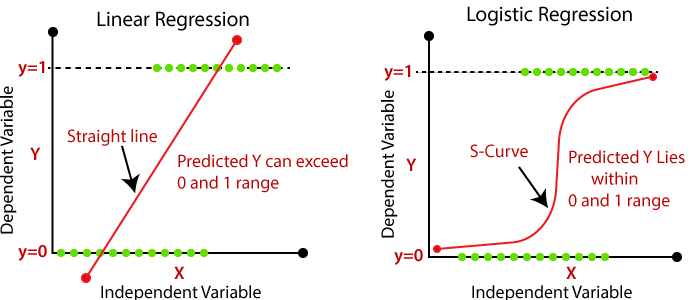
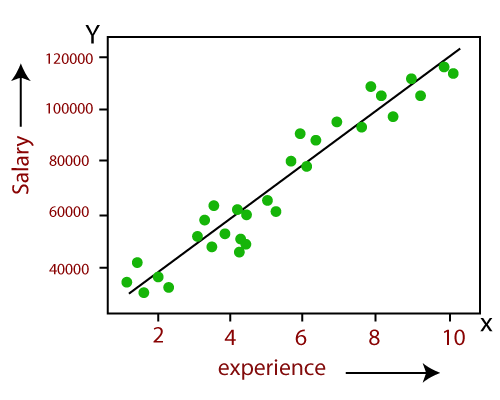
**Linear Regression vs Logistic Regression**

Linear Regression and Logistic Regression are the two famous Machine Learning Algorithms which come under supervised learning technique. Since both the algorithms are of supervised in nature hence these algorithms use labelled dataset to make the predictions. But the main difference between them is how they are being used. The Linear Regression is used for solving Regression problems whereas Logistic Regression is used for solving the Classification problems. The description of both the algorithms is given below along with difference table.



**Linear Regression:**

* Linear Regression is one of the most simple Machine learning algorithm that comes under Supervised Learning technique and used for solving regression problems.
* It is used for predicting the continuous dependent variable with the help of independent variables.
* The goal of the Linear regression is to find the best fit line that can accurately predict the output for the continuous dependent variable.
* If single independent variable is used for prediction then it is called Simple Linear Regression and if there are more than two independent variables then such regression is called as Multiple Linear Regression.
* By finding the best fit line, algorithm establish the relationship between dependent variable and independent variable. And the relationship should be of linear nature.
* The output for Linear regression should only be the continuous values such as price, age, salary, etc. The relationship between the dependent variable and independent variable can be shown in below image:



In above image the dependent variable is on Y-axis (salary) and independent variable is on x-axis(experience). The regression line can be written as:

y= a0+a1x+ ε

Where, a0 and a1 are the coefficients and ε is the error term.

**Logistic Regression:**

* Logistic regression is one of the most popular Machine learning algorithm that comes under Supervised Learning techniques.
* It can be used for Classification as well as for Regression problems, but mainly used for Classification problems.
* Logistic regression is used to predict the categorical dependent variable with the help of independent variables.
* The output of Logistic Regression problem can be only between the 0 and 1.
* Logistic regression can be used where the probabilities between two classes is required. Such as whether it will rain today or not, either 0 or 1, true or false etc.
* Logistic regression is based on the concept of Maximum Likelihood estimation. According to this estimation, the observed data should be most probable.
* In logistic regression, we pass the weighted sum of inputs through an activation function that can map values in between 0 and 1. Such activation function is known as **sigmoid function** and the curve obtained is called as sigmoid curve or S-curve. Consider the below image:



* The equation for logistic regression is:

inear Regression vs Logistic Regression

Difference between Linear Regression and Logistic Regression:

|  |  |
| --- | --- |
| **Linear Regression** | **Logistic Regression** |
| Linear regression is used to predict the continuous dependent variable using a given set of independent variables. | Logistic Regression is used to predict the categorical dependent variable using a given set of independent variables. |
| Linear Regression is used for solving Regression problem. | Logistic regression is used for solving Classification problems. |
| In Linear regression, we predict the value of continuous variables. | In logistic Regression, we predict the values of categorical variables. |
| In linear regression, we find the best fit line, by which we can easily predict the output. | In Logistic Regression, we find the S-curve by which we can classify the samples. |
| Least square estimation method is used for estimation of accuracy. | Maximum likelihood estimation method is used for estimation of accuracy. |
| The output for Linear Regression must be a continuous value, such as price, age, etc. | The output of Logistic Regression must be a Categorical value such as 0 or 1, Yes or No, etc. |
| In Linear regression, it is required that relationship between dependent variable and independent variable must be linear. | In Logistic regression, it is not required to have the linear relationship between the dependent and independent variable. |
| In linear regression, there may be collinearity between the independent variables. | In logistic regression, there should not be collinearity between the independent variable. |

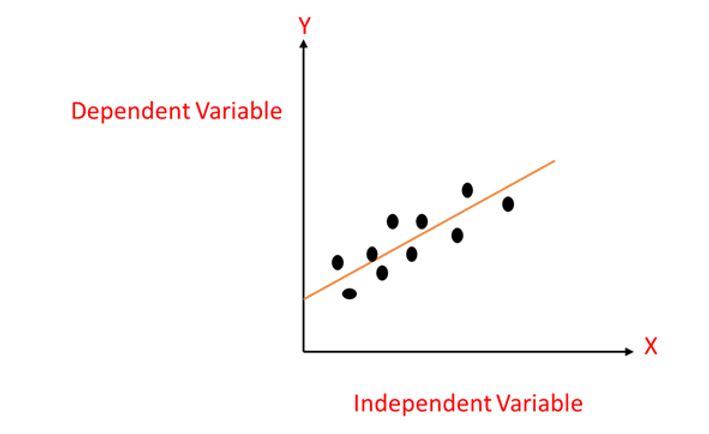
**Regression Analysis**

Regression analysis is a form predictive modelling technique which builds the relationship between two or independent variable and dependent variables.

