Machine Learning
(6CS4-02)

Unit-1 Notes

Vision of the Institute

To become a renowned center of outcome based learning and work towards academic, professional, cultural and social enrichment of the lives of individuals and communities.

Mission of the Institute

M1- Focus on evaluation of learning outcomes and motivate students to inculcate research aptitude by project based learning.
M2- Identify, based on informed perception of Indian, regional and global needs, the areas of focus and provide platform to gain knowledge and solutions.
M3- Offer opportunities for interaction between academia and industry.
M4- Develop human potential to its fullest extent so that intellectually capable and imaginatively gifted leaders can emerge in a range of professions.

Vision of the Department

To become renowned Centre of excellence in computer science and engineering and make competent engineers & professionals with high ethical values prepared for lifelong learning.

Mission of the Department

M1-To impart outcome based education for emerging technologies in the field of computer science and engineering.
M2-To provide opportunities for interaction between academia and industry.
M3- To provide platform for lifelong learning by accepting the change in technologies
M4- To develop aptitude of fulfilling social responsibilities.

Program Outcomes (PO)

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problemsand design system components or processes that meet thespecified needs with appropriate consideration for the public health and safety, andthe cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issuesand the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Educational Objectives (PEO)

1. To provide students with the fundamentals of Engineering Sciences with more emphasis in Computer Science &Engineering by way of analyzing and exploiting engineering challenges.
2. To train students with good scientific and engineering knowledge so as to comprehend, analyze, design, and create novel products and solutions for the real life problems.
3. To inculcate professional and ethical attitude, effective communication skills, teamwork skills, multidisciplinary approach, entrepreneurial thinking and an ability to relate engineering issues with social issues.
4. To provide students with an academic environment aware of excellence, leadership, written ethical codes and guidelines, and the self-motivated life-long learning needed for a successful professional career.
5. To prepare students to excel in Industry and Higher education by Educating Students along with High moral values and Knowledge

Program Specific Outcomes (PSO)

PSO1: Ability to interpret and analyze network specific and cyber security issues, automation in real word environment.

PSO2: Ability to Design and Develop Mobile and Web-based applications under realistic constraints.

Course Outcome:

CO1: Understand the concept of machine learning and apply supervised learning techniques.
CO2: Illustrate various unsupervised leaning algorithm for clustering, and market basket analysis.
CO3: Analyze statistical learning theory for dimension reduction and model evaluation in machine learning.
CO4: Apply the concept of semi supervised learning, reinforcement learning and recommendation system.

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
Understand the concept of machine learning and apply supervised learning techniques.	3	3	3	3	2	1	1	1	1	2	1	3
Illustrate various unsupervised leaning algorithm for clustering, and market basket analysis.	3	3	3	2	2	1	1	1	1	1	1	3
Analyze statistical learning theory for dimension reduction and model evaluation in machine learning.	3	3	3	3	2	2	2	2	1	2	2	3
Apply the concept of semi supervised learning, reinforcement learning and recommendation system.	3	3	3	3	2	1	1	1	1	2	1	3

SYLLABUS:

RAJASTHAN TECHNICAL UNIVERSITY, KOTA

Syllabus

III Year-VI Semester: B.Tech. Computer Science and Engineering

6CS4-02:Machine Learning

Credit: 3 **Max. Marks: 150(IA:30, ETE:120)**

3L+0T+0P **End Term Exam: 3 Hours**

SN	Contents	Hours
1	**Introduction:** Objective, scope and outcome of the course.	01
2	**Supervised learning algorithm:** Introduction, types of learning, application, Supervised learning: Linear Regression Model, Naive Bayes classifier Decision Tree, K nearest neighbor, Logistic Regression, Support Vector Machine, Random forest algorithm	09
3	**Unsupervised learning algorithm:** Grouping unlabelled items using k-means clustering, Hierarchical Clustering, Probabilistic clustering, Association rule mining, Apriori Algorithm, f-p growth algorithm, Gaussian mixture model.	08
4	**Introduction to Statistical Learning Theory**, Feature extraction - Principal component analysis, Singular value decomposition. Feature selection – feature ranking and subset selection, filter, wrapper and embedded methods, Evaluating Machine Learning algorithms and Model Selection.	08
5	**Semi supervised learning, Reinforcement learning:** Markov decision process (MDP), Bellman equations, policy evaluation using Monte Carlo, Policy iteration and Value iteration, Q-Learning, State-Action-Reward-State-Action (SARSA), Model-based Reinforcement Learning.	08
6	**Recommended system,** Collaborative filtering, Content-based filtering Artificial neural network, Perceptron, Multilayer network, Backpropagation, Introduction to Deep learning.	08
	Total	42

LECTURE PLAN:

Unit No./ Total Lecture Reqd.	Topics	Lect. Reqd.	Lect. No.
Unit-I (10)	1. Introduction to subject and scope	1	1
	2. Introduction to learning, Types of learning and Applications	1	2
	3. Supervised Learning	1	3
	4. Linear Regression Model	1	4
	5. Naïve Bayes Classifier	1	5
	6. Decision Tree	1	6
	7. K-nearest Neighbor	1	7
	8. Logistic Regression	1	8
	9. Support Vector Machine	1	9
	10. Random Forest Algorithm	1	10
BC-1	**Logistic Regression Model**	1	11
Unit-II (8)	1. Introduction to clustering, K-mean clustering	2	12
	2. Hierarchical Clustering	1	14
	3. Probabilistic Clustering	1	15
	4. Association Rule Mining	1	16
	5. Apriori Algorithm	1	17
	6. f-p Growth Algorithm	1	18
	7. Gaussian Mixture Model	1	19
Unit-III (8)	1. Feature Extraction- PCA and SVD	3	22
	2. Feature Selection- Feature Ranking and Subset Selection	2	24
	3. Filter, Wrapper and Embedded Methods	1	25
	4. Evaluating Machine Learning Algorithms	1	26
	5. Evaluating Model Selection	1	27
Unit- IV (8)	1. Semi supervised learning: Markov Decision Process (MDP)	2	29
	2. Bellman Equations	1	30
	3. Policy Evaluation using Monte Carlo	1	31
	4. Policy iteration and Value iteration	1	32
	5. Q-Learning	1	33
	6. State-Action-Reward-State-Action (SARSA)	1	34
	7. Model-based Reinforcement Learning	1	35

Unit- V (8)	1. Recommendation system: Collborative Filtering	1	36
	2. Content based filtering	1	37
	3. Artificial neural network	1	38
	4. Perceptron	1	39
	5. Multilayer network	1	40
	6. Backpropagation	1	41
	7. Introduction to Deep learning.	2	42
BC-2	**Genetic Algorithms**	1	44

Text Book:
Machine learning- Tom M Mitchell

Introduction:

Machine learning is a subfield of artificial intelligence (AI). The goal of machine learning generally is to understand the structure of data and fit that data into models that can be understood and utilized by people.

Although machine learning is a field within computer science, it differs from traditional computational approaches. In traditional computing, algorithms are sets of explicitly programmed instructions used by computers to calculate or problem solve. Machine learning algorithms instead allow for computers to train on data inputs and use statistical analysis in order to output values that fall within a specific range. Because of this, machine learning facilitates computers in building models from sample data in order to automate decision-making processes based on data inputs.

Machine Learning Methods:

In machine learning, tasks are generally classified into broad categories. These categories are based on how learning is received or how feedback on the learning is given to the system developed.

Two of the most widely adopted machine learning methods are supervised learning which trains algorithms based on example input and output data that is labeled by humans, and unsupervised learning which provides the algorithm with no labeled data in order to allow it to find structure within its input data.

Supervised Learning

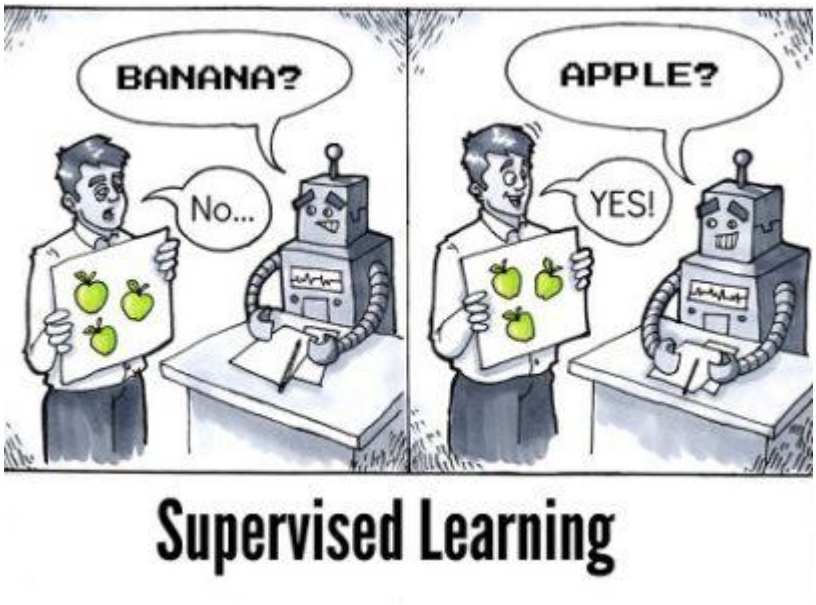
- **Supervised learning:-**Supervised learning is when the model is getting trained on a labelled dataset. Labelled dataset is one which have both input and output parameters. In this type of learning both training and validation datasets are labelled.
- **Learning (training):** Learn a model using the training data.
- **Testing:** Test the model using unseen test data to assess the model accuracy

