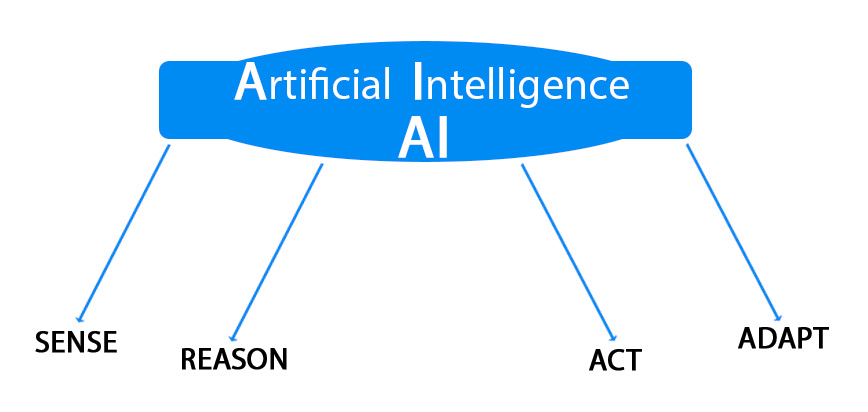
**Artificial Intelligence**:  
→ Artificial intelligence(AI) is a way of artificially making a computer intelligent.



**Machine Learning**:

→Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being **explicitly programmed**.

→Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

→The process of learning begins with

(i).observations or data,

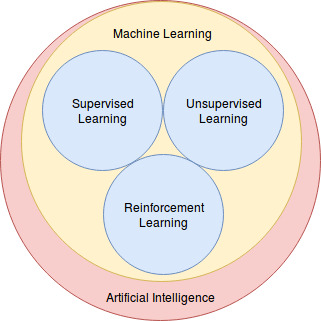
(ii).examples

(iii).direct experience

(iv). Instruction

(v).to look for patterns in data and make better decisions in the future based on the examples that we provide.

→The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.



**Branches of Machine Learning**:

1.Computational Learning Theory:

→studying and analyzing the algorithms of machine learning

Adversarial Machine Learning:

→interaction of machine learning and computer security.

2.Quantum Machine Learning :

→machine learning deals with quantum physics.

Predictive Analysis :

→Predictive Analysis uses statistical techniques from data modeling, machine learning and data mining to analyze current and historical data to predict the future.

Customer relationship management(CRM) is the common application of predictive analysis.

3.Robot Learning :

→This area deals with the interaction of machine learning and robotics.

Grammar Induction :

→It is a process in machine learning to learn formal grammar from a given set of observations to identify characteristics of the observed model

4.Meta-Learning :

→In this process learning algorithms are applied on meta-data and mainly deals with automatic learning algorithms.

Best Machine Learning Tools:

Here is a list of artificial intelligence and machine learning tools for developers:

ai-one

Protege –

IBM Watson

DiffBlue

TensorFlow

Amazon Web Services

OpenNN

Apache Spark –

Caffe –

Veles

Data

**Data**:

Data can be from Netflix movies watched to recommend,credit history to give credit card, price of house in particular city over the 5 years etc..

1. .Labbelled
2. .Unlabelled

**Labelled data**:

→When data is labelled, it means we have historic data.

→Data that comes with a label.

**Example of labeled data**:

* Image: [cat.jpg, dog.jpg, cat.jpg]
* Labels: ["cat", "dog", "cat"]

If the array represents **image files** like image1.jpg, image2.jpg, and image3.jpg, those are just file paths to the images or can be some thing like binary data instances of images

**Unlabelled data**:

→Data is unlabeled. We don’t have any historic data

→Data that comes without a label.

A picture containing text, different, screen, several

Description automatically generated

**Example of unlabeled data**:

* Image: [image1.jpg, image2.jpg, image3.jpg]
* No labels available.

### **Key Difference**:

**Labeled Data**: You already know the correct answer (label) for each data point, and the goal is to train a model to predict that label.

**Unlabeled Data**: You don't know the correct answer (label), and the goal is to learn patterns or structures from the data itself.

### **Common Use Cases**:

* **Labeled Data**: Supervised learning (e.g., classification, regression).
* **Unlabeled Data**: Unsupervised learning (e.g., clustering, anomaly detection).

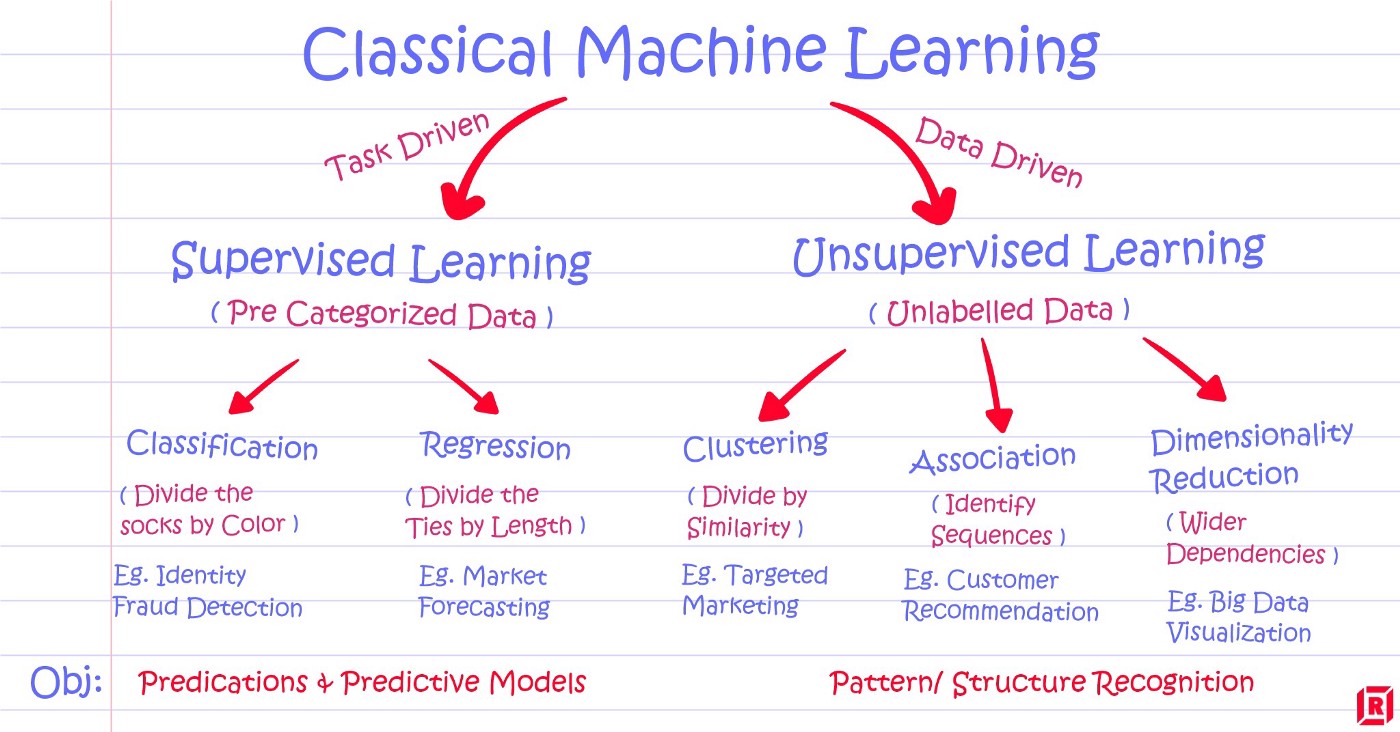
**Some machine learning methods**:

(i). Supervised machine learning algorithms

(ii). Unsupervised machine learning algorithms

(iii). Semi-supervised machine learning algorithms

(iv). Reinforcement machine learning algorithms



**Task-Driven**:

The task is explicit — for example, classifying an email as spam or not spam, predicting house prices, or recognizing handwritten digits.

The goal of supervised learning is to optimize the model for this specific task, which is defined by the available labels in the training data.

* + **Classification Task**: You have a dataset of images of cats and dogs (inputs) and labels indicating whether the image is a cat or a dog (outputs). The task is to classify new images into "cat" or "dog."
  + **Regression Task**: You have a dataset of house features (inputs) and prices (outputs). The task is to predict the price of a house based on its features.

In summary, supervised learning is **task-driven** because you know the goal (task) from the start — you are trying to solve a specific problem or answer a specific question, and the labels guide the learning process toward that goal.

**Analogy**:

You say to the student:  
"Here are several pictures of animals, and I will tell you what each animal is. I want you to learn to recognize them in the future."

* You show pictures of different animals (like dogs, cats, lions, etc.) and tell the student, "This is a dog," "This is a cat," "This is a lion."
* The task is clear: the student needs to learn to identify the animals based on the images.
* The student is **supervised** by your labels (dog, cat, lion). Every time they see a new animal, they can use their learned knowledge to correctly classify it.

**Data -driven**:  
Unsupervised learning is **data-driven** because it focuses on understanding the structure, patterns, or relationships in the data without explicit labels.

Here, the model is not given a specific task in the same way as supervised learning, but rather it seeks to find hidden patterns or structures within the dataset itself.

In unsupervised learning, the model is given **input data** without any labels or target outputs. The model tries to extract meaningful insights, such as grouping similar data points or discovering hidden features, based purely on the data’s inherent structure.

The learning process is **driven by the data itself** — the model explores the data to discover clusters, patterns, or correlations without being explicitly told what to look for.

**Clustering**: In a dataset of customer data (without labels), unsupervised learning might group customers into different clusters based on purchasing behavior, geographical location, or other factors.

**Dimensionality Reduction**: The model might reduce the number of features in the dataset while preserving the important structure, making it easier to visualize or use for other tasks.

#### Example:

You give the student a pile of pictures of various animals without telling them what any of them are. You simply say:  
"Here are some pictures. Try to find patterns or groupings in them."

* The student might start grouping animals based on similar features: they might notice that some animals have fur, others have scales, some are small, others are large.
* They may group animals into categories like "furry animals" or "scaly animals," but they don’t know if those groupings correspond to any known species like dogs or cats.
* The student isn't given any labels (like "dog" or "cat"), so they have to discover the patterns and categories by themselves. The learning process is **data-driven**, as the student is making sense of the data without explicit supervision.

**Regression**:

In general, **regression** refers to a statistical method or a type of predictive modeling technique where the goal is to **predict a continuous output** (a number) based on one or more input features (variables). In other words, regression tries to find the relationship between dependent (output) and independent (input) variables.

For example, if you are trying to predict a **house price** based on features like the **size of the house**, **number of bedrooms**, and **location**, that's a **regression problem** because the price is a continuous value, and you are trying to find a function that relates the input features to that output.

#### Types of Regression:

* **Linear Regression**: Predicts a continuous value by fitting a straight line (or hyperplane in higher dimensions) to the data.
* **Polynomial Regression**: Uses higher-degree polynomials to fit the data if the relationship between the variables is non-linear.
* **Logistic Regression**: Despite the name "regression," it's used for classification tasks, specifically when the output is categorical (e.g., yes/no, 0/1).

