functions

2. Weak Learner:

Weak learners are the base learner models that learn from past errors and help in building a strong predictive model design for boosting algorithms in machine learning. Generally, decision trees work as a weak learners in boosting algorithms.

Boosting is defined as the framework that continuously works to improve the output from base models. Many gradient boosting applications allow you to "plugin" various classes of weak learners at your disposal. Hence, decision trees are most often used for weak (base) learners.

How to train weak learners:

Machine learning uses training datasets to train base learners and based on the prediction from the previous learner, it improves the performance by focusing on the rows of the training data where the previous tree had the largest errors or residuals. E.g. shallow trees are considered weak learner to decision trees as it contains a few splits. Generally, in boosting algorithms, trees having up to 6 splits are most common.

Below is a sequence of training the weak learner to improve their performance where each tree is in the sequence with the previous tree's residuals. Further, we are introducing each new tree so that it can learn from the previous tree's errors. These are as follows:

1. Consider a data set and fit a decision tree into it.

   F1(x)=y

2. Fit the next decision tree with the largest errors of the previous tree.

   h1(x)=y?F1(x)

3. Add this new tree to the algorithm by adding both in steps 1 and 2.

   F2(x)=F1(x)+h1(x)

4. Again fit the next decision tree with the residuals of the previous tree.

   h2(x)=y?F2(x)

5. Repeat the same which we have done in step 3.

   F3(x)=F2(x)+h2(x)

Continue this process until some mechanism (i.e. cross-validation) tells us to stop. The final model here is a stagewise additive model of b individual trees:

f(x)=B∑b=1fb(x)

Hence, trees are constructed greedily, choosing the best split points based on purity scores like Gini or minimizing the loss.

3. Additive Model:

The additive model is defined as adding trees to the model. Although we should not add multiple trees at a time, only a single tree must be added so that existing trees in the model are not changed. Further, we can also prefer the gradient descent method by adding trees to reduce the loss.

In the past few years, the gradient descent method was used to minimize the set of parameters such as the coefficient of the regression equation and weight in a neural network. After calculating error or loss, the weight parameter is used to minimize the error. But recently, most ML experts prefer weak learner sub-models or decision trees as a substitute for these parameters. In which, we have to add a tree in the model to reduce the error and improve the performance of that model. In this way, the prediction from the newly added tree is combined with the prediction from the existing series of trees to get a final prediction. This process continues until the loss reaches an acceptable level or is no longer improvement required.

This method is also known as functional gradient descent or gradient descent with functions.

FUNDAMENTALS OF ML

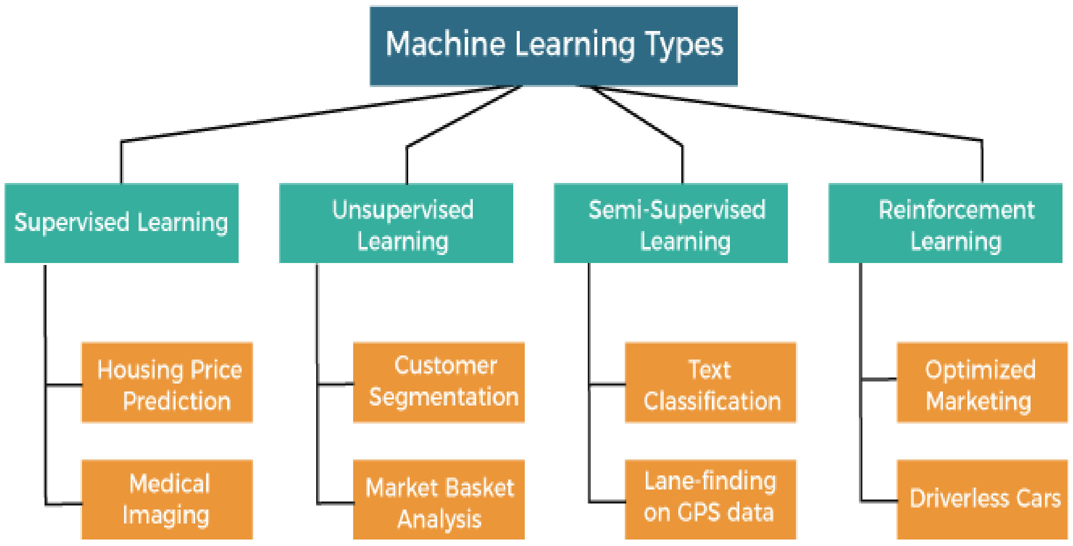
TYPES OF MACHINE LEARNING:

