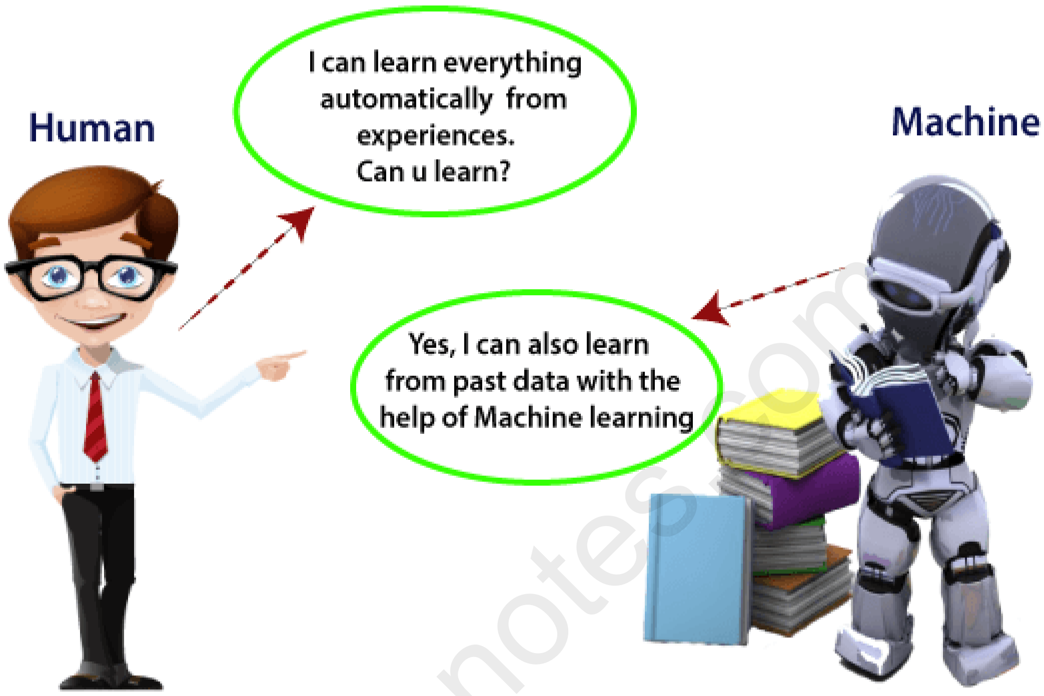
Kerala Notes

What is Machine Learning

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



Machine Learning is said as a subset of **artificial intelligence** that is mainly concerned with the development of algorithms which allow a computer to learn from the data and past experiences on their own. The term machine learning was first introduced by **Arthur Samuel** in **1959**. We can define it in a summarized way as:

> Machine learning enables a machine to automatically learn from data, improve performance from experiences, and predict things without being explicitly programmed.

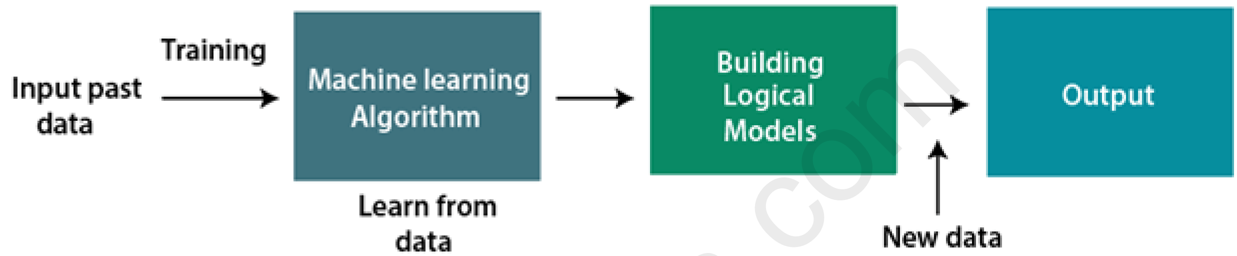
With the help of sample historical data, which is known as **training data**, machine learning algorithms build a **mathematical model** that helps in making predictions or decisions without being explicitly programmed. Machine learning brings computer science and statistics together for creating predictive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

A machine has the ability to learn if it can improve its performance by gaining more data.

