PROGRAMME DESIGN COMMITTEE

Prof. (Retd.) S.K. Gupta , IIT, Delhi
Prof. Ela Kumar, IGDTUW, Delhi
Prof. T.V. Vijay Kumar JNU, New Delhi
Prof. Gayatri Dhingra, GVMITM, Sonipat
Mr. Milind Mahajan,. Impressico Business Solutions, New Delhi
Sh. Shashi Bhushan Sharma, Associate Professor, SOCIS, IGNOU
Sh. Akshay Kumar, Associate Professor, SOCIS, IGNOU
Dr. P. Venkata Suresh, Associate Professor, SOCIS, IGNOU
Dr. V.V. Subrahmanyam, Associate Professor, SOCIS, IGNOU
Sh. M.P. Mishra, Assistant Professor, SOCIS, IGNOU
Dr. Sudhansh Sharma, Assistant Professor, SOCIS, IGNOU

COURSE DESIGN COMMITTEE

Prof. T.V. Vijay Kumar, JNU, New Delhi
Prof. S.Balasundaram, JNU, New Delhi
Prof D.P. Vidyarthi, JNU, New Delhi
Prof. Anjana Gosain, USICT, GGSIPU, New Delhi
Dr. Ayesha Choudhary, JNU, New Delhi
Sh. Shashi Bhushan Sharma, Associate Professor, SOCIS, IGNOU
Sh. Akshay Kumar, Associate Professor, SOCIS, IGNOU
Dr. P. Venkata Suresh, Associate Professor, SOCIS, IGNOU
Dr. V.V. Subrahmanyam, Associate Professor, SOCIS, IGNOU
Sh. M.P. Mishra, Assistant Professor, SOCIS, IGNOU
Dr. Sudhansh Sharma, Assistant Professor, SOCIS, IGNOU

SOCIS FACULTY

Prof. P. Venkata Suresh, Director, SOCIS, IGNOU
Prof. V.V. Subrahmanyam, SOCIS, IGNOU
Dr. Akshay Kumar, Associate Professor, SOCIS, IGNOU
Dr. Naveen Kumar, Associate Professor, SOCIS, IGNOU (on EOL)
Dr. M.P. Mishra, Associate Professor, SOCIS, IGNOU
Dr. Sudhansh Sharma, Assistant Professor, SOCIS, IGNOU
Dr. Manish Kumar, Assistant Professor, SOCIS, IGNOU

PREPARATION TEAM

Mr. VenuGopal, General Manager.(Writer- Unit 9)
Sify Technologies, Noida,U.P

Dr.Sudhansh Sharma, (Writer – Unit 10)
Assistant .Professor, SOCIS, IGNOU
(Unit-10 : Partially Adapted from MCS 043
Advanced Database Management Systems)

Dr. Parmod Kumar, Assistanr Professor(Sr.G.)
Department of Computer Applications,
SRM Institute of Science and Technology,
Delhi NCR Campur Modinagar,U.P.(Writer-Unit 11)

Prof Anjana Gosain (Content Editor)
USICT-GGSIPU,Delhi

Dr. Rajesh Kumar(Language Editor)
SOH, IGNOU, New Delhi

Prof. Sachin Kumar (Writer-Unit 12)
Department of Computer Science and Engineering
Ajay Kumar Garg Engineering College, Ghaziabad, U.P.

Course Coordinator: Dr.Sudhansh Sharma,

Print Production

Sh Sanjay Aggarwal,Assistant Registrar, MPDD

, 2022

ISBN-

Further information on the Indira Gandhi National Open University courses may be obtained from the University's office at Maidan Garhi, New Delhi-110068.
Printed and published on behalf of the Indira Gandhi National Open University, New Delhi by MPDD, IGNOU.

UNIT 9 INTRODUCTION TO MACHINE LEAR METHODS

9.0 INTRODUCTION

After Artificial Intelligence was introduced, in Computing World. There was a need for a machine that would automatically make things better. This needs to be kept in check, so there should be some rules that apply to all learning processes.

The main goal of machine learning, even at its most basic level, is to be able to analyse and adapt data on its own and make decisions based on calculations and analyses. Machine learning is a way to try to improve computers by imitating how the human brain learns. A computer that doesn't have intelligence is just a fast machine for processing data. The devices that don't have AI or ML are just data processing units that use the information they are given. Machine Learning is what we need to make devices that can make decisions based on data.

To get to this level of intelligence, you need to put algorithms and data into a machine in a way that lets it make decisions.

For example, Real-time GPS data is used by Maps on devices to show the quickest and fastest route. Several algorithms, such as the shortest path (Dijkstra's algorithm) and the travelling salesman (an algorithm that works like water flow), can be used to make the decision (WFA). These are algorithms that have been used and can be made better, but they are useful for learning. Here, we can see that the Machine, which is your computer or mobile device, uses GPS coordinates, traffic data based on density, and predefined map routes to figure out the fastest way to get from Point A to Point B..

This is one of the simplest examples that can help us understand how Machine Learning can help in independent decision making by devices and how it can help in making decision making easier and more accurate.

The accuracy of the data as a whole is a topic of debate since decisions based on the data might be accurate, but it is one of the issues whether or not they are acceptable within the limitations.

Consequently, it is necessary to set these boundaries for the entirety of the machine learning algorithms and engines.

Simplest example would be if we instruct an auto driving car to reach a destination at a specified time. IT should also work within the legal boundaries of the land not to break traffic rules to achieve the desired result. The Boundaries and restriction cannot be ignored as they are very important for any Self learning system.

Data/Inputs is a soul of all the business .Data has been a key component for making any decision . Data is the key to successes from prehistoric era. The more you have the data more is the probability of making the right decision. Machine learning is the key to unlock new world where customer data , corporate data , demographic data, or related dimension data relevant to the decision can help you make right and more informed decision to stay ahead of competition.

Both artificial intelligence and statistics research groups contributed to the development of machine learning. Companies like Google, Microsoft, Facebook, and Amazon all use machine learning as part of their decision-making processes.

The most common applications of machine learning nowadays are to interpret and investigate cyber phenomena, to extract and project the future values of those phenomena, and to detect anomalies.

There are a number of open-source solutions for machine learning that can be used with API calls or without programming. Some of the Open-source Machine Learning projects, such as Weka, Orange, and Rapid-Miner. To see how data that has been processed by an algorithm looks, you can put the results into tools like Tableau, Pivotal, or Spotfire and use them to make dashboards and workflow strategies.

Michie et al. (D. Michie, 1994) says that Machine Learning usually refers to automatic computing procedures based on logical or binary operations that learn how to do a task by looking at a series of examples. Machine learning is used in a lot of ways today, but whether or not they are all ready is up for debate. There is a lot of room for improvement when it comes to accuracy, which is a process that never ends and changes every day.

9.1 OBJECTIVES

After going through this unit, you should be able to:

- Understand the basics of Machine learning
- Identify various techniques of Machine Learning
- Understand the concept of Reinforcement Learning
- Understand the concept of Deep Learning
- Understand Ensemble Methods

9.2 INTRODUCTION TO MACHINE LEARNING

Understanding data, describing the characteristics of a data collection, and locating hidden connections and patterns within that data are all necessary steps in the process of developing a model. These steps can be accomplished through the application of statistics, data mining, and machine learning. When it comes to finding solutions to business issues, the methods and tools that are employed by these fields share a lot in common with one another.

The more conventional forms of statistical investigation are the origin of a great deal of the prevalent data mining and machine learning techniques. Data scientists have a background in technology and also have expertise in areas such as statistics, data mining, and machine learning. This allows them to collaborate effectively across all fields.

The process of "data mining" refers to the extraction of information from data that is latent, previously unknown, and has the potential to be beneficial. Building computer algorithms that can automatically search through large databases for recurring structures or patterns is the goal of this project. In the event that robust patterns are discovered, it is likely that they will generalise to enable accurate predictions on future data.

In the renowned book "Data Mining: Practical Machine Learning Tools and Techniques," written by Ian Witten and Eibe Frank, the subject matter is thoroughly covered. The activity known as "data mining" refers to the practise of locating patterns within data. The procedure needs to be fully automatic or, at the very least, semiautomatic. The patterns that are found have to be significant in the sense that they lead to some kind of benefit, most commonly an economic one. Consistently and in substantial amounts, the statistics are there to be found.