In the context of predictive modeling, **regression** is used because it often deals with predicting values (like a mean or average value) based on some inputs.

**Supervised machine learning algorithms**:

→ Supervised Machine Learning deals with labelled data.**supervised learning** gets its name because the algorithm is **supervised** by labeled data.The "supervision" refers to the fact that the algorithm is **guided** or "taught" by the labeled data.

→Inputs and outputs are labelled.we have a **dataset with both inputs and outputs** (labels or target values).

→We have specific input data and specific output.

→Supervision will be there.

**Unsupervised machine learning algorithms**:

→ Supervision won’t be there.**Unsupervised learning** gets its name because, unlike supervised learning, there are **no labels** (no supervision) for the data.

→the data has no labels.

→It is a dataset with only features, and no target to predict.

**Example 1**:

**We contain set of images of dog and cat. We need to find the animal name.**

**Explanation**:

Supervised:

→Let the data set is formed by images of dogs and cats, and the labels in the image are ‘dog’ and ‘cat’.

→ The machine learning model would simply use previous data to predict the label of new data points.

This means, if we bring in a new image without a label, the model would guess if the image is of a dog or a cat, thus predicting the label of the data point.

Graphical user interface

Description automatically generated with low confidence

We have two types of datasets, one in which the labels are numbers (the weight of the animal), and one in which the labels are states, or classes (the type of animal, namely cat or dog). This gives rise to two types of supervised learning models.

A picture containing text, different, screen, several

Description automatically generated

→ Remember-Formulate-Predict. This is precisely how supervised learning works.

The model first remembers the dataset of dogs and cats, then formulates a model, or a rule for what is a dog and what is a cat, and when a new image comes in, the model makes a prediction about what the label of the image is, namely, is it a dog or a cat.

Graphical user interface

Description automatically generated with medium confidence

Unsupervised:

→Unfortunately, in unsupervised the main thing we are aiming to predict is not there.

However, we can still extract a lot of information from an unlabeled dataset.

→If our dataset has no labels, then we simply have a bunch of pictures of dogs and cats, and we do not know what type of pet each one represents.

→Our model can still tell us if two pictures of dogs are similar to each other, and different to a picture of a cat.

→Maybe it can group them in some way by similarity, even without knowing what each group represents.

Graphical user interface, text

Description automatically generated

Example 2:

**Suppose you had a basket and it is filled with some different kinds of fruits, your task is to arrange them as groups.**

A bowl of fruits

Description automatically generated with low confidence

**Supervised Learning**:

Size:

|  |  |
| --- | --- |
| Banana | Cylander big |
| Apple | big |
| Grapes | Very small |
|  |  |

Color:

|  |  |
| --- | --- |
| Banana | Yellow |
| Apple | Red |
| Grapes | Green |
|  |  |

→You already learn from your previous work about the physical characters of fruits.

So arranging the same type of fruits at one place is easy now.

→Your previous work is called as training data in data mining.

→so you already learn the things from your train data, this is because of response variable.

|  |
| --- |
| Response variable mean just a decision variable. |

You can observe response variable below (FRUIT NAME) .

Table

Description automatically generated

→Suppose you have taken an new fruit from the basket then you will see the size , color and shape of that particular fruit.

If size is Big , color is Red , shape is rounded shape with a depression at the top, you will conform the fruit name as apple and you will put in apple group.

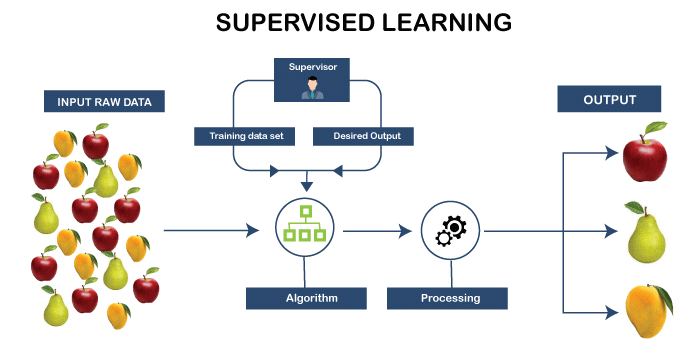
Likewise for other fruits also.

→Job of groping fruits was done and happy ending.

You can observe in the table that a column was labeled as “FRUIT NAME” this is called as response variable.

If you learn the thing before from training data and then applying that knowledge to the test data(for new fruit), This type of learning is called as Supervised Learning.

Classification come under Supervised learning.



**Unsupervised learning**:

→Suppose you had a basket and it is filled with some different types fruits, your task is to arrange them as groups.

→This time you don’t know any thing about that fruits, honestly saying this is the first time you have seen them.

so how will you arrange them. What will you do first???

→You will take a fruit and you will arrange them by considering physical character of that particular fruit. suppose you have considered color.

Then you will arrange them on considering base condition as color.

Then the groups will be some thing like this.

|  |  |
| --- | --- |
|  |  |
| RED COLOR GROUP | apples & cherry fruits. |
| GREEN COLOR GROUP | bananas & grapes. |
|  |  |

Size:

|  |  |
| --- | --- |
| RED COLOR AND BIG SIZE | apple |
| RED COLOR AND SMALL SIZE | cherry fruits |
| GREEN COLOR AND BIG SIZE | bananas |
| GREEN COLOR AND SMALL SIZE | grapes |

job done happy ending.

Here you didn’t know learn any thing before ,means no train data and no response variable.

→This type of learning is known unsupervised learning.

**Example** :

**Features: Height and weight**

**Label: Gender**

**Task: Given a persons’ height and weight predict their gender**

Chart, scatter chart

Description automatically generated

**Example 1**:

Let's, take the case of a baby and her family dog.



She knows and identifies this dog. Few weeks later a family friend brings along a dog and tries to play with the baby.

[](https://www.guru99.com/images/1/030819_1030_Unsupervise2.png)

Baby has not seen this dog earlier. But it recognizes many features (2 ears, eyes, walking on 4 legs) are like her pet dog.

She identifies the new animal as a dog.

### What the Baby Does:

* The baby **already knows** and can **recognize** the family dog, which she has learned to identify based on previous exposure. This is like **supervised learning**.
* When the family friend brings a new dog, the baby **recognizes** the new dog by identifying features that are similar to the family dog (e.g., ears, eyes, four legs, etc.). This is an example of the baby **generalizing** from what she has learned about her own dog to the new dog.

### Supervised Learning (Related to the Baby's Initial Learning of Her Dog):

In the beginning, when the baby learns to identify her family dog, this could be seen as a **supervised learning** process:

* **Supervision**: The baby is taught (implicitly) that this particular animal is a dog, likely through feedback from her family (e.g., “This is our dog, Spot”).
* **Labeling**: The family provides the label (the dog’s name or calling it "dog"), and the baby learns to associate the features of this specific animal with the label “dog.”

### Unsupervised Learning (Related to the Baby Recognizing the New Dog):

When the baby encounters the new dog that she has never seen before, she **doesn’t have a label** or any prior supervision regarding this specific dog. However, she can **identify** the new dog by recognizing certain **common features** (ears, eyes, walking on four legs).No labels but she recognized with features.

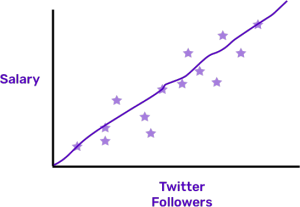
This part is like **unsupervised learning** because the baby is **not explicitly told** that the new dog is a dog, but she identifies the animal based on the features that are similar to the first dog. She **discovers a pattern** (the shared characteristics between dogs) on her own.

The baby is making use of **previously learned information** (recognizing the dog based on its features) and **generalizing** that knowledge to the new situation. While the baby doesn’t have a label for the new dog yet, she is **using patterns** to identify it.

Example 2:

Traditional datasets in ML have labels (think: the answer key), and follow the logic of “X leads to Y.”

For example: we might want to figure out if people with more Twitter followers typically make higher salaries. We think that our input (Twitter followers) might lead to our output (salary), and we try to approximate what that relationship is.

[](https://blog.algorithmia.com/wp-content/uploads/2018/04/unsupervised.png)

The stars are data points, and machine learning works on creating a line that explains how the input and outcomes are related. But in unsupervised learning, there are no outcomes! We’re just looking to analyze in the input, which is our Twitter followers.

There is no salary, or Y, involved at all. Just like there not being an answer key for the test.

[Unsupervised Learning Example](https://blog.algorithmia.com/wp-content/uploads/2018/04/unsupervised2.png)

Maybe we don’t have access to salary data, or we’re just interested in different questions. It doesn’t matter! The important thing is that there is no output to match to, and no line to draw that represents a relationship.

**Why Unsupervised Learning**:

Here, are prime reasons for using Unsupervised Learning:

Unsupervised machine learning finds all kind of unknown patterns in data.

Unsupervised methods help you to find features which can be useful for categorization.

It is taken place in real time, so all the input data to be analyzed and labeled in the presence of learners.

It is easier to get unlabeled data from a computer than labeled data, which needs manual intervention.

| **Parameters** | **Supervised machine learning technique** | **Unsupervised machine learning technique** |
| --- | --- | --- |
| Process | In a supervised learning model, input and output variables will be given. | In unsupervised learning model, only input data will be given |
| Input Data | Algorithms are trained using labeled data. | Algorithms are used against data which is not labeled |
| Algorithms Used | Support vector machine, Neural network, Linear and logistics regression, random forest, and Classification trees. | Unsupervised algorithms can be divided into different categories: like Cluster algorithms, K-means, Hierarchical clustering, etc. |
| Computational Complexity | Supervised learning is a simpler method. | Unsupervised learning is computationally complex |
| Use of Data | Supervised learning model uses training data to learn a link between the input and the outputs. | Unsupervised learning does not use output data. |
| Accuracy of Results | Highly accurate and trustworthy method. | Less accurate and trustworthy method. |
| Real Time Learning | Learning method takes place offline. | Learning method takes place in real time. |
| Number of Classes | Number of classes is known. | Number of classes is not known. |
| Main Drawback | Classifying big data can be a real challenge in Supervised Learning. | You cannot get precise information regarding data sorting, and the output as data used in unsupervised learning is labeled and not known. |

Supervised Learning

**Supervised Learning**:

→The majority of practical machine learning uses supervised learning.

→It can apply what has been learned in the past to new data using labeled examples to predict future events.

→It uses labelled data(historical data) to predict a label given some features.

**Predictive modelling**:

→Predictive modelling can be described as the mathematical problem of approximating a mapping function (f) from input variables (X) to output variables (y).