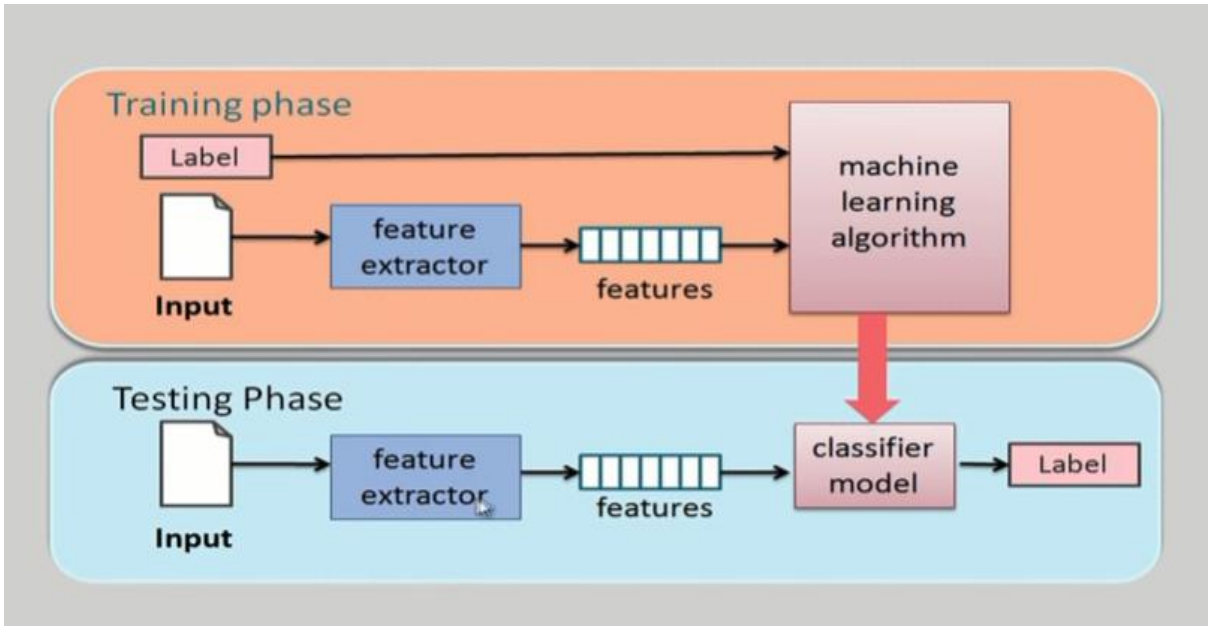
What is Supervised Learning?

Supervised Learning is the process of making an algorithm to learn to map an input to a particular output. This is achieved using the labelled datasets that you have collected. If the mapping is correct, the algorithm has successfully learned. Else, you make the necessary changes to the algorithm so that it can learn correctly. Supervised Learning algorithms can help make predictions for new unseen data that we obtain later in the future.

This is similar to a teacher-student scenario. There is a teacher who guides the student to learn from books and other materials. The student is then tested and if correct, the student

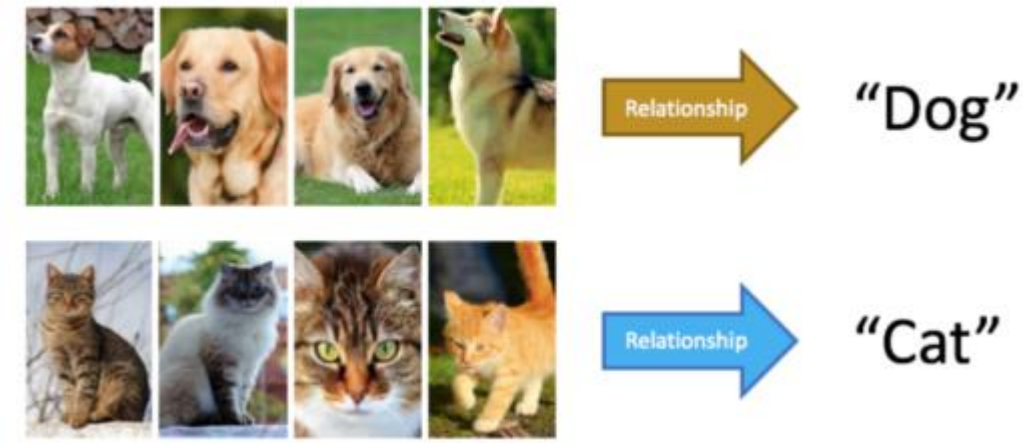
passes. Else, the teacher tunes the student and makes the student learn from the mistakes that he or she had made in the past. That is the basic principle of Supervised Learning.





Example of Supervised Learning

Suppose you have a niece who has just turned 2 years old and is learning to speak. She knows the words, Papa and Mumma, as her parents have taught her how she needs to call them. You want to teach her what a dog and a cat is. So what do you do? You either show her videos of dogs and cats or you bring a dog and a cat and show them to her in real-life so that she can understand how they are different.



Now there are certain things you tell her so that she understands the differences between the 2 animals.

- Dogs and cats both have 4 legs and a tail.
- Dogs come in small to large sizes. Cats, on the other hand, are always small.
- Dogs have a long mouth while cats have smaller mouths.
- Dogs bark while cats meow.
- Different dogs have different ears while cats have almost the same kind of ears.

Now you take your niece back home and show her pictures of different dogs and cats. If she is able to differentiate between the dog and cat, you have successfully taught her.

So what happened here? You were there to guide her to the goal of differentiating between a dog and a cat. You taught her every difference there is between a dog and a cat. You then tested her if she was able to learn. If she was able to learn, she called the dog as a dog and a cat as a cat. If not, you taught her more and were able to teach her. You acted as the supervisor and your niece acted as the algorithm that had to learn. You even knew what was a dog and what was a cat. Making sure that she was learning the correct thing. That is the principle that Supervised Learning follows.

.

Why is it Important?

- Learning gives the algorithm experience which can be used to output the predictions for new unseen data
- Experience also helps in optimizing the performance of the algorithm
- Real-world computations can also be taken care of by the Supervised Learning algorithms

.

Types of Supervised Learning

Supervised Learning has been broadly classified into 2 types.

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

Regression is the kind of Supervised Learning that learns from the Labelled Datasets and is then able to **predict a continuous-valued output** for the new data given to the algorithm. It is used whenever the output required is a number such as money or height etc.

- It is a Supervised Learning task where output is having continuous value.
- Regression means to predict the output value using training data.
- Example in above Figure B, Output – Wind Speed is not having any discrete value but is continuous in the particular range.
- The goal here is to predict a value as much closer to actual output value as our model can and then evaluation is done by calculating error value.
- The smaller the error the greater the accuracy of our regression model

Classification:-

- It is a Supervised Learning task where output is having defined labels(discrete value).
- Classification means to group the output into a class
- For example in below Figure A, Output – Purchased has defined labels i.e. 0 or 1 ; 1 means the customer will purchase and 0 means that customer won't purchase. The goal here is to predict discrete values belonging to a particular class and evaluate on the basis of accuracy.
  It can be either binary or multi class classification. In binary classification, model predicts either 0 or 1 ; yes or no but in case of multi class classification, model predicts more than one class.
  Example: Gmail classifies mails in more than one classes like social, promotions, updates, forum.

Applications of Supervised Learning

Supervised Learning Algorithms are used in a variety of applications. Let's go through some of the most well-known applications.

- **BioInformatics** – This is one of the most well-known applications of Supervised Learning because most of us use it in our day-to-day lives. BioInformatics is the storage of Biological Information of us humans such as fingerprints, iris texture, earlobe and so on. Cellphones of today are capable of learning our biological information and are then able to authenticate us bringing up the security of the system. Smartphones such as iPhones, Google Pixel are capable of facial recognition while OnePlus, Samsung is capable of In-display finger recognition.
- **Speech Recognition** – This is the kind of application where you teach the algorithm about your voice and it will be able to recognize you. The most well-known real-world applications are virtual assistants such as Google Assistant and Siri, which will wake up to the keyword with your voice only.

- **Spam Detection** – This application is used where the unreal or computer-based messages and E-Mails are to be blocked. G-Mail has an algorithm that learns the different keywords which could be fake such as "You are the winner of something" and so forth and blocks those messages directly. OnePlus Messages App gives the user the task of making the application learn which keywords need to be blocked and the app will block those messages with the keyword.
- **Object-Recognition for Vision** – This kind of application is used when you need to identify something. You have a huge dataset which you use to teach your algorithm and this can be used to recognize a new instance. Raspberry Pi algorithms which detect objects are the most well-known example.

.

Supervised vs. Unsupervised Learning

Parameter	Supervised Learning	Unsupervised Learning
Dataset	Labelled	Unlabelled
Method of Learning	Guided learning	The algorithm learns by itself using dataset
Complexity	Simpler method	Computationally complex
Accuracy	More Accurate	Less Accurate

Disadvantages of Supervised Learning

Supervised Learning has a lot of challenges and disadvantages that you could face while working with these algorithms. Let's take a look at these.