Machine learning, on the other hand, is the core of data mining's technical infrastructure. It is used to extract information from the raw data that is stored in databases; this information is then expressed in a form that is understandable and can be applied to a range of situations.

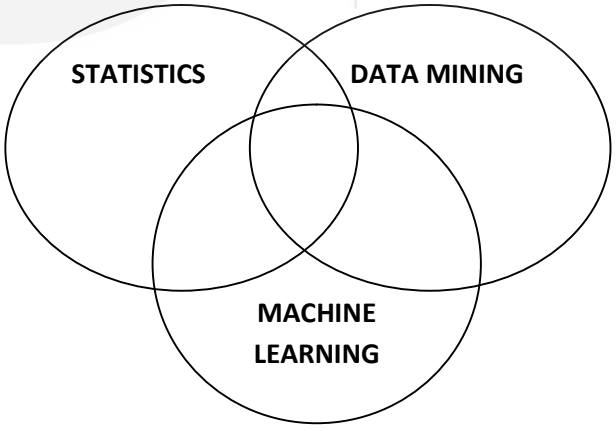


Figure 1 : Machine Learning – Data Mining - Statistics

We learned the differences between Machine Learning and Data Mining from the conversation; nevertheless, because all three, machine learning, data mining, and statistics, are intertwined, we must grasp their relationships. So, how do machine learning and statistics differ? In reality, there is no clear cut between machine learning and statistics because data analysis techniques are a continuum—and a multidimensional one at that. Some are derived from statistical skills, while others are more strongly linked to the type of machine learning that has emerged from computer science. Both sides have had quite diverse traditions throughout history. If forced to choose one point of emphasis, statistics may have been more concerned with testing hypotheses, whereas machine learning has been more interested with articulating the generalisation process as a search through possible hypotheses. This, however, is an exaggeration: Many machine learning algorithms do not require any searching at all, and statistics is significantly more than just hypothesis testing.

Most learning algorithms use statistical tests to build rules or trees and fix models that are "overfitted," or too dependent on the details of the examples that were used to make them. So, a lot of statistical thinking goes into the techniques we will talk about in this unit. Statistical tests are used to evaluate and validate machine learning models and algorithms.

Machine learning is when a computer learns how to do a task by using algorithms that are logical and can be turned into models that can be used. Artificially intelligent communities are the main reason why Machine Learning is growing.The most important factor contributing to this expansion was that it assisted in the collection of statistical and computational methods that could automatically construct usable models from data. Companies such as Google, Microsoft, Facebook, and Netflix have been putting in consistent effort over the past decade to make this more accurate and mature.

The primary function or application of machine learning algorithms can be summarized as follows:

(a) To gain an understanding of the cyber phenomenon that produced the data that is being investigated;
(b) To abstract the understanding of underlying phenomena in the form of a model;
(c) To predict the future values of a phenomenon by using the model that was just generated; and
(d) To identify anomalous behavior exhibited by a phenomenon that is being observed.

There are various open-source implementations of machine learning algorithms that can be utilised with either application programming interface (API) calls or non-programmatic applications. These methods can also be used in conjunction with each other. Weka, Orange, and Rapid Miner are a few instances of open-source application programming interfaces. These algorithms' outputs can be fed into visual analytics tools like Tableau4 and Spotfire5, which can then be used to build dashboards and actionable pipelines.

Almost all of the Frameworks have emphasised decision-tree techniques, in which classification is determined by a series of logical processes. Given enough data (which may be a lot!), these are capable of representing even the most complex problems. Other techniques, such as genetic algorithms and inductive logic procedures (ILP), are currently in development and, in theory, would allow us to deal with a wider range of data, including cases where the number and type of attributes vary, where additional layers of learning are superimposed, and where attributes and classes are organised hierarchically, and so

on. Machine Learning seeks to provide classification phrases that are basic enough for humans to understand. They must be able to sufficiently simulate human reasoning in order to provide insight into the decision-making process. Background knowledge, including statistical techniques, can be used in development, but operation is assumed to be without human interference.

The expression “To learn” can be understood as :

- To learn means to acquire knowledge.to gain knowledge or understanding of something through experiencing it or through learning about it (some art or practice)
- To gain experience with something, learn a new skill, or master a talent.
- To memorize (something), to acquire something through the experience, example, or practice of doing so.

Machine Learning is a methodology for automatically improving computer systems through the process of developing using experience and implementing a learning process. There are various techniques for imparting machine learning and we will learn about few of those in the subsequent section.

Check Your Progress - 1

Q1. How machine learning differs from Artificial intelligence ?

……
……

Q2 Briefly discuss the major function or use of Machine learning algorithms.

……
……

9.3 TECHNIQUES OF MACHINE LEARNING

Machine learning uses various algorithms to improve, describe, and predict outcomes by repeated learning from data. It is possible to make models that are more accurate as the algorithms learn from the training data. A machine learning model is what you get when you use data to train your machine learning algorithm. After it has been trained, a model will give you an output when you give it an input. A predictive model is made, for example, by a predictive algorithm. Then, when you put data into the predictive model, you'll get a prediction based on the data that was used to train the model. At the moment, analytics models can't be made without machine learning.

Machine learning approaches are needed to make prediction models more accurate. Depending on the type and amount of data and the business problem being solved, there are different ways to approach the problem. In this section, we talk about the machine learning cycle.

The Machine Learning Cycle: Making a machine learning application is similar to making a machine learning algorithm work, which is an iterative process. You can't just train a model once and leave it alone, because data changes, preferences change, and new competitors come along. So, when your model goes into production, you need to keep it updated. Even though you won't need as much training as when you created the model, don't expect it to run on its own.

Figure 2 :Machine Learning Cycle at a Glance

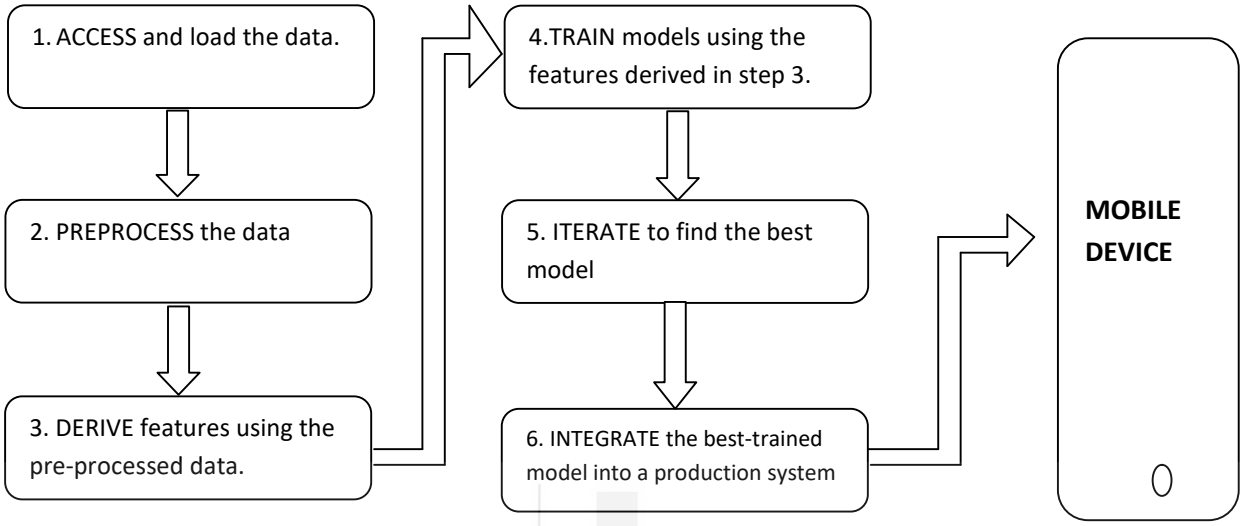


Figure 2 : Machine Learning Cycle at a Glance

To use machine learning techniques effectively, you need to know how they work. You can't just use them without knowing how they work and expect to get good results. Different techniques work for different kinds of problems, but it's not always clear which techniques will work in a given situation. You need to know something about the different kinds of solutions. We talk about a very large number of techniques.

