

# MACHINE LEARNING

## UNIT-1

### Introduction: What is Machine Learning

In the real world, we are surrounded by humans who can learn everything from their experiences with their learning capability, and we have computers or machines which work on our instructions. But can a machine also learn from experiences or past data like a human does? So here comes the role of **Machine Learning**.

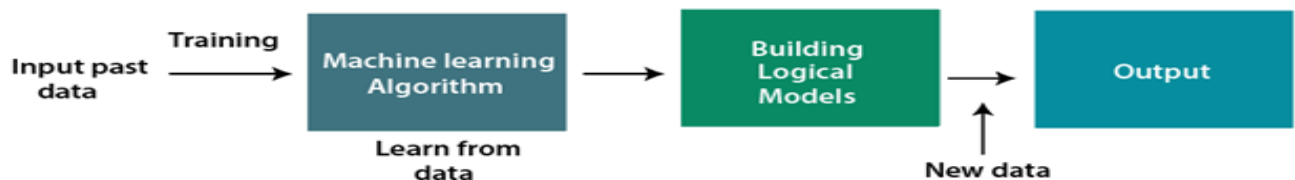
Machine Learning is said as a subset of **artificial intelligence** that is mainly concerned with the development of algorithms which allow a computer to learn from the data and past experiences on their own. The term machine learning was first introduced by **Arthur Samuel** in **1959**. We can define it in a summarized way as: Machine learning enables a machine to automatically learn from data, improve performance from experiences, and predict things without being explicitly programmed.

With the help of sample historical data, which is known as **training data**, machine learning algorithms build a **mathematical model** that helps in making predictions or decisions without being explicitly programmed. Machine learning brings computer science and statistics together for creating predictive models. Machine learning constructs or uses the algorithms that learn from historical data. The more we will provide the information, the higher will be the performance

### How does Machine Learning work

A Machine Learning system **learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it**. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.

Suppose we have a complex problem, where we need to perform some predictions, so instead of writing a code for it, we just need to feed the data to generic algorithms, and with the help of these algorithms, machine builds the logic as per the data and predict the output. Machine learning has changed our way of thinking about the problem. The below block diagram explains the working of Machine Learning algorithm:



## Features of Machine Learning:

- Machine learning uses data to detect various patterns in a given dataset.
- It can learn from past data and improve automatically.
- It is a data-driven technology.
- Machine learning is much similar to data mining as it also deals with the huge amount of the data.

## Need for Machine Learning

The need for machine learning is increasing day by day. The reason behind the need for machine learning is that it is capable of doing tasks that are too complex for a person to implement directly. As a human, we have some limitations as we cannot access the huge amount of data manually, so for this, we need some computer systems and here comes the machine learning to make things easy for us.

We can train machine learning algorithms by providing them the huge amount of data and let them explore the data, construct the models, and predict the required output automatically. The performance of the machine learning algorithm depends on the amount of data, and it can be determined by the cost function. With the help of machine learning, we can save both time and money.

The importance of machine learning can be easily understood by its uses cases, Currently, machine learning is used in **self-driving cars, cyber fraud detection, face recognition, and friend suggestion by Facebook**, etc. Various top companies such as Netflix and Amazon have build machine learning models that are using a vast amount of data to analyze the user interest and recommend product accordingly.