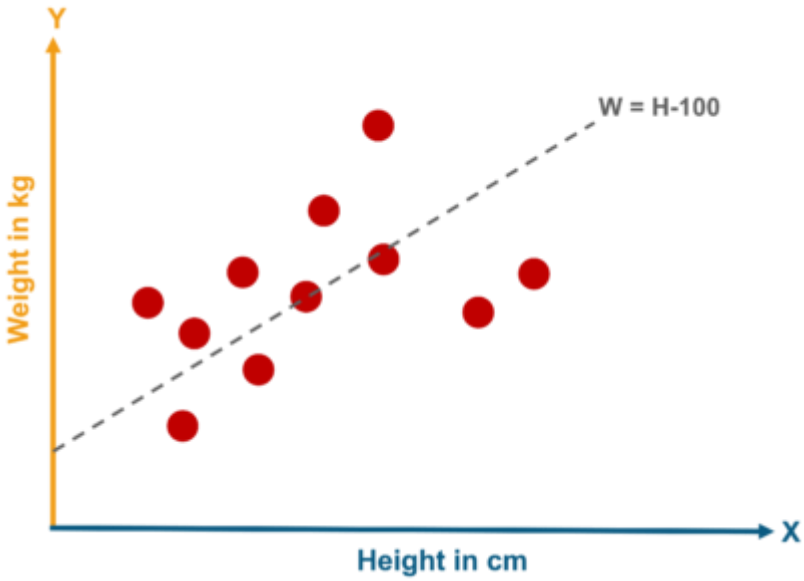
- You could over fit your algorithm easily
- Good examples need to be used to train the data
- Computation time is very large for Supervised Learning
- Unwanted data could reduce the accuracy
- Pre-Processing of data is always a challenge
- If the dataset is incorrect, you make your algorithm learn incorrectly which can bring losses

So now that we have finished all the disadvantages, let's retrace back and summarize what we have learnt today.

We had an overview of what Machine Learning is and its various types. We then understood in depth of what supervised learning is, why is it so important. Later, we went through the various types of supervised Learning which are regression and classification. After that, we discussed the various algorithms, the applications of supervised Learning, differences between Supervised and Unsupervised Learning and the disadvantages that you may face when you work with supervised Learning Algorithms.

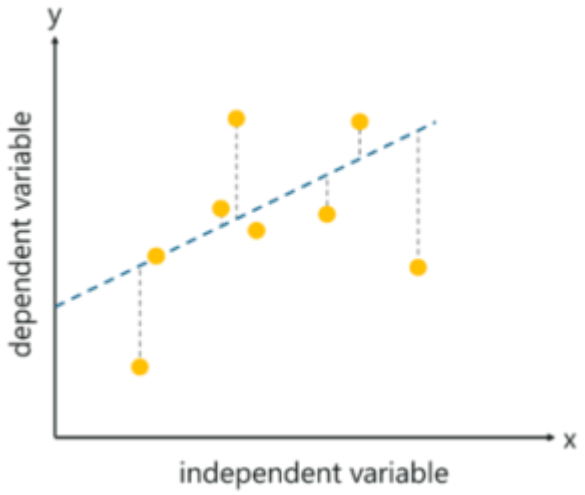
A linear regression is one of the easiest statistical models in machine learning. Understanding its algorithm is a crucial part of the Data Science Certification's course curriculum. It is used

to show the linear relationship between a dependent variable and one or more independent variables.

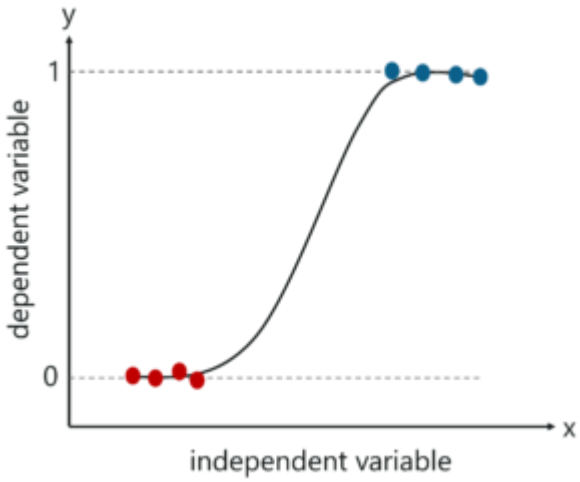


Before we drill down to linear regression in depth, let me just give you a quick overview of what is a regression as Linear Regression is one of a type of Regression algorithm

Linear Regression –This algorithm assumes that there is a linear relationship between the 2 variables, Input (X) and Output (Y), of the data it has learnt from. The Input variable is called the Independent Variable and the Output variable is called the Dependent Variable. When unseen data is passed to the algorithm, it uses the function, calculates and maps the input to a continuous value for the output.



- **Logistic Regression** – This algorithm predicts discrete values for the set of Independent variables that have been passed to it. It does the prediction by mapping the unseen data to the logit function that has been programmed into it. The algorithm predicts the probability of the new data and so it's output lies between the range of 0 and 1.



Classification, on the other hand, is the kind of learning where the algorithm needs to map the new data that is obtained to any one of the 2 classes that we have in our dataset. The classes need to be mapped to either 1 or 0 which in real-life translated to 'Yes' or 'No', 'Rains' or 'Does Not Rain' and so forth. The output will be either one of the classes and not a number as it was in Regression. Some of the most well-known algorithms are discussed below:

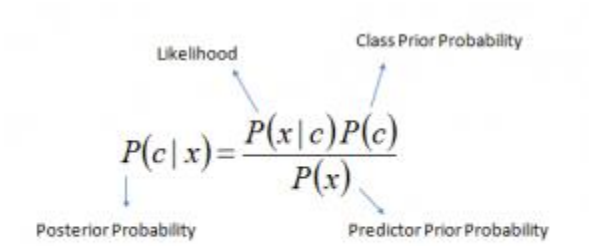
What is Naive Bayes algorithm?

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Above,

- $P(c|x)$ is the posterior probability of *class* (c, *target*) given *predictor* (x, *attributes*).
- $P(c)$ is the prior probability of *class*.
- $P(x|c)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ is the prior probability of *predictor*.

How Naive Bayes algorithm works?

I have a training data set of weather and corresponding target variable 'Play' (suggesting possibilities of playing). Now, we need to classify whether players will play or not based on weather condition. Let's follow the below steps to perform it.

Step 1: Convert the data set into a frequency table

Step 2: Create Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64.

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

Step 3: Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

Problem: Players will play if weather is sunny. Is this statement is correct?

We can solve it using above discussed method of posterior probability.

P(Yes | Sunny) = P(Sunny | Yes) * P(Yes) / P (Sunny)

Here we have P (Sunny |Yes) = 3/9 = 0.33, P(Sunny) = 5/14 = 0.36, P(Yes)= 9/14 = 0.64

Now, P (Yes | Sunny) = 0.33 * 0.64 / 0.36 = 0.60, which has higher probability.

Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes.

What are the Pros and Cons of Naive Bayes?

Pros:

- It is easy and fast to predict class of test data set. It also perform well in multi class prediction
- When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
- It perform well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

Cons:

- If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as "Zero Frequency". To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation.
- On the other side naive Bayes is also known as a bad estimator, so the probability outputs from predict_proba are not to be taken too seriously.
- Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

Applications of Naive Bayes Algorithms

- **Real time Prediction:** Naive Bayes is an eager learning classifier and it is sure fast. Thus, it could be used for making predictions in real time.
- **Multi class Prediction:** This algorithm is also well known for multi class prediction feature. Here we can predict the probability of multiple classes of target variable.
- **Text classification/ Spam Filtering/ Sentiment Analysis:** Naive Bayes classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)
- **Recommendation System:** Naive Bayes Classifier and Collaborative Filtering together builds a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not

Why Decision Tree Algorithm?