One step in the machine learning cycle is choosing the right machine learning algorithm. So, let's look at how the machine learning cycle works.

The steps in the machine learning cycle are as follows:

1. *Data Identification*
2. *Data Preparation*
3. *Selection of machine learning algorithm:*
4. *Training the algorithm to develop a model*
5. *Evaluating the model*
6. *Deploying the model*
7. *Performing Prediction*
8. *Assess the predictions*

When your model has reached the point where it can make accurate predictions, you can restart the process by re-evaluating it using questions such as "Is all of the information important?" Exist any more data sets that could be used to improve the accuracy of the predictions? You may maintain the usefulness of your applications that are based on machine learning by continually improving the models and assessing new approaches.

When should you use machine learning? Think about using machine learning when you have a hard task or problem that involves a lot of data and many different factors but no formula or equation to solve it. For example, machine learning is a good choice if you need to deal with situations like face and speech

recognition, fraud detection by analysing transaction records, automated trading, energy demand forecasting, predicting shopping trends, and many more.

When it comes to machine learning, there's rarely a straight line from the beginning to the end. Instead, you'll find yourself constantly iterating and trying out new ideas and methods.

This unit talks about a step-by-step process for machine learning and points out some important decision points along the way. The most common problem with machine learning is getting your data in order and finding the right model. Here are some of the most important things to worry about with the data:

• **Data comes in all shapes and sizes** : There are many different kinds of data. Datasets from the real world can be messy, with missing values, and may be in different formats. You might just have simple numeric data. But sometimes you have to combine different kinds of data, like sensor signals, text, and images from a camera that are being sent in real time.

• **Preprocessing your data might require specialized knowledge and tools** : You might need specialised tools and knowledge to prepare your data before you use it. For example, you need to know a lot about image processing to choose features to train an object detection algorithm. Preprocessing needs to be done in different ways for different kinds of data.

• **It takes time to find the best model to fit the data :** Finding the best model to fit the data takes time. Finding the right model is like walking a tightrope. Highly flexible models tend to fit the data too well by trying to explain small differences that could just be noise. On the other hand, models that are too simple might assume too much. Model speed, accuracy, and complexity are always at odds with each other.

Does it appear to be a challenge? Try not to let this discourage you. Keep in mind that the process of machine learning relies on trial and error. You merely go on to the next method or algorithm in the event that the first one does not succeed. On the other hand, a well-organized workflow will assist you in getting off to a good start.

Every machine learning workflow begins with three questions:

- What kind of information do you have to work with?
- How do you want to learn something from it?
- How and where will these new ideas come from?

Your answers to these questions help you decide whether to use supervised or unsupervised learning algorithm, before proceeding to other details we will discuss these two types of learning algorithms.

Fundamentally Machine learning involves two classes of Learning algorithms:

a) Supervised learning, which requires training a model with data whose inputs and outputs are already known in order for the model to be able to predict future outputs, such as whether or not an email is authentic or spam or whether or not a tumor is cancerous. Classification models classify given data into categories. Imaging for medical purposes, speech recognition, and rating credit are a few examples of typical applications.

b) Unsupervised learning analyses data to uncover previously unknown patterns or structures. It is used to infer conclusions from sets of data that contain inputs but no tagged answers. The most prevalent method of learning without being observed is clustering. Exploratory data analysis is used to uncover hidden patterns or groups in data. Clustering can be used for gene sequence analysis, market research, and object recognition.

Note: In semi-supervised learning, algorithms are trained on small sets of labelled data before being applied to unlabeled data, like in unsupervised learning. When there is a dearth of quality data, this method is frequently used.

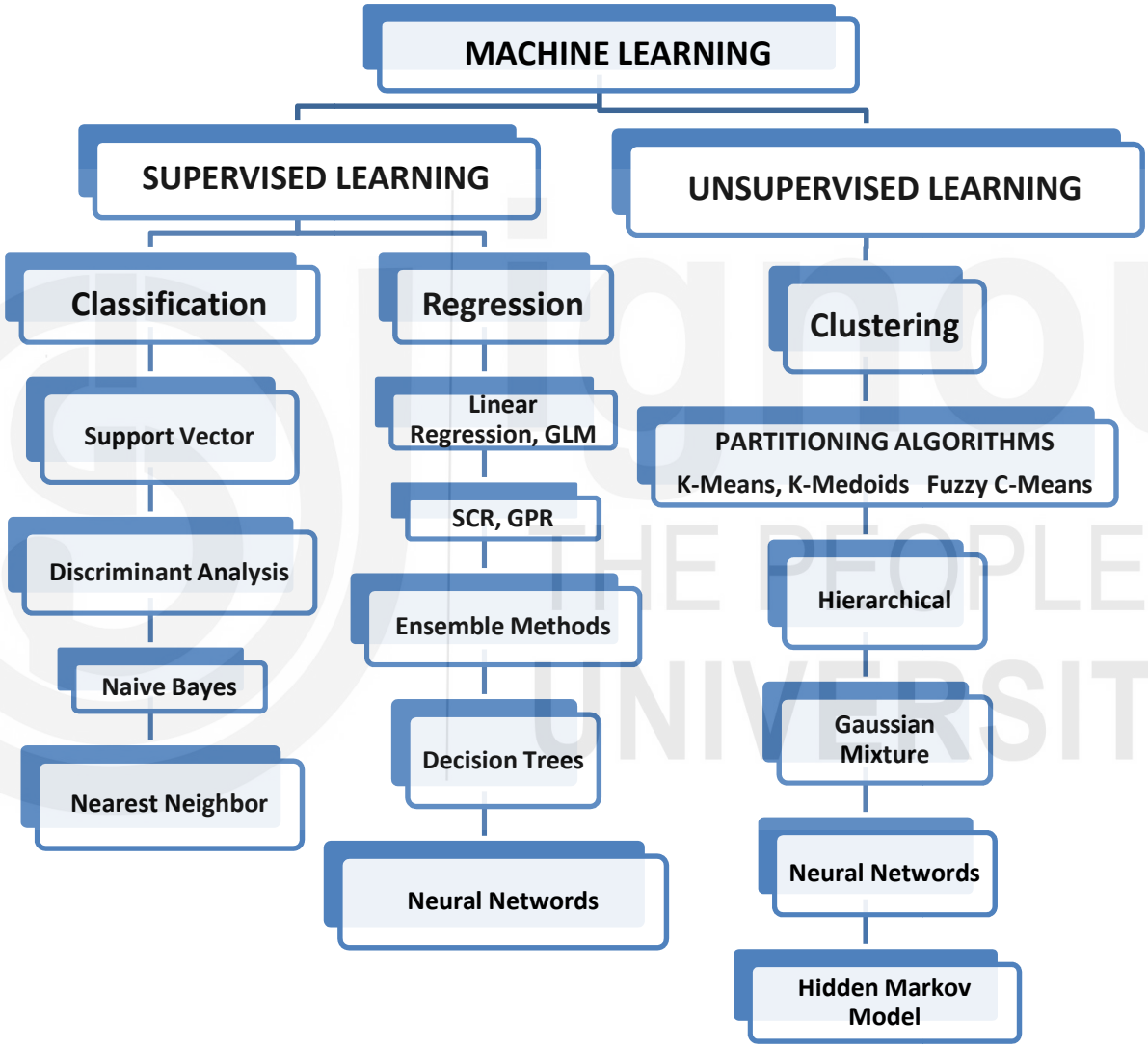


Figure – 3 Machine Learning Algorithms

"How Do You Choose Which Algorithm to Use?" is a crucial question. There are numerous supervised and unsupervised machine learning algorithms, each with its own learning strategy. This can make picking the appropriate one difficult. There is no alternative solution or strategy that will work for everyone. It takes some trial and error to find the proper algorithm. Even the most seasoned data scientists can't predict whether or not an algorithm would work without putting it to the test. However, the size and

type of data you're working with, the insights you want to gain from the data, and how those insights will be used all go into the algorithm you choose.