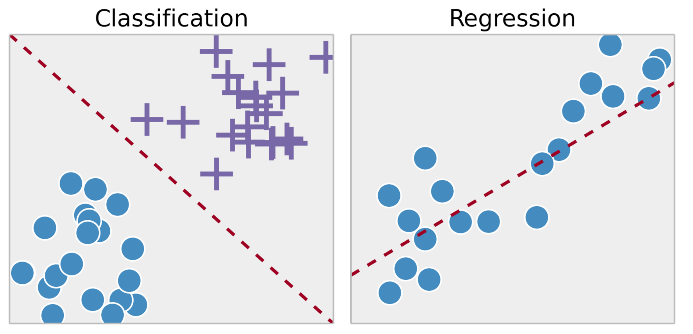
→This is called the problem of function approximation.

→Supervised learning is where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output.

Y = f(X)

The goal is to approximate the mapping function so well that when you have new input data (x) that you can predict the output variables (Y) for that data.

**Supervised Learning**:



→Types of Supervised machine learning:

(i).Classification

(ii).Regression

Diagram

Description automatically generated

|  |
| --- |
| The main difference between them is that the output variable in regression is numerical (or continuous) while that for classification is categorical state (or discrete). |

**Classification in machine learning:**

→Classification: Classifying labelled data

→The main goal of classification is to predict the target class (Yes/ No).

→Classification algorithms attempt to estimate the mapping function (f) from the input variables (x) to discrete or categorical output variables (y).

In this case, y is a category that the mapping function predicts.

→If provided with a single or several input variables, a classification model will attempt to predict the value of a single or several conclusions.

→A classification problem requires that examples be classified into one of two or more classes.

→A classification can have real-valued or discrete input variables.

→Classification algorithims:

1.Logistic Regression

2.K-Nearest Neighborhood

3.Support Vector Machines

4.Kernal SVM

5.Naive Bayes

6.Decission Tree classification

7.Random Forest classification

**Regression in machine learning:**

→In machine learning, regression algorithms attempt to estimate the mapping function (f) from the input variables (x) to numerical or continuous output variables (y).

→In this case, y is a real value, which can be an integer or a floating point value. Therefore, regression prediction problems are usually quantities or sizes.

→A regression problem requires the prediction of a quantity.

→A regression can have real valued or discrete input variables.

→A problem with multiple input variables is often called a multivariate regression problem.

→A regression problem where input variables are ordered by time is called a time series forecasting problem.

→Regression Algorithms:

1.Simple Linear

2.Multiple Linear

3.Polynomial

4.Support Vectors for Regression

5.Decision tree classification

6.Random forest 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 value. |
| 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 input 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, which 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 classification problems such as Identification of spam emails, Speech 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 Binary Classifier and Multi-class Classifier. |

**Example : Housing prices model:**

**In this model, each data point is a house. You will** provided with a dataset about houses.

**The label of each house is its price.**

**Regression:**

**Our goal is, when a new house (data point) comes in the market, we would like to predict its label, namely, its price. This is called regression.**

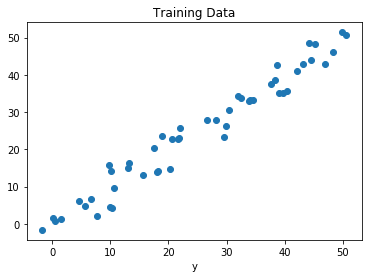
When provided with a dataset about houses, and you are asked to predict their prices, that is a regression task because price will be a continuous output.

**Classification:**

When, a classification algorithm can try to predict whether the prices for the houses “sell more or less than the recommended retail price.”

Here, the houses will be classified whether their prices fall into two discrete categories: above or below the said price.

Rain forecast:

we are finding the possibility of rain in some particular regions with the help of some parameters recorded earlier. Then there is a probability associated with the rain.  
  
 Fig : Regression of Day vs Rainfall (in mm)

**Example** :

**Email Spam**:

(classification), email spam detection model: In this model, each data point is an email. The label of each email is either spam or ham.

Our goal is, when a new email (data point) comes into our inbox, we would like to predict its label, namely, if it is spam or ham.

**Example** :

Classification and Regression difference with an example.Movies



**Classification Example**:

Suppose from your past data ( train data ) you come to know that your best friend likes the above movies.

Now one new movie ( test data ) released. Hopefully, you want to know your best friend like it or not. If you strongly confirmed about the chances of your friend like the move.  You can take your friend to a movie this weekend.

If you clearly observe the problem it is just whether your friend like or not. Finding a solution to this type of problem is called as classification.

This is because we are classifying the things to their belongings (yes or no, like or dislike ).

Keep in mind here we are  forecasting target class( classification ) and the other thing this classification belongs to Supervised learning. This is because you are learning this from your train data.

In this case,  the problem is a binary classification in which we have to predict whether output belongs to class 1 or class 2 (class 1 : yes, class 2: no ).

Regression Example:

Suppose from your past data ( train data ) you come to know that your best friend likes the above movies.

You also know how many times each particular movie seen by your friend.

Now one new movie ( test data ) released. Now your are going to find how many times this newly released movie will your friend watch. It could be 5 times, 6 times,10 times etc…

If you clearly observe the problem is about finding the count, sometimes we can say this as predicting the value. Keep in mind, here we are forecasting a value ( Prediction ) and the other thing this prediction also belongs to Supervised learning.

This is because you are learning this from you train data.

Summary

If forecasting target class ( Classification )

If forecasting a value ( Regression )

Q. **Which of the following is/are classification problem(s)?**

1.Predicting the gender of a person by his/her handwriting style

2.Predicting house price based on area

3.Predicting whether monsoon will be normal next year

4.Predict the number of copies a music album will be sold next month

Solution :

Predicting the gender of a person

Predicting whether monsoon will be normal next year. The other two are regression.

**Binary and Multi-class classification**:

**Binary classification**:

→A problem with two classes is often called a two-class or binary classification problem.

Eg:

(i).Considering the student profile to predict whether the student will pass or fail. (ii).Considering the customer, transaction details to predict whether he will buy the new product or not.

**Multi Labeled** :

→A problem where an example is assigned multiple classes is called a multi-label classification problem.

**Multi-class classification**

→A problem with more than two classes is often called a multi-class classification problem.

Eg:

Considering all subject details of a student to  predict which subject the student will score more.

Identifying the object in an image. These kind problems are known as multi-classification problems.

**Example** :Match output predition

Let’s take an example, suppose we want to predict the possibility of the wining of match by Team A on the basis of some parameters recorded earlier. Then there would be two labels Yes and No.

*Chart

Description automatically generated*

*Diagram

Description automatically generated*

Un Supervised ML

**Unsupervised ML**:

→They are used when the information used to train is neither classified nor labeled.

→Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data.

→The system doesn’t figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

**Types of Unsupervised Learning**:

Unsupervised learning problems further grouped into

(i). clustering and

(ii).association problems.

(iii).Dimensionality reduction

**Association rule :**

→It is the process of measuring the degree of association between any 2 items.

An association rule is typically written as:

* **A → B**: This means if item A is bought, item B is likely to be bought as well.

**For example**,

If we go to a grocery shop, there is a high probability that we will buy a jam if we already bought bread there. This is because bread and jam are 2 items that are closely associated.

But, there is only a low probability that we will buy a biscuit if we already bought a book. This is because biscuits and books are not closely associated.

**Clustering**:

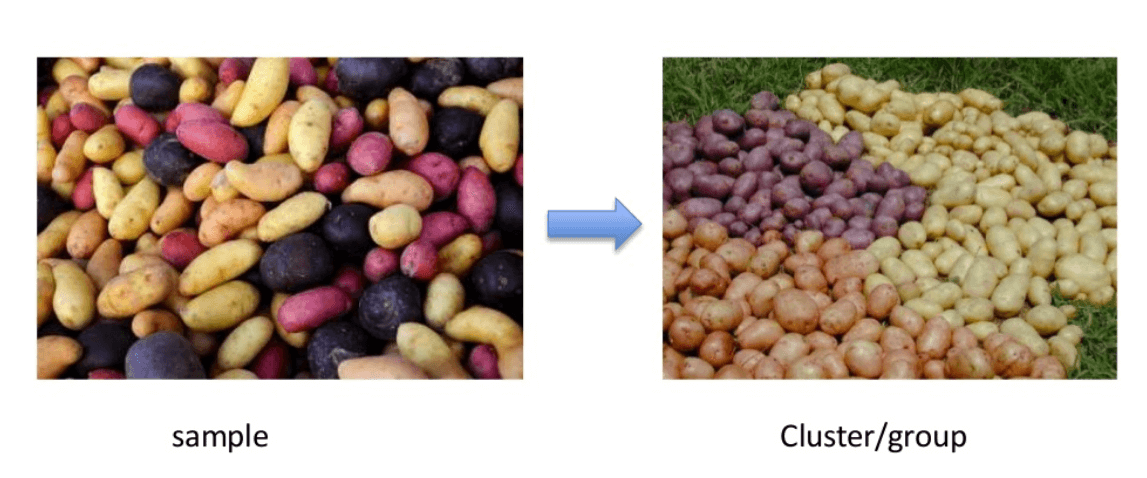
|  |
| --- |
| This is the task of grouping our data into clusters based on similarity. |

**Eg**:

→**For example**, when you go out for grocery shopping, you easily distinguish between apples and oranges in a given set containing both of them.

You distinguish these two objects based on their color, texture and other sensory information that is processed by your brain.

→Clustering is an emulation of this process so that machines are able to distinguish between different objects.



Diagram

Description automatically generated with medium confidence

→It mainly deals with finding a structure or pattern in a collection of uncategorized data. Clustering algorithms will process your data and find natural clusters(groups) if they exist in the data.

→You can also modify how many clusters your algorithms should identify.

→It allows you to adjust the granularity of these groups.

## **Types of Clustering Algorithms:**

→In total, there are five distinct types of clustering algorithms. They are as follows –

* Partitioning Based Clustering
* Hierarchical Clustering
* Model-Based Clustering
* Density-Based Clustering
* Fuzzy Clustering

→There are different types of clustering you can utilize:

Exclusive (partitioning):

→In this clustering method, Data are grouped in such a way that one data can belong to one cluster only.

Example: K-means

Agglomerative:

→In this clustering technique, every data is a cluster.

→The iterative unions between the two nearest clusters reduce the number of clusters.

Example: Hierarchical clustering

Overlapping:

→In this technique, fuzzy sets is used to cluster data.

→Each point may belong to two or more clusters with separate degrees of membership.

Probabilistic:

→This technique uses probability distribution to create the clusters

Example: Following keywords

"man's shoe."

"women's shoe."

"women's glove."

"man's glove."

can be clustered into two categories "shoe" and "glove" or "man" and "women."

**Clustering Types**:

1.Hierarchical clustering

2.K-means clustering

3.K-NN (k nearest neighbors)

4.Principal Component Analysis

5.Singular Value Decomposition

6.Independent Component Analysis

7.[Probabilistic Clustering](https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/) –

**Hierarchical Clustering**:

→Hierarchical clustering is an algorithm which builds a hierarchy of clusters.

→It begins with all the data which is assigned to a cluster of their own.

→Here, two close cluster are going to be in the same cluster.

→This algorithm ends when there is only one cluster left.