How does Machine Learning work

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

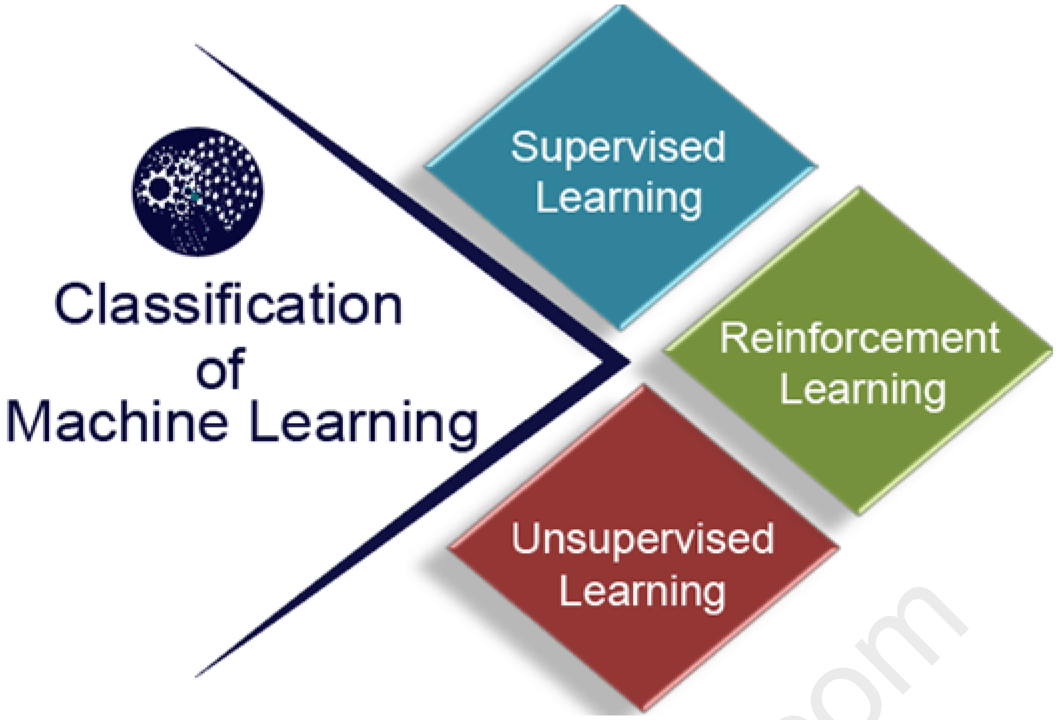
Suppose we have a complex problem, where we need to perform some predictions, so instead of writing a code for it, we just need to feed the data to generic algorithms, and with the help of these algorithms, machine builds the logic as per the data and predict the output. Machine learning has changed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



Classification of Machine Learning

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

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



1) Supervised Learning

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

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

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

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

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

2) Unsupervised Learning

Unsupervised learning is a learning method in which a machine learns without any supervision.