• *If you need to train a model to produce a forecast, such as the future value of a continuous variable like temperature or a stock price, or a classification, such as determining what kind of automobile is on a webcam footage, go with supervised learning.*

• *If you want to look at your data and train a model to identify an appropriate way to represent it internally, for as by grouping it, use unsupervised learning.*

The purpose of supervised machine learning is to create a model capable of making predictions based on data even when there is ambiguity. A supervised learning technique trains a model to generate good predictions about the response to new data using a known set of input data and previous responses to the data (output).

Using Supervised Learning to Predict Heart Attacks as an Example: Assume doctors want to determine if someone will suffer a heart attack in the coming year. They have information on former patients' age, weight, height, and blood pressure. They know if any of the previous patients had heart attacks within a year. The challenge is to create a model using existing data that can predict if a new person will have a heart attack in the coming year.

Supervised Learning Techniques*:* Every supervised learning method may be broken down into one of two categories: classification or regression. Methods like as classification and regression, which are employed in supervised learning, are put to use in the development of models that are able to forecast the future.

• **Classification techniques** : Classification methods make predictions about discrete outcomes, such as whether an e-mail is genuine or spam or if a tumour is cancerous or not. Classification models classify incoming data into categories. Imaging for medical purposes, speech recognition, and rating credit are a few examples of typical applications.

• **Regression methods** : Predictions can be made with regression algorithms about things like shifts in temperature or alterations in the quantity of power consumed. The most typical applications are stock price predicting, handwriting recognition, electricity load forecasting, acoustic signal processing, and other similar tasks.

Note:

- Is it possible to tag or categorise your data? Use classification techniques if your data can be divided into distinct groups or classes.

- Working with a collection of data? Use regression techniques if your answer is a real number, such as the temperature or the time until a piece of equipment fails.

- Binary vs. Multiclass Classification: Before you start working on a classification problem, figure out whether it's a binary or multiclass problem. A single training or test item (instance) can only be classified into two classes in a binary classification task, such as determining whether an email

is real or spam. If you wish to train a model to categorise a picture as a dog, cat, or other animal, for example, a multiclass classification problem might be separated into more than two categories. Remember that a multiclass classification problem is more difficult to solve since it necessitates a more sophisticated model. Certain techniques (such as logistic regression) are specifically intended for binary classification situations. These methods are more efficient than multiclass algorithms during training.

Now it's time to talk about the role of algorithms in machine learning. Algorithms are a very important part of how machine learning works, so it's important to talk about both of them. Discussion about algorithms and machine learning go hand in hand. They're the most important part of learning. In the world of computers, algorithms have been used for a long time to help us solve hard problems. They are a set of computer instructions for working with, changing, and interacting with data. An algorithm can be as simple as adding a column or as complicated as figuring out how to recognize anyone's face in a picture.

For an algorithm to work, it must be written as a programme that a computer can understand. Machine learning algorithms are usually written in either Java, Python, or R. Each of these languages has machine learning libraries that support a wide range of machine learning algorithms.

Active user communities for these languages share code and talk about ideas, problems, and ways to solve business problems. Machine learning algorithms are different from other algorithms. Most of the time, a programmer starts by typing in the algorithm. Machine learning turns the process around. With machine learning, the data itself creates the model. When you add more data to an algorithm, it gets harder to understand. As the machine learning algorithm gets more and more information, it can make more accurate algorithms.

It's a mix of science and art, to choose the right kind of machine learning algorithm. If you ask two data scientists to solve the same business problem, they might do it in different ways. But data scientists can figure out which machine learning algorithms work best if they know the different kinds. So, the most important step after getting the data in the right format is to choose the right machine learning algorithm.

As a result of our earlier discussion, we understood that choosing a right algorithm for machine learning is a process of trial and error. There is also a contradiction between certain aspects of the algorithms, such as:

- the amount of time spent in training;
- the amount of memory needed;
- the accuracy with which predictions are made on new data; and
- the level of transparency or interpretability (how easily you can understand the reasons an algorithm makes its predictions)

Let’s take a closer look at the most commonly used machine learning algorithms.

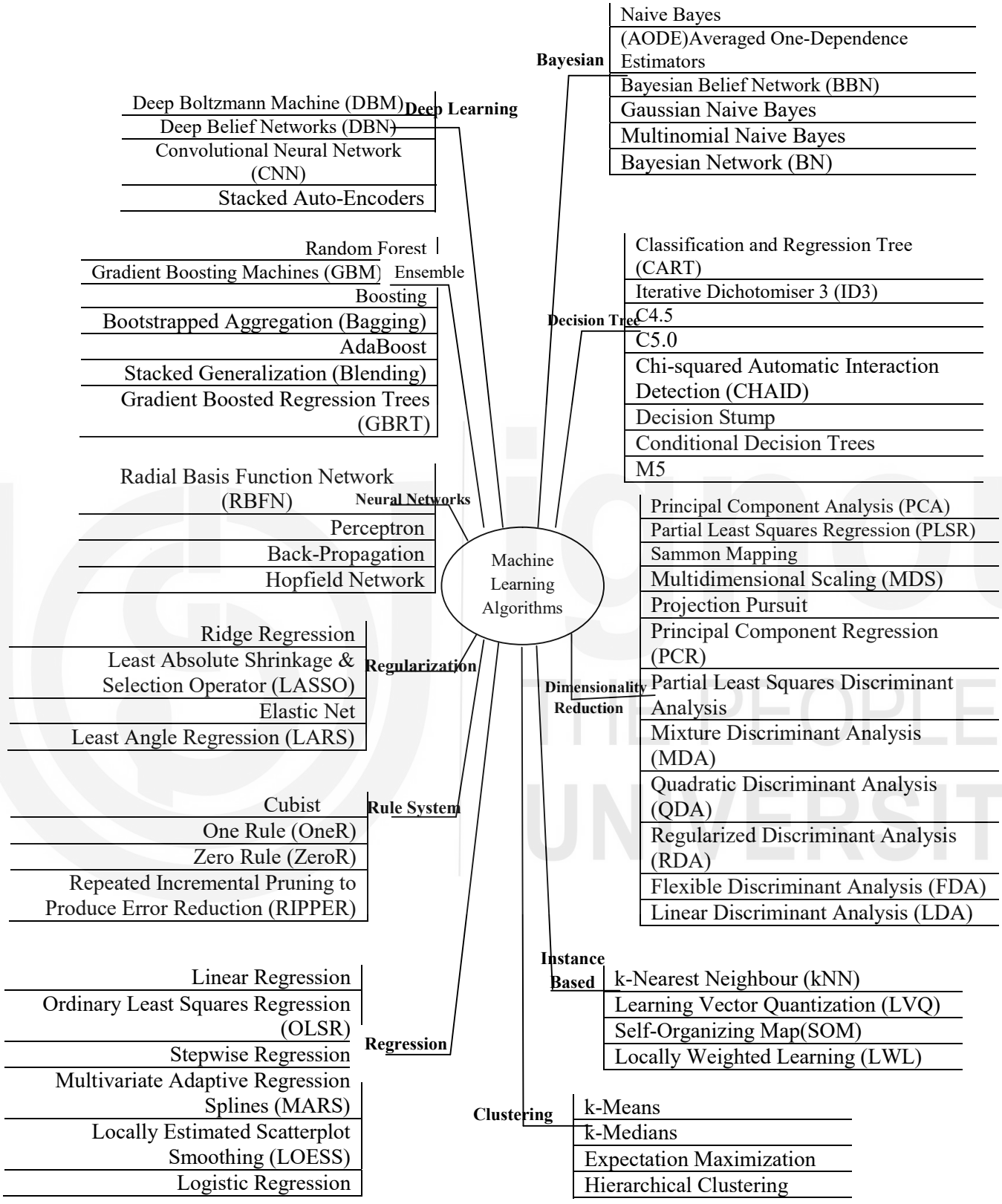


Figure – 4 : Types of machine learning algorithms

A brief discussion of the main types of machine learning algorithms, is given below : .

- **Bayesian:** Regardless of what the data shows, data scientists can use Bayesian algorithms to save their ideas about how models should look. Given how much attention is devoted to how the data shapes the model, you might ask why anyone would be interested in Bayesian algorithms. When you don't have much data to work with, Bayesian techniques come in handy.

