Subject: Artificial Intelligence

Note: UNIT VI: Learning

Machine learning introduction, Types of Learning (Rote, Direct instruction Analogy, Induction, Deduction), Knowledge and learning, Learning in Problem Solving, learning from example, learning probabilistic models, Formal Learning Theory, k-nearest neighbor, Decision Tree Learning, Recent applications of machine learning.

What is learning?

- According to **Herbert Simon**, learning denotes changes in a system that enable a system to do the same task more efficiently the next time.
- **Arthur Samuel stated that**, "Machine learning is the subfield of computer science, that gives computers the ability to learn without being explicitly programmed ".
- In 1997, **Mitchell** proposed that, " A computer program is said to learn from experience **'E'** with respect to some class of tasks **'T'** and performance measure **'P'**, if its performance at tasks in **'T'**, as measured by **'P'**, improves with experience **E** ".
- The main purpose of machine learning is to study and design the algorithms that can be used to produce the predicates from the given dataset.
- Besides these, the machine learning includes the agents percepts for acting as well as to improve their future performance.

The following tasks must be learned by an agent.

- To predict or decide the result state for an action.
- To know the values for each state(understand which state has high or low vale).
- To keep record of relevant percepts.

Why do we require machine learning?

- Machine learning plays an important role in improving and understanding the efficiency of human learning.
- Machine learning is used to discover a new things not known to many human beings.

What is rote learning

Rote learning is a memorization technique based on repetition. The method rests on the premise that the recall of repeated material becomes faster the more one repeats it. Some of the alternatives to rote learning include meaningful learning, associative learning, spaced repetition and active learning.

The meaning of rote in ‘rote learning’ itself means learning by repetition. The process of repeating something over and over engages the short-term memory and allows us to quickly remember basic things like facts, dates, names, multiplication tables, etc. It differs from other forms of learning in that it doesn't require the learner to carefully think about something, and is rather dependent on the act of repetition itself.

Even though complete and holistic learning is not dependent on rote learning techniques alone, they do allow students to quickly recall basic facts and laws and master foundational knowledge of a topic in students. Some examples of rote learning in schools can be found in the following:

- Repeating words to instil them in your vocabulary.
- Learning scales in music.
- Memorizing the periodic table.
- Learning the basic laws and formulae in physics and sundry sciences.
- Learning statutes and cases in law.

Having to memorize the basic facts and principles of a field is an important prerequisite to later analyze and study them. This is where rote learning techniques come in handy and allow you to remember the building blocks of concepts without having to dive deep into them.

Rote learning techniques

Rote learning techniques are aplenty, and they all require time and effort in repetition. The more you repeat for longer periods, the easier it will be to recall. Even if you only have a few hours to memorize something, the following rote learning techniques will help you remember quickly:

Read it aloud - Read the text out loud with understanding. You can even try it before a mirror, ask a friend to listen to you, or read it out just under your breath. You can experiment with how slow or fast you want to read, how expressive you want to be, and internalize the rhythm of the text. Auditory learners will greatly benefit from this rote learning technique.

Write it down - Writing down the text information after reading is one of the best rote learning techniques. Doing so will help identify difficult passages and areas that need more practice. If you're preparing for a written exam, this kinesthetic rote technique will serve as a rehearsal and commit the information for easy retrieval.

Sing it out - There's a reason why songs commit to memory easier than text that is spoken simply and without any variation in pitch. So try putting the text that you want to learn with a melody that you like. Or, if you're feeling creative, come up with your own catchy tune to remember the text.

Visualize - Humans are visual creatures and our brains are wired to remember things better with images. For every line and connected phrase, come up with ways to visualize it and remember it. The memory palace can be a useful trick for such rote learning techniques.

Free association - Free association is one of the more interesting rote learning techniques, and a very useful way of remembering things quickly, especially if they are too messy for the traditional rote learning techniques. The main idea of this method is to combine new information with what you already know in a fun and personal way. For instance, if you're learning the 'Circle of Fifths' in music, you can associate each note to the numbers on the clock, one for each of the 12 notes in music. You are free to form your own associations as you see fit, as long as it helps you to recall the information.

Advantages of rote learning techniques

Rote learning is considered useful for a variety of reasons. Here are a few:

- Rote learning requires very little analysis.
- With rote, one can remember just about anything over time and repetition.
- Rote learning allows one to recall information wholly, and even to retain it for life.
- Rote learning makes it easier for people to score who find it difficult to understand or master reading and maths concepts.
- Rote learning can help improve short-term memory.

Disadvantages of rote learning techniques

On the other hand, there are a few drawbacks of rote learning that you need to be aware of as well.

- The repetitive nature of rote learning can become dull.
- One can easily lose focus while rote learning.
- Rote learning is not holistic.
- There is no connection between new and old information with rote learning.
- Rote learning doesn't lead to a deeper understanding of the information.

Main Points:

Example: Factorial 5!, Tables, Cookies, Caches,

- Rote learning is possible on the basis of memorization.
- This technique mainly focuses on memorization by avoiding the inner complexities.
  So, it becomes possible for the learner to recall the stored knowledge.
  For example: When a learner learns a poem or song by reciting or repeating it, without knowing the actual meaning of the poem or song.

Induction learning (Learning by example).

- Induction learning is carried out on the basis of supervised learning.

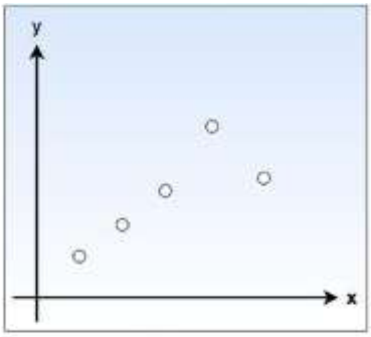
- In this learning process, a general rule is induced by the system from a set of observed instance.
- However, class definitions can be constructed with the help of a classification method.

  For Example:

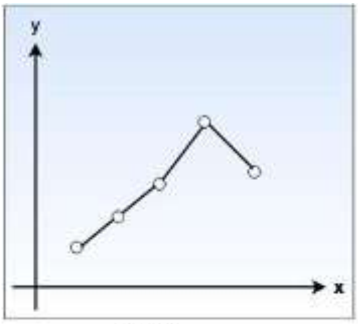
  Consider that **'*f*'** is the target function and example is a pair (x *f*(x)), where 'x' is input and *f*(x) is the output function applied to 'x'.

  Given problem: Find hypothesis h such as h ≈ *f*

- So, in the following fig-a, points (x,y) are given in plane so that y = *f*(x), and the task is to find a function h(x) that fits the point well.

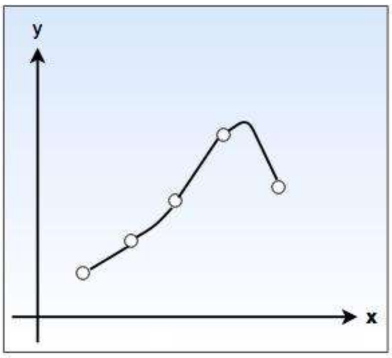


Fig(a)



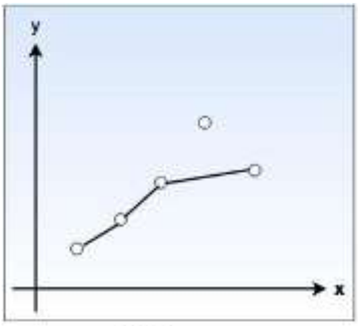
Fig(b)

- In fig-b, a piecewise-linear 'h' function is given, while the fig-c shows more complicated 'h' function.



Fig(c)

- Both the functions agree with the example points, but differ with the values of 'y' assigned to other x inputs.



Fig(d)

- As shown in fig.(d), we have a function that apparently ignores one of the example points, but fits others with a simple function. The true/ is unknown, so there are many choices for h, but without further knowledge, we have no way to prefer (b), (c), or (d).

Learning by taking advice

- This type is the easiest and simple way of learning.
- In this type of learning, a programmer writes a program to give some instructions to perform a task to the computer. Once it is learned (i.e. programmed), the system will be able to do new things.
- Also, there can be several sources for taking advice such as humans(experts), internet etc.
- However, this type of learning has a more necessity of inference than rote learning.
- As the stored knowledge in knowledge base gets transformed into an operational form, the reliability of the knowledge source is always taken into consideration.
- Request----Interpret----Operational----Integrate---Evaluate

Explanation based learning

- Explanation-based learning (EBL) deals with an idea of single-example learning.
- This type of learning usually requires a substantial number of training instances but there are two difficulties in this:
  I. it is difficult to have such a number of training instances
  ii. Sometimes, it may help us to learn certain things effectively, specially when we have enough knowledge.
  Hence, it is clear that instance-based learning is more data-intensive, data-driven while EBL is more knowledge-intensive, knowledge-driven.
- Initially, an EBL system accepts a training example.
- On the basis of the given goal concept, an operationality criteria and domain theory, it "generalizes" the training example to describe the goal concept and to satisfy the

operationality criteria (which are usually a set of rules that describe relationships between objects and actions in a domain).

- Thus, several applications are possible for the knowledge acquisition and engineering aspects.

Intuition:

The objective of EBL is to understand the essential properties of a particular concept. So, we need to find out what makes an example, part of a particular concept. Unlike FOIL algorithm, here we focus on the one example instead of collecting multiple examples.