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

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

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

3) Reinforcement Learning

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

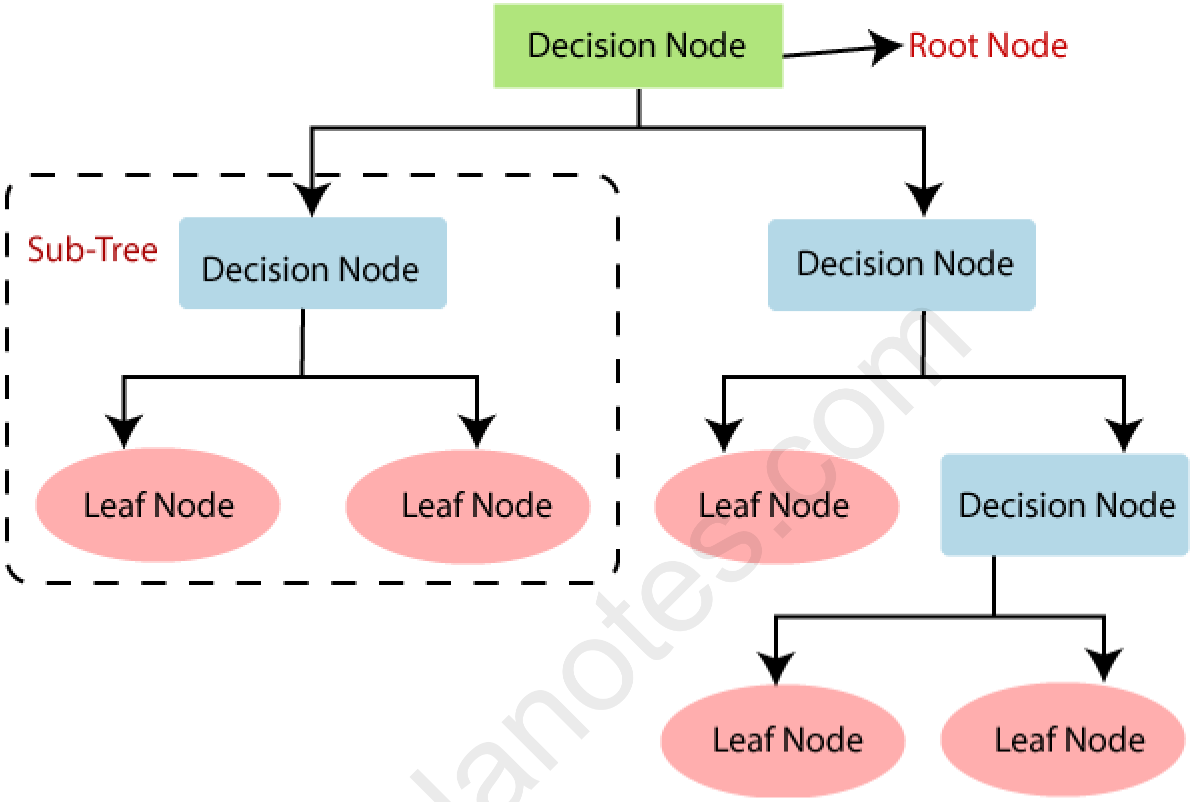
The robotic dog, which automatically learns the movement of his arms, is an example of Reinforcement learning.

Decision Tree Classification Algorithm

- Decision Tree is a **Supervised learning technique** that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where **internal nodes represent the features of a dataset, branches represent the decision rules** and **each leaf node represents the outcome.**
- In a Decision tree, there are two nodes, which are the **Decision Node** and **Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.
- ***It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.***
- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
- In order to build a tree, we use the **CART algorithm,** which stands for **Classification and Regression Tree algorithm.**

- A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.
- Below diagram explains the general structure of a decision tree:

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



Why use Decision Trees?

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

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

Decision Tree Terminologies

- **Root Node:** Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.

- **Leaf Node:** Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
- **Splitting:** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
- **Branch/Sub Tree:** A tree formed by splitting the tree.
- **Pruning:** Pruning is the process of removing the unwanted branches from the tree.
- **Parent/Child node:** The root node of the tree is called the parent node, and other nodes are called the child nodes.

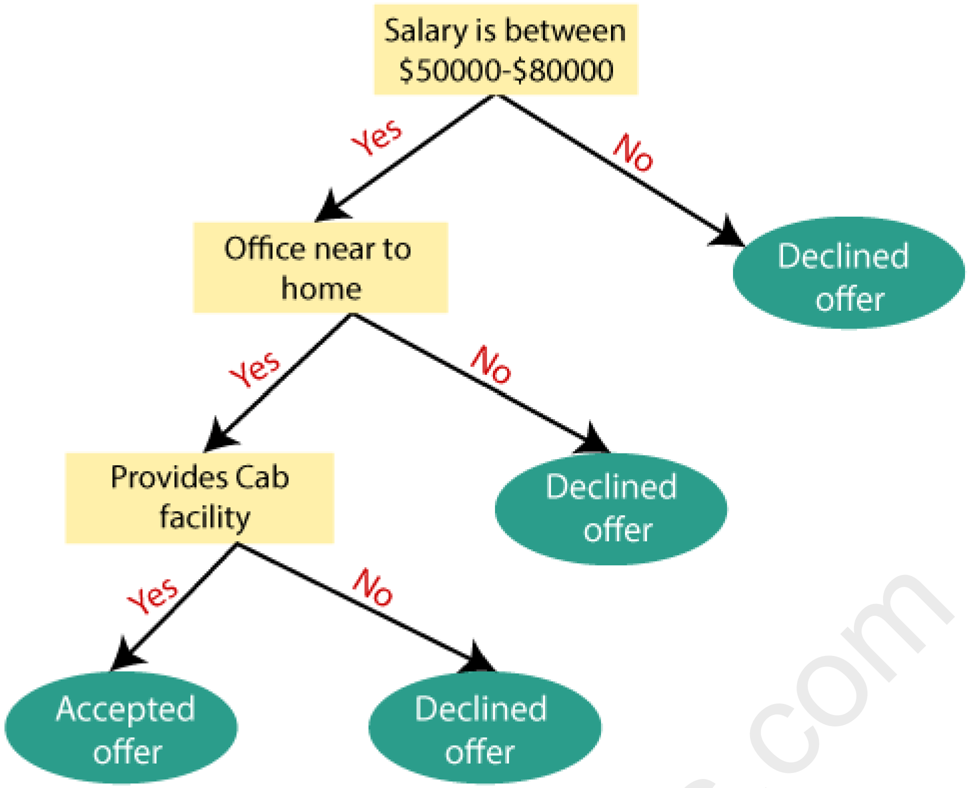
How does the Decision Tree algorithm Work?