  If you already knew something about a part of the model and could code that part directly, a Bayesian algorithm might make sense. Consider a medical imaging system that looks for signs of lung disease. These estimates can be incorporated into the model if a study published in a journal calculates the likelihood of various lung diseases based on a person's lifestyle.

- **Clustering :** Clustering is an easy-to-understand approach. Objects with comparable properties are combined (in a cluster). A cluster's contents are more similar than those of other clusters. Because the data are not labelled, clustering is a sort of unsupervised learning. Based on the parameters, the algorithm determines what each item is made of and assigns it to the appropriate group.

- **Decision tree :** Decision tree algorithms show what will happen when a choice is made by using a structure with branches. Decision trees can be used to show all the possible outcomes of a choice. A decision tree shows all the possible outcomes at each branch. The likelihood of the outcome is shown as a percentage for each node.

  Sometimes, online sales use decision trees. You might want to figure out who is most likely to use a 50% off coupon before sending it to them. Customers can be split into four groups:

  a) Customers who are likely to use the code if they get a personal message.
  b) Customers who will buy no matter what.
  c) Customers who will never buy.
  d) Customers who are likely to be upset if someone tries to reach out to them.

  If you send out a campaign, it's obvious that you don't want to send items to three of the groups since they will either ignore them or respond negatively. You'll get the best return on investment (ROI) if you go after the convenience.

  A decision tree will assist you in identifying these four client categories and organizing prospects and customers according to who will respond best to the marketing campaign.

- **Dimensionality reduction :** Dimensionality reduction allows systems to eliminate redundant data. These approaches remove data that is redundant, outliers, or otherwise useless. Sensor data and other IoT use cases can benefit from dimensionality reduction. The status of a sensor in an IoT system can be communicated using thousands of data points. Storing and analyzing "on" data is wasteful and wastes storage. Furthermore, minimizing redundant data increases machine

learning system performance. Finally, data visualization is also benefited from dimensionality reduction.

- **Instance based :** Instance-based algorithms are used to classify new data points based on training data. Because there's no training phase, these algorithms are called "lazy learners." Instead, instance-based algorithms compare new data to training data and classify it based on how similar it is. Data sets with random changes, irrelevant data, or missing values are not good for instance-based learning.

  They can be quite good at finding patterns.

  For example, instance learning is used in spatial and chemical structure analysis. There are many instance-based algorithms used in biology, pharmacology, chemistry, and engineering.

- **Neural networks and deep learning :** A neural network is an artificial intelligence system that attempts to solve problems in the same way that the human brain does. This is accomplished by the utilisation of many layers of interconnected units that acquire knowledge from data and infer linkages. In a neural network, the layers can be connected to one another in various ways. When referring to the process of learning that takes place within a neural network with multiple hidden layers, the term "deep learning" is frequently used. Models built with neural networks are able to adapt to new information and gain knowledge from it. Neural networks are frequently utilised in situations in which the data in question is not tagged or is not organised in a particular fashion. The field of computer vision is quickly becoming one of the most important applications for neural networks. Today, one can find applications for deep learning in a diverse range of contexts.

  The process of deep learning is utilised to assist self-driving autos in figuring out what is going on in their surroundings. Deep learning algorithms analyse the unstructured data that is being collected by the cameras as they capture pictures of the environment around them. This allows the system to make judgments in what is essentially real time. The apps that radiologists use to better analyse medical images also include deep learning as an integral part of their design.

- **Linear regression :** Regression algorithms are important in machine learning and are often used for statistical analysis. Regression algorithms help analysts figure out how data points are related.

  Regression algorithms can measure how strongly two variables in a set of data are linked to each other. Regression analysis can also be used to predict the values of data in the future based on their past values. But it's important to remember that regression analysis is based on the idea that correlation means cause. Regression analysis can lead to wrong conclusions if you don't understand the context of the data.

- **Regularization to avoid over-fitting** : The process of regularisation involves modifying models in such a way that they no longer fit too well into the data . Any model that is used for machine learning can benefit from using regularisation. For example, you can regularise a decision tree

model. Models that are excessively intricate and have a tendency to be overfit can become easier to grasp with the help of regularisation. A model that has been overfit to the available data will produce accurate predictions when additional data sets are added to it. When a model is developed for a particular set of data, but that model is unable to produce accurate predictions when applied to a more general set of data, this is an example of overfitting.

- **Rule-based machine learning :** Rule-based machine learning algorithms describe data with the help of rules about relationships. A rule-based system is different from a machine learning system, which builds a model that can be used on all the data. Rule-based systems are easy to understand in general: if X data is put in, do Y. A rule-based approach to machine learning, on the other hand, can get very complicated as systems get more complicated. For example, a system might have 100 rules that are already set. As the system gets more and more data and learns how to use it, it is likely that hundreds of rules will be broken. When making a rule-based approach, it's important to make sure it doesn't get so complicated that it stops being clear.

Think about how hard it would be to make an algorithm based on rules to apply the GST codes.

Check your progress - 2

Q3. Discuss the various phases of Machine Learning.

……
……

Q4 When should we use machine learning ?

……
……

Q5 Compare the concept of Classification, Regression and Clustering? List the algorithms in respective categories.

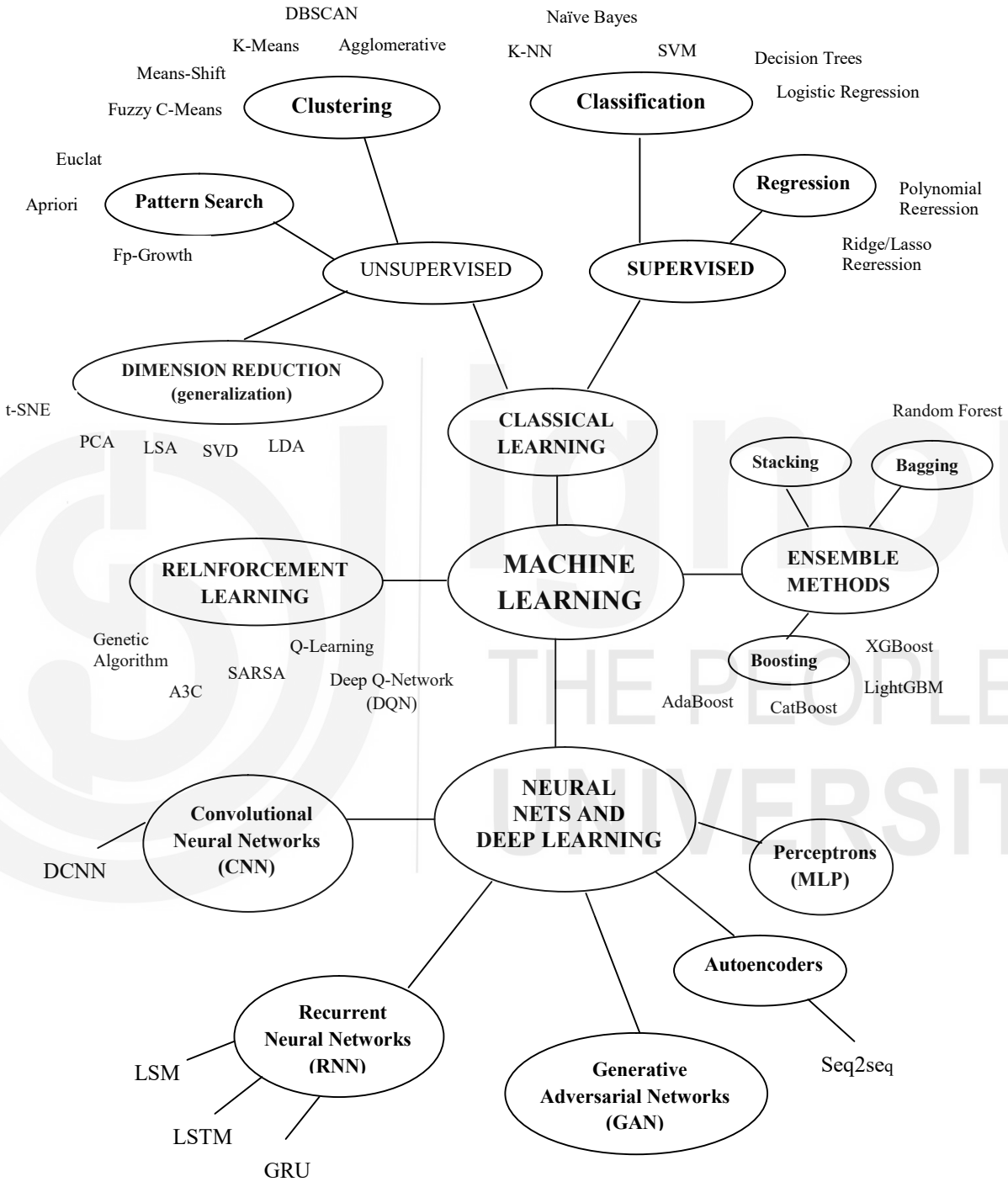
……
……

9.4 REINFORCEMENT LEARNING AND ALGORITHMS

According to what we observed in the previous section, learning can be broken down into three main categories: supervised, unsupervised, and semi-supervised. However, in addition to these two categories, there are also other types of learning, such as reinforcement learning (RL), deep learning (DL), adaptive learning, and so on.