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.

These ML algorithms help to solve different business problems like Regression, Classification, Forecasting, Clustering, and Associations, etc.

Based on the methods and way of learning, machine learning is divided into mainly four types, which are:

1. Supervised Machine Learning
2. Unsupervised Machine Learning
3. Semi-Supervised Machine Learning
4. Reinforcement Learning



In this topic, we will provide a detailed description of the types of Machine Learning along with their respective algorithms:

1. Supervised Machine Learning

As its name suggests, Supervised machine learning is based on supervision. It means in the supervised learning technique, we train the machines using the "labelled" dataset, and based on the training, the machine predicts the output. Here, the labelled data specifies that some of the inputs are already mapped to the output. More preciously, we can say; first, we train the machine with the input and corresponding output, and then we ask the machine to predict the output using the test dataset.

Let's understand supervised learning with an example. Suppose we have an input dataset of cats and dog images. So, first, we will provide the training to the machine to understand the images, such as the **shape & size of the tail of cat and dog, Shape of eyes, colour, height (dogs are taller, cats are smaller), etc.** After completion of training, we input the picture of a cat and ask the machine to identify the object and predict the output. Now, the machine is well trained, so it will check all the features of the object, such as height, shape, colour, eyes, ears, tail, etc., and find that it's a cat. So, it will put it in the Cat category. This is the process of how the machine identifies the objects in Supervised Learning.

The main goal of the supervised learning technique is to map the input variable(x) with the output variable(y). Some real-world applications of supervised learning are **Risk Assessment, Fraud Detection, Spam filtering,** etc.

Categories of Supervised Machine Learning

Supervised machine learning can be classified into two types of problems, which are given below:

- **Classification**
- **Regression**

a) Classification

Classification algorithms are used to solve the classification problems in which the output variable is categorical, such as "**Yes" or No, Male or Female, Red or Blue, etc**. The classification algorithms predict the categories present in the dataset. Some real-world examples of classification algorithms are **Spam Detection, Email filtering, etc.**

Some popular classification algorithms are given below:

- **Random Forest Algorithm**
- **Decision Tree Algorithm**
- **Logistic Regression Algorithm**
- **Support Vector Machine Algorithm**

b) Regression

Regression algorithms are used to solve regression problems in which there is a linear relationship between input and output variables. These are used to predict continuous output variables, such as market trends, weather prediction, etc.

Some popular Regression algorithms are given below:

- **Simple Linear Regression Algorithm**
- **Multivariate Regression Algorithm**
- **Decision Tree Algorithm**
- **Lasso Regression**

Advantages and Disadvantages of Supervised Learning

Advantages:

- Since supervised learning work with the labelled dataset so we can have an exact idea about the classes of objects.
- These algorithms are helpful in predicting the output on the basis of prior experience.

Disadvantages:

- These algorithms are not able to solve complex tasks.
- It may predict the wrong output if the test data is different from the training data.
- It requires lots of computational time to train the algorithm.