The ability to explain single examples is known as **"Domain Theory".**
An EBL accepts 4 kinds of input:

i) A training example: what the learning model sees in the world.
ii) A goal concept: a high level description of what the model is supposed to learn.
iii) A operational criterion: states which other terms
can appear in the generalized result.
iv) A domain theory: set of rules that describe relationships between objects and actions in a domain.
From the above 4 parameters, EBL uses the domain theory to find that training example, that best describes the goal concept while abiding by the operational criterion and keeping our justification as general as possible.

EBL involves 2 steps:

1. **Explanation** — The domain theory is used to eliminate all the unimportant training example while retaining the important ones that best describe the goal concept.
2. **Generalization** — The explanation of the goal concept is made as general and widely applicable as possible. This ensures that all cases are covered, not just certain specific ones.

EBL Architecture:

- **EBL model during training**
    - During training, the model generalizes the training example in such a way that all scenarios lead to the Goal Concept, not just in specific cases. **(As shown in Fig 1)**

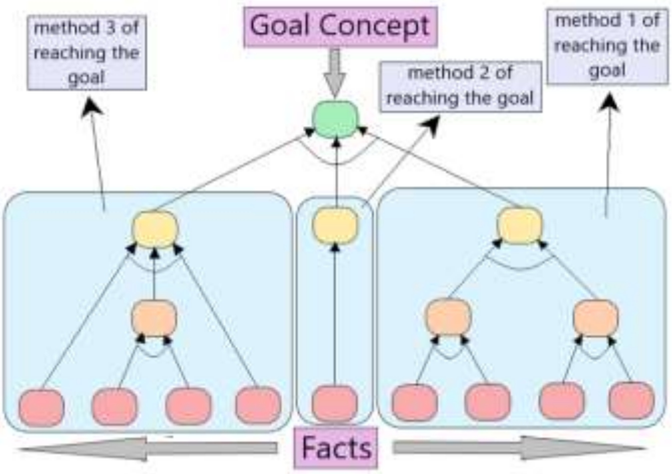


Fig 1 : Training EBL Model

-

Learning in Problem Solving

- Humans have a tendency to learn by solving various real world problems.

- The forms or representation, or the exact entity, problem solving principle is based on reinforcement learning.
- Therefore, repeating certain action results in desirable outcome while the action is avoided if it results into undesirable outcomes.
- As the outcomes have to be evaluated, this type of learning also involves the definition of a utility function. This function shows how much is a particular outcome worth?
- There are several research issues which include the identification of the learning rate, time and algorithm complexity, convergence, representation (frame and qualification problems), handling of uncertainty (ramification problem), adaptivity and "unlearning" etc.
- In reinforcement learning, the system (and thus the developer) know the desirable outcomes but does not know which actions result into desirable outcomes.
- In such a problem or domain, the effects of performing the actions are usually compounded with side-effects. Thus, it becomes impossible to specify the actions to be performed in accordance to the given parameters.
- Q-Learning is the most widely used reinforcement learning algorithm.
- The main part of an algorithm is a simple value iteration update. For each state 'S', from the state set S, and for each action, a, from the action set 'A', it is possible to calculate an update to its expected reduction reward value, with the following expression:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t (s_t, a_t) [r_t + \gamma \max_a Q (s_{t+1}, a) - Q(s_t, a_t)]$$

- where r_t is a real reward at time t, $\alpha_t(s,a)$ are the learning rates such that $0 \leq \alpha_t(s,a) \leq 1$, and γ is the discount factor such that $0 \leq \gamma < 1$.

What Is Discovery Learning?

To explain what discovery learning is, it helps to pinpoint its opposite: direct instruction. In direct instruction, teachers lecture students in a classroom, giving them the information they need to know about a topic.

On the other hand, Discovery learning is an approach to education in which students learn the material from their interests, experiences, and exploration.

This is opposed to the idea of direct instruction which is the traditional method of teaching. Direct instruction involves teachers giving students specific discrete facts or learning objectives instead.

What Are the Types of Discovery Learning?

1. Pure Discovery Learning

Pure discovery learning is characterized by the techniques used. Essentially, these are techniques that include no direct assistance at all, other than the encouragement given to learners by the teacher.

This type of discovery learning is passive and can be compared to the type of learning that takes place when you go on a family trip and tour a museum.

You basically stand around and absorb what you can from a variety of methods such as reading the information, viewing pictures, and observing demonstrations.

However, you learn primarily by paying attention to what is going on around you. So, in pure discovery learning, students have to discover the rules and concepts on their own, with little to no assistance from a teacher.

A teacher merely provides a problem to the learners that they have not seen before. The teacher can then encourage them to work toward solving the problem on their own. They will do that with no explanation of how to solve the problem.

2. Expositional Discovery Learning

Expositional discovery is not a passive method where students are simply listening or reading; rather, expositional discovery involves hands-on activities such as conducting an experiment to test a hypothesis or building a model to test an idea. However, with maximal help from the teacher.

The teacher provides learners with some initial knowledge, then through a structured activity enables them to come to an understanding of a related principle.

In this type of learning, the instructor determines the procedures that should be followed to arrive at the solution to a problem. The expositional approach is often called expository teaching.

Generally, this type of learning is similar to Direct Instruction because the students or learners will eventually not make any discoveries themselves.

The teacher helps them with all they need and even what they will learn in the end. Therefore, students have no or little discovery to make.

3. Guided Discovery Learning

Guided Discovery Learning which is also known as directed discovery, is a type of learning where the teacher guides the students to solve problems, complete tasks, and make meaningful associations.

The teacher provides clues and information, but the learner makes discoveries on their own.

Guided discovery learning focuses on the process of problem-solving with the teacher acting in a supportive role. This type of learning is often used in advanced placement courses that focus on authentic learning, such as AP Physics or AP literature.

The teacher may point out specific details that students may not have noticed on their own. For example, the teacher might ask, “What’s different in this picture than in the previous one?” Or they might say, “Look at the title of these books. What do you imagine they’re about?”

In my opinion, it is a worthwhile method of achieving your goals in teaching and learning, especially if you are embarking on this path for the first time.

Guided Discovery provides you with more structure rather than using Trial and Error.

What Are the Benefits of Discovery Learning?

1. Increases engagement

Discovery learning is a way to engage this curiosity in the classroom, and let children explore on their own. Instead of giving them the answers, you give them problems, and let them work out the solutions. It’s an effective way to help kids learn math and other subjects.

Have your students talk through their approaches to equations, discuss their previous knowledge, and use it to solve problems (instead of telling them how).

Even better, make math a communal activity. Making math social enhances engagement and motivation while completing mathematical activities and projects.

When working in groups, however, the possibility of a single student dominating the debate cannot be discounted.

2. Promotes autonomy and independence

One of the main benefits of discovery learning is that it allows students to have a greater role in their own education.

Through discovery learning, students become more independent and self-motivated. Discovery learning also encourages your students to think critically.

This keeps them further engaged and encourages them. It is their responsibility to master each level before moving forward.

To engage in problem-solving, even when working in groups or participating in math talks, students must first think independently and creatively.

They must take information from multiple sources and formulate conclusions based on their findings. With practice, the skills learned through discovery learning can translate into a student's personal life and professional career.

3. It motivates students to learn

Discovery learning through hands-on projects offers a way to motivate students from the very start, with active learning that is exciting, and challenging and encourages deeper levels of learning.

The best part? Since students have an interest in the subject matter from the start, it's easier for instructors to reengage and motivate students in the lesson when distractions arise.

Students receive encouragement to learn about and explore the world around them. Since discovery learning is self-paced, your students can advance only when they are ready.

With this type of learning, students can take an active role in their education and gain knowledge via direct experiences, reflection, and interaction with their classmates.

4. Increases levels of retention

Discovery Learning is based on the idea that if students can come up with an answer by themselves, they will be more likely to remember it for a longer period than if they were taught the information by their teacher.

Unlike other types of learning that are rooted in memorization, discovery learning is a learner-centered approach that allows people to explore new ways of thinking and truly discover the meaning of the subject they're learning in order to reach the desired outcome all by themselves.

Though this process takes considerably longer than if the teacher were simply to tell the students the answers and ask them to commit these to memory. But the long-term retention is much greater with discovery learning.

5. Generates life-long results

The Discovery Learning model creates life-long results. Your students will learn at their own pace with plenty of hands-on and project-based activities, keeping them engaged and excited to come back for more.

This model is especially effective in science, overall knowledge of the world, and life-long learning. With this method, students learn by doing.

Machine learning definition

Machine learning is a subfield of artificial intelligence (AI) that uses algorithms trained on data sets to create self-learning models that are capable of predicting outcomes and classifying information without human intervention. Machine learning is used today for a wide range of commercial purposes, including suggesting products to consumers based on their past purchases, predicting stock market fluctuations, and translating text from one language to another.

In common usage, the terms "machine learning" and "artificial intelligence" are often used interchangeably with one another due to the prevalence of machine learning for AI purposes in the world today. But, the two terms are meaningfully distinct. While AI refers to the general attempt to create machines capable of human-like cognitive abilities, machine learning specifically refers to the use of algorithms and data sets to do so.

Examples and use cases

Machine learning is typically the most mainstream type of AI technology in use around the world today. Some of the most common examples of machine learning that you may have interacted with in your day-to-day life include:

- Recommendation engines that suggest products, songs, or television shows to you, such as those found on Amazon, Spotify, or Netflix.

- Speech recognition software that allows you to convert voice memos into text.
- A bank's fraud detection services automatically flag suspicious transactions.
- Self-driving cars and driver assistance features, such as blind-spot detection and automatic stopping, improve overall vehicle safety.