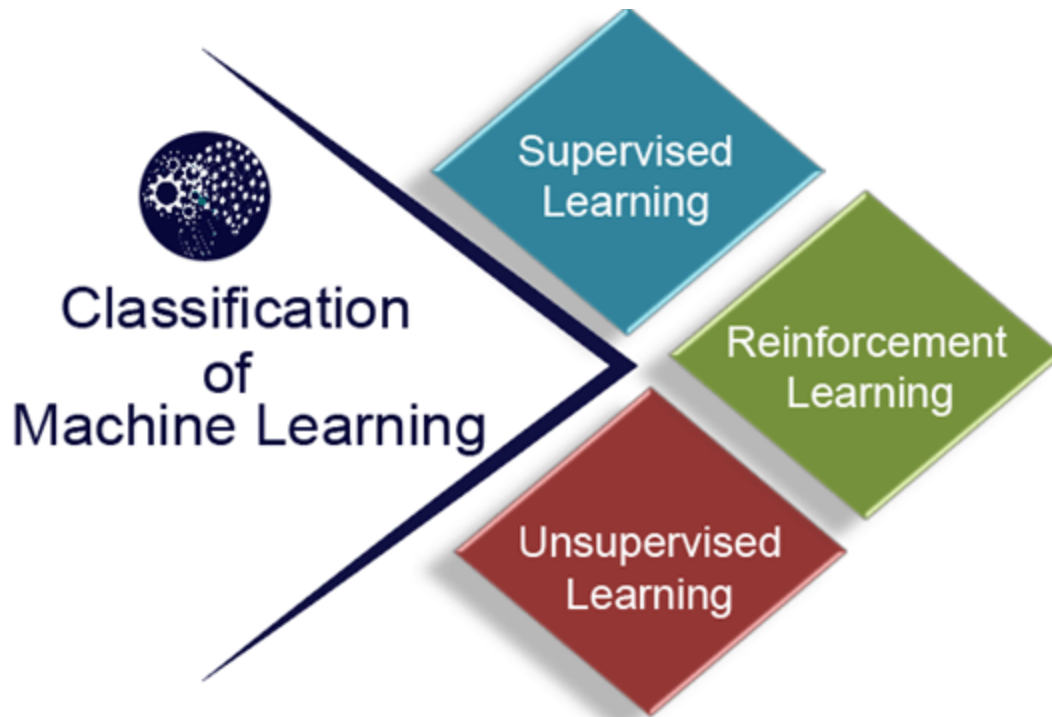
## Following are some key points which show the importance of Machine Learning:

- Rapid increment in the production of data
- Solving complex problems, which are difficult for a human
- Decision making in various sector including finance
- Finding hidden patterns and extracting useful information from data.

## Classification of Machine Learning

At a broad level, machine learning can be classified into three types:

1. Supervised learning
2. Unsupervised learning
3. Reinforcement learning



## 1) Supervised Learning

Supervised learning is a type of machine learning method in which we provide sample labeled data to the machine learning system in order to train it, and on that basis, it predicts the output.

The system creates a model using labeled data to understand the datasets and learn about each data, once the training and processing are done then we test the model by providing a sample data to check whether it is predicting the exact output or not.

The goal of supervised learning is to map input data with the output data. The supervised learning is based on supervision, and it is the same as when a student learns things in the supervision of the teacher. The example of supervised learning is **spam filtering**.

Supervised learning can be grouped further in two categories of algorithms:

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

## 2) Unsupervised Learning:

**Unsupervised learning is a learning method in which a machine learns without any supervision. The training is provided to the machine with the set of data that has not been labeled, classified, or categorized, and the algorithm needs to act on that data without any supervision. The goal of unsupervised learning is to restructure the input data into new features or a group of objects with similar patterns.**

In unsupervised learning, we don't have a predetermined result. The machine tries to find useful insights from the huge amount of data. It can be further classified into two categories of algorithms:

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

## 3) Reinforcement Learning

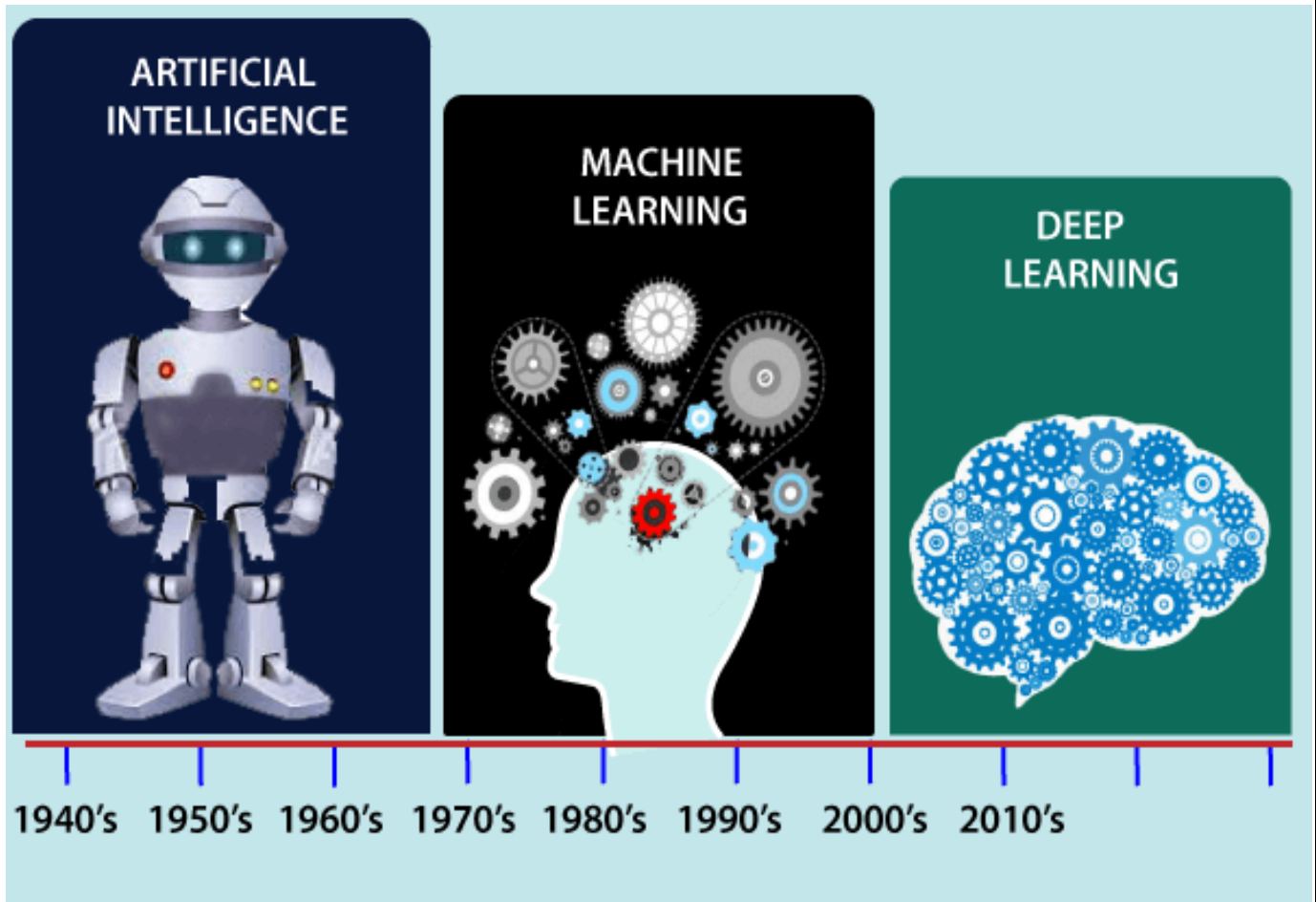
Reinforcement learning is a feedback-based learning method, in which a learning agent gets a reward for each right action and gets a penalty for each wrong action. The agent learns automatically with these feedbacks and improves its performance. In reinforcement learning, the agent interacts with the environment and explores it. The goal of an agent is to get the most reward points, and hence, it improves its performance.

The robotic dog, which automatically learns the movement of his arms, is an example of Reinforcement learning.

***Note:** We will learn about the above types of machine learning in detail in later chapters.*

## History of Machine Learning

Before some years (about 40-50 years), machine learning was science fiction, but today it is the part of our daily life. Machine learning is making our day to day life easy from **self-driving cars** to **Amazon virtual assistant "Alexa"**. However, the idea behind machine learning is so old and has a long history. Below some milestones are given which have occurred in the history of machine learning:



### The early history of Machine Learning (Pre-1940):

- **1834:** In 1834, Charles Babbage, the father of the computer, conceived a device that could be programmed with punch cards. However, the machine was never built, but all modern computers rely on its logical structure.
- **1936:** In 1936, Alan Turing gave a theory that how a machine can determine and execute a set of instructions.