Decision Tree is considered to be one of the most useful Machine Learning algorithms since it can be used to solve a variety of problems. Here are a few reasons why you should use Decision Tree:

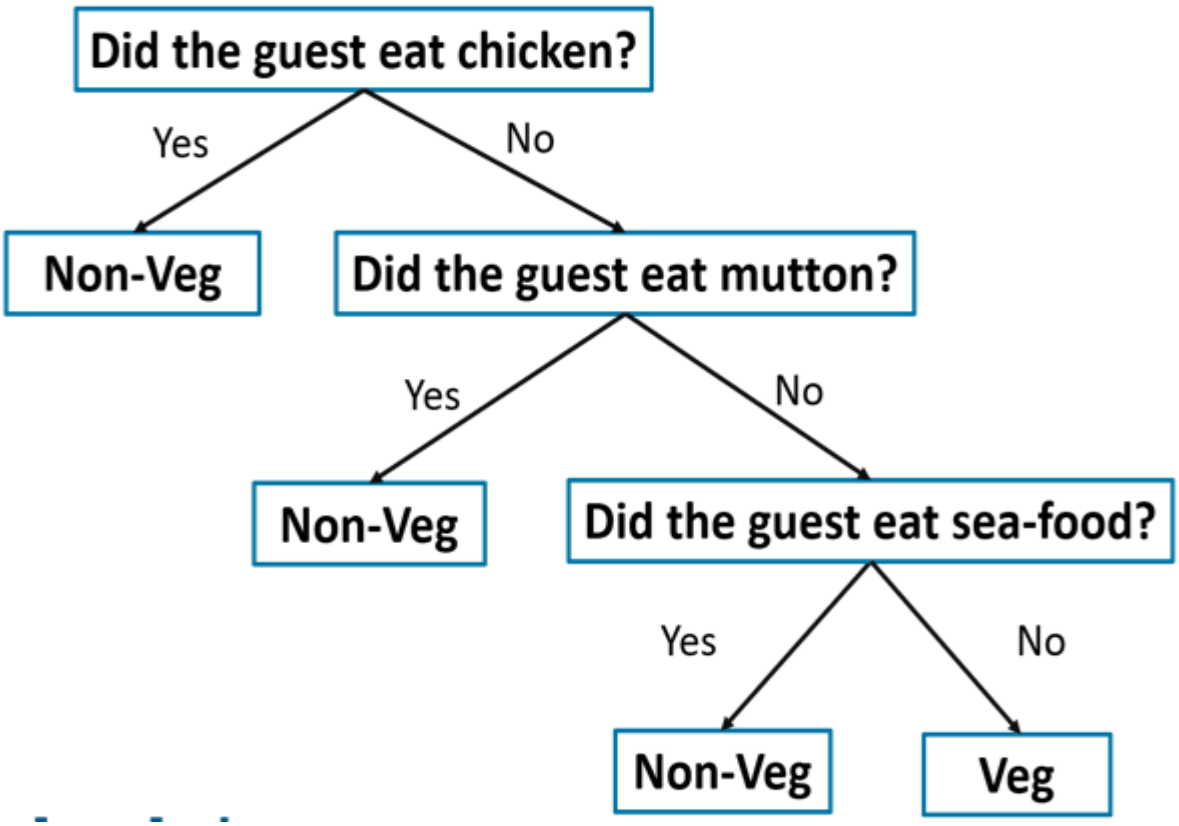
1. It is considered to be the most understandable Machine Learning algorithm and it can be easily interpreted.
2. It can be used for classification and regression problems.
3. Unlike most Machine Learning algorithms, it works effectively with non-linear data.
4. Constructing a Decision Tree is a very quick process since it uses only one feature per node to split the data.

What Is A Decision Tree Algorithm?

A Decision Tree is a Supervised Machine Learning algorithm which looks like an inverted tree, wherein each node represents a predictor variable (feature), the link between the nodes represents a Decision and each leaf node represents an outcome (response variable).

To get a better understanding of a Decision Tree, let's look at an example:

Let's say that you hosted a huge party and you want to know how many of your guests were non-vegetarians. To solve this problem, let's create a simple Decision Tree.

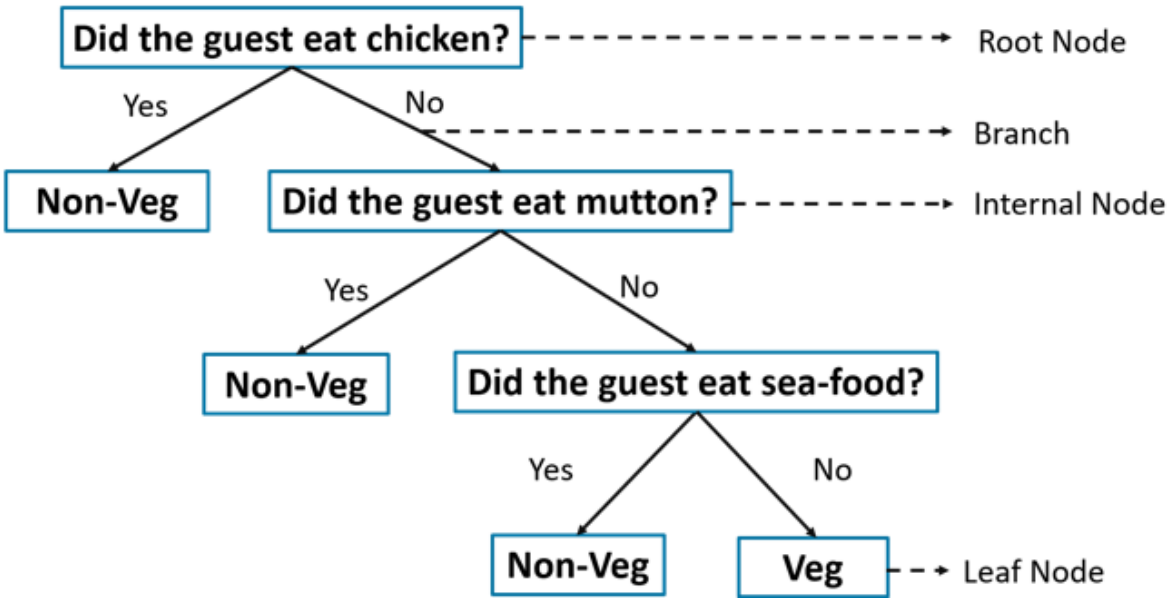


Decision Tree Example – Decision Tree Algorithm

In the above diagram I've created a Decision tree that classifies a guest as either vegetarian or non-vegetarian. Each node represents a predictor variable that will help to conclude whether or not a guest is a non-vegetarian. As you traverse down the tree, you must make decisions at each node, until you reach a dead end.

Now that you know the logic of a Decision Tree, let's define a set of terms related to a Decision Tree.

Structure Of A Decision Tree



Decision Tree Structure – Decision Tree Algorithm

A Decision Tree has the following structure:

- **Root Node:** The root node is the starting point of a tree. At this point, the first split is performed.
- **Internal Nodes:** Each internal node represents a decision point (predictor variable) that eventually leads to the prediction of the outcome.
- **Leaf/ Terminal Nodes:** Leaf nodes represent the final class of the outcome and therefore they're also called terminating nodes.
- **Branches:** Branches are connections between nodes, they're represented as arrows. Each branch represents a response such as yes or no.

So that is the basic structure of a Decision Tree. Now let's try to understand the workflow of a Decision Tree.

How Does The Decision Tree Algorithm Work?

The Decision Tree Algorithm follows the below steps:

Step 1: Select the feature (predictor variable) that best classifies the data set into the desired classes and assign that feature to the root node.
Step 2: Traverse down from the root node, whilst making relevant decisions at each internal node such that each internal node best classifies the data.
Step 3: Route back to step 1 and repeat until you assign a class to the input data.

The above-mentioned steps represent the general workflow of a Decision Tree used for classification purposes.

Now let's try to understand how a Decision Tree is created.

Build A Decision Tree Using ID3 Algorithm

There are many ways to build a Decision Tree, in this blog we'll be focusing on how the ID3 algorithm is used to create a Decision Tree.

What Is The ID3 Algorithm?
ID3 or the Iterative Dichotomiser 3 algorithm is one of the most effective algorithms used to build a Decision Tree. It uses the concept of *Entropy* and *Information Gain* to generate a Decision Tree for a given set of data.

ID3 Algorithm:

The ID3 algorithm follows the below workflow in order to build a Decision Tree:

1. Select **Best Attribute** (A)
2. Assign A as a decision variable for the root node.
3. For each value of A, build a descendant of the node.
4. Assign classification labels to the leaf node.
5. If data is correctly classified: Stop.
6. Else: Iterate over the tree.

The first step in this algorithm states that we must select the best attribute. What does that mean?

The best attribute (predictor variable) is the one that, separates the data set into different classes, most effectively or it is the feature that best splits the data set.

Now the next question in your head must be, "*How do I decide which variable/ feature best splits the data?*"

Two measures are used to decide the best attribute:

1. Information Gain
2. Entropy

What Is Entropy?

Entropy measures the impurity or uncertainty present in the data. It is used to decide how a Decision Tree can split the data.

Equation For Entropy:

$$Entropy = -\sum p(x) \log p(x)$$

What Is Information Gain?

Information Gain (IG) is the most significant measure used to build a Decision Tree. It indicates how much "information" a particular feature/ variable gives us about the final outcome.

Information Gain is important because it used to choose the variable that best splits the data at each node of a Decision Tree. The variable with the highest IG is used to split the data at the root node.

Equation For Information Gain (IG):

$$Information\ Gain = entropy(parent) - [weighted\ average] * entropy(children)$$

how Information Gain and Entropy are used to create a Decision Tree, let's look at an example. The below data set represents the speed of a car based on certain parameters.

Road type	Obstruction	Speed limit	Speed
steep	yes	yes	slow
steep	no	yes	slow
flat	yes	no	fast
steep	no	no	fast

Speed Data Set – Decision Tree Algorithm

Your problem statement is to study this data set and create a Decision Tree that classifies the speed of a car (response variable) as either slow or fast, depending on the following predictor variables:

- Road type
- Obstruction
- Speed limit

We'll be building a Decision Tree using these variables in order to predict the speed of a car. Like I mentioned earlier we must first begin by deciding a variable that best splits the data set and assign that particular variable to the root node and repeat the same thing for the other nodes as well.