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

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

- **Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.
- **Step-2:** Find the best attribute in the dataset using **Attribute Selection Measure (ASM).**
- **Step-3:** Divide the S into subsets that contains possible values for the best attributes.
- **Step-4:** Generate the decision tree node, which contains the best attribute.
- **Step-5:** Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

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



Attribute Selection Measures

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

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

1. Information Gain:

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

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

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

```
Entropy(s)= -P(yes)log2 P(yes)- P(no) log2 P(no)
```

Where,

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

2. Gini Index:

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

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

Pruning: Getting an Optimal Decision tree

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

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

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

Advantages of the Decision Tree

- It is simple to understand as it follows the same process which a human follow while making any decision in real-life.
- It can be very useful for solving decision-related problems.

- It helps to think about all the possible outcomes for a problem.
- There is less requirement of data cleaning compared to other algorithms.

Disadvantages of the Decision Tree

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

Overfitting in Machine Learning

In the real world, the dataset present will never be clean and perfect. It means each dataset contains impurities, noisy data, outliers, missing data, or imbalanced data. Due to these impurities, different problems occur that affect the accuracy and the performance of the model. One of such problems is Overfitting in Machine Learning. *Overfitting is a problem that a model can exhibit.*

A statistical model is said to be overfitted if it can't generalize well with unseen data.

Before understanding overfitting, we need to know some basic terms, which are:

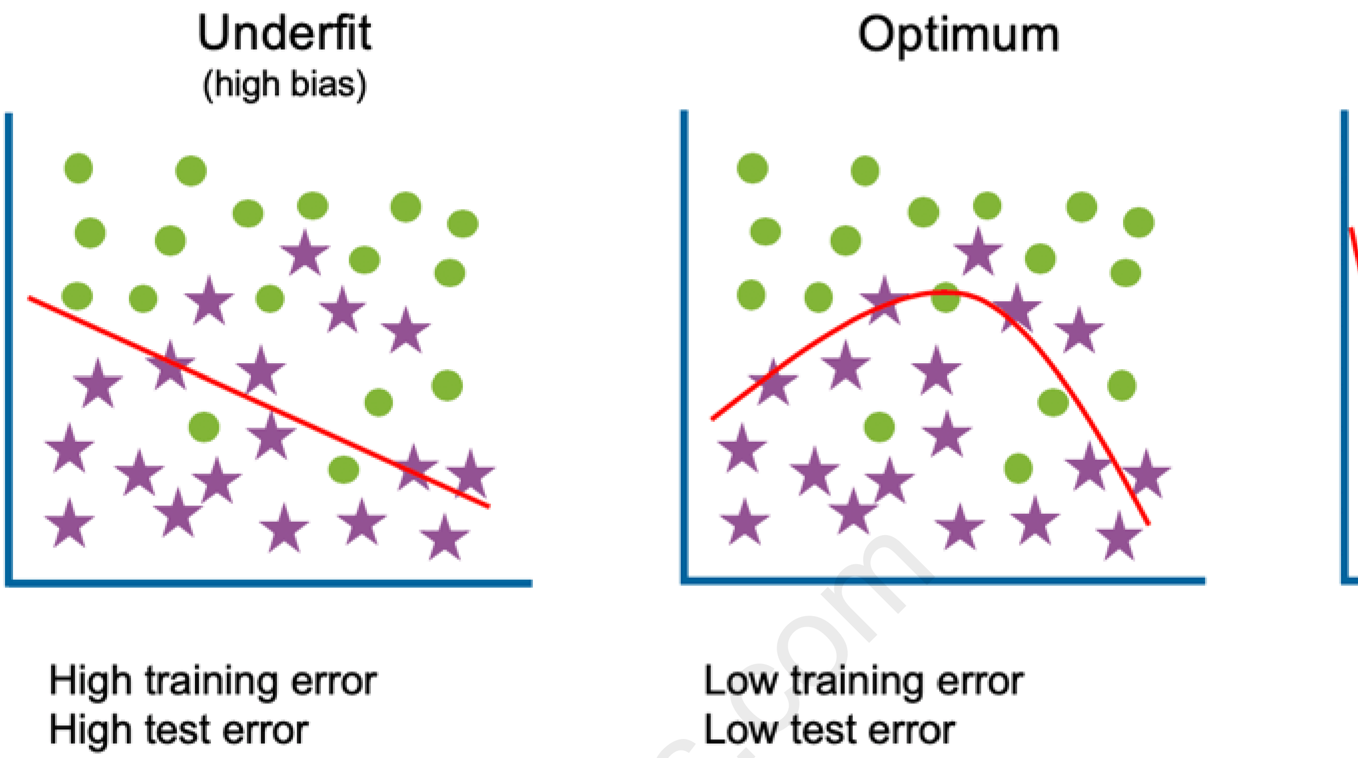
Noise: Noise is meaningless or irrelevant data present in the dataset. It affects the performance of the model if it is not removed.