**K-means Clustering**:

→K means it is an iterative clustering algorithm which helps you to find the highest value for every iteration.

→Initially, the desired number of clusters are selected. In this clustering method, you need to cluster the data points into k groups.

→A larger k means smaller groups with more granularity in the same way.

→A lower k means larger groups with less granularity.

→The output of the algorithm is a group of "labels."

→It assigns data point to one of the k groups.

In k-means clustering, each group is defined by creating a centroid for each group. The centroids are like the heart of the cluster, which captures the points closest to them and adds them to the cluster.

K-mean clustering further defines two subgroups:

(i).Agglomerative clustering

(ii).Dendrogram

**Agglomerative clustering**:

→This type of K-means clustering starts with a fixed number of clusters.

→It allocates all data into the exact number of clusters.

→This clustering method does not require the number of clusters K as an input. →Agglomeration process starts by forming each data as a single cluster.

→This method uses some distance measure, reduces the number of clusters (one in each iteration) by merging process.

Lastly, we have one big cluster that contains all the objects.

**Dendrogram**:

→In the Dendrogram clustering method, each level will represent a possible cluster.

→The height of dendrogram shows the level of similarity between two join clusters.

→The closer to the bottom of the process they are more similar cluster which is finding of the group from dendrogram which is not natural and mostly subjective.

**K- Nearest neighbors** :

→K- nearest neighbour is the simplest of all machine learning classifiers.

→It differs from other machine learning techniques, in that it doesn't produce a model.

→It is a simple algorithm which stores all available cases and classifies new instances based on a similarity measure.

It works very well when there is a distance between examples.

→The learning speed is slow when the training set is large, and the distance calculation is nontrivial.

**Principal Components Analysis**:

→In case you want a higher-dimensional space. You need to select a basis for that space and only the 200 most important scores of that basis.

→This base is known as a principal component. The subset you select constitute is a new space which is small in size compared to original space.

→It maintains as much of the complexity of data as possible.

**Association**:

→Association rules allow you to establish associations amongst data objects inside large databases.

→This unsupervised technique is about discovering interesting relationships between variables in large databases.

For example, people that buy a new home most likely to buy new furniture.

Other Examples:

A subgroup of cancer patients grouped by their gene expression measurements

Groups of shopper based on their browsing and purchasing histories

Movie group by the rating given by movies viewers

**Applications of unsupervised machine learning**:

→Clustering automatically split the dataset into groups base on their similarities

→Anomaly detection can discover unusual data points in your dataset. It is useful for finding fraudulent transactions

→Association mining identifies sets of items which often occur together in your dataset

→Latent variable models are widely used for data preprocessing. Like reducing the number of features in a dataset or decomposing the dataset into multiple components

**Disadvantages of Unsupervised Learning**:

→You cannot get precise information regarding data sorting, and the output as data used in unsupervised learning is labeled and not known

→Less accuracy of the results is because the input data is not known and not labeled by people in advance. This means that the machine requires to do this itself.

→The spectral classes do not always correspond to informational classes.

→The user needs to spend time interpreting and label the classes which follow that classification.

→Spectral properties of classes can also change over time so you can't have the same class information while moving from one image to another.

Semi supervised ML

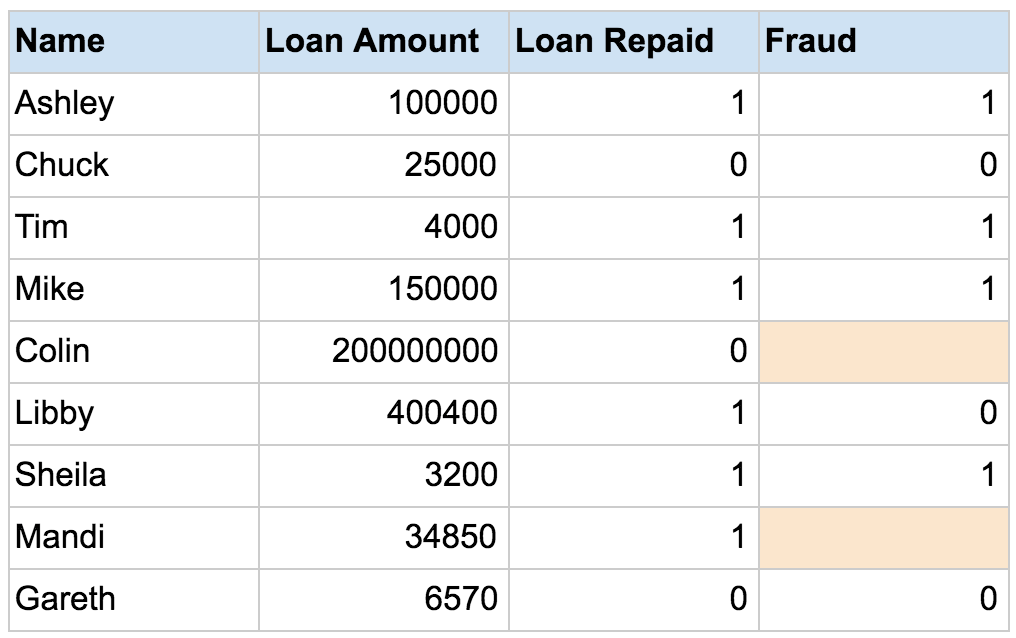
**Semi-supervised machine learning algorithms** :

→It fall somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training – typically a small amount of labeled data and a large amount of unlabeled data.

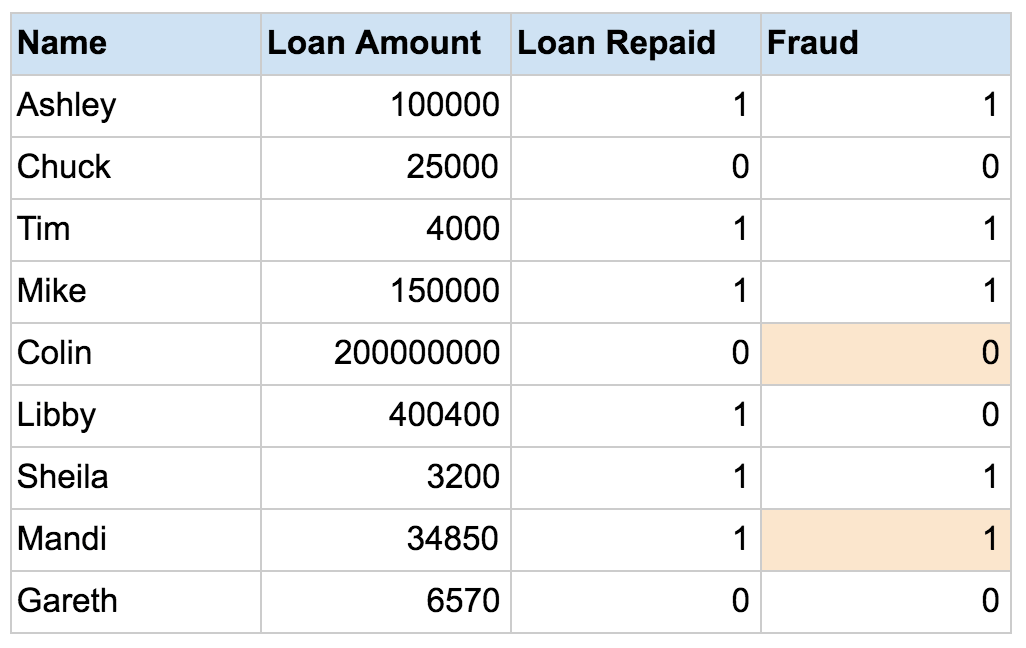
→The systems that use this method are able to considerably improve learning accuracy.

→Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring unlabelled data generally doesn’t require additional resources.

When you don’t have enough labeled data to produce an accurate model and you don’t have the ability or resources to get more data, you can use semi-supervised techniques to increase the size of your training data. For example, imagine you are developing a model intended to detect fraud for a large bank. Some fraud you know about, but other instances of fraud are slipping by without your knowledge. You can label the dataset with the fraud instances you’re aware of, but the rest of your data will remain unlabelled:



You can use a semi-supervised learning algorithm to label the data, and retrain the model with the newly labeled dataset:



Then, you apply the retrained model to new data, more accurately identifying fraud using supervised machine learning techniques. However, there is no way to verify that the algorithm has produced labels that are 100% accurate, resulting in less trustworthy outcomes than traditional supervised techniques.

A Semi-Supervised algorithm assumes the following about the data –

**Continuity Assumption:** The algorithm assumes that the points which are closer to each other are more likely to have the same output label.

**Cluster Assumption:** The data can be divided into discrete clusters and points in the same cluster are more likely to share an output label.

**Manifold Assumption:** The data lie approximately on a manifold of much lower dimension than the input space. This assumption allows the use of distances and densities which are defined on a [manifold](https://en.wikipedia.org/wiki/Manifold).

**Practical applications of Semi-Supervised Learning –**

**Speech Analysis:** Since labeling of audio files is a very intensive task, Semi-Supervised learning is a very natural approach to solve this problem.

**Internet Content Classification:** Labeling each webpage is an impractical and unfeasible process and thus uses Semi-Supervised learning algorithms. Even the Google search algorithm uses a variant of Semi-Supervised learning to rank the relevance of a webpage for a given query.

**Protein Sequence Classification:** Since DNA strands are typically very large in size, the rise of Semi-Supervised learning has been imminent in this field.

Reinforcement Machine Learning

**Reinforcement machine learning algorithms**:

|  |
| --- |
| Trying to control a system with minimal control is called reinforcement learning. |

→It is a learning method that interacts with its environment by producing actions and discovers errors or rewards.

→Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning.

→This method allows machines and software agents to automatically determine the ideal behavior within a specific context to maximize its performance.

→Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.

→The end result is to maximize the numerical reward signal.

→Reinforcement Learning is learning what to do and how to map situations to actions.

→ The learner is not told which action to take, but instead must discover which action will yield the maximum reward.

### **Reinforcement Learning Analogy:**

Think of **reinforcement learning** like teaching a pet (say, a dog) to perform tricks using **rewards** and **punishments**.

#### The Scenario:

Imagine you are training your dog to learn a new trick, like sitting on command.

* **The Environment**: The environment is where the dog learns, which is your living room or backyard. It’s the space where the dog can perform actions (like sitting, standing, jumping, etc.).
* **The Agent**: The dog is the agent that will learn through interactions.
* **Actions**: The actions are the things the dog can do, like sitting, standing, or walking.
* **Rewards**: When the dog performs the desired action (like sitting when you say “sit”), you give it a treat or praise (a **positive reward**).
* **Punishments**: If the dog does something wrong (like jumping on you instead of sitting), you might give a mild correction (a **negative punishment** or withhold rewards).

#### How It Works:

* **Exploration**: At first, the dog doesn’t know what to do. It might try different things—running, barking, jumping—because it’s exploring its environment and actions.
* **Learning**: Over time, the dog **learns** that sitting (the correct action) results in a treat, while jumping or not sitting does not. Through **trial and error**, it **discovers** which actions lead to the best outcomes (rewards).
* **Maximizing Reward**: The dog’s goal is to maximize the reward (getting treats), so it eventually **learns** that sitting on command is the best strategy.