### The era of stored program computers:

- **1940:** In 1940, the first manually operated computer, "ENIAC" was invented, which was the first electronic general-purpose computer. After that stored program computer such as EDSAC in 1949 and EDVAC in 1951 were invented.
- **1943:** In 1943, a human neural network was modeled with an electrical circuit. In 1950, the scientists started applying their idea to work and analyzed how human neurons might work.

## Computer machinery and intelligence:

- **1950:** In 1950, Alan Turing published a seminal paper, "**Computer Machinery and Intelligence**," on the topic of artificial intelligence. **In his paper, he asked, "Can machines think?"**

## Machine intelligence in Games:

- **1952:** 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.
- **1959:** In 1959, the term "Machine Learning" was first coined by **Arthur Samuel**.

**Well Posed Learning Problem** – A computer program is said to learn from experience E in context to some task T and some performance measure P, if its performance on T, as was measured by P, upgrades with experience E.

Any problem can be segregated as well-posed learning problem if it has three traits –

- Task
- Performance Measure
- Experience

**Certain examples that efficiently defines the well-posed learning problem are –**

### **1. To better filter emails as spam or not**

- Task – Classifying emails as spam or not
- Performance Measure – The fraction of emails accurately classified as spam or not spam
- Experience – Observing you label emails as spam or not spam

### **2. A checkers learning problem**

- Task – Playing checkers game
- Performance Measure – percent of games won against opposer
- Experience – playing implementation games against itself

### **3. Handwriting Recognition Problem**

- Task – Acknowledging handwritten words within portrayal
- Performance Measure – percent of words accurately classified
- Experience – a directory of handwritten words with given classifications

### **4. A Robot Driving Problem**

- Task – driving on public four-lane highways using sight scanners
- Performance Measure – average distance progressed before a fallacy
- Experience – order of images and steering instructions noted down while observing a human driver

### **5. Fruit Prediction Problem**

- Task – forecasting different fruits for recognition
- Performance Measure – able to predict maximum variety of fruits
- Experience – training machine with the largest datasets of fruits images

### **6. Face Recognition Problem**

- Task – predicting different types of faces
- Performance Measure – able to predict maximum types of faces
- Experience – training machine with maximum amount of datasets of different face images

### **7. Automatic Translation of documents**

- Task – translating one type of language used in a document to other language

- Performance Measure – able to convert one language to other efficiently
- Experience – training machine with a large dataset of different types of languages

### **Designing a Learning System in Machine Learning :**

According to Tom Mitchell, “A computer program is said to be learning from experience (E), with respect to some task (T). Thus, the performance measure (P) is the performance at task T, which is measured by P, and it improves with experience E.”

**Example:** In Spam E-Mail detection,

- **Task, T:** To classify mails into Spam or Not Spam.
- **Performance measure, P:** Total percent of mails being correctly classified as being “Spam” or “Not Spam”.
- **Experience, E:** Set of Mails with label “Spam”

**Steps for Designing Learning System are:**

**Step 1) Choosing the Training Experience:** The very important and first task is to choose the training data or training experience which will be fed to the Machine Learning Algorithm. It is important to note that the

data or experience that fed to the algorithm must have a significant impact on the Success or Failure of the Model. So Training data or experience should be chosen wisely.

Below are the attributes which will impact on Success and Failure of Data:

- The training experience will be able to provide direct or indirect feedback regarding choices. For example: While Playing chess the training data will provide feedback to itself like instead of this move if this is chosen the chances of success increases.
- Second important attribute is the degree to which the learner will control the sequences of training examples. For example: when training data is fed to the machine then at that time accuracy is very less but when it gains experience while playing again and again with itself or opponent the machine algorithm will get feedback and control the chess game accordingly.
- Third important attribute is how it will represent the distribution of examples over which performance will be measured. For example, a Machine learning algorithm will get experience while going through a number of different cases and different examples. Thus, Machine Learning Algorithm will get more and more experience by passing through more and more examples and hence its performance will increase.

**Step 2- Choosing target function:** The next important step is choosing the target function. It means according to the knowledge fed to the algorithm the machine learning will choose NextMove function which will describe what type of legal moves should be taken. For example : While playing chess with the opponent, when opponent will play then the machine learning algorithm will decide what be the number of possible legal moves taken in order to get success.