At this point, you might be wondering how do you know which variable best separates the data? The answer is, the variable with the highest Information Gain best divides the data into the desired output classes.

So, let's begin by calculating the Entropy and Information Gain (IG) for each of the predictor variables, starting with 'Road type'.

In our data set, there are four observations in the 'Road type' column that correspond to four labels in the 'Speed of car' column. We shall begin by calculating the entropy of the parent node (Speed of car).

Step one is to find out the fraction of the two classes present in the parent node. We know that there are a total of four values present in the parent node, out of which two samples belong to the 'slow' class and the other 2 belong to the 'fast' class, therefore:

- P(slow) -> fraction of 'slow' outcomes in the parent node
- P(fast) -> fraction of 'fast' outcomes in the parent node

The formula to calculate P(slow) is:

p(slow) = no. of 'slow' outcomes in the parent node / total number of outcomes

$$P_{\text{slow}} = \frac{2}{4} = 0.5$$

Similarly, the formula to calculate P(fast) is:

p(fast) = no. of 'fast' outcomes in the parent node / total number of outcomes

$$P_{\text{fast}} = \frac{2}{4} = 0.5$$

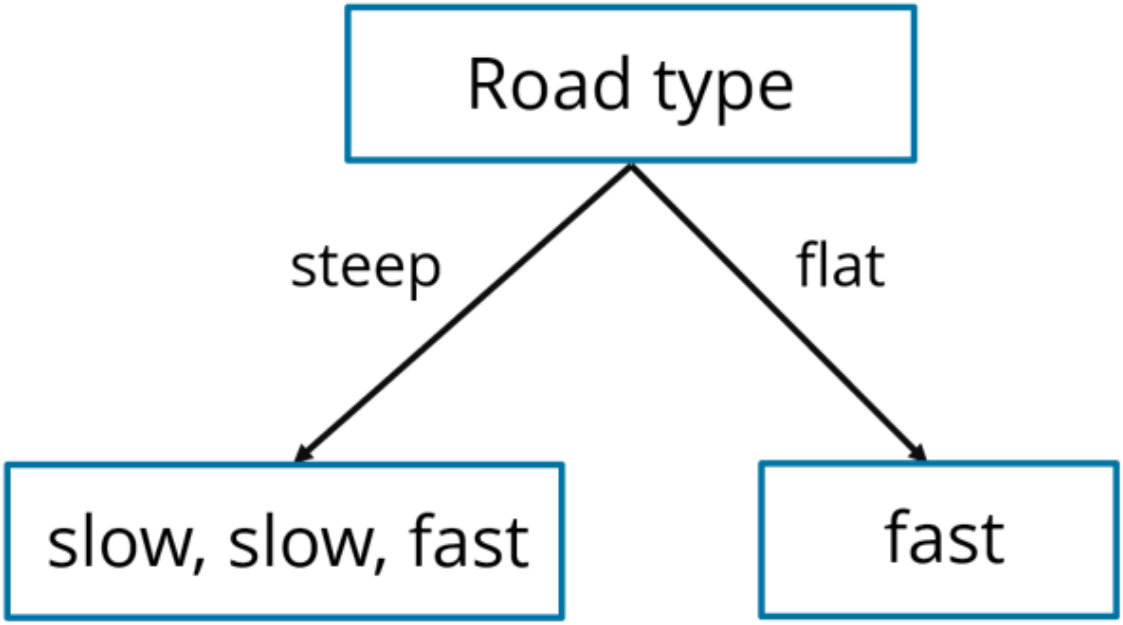
Therefore, the entropy of the parent node is:

$$Entropy_{\text{parent}} = -\Sigma p_{slow} log_2(p_{slow}) + p_{fast} log_2(p_{fast})$$

Entropy(parent) = – {0.5 log2(0.5) + 0.5 log2(0.5)} = – {-0.5 + (-0.5)} = 1

Now that we know that the entropy of the parent node is 1, let's see how to calculate the Information Gain for the 'Road type' variable. Remember that, if the Information gain of the 'Road type' variable is greater than the Information Gain of all the other predictor variables, only then the root node can be split by using the 'Road type' variable.

In order to calculate the Information Gain of 'Road type' variable, we first need to split the root node by the 'Road type' variable.



Decision Tree (Road type) – Decision Tree Algorithm

In the above illustration, we've split the parent node by using the 'Road type' variable, the child nodes denote the corresponding responses as shown in the data set. Now, we need to measure the entropy of the child nodes.

The entropy of the right-hand side child node (fast) is 0 because all of the outcomes in this node belongs to one class (fast). In a similar manner, we must find the Entropy of the left-hand side node (slow, slow, fast).

In this node there are two types of outcomes (fast and slow), therefore, we first need to calculate the fraction of slow and fast outcomes for this particular node.

P(slow) = 2/3 = 0.667
P(fast) = 1/3 = 0.334

Therefore, entropy is:

Entropy(left child node) = – {0.667 log2(0.667) + 0.334 log2(0.334)} = – {-0.38 + (-0.52)} = 0.9

Our next step is to calculate the Entropy(children) with weighted average:

- Total number of outcomes in parent node: 4
- Total number of outcomes in left child node: 3
- Total number of outcomes in right child node: 1

Formula for Entropy(children) with weighted avg. :

[Weighted avg]Entropy(children) = (no. of outcomes in left child node) / (total no. of outcomes in parent node) * (entropy of left node) + (no. of outcomes in right child node)/ (total no. of outcomes in parent node) * (entropy of right node)

By using the above formula you'll find that the, Entropy(children) with weighted avg. is = 0.675

Our final step is to substitute the above weighted average in the IG formula in order to calculate the final IG of the 'Road type' variable:

*Information Gain = entropy(parent) – [weighted average] * entropy(children)*

Therefore,

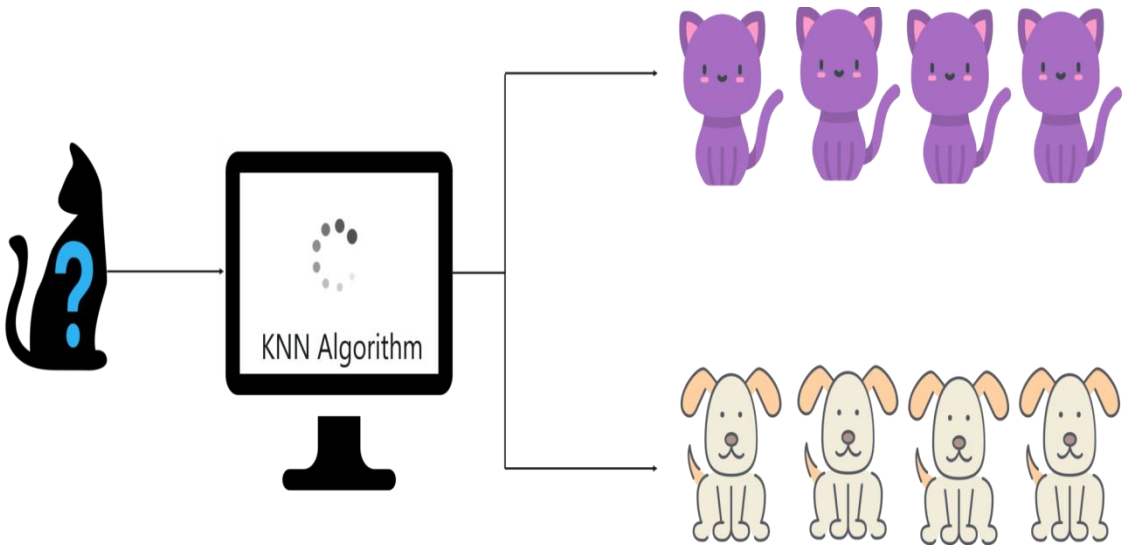
Information gain(Road type) = 1 – 0.675 = 0.325

Information gain of Road type feature is 0.325.

What Is KNN Algorithm?

KNN which stand for K Nearest Neighbor is a Supervised Machine Learning algorithm that classifies a new data point into the target class, depending on the features of its neighboring data points.

Let's try to understand the KNN algorithm with a simple example. Let's say we want a machine to distinguish between images of cats & dogs. To do this we must input a dataset of cat and dog images and we have to train our model to detect the animals based on certain features. For example, features such as pointy ears can be used to identify cats and similarly we can identify dogs based on their long ears.