Applications of Supervised Learning

Some common applications of Supervised Learning are given below:

- **Image Segmentation:**
  Supervised Learning algorithms are used in image segmentation. In this process, image classification is performed on different image data with pre-defined labels.
- **Medical Diagnosis:**
  Supervised algorithms are also used in the medical field for diagnosis purposes. It is done by using medical images and past labelled data with labels for disease conditions. With such a process, the machine can identify a disease for the new patients.
- **Fraud Detection -** Supervised Learning classification algorithms are used for identifying fraud transactions, fraud customers, etc. It is done by using historic data to identify the patterns that can lead to possible fraud.
- **Spam detection -** In spam detection & filtering, classification algorithms are used. These algorithms classify an email as spam or not spam. The spam emails are sent to the spam folder.

- **Speech Recognition -** Supervised learning algorithms are also used in speech recognition. The algorithm is trained with voice data, and various identifications can be done using the same, such as voice-activated passwords, voice commands, etc.

2. Unsupervised Machine Learning

Unsupervised learning is different from the Supervised learning technique; as its name suggests, there is no need for supervision. It means, in unsupervised machine learning, the machine is trained using the unlabeled dataset, and the machine predicts the output without any supervision.

In unsupervised learning, the models are trained with the data that is neither classified nor labelled, and the model acts on that data without any supervision.

The main aim of the unsupervised learning algorithm is to group or categories the unsorted dataset according to the similarities, patterns, and differences. Machines are instructed to find the hidden patterns from the input dataset.

Let's take an example to understand it more preciously; suppose there is a basket of fruit images, and we input it into the machine learning model. The images are totally unknown to the model, and the task of the machine is to find the patterns and categories of the objects.

So, now the machine will discover its patterns and differences, such as colour difference, shape difference, and predict the output when it is tested with the test dataset.

Categories of Unsupervised Machine Learning

Unsupervised Learning can be further classified into two types, which are given below:

- **Clustering**
- **Association**

1) Clustering

The clustering technique is used when we want to find the inherent groups from the data. It is a way to group the objects into a cluster such that the objects with the most similarities remain in one group and have fewer or no similarities with the objects of other groups. An example of the clustering algorithm is grouping the customers by their purchasing behaviour.

Some of the popular clustering algorithms are given below:

- **K-Means Clustering algorithm**
- **Mean-shift algorithm**
- **DBSCAN Algorithm**
- **Principal Component Analysis**

- **Independent Component Analysis**

2) Association

Association rule learning is an unsupervised learning technique, which finds interesting relations among variables within a large dataset. The main aim of this learning algorithm is to find the dependency of one data item on another data item and map those variables accordingly so that it can generate maximum profit. This algorithm is mainly applied in **Market Basket analysis, Web usage mining, continuous production**, etc.

Some popular algorithms of Association rule learning are **Apriori Algorithm, Eclat, FP-growth algorithm.**

Advantages and Disadvantages of Unsupervised Learning Algorithm

Advantages:

- These algorithms can be used for complicated tasks compared to the supervised ones because these algorithms work on the unlabeled dataset.
- Unsupervised algorithms are preferable for various tasks as getting the unlabeled dataset is easier as compared to the labelled dataset.

Disadvantages:

- The output of an unsupervised algorithm can be less accurate as the dataset is not labelled, and algorithms are not trained with the exact output in prior.
- Working with Unsupervised learning is more difficult as it works with the unlabelled dataset that does not map with the output.

Applications of Unsupervised Learning

- **Network Analysis:** Unsupervised learning is used for identifying plagiarism and copyright in document network analysis of text data for scholarly articles.
- **Recommendation Systems:** Recommendation systems widely use unsupervised learning techniques for building recommendation applications for different web applications and e-commerce websites.
- **Anomaly Detection:** Anomaly detection is a popular application of unsupervised learning, which can identify unusual data points within the dataset. It is used to discover fraudulent transactions.
- **Singular Value Decomposition:** Singular Value Decomposition or SVD is used to extract particular information from the database. For example, extracting information of each user located at a particular location.