Definition of learning:

A computer program is said to *learn* from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks T, as measured by P, improves with experience E.

Examples

- Handwriting recognition learning problem
    - Task T : Recognizing and classifying handwritten words within images
    - Performance P : Percent of words correctly classified
    - Training experience E : A dataset of handwritten words with given classifications
- A robot driving learning problem
    - Task T : Driving on highways using vision sensors
    - Performance P : Average distance traveled before an error
    - Training experience E : A sequence of images and steering commands recorded while observing a human driver

Definition: A computer program which learns from experience is called a machine learning program or simply a learning program .

How machine learning works

UC Berkeley (link resides outside ibm.com) breaks out the learning system of a machine learning algorithm into three main parts.

1. A Decision Process: In general, machine learning algorithms are used to make a prediction or classification. Based on some input data, which can be labeled or unlabeled, your algorithm will produce an estimate about a pattern in the data.
2. An Error Function: An error function evaluates the prediction of the model. If there are known examples, an error function can make a comparison to assess the accuracy of the model.
1. A Model Optimization Process: If the model can fit better to the data points in the training set, then weights are adjusted to reduce the discrepancy between the known example and the model estimate. The algorithm will repeat this "evaluate and optimize" process, updating weights autonomously until a threshold of accuracy has been met.

Difference between Machine Learning and Traditional Programming

The Difference between Machine Learning and Traditional Programming is as follows:

Machine Learning	Traditional Programming	Artificial Intelligence
Machine Learning is a subset of artificial intelligence(AI) that focus on learning from data to develop an algorithm that can be used to make a prediction.	In traditional programming, rule-based code is written by the developers depending on the problem statements.	Artificial Intelligence involves making the machine as much capable, So that it can perform the tasks that typically require human intelligence.

Machine Learning	Traditional Programming	Artificial Intelligence
Machine Learning uses a data-driven approach, It is typically trained on historical data and then used to make predictions on new data.	Traditional programming is typically rule-based and deterministic. It hasn't self-learning features like Machine Learning and AI.	AI can involve many different techniques, including Machine Learning and Deep Learning, as well as traditional rule-based programming.
ML can find patterns and insights in large datasets that might be difficult for humans to discover.	Traditional programming is totally dependent on the intelligence of developers. So, it has very limited capability.	Sometimes AI uses a combination of both Data and Pre-defined rules, which gives it a great edge in solving complex tasks with good accuracy which seem impossible to humans.
Machine Learning is the subset of AI. And Now it is used in various AI-based tasks like Chatbot Question answering, self-driven car., etc.	Traditional programming is often used to build applications and software systems that have specific functionality.	AI is a broad field that includes many different applications, including natural language processing, computer vision, and robotics.

How machine learning algorithms work
Machine Learning works in the following manner.

- **Forward Pass:** In the Forward Pass, the machine learning algorithm takes in input data and produces an output. Depending on the model algorithm it computes the predictions.
- **Loss Function:** The loss function, also known as the error or cost function, is used to evaluate the accuracy of the predictions made by the model. The function compares the predicted output of the model to the actual output and calculates the difference between them. This difference is known as error or loss. The goal of the model is to minimize the error or loss function by adjusting its internal parameters.
- **Model Optimization Process:** The model optimization process is the iterative process of adjusting the internal parameters of the model to minimize the error or loss function. This is done using an optimization algorithm, such as **gradient descent**. The optimization algorithm calculates the gradient of the error function with respect to the model's parameters and uses this information to adjust the parameters to reduce the error. The algorithm repeats this process until the error is minimized to a satisfactory level.

Once the model has been trained and optimized on the training data, it can be used to make predictions on new, unseen data. The accuracy of the model's predictions can be evaluated using various performance metrics, such as accuracy, precision, recall, and F1-score.

Machine Learning lifecycle:

The lifecycle of a machine learning project involves a series of steps that include:

1. **Study the Problems:** The first step is to study the problem. This step involves understanding the business problem and defining the objectives of the model.
2. **Data Collection:** When the problem is well-defined, we can collect the relevant data required for the model. The data could come from various sources such as databases, APIs, or web scraping.

3. **Data Preparation:** When our problem-related data is collected. then it is a good idea to check the data properly and make it in the desired format so that it can be used by the model to find the hidden patterns. This can be done in the following steps:
    - Data cleaning
    - Data Transformation
    - Explanatory Data Analysis and Feature Engineering
    - Split the dataset for training and testing.
4. **Model Selection:** The next step is to select the appropriate machine learning algorithm that is suitable for our problem. This step requires knowledge of the strengths and weaknesses of different algorithms. Sometimes we use multiple models and compare their results and select the best model as per our requirements.
5. **Model building and Training:** After selecting the algorithm, we have to build the model.
    1. In the case of traditional machine learning building mode is easy it is just a few hyperparameter tunings.
    2. In the case of deep learning, we have to define layer-wise architecture along with input and output size, number of nodes in each layer, loss function, gradient descent optimizer, etc.
    3. After that model is trained using the preprocessed dataset.
6. **Model Evaluation:** Once the model is trained, it can be evaluated on the test dataset to determine its accuracy and performance using different techniques like classification report, F1 score, precision, recall, ROC Curve, Mean Square error, absolute error, etc.
7. **Model Tuning:** Based on the evaluation results, the model may need to be tuned or optimized to improve its performance. This involves tweaking the hyperparameters of the model.
8. **Deployment:** Once the model is trained and tuned, it can be deployed in a production environment to make predictions on new data. This step requires integrating the model into an existing software system or creating a new system for the model.
9. **Monitoring and Maintenance:** Finally, it is essential to monitor the model's performance in the production environment and perform maintenance tasks as required. This involves monitoring for data drift, retraining the model as needed, and updating the model as new data becomes available.

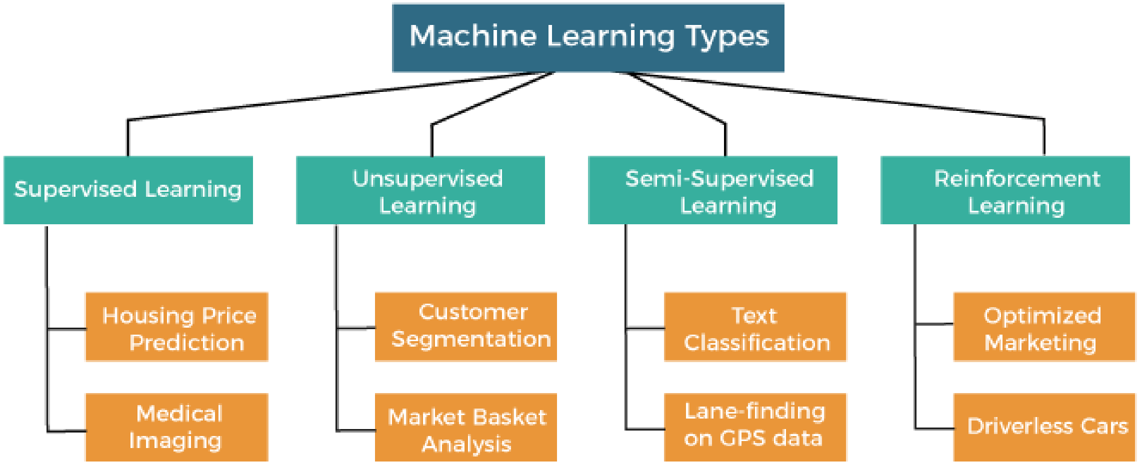
Types of Machine Learning

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.
- Supervised learning allows collecting data and produces data output from previous experiences.
- Helps to optimize performance criteria with the help of experience.
- Supervised machine learning helps to solve various types of real-world computation problems.
- It performs classification and regression tasks.
- It allows estimating or mapping the result to a new sample.
- We have complete control over choosing the number of classes we want in the training data.

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.
- Classifying big data can be challenging.
- Training for supervised learning needs a lot of computation time. So, it requires a lot of time.
- Supervised learning cannot handle all complex tasks in Machine Learning.

- Computation time is vast for supervised learning.
- It requires a labelled data set.
- It requires a training process

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.
- It does not require training data to be labeled.
- Dimensionality reduction can be easily accomplished using unsupervised learning.
- Capable of finding previously unknown patterns in data.
- Flexibility: Unsupervised learning is flexible in that it can be applied to a wide variety of problems, including clustering, anomaly detection, and association rule mining.

- Exploration: Unsupervised learning allows for the exploration of data and the discovery of novel and potentially useful patterns that may not be apparent from the outset.
- Low cost: Unsupervised learning is often less expensive than supervised learning because it doesn't require labeled data, which can be time-consuming and costly to obtain.

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.
- Difficult to measure accuracy or effectiveness due to lack of predefined answers during training.
- The results often have lesser accuracy.
- The user needs to spend time interpreting and label the classes which follow that classification.
- Lack of guidance: Unsupervised learning lacks the guidance and feedback provided by labeled data, which can make it difficult to know whether the discovered patterns are relevant or useful.
- Sensitivity to data quality: Unsupervised learning can be sensitive to data quality, including missing values, outliers, and noisy data.
- Scalability: Unsupervised learning can be computationally expensive, particularly for large datasets or complex algorithms, which can limit its scalability.

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.