In **reinforcement learning**:

* The **dog** is the **agent**.
* The **living room** is the **environment**.
* The **treats/praise** are the **rewards**.
* The dog is trying to figure out the best **action** (sitting) to get the maximum reward (the treat).

This process is iterative — the dog continues to improve and optimize its behavior over time based on feedback from the environment.

### **Reinforcement Learning Example: Self-Driving Car**

#### The Scenario:

Imagine you're developing a **self-driving car** that needs to learn how to navigate through city streets. The car must make decisions like when to stop, go, turn left or right, and how to avoid obstacles.

### **Key Elements in This Example:**

* **The Agent**: The self-driving car (AI) is the agent.
* **The Environment**: The environment is the **city streets**, with roads, traffic lights, other vehicles, pedestrians, and obstacles.
* **Actions**: The car can perform actions like:
  + **Accelerate** (go faster)
  + **Brake** (slow down or stop)
  + **Turn Left** or **Turn Right**
  + **Change Lanes**
* **Rewards**: The car gets feedback (rewards or punishments) based on its actions:
  + **Positive Reward**: If the car safely completes a task, like successfully turning without hitting anything or crossing an intersection at the right time, it gets a positive reward (for example, +10 points).
  + **Negative Reward**: If the car makes a mistake, like running a red light or hitting another vehicle, it gets a negative reward (for example, -100 points).

### **How It Works (Learning Process):**

**Exploration**:

The car starts with no knowledge about how to drive safely. It begins exploring by randomly choosing actions (accelerating, braking, turning) in different driving situations, like navigating intersections or avoiding pedestrians.

At this stage, the car may make mistakes (e.g., running a red light), but it doesn’t know what the right action is yet.

**Feedback**:

After each action, the car receives feedback from its environment. For example, if it runs a red light, it gets a **negative reward** (penalty). If it stops at the red light safely, it gets a **positive reward**.

The feedback can also come in the form of **delays**. For instance, if the car crashes, it might receive a **large negative reward** and have to "reset" its position.

**Learning**:

Over time, through **trial and error**, the car begins to learn which actions lead to **positive rewards** and which actions lead to **negative rewards**.

It learns that **stopping at red lights** and **yielding to pedestrians** lead to higher rewards (i.e., avoiding crashes, obeying traffic rules), while actions like **crashing into other cars** or **running red lights** lead to penalties.

**Optimization**:

* + After experiencing many different driving scenarios, the car becomes more **efficient** and **optimized** in its decisions.
  + For example, it learns the best way to **avoid traffic jams**, **optimize fuel efficiency**, and **choose the safest route** to a destination.

**Goal**:

* + The goal of the reinforcement learning process is to **maximize cumulative rewards** over time. The car’s overall objective is to drive safely, follow traffic rules, and reach its destination in the shortest possible time with minimal risks.

### **Real-World Impact of Reinforcement Learning in Self-Driving Cars:**

In a real-world scenario, companies like **Tesla**, **Waymo**, and **Uber** use reinforcement learning (along with other techniques) to improve the performance of their autonomous vehicles. The vehicles need to make real-time decisions based on their environment, such as reacting to pedestrians crossing the street, navigating through complex intersections, or changing lanes safely in traffic.

#### Key Challenges:

**Exploration vs. Exploitation**: In reinforcement learning, there's a balance between exploring new actions (to learn more about the environment) and exploiting what the model already knows (to maximize reward). For instance, the car might need to **explore** new routes to avoid traffic but will **exploit** familiar paths once it knows which ones are fastest and safest.

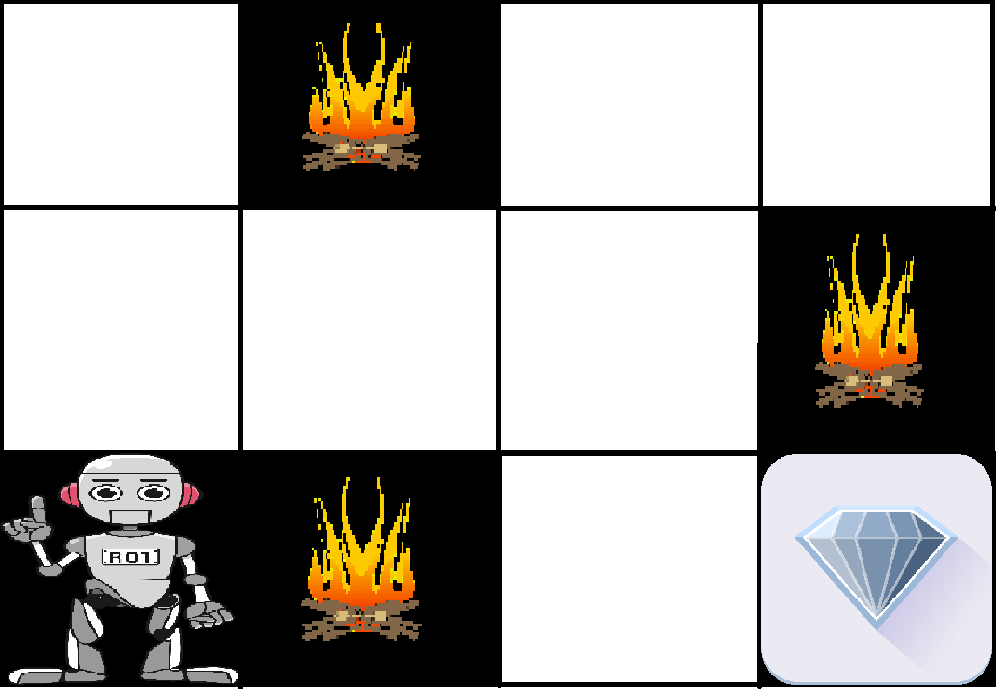
**Delays in Rewards**: In real driving, the rewards may not be immediate. For example, the car's action of stopping at a red light may only result in a positive reward much later (avoiding an accident or getting a green light). This is called the **temporal credit assignment problem** in reinforcement learning, where the agent needs to figure out how to connect long-term rewards to immediate actions.

**Example : Find valid puzzle path(Mize)**

The problem is as follows:

We have an agent and a reward, with many hurdles in between. The agent is supposed to find the best possible path to reach the reward. The following problem explains the problem more easily.

The below image shows robot, diamond and fire. The goal of the robot is to get the reward that is the diamond and avoid the hurdles that is fire.



→The robot learns by trying all the possible paths and then choosing the path which gives him the reward with the least hurdles.

→Each right step will give the robot a reward and each wrong step will subtract the reward of the robot.

→The total reward will be calculated when it reaches the final reward that is the diamond.

**Example** : **Learning cycle**

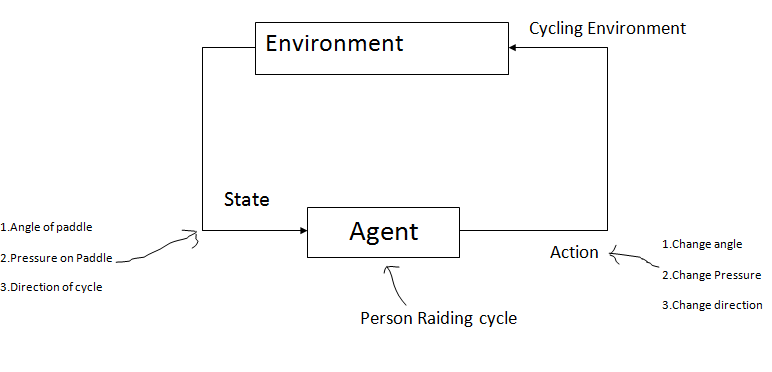
**Steps to learn cycle** :

1.Take cycle

2.You may fall

3.Take feedback

4.Try again



→It is learned from closed interaction

→Stochastic environment(Environment changes continuously)

→Noise delayed scalar evolution.

→Learn a policy. Maximize a long term performance.

**is cycling Supervised**?

No body tells us how much angle we have to put, how much gravity we have to put. So it is not

**is cycling unsupervised?**

By watching 100 videos on Youtube , you can not learn .So it is not unsupervised.

**Example** : **Child leant to walk**

Consider an example of a child learning to walk.

Here are the steps a child will take while learning  to walk:

1.The first thing the child will observe is to **notice** how you are walking.

You use two legs, taking a step at a time in order to walk. Grasping this concept, the child tries to replicate you.

2.But soon he/she will understand that before walking, the child has to stand up! This is a challenge that comes along while trying to walk. So now the child **attempts to get up,** staggering and slipping but still determinant to get up.

3.Then there’s another challenge to cope up with. Standing up was easy, but to **remain still** is another task altogether! Clutching thin air to find support, the child manages to stay standing.

4.Now the real task for the child is to start walking. But it’s easy to say than actually do it. There are so **many things to keep in mind**, like balancing the body weight, deciding which foot to put next and where to put it.

Sounds like a difficult task right? It actually is a bit challenging to get up and start walking, but you have become so use to it that you are not fazed by the task. But now you can get the gist of how difficult it is for a child.

Let’s formalize the above example, the “problem statement” of the example is **to walk**, where **the child is an agent** trying to manipulate the **environment (which is the surface on which it walks)** by **taking actions (viz walking)** and he/she tries to go from one **state (viz each step he/she takes)** to another.

The child gets a **reward** **(let’s say chocolate)** when he/she accomplishes a **submodule of the task (viz taking couple of steps)** and will not receive any chocolate**(a.k.a negative reward)** when he/she is not able to walk.

This is a simplified description of a reinforcement learning problem.