For example, Let’s say you want to predict the house price and that will depend on various factors like plot size, number of bedrooms, year it is build and so on...

*The dependent features are called dependent variables or outputs.*

*The independent features are called independent variables or inputs*.



**What is Linear regression**

Linear regression in simple term is answering a question on “How can I use X to predict Y?” where X is some information that you have, and Y is some information that you want.

Let’s say you wanted a sell a house and you wanted to know how much you can sell it for. You have information about the house that is your X and the selling price that you wanted to know will be your Y.

Linear regression creates an equation in which you input your given numbers (X) and it outputs the target variable that you want to find out (Y).

**Linear Regression model representation**

Linear regression is such a useful and established algorithm, that it is both a statistical model and a machine learning model. Linear regression tries a draw a best fit line that is close to the data by finding the slope and intercept.

Linear regression equation is,

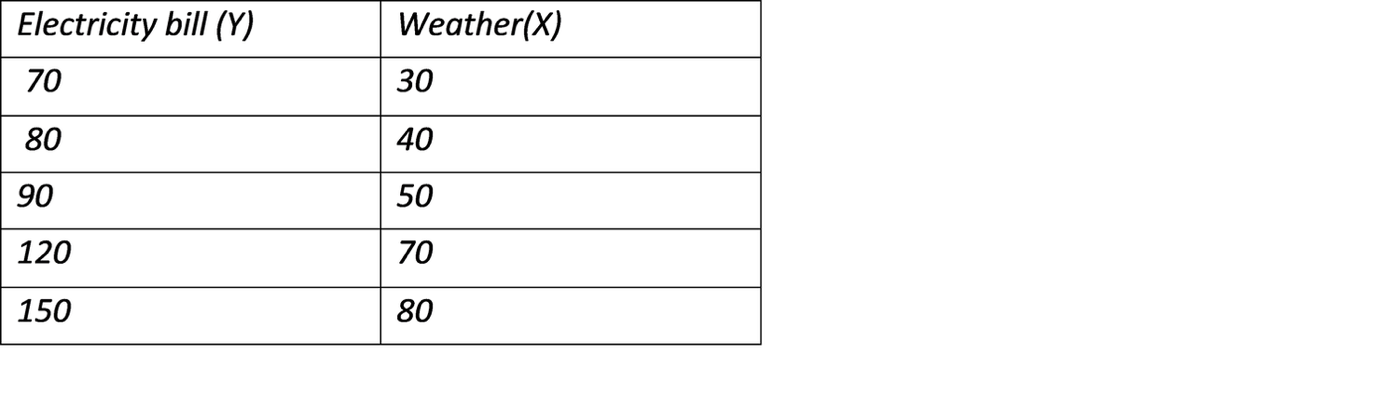
**Y=a+bx**

In this equation:

* y is the output variable. It is also called the target variable in machine learning or the dependent variable.
* x is the input variable. It is also referred to as the feature in machine learning or it is called the independent variable.
* a is the constant
* b is the coefficient of independent variable

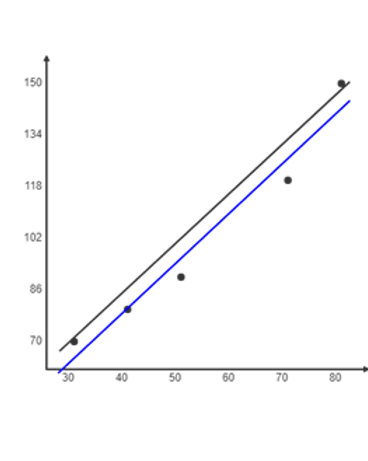
Let’s take an example and see how simple linear regression works

Let’s say we need to predict our electricity bill. The electricity bill will totally depend upon the weather.



Here the weather is given in Fahrenheit and Electricity bill is in dollars...

Lets plot in graph and find the best fit line



Intercept:

18.488372

Slope:

1.546512

Line of Best Fit:

y = 1.546512x + 18.488372.

**Multiple linear regression**

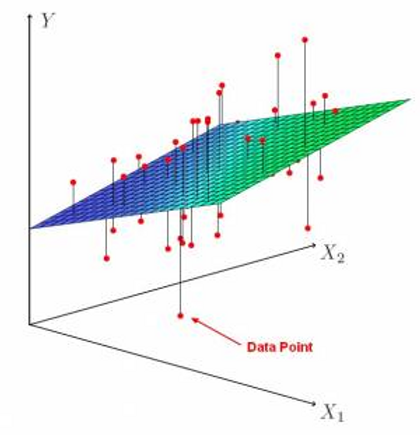
Multiple Linear Regression assumes there is a linear relationship between two or more independent variables and one dependent variable.

The Formula for multiple linear regression:

Y=B0+B0X1+B2X2+……+BnXn+e

* **Y** = the predicted value of the dependent variable
* **B0** = the y-intercept (value of y when all other parameters are set to 0)
* **B1X1**= the regression coefficient (B1) of the first independent variable (**X1**)
* **BnXn** = the regression coefficient of the last independent variable
* **e** = model error

The multiple linear regression model can be represented as a plane (in 2-dimensions) or a hyperplane (in higher dimensions).



**[What is the difference between simple regression and multiple regression?](https://www.quora.com/What-is-the-difference-between-simple-regression-and-multiple-regression?no_redirect=1" \t "_blank)**

“Simple linear regression” is the term used to describe the process of finding a “best fit line” that models the relationship between two variables with a straight line. Typically, one of the variables is designated as the independent variable, usually denoted by X. The other is the dependent variable denoted by Y. This language reflects an assumption that X holds predictive value for Y, that is, knowing X gives you information about what Y is, or rather, what the distribution of Y is at a particular X value. An equation representing this relationship is derived from the data and given in the form Y=b0 + b1X, which is the equation of a straight line (hence: linear).

“Multiple regression” simply means that there are more X variables (independent variables) used in the prediction of Y. For k variables, an equation is derived of the form Y=b0+b1X1+b2X2+…+bkXk. This is still a linear equation.