Supervised vs. Unsupervised Machine Learning:

Parameters	**Supervised machine learning**	**Unsupervised machine learning**
Input Data	Algorithms are trained using labeled data.	Algorithms are used against data that is not labeled
Computational Complexity	Simpler method	Computationally complex
Accuracy	Highly accurate	Less accurate
No. of classes	No. of classes is known	No. of classes is not known
Data Analysis	Uses offline analysis	Uses real-time analysis of data
Algorithms used	Linear and Logistics regression, Random forest, Support Vector Machine, Neural Network, etc.	K-Means clustering, Hierarchical clustering, Apriori algorithm, etc.
Output	Desired output is given.	Desired output is not given.
Training data	Use training data to infer model.	No training data is used.
Complex model	It is not possible to learn larger and more complex models than with supervised learning.	It is possible to learn larger and more complex models with unsupervised learning.
Model	We can test our model.	We can not test our model.
Called as	Supervised learning is also called classification.	Unsupervised learning is also called clustering.
Example	Example: Optical character recognition.	Example: Find a face in an image.

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.

Criteria	**Supervised ML**	**Unsupervised ML**	**Reinforcement ML**
Definition	Learns by using labelled data	Trained using unlabelled data without any guidance.	Works on interacting with the environment
Type of data	Labelled data	Unlabelled data	No – predefined data
Type of problems	Regression and classification	Association and Clustering	Exploitation or Exploration
Supervision	Extra supervision	No supervision	No supervision
Algorithms	Linear Regression, Logistic Regression, SVM, KNN etc.	K – Means, C – Means, Apriori	Q – Learning, SARSA
Aim	Calculate outcomes	Discover underlying patterns	Learn a series of action
Application	Risk Evaluation, Forecast Sales	Recommendation System, Anomaly Detection	Self Driving Cars, Gaming, Healthcare

Probabilistic Models in Machine Learning

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.

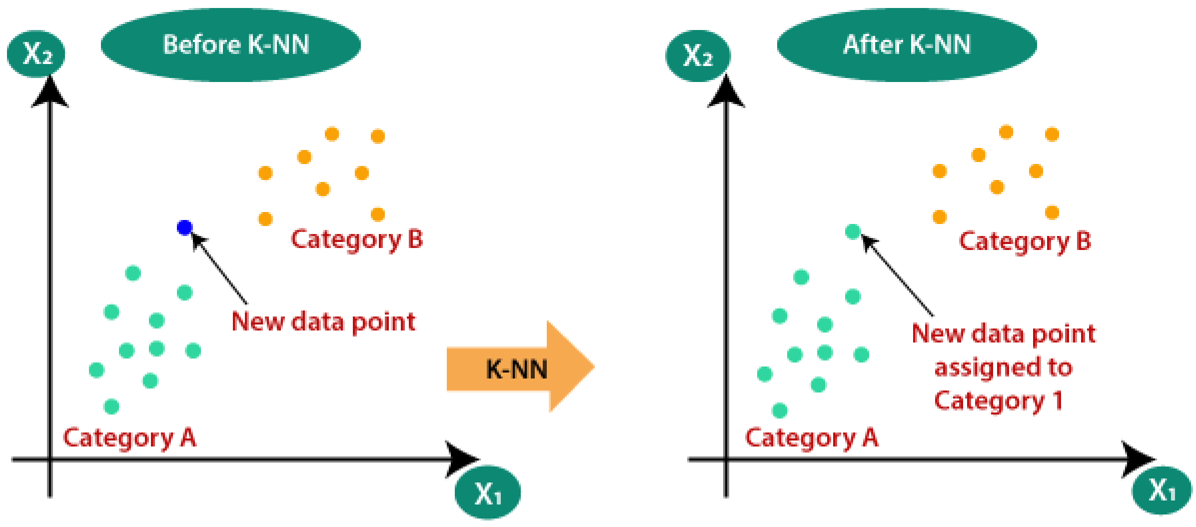
K-Nearest Neighbor(KNN) Algorithm for Machine Learning

- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.

- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
- **Example:** Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

<u>Why do we need a K-NN Algorithm?</u>

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:

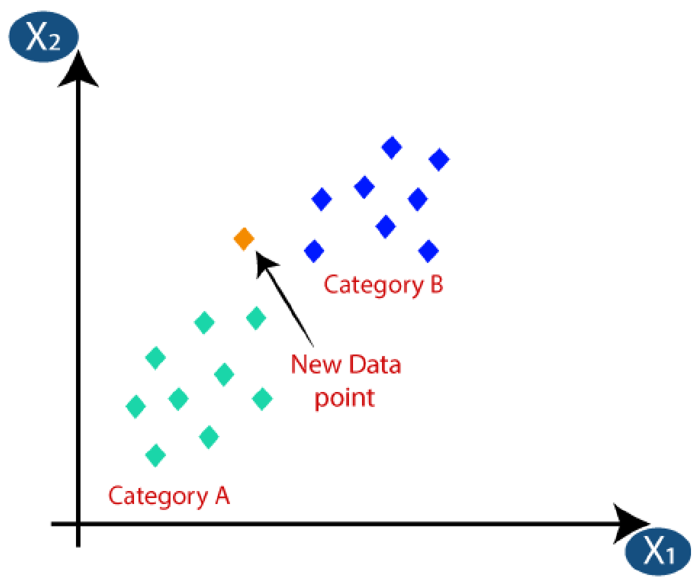


How does K-NN work?

The K-NN working can be explained on the basis of the below algorithm:

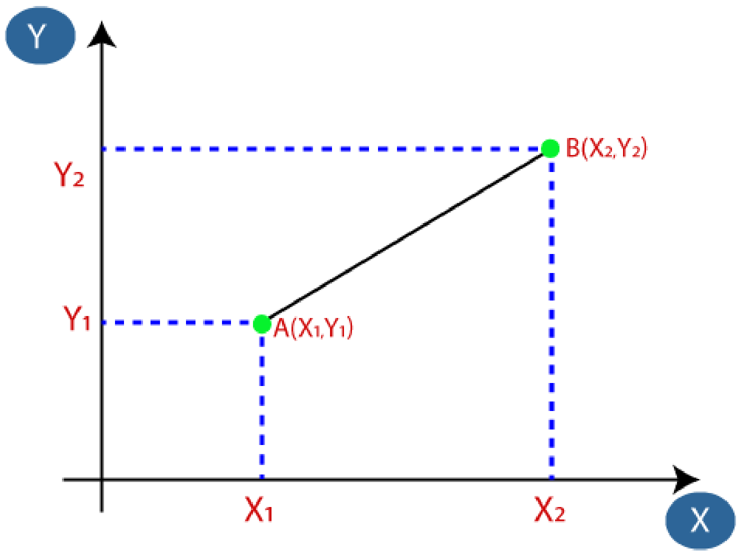
- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Our model is ready.

Suppose we have a new data point and we need to put it in the required category. Consider the below image:



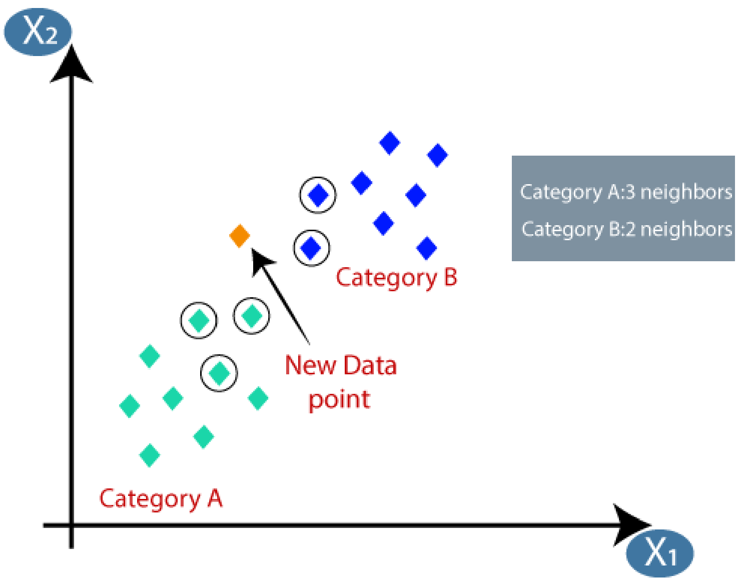
- Firstly, we will choose the number of neighbors, so we will choose the k=5.

- Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:



Euclidean Distance between A_1 and B_2 = $\sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$

- By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image:



- As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.

How to select the value of K in the K-NN Algorithm?

Below are some points to remember while selecting the value of K in the K-NN algorithm:

- There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.

- A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
- Large values for K are good, but it may find some difficulties.

Applications of the KNN Algorithm

- **Data Preprocessing** – While dealing with any Machine Learning problem we first perform the EDA part in which if we find that the data contains missing values then there are multiple imputation methods are available as well. One of such method is KNN Imputer which is quite effective ad generally used for sophisticated imputation methodologies.
- **Pattern Recognition** – KNN algorithms work very well if you have trained a KNN algorithm using the MNIST dataset and then performed the evaluation process then you must have come across the fact that the accuracy is too high.
- **Recommendation Engines** – The main task which is performed by a KNN algorithm is to assign a new query point to a pre-existed group that has been created using a huge corpus of datasets. This is exactly what is required in the recommender systems to assign each user to a particular group and then provide them recommendations based on that group's preferences.

Advantages of the KNN Algorithm

- **Easy to implement** as the complexity of the algorithm is not that high.
- **Adapts Easily** – As per the working of the KNN algorithm it stores all the data in memory storage and hence whenever a new example or data point is added then the algorithm adjusts itself as per that new example and has its contribution to the future predictions as well.
- **Few Hyperparameters** – The only parameters which are required in the training of a KNN algorithm are the value of k and the choice of the distance metric which we would like to choose from our evaluation metric.