Diagram

Description automatically generated Diagram

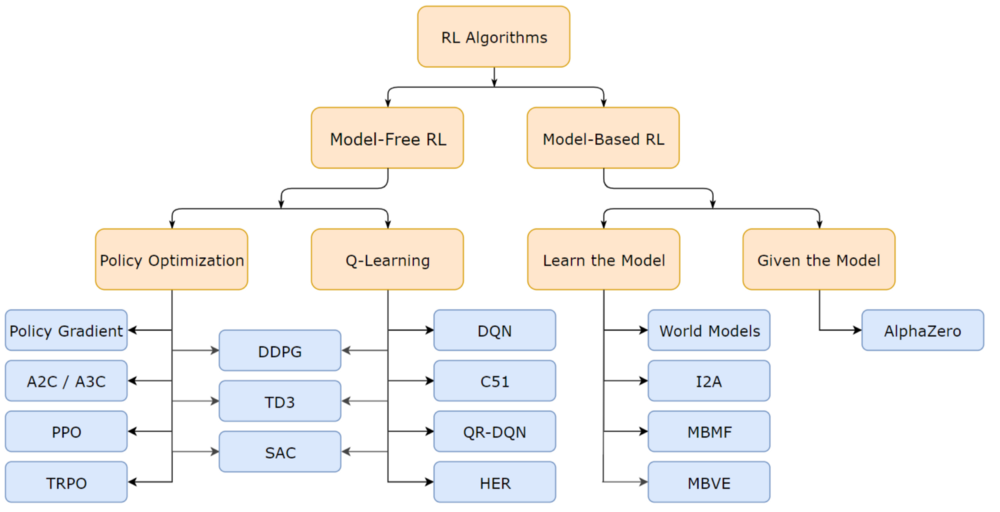
Description automatically generated

**List of Common Algorithms**:

Q-Learning

Temporal Difference (TD)

Deep Adversarial Networks



Feature Extraction Techniques:

1.PCA

2.LDA

3.Kernal PCA

4.Quadric Discriminal Analysis(QDA)

Clustering:

1.K-means clusterring

2.Hierichial clussstering

Model selection:

K-Fold cross validation

Grid Search

Assocition Rule Learning:

1.Apriori

2.Eclat

Deep Learning

**Deep Learning**:

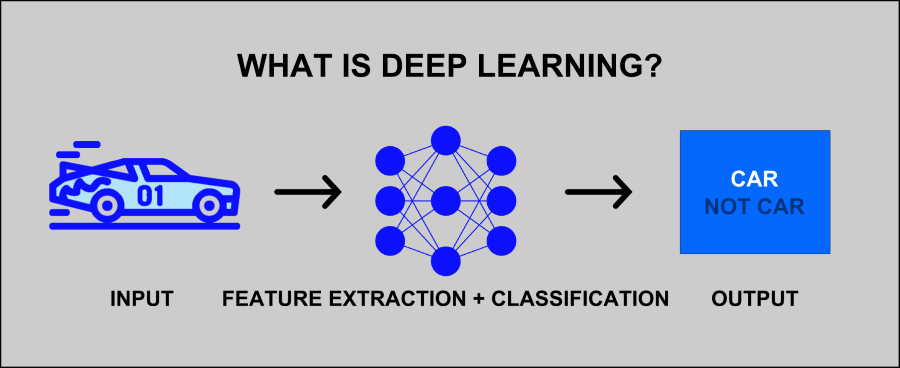
|  |
| --- |
| Deep Learning or Hierarchical Learning is a subset of Machine Learning in Artificial Intelligence that can imitate the data processing function of the human brain and create similar patterns the brain used for decision making. |

→Since neural networks imitate the human brain and so deep learning will do.

→In deep learning, nothing is programmed explicitly.

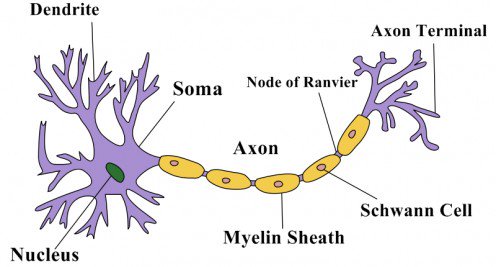
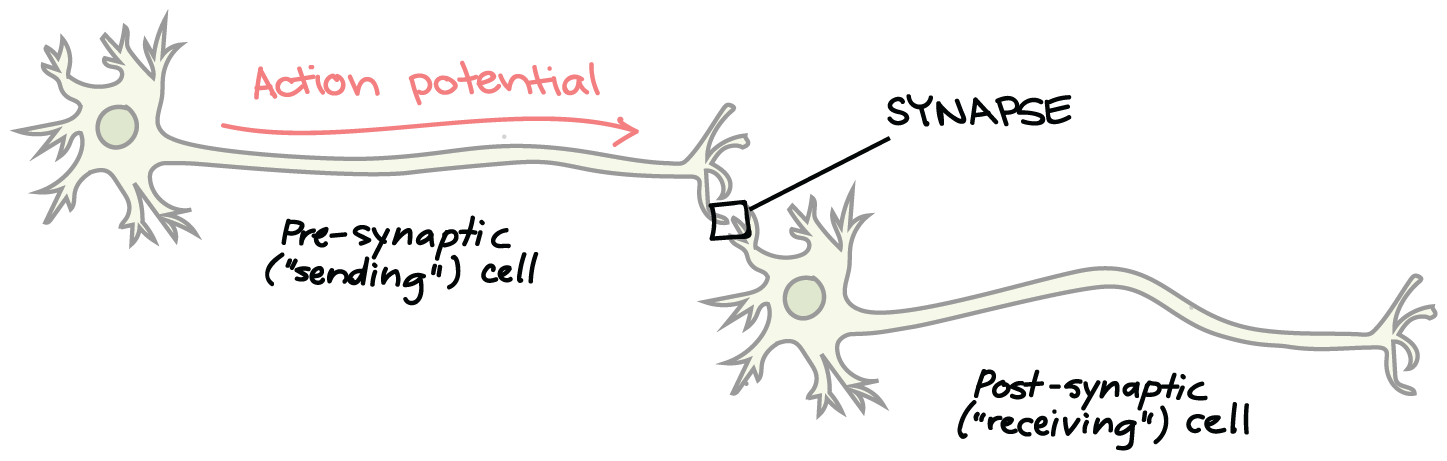
→Basically, it is a machine learning class that makes use of numerous nonlinear processing units to perform feature extraction as well as transformation.

→The output from each preceding layer is taken as input by each one of the successive layers.



Depending on the task and the available data, deep learning can be applied in both supervised and unsupervised settings.

**Neuron**:

**Axon**: It generated the signal

**Dendrite**: Receives the signal from other neuron

**Synapses:**

→Neuron-to-neuron connections are made onto the dendrites and cell bodies of other neurons. These connections, known as **synapses**, are the sites at which information is carried from the first neuron, the **presynaptic neuron**, to the target neuron (the **postsynaptic neuron**).

## **Neural Network**:

→A Neural Networks is made of an assortment of algorithms that are modelled on the human brain.

→These algorithms can interpret sensory data via machine perception and label or cluster the raw data.

→They are designed to recognize numerical patterns that are contained in vectors within which all the real-world data (images, sound, text, time series, etc.) has to be translated.

Essentially, the primary task of a Neural Networks is to cluster and classify the raw data – they group the unlabeled data based on the similarities found in the input data and then classify the data based on the labelled training dataset.

→Neural Networks can automatically adapt to changing input. So, you need not redesign the output criteria each time the input changes to generate the best possible result.

## **Deep Learning vs Neural Network**:

→While Deep Learning incorporates Neural Networks within its architecture, there’s a stark difference between Deep Learning and Neural Networks.

→Here we’ll shed light on the three major points of difference between Deep Learning and Neural Networks.

### 1. Definition

[Neural Networks](https://www.upgrad.com/blog/neural-network-tutorial-step-by-step-guide-for-beginners/) – It is a structure consisting of ML algorithms wherein the artificial neurons make the core computational unit that focuses on uncovering the underlying patterns or connections within a dataset, just like the human brain does while decision making.

Deep Learning – It is a branch of Machine Learning that leverages a series of nonlinear processing units comprising multiple layers for feature transformation and extraction. It has several layers of artificial neural networks that carry out the ML process. The first layer of the neural network processes the raw data input and passes the information to the second layer

The second later then processes that information further by adding additional information (for example, user’s IP address) and passes it to the next layer. This process continues throughout all layers of the Deep Learning network until the desired result is achieved.

### 2. Structure

A Neural Network consists of the following components:

* **Neurons** – A neuron is a mathematical function designed to imitate the functioning of a biological neuron. It computes the weighted average of the data input and passes the information through a nonlinear function, a.k.a. The activation function (for examples, the sigmoid).
* **Connection and weights** – As the name suggests, connections connect a neuron in one layer to another neuron in the same layer or another layer. Each connection has a weight value linked to it. Here, a weight represents the strength of the connection between the units. The aim is to reduce the weight value to decrease the possibilities of loss (error).
* **Propagation function** – Two propagation functions work in a Neural Network: forward propagation that delivers the “predicted value” and backward propagation that delivers the “error value.”
* **Learning rate** – Neural Networks are trained using Gradient Descent to optimize the weights. Back-propagation is used at each iteration to calculate the derivative of the loss function in reference to each weight value and subtract it from that weight. Learning rate decides how quickly or slowly you want to update the weight (parameter) values of the model.

A Deep Learning model consists of the following components:

* **Motherboard** – The motherboard chipset of the model is usually based on PCI-e lanes.
* **Processors** – The GPU required for Deep Learning must be determined according to the number of cores and cost of the processor.
* **RAM** – This is the physical memory and storage. Since Deep Learning algorithms demand greater CPU usage and storage area, the RAM must be huge.
* **PSU** – As the memory demands increase, it becomes crucial to employ a large PSU that can handle massive and complex Deep Learning functions.

### 3. Architecture

The architecture of a Neural Network includes:

* **Feed Forward Neural Networks** – This is the most common kind of Neural Network architecture wherein the first layer is the input layer, and the final layer is the output layer. All intermediary layers are hidden layers.
* **Recurrent Neural Networks** – This network architecture is a series of artificial neural networks wherein the connections between nodes make a directed graph along a temporal sequence. Hence, this type of network depicts temporal dynamic behaviour.
* **Symmetrically Connected Neural Networks** – These are similar to recurrent neural networks with the only difference being that in Symmetrically Connected Neural Networks, the connections between units are symmetrical (they have the same weight values in both directions).

**The architecture of a Deep Learning model includes:**

* **Unsupervised Pre-trained Networks** – As the name suggests, this architecture need no formal training since it is pre-trained on past experiences. These include Autoencoders, Deep Belief Networks, and Generative Adversarial Networks.
* **Convolutional Neural Networks** – This is a Deep Learning algorithm that can take in an input image, assign importance (learnable weights and biases) to different objects in the image, and also differentiate between those objects.
* **Recurrent Neural Networks** – Recurrent Neural Networks refer to a specific kind of artificial neural network that adds additional weights to the network to create cycles in the network graph so as to maintain an internal state.
* **Recursive Neural Networks** – This is a type of Deep Neural Network that is created by applying the same set of weights recursively over a structured input, to produce a structured prediction over or a scalar prediction on variable-size input structures by passing a topological structure.

### **Deep Learning Analogy: Teaching a Child to Recognize Objects**

Imagine you're teaching a child to recognize and classify objects in the world, like fruits, animals, or vehicles. The child is like the **deep learning model**, and you, as the teacher, help guide the learning process.

### **Step-by-Step Analogy**:

**The Child (Agent)**:

Think of the child as the **deep learning model**. Initially, the child knows very little about objects in the world but is eager to learn from experience and feedback.

**Examples of Objects (Training Data)**:

You give the child a set of examples, such as pictures of different types of fruits: apples, bananas, oranges, and strawberries. These images are similar to **training data** in deep learning.

Some of the pictures might be labeled ("This is an apple," "This is a banana"), and others might not be ("This is a picture of a fruit").

**Learning Features (Hidden Layers)**:

As the child looks at each picture, they start noticing **features** that make each fruit unique. For example, they might observe that apples are typically red or green, bananas are yellow, and strawberries have a particular shape.