3. Semi-Supervised Learning

Semi-Supervised learning is a type of Machine Learning algorithm that lies between Supervised and Unsupervised machine learning. It represents the intermediate ground between Supervised (With Labelled training data) and Unsupervised learning (with no labelled training data) algorithms and uses the combination of labelled and unlabeled datasets during the training period.

Although Semi-supervised learning is the middle ground between supervised and unsupervised learning and operates on the data that consists of a few labels, it mostly consists of unlabeled data. As labels are costly, but for corporate purposes, they may have few labels. It is completely different from supervised and unsupervised learning as they are based on the presence & absence of labels.

To overcome the drawbacks of supervised learning and unsupervised learning algorithms, the concept of Semi-supervised learning is introduced. The main aim of semi-supervised learning is to effectively use all the available data, rather than only labelled data like in supervised learning. Initially, similar data is clustered along with an unsupervised learning algorithm, and further, it helps to label the unlabeled data into labelled data. It is because labelled data is a comparatively more expensive acquisition than unlabeled data.

We can imagine these algorithms with an example. Supervised learning is where a student is under the supervision of an instructor at home and college. Further, if that student is self-analysing the same concept without any help from the instructor, it comes under unsupervised learning. Under semi-supervised learning, the student has to revise himself after analyzing the same concept under the guidance of an instructor at college.

Advantages and disadvantages of Semi-supervised Learning

Advantages:

- It is simple and easy to understand the algorithm.
- It is highly efficient.
- It is used to solve drawbacks of Supervised and Unsupervised Learning algorithms.

Disadvantages:

- Iterations results may not be stable.
- We cannot apply these algorithms to network-level data.
- Accuracy is low.

4. Reinforcement Learning

Reinforcement learning works on a feedback-based process, in which an AI agent (A software component) automatically explore its surrounding by hitting & trail, taking action, learning from experiences, and improving its performance. Agent gets rewarded for each good action and get punished for each bad action; hence the goal of reinforcement learning agent is to maximize the rewards.

In reinforcement learning, there is no labelled data like supervised learning, and agents learn from their experiences only.

The reinforcement learning process is similar to a human being; for example, a child learns various things by experiences in his day-to-day life. An example of reinforcement learning is to play a game, where the Game is the environment, moves of an agent at each step define states, and the goal of the agent is to get a high score. Agent receives feedback in terms of punishment and rewards.

Due to its way of working, reinforcement learning is employed in different fields such as **Game theory, Operation Research, Information theory, multi-agent systems.**

A reinforcement learning problem can be formalized using **Markov Decision Process(MDP).** In MDP, the agent constantly interacts with the environment and performs actions; at each action, the environment responds and generates a new state.

Categories of Reinforcement Learning

Reinforcement learning is categorized mainly into two types of methods/algorithms:

- **Positive Reinforcement Learning:** Positive reinforcement learning specifies increasing the tendency that the required behaviour would occur again by adding something. It enhances the strength of the behaviour of the agent and positively impacts it.
- **Negative Reinforcement Learning:** Negative reinforcement learning works exactly opposite to the positive RL. It increases the tendency that the specific behaviour would occur again by avoiding the negative condition.

Real-world Use cases of Reinforcement Learning

- **Video Games:**
  RL algorithms are much popular in gaming applications. It is used to gain super-human performance. Some popular games that use RL algorithms are **AlphaGO** and **AlphaGO Zero**.
- **Resource Management:**
  The "Resource Management with Deep Reinforcement Learning" paper showed that how to use RL in computer to automatically learn and schedule resources to wait for different jobs in order to minimize average job slowdown.
- **Robotics:**
  RL is widely being used in Robotics applications. Robots are used in the industrial

and manufacturing area, and these robots are made more powerful with reinforcement learning. There are different industries that have their vision of building intelligent robots using AI and Machine learning technology.

- **Text Mining**
  Text-mining, one of the great applications of NLP, is now being implemented with the help of Reinforcement Learning by Salesforce company.