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.
-

Disadvantages of the KNN Algorithm

- **Does not scale** – As we have heard about this that the KNN algorithm is also considered a Lazy Algorithm. The main significance of this term is that this takes lots of computing power as well as data storage. This makes this algorithm both time-consuming and resource exhausting.
- **Curse of Dimensionality** – There is a term known as the peaking phenomenon according to this the KNN algorithm is affected by the curse of dimensionality which implies the algorithm faces a hard time classifying the data points properly when the dimensionality is too high.
- **Prone to Overfitting** – As the algorithm is affected due to the curse of dimensionality it is prone to the problem of overfitting as well. Hence generally feature selection as well as dimensionality reduction techniques are applied to deal with this problem.

- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.
- KNN stands for K-nearest neighbour, it's one of the Supervised learning algorithm mostly used for classification of data on the basis how it's neighbour are classified. KNN stores all available cases and classifies new cases based on a similarity

measure. K in KNN is a parameter that refers to the number of the nearest neighbours to include in the majority voting process.

- **How do we choose K?**
- Sqrt(n), where n is a total number of data points(if in case n is even we have to make the value odd by adding 1 or subtracting 1 that helps in select better)
- **When to use KNN?**
- We can use KNN when Dataset is labelled and noise-free and it's must be small because KNN is a *"Lazy learner"*. Let's understand KNN algorithm with the help of an example

NAME	AGE	GENDER	CLASS OF SPORTS
Ajay	32	0	Football
Mark	40	0	Neither
Sara	16	1	Cricket
Zaira	34	1	Cricket
Sachin	55	0	Neither
Rahul	40	0	Cricket
Pooja	20	1	Neither
Smith	15	0	Cricket
Laxmi	55	1	Football
Michael	15	0	Football

- Here male is denoted with numeric value 0 and female with 1. Let's find in which class of people Angelina will lie whose k factor is 3 and age is 5. So we have to find out the distance using
- $d=\sqrt{(x2-x1)^2+(y2-y1)^2}$ *to find the distance between any two points.*
- So let's find out the distance between Ajay and Angelina using formula
- $d=\sqrt{(age2-age1)^2+(gender2-gender1)^2}$
- $d=\sqrt{(5-32)^2+(1-0)^2}$
- $d=\sqrt{729+1}$
- $d=27.02$
- Similarly, we find out all distance one by one.

Distance between Angelina and	Distance
Ajay	27.02
Mark	35.01

Distance between Angelina and	Distance
Sara	11.00
Zaira	29.00
Sachin	50.01
Rahul	35.01
Pooja	15.00
Smith	10.05
Laxmi	50.00
Michael	10.05

- So the value of ***k*** factor is 3 for Angelina. And the closest to 3 is 9,10,10.5 that is closest to Angelina are Zaira, Smith and Michael.
-

Zaira	9	cricket
Michael	10	cricket
smith	10.5	football

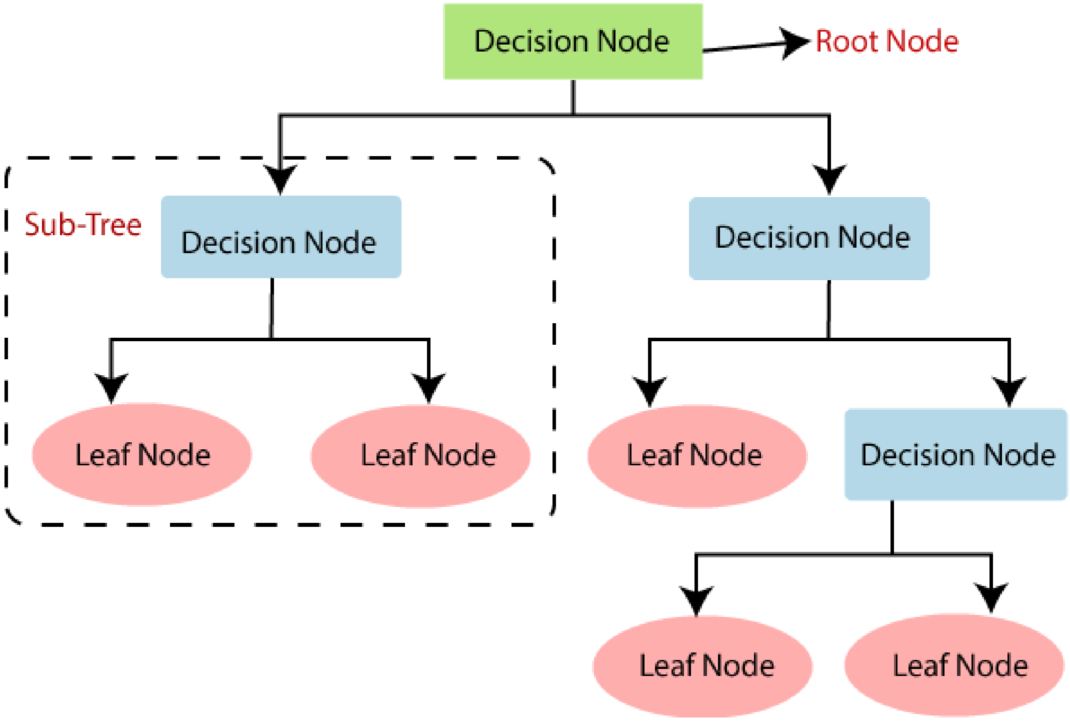
- so according to KNN algorithm, Angelina will be in the class of people who like cricket. So this is how KNN algorithm works.

Decision Tree Classification Algorithm

- 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.

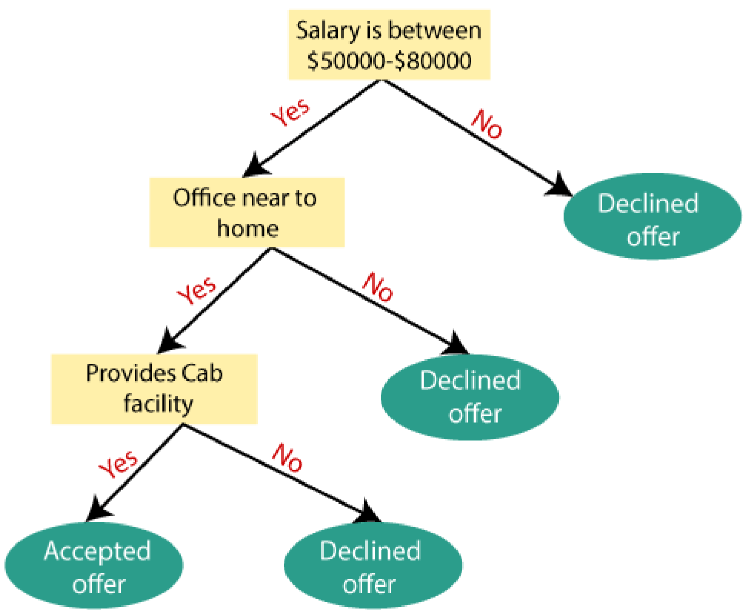
How does the Decision Tree algorithm Work?

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.

Python Implementation of Decision Tree

Now we will implement the Decision tree using Python. For this, we will use the dataset "**user_data.csv**," which we have used in previous classification models. By using the same dataset, we can compare the Decision tree classifier with other classification models such as KNN SVM, LogisticRegression, etc.

Steps will also remain the same, which are given below:

- **Data Pre-processing step**
- **Fitting a Decision-Tree algorithm to the Training set**
- **Predicting the test result**
- **Test accuracy of the result(Creation of Confusion matrix)**
- **Visualizing the test set result.**

Example:-

Day	Outlook	Temp	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Day	Outlook	Temp	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Attribute: Outlook

$Values(Outlook) = Sunny, Overcast, Rain$

$S = [9+, 5-]$ $\quad Entropy(S) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.94$

$S_{Sunny} \leftarrow [2+, 3-]$ $\quad Entropy(S_{Sunny}) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.971$

$S_{Overcast} \leftarrow [4+, 0-]$ $\quad Entropy(S_{Overcast}) = -\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4} = 0$

$S_{Rain} \leftarrow [3+, 2-]$ $\quad Entropy(S_{Rain}) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.971$

$$Gain(S, Outlook) = Entropy(S) - \sum_{v \in \{Sunny, Overcast, Rain\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$Gain(S, Outlook)$

$$= Entropy(S) - \frac{5}{14}Entropy(S_{Sunny}) - \frac{4}{14}Entropy(S_{Overcast}) - \frac{5}{14}Entropy(S_{Rain})$$

$$Gain(S, Outlook) = 0.94 - \frac{5}{14}0.971 - \frac{4}{14}0 - \frac{5}{14}0.971 = 0.2464$$

Day	Outlook	Temp	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Attribute: Temp

$Values(Temp) = Hot, Mild, Cool$

$S = [9+, 5-]$ $\quad Entropy(S) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.94$

$S_{Hot} \leftarrow [2+, 2-]$ $\quad Entropy(S_{Hot}) = -\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4} = 1.0$

$S_{Mild} \leftarrow [4+, 2-]$ $\quad Entropy(S_{Mild}) = -\frac{4}{6}\log_2\frac{4}{6} - \frac{2}{6}\log_2\frac{2}{6} = 0.9183$

$S_{Cool} \leftarrow [3+, 1-]$ $\quad Entropy(S_{Cool}) = -\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4} = 0.8113$

$$Gain(S, Temp) = Entropy(S) - \sum_{v \in \{Hot, Mild, Cool\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$Gain(S, Temp)$

$$= Entropy(S) - \frac{4}{14}Entropy(S_{Hot}) - \frac{6}{14}Entropy(S_{Mild}) - \frac{4}{14}Entropy(S_{Cool})$$