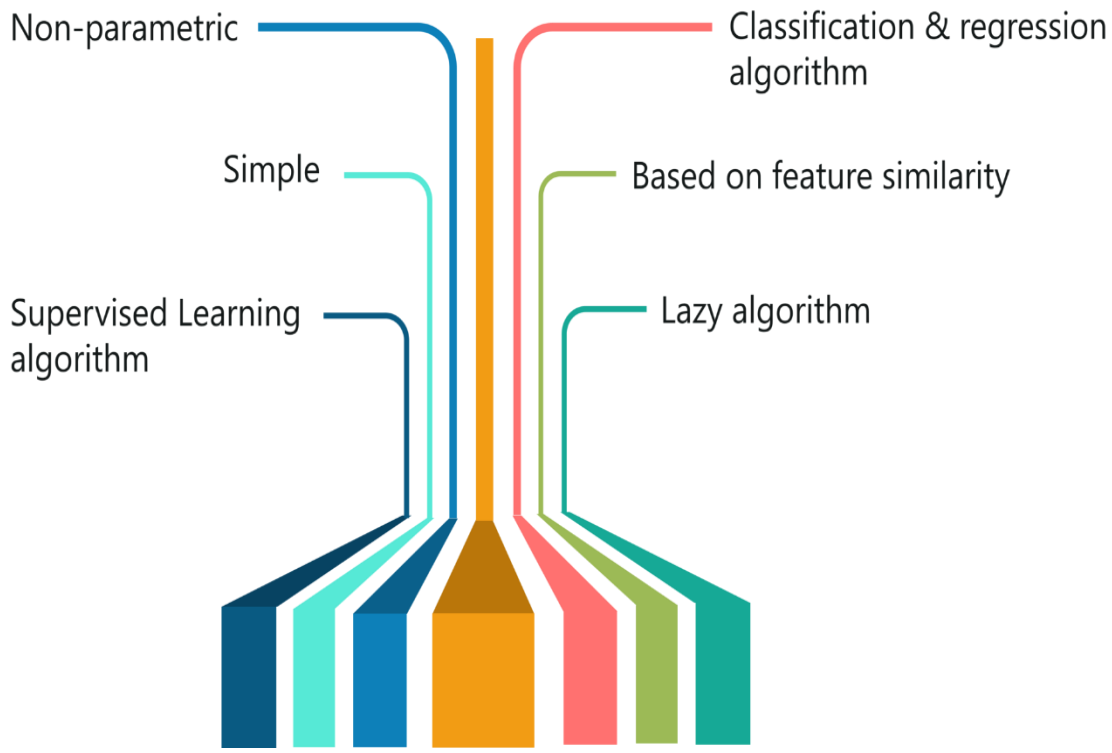
What is KNN Algorithm? – KNN Algorithm In R

the dataset during the training phase, when a new image is given to the model, the KNN algorithm will classify it into either cats or dogs depending on the similarity in their features. So if the new image has pointy ears, it will classify that image as a cat because it is similar to the cat images. In this manner, the KNN algorithm classifies data points based on how similar they are to their neighboring data points.

Features Of KNN Algorithm

The KNN algorithm has the following features:

- KNN is a Supervised Learning algorithm that uses labeled input data set to predict the output of the data points.
- It is one of the most simple Machine learning algorithms and it can be easily implemented for a varied set of problems.
- It is mainly based on feature similarity. KNN checks how similar a data point is to its neighbor and classifies the data point into the class it is most similar to.

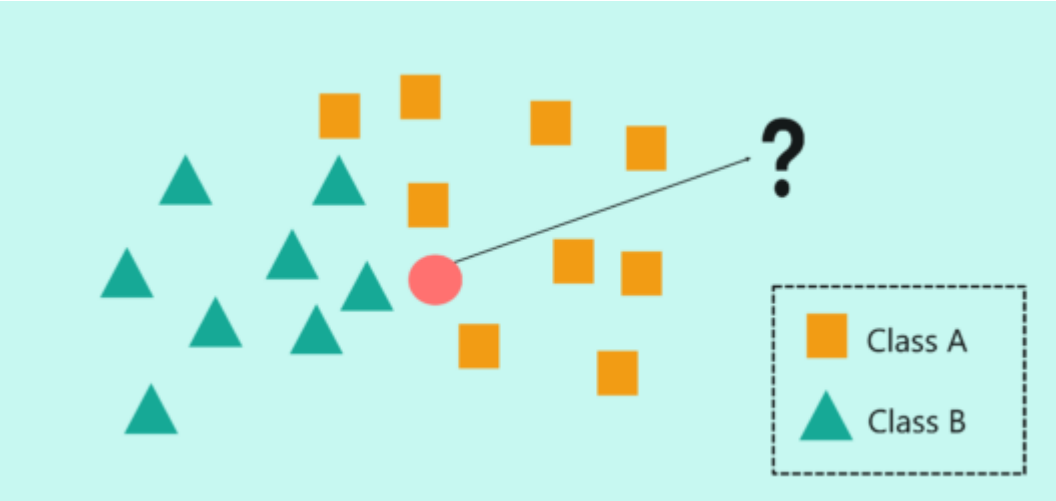


Features of KNN – KNN Algorithm

- Unlike most algorithms, KNN is a non-parametric model which means that it does not make any assumptions about the data set. This makes the algorithm more effective since it can handle realistic data.
- KNN is a lazy algorithm, this means that it memorizes the training data set instead of learning a discriminative function from the training data.
- KNN can be used for solving both classification and regression problems.

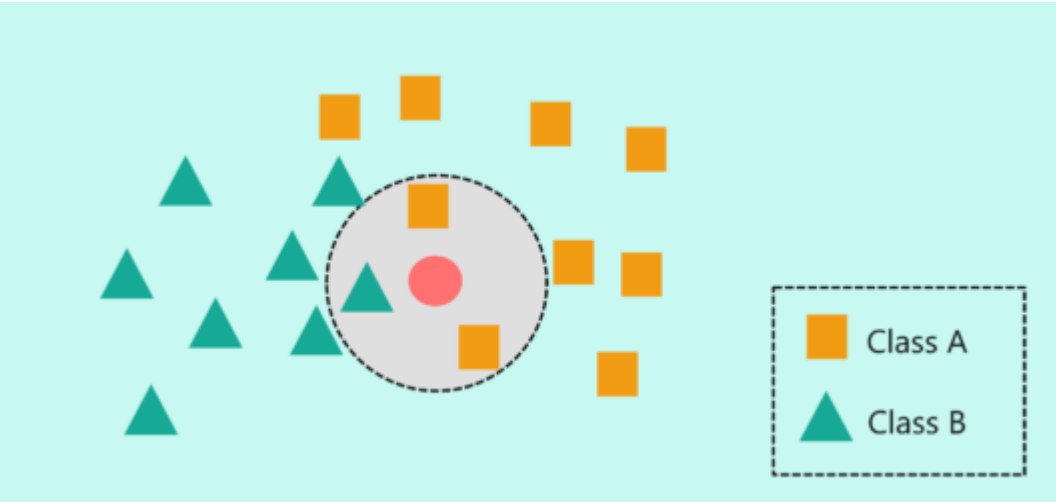
KNN Algorithm Example

To make you understand how KNN algorithm works, let's consider the following scenario:



How does KNN Algorithm work? – KNN Algorithm

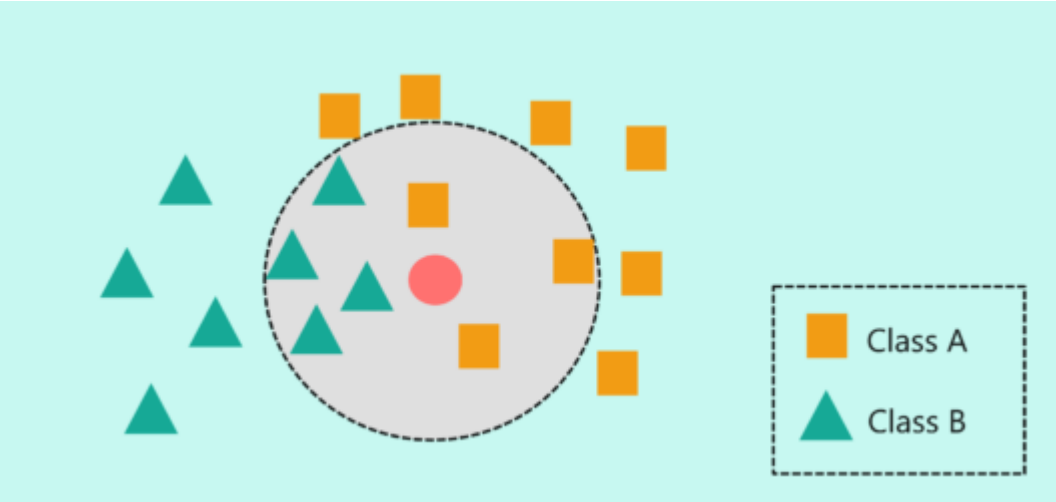
- In the above image, we have two classes of data, namely class A (squares) and Class B (triangles)
- The problem statement is to assign the new input data point to one of the two classes by using the KNN algorithm
- The first step in the KNN algorithm is to define the value of 'K'. But what does the 'K' in the KNN algorithm stand for?
- 'K' stands for the number of Nearest Neighbors and hence the name K Nearest Neighbors (KNN).