Advantages and Disadvantages of Reinforcement Learning

Advantages

- It helps in solving complex real-world problems which are difficult to be solved by general techniques.
- The learning model of RL is similar to the learning of human beings; hence most accurate results can be found.
- Helps in achieving long term results.

Disadvantage

- RL algorithms are not preferred for simple problems.
- RL algorithms require huge data and computations.
- Too much reinforcement learning can lead to an overload of states which can weaken the results.

The curse of dimensionality limits reinforcement learning for real physical systems.

EVALUATING MACHINE LEARNING MODELS:

It is an essential step in the development of any machine learning project. It helps to assess the accuracy and efficiency of the algorithm in solving solving the given problem. Here are some common methods for evaluating machine learning algorithms:

- TrainTest Split: The most common method is to divide the dataset into two parts - training set and test set. The algorithm is trained on the training set and evaluated on the test set. The accuracy of the model is measured by comparing the predicted output with the actual output of the test set.
- Cross-Validation: Cross-validation is a more advanced method for evaluating machine learning algorithms. It involves dividing the dataset into k-folds, where k is typically 5 or 10. The algorithm is trained on k-1 folds and validated on the remaining fold. This process is repeated k times, with each fold serving as the validation set once. The average accuracy of all k-folds is used as the final evaluation metric.
- Confusion Matrix: A confusion matrix is a table used to evaluate the performance of a classification model. It summarizes the predicted output of the model and compares it

with the actual output. The confusion matrix is used to calculate metrics such as accuracy, precision, recall, and F1 score.

- Receiver Operating Characteristic (ROC) Curve: ROC curve is a plot that summarizes the performance of a binary classification model at different classification thresholds. It plots the true positive rate (TPR) against the false positive rate (FPR). The area under the curve (AUC) of the ROC curve is used as the evaluation metric.
- Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE): MAE and RMSE are used to evaluate regression models. MAE calculates the average absolute difference between the predicted and actual output, while RMSE calculates the square root of the average squared difference between the predicted and actual output.
- Precision-Recall Curve: The precision-recall curve is another way to evaluate the performance of a classification model. It plots the precision against the recall at different classification thresholds. The area under the precision-recall curve (AUPRC) is used as the evaluation metric.

It's important to note that the choice of evaluation metric depends on the type of problem and the specific goals of the project. Therefore, it's essential to understand the strengths and limitations of each evaluation method and choose the appropriate one for the problem at hand.

UNDERFITTING OF TRAINING DATA

This process occurs when data is unable to establish an accurate relationship between input and output variables. It simply means trying to fit in undersized jeans. It signifies the data is too simple to establish a precise relationship. To overcome this issue:
 Maximize the training time
 Enhance the complexity of the model
 Add more features to the data
 Reduce regular parameters
 Increasing the training time of model

OVERFITTING OF TRAINING DATA

-> Overfitting refers to a machine learning model trained with a massive amount of data that negatively affect its performance.
-> It is like trying to fit in Oversized jeans. Unfortunately, this is one of the significant issues faced by machine learning professionals.
-> This means that the algorithm is trained with noisy and biased data, which will affect its overall performance.
-> Let’s understand this with the help of an example. Let’s consider a model trained to differentiate between a cat, a rabbit, a dog, and a tiger. The training data contains 1000 cats, 1000 dogs, 1000 tigers, and 4000 Rabbits. Then there is a considerable probability that it will identify the cat as a rabbit. In this example, we had a vast amount of data, but it was biased; hence the prediction was negatively affected.
We can tackle this issue by:
 Analyzing the data with the utmost level of perfection
 Use data augmentation technique

 Remove outliers in the training set
 Select a model with lesser features

Underfitting in Machine Learning

A statistical model or a machine learning algorithm is said to have underfitting when it cannot capture the underlying trend of the data, i.e., it only performs well on training data but performs poorly on testing data. (It's just like trying to fit undersized pants!) 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. In such cases, the rules of the machine learning model are too easy and flexible to be applied to such minimal data, and therefore the model will probably make a lot of wrong predictions. Underfitting can be avoided by using more data and also reducing the features by feature selection.