The graph shown below, depicts the various branches and sub-branches of Machine learning, including the various algorithms involved in each sub-branch. Let’s understand them in brief, as the entire coverage

of the said Machine Learning techniques is out of the scope of this unit. We will begin our discussion with Reinforcement learning.



In Reinforcement Learning (RL), algorithms get a set of instructions and rules and then figure out how to handle a task by trying things out and seeing what works and what doesn't. As a way to help the AI find the best way to solve a problem, decisions are either rewarded or punished.

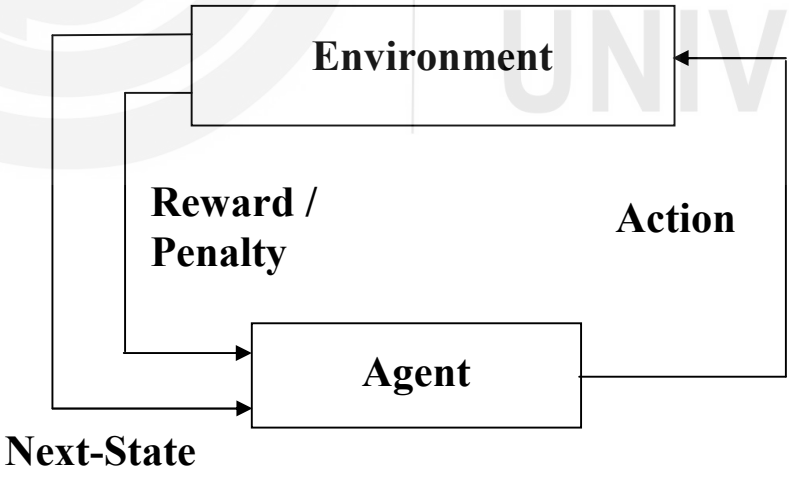
Machine learning models are taught through reinforcement learning to make a series of decisions. It is set up so that an Agent talks to an Environment.

Reinforcement Learning (RL) is a type of Machine Learning in which the agent gets a delayed re ward in the next time step to evaluate how well it did in the previous time step. It was mostly use d in games, like Atari and Mario, where it could do as well as or better than a person. Since Neur al Networks have been added to the algorithm, it has been able to do more complicated tasks.

In reinforcement learning, an AI system is put in a situation that is like a game (i.e. a simulation). The AI system tries until it finds a solution to the problem. Slowly but surely, the agent learns how to reach a goal in an uncertain, potentially complicated environment, but we can't expect the agent to slip upon the perfect solution by accident. This is where the interactions come into play, the Agent is provided with the State of the Environment which becomes the input/basis for the Agent to take Action. An Action first gives the Agent a Reward. (Note that rewards can be both positive and negative depending on the fitness function for the problem.) Based on this reward, the Policy (ML model) inside the Agent adapts and learns. Second, it affects the Environment and changes its State, which means the input for the next cycle changes.

This cycle will keep going until the best Agent is created. This cycle tries to imitate the way that organisms learn over the course of their lives. Most of the time, the Environment is reset after a certain number of cycles or if something goes wrong. Note that you can run more than one Agent at the same time to get to the solution faster, but each Agent runs on its own, independently.

Reinforcement Learning (RL) refers to a kind of Machine Learning method in which the agent receives a delayed reward in the next time step to evaluate its previous action. It was mostly used in games. Typically, a RL setup is composed of two components, an agent and an environment.



The following are the meanings of the different parts of reinforcement learning:

1. AGENT : The agent is the person who learns and makes decisions.

2. ENVIRONMENT : The agent's environment is where it learns and decides what to do.

3.ACTION : A group of things that the agent can do.

4. STATE : How the agent is doing in its environment.

5. REWARD : The environment gives the agent a reward for each action they choose. Usually a scalar value.

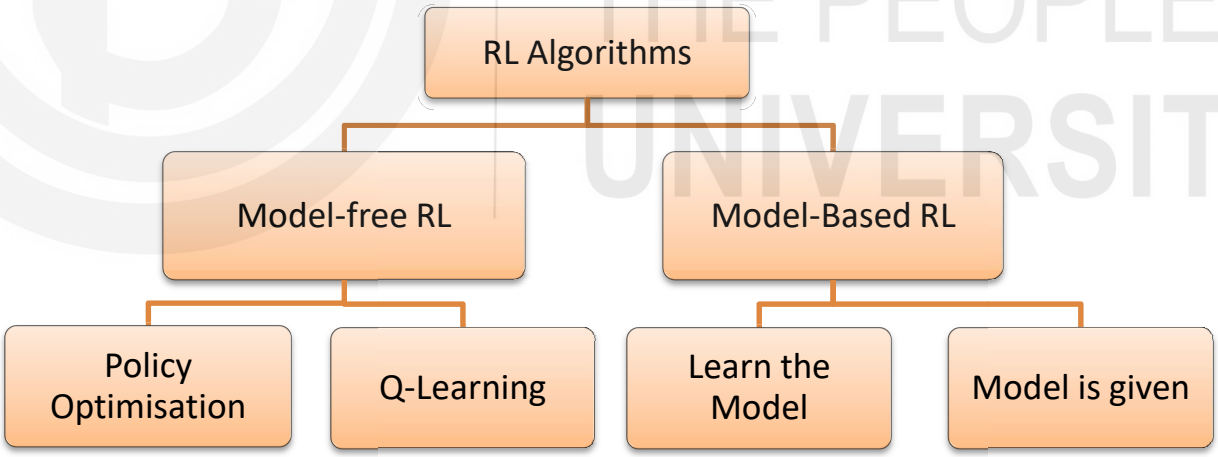
6. POLICY : Policy is the agent's way of deciding what to do (its control strategy), which is a mapping from situations to actions.

7. VALUE FUNCTION : A way to map states to real numbers, where the value of a state is the long-term reward that can be earned by starting in that state and following a certain policy.

8. FUNCTION APPROXIMATOR : is a term for the problem of figuring out what a function is by looking at training examples. Decision trees, neural networks, and nearest-neighbor methods are all examples of standard approximators.

9. MODEL : The agent's view of the environment, which maps state-action pairs to probability distributions over states. Note that not every agent that learns from its environment uses a model of its environment.

In spite of the fact that there is a large number of RL algorithms, it does not appear that there is a comparison that is exhaustive of each of them. It is quite challenging to determine which algorithms should be used for which type of activity. This section will attempt to provide an introduction to several well-known algorithms.



The algorithms for reinforcement learning may be broken down into two broad categories: model-free and model-based. In this section, we will analyse the key differences between these two types of reinforcement learning algorithms.