Bias: Bias is a prediction error that is introduced in the model due to oversimplifying the machine learning algorithms. Or it is the difference between the predicted values and the actual values.

Variance: If the machine learning model performs well with the training dataset, but does not perform well with the test dataset, then variance occurs.

Generalization: It shows how well a model is trained to predict unseen data.

What is Overfitting?



- Overfitting & underfitting are the two main errors/problems in the machine learning model, which cause poor performance in Machine Learning.
- Overfitting occurs when the model fits more data than required, and it tries to capture each and every datapoint fed to it. Hence it starts capturing noise and inaccurate data from the dataset, which degrades the performance of the model.
- An overfitted model doesn't perform accurately with the test/unseen dataset and can't generalize well.
- An overfitted model is said to have low bias and high variance.

Example to Understand Overfitting

We can understand overfitting with a general example. Suppose there are three students, X, Y, and Z, and all three are preparing for an exam. X has studied only three sections of the book and left all other sections. Y has a good memory, hence memorized the whole book. And the third student, Z, has studied and practiced all the questions. So, in the exam, X will only be able to solve the questions if the exam has questions related to section 3. Student Y will only be able to solve questions if they appear exactly the same as given in the book. Student Z will be able to solve all the exam questions in a proper way.

The same happens with machine learning; if the algorithm learns from a small part of the data, it is unable to capture the required data points and hence under fitted.

Suppose the model learns the training dataset, like the Y student. They perform very well on the seen dataset but perform badly on unseen data or unknown instances. In such cases, the model is said to be Overfitting.

And if the model performs well with the training dataset and also with the test/unseen dataset, similar to student Z, it is said to be a good fit.