**Step 3- Choosing Representation for Target function:** When the machine algorithm will know all the possible legal moves the next step is to choose the optimized move using any representation i.e. using linear Equations, Hierarchical Graph Representation, Tabular form etc. The NextMove function will move the Target move like out of these move which will provide more success rate. For Example : while playing chess machine have 4 possible moves, so the machine will choose that optimized move which will provide success to it.

**Step 4- Choosing Function Approximation Algorithm:** An optimized move cannot be chosen just with the training data. The training data had to go through with set of example and through these examples the training data will approximates which steps are chosen and after that machine will

provide feedback on it. For Example : When a training data of Playing chess is fed to algorithm so at that time it is not machine algorithm will fail or get success and again from that failure or success it will measure while next move what step should be chosen and what is its success rate.

**Step 5- Final Design:** The final design is created at last when system goes from number of examples , failures and success , correct and incorrect decision and what will be the next step etc. Example: DeepBlue is an intelligent computer which is ML-based won chess game against the chess expert Garry Kasparov, and it became the first computer which had beaten a human chess expert.

## **Perspectives and issues in machine learning:**

### **PERSPECTIVES:**

- Machine learning is a subfield of artificial intelligence and machine learning algorithms are used in other related fields like natural language processing and computer vision.
- In general, there are three types of learning and these are supervised learning, unsupervised learning, and reinforcement learning.
- Their names tell the main idea behind them actually.
- In supervised learning, your system learns under the supervision of the data outputs so supervised algorithms are preferred if your dataset contains output information.
- Let me give you an example in there.

Let's assume you have a medical statistic company and you have a dataset which contains patients' features like blood pressure, sugar rate in their blood, heart rate per minute, etc.

### **ISSUES:**

#### **LACK OF QUALITY DATA**

- One of the main issues in Machine Learning is the absence of good data.
- While, algorithms tend to make developers exhaust most of their time on artificial intelligence.

#### **FAULT IN CREDIT CARD FRAUD DETECTION**

- Although this AI-driven software helps to successfully detect credit card fraud, there are issues in Machine Learning that make the process redundant.

#### **GETTING BAD RECOMMENDATIONS**

- Proposal engines are quite regular today.
- While some might be dependable, others may not appear to provide the necessary results.

#### **TALENT DEFICIT**

- Albeit numerous individuals are pulled into the ML business, however, there are still not many experts who can take complete control of this innovation.

#### **MAKING THE WRONG ASSUMPTIONS**

- ML models can't manage datasets containing missing data points.
- Thus, highlights that contain a huge part of missing data should be erased.

Concept learning and general to specific ordering:

Introduction: If we really wanted to simplify the whole story behind "Learning" in machine learning, we could say that a machine learning algorithm strives to learn a **general function** out of a given **limited training examples**. In general, we can think of concept learning as a search problem. The learner searches through a space of hypotheses (we will explain what they are), to find the best one. Which one would be the best one? The answer is the one that **fits** the training examples the best. And while searching and trying different hypotheses, we would hope for the learner to eventually **converge** to the correct hypothesis. This convergence is the same idea as **Learning!** Now, let's go deeper.



### A concept learning task:

Taking a very simple example, one possible target concept may be to *Find the day when my friend Ramesh enjoys his favorite sport*. We have some attributes/features of the day like, *Sky, Air Temperature, Humidity, Wind, Water, Forecast* and based on this we have a target Concept named **EnjoySport**.

We have the following training example available:

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

Let's Design the problem formally with **TPE**(Task, Performance, Experience

**Problem:** Learning the day when Ramesh enjoys the sport.

**Task T:** Learn to predict the value of **EnjoySport** for an arbitrary day, based on the values of the attributes of the day.

**Performance measure P:** Total percent of days (EnjoySport) correctly predicted.

**Training experience E:** A set of days with given labels (EnjoySport: Yes/No)

Let us take a very simple hypothesis representation which consists of a **conjunction** of constraints in the instance attributes. We get a hypothesis  $h_i$  with the help of example  $i$  for our training set as below:

$h_i(x) := \langle x_1, x_2, x_3, x_4, x_5, x_6 \rangle$

where  $x_1, x_2, x_3, x_4, x_5$  and  $x_6$  are the values of **Sky, AirTemp, Humidity, Wind, Water** and **Forecast**.