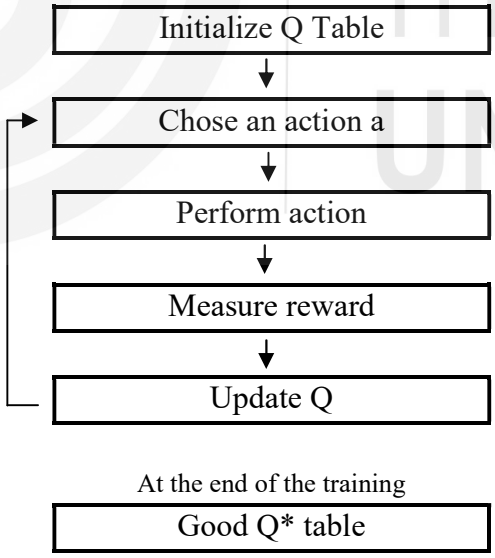
Model-Free Vs Model Based RL: The model is used to perform a simulation of the dynamic processes that take place in the environment. In other words, the model learns the transition

probability T(s1|(s0, a)) from the present state s0 and action a to the next state s1, and it does so by pairing the two states together. If the agent is able to successfully learn the transition probability, then the agent will be aware of how probable it is to reach a particular state given the present state and activity. On the other hand, as the state space and the action space grow, model-based algorithms become less practical.

On the other hand, model-free algorithms acquire new information through an iterative process of trial and error. As a consequence of this, it does not need any additional space in order to store every possible combination of states and actions.

Within the realm of Model-Free RL, policy optimization serves as a subclass, and it is comprised of two distinct sorts of policies. i.e. On-Policy Vs Off-Policy: The value is learned by an on-policy agent based on its current action "a" which is derived from the current policy, but the value is learned by an off-policy agent's counterpart based on the action "a*" which is received from another policy. This policy is referred to as the greedy policy in Q-learning.

The Q-learning or value-iteration methods are the next subcategory that is included in Model-Free RL. Q-learning is responsible for the acquisition of the action-value function. How advantageous would it be to perform a certain action at a certain state? In its most basic form, the action "a" receives a scalar value that is determined by the state "s". The algorithm is shown in the following chart, which does a good job of conveying its details.



Lets extend our discussion to some more Reinforcement Learning Algorithms i.e. DQN and SARSA

Deep Q Neural Network (DQN): It is Q-learning with Neural Networks . The motivation behind is simply related to big state space environments where defining a Q-table would be a very complex, challenging and time-consuming task. Instead of a Q-table Neural Networks approximate Q-values for each action based on the state.

State–action–reward–state–action (SARSA): SARSA algorithm is a slight variation of the popular Q-Learning algorithm. For a learning agent in any Reinforcement Learning algorithm it's policy can be of two types:-

- **On Policy:** In this, the learning agent learns the value function according to the current action derived from the policy currently being used.

- **Off Policy:** In this, the learning agent learns the value function according to the action derived from another policy.

The Q-Learning technique is considered an Off Policy technique that employs the greedy learning strategy in order to acquire knowledge of the Q-value. On the other hand, the SARSA approach is an On Policy and it makes advantage of the action that is being performed by the current policy in order to learn the Q-value.

Text mining, facial recognition, city planning, and targeted marketing are some of the applications which are actually the implementation of unsupervised learning algorithms. In a similar manner, the classification methods that fall under the supervised learning umbrella have applications in the areas of fraud detection, spam detection, diagnostics, picture classification, and score prediction. Similarly , reinforcement learning has a wide range of applications in a variety of fields, including the gaming industry, manufacturing, inventory management, and the financial sector, among many others..

Check Your Progress – 3

Q6 What is Reinforcement Learning ? List the components involved in it.

……
……

Q7 Briefly discuss the various algorithms of Reinforcement Learning.

……
……

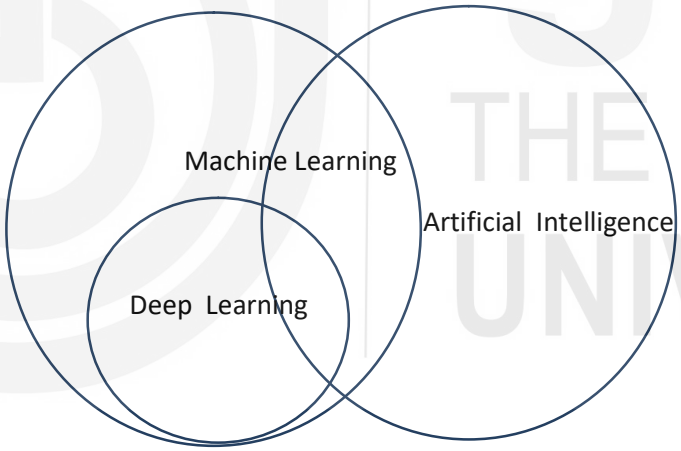
9.5 DEEP LEARNING AND ALGORITHMS

Deep learning is a type of machine learning that uses artificial neural networks and representation learning. It is also called deep structured learning or differential programming.Deep learning is a way for machines to learn through deep neural networks. It is used a lot to solve practical problems in fields like computer vision (image), natural language processing (text), and automated speech recognition (audio). Machine learning is often thought of as a tool with several algorithms. However, deep learning is actually just a subset of approaches that mostly use neural networks, which are a type of algorithm loosely based on the human brain.

A deep learning model learns to solve classification tasks directly from images, text, or sound. A neural network architecture is commonly used to implement deep learning. The number of layers in a network defines the depth of the network; the more layers, the deeper the network. Traditional neural networks have two or three layers, whereas deep neural networks include hundreds.

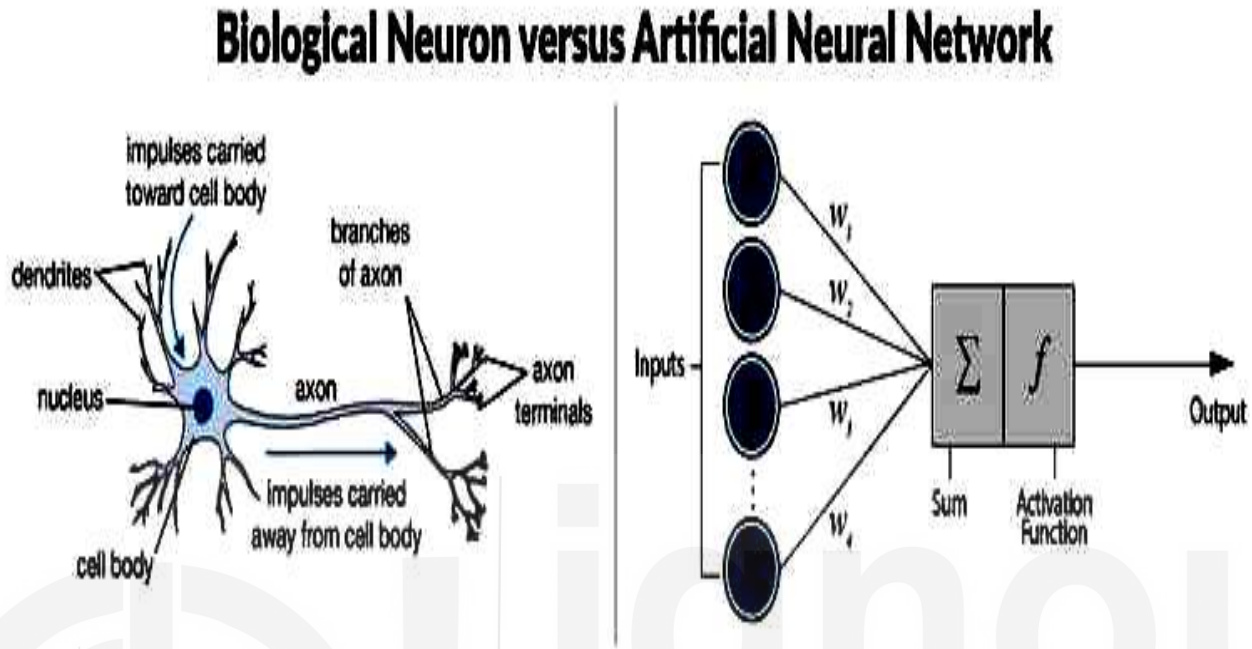
Deep learning is especially well-suited to identification applications such as face recognition, text translation, voice recognition, and advanced driver assistance systems, including, lane classification and traffic sign recognition.