DECISION TREE:

**ID3 algorithm**, stands for **Iterative Dichotomiser 3**, is a **classification algorithm** that follows a **greedy approach** of **building a decision tree** by selecting a best attribute that yields **maximum Information Gain (IG) or minimum Entropy (H)**.

we will use the **ID3 algorithm to build a decision tree based on a weather data** and illustrate how we can use this procedure to make a decision on an action (like whether to play outside) based on the current data using the previously collected data.

# What is a Decision Tree?

A Supervised Machine Learning Algorithm, used to build classification and regression models in the form of a tree structure.

A decision tree is a tree where each -

* Node - a feature(attribute)
* Branch - a decision(rule)
* Leaf - an outcome(categorical or continuous)

There are many algorithms to build decision trees, here we are going to discuss ID3 algorithm with an example.

# What is an ID3 Algorithm?

ID3 stands for **Iterative Dichotomiser 3**

It is a classification algorithm that follows a greedy approach by selecting a best attribute that yields maximum Information Gain(IG) or minimum Entropy(H).

## What is Entropy and Information gain?

**Entropy is a measure of the amount of uncertainty in the dataset S. Mathematical Representation of Entropy is shown here -**

H ( S ) = ∑ c ∈ C − p ( c ) l o g 2 p ( c )

Where,

* S - The current dataset for which entropy is being calculated(changes every iteration of the ID3 algorithm).
* C - Set of classes in S {example - C ={yes, no}}
* p(c) - The proportion of the number of elements in class c to the number of elements in set S.

In ID3, entropy is calculated for each remaining attribute. The attribute with the smallest entropy is used to split the set S on that particular iteration.

Entropy = 0 implies it is of pure class, that means all are of same category.

**Information Gain IG(A) tells us how much uncertainty in S was reduced after splitting set S on attribute A. Mathematical representation of Information gain is shown here -**

I G ( A , S ) = H ( S ) − ∑ t ∈ T p ( t ) H ( t )

Where,

* H(S) - Entropy of set S.
* T - The subsets created from splitting set S by attribute A such that

S = ⋃ t ϵ T t

* p(t) - The proportion of the number of elements in t to the number of elements in set S.
* H(t) - Entropy of subset t.

In ID3, information gain can be calculated (instead of entropy) for each remaining attribute. The attribute with the largest information gain is used to split the set S on that particular iteration.

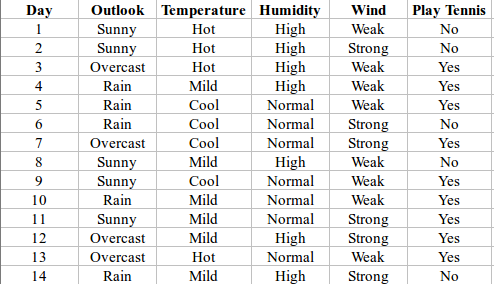
# What are the steps in ID3 algorithm?

The **steps in ID3 algorithm** are as follows:

1. Calculate entropy for dataset.
2. For each attribute/feature.  
   2.1. Calculate entropy for all its categorical values.  
   2.2. Calculate information gain for the feature.
3. Find the feature with maximum information gain.
4. Repeat it until we get the desired tree.

# Use ID3 algorithm on a data

We'll discuss it here mathematically and later see it's implementation in Python.  
So, Let's take an example to make it more clear.



Here,dataset is of binary classes(yes and no), where 9 out of 14 are "yes" and 5 out of 14 are "no".

Complete entropy of dataset is:

H(S) = - p(yes) \* log2(p(yes)) - p(no) \* log2(p(no))

= - (9/14) \* log2(9/14) - (5/14) \* log2(5/14)

= - (-0.41) - (-0.53)

= 0.94

For each attribute of the dataset, let's follow the step-2 of pseudocode : -

## First Attribute - Outlook

Categorical values - sunny, overcast and rain

H(Outlook=sunny) = -(2/5)\*log(2/5)-(3/5)\*log(3/5) =0.971

H(Outlook=rain) = -(3/5)\*log(3/5)-(2/5)\*log(2/5) =0.971

H(Outlook=overcast) = -(4/4)\*log(4/4)-0 = 0

Average Entropy Information for Outlook -

I(Outlook) = p(sunny) \* H(Outlook=sunny) + p(rain) \* H(Outlook=rain) + p(overcast) \* H(Outlook=overcast)

= (5/14)\*0.971 + (5/14)\*0.971 + (4/14)\*0

= 0.693

Information Gain = H(S) - I(Outlook)

= 0.94 - 0.693

= 0.247

I suggest you to do the same calculations for all the remaining attributes without scrolling down....

### Second Attribute - Temperature

Categorical values - hot, mild, cool

H(Temperature=hot) = -(2/4)\*log(2/4)-(2/4)\*log(2/4) = 1

H(Temperature=cool) = -(3/4)\*log(3/4)-(1/4)\*log(1/4) = 0.811

H(Temperature=mild) = -(4/6)\*log(4/6)-(2/6)\*log(2/6) = 0.9179

Average Entropy Information for Temperature -

I(Temperature) = p(hot)\*H(Temperature=hot) + p(mild)\*H(Temperature=mild) + p(cool)\*H(Temperature=cool)

= (4/14)\*1 + (6/14)\*0.9179 + (4/14)\*0.811

= 0.9108

Information Gain = H(S) - I(Temperature)

= 0.94 - 0.9108

= 0.0292

### Third Attribute - Humidity

Categorical values - high, normal

H(Humidity=high) = -(3/7)\*log(3/7)-(4/7)\*log(4/7) = 0.983

H(Humidity=normal) = -(6/7)\*log(6/7)-(1/7)\*log(1/7) = 0.591

Average Entropy Information for Humidity -

I(Humidity) = p(high)\*H(Humidity=high) + p(normal)\*H(Humidity=normal)

= (7/14)\*0.983 + (7/14)\*0.591

= 0.787

Information Gain = H(S) - I(Humidity)