These features are learned in **multiple layers** (just like deep learning models have layers of neurons). In the beginning, the child notices simple features like color, shape, or size. But as they see more examples, they start recognizing more complex features, like texture or patterns.

The child is going through a process of **feature extraction**, just as a deep learning model learns from data in layers of neurons.

**Feedback (Training Process)**:

* 1. Each time the child guesses the wrong object, you correct them ("No, that’s not an apple, that’s a banana"). This feedback helps the child improve their understanding.
  2. In deep learning, the feedback comes in the form of **error** (how wrong the model's prediction was), which is used to adjust the model's parameters (like weights in a neural network) to improve future predictions.

**Making Predictions (Using the Model)**:

* 1. After enough practice, the child begins to **recognize** new pictures of fruits they’ve never seen before. They might look at a picture of an unknown fruit and say, “I think that’s an apple, because it’s red and round.”
  2. In deep learning, once the model has been trained on enough data, it can **generalize** and make predictions on new, unseen data.

**Refining Understanding (Learning over Time)**:

* 1. The more pictures the child sees, the more their understanding of what makes an apple an apple (or a banana a banana) improves. Eventually, they might even recognize complex variations, like green apples or overripe bananas, and still classify them correctly.
  2. This is similar to how deep learning models get better with more data and iterations, refining their understanding of patterns and relationships.

### **How Deep Learning "Learns" to Recognize Objects (like an Apple):**

**Multiple Layers**: In a deep learning model (such as a **neural network**), the network consists of **multiple layers**. Each layer of neurons (or nodes) processes the data in a different way. In the context of **image recognition** (e.g., identifying fruits), the first layers might learn basic patterns, such as **edges**, **textures**, and **shapes**, while deeper layers learn more complex features, like **color patterns** or **distinct characteristics** that define an apple (or any other object).

**Layer 1 - Simple Features (Edges and Colors)**: The first layer might learn basic features, such as **edges** (where the apple's shape starts and ends), or **color patterns** (red or green). It might see a **red color** and say, "This could be an apple" because red is a common color for apples.

**Layer 2 - Combining Features**: As you go deeper into the network, the next layer might combine these basic features into more complex ones. For example, it might identify a **round shape** combined with the red color and say, "This is likely an apple," but it still doesn’t know for sure.

**Deeper Layers - More Complex Features**: In the deeper layers of the network, it might start to recognize that there are **variations** within the "apple" class, such as **green apples**, **yellow apples**, or apples with different textures. By combining these deeper learned features, the network can now confidently say, “This is an apple, whether it’s red, green, or yellow.”

**Final Decision**: By the time you get to the **output layer** of the network, the model has used all the learned features and is able to make a **final classification**. The output layer might say, "This fruit is an apple," based on the features it learned through the layers (such as color, shape, size, and texture). The network can also account for variations, like a green apple or a red apple, because it has learned those features in deeper layers.

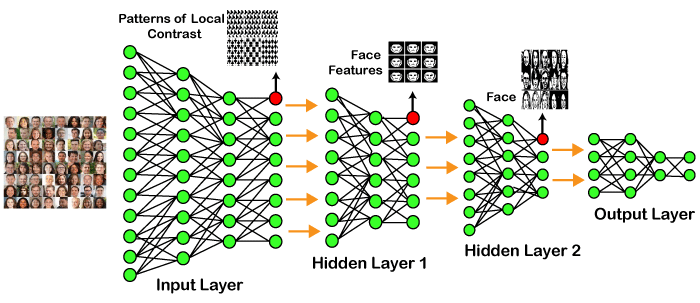
### **Example: Classifying an Apple**

* **Layer 1** might detect color and shape.
* **Layer 2** might combine that with texture and size.
* **Layer 3** might recognize that this combination of features is common in both red apples and green apples, increasing the model's certainty.
* **Layer 4 (Final Decision)** would confidently output **“Apple”**, regardless of the specific color, shape, or size.

### **So, is your understanding correct?**

Yes, increasing the number of layers helps the model learn **more complex features** and relationships. Initially, the network might be unsure (like identifying **red apples** as "apple" but still considering other possibilities). But as the network goes deeper and learns more complex features, it becomes more **confident** in its ability to say whether the fruit is an apple, regardless of whether it's **red**, **green**, or even a different variety.

Example of Deep Learning:



In the example given above, we provide the raw data of images to the first layer of the input layer.

After then, these input layer will determine the patterns of local contrast that means it will differentiate on the basis of colors, luminosity, etc.

Then the 1st hidden layer will determine the face feature, i.e., it will fixate on eyes, nose, and lips, etc. And then, it will fixate those face features on the correct face template.

So, in the 2nd hidden layer, it will actually determine the correct face here as it can be seen in the above image, after which it will be sent to the output layer.

Likewise, more hidden layers can be added to solve more complex problems, for example, if you want to find out a particular kind of face having large or light complexions.

So, as and when the hidden layers increase, we are able to solve complex problems.

**How Neural networks works:**

# How much will it cost to own house in Manhattan, New York?

Depends on:

**Area**:

cost Area

**Bedrooms**:

**Size**

The larger the **property area** and the home, the more expensive a house can be. If your home has several bedrooms, it is more likely to sell for a higher price as opposed to a bungalow with just one bedroom.

cost No of Bedrooms

**Design:**

An open layout with less walls and partitions, looks more spacious and this illusion can convince a home buyer to pay more because it looks bigger than its actual size.

**Location**:

1. One of the most important factors that affect property value

2. location will be at the top of the list.

3.

**Proximity to Local Transport**

Another important factor that affects how much a house is priced at would be it’s access to public transport. Properties near bus and train stations, as well as those near supermarkets, parks and hospitals are preferred by more buyers and sellers can demand higher prices.

To Airport

To school

**Condition**:

Will you pay more for a property that’s older but well-kept and in need of a little renovation or one that’s newer but requires major renovation? Chances are, you’ll go for the former. More often than not, the majority of home buyers will have the same answer. The condition of a home matters and will definitely increase or decrease its price.

**Neighbourhood**:

**Age**:

How long the structure has been in existence is also a deciding factor on pricing a home. Normally, the newer the home, the higher the price will be simply because the structure is almost new and no major repairs and renovations are needed however there are homes that have been built decades ago and are still worth far more than the **modern homes**. These are old homes that have historical significance and have been well-maintained.

**View**

Another aspect **potential buyers** consider when looking for a property to buy is the view they will see when they look out of the window. A home with a seaside view can demand for a much heftier price than one with a blocked view. Harbour views are enticing to buyers and if your property has this, it is best to include this in your **listing**.

|  |  |  |
| --- | --- | --- |
| Age | 2 years | 6 |
| 15 years | 2 |
| 100 years | 8 |
| Area(**square** feet.) | [500-600](https://www.theplancollection.com/house-plans/square-feet-500-600) | 4 |
| 600-700 | 5 |
| 700-900 | 6 |
| No of Bed rooms | 1 | 4 |
| 2 | 5 |
| 3 | 6 |
| Location | Far from City | 2 |
| Center of city | 3 |
| Schools,Transport | 4 |

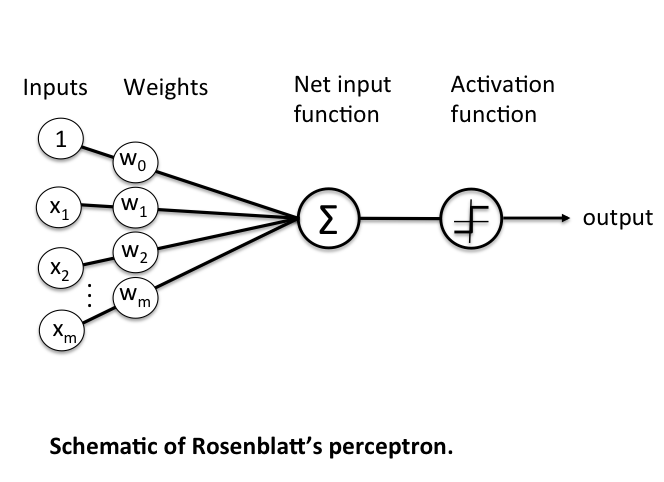
**Price=(Area\*Area Weight)+ (Age\*Age Weight)+ (# bedrooms\*# bedrooms Weight)+ (Location\*Location Weight)**

**Diagram

Description automatically generated**

**Perceptrons:**  
Considered the first generation of neural networks, ***perceptrons*** are simply computational models of a single neuron.

Also called feed-forward neural network, a perceptron feeds information from the front to the back. Training perceptrons usually require back-propagation, giving the network paired datasets of inputs and outputs. Inputs are sent into the neuron, processed, and result in an output. The error that is back propagated is usually the difference between the input and the output data. If the network has enough hidden neurons, it can always model the relationship between the input and output. Practically, their use is a lot more limited, but they are popularly combined with other networks to form new networks.



## **Architectures:**

**Deep Neural Networks:**

→It is a neural network that incorporates the complexity of a certain level, which means several numbers of hidden layers are encompassed in between the input and output layers. They are highly proficient on model and process non-linear associations.

**Deep Belief Networks:**

A deep belief network is a class of Deep Neural Network that comprises of multi-layer belief networks.

**Steps to perform DBN:**

1.With the help of the Contrastive Divergence algorithm, a layer of features is learned from perceptible units.

2.Next, the formerly trained features are treated as visible units, which perform learning of features.

3.Lastly, when the learning of the final hidden layer is accomplished, then the whole DBN is trained.

**Recurrent Neural Networks:**

→It permits parallel as well as sequential computation, and it is exactly similar to that of the human brain (large feedback network of connected neurons).

→Since they are capable enough to reminisce all of the imperative things related to the input they have received, so they are more precise.

## **Types of Deep Learning Networks:**

### **1. Feed Forward Neural Network**

→A feed-forward neural network is none other than an Artificial Neural Network ,which ensures that the nodes do not form a cycle. In this kind of neural network, all the perceptrons are organized within layers, such that the input layer takes the input, and the output layer generates the output.

→Since the hidden layers do not link with the outside world, it is named as hidden layers. Each of the perceptrons contained in one single layer is associated with each node in the subsequent layer. It can be concluded that all of the nodes are fully connected.

→It does not contain any visible or invisible connection between the nodes in the same layer. There are no back-loops in the feed-forward network.

→To minimize the prediction error, the backpropagation algorithm can be used to update the weight values.

**Applications:**

* Data Compression
* Pattern Recognition
* Computer Vision
* Sonar Target Recognition
* Speech Recognition
* Handwritten Characters Recognition

### **2. Recurrent Neural Network**

→They are yet another variation of feed-forward networks.

→Here each of the neurons present in the hidden layers receives an input with a specific delay in time.

→The Recurrent neural network mainly accesses the preceding info of existing iterations. For example, to guess the succeeding word in any sentence, one must have knowledge about the words that were previously used.

→It not only processes the inputs but also shares the length as well as weights crossways time.