Hence  $h_1$  will look like (the first row of the table above):

$h_1(x=1): \langle \text{Sunny, Warm, Normal, Strong, Warm, Same} \rangle$  *Note:  $x=1$  represents a positive hypothesis / Positive example*

We want to find the most suitable hypothesis which can represent the concept. For example, Ramesh enjoys his favorite sport only on **cold days** with **high humidity** (This seems independent of the values of the other attributes present in the training examples).

$h(x=1) = \langle ?, \text{Cold, High, }, ?, ? \rangle$

Here ? indicates that any value of the attribute is acceptable.

**Note:** The most generic hypothesis will be  $\langle ?, ?, ?, ?, ? \rangle$  where every day is a positive example and the most specific hypothesis will be  $\langle ?, ?, ?, ?, ? \rangle$  where no day is a positive example.

Concept learning search: What is concept learning? In terms of machine learning, the concept learning can be formulated as “**Problem of searching through a predefined space of potential hypotheses for the hypothesis that best fits the training examples**”

### Find-S Algorithm:

Following are the steps for the Find-S algorithm:

1. Initialize **h** to the most specific hypothesis in **H**
2. For each positive training example,
  1. For each attribute, constraint **ai** in **h**
    1. If the constraints **ai** is satisfied by **x**
    2. Then do nothing
    3. Else replace **ai** in **h** by the next more general constraint that is satisfied by **x**
3. Output hypothesis **h**

### Version space and candidate elimination algorithm:

The candidate elimination algorithm incrementally builds the version space given a hypothesis space **H** and a set **E** of examples. The examples are added one by one; each example possibly shrinks the version space by removing the hypotheses that are inconsistent with the example. The candidate elimination algorithm does this by updating the general and specific boundary for each new example.

- You can consider this as an extended form of Find-S algorithm.
- Consider both positive and negative examples.
- Actually, positive examples are used here as Find-S algorithm (Basically they are generalizing from the specification).
- While the negative example is specified from generalize form.

#### Terms Used:

- **Concept learning:** Concept learning is basically learning task of the machine (Learn by Train data)
- **General Hypothesis:** Not Specifying features to learn the machine.
- **G** = {‘?’ , ‘?’ , ‘?’ , ‘?’ ...}: Number of attributes
- **Specific Hypothesis:** Specifying features to learn machine (Specific feature)
- **S** = {‘pi’ , ‘pi’ , ‘pi’ ...}: Number of pi depends on number of attributes.
- **Version Space:** It is intermediate of general hypothesis and Specific hypothesis. It not only just written one hypothesis but a set of all possible hypothesis based on training data-set.

#### Algorithm:

**Step1:** Load Data set

**Step2:** Initialize General Hypothesis and Specific Hypothesis.

**Step3:** For each training example

**Step4:** If example is positive example

if attribute\_value == hypothesis\_value:

Do nothing

else:

replace attribute value with '?' (Basically generalizing it)

**Step5:** If example is Negative example

Make generalize hypothesis more specific.

**Example:**

**Consider the dataset given below:**

**Algorithmic steps:**

**Initially :**  $G = [[?, ?, ?, ?, ?], [?, ?, ?, ?, ?], [?, ?, ?, ?, ?], [?, ?, ?, ?, ?], [?, ?, ?, ?, ?]]$   
 $S = [Null, Null, Null, Null, Null]$

**For instance 1 :** <'sunny','warm','normal','strong','warm ','same'> and positive output.

$G1 = G$

$S1 = ['sunny','warm','normal','strong','warm ','same']$

**For instance 2 :** <'sunny','warm','high','strong','warm ','same'> and positive output.

$G2 = G$

$S2 = ['sunny','warm','?','strong','warm ','same']$

**For instance 3 :** <'rainy','cold','high','strong','warm ','change'> and negative output.

$G3 = [['sunny', ?, ?, ?, ?], [?, 'warm', ?, ?, ?], [?, ?, ?, ?, ?],$

$[?, ?, ?, ?, ?], [?, ?, ?, ?, ?], [?, ?, ?, ?, 'same']]$

$S3 = S2$

**For instance 4 :** <'sunny','warm','high','strong','cool','change'> and positive output.

$G4 = G3$

$S4 = ['sunny','warm','?','strong', ?, ?]$

At last, by synchronizing the  $G4$  and  $S4$  algorithm produce the output.