= 0.94 - 0.787

= 0.153

### Fourth Attribute - Wind

Categorical values - weak, strong

H(Wind=weak) = -(6/8)\*log(6/8)-(2/8)\*log(2/8) = 0.811

H(Wind=strong) = -(3/6)\*log(3/6)-(3/6)\*log(3/6) = 1

Average Entropy Information for Wind -

I(Wind) = p(weak)\*H(Wind=weak) + p(strong)\*H(Wind=strong)

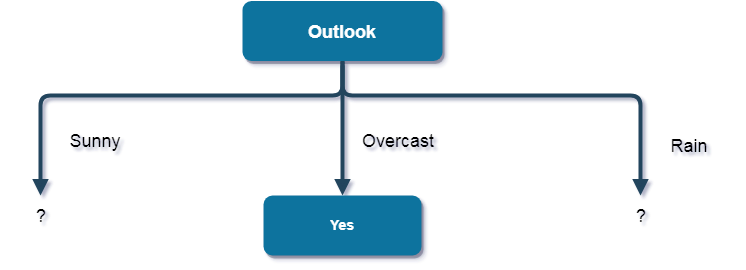
= (8/14)\*0.811 + (6/14)\*1

= 0.892

Information Gain = H(S) - I(Wind)

= 0.94 - 0.892

= 0.048

Here, the attribute with maximum information gain is Outlook. So, the decision tree built so far -  


Here, when Outlook == overcast, it is of pure class(Yes).  
Now, we have to repeat same procedure for the data with rows consist of Outlook value as Sunny and then for Outlook value as Rain.

**I again recommend you to perform further calculations and then cross check by scrolling down...**

Now, finding the best attribute for splitting the data with Outlook=Sunny values{ Dataset rows = [1, 2, 8, 9, 11]}.

Complete entropy of Sunny is -

H(S) = - p(yes) \* log2(p(yes)) - p(no) \* log2(p(no))

= - (2/5) \* log2(2/5) - (3/5) \* log2(3/5)

= 0.971

### First Attribute - Temperature

Categorical values - hot, mild, cool

H(Sunny, Temperature=hot) = -0-(2/2)\*log(2/2) = 0

H(Sunny, Temperature=cool) = -(1)\*log(1)- 0 = 0

H(Sunny, Temperature=mild) = -(1/2)\*log(1/2)-(1/2)\*log(1/2) = 1

Average Entropy Information for Temperature -

I(Sunny, Temperature) = p(Sunny, hot)\*H(Sunny, Temperature=hot) + p(Sunny, mild)\*H(Sunny, Temperature=mild) + p(Sunny, cool)\*H(Sunny, Temperature=cool)

= (2/5)\*0 + (1/5)\*0 + (2/5)\*1

= 0.4

Information Gain = H(Sunny) - I(Sunny, Temperature)

= 0.971 - 0.4

= 0.571

### Second Attribute - Humidity

Categorical values - high, normal

H(Sunny, Humidity=high) = - 0 - (3/3)\*log(3/3) = 0

H(Sunny, Humidity=normal) = -(2/2)\*log(2/2)-0 = 0

Average Entropy Information for Humidity -

I(Sunny, Humidity) = p(Sunny, high)\*H(Sunny, Humidity=high) + p(Sunny, normal)\*H(Sunny, Humidity=normal)

= (3/5)\*0 + (2/5)\*0

= 0

Information Gain = H(Sunny) - I(Sunny, Humidity)

= 0.971 - 0

= 0.971

### Third Attribute - Wind

Categorical values - weak, strong

H(Sunny, Wind=weak) = -(1/3)\*log(1/3)-(2/3)\*log(2/3) = 0.918

H(Sunny, Wind=strong) = -(1/2)\*log(1/2)-(1/2)\*log(1/2) = 1

Average Entropy Information for Wind -

I(Sunny, Wind) = p(Sunny, weak)\*H(Sunny, Wind=weak) + p(Sunny, strong)\*H(Sunny, Wind=strong)

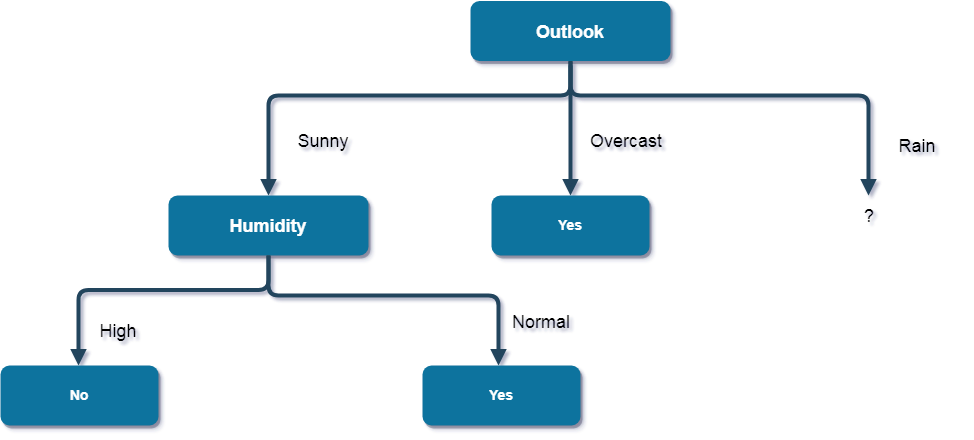
= (3/5)\*0.918 + (2/5)\*1

= 0.9508

Information Gain = H(Sunny) - I(Sunny, Wind)

= 0.971 - 0.9508

= 0.0202

Here, the attribute with maximum information gain is Humidity. So, the decision tree built so far -  


Here, when Outlook = Sunny and Humidity = High, it is a pure class of category "no". And When Outlook = Sunny and Humidity = Normal, it is again a pure class of category "yes". Therefore, we don't need to do further calculations.

Now, finding the best attribute for splitting the data with Outlook=Sunny values{ Dataset rows = [4, 5, 6, 10, 14]}.