Day	Outlook	Temp	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Attribute: Humidity

$Values\ (Humidity) = High, Normal$

$S = [9+, 5-]$ $\quad Entropy(S) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.94$

$S_{High} \leftarrow [3+, 4-]$ $\quad Entropy(S_{High}) = -\frac{3}{7}\log_2\frac{3}{7} - \frac{4}{7}\log_2\frac{4}{7} = 0.9852$

$S_{Normal} \leftarrow [6+, 1-]$ $\quad Entropy(S_{Normal}) = -\frac{6}{7}\log_2\frac{6}{7} - \frac{1}{7}\log_2\frac{1}{7} = 0.5916$

$$Gain\ (S, Humidity) = Entropy(S) - \sum_{v \in \{High, Normal\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S, Humidity)$$

$$= Entropy(S) - \frac{7}{14}Entropy(S_{High}) - \frac{7}{14}Entropy(S_{Normal})$$

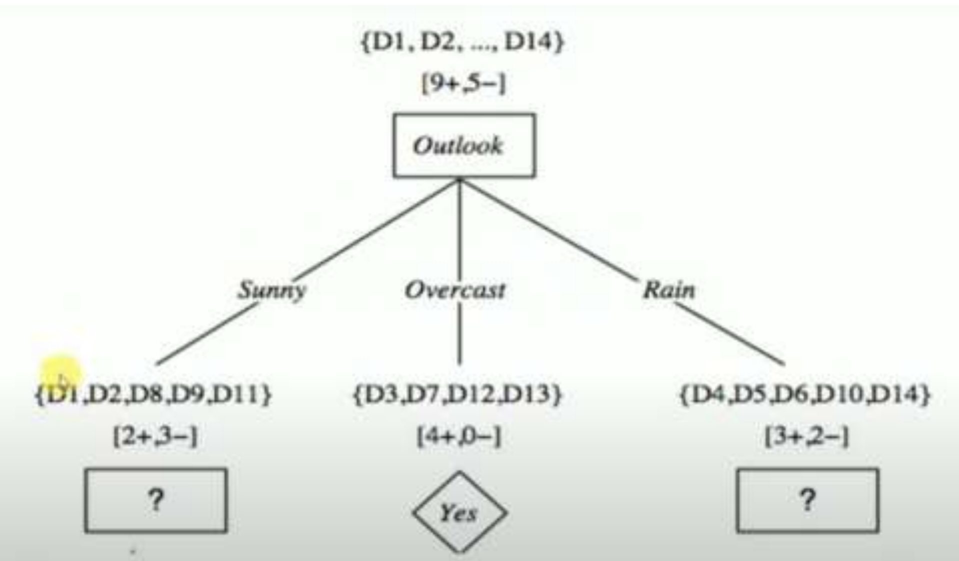
Day	Outlook	Temp	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$$Gain(S, Outlook) = 0.2464$$

$$Gain(S, Temp) = 0.0289$$

$$Gain(S, Humidity) = 0.1516$$

$$Gain(S, Wind) = 0.0478$$



Day	Temp	Humidity	Wind	Play Tennis
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

Attribute: Temp

Values (Temp) = Hot, Mild, Cool

$S_{Sunny} = [2+, 3-]$ $\quad Entropy(S_{Sunny}) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.97$

$S_{Hot} \leftarrow [0+, 2-]$ $\quad Entropy(S_{Hot}) = 0.0$

$S_{Mild} \leftarrow [1+, 1-]$ $\quad Entropy(S_{Mild}) = 1.0$

$S_{Cool} \leftarrow [1+, 0-]$ $\quad Entropy(S_{Cool}) = 0.0$

$$Gain\ (S_{Sunny}, Temp) = Entropy(S) - \sum_{v \in \{Hot, Mild, Cool\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Sunny}, Temp)$$

$$= Entropy(S) - \frac{2}{5}Entropy(S_{Hot}) - \frac{2}{5}Entropy(S_{Mild})$$

$$- \frac{1}{5}Entropy(S_{Cool})$$

$$Gain(S_{sunny}, Temp) = 0.97 - \frac{2}{5}0.0 - \frac{2}{5}1 - \frac{1}{5}0.0 = 0.570$$

Day	Temp	Humidity	Wind	Play Tennis
DI	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
DI1	Mild	Normal	Strong	Yes

Attribute: Humidity

Values (Humidity) = High, Normal

$S_{Sunny} = [2+, 3-]$ $\quad Entropy(S) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.97$

$S_{high} \leftarrow [0+, 3-]$ $\quad Entropy(S_{High}) = 0.0$

$S_{Normal} \leftarrow [2+, 0-]$ $\quad Entropy(S_{Normal}) = 0.0$

$$Gain\ (S_{Sunny}, Humidity) = Entropy(S) - \sum_{v \in \{High, Normal\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Sunny}, Humidity) = Entropy(S) - \frac{3}{5}Entropy(S_{High}) - \frac{2}{5}Entropy(S_{Normal})$$

Day	Temp	Humidity	Wind	Play Tennis
DI	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
DI1	Mild	Normal	Strong	Yes

Attribute: Wind

Values (Wind) = Strong, Weak

$S_{Sunny} = [2+, 3-]$ $\quad Entropy(S) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.97$

$S_{Strong} \leftarrow [1+, 1-]$ $\quad Entropy(S_{Strong}) = 1.0$

$S_{Weak} \leftarrow [1+, 2-]$ $\quad Entropy(S_{Weak}) = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0.9183$

$$Gain\ (S_{Sunny}, Wind) = Entropy(S) - \sum_{v \in \{Strong, Weak\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Sunny}, Wind) = Entropy(S) - \frac{2}{5}Entropy(S_{Strong}) - \frac{3}{5}Entropy(S_{Weak})$$

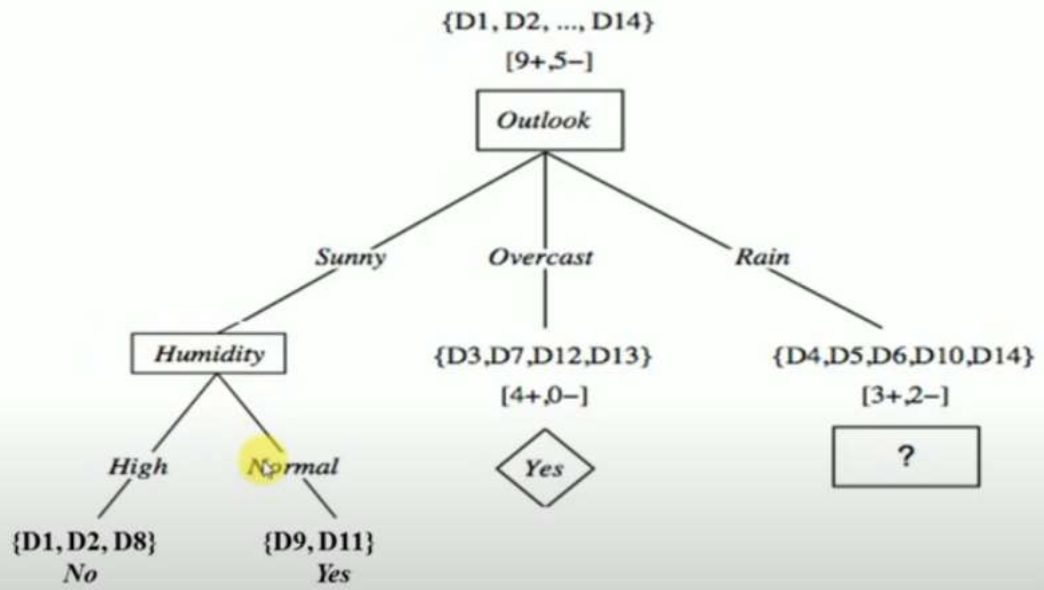
$$Gain(S_{sunny}, Wind) = 0.97 - \frac{2}{5}1.0 - \frac{3}{5}0.918 = 0.0192$$

Day	Temp	Humidity	Wind	Play Tennis
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

$$Gain(S_{sunny}, Temp) = 0.570$$

$$Gain(S_{sunny}, Humidity) = 0.97$$

$$Gain(S_{sunny}, Wind) = 0.0192$$



Day	Temp	Humidity	Wind	Play Tennis
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
D10	Mild	Normal	Weak	Yes
D14	Mild	High	Strong	No

Attribute: Temp

$Values\ (Temp) = Hot, Mild, Cool$

$S_{Rain} = [3+, 2-]$ $\quad$ $Entropy(S_{Sunny}) = -\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} = 0.97$

$S_{Hot} \leftarrow [0+, 0-]$ $\quad$ $Entropy(S_{Hot}) = 0.0$

$S_{Mild} \leftarrow [2+, 1-]$ $\quad$ $Entropy(S_{Mild}) = -\frac{2}{3}log_2\frac{2}{3} - \frac{1}{3}log_2\frac{1}{3} = 0.9183$

Day	Temp	Humidity	Wind	Play Tennis
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
D10	Mild	Normal	Weak	Yes
D14	Mild	High	Strong	No

Attribute: Temp

$Values\ (Temp) = Hot, Mild, Cool$

$S_{Rain} = [3+, 2-]$ $\quad Entropy(S_{Sunny}) = -\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} = 0.97$