**Output :**

$G = [['sunny', ?, ?, ?, ?], [?, 'warm', ?, ?, ?]]$

$S = ['sunny','warm','?','strong', ?, ?]$

**Remarks on version space and candidate elimination algorithm:**

The version space learned by the CANDIDATE-ELIMINATION algorithm will converge toward the hypothesis that correctly describes the target concept, provided (1) there are no errors in the training examples, and (2) there is some hypothesis in  $H$  that correctly describes ...

Inductive bias: In machine learning, the term *inductive bias* refers to a set of (explicit or implicit) assumptions made by a learning algorithm in order to perform induction, that is, to generalize a finite set of observation (training data) into a general model of the domain. Without a bias of that kind, induction would not be possible, since the observations can normally be generalized in many ways. Treating all these possibilities in equally, i.e., without any bias in the sense of a preference for specific types of generalization (reflecting background knowledge about the target function to be learned), predictions for new situations could not be made.

More specifically, consider the problem of learning a mapping (model)  $f \in \mathcal{F} = \mathcal{Y}^{\mathcal{X}}$

## Decision Tree Learning:

- Introduction: Decision tree algorithm falls under the category of supervised learning. They can be used to solve both regression and classification problems.
- Decision tree uses the tree representation to solve the problem in which each leaf node corresponds to a class label and attributes are represented on the internal node of the tree.
- We can represent any boolean function on discrete attributes using the decision tree.
- 

### Decision tree representation:

- 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.

### 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.

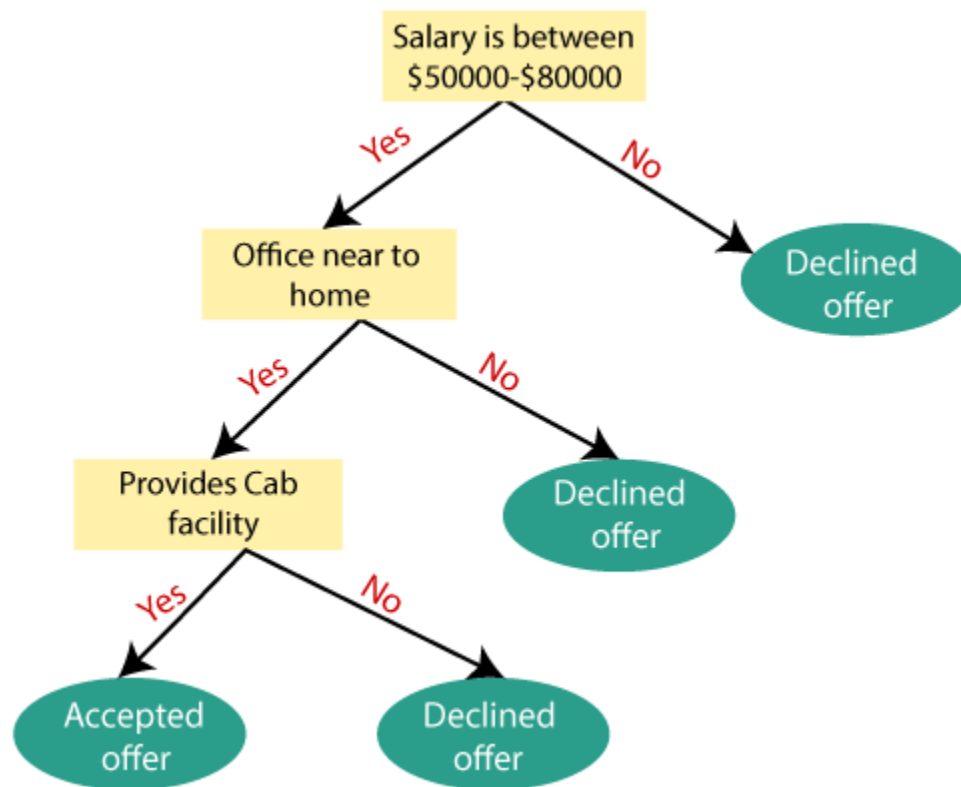
- **Leaf Node:** Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
- **Splitting:** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
- **Branch/Sub Tree:** A tree formed by splitting 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: Machine Learning - Observational and Experimental Studies

- **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:



#### 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 over fitting issue, which can be resolved using the **Random Forest algorithm**.
- For more class labels, the computational complexity of the decision tree may increase.