Complete entropy of Rain is -

H(S) = - p(yes) \* log2(p(yes)) - p(no) \* log2(p(no))

= - (3/5) \* log(3/5) - (2/5) \* log(2/5)

= 0.971

### First Attribute - Temperature

Categorical values - mild, cool

H(Rain, Temperature=cool) = -(1/2)\*log(1/2)- (1/2)\*log(1/2) = 1

H(Rain, Temperature=mild) = -(2/3)\*log(2/3)-(1/3)\*log(1/3) = 0.918

Average Entropy Information for Temperature -

I(Rain, Temperature) = p(Rain, mild)\*H(Rain, Temperature=mild) + p(Rain, cool)\*H(Rain, Temperature=cool)

= (2/5)\*1 + (3/5)\*0.918

= 0.9508

Information Gain = H(Rain) - I(Rain, Temperature)

= 0.971 - 0.9508

= 0.0202

### Second Attribute - Wind

Categorical values - weak, strong

H(Wind=weak) = -(3/3)\*log(3/3)-0 = 0

H(Wind=strong) = 0-(2/2)\*log(2/2) = 0

Average Entropy Information for Wind -

I(Wind) = p(Rain, weak)\*H(Rain, Wind=weak) + p(Rain, strong)\*H(Rain, Wind=strong)

= (3/5)\*0 + (2/5)\*0

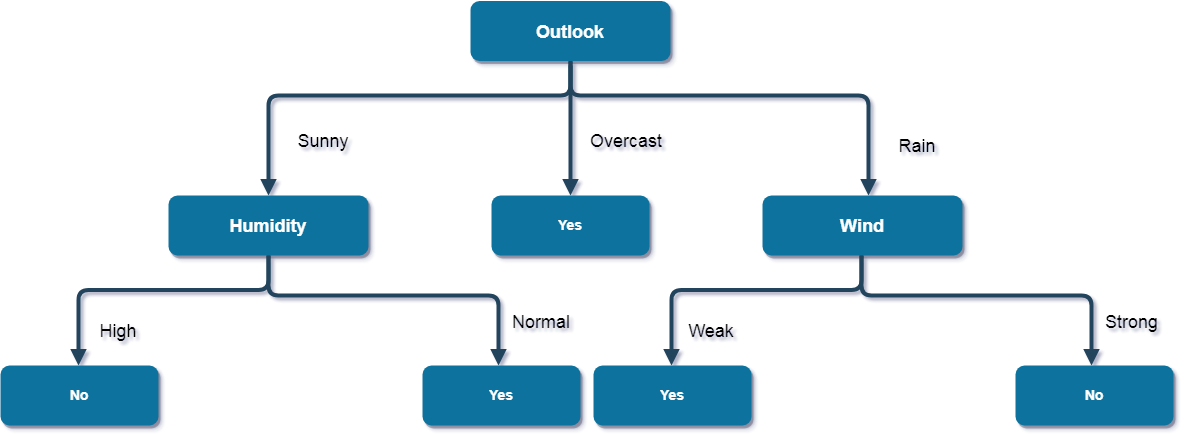
= 0

Information Gain = H(Rain) - I(Rain, Wind)

= 0.971 - 0

= 0.971

Here, the attribute with maximum information gain is Wind. So, the decision tree built so far -



Here, when Outlook = Rain and Wind = Strong, it is a pure class of category "no". And When Outlook = Rain and Wind = Weak, it is again a pure class of category "yes".  
**And this is our final desired tree for the given dataset.**

# What are the characteristics of ID3 algorithm?

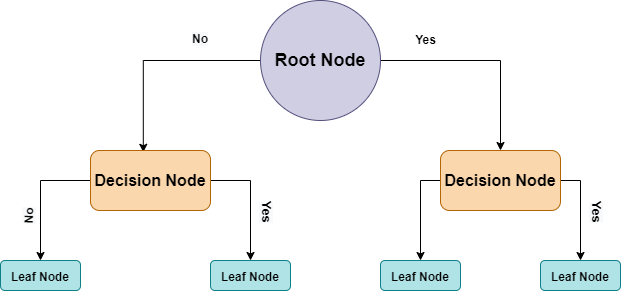
Characteristics of ID3 Algorithm are as follows:

1. ID3 uses a greedy approach that's why it does not guarantee an optimal solution; it can get stuck in local optimums.
2. ID3 can overfit to the training data (to avoid overfitting, smaller decision trees should be preferred over larger ones).
3. This algorithm usually produces small trees, but it does not always produce the smallest possible tree.
4. ID3 is harder to use on continuous data (if the values of any given attribute is continuous, then there are many more places to split the data on this attribute, and searching for the best value to split by can be time consuming).

**What is a Decision Tree?**

Decision trees are one of the predictive modeling approaches used in machine learning. It uses a decision tree to travel from observations about an object (represented by the branches) to inferences about the item’s target value (represented by the leaves) (as a predictive model)

A decision tree’s main idea is to locate the features that contain the most information about the target feature and then split the dataset along with their values. The characteristic that best isolates the uncertainty from knowledge about the target feature is the most informative. The search for the most informative attribute continues until all we have are pure leaf nodes.



**Decision Tree Terminologies**

**Root Node**: Represents the entire sample. This will further get divided into two or more homogeneous sets.

**Decision Node:** Nodes Branched from Root nodes are Decision nodes.

**Branch:** Formed by splitting the tree.

To summarize, The inputs are routed through the root node of every tree. This root node is further segmented into decision nodes that are conditionally dependent on results and observations.

The process of splitting a single node into many nodes is known as splitting. A leaf node, also known as a terminal node, is a node that does not break into other nodes. A branch, sometimes known as a sub-tree, is a section of a decision tree. Splitting is not the only concept that is diametrically opposite it.

Decision trees classify cases by sorting them from the root to some leaf/terminal node, with the leaf/terminal node categorizing the example. Each node in the tree is a test case for a property, and each edge descending from it represents one of the test case’s possible solutions. This is a recursive procedure that is carried out for each new node-rooted subtree.