In a nutshell, Underfitting refers to a model that can neither performs well on the training data nor generalize to new data.

Reasons for Underfitting

1. High bias and low variance.
2. The size of the training dataset used is not enough.
3. The model is too simple.
4. Training data is not cleaned and also contains noise in it.

Techniques to Reduce Underfitting

1. Increase model complexity.
2. Increase the number of features, performing feature engineering.
3. Remove noise from the data.
4. Increase the number of epochs or increase the duration of training to get better results.

Overfitting in Machine Learning

A statistical model is said to be overfitted when the model does not make accurate predictions on testing data. When a model gets trained with so much data, it starts learning from the noise and inaccurate data entries in our data set. And when testing with test data results in High variance. Then the model does not categorize the data correctly, because of too many details and noise. The causes of overfitting are the non-parametric and non-linear methods because these types of machine learning algorithms have more freedom in building the model based on the dataset and therefore they can really build unrealistic models. A solution to avoid overfitting is using a linear algorithm if we have linear data or using the parameters like the maximal depth if we are using decision trees.
In a nutshell, Overfitting is a problem where the evaluation of machine learning algorithms on training data is different from unseen data.

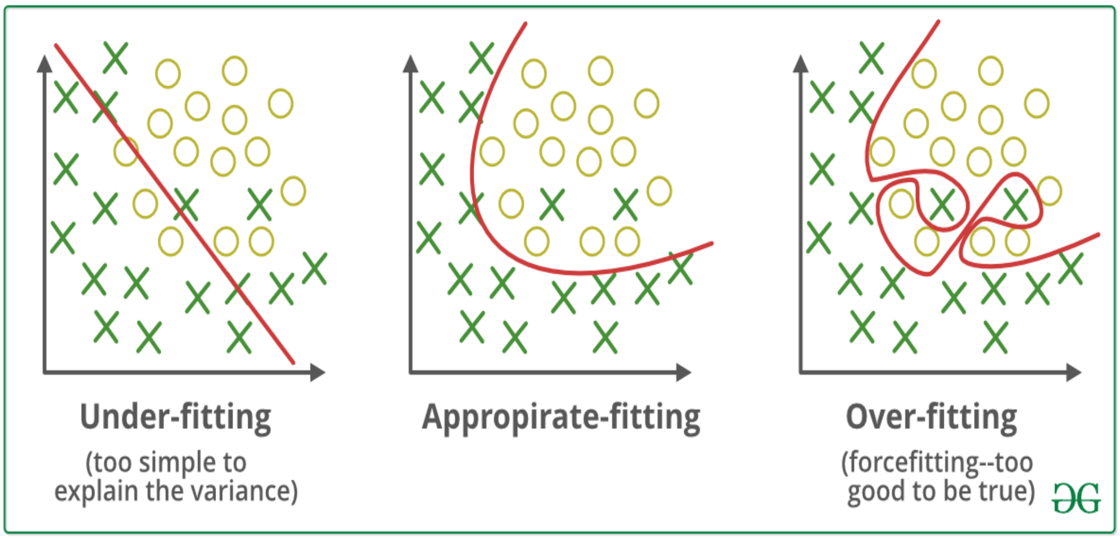
Reasons for Overfitting:

1. High variance and low bias.

2. The model is too complex.
3. The size of the training data.

Techniques to Reduce Overfitting

1. Increase training data.
2. Reduce model complexity.
3. Early stopping during the training phase (have an eye over the loss over the training period as soon as loss begins to increase stop training).
4. Ridge Regularization and Lasso Regularization.
5. Use dropout for neural networks to tackle overfitting.


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

Underfitting and Overfitting in Machine Learning