#### Appropriate problems for decision tree learning:

Although a variety of decision tree learning methods have been developed with somewhat differing capabilities and requirements, decision tree learning is generally best suited to problems with the following characteristics:

**1. Instances are represented by attribute-value pairs.**

“Instances are described by a fixed set of attributes (e.g., Temperature) and their values (e.g., Hot). The easiest situation for decision tree learning is when each attribute takes on a small number of disjoint possible values (e.g., Hot, Mild, Cold). However, extensions to the basic algorithm allow handling real-valued attributes as well (e.g., representing Temperature numerically).”

**2. The target function has discrete output values.**

“The decision tree is usually used for Boolean classification (e.g., **yes** or **no**) kind of example. Decision tree methods easily extend to learning functions with more than two possible output values. A more substantial extension allows learning target functions with real-valued outputs, though the application of decision trees in this setting is less common.”

**3. Disjunctive descriptions may be required.**

Decision trees naturally represent disjunctive expressions.

**4. The training data may contain errors.** “Decision tree learning methods are robust to errors, both errors in classifications of the training examples and errors in the attribute values that describe these examples.”

**5. The training data may contain missing attribute values.**

“Decision tree methods can be used even when some training examples have unknown values (e.g., if the **Humidity** of the day is known for only some of the training examples).”

The basic decision tree learning algorithm:

Decision tree learning is a type of supervised machine learning this decision tree algorithm classification and regression algorithm mostly it is following classification algorithm.

Algorithm:

- 1.function d-tree learning
- 2.inputs-collection of examples, collection of attributes ,default values.
- 3.if ‘there is no example’;then return default value.
- 4.else-‘the categorization is same for all examples’ then return the categorization.
- 5.else if-‘there is no attribute’.then the return MAX-val.
- 6.else best-SELECT\_ATTRIBUTE tree-A new d-tree with root text best M-Max-val examples  
Subtree- D-TREE\_LEARNING.

Hypothesis space search in decision tree learning:

1. ID3 searches the space of possible decision trees: doing hill-climbing on information gain.
2. It searches the complete space of all finite discrete-valued functions. ...
3. It maintains only one hypothesis (unlike Candidate-Elimination). ...
4. It does not do back tracking.
5. It can be extended easily to noisy data also.
6. ID3 cannot determine alternative decision tree.
7. ID3 has only single current hypothesis.
8. ID3 algorithm avoids major risk of searching incomplete hypothesis.
9. First start with empty tree and keep on adding nodes.
10. ID3 algorithm performs simple to complex searching.

**Inductive bias in decision tree learning:**

The inductive bias (also known as learning bias) of a learning algorithm is the set of assumptions that the learner uses to predict outputs of given inputs that it has not encountered. In machine learning, one aims to construct algorithms that are able to learn to predict a certain target output.

1. from examples we derive the rules.

2. inductive bias of ID3 consist of describing the basis by which ID3 chooses one consist decision tree overall the possible decision trees.

ID3 searching strategies:

Two types they are

1. select infavour of shorter trees over longer trees.

2. selects element with highest information gaining as root attribute over lowest information gaining ones.

Types of inductive bias:

1. restrictive bias

It will works based on condition

Eg: version space and candidate elimination algorithm.

2. preference bias

It will works based on priorities.

Eg: decision tree or ID3 algorithm.

Issues in decision tree learning:

1. Overfitting the data

2. incorporating continues valued attributes.

3. determining the depth search tree for correct classification

4. handling attributes with different costs.

5. alternative measures for selecting attributes.

Example of decision tree construct a loan system.