**How do you split nodes in a Decision tree?**

Although their algorithms differ from those used in classification and regression trees, decision trees completely depend on the objective variable. There are a variety of methods for selecting how to partition the data.

The essence of decision trees is that they divide data sets into sections, resulting in an inverted decision tree with root nodes at the top. Through the pass-over nodes of the trees, the layered model of the decision tree leads to the end outcome.

Each node has an attribute (feature) that acts as the catalyst for further splitting in the downward direction.

Multiple features are included in the decision-making process, and it is necessary to consider the relevance and repercussions of each feature, thereby assigning the relevant feature at the root node and traversing the node splitting downward.

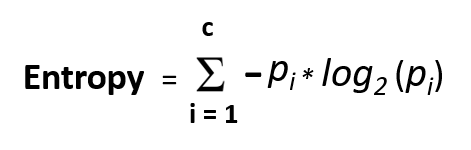
**Methods to split Decision Tree**

To address the significant concerns described above, there are some key splitting parameters. Yes, we shall discuss **Entropy, Gini Index**, and **Information Gain** within the scope of this post.

**Entropy**

Entropy is a measure of purity or the degree of uncertainty, impurity, or disorder of a random variable. It is, in essence, the assessment of impurity or unpredictability in data points

If all of the elements belong to the same class, the distribution is called “Pure,” and if they don’t, it’s called “Impurity”.

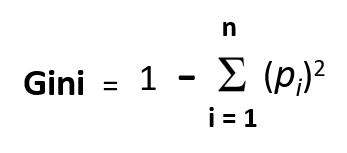


To put it another way, a high order of disorder indicates a low level of impurity. Entropy is a measure of disorder that ranges from 0 to 1. It can be higher than 1 depending on the number of groups or classes present in the data collection, but it has the same meaning.

**Gini Impurity**

If all elements are accurately split into different classes, the division is called pure (an ideal scenario). The Gini impurity (pronounced “genie”) is used to predict the likelihood of a randomly chosen example being incorrectly classified by a particular node. It’s referred to as an “impurity” measure because it demonstrates how the model departs from a simple division.

Gini impurity is measured on a scale of 0 to 1, with 0 indicating that all elements belong to the same class and 1 indicating that only one class exists. A Gini impurity of 1 suggests that all items are scattered randomly across various classes, whereas a value of 0.5 shows that the elements are distributed uniformly across some classes.



Now that we have seen what Gini Impurity is? let us see how to calculate it.

* Calculate Gini coefficients for sub-nodes using the success(p) and failure(q) formulas (p2+q2)
* Next, Calculate the impurity for each node using a weighted Gini score.

**Information Gain**

When it comes to measuring information gain, the concept of entropy is key.  “Information gain, on the other hand, is based on information theory.” “Information gain” refers to the process of identifying the most important features/attributes that convey the most information about a class. The entropy principle is followed with the goal of reducing entropy from the root node to the leaf nodes. Information gain is the difference in entropy before and after splitting, which describes the impurity of in-class items.

**Information Gain = 1-Entropy**

The entropy generally changes when we use a node in a decision tree to partition the training instances into smaller subsets. Information gain is a metric for entropy change.

The more information there is, the higher the entropy.

Now that we have seen what Information Gain is? Let us see how to calculate it.

* For each split, calculate the entropy of each child node independently
* Calculate the entropy of each split using the weighted average entropy of child nodes
* Choose the split with the lowest entropy or the greatest gain in information
* Repeat these steps to obtain homogeneous split nodes

Now, let us compare Information Gain and Gini Impurity

**Information Gain Vs Gini Impurity**

We’ll go over some comparison points gleaned from the preceding discussion to assist in deciding which strategy to adopt.

* The likelihood of a class is multiplied by the log base 2 of that class’s probability to calculate information gain. Gini impurity is determined by subtracting the total of each class’s squared probability from one.
* The Gini Impurity prefers larger partitions (distributions) and is easy to apply, whereas information gains prefer smaller partitions (distributions) with a wide range of values, needing a data and splitting criterion experiment.
* CART algorithms employ the Gini Index approach, whereas ID3, C4.5 methods employ the Information Gain method
* In contrast to the Gini index, which computes the difference between entropy before and after the split and indicates impurity in classes of elements, Information Gain computes the difference between entropy before and after the split and indicates impurity in classes of elements.

## ****What is the Naive Bayes Classifier?****

The Naive Bayes classifier separates data into different classes according to the Bayes’ Theorem, along with the assumption that all the predictors are independent of one another. It assumes that a particular feature in a class is not related to the presence of other features.

**Advantages of Naive Bayes**

The **Naive Bayes is a popular algorithm** due to its following advantages:

* This algorithm works very fast and can easily predict the class of a test dataset.
* You can use it to solve multi-class prediction problems as it’s quite useful with them.
* Naive Bayes classifier performs better than other models with less training data if the assumption of independence of features holds.
* If you have categorical input variables, the Naive Bayes algorithm performs exceptionally well in comparison to numerical variables.
* It can be used for Binary and Multi-class Classifications.
* It effectively works in Multi-class predictions.

Now let’s go through the **disadvantages of Naive Bayes classifier MCQ.**

**Disadvantages of Naive Bayes**

* If your test data set has a categorical variable of a category that wasn’t present in the training data set, the Naive Bayes model will assign it zero probability and won’t be able to make any predictions in this regard. This phenomenon is called ‘Zero Frequency,’ and you’ll have to use a smoothing technique to solve this problem.
* This algorithm is also notorious as a lousy estimator. So, you shouldn’t take the probability outputs of ‘predict\_proba’ too seriously.
* It assumes that all the features are independent. While it might sound great in theory, in real life, you’ll hardly find a set of independent features.

After understanding these **disadvantages of Naive Bayes classifier MCQ**, you can now better understand this algorithm’s applications.