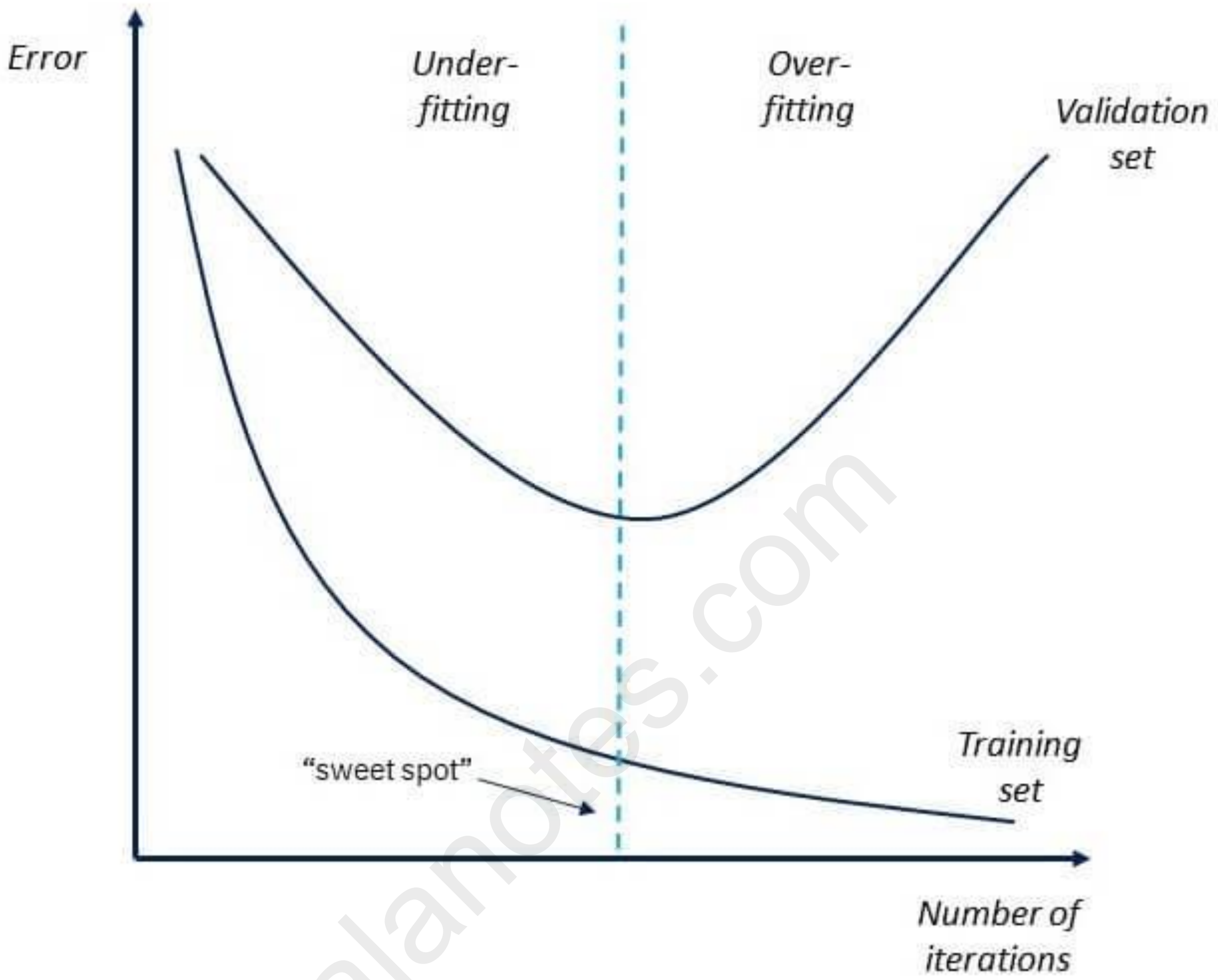
How to detect Overfitting?

Overfitting in the model can only be detected once you test the data. To detect the issue, we can perform **Train/test split.**

In the train-test split of the dataset, we can divide our dataset into random test and training datasets. We train the model with a training dataset which is about 80% of the total dataset. After training the model, we test it with the test dataset, which is 20 % of the total dataset.

Now, if the model performs well with the training dataset but not with the test dataset, then it is likely to have an overfitting issue.

For example, if the model shows 85% accuracy with training data and 50% accuracy with the test dataset, it means the model is not performing well.



Ways to prevent the Overfitting

Although overfitting is an error in Machine learning which reduces the performance of the model, however, we can prevent it in several ways. With the use of the linear model, we can avoid overfitting; however, many real-world problems are non-linear ones. It is important to prevent overfitting from the models. Below are several ways that can be used to prevent overfitting:

1. **Early Stopping**
2. **Train with more data**
3. **Feature Selection**
4. **Cross-Validation**
5. **Data Augmentation**

6. **Regularization**

Hypothesis in Machine Learning

The hypothesis is a common term in Machine Learning and data science projects. As we know, machine learning is one of the most powerful technologies across the world, which helps us to predict results based on past experiences. Moreover, data scientists and ML professionals conduct experiments that aim to solve a problem. These ML professionals and data scientists make an initial assumption for the solution of the problem.