$S_{Hot} \leftarrow [0+, 0-]$ $\quad Entropy(S_{Hot}) = 0.0$

$S_{Mild} \leftarrow [2+, 1-]$ $\quad Entropy(S_{Mild}) = -\frac{2}{3}log_2\frac{2}{3} - \frac{1}{3}log_2\frac{1}{3} = 0.9183$

$S_{Cool} \leftarrow [1+, 1-]$ $\quad Entropy(S_{Cool}) = 1.0$

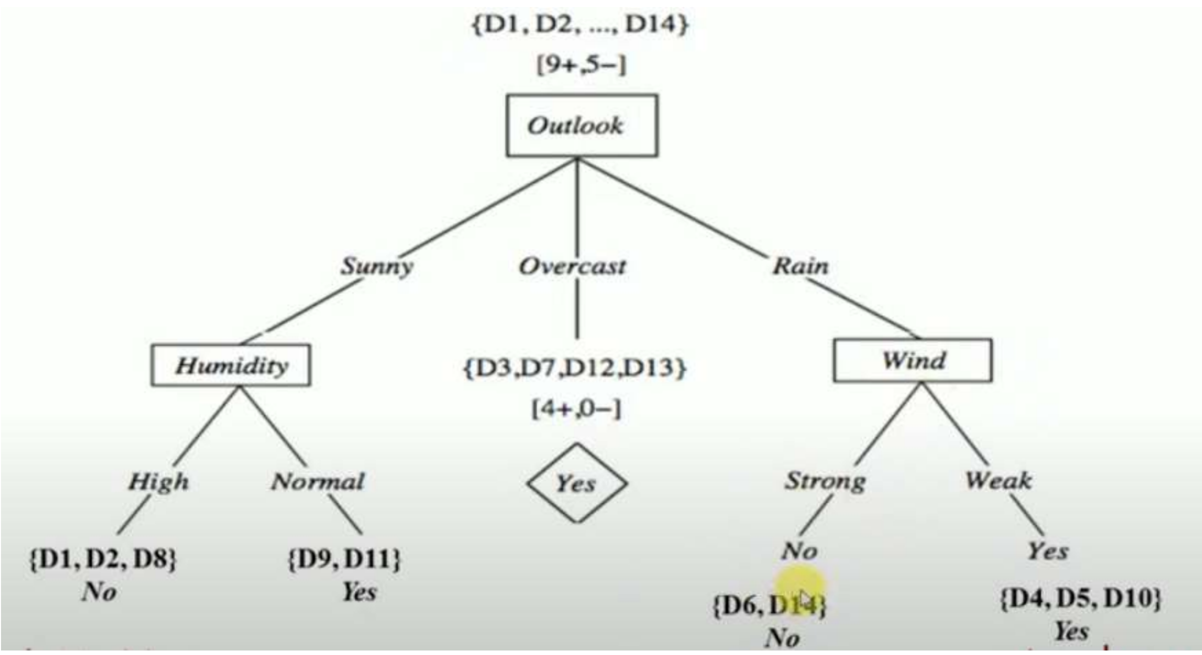
$$Gain\ (S_{Rain}, Temp) = Entropy(S) - \sum_{v \in \{Hot, Mild, Cool\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Rain}, Temp)$$

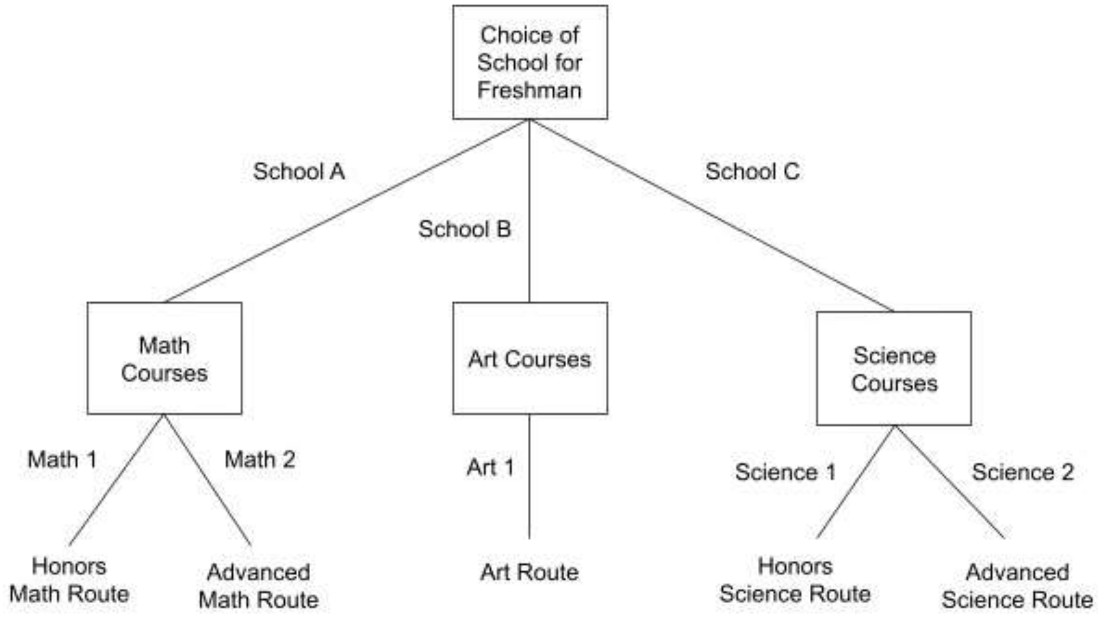
$$= Entropy(S) - \frac{0}{5}Entropy(S_{Hot}) - \frac{3}{5}Entropy(S_{Mild})$$

$$-\frac{2}{5}Entropy(S_{Cool})$$

$$Gain(S_{Rain}, Temp) = 0.97 - \frac{0}{5}0.0 - \frac{3}{5}0.918 - \frac{2}{5}1.0 = 0.0192$$



Example 2:



Random Forest Algorithm

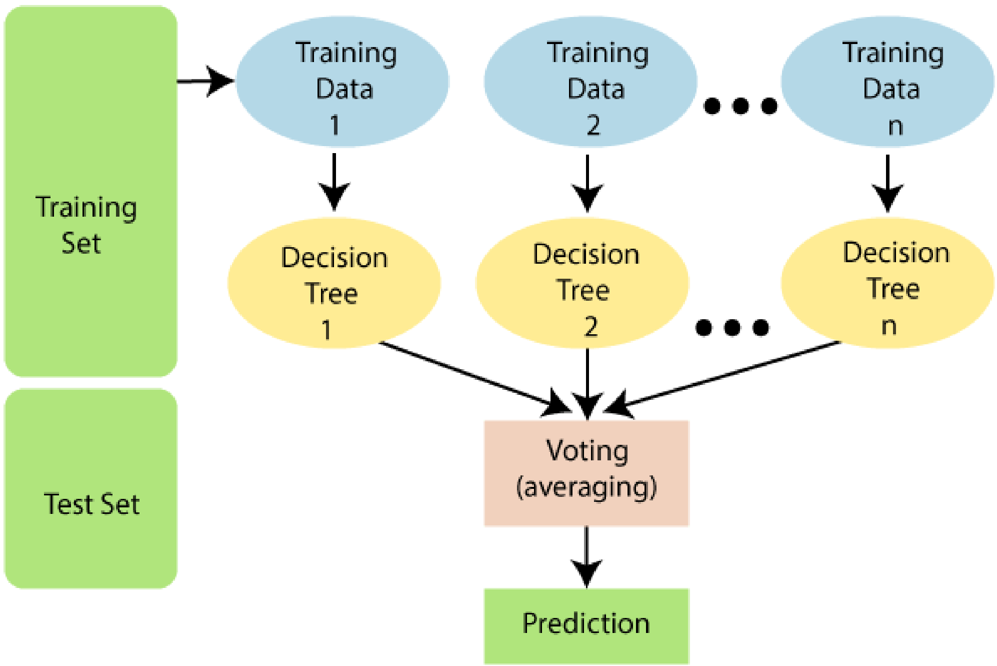
Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning,** which is a process of *combining multiple classifiers to solve a complex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

Backward Skip 10sPlay VideoForward Skip 10s



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:

- 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.
- 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:

<="" li="">

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

How does 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).

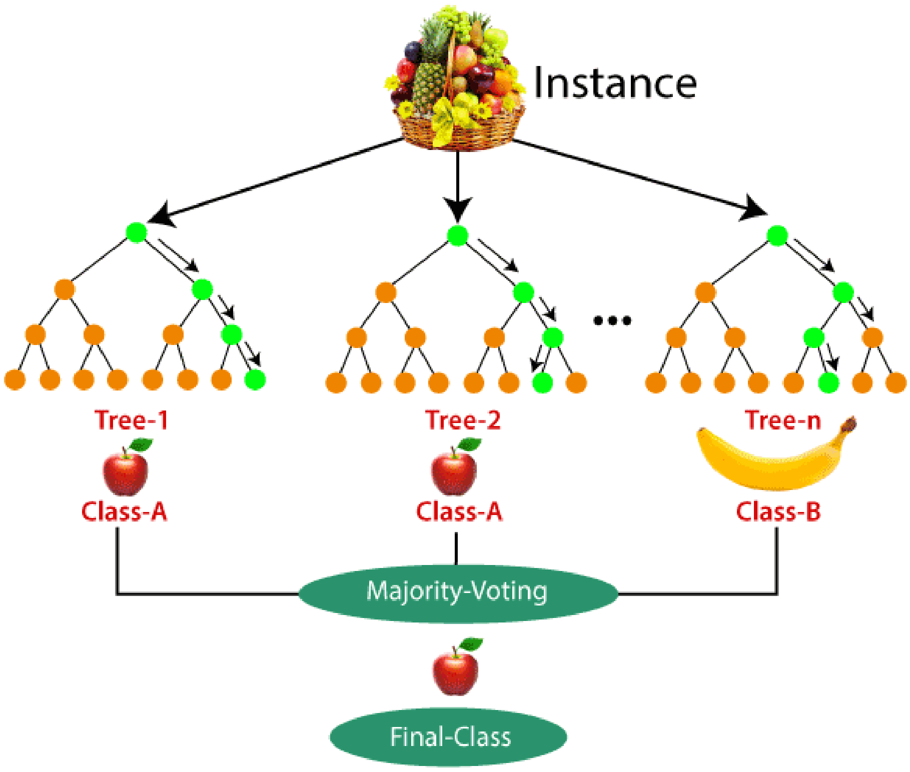
Step-3: Choose the number N for decision trees that you want to build.