**Applications of Naive Bayes Algorithm**

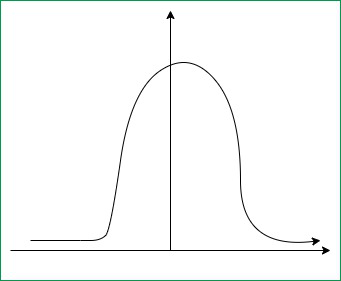
As you must’ve noticed, this algorithm offers plenty of advantages to its users. That’s why it has a lot of applications in various sectors too. Here are some applications of Naive Bayes algorithm:

* As this algorithm is fast and efficient, you can use it to make real-time predictions.
* This algorithm is popular for multi-class predictions. You can find the probability of multiple target classes easily by using this algorithm.
* Email services (like Gmail) use this algorithm to figure out whether an email is a spam or not. This algorithm is excellent for spam filtering.
* Its assumption of feature independence, and its effectiveness in solving multi-class problems, makes it perfect for performing Sentiment Analysis. Sentiment Analysis refers to the identification of positive or negative sentiments of a target group (customers, audience, etc.)
* Collaborative Filtering and the Naive Bayes algorithm work together to build recommendation systems. These systems use data mining and machine learning to predict if the user would like a particular resource or not.

**SVM:**

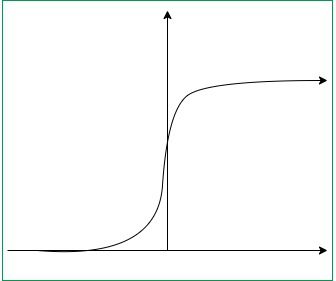
**Kernel Function** is a method used to take data as input and transform it into the required form of processing data. “Kernel” is used due to a set of mathematical functions used in Support Vector Machine providing the window to manipulate the data. So, Kernel Function generally transforms the training set of data so that a non-linear decision surface is able to transform to a linear equation in a higher number of dimension spaces. Basically, It returns the inner product between two points in a standard feature dimension.   
**Standard Kernel Function Equation :**   
**Major Kernel Functions :-**   
For Implementing Kernel Functions, first of all, we have to install the “scikit-learn” library using the command prompt terminal:

* **Gaussian Kernel:** It is used to perform transformation when there is no prior knowledge about data.
* **Gaussian Kernel Radial Basis Function (RBF):** Same as above kernel function, adding radial basis method to improve the transformation.



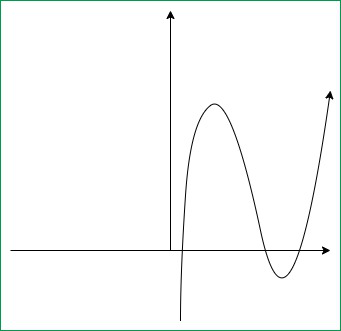
**Gaussian Kernel Graph**

* **Sigmoid Kernel:** this function is equivalent to a two-layer, perceptron model of the neural network, which is used as an activation function for artificial neurons.



**Sigmoid Kernel Graph**

* **Polynomial Kernel:** It represents the similarity of vectors in the training set of data in a feature space over polynomials of the original variables used in the kernel.



**Polynomial Kernel Graph**

**UNIT 3:**

# What is Artificial Neural Network – Structure, Working, Applications

## What is Neural Network in Artificial Intelligence(ANN)?

ANN stands for Artificial Neural Networks. Basically, it’s a computational model. That is based on structures and functions of biological neural networks. Although, the structure of the ANN affected by a flow of information. Hence, neural network changes were based on input and output.

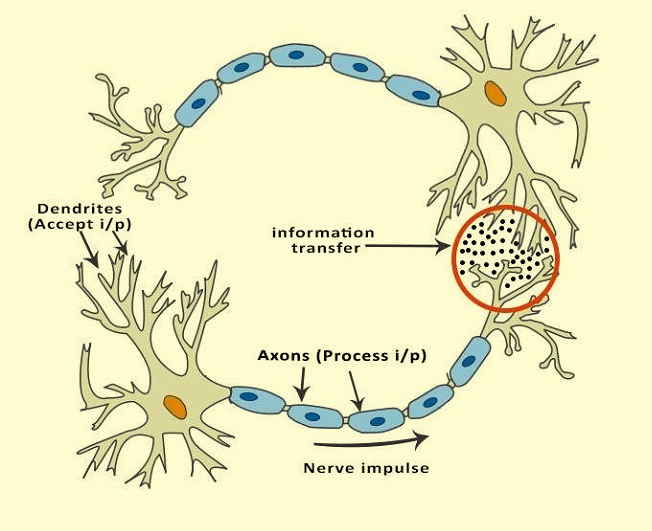
Basically, we can consider ANN as nonlinear statistical data. That means complex relationship defines between input and output. As a result, we found different patterns. Also, we call the ANN as a neural network.

## Structure of Artificial Neural Network

Generally, the working of a human brain by making the right connections is the idea behind ANNs. That was limited to use of silicon and wires as living neurons and dendrites.

Here, neurons, part of human brain. That was composed of 86 billion nerve cells. Also, connected to other thousands of cells by Axons. Although, there are various inputs from sensory organs. That was accepted by dendrites.

As a result, it creates electric impulses. That is used to travel through the Artificial neural network. Thus, to handle the different issues, neuron send a message to another neuron.

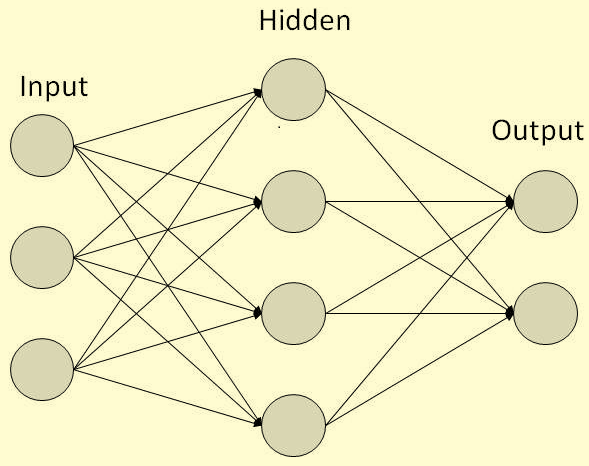
[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2018/01/image-2.png)

Basic Structure of Artificial Neural Network

As a result, we can say that ANNs are composed of multiple nodes. That imitate biological neurons of the human brain. Although, we connect these neurons by links. Also, they interact with each other.