This assumption in Machine learning is known as Hypothesis. In Machine Learning, at various times, Hypothesis and Model are used interchangeably. However, a Hypothesis is an assumption made by scientists, whereas a model is a mathematical representation that is used to test the hypothesis. In this topic, "Hypothesis in Machine Learning," we will discuss a few important concepts related to a hypothesis in machine learning and their importance. So, let's start with a quick introduction to Hypothesis.

What is Hypothesis?

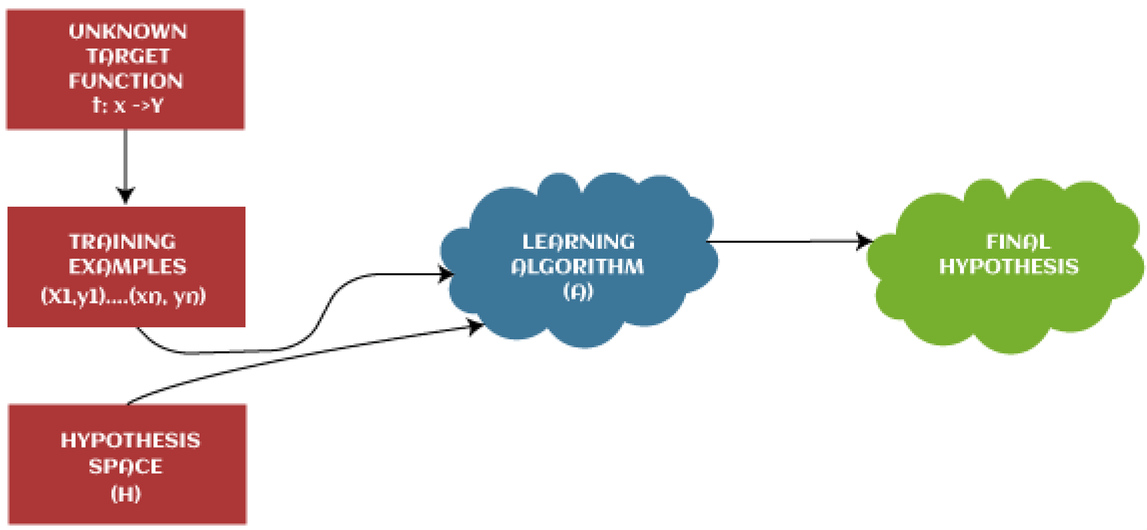
The hypothesis is defined as the supposition or proposed explanation based on insufficient evidence or assumptions. It is just a guess based on some known facts but has not yet been proven. A good hypothesis is testable, which results in either true or false.

Example: Let's understand the hypothesis with a common example. Some scientist claims that ultraviolet (UV) light can damage the eyes then it may also cause blindness.

In this example, a scientist just claims that UV rays are harmful to the eyes, but we assume they may cause blindness. However, it may or may not be possible. Hence, these types of assumptions are called a hypothesis.

Hypothesis in Machine Learning (ML)

The hypothesis is one of the commonly used concepts of statistics in Machine Learning. It is specifically used in Supervised Machine learning, where an ML model learns a function that best maps the input to corresponding outputs with the help of an available dataset.



In supervised learning techniques, the main aim is to determine the possible hypothesis out of hypothesis space that best maps input to the corresponding or correct outputs.

There are some common methods given to find out the possible hypothesis from the Hypothesis space, where hypothesis space is represented by **uppercase-h (H)** and hypothesis by **lowercase-h (h).** Th ese are defined as follows:

Hypothesis space (H):

Hypothesis space is defined as a set of all possible legal hypotheses; hence it is also known as a hypothesis set. It is used by supervised machine learning algorithms to determine the best possible hypothesis to describe the target function or best maps input to output.

It is often constrained by choice of the framing of the problem, the choice of model, and the choice of model configuration.

Hypothesis (h):

It is defined as the approximate function that best describes the target in supervised machine learning algorithms. It is primarily based on data as well as bias and restrictions applied to data.

Hence hypothesis (h) can be concluded as a single hypothesis that maps input to proper output and can be evaluated as well as used to make predictions.