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:



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.

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.

Disadvantages of Random Forest

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

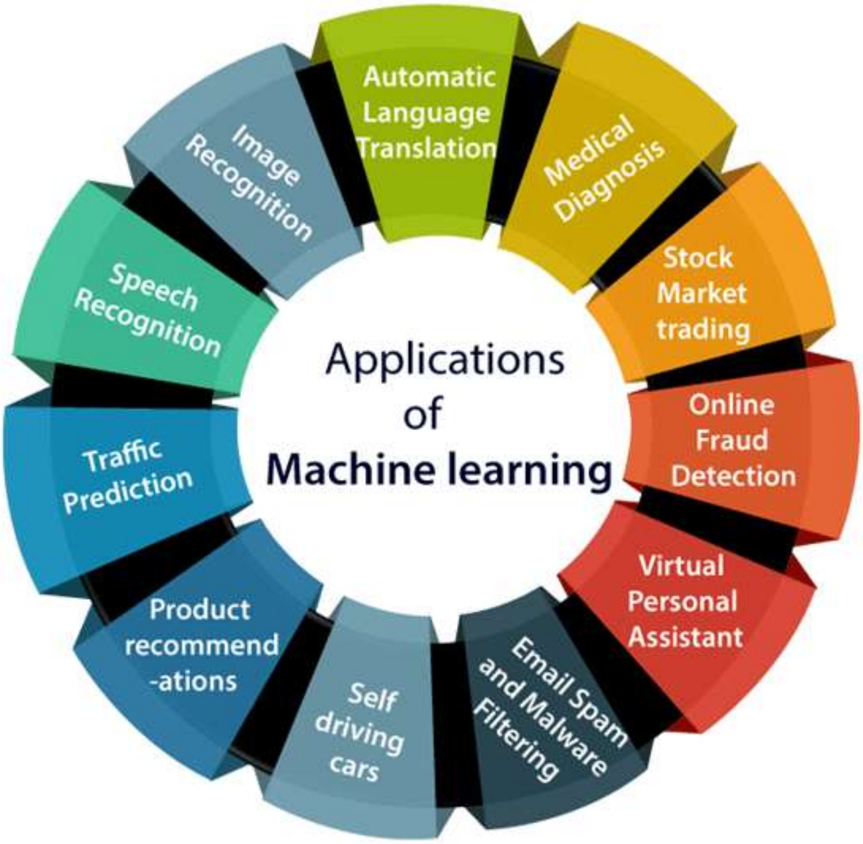
Overview of Random Forest vs Decision Tree

Aspect	Random Forest	Decision Tree
Nature	Ensemble of multiple decision trees	Single decision tree
Bias-Variance Trade-off	Lower variance, reduced overfitting	Higher variance, prone to overfitting
Predictive Accuracy	Generally higher due to ensemble	Prone to overfitting, may vary
Robustness	More robust to outliers and noise	Sensitive to outliers and noise
Training Time	Slower due to multiple tree construction	Faster as it builds a single tree
Interpretability	Less interpretable due to ensemble	More interpretable as a single tree
Feature Importance	Provides feature importance scores	Provides feature importance, but less reliable
Usage	Suitable for complex tasks, high-dimensional data	Simple tasks, easy interpretation
	1. Decision trees normally suffer from the problem of overfitting if it's allowed to grow without any control.	1. Random forests are created from subsets of data, and the final output is based on average or majority ranking; hence the problem of overfitting is taken care of.
	2. A single decision tree is faster in computation.	2. It is comparatively slower.
	3. When a data set with features is taken as input by a decision tree, it will formulate some rules to make predictions.	3. Random forest randomly selects observations, builds a decision tree, and takes the average result. It doesn't use any set of formulas.

Applications of Machine learning

Machine learning is a buzzword for today's technology, and it is growing very rapidly day by day. We are using machine learning in our daily life even without knowing it such as Google

Maps, Google assistant, Alexa, etc. Below are some most trending real-world applications of Machine Learning:



1. Image Recognition:

Image recognition is one of the most common applications of machine learning. It is used to identify objects, persons, places, digital images, etc. The popular use case of image recognition and face detection is, **Automatic friend tagging suggestion**:

Facebook provides us a feature of auto friend tagging suggestion. Whenever we upload a photo with our Facebook friends, then we automatically get a tagging suggestion with name, and the technology behind this is machine learning's **face detection** and **recognition algorithm**.

It is based on the Facebook project named "**Deep Face**," which is responsible for face recognition and person identification in the picture.

2. Speech Recognition

While using Google, we get an option of "**Search by voice**," it comes under speech recognition, and it's a popular application of machine learning.

Speech recognition is a process of converting voice instructions into text, and it is also known as "**Speech to text**", or "**Computer speech recognition**." At present, machine learning algorithms are widely used by various applications of speech recognition. **Google assistant**, **Siri**, **Cortana**, and **Alexa** are using speech recognition technology to follow the voice instructions.

3. Traffic prediction:

If we want to visit a new place, we take help of Google Maps, which shows us the correct path with the shortest route and predicts the traffic conditions.

It predicts the traffic conditions such as whether traffic is cleared, slow-moving, or heavily congested with the help of two ways:

- **Real Time location** of the vehicle form Google Map app and sensors
- **Average time has taken** on past days at the same time.

Everyone who is using Google Map is helping this app to make it better. It takes information from the user and sends back to its database to improve the performance.

4. Product recommendations:

Machine learning is widely used by various e-commerce and entertainment companies such as **Amazon**, **Netflix**, etc., for product recommendation to the user. Whenever we search for some product on Amazon, then we started getting an advertisement for the same product while internet surfing on the same browser and this is because of machine learning.

Google understands the user interest using various machine learning algorithms and suggests the product as per customer interest.

As similar, when we use Netflix, we find some recommendations for entertainment series, movies, etc., and this is also done with the help of machine learning.

5. Self-driving cars:

One of the most exciting applications of machine learning is self-driving cars. Machine learning plays a significant role in self-driving cars. Tesla, the most popular car manufacturing company is working on self-driving car. It is using unsupervised learning method to train the car models to detect people and objects while driving.

6. Email Spam and Malware Filtering:

Whenever we receive a new email, it is filtered automatically as important, normal, and spam. We always receive an important mail in our inbox with the important symbol and spam emails in our spam box, and the technology behind this is Machine learning. Below are some spam filters used by Gmail:

- Content Filter
- Header filter
- General blacklists filter
- Rules-based filters
- Permission filters

Some machine learning algorithms such as **Multi-Layer Perceptron**, **Decision tree**, and **Naïve Bayes classifier** are used for email spam filtering and malware detection.

7. Virtual Personal Assistant:

We have various virtual personal assistants such as **Google assistant**, **Alexa**, **Cortana**, **Siri**. As the name suggests, they help us in finding the information using our voice instruction. These assistants can help us in various ways just by our voice instructions such as Play music, call someone, Open an email, Scheduling an appointment, etc.

These virtual assistants use machine learning algorithms as an important part.

These assistant record our voice instructions, send it over the server on a cloud, and decode it using ML algorithms and act accordingly.

8. Online Fraud Detection:

Machine learning is making our online transaction safe and secure by detecting fraud transaction. Whenever we perform some online transaction, there may be various ways that a fraudulent transaction can take place such as **fake accounts**, **fake ids**, and **steal money** in the middle of a transaction. So to detect this, **Feed Forward Neural network** helps us by checking whether it is a genuine transaction or a fraud transaction.

For each genuine transaction, the output is converted into some hash values, and these values become the input for the next round. For each genuine transaction, there is a specific pattern which gets change for the fraud transaction hence, it detects it and makes our online transactions more secure.

9. Stock Market trading:

Machine learning is widely used in stock market trading. In the stock market, there is always a risk of up and downs in shares, so for this machine learning's **long short term memory neural network** is used for the prediction of stock market trends.

10. Medical Diagnosis:

In medical science, machine learning is used for diseases diagnoses. With this, medical technology is growing very fast and able to build 3D models that can predict the exact position of lesions in the brain.

It helps in finding brain tumors and other brain-related diseases easily.

11. Automatic Language Translation:

Nowadays, if we visit a new place and we are not aware of the language then it is not a problem at all, as for this also machine learning helps us by converting the text into our known languages. Google's GNMT (Google Neural Machine Translation) provide this feature, which is a Neural Machine Learning that translates the text into our familiar language, and it called as automatic translation.

The technology behind the automatic translation is a sequence to sequence learning algorithm, which is used with image recognition and translates the text from one language to another language.