How does KNN Algorithm work? – KNN Algorithm

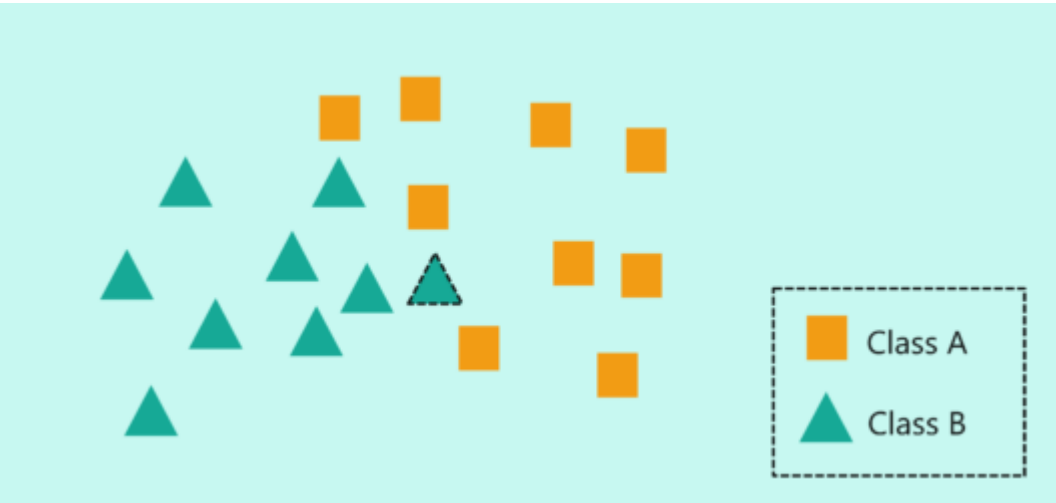
- In the above image, I've defined the value of 'K' as 3. This means that the algorithm will consider the three neighbors that are the closest to the new data point in order to decide the class of this new data point.

- The closeness between the data points is calculated by using measures such as Euclidean and Manhattan distance, which I'll be explaining below.
- At 'K' = 3, the neighbors include two squares and 1 triangle. So, if I were to classify the new data point based on 'K' = 3, then it would be assigned to Class A (squares).



How does KNN Algorithm work? – KNN Algorithm

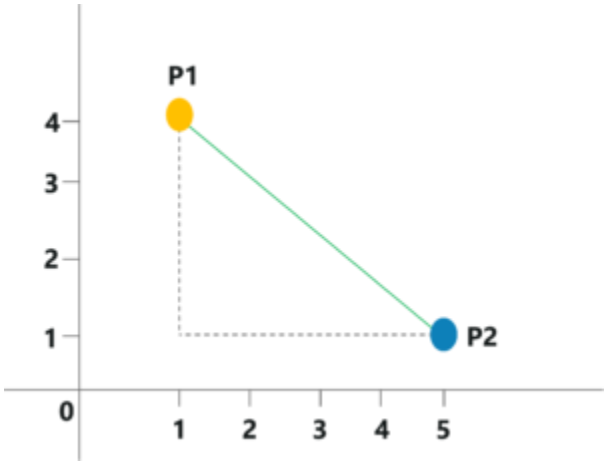
- But what if the 'K' value is set to 7? Here, I'm basically telling my algorithm to look for the seven nearest neighbors and classify the new data point into the class it is most similar to.
- At 'K' = 7, the neighbors include three squares and four triangles. So, if I were to classify the new data point based on 'K' = 7, then it would be assigned to Class B (triangles) since the majority of its neighbors were of class B.



How does KNN Algorithm work? – KNN Algorithm

In practice, there's a lot more to consider while implementing the KNN algorithm. This will be discussed in the demo section of the blog.

Earlier I mentioned that KNN uses Euclidean distance as a measure to check the distance between a new data point and its neighbors, let's see how.



Euclidian Distance – KNN Algorithm

- Consider the above image, here we're going to measure the distance between P1 and P2 by using the Euclidian Distance measure.
- The coordinates for P1 and P2 are (1,4) and (5,1) respectively.
- The Euclidian Distance can be calculated like so:

Point P1 = (1,4)

Point P2 = (5,1)

Euclidian distance = $\sqrt{(5-1)^2 + (4-1)^2} = 5$

Euclidian Distance Calculations – KNN Algorithm

It is as simple as that! KNN makes use of simple measure in order to solve complex problems, this is one of the reasons why KNN is such a commonly used algorithm.

To sum it up, let's look at the pseudocode for KNN Algorithm.

KNN Algorithm Pseudocode

Consider the set, (Xi, Ci),

- Where Xi denotes feature variables and 'i' are data points ranging from i=1, 2, ….., n
- Ci denotes the output class for Xi for each i

The condition, $Ci \in \{1, 2, 3, \ldots\ldots, c\}$ is acceptable for all values of 'i' by assuming that the total number of classes is denoted by 'c'.

Now let's pretend that there's a data point 'x' whose output class needs to be predicted. This can be done by using the K-Nearest Neighbour (KNN) Algorithm.

KNN Algorithm Pseudocode:
Calculate D(x, xi), where 'i' =1, 2, ….., n and 'D' is the Euclidean measure between the data points.
The calculated Euclidean distances must be arranged in ascending order.
Initialize k and take the first k distances from the sorted list.
Figure out the k points for the respective k distances.
Calculate ki, which indicates the number of data points belonging to the ith class among k points i.e. $k \geq 0$
If $ki > kj \forall i \neq j$; put x in class i.
The above pseudocode can be used for solving a classification problem by using the KNN Algorithm.

What is the Support Vector Machine

A Support Vector Machine was first introduced in the 1960s and later improvised in the 1990s. It is a supervised learning machine learning classification algorithm that has become **extremely popular** nowadays owing to its extremely efficient results.

An SVM is implemented in a slightly different way than other machine learning algorithms. It is capable of performing classification, regression and outlier detection.

Support Vector Machine is a discriminative classifier that is formally designed by a separative hyperplane. It is a representation of examples as points in space that are mapped so that the points of different categories are separated by a gap as wide as possible. In addition to this, an SVM can also perform non-linear classification. Let us take a look at how the Support Vector Machine work.

Advantages of SVM

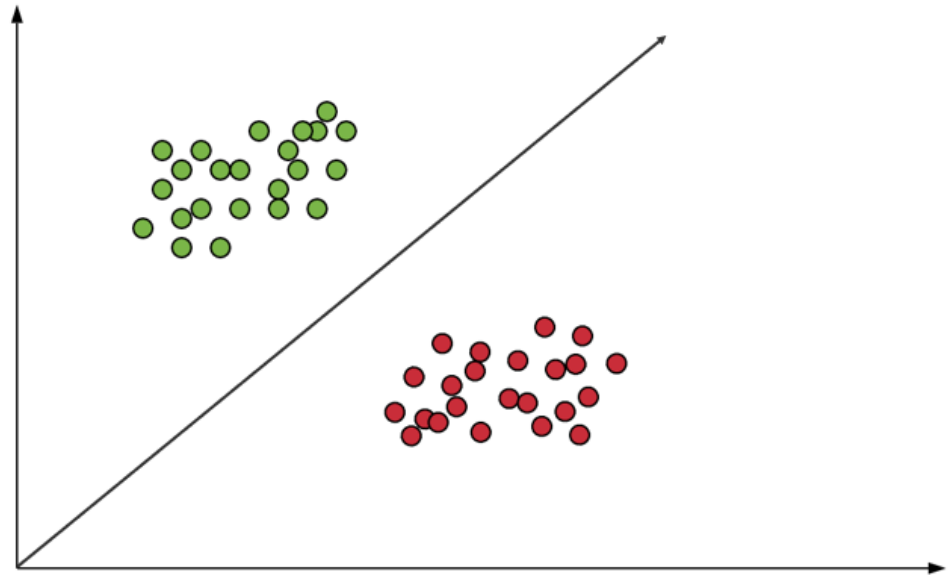
- Effective in high dimensional spaces
- Still effective in cases where the number of dimensions is greater than the number of samples
- Uses a subset of training points in the decision function that makes it memory efficient
- Different kernel functions can be specified for the decision function that also makes it versatile

Disadvantages of SVM

- If the number of features is much larger than the number of samples, avoid over-fitting in choosing kernel functions and regularization term is crucial.
- SVMs do not directly provide probability estimates, these are calculated using five-fold cross-validation.

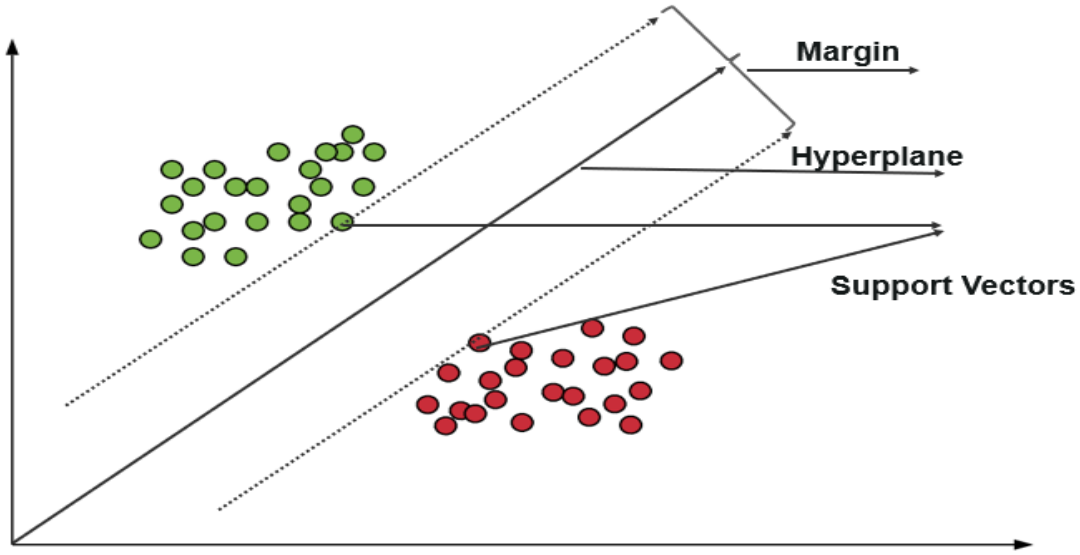
How Does SVM Work?