The hypothesis (h) can be formulated in machine learning as follows:

$y = mx + b$

Where,

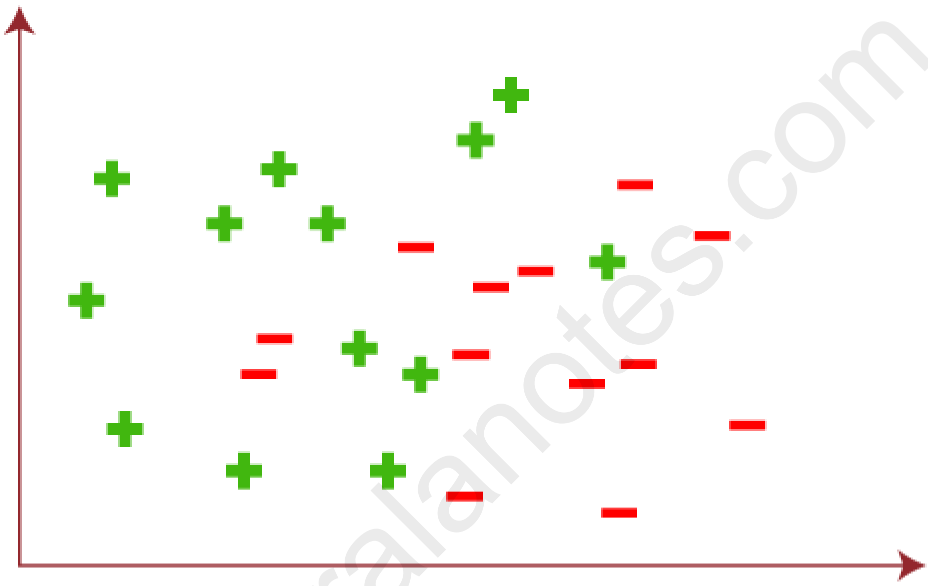
Y: Range

m: Slope of the line which divided test data or changes in y divided by change in x.

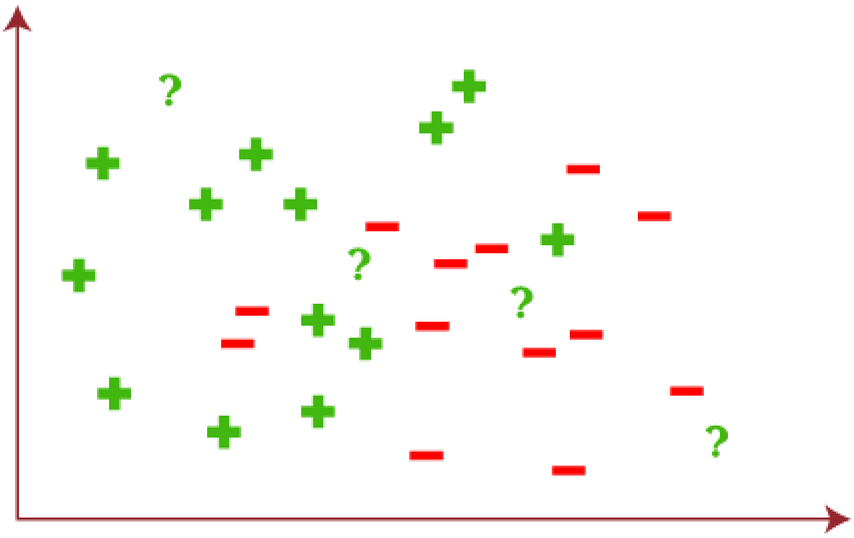
x: domain

c: intercept (constant)

Example: Let's understand the hypothesis (h) and hypothesis space (H) with a two-dimensional coordinate plane showing the distribution of data as follows:



Now, assume we have some test data by which ML algorithms predict the outputs for input as follows:



Difference between Regression and Classification

Regression Algorithm	Classification Algorithm
In Regression, the output variable must be of continuous nature or real value.	In Classification, the output variable must be a discrete v
The task of the regression algorithm is to map the input value (x) with the continuous output variable(y).	The task of the classification algorithm is to map the value(x) with the discrete output variable(y).
Regression Algorithms are used with continuous data.	Classification Algorithms are used with discrete data.
In Regression, we try to find the best fit line, which can predict the output more accurately.	In Classification, we try to find the decision boundary, v can divide the dataset into different classes.
Regression algorithms can be used to solve the regression problems such as Weather Prediction, House price prediction, etc.	Classification Algorithms can be used to solve classific problems such as Identification of spam emails, Sp Recognition, Identification of cancer cells, etc.
The regression Algorithm can be further divided into Linear and Non-linear Regression.	The Classification algorithms can be divided into E Classifier and Multi-class Classifier.