→ It does not let the size of the model to increase with the increase in the input size. However, the only problem with this recurrent neural network is that it has slow computational speed as well as it does not contemplate any future input for the current state.

→It has a problem with reminiscing prior information.

**Applications:**

* Machine Translation
* Robot Control
* Time Series Prediction
* Speech Recognition
* Speech Synthesis
* Time Series Anomaly Detection
* Rhythm Learning
* Music Composition

### **3. Convolutional Neural Network**

→They are a special kind of neural network mainly used for image classification, clustering of images and object recognition.

→DNNs enable unsupervised construction of hierarchical image representations.

→To achieve the best accuracy, deep convolutional neural networks are preferred more than any other neural network.

**Applications:**

* Identify Faces, Street Signs, Tumors.
* Image Recognition.
* Video Analysis.
* NLP.
* Anomaly Detection.
* Drug Discovery.
* Checkers Game.
* Time Series Forecasting.

### **4. Restricted Boltzmann Machine**

→They are yet another variant of Boltzmann Machines.

→Here the neurons present in the input layer and the hidden layer encompasses symmetric connections amid them.

→However, there is no internal association within the respective layer. But in contrast to RBM, Boltzmann machines do encompass internal connections inside the hidden layer. These restrictions in BMs helps the model to train efficiently.

**Applications:**

* Filtering.
* Feature Learning.
* Classification.
* Risk Detection.
* Business and Economic analysis.

### **5. Autoencoders:**

### →An autoencoder neural network is another kind of unsupervised machine learning algorithm. Here the number of hidden cells is merely small than that of the input cells.

### →But the number of input cells is equivalent to the number of output cells.

### →An autoencoder network is trained to display the output similar to the fed input to force AEs to find common patterns and generalize the data.

### →The autoencoders are mainly used for the smaller representation of the input. It helps in the reconstruction of the original data from compressed data.

### →This algorithm is comparatively simple as it only necessitates the output identical to the input.

* **Encoder:** Convert input data in lower dimensions.
* **Decoder:** Reconstruct the compressed data.

**Applications:**

* Classification.
* Clustering.
* Feature Compression.

## **Deep learning applications:**

**Self-Driving Cars**In self-driven cars, it is able to capture the images around it by processing a huge amount of data, and then it will decide which actions should be incorporated to take a left or right or should it stop. So, accordingly, it will decide what actions it should take, which will further reduce the accidents that happen every year.

**Voice Controlled Assistance:**

When we talk about voice control assistance, then **Siri** is the one thing that comes into our mind. So, you can tell Siri whatever you want it to do it for you, and it will search it for you and display it for you.

**Automatic Image Caption Generation:**

Whatever image that you upload, the algorithm will work in such a way that it will generate caption accordingly. If you say blue colored eye, it will display a blue-colored eye with a caption at the bottom of the image.

**Automatic Machine Translation:**

With the help of automatic machine translation, we are able to convert one language into another with the help of deep learning.

## **Limitations:**

* It only learns through the observations.
* It comprises of biases issues.

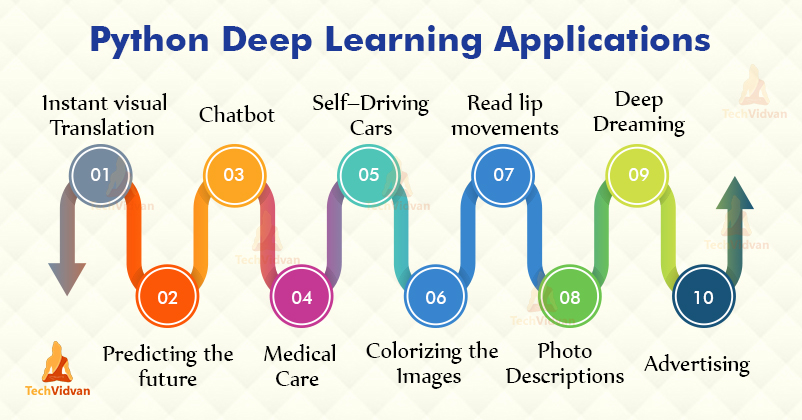
## **Advantages:**

* It lessens the need for feature engineering.
* It eradicates all those costs that are needless.
* It easily identifies difficult defects.
* It results in the best-in-class performance on problems.

## **Disadvantages:**

* It requires an ample amount of data.
* It is quite expensive to train.
* It does not have strong theoretical groundwork

Applications:



### **NLP**: **What is NLP?**

**NLP (Natural Language Processing)** is a subfield of artificial intelligence (AI) focused on enabling machines to understand, interpret, and generate human language. It involves the interaction between computers and human languages, and its goal is for computers to process and respond to text or spoken words in a way that is both meaningful and useful.

In simple terms, NLP helps machines "understand" and "work with" human language, whether it's in the form of text (like emails or books) or speech (like voice assistants).

### **Key Tasks in NLP**:

* **Text classification** (e.g., spam detection, sentiment analysis).
* **Machine translation** (e.g., translating text from one language to another).
* **Speech recognition** (e.g., converting spoken language to text).
* **Question answering** (e.g., answering questions based on a body of text).
* **Text generation** (e.g., generating coherent text like ChatGPT responses).

### **Analogy for NLP: A Translator or Interpreter**

Imagine you have a **translator** or an **interpreter** who helps people communicate across different languages.

#### **Scenario**:

* **Person A** speaks **English** and wants to convey a message to **Person B**, who only speaks **Spanish**.
* The **translator** listens to **Person A**'s words, understands the meaning of those words in **English**, and then speaks the equivalent words in **Spanish** for **Person B** to understand.

This is similar to what NLP does but in the digital world. Let's break down the analogy:

**Understanding the Message (Language Understanding)**:

The **translator** listens to the English message, decodes it, and understands its meaning. In NLP, this is similar to **text analysis** where the machine needs to understand the words, the context, and the relationships between them (e.g., breaking down sentences, identifying nouns, verbs, and adjectives).

**Converting the Message (Language Generation)**:

Once the translator understands the meaning, they convert it into Spanish. In NLP, this step corresponds to **text generation** or **machine translation**, where the system takes the understood meaning from one language (e.g., English) and translates it into another (e.g., Spanish) or generates new text (e.g., ChatGPT’s responses).

**Context Awareness**:

Just like an experienced translator knows that "cold" in a weather context is different from "cold" in an emotional context, NLP models need to be aware of **context**. The meaning of a word can change depending on how it’s used in a sentence, and an NLP system must learn these nuances.

**Helping with Communication (Practical Application)**:

The translator allows **Person A** and **Person B** to communicate, despite speaking different languages. Similarly, NLP enables machines to **communicate with humans** in natural language—whether it's responding to queries, translating text, or generating meaningful dialogue (like ChatGPT's responses to your questions).

### **Dimensionality Reduction?**

**Dimensionality reduction** is a technique used in machine learning and data analysis to reduce the number of input variables or features in a dataset while retaining as much important information as possible. High-dimensional datasets (datasets with many features or variables) can be challenging to work with because:

* They are computationally expensive to process.
* They may lead to overfitting in models.
* They can be difficult to visualize or interpret.

Dimensionality reduction helps solve these problems by simplifying the data without losing critical information.

### **Common Techniques for Dimensionality Reduction:**

1. **Principal Component Analysis (PCA)**: PCA reduces the dimensions of the data by projecting it onto a smaller set of "principal components" that capture the most variance in the data.
2. **t-Distributed Stochastic Neighbor Embedding (t-SNE)**: t-SNE is often used for visualizing high-dimensional data in 2 or 3 dimensions.
3. **Linear Discriminant Analysis (LDA)**: LDA is used for supervised dimensionality reduction, where the data is projected in such a way that it maximizes the separation between different classes.

### **Example of Dimensionality Reduction:**

Imagine you have a dataset of **photos of fruits** (apples, oranges, bananas) where each photo has many **pixels** (features), each with a red, green, or blue value. This creates a **high-dimensional** dataset, as each photo might have thousands or millions of features (depending on the resolution of the image).

If you want to build a machine learning model to classify the fruits, it's computationally expensive to work with all the pixel values. Dimensionality reduction can be applied to reduce the number of pixels (features) while still retaining enough information to differentiate between apples, oranges, and bananas.

* **Before Dimensionality Reduction**: Each photo might have thousands of features (pixels).
* **After Dimensionality Reduction (e.g., PCA)**: The dataset could be reduced to a few key features (for example, color patterns or shapes), which still contain enough information to classify the fruits accurately.

### **Analogy for Dimensionality Reduction:**

Imagine you're packing for a trip, and you need to fit everything into a suitcase. You have **a lot of items**, but the suitcase has limited space. Instead of just trying to stuff everything into the suitcase, you decide to **select the most important items** that you’ll actually need, reducing the total number of items you bring along.

* **High-Dimensional Data (Many Items)**: Initially, you have all sorts of items, like clothes, shoes, books, gadgets, and toiletries. These are like the features in your high-dimensional dataset.
* **Dimensionality Reduction (Selecting Important Items)**: To reduce the load, you decide to pick the most essential items. You might leave behind a few unnecessary things (e.g., a second pair of shoes), but you ensure that you’re still bringing everything important for your trip.
* **Result**: After selecting the most important items, your suitcase is lighter and more manageable. Similarly, after dimensionality reduction, the data is more manageable while still retaining key information to make decisions or predictions.

### **Anomaly Detection in Machine Learning**

**Anomaly detection** in machine learning refers to the process of identifying patterns or data points that deviate significantly from the normal behavior in a dataset. These anomalies are also called **outliers** or **novelties**. In machine learning, anomaly detection can be supervised, semi-supervised, or unsupervised depending on the availability of labeled data.

Anomaly detection is widely used in various domains, including:

* **Fraud detection** (e.g., credit card fraud)
* **Intrusion detection** (e.g., cybersecurity)
* **Manufacturing quality control** (e.g., defect detection in products)
* **Healthcare** (e.g., detecting unusual patterns in medical records)

### **Types of Anomaly Detection in Machine Learning:**

**Supervised Anomaly Detection**:

* + In **supervised learning**, you have **labeled data** where anomalies are already identified. The model learns to distinguish between normal and anomalous data points by training on this labeled dataset.
  + Example: A dataset of financial transactions where each transaction is labeled as either **fraudulent** or **non-fraudulent**. The model can be trained to classify future transactions into these categories.

**Unsupervised Anomaly Detection**:

* + In **unsupervised learning**, you do not have labeled data. The model tries to identify anomalies by observing patterns in the data and recognizing which data points deviate from the expected pattern.
  + Example: Network traffic data where no explicit label tells you which traffic is normal and which is an attack. The model finds anomalies based on patterns in the traffic flow.

**Semi-Supervised Anomaly Detection**:

* + In **semi-supervised learning**, the model is trained on mostly normal (non-anomalous) data and then tested to identify anomalies. This approach is useful when you have a limited amount of labeled anomaly data but a larger dataset of normal data.
  + Example: Detecting unusual activity in a system where most data is normal, but you only have a few examples of anomalous events.