The main objective of a support vector machine is to segregate the given data in the best possible way. When the segregation is done, the distance between the nearest points is known as the margin. The approach is to select a hyperplane with the maximum possible margin between the support vectors in the given data-sets.



To select the maximum hyperplane in the given sets, the support vector machine follows the following sets:

- Generate hyperplanes which segregates the classes in the best possible way
- Select the right hyperplane with the maximum segregation from either nearest data points



How to deal with inseparable and non-linear planes

In some cases, hyperplanes can not be very efficient. In those cases, the support vector machine uses a kernel trick to *transform the input into a higher-dimensional space*. With this, it becomes easier to segregate the points. Let us take a look at the SVM kernels.

SVM Kernels

An SVM kernel basically adds more dimensions to a low dimensional space to make it easier to segregate the data. It converts the inseparable problem to separable problems by adding more dimensions using the kernel trick. A support vector machine is implemented in practice by a kernel. The kernel trick helps to make a more accurate classifier. Let us take a look at the different kernels in the Support vector machine.

- **Linear Kernel** – A linear kernel can be used as a normal dot product between any two given observations. The product between the two vectors is the sum of the multiplication of each pair of input values. Following is the linear kernel equation.

$$f(x) = B(0) + sum(ai * (x,xi))$$

- **Polynomial Kernel** – It is a rather generalized form of the linear kernel. It can distinguish curved or nonlinear input space. Following is the polynomial kernel equation.

$$K(X_1, X_2) = (a + X_1^T X_2)^b$$

b = degree of kernel & a = constant term.

- **Radial Basis Function Kernel** – The radial basis function kernel is commonly used in SVM classification, it can map the space in infinite dimensions. Following is the

RBF kernel equation.

$$K(X_1, X_2) = exponent(-\gamma ||X_1 - X_2||^2)$$

||X1 — X2 || = Euclidean distance between X1 & X2

Support Vector Machine Use Cases

- Face Detection
- Text And Hyper Text Categorization
- Classification Of Images
- Bioinformatics
- Protein Fold and Remote Homology Detection
- Handwriting Recognition
- Generalized Predictive Control

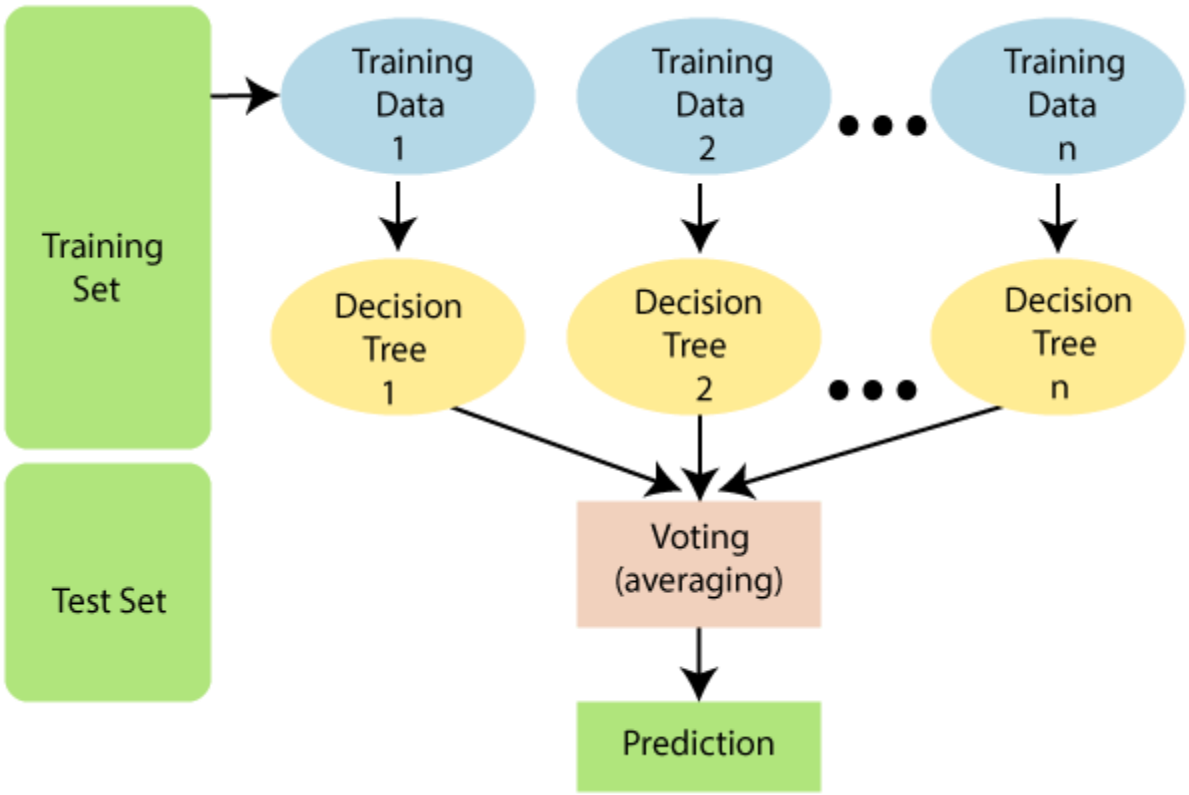
Random Forest Algorithm

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

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

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

The below diagram explains the working of the Random Forest algorithm:



Note: To better understand the Random Forest Algorithm, you should have knowledge of the Decision Tree Algorithm.

Assumptions for Random Forest

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:

- 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).

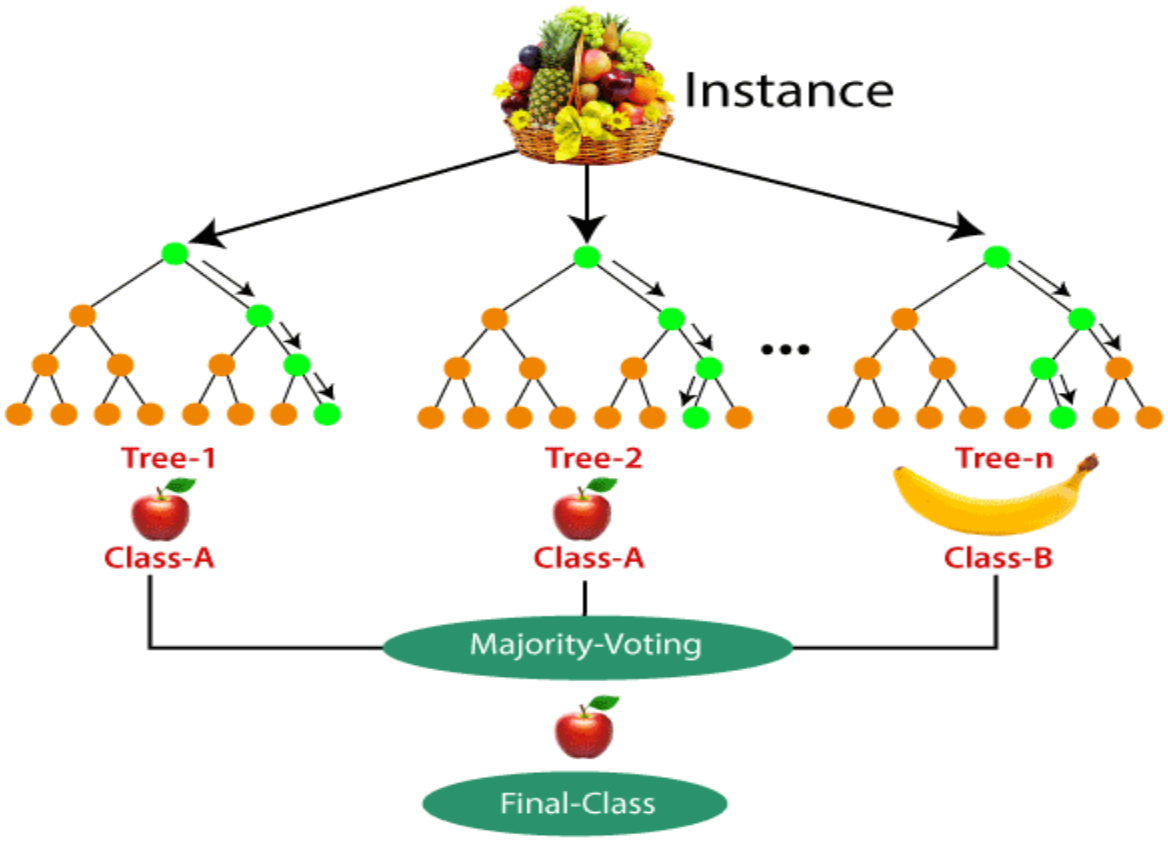
Step-3: Choose the number N for decision trees that you want to build.

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.