Although, nodes are used to take input data. Further, perform simple operations on the data. As a result, these operations are passed to other neurons. Also, output at each node is called its activation or node value.

As each link is associated with weight. Also, they are capable of learning. That takes place by altering weight values. Hence, the following illustration shows a simple ANN −

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2018/01/image-6.png)

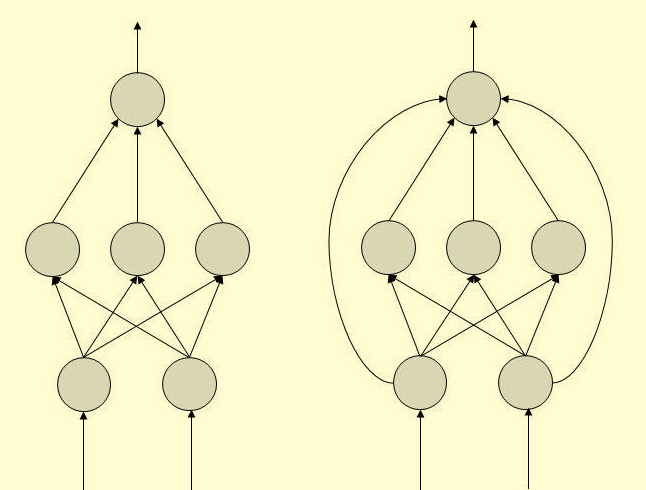
Artificial Neural Network Structure

## Types of Artificial Neural Networks

Generally, there are two types of ANN. Such as FeedForward and Feedback.

### a. FeedForward ANN

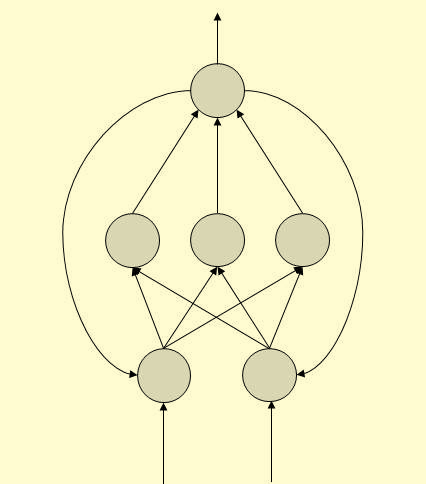
In this network flow of information is unidirectional. A unit used to send information to another unit that does not receive any information. Also, no feedback loops are present in this. Although, used in recognition of a pattern. As they contain fixed inputs and outputs.

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2018/01/image-4.png)

Types of Artificial Neural Networks – FeedForward ANN

### b. FeedBack ANN

In this particular Artificial Neural Network, it allows feedback loops. Also, used in content addressable memories.

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2018/01/image-3.png)

Types of Artificial Neural Networks – FeedBack ANN

## How Does Artificial Neural Networks Works?

In this topology diagrams, you will learn everything in a detailed manner.

In this, each arrow represents a connection between two neurons. Also, they used to indicate the pathway for the flow of information. As it was noticed that each connection has a weight, an integer number. That used to controls the signal between the two neurons.

If the output is good that was generated by the network then we don’t require to adjust the weights. Although, if poor output generated. Then surely system will alter the weight to improve results.

## Machine Learning in ANNs

As there are too many Machine learning strategies are present, let’s see them one by one:

### **a. Supervised Learning**

Generally, in this learning a teacher is present to teach. That teacher must be aware of ANN.

**For example:**

The teacher feeds only example data. That teacher already knows the answers.

### **b. Unsupervised Learning**

If there is present no data set. Then we need this learning technique.

### **c. Reinforcement Learning**

As this Machine learning technique is based on the observation. Although, if it’s negative the networks need to adjust its weights. That is able to make a different required decision the next time.

## Back Propagation Algorithm

Generally, we use to call it as training and learning algorithm. As these networks are ideal for simple Pattern Recognition and Mapping Task.

## Bayesian Networks (BN)

Basically, we use to call it as graphical structures. Generally, we use this network to represent probabilistic representation. This represents among a set of random variables. Also, we used to call this network as Belief networks or Bayes Nets.

In these networks, each node represents a random variable with specific propositions.

In this only constraint arcs present in BN. Thus, doesn’t need to return node by following directed arcs.

Hence, we can say BNs are known as Directed Acyclic Graphs (DAGs). Hence, we use BNs to handle multivalued variables simultaneously.

**Thus, BN variables composed of two dimensions −**

* Range of prepositions
* Probability assigned to each of the prepositions.

## Artificial Neural Networks Applications

Artificial Neural Network used to perform a various task. Also, this task performs that are busy with humans but difficult for a machine.

**a. Aerospace**

Generally, we use ANN a for Autopilot aircrafts. They used for aircraft fault detection.

**b. Military**

In various ways, we use ANN an in the military. Such as Weapon orientation and steering, target tracking.

**c. Electronics**

Basically, we use an Artificial neural network in electronics in many ways. That are code sequence prediction, IC chip layout, and chip failure analysis.

**d. Medical**

As medical has too many machines. That use in various ways. Such as cancer cell analysis, EEG and ECG analysis.

**e. Speech**

We use ANN in speech recognition and speech classification.

**f. Telecommunications**

Generally, it has different applications. Thus, we use an Artificial neural network in many ways. Such as image and data compression, automated information services.

**g. Transportation**

Generally, we use an Artificial neural network in transportation in many ways. That are truck Brake system diagnosis and vehicle scheduling, routing systems.

**h. Software**

It also uses an ANN in pattern Recognition. Such as in facial recognition, optical character recognition, etc.

**i. Time Series Prediction**

We use an Artificial neural network to predict time. Also, we use ANNs to make predictions on stocks and natural calamities.