Relation Between Machine learning , Deep Learning and Artificial Intelligence



As seen in the diagram above, machine learning (ML), deep learning (DL), and artificial intelligence (AI) are all related. Deep Learning is a collection of algorithms inspired by the human brain's workings in processing the data and creating patterns for use in decision making, which are expanding and improving or refining the idea of a single model architecture termed Artificial Neural Network (ANN). Later in this course, we shall go deeper into neural networks. However fr now, a quick overview of neural networks is provided below, followed by a discussion of the various Deep Learning algorithms, such as CNN, RNN, Auto Encoders, GAN, and others..

Neural Networks: Just like the human brain, Neural Networks consist of Neurons. Each Neuron takes in signals as input, multiplies them by weights, adds them together, and then applies a non-linear function. These neurons are arranged in layers and stacked close to each other.



Neural Networks have proven to be effective function approximators. We can presume that every behaviour and system can be represented mathematically at some point (sometimes an incredible complex one). If we can find that function, we will know everything there is to know about the system. However, locating the function can be difficult. As a result, we must use Neural Networks to estimate it.

A deep neural network is one that incorporates several nonlinear processing layers, makes use of simple pieces that work in parallel, and takes its cues from the biological nervous systems of living things. There is an input layer, numerous hidden layers, and an output layer that make up this structure. Each hidden layer takes as its input the information that was output by the layer that came before it and is connected to the other layers via nodes, also known as neurons.

To understand the basic deep neural networks we need to have brief understanding of various algorithms, the same are given below:

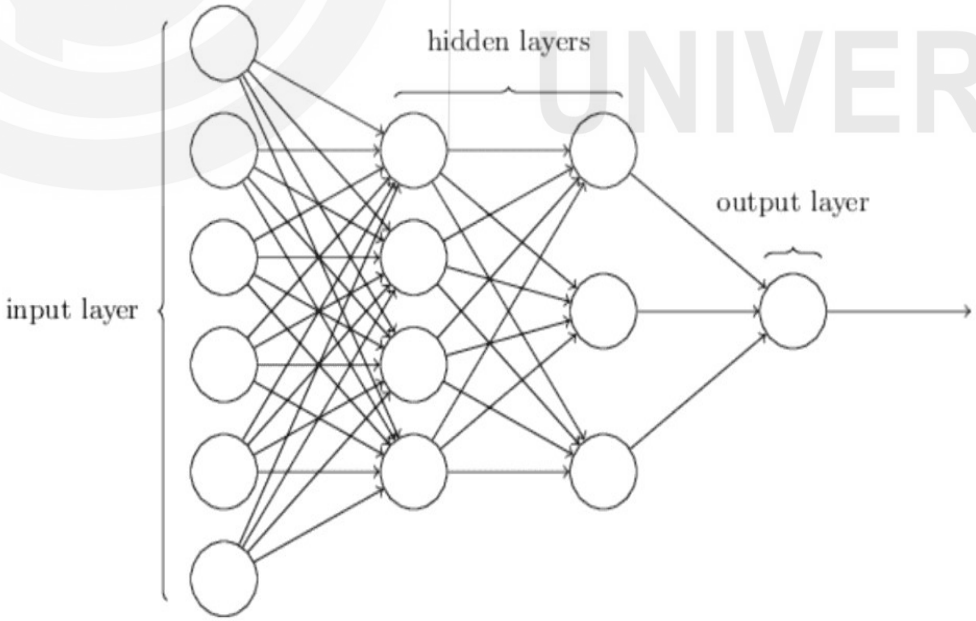
Back Propagation Neural Networks:Propagation in reverse Back propagation is an iterative process that allows neural networks to learn the desired function by utilising large quantities of data and learning from their previous mistakes. We feed the network data, and in return, it provides us with an output. We start by comparing the output to what we want using a loss function. Then, based on the gap that we find, we iteratively adjust the weights of the various variables. This non-linear optimization procedure is termed stochastic gradient descent, and it is used to make the necessary modification to the weights.

After some time, the network will improve its ability to provide the output to a very high standard. As a result, the training is complete. As a result, we are able to come close to approximating our function. In addition, if we give the network an input for which we do not know the corresponding output, it will provide us with an answer based on the approximated function.

To further understand, let's look at an example. Let's say we need to recognise pictures that contain a tree. Photos are input into the network, and the system produces results. We might evaluate the results in light of our current situation and make adjustments to the network accordingly.

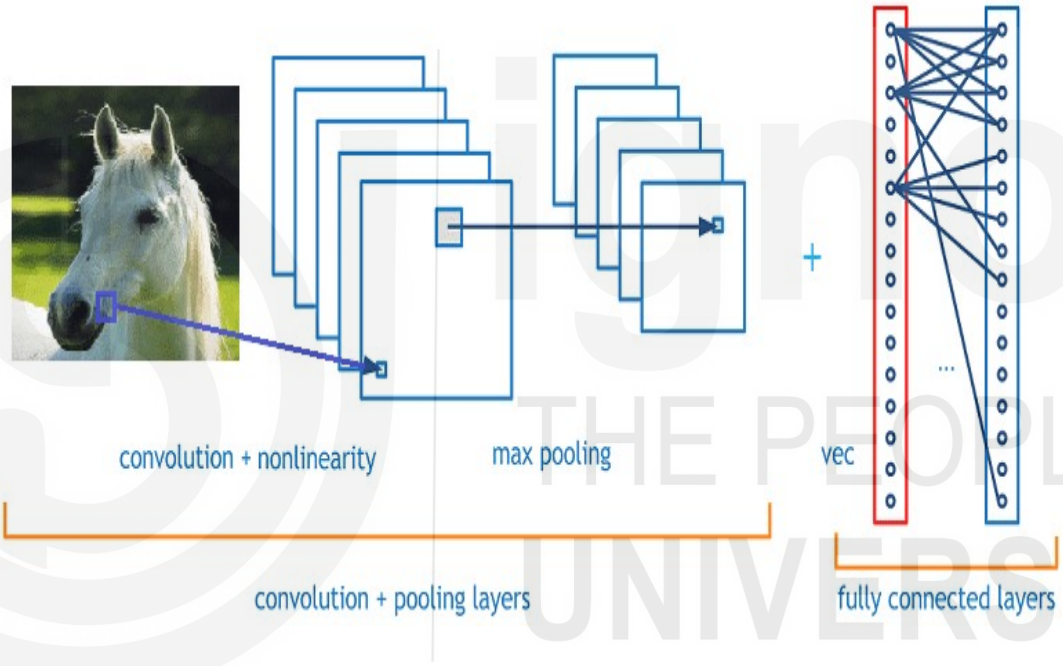
As more photographs are passed via the network, the number of errors that occur decreases. We can now feed it an unknown image, and it will tell us whether or not it has a tree. or not, that is astounding either way.

Feed-forward neural networks (FNN) : Typically, feed-forward neural networks, also known as FNN, are completely connected, this implies that each neuron in one layer is connected to each neuron in the layer next to it. A "Multilayer Perceptron" is the name given to the structure shown below and that is the topic of discussion here. A multilayer perceptron, has the ability to learn associations between the data that are not linear, in contrast to a single-layer perceptron, which can only learn patterns that can be separated in a linear manner. FNN are exceptionally well on tasks like classification and regression. Contrary to other machine learning algorithms, they don't converge so easily. The more data they have, the higher their accuracy.



Convolutional Neural Networks (CNN) The term "convolution" refers to the function that is utilised by convolutional neural networks (CNN). The idea that undelies them is that rather than linking each neuron with all of the ones that come after it, we just connect it with a select few of those that come after it (the receptive field). They strive to regularise feed-forward networks in order to avoid overfitting, which is when the model is unable to generalise its findings since it can only learn from the data it has already seen. Because of this, they are particularly skilled at determining how the data are related to one another spatially. As a result, computer vision is their primary application, which includes image classification, video identification, medical image analysis, and self-driving automobiles. These are the types of tasks where they achieve near-superhuman results.

Due to their adaptability, they are also ideal for merging with other types of models, such as Recurrent Networks and Auto-encoders. The recognition of sign languages is one such exa**mple.**



Face Recognition Based on Convolutional Neural Network

Recurrent Neural Networks (RNN) are utilised in time series forecasting because they are ideal for time-related data. They employ some type of feedback, in which the output is fed back into the input. You can think of it as a loop that passes data back to the network from the output to the input. As a result, they are able to recall previous data and use it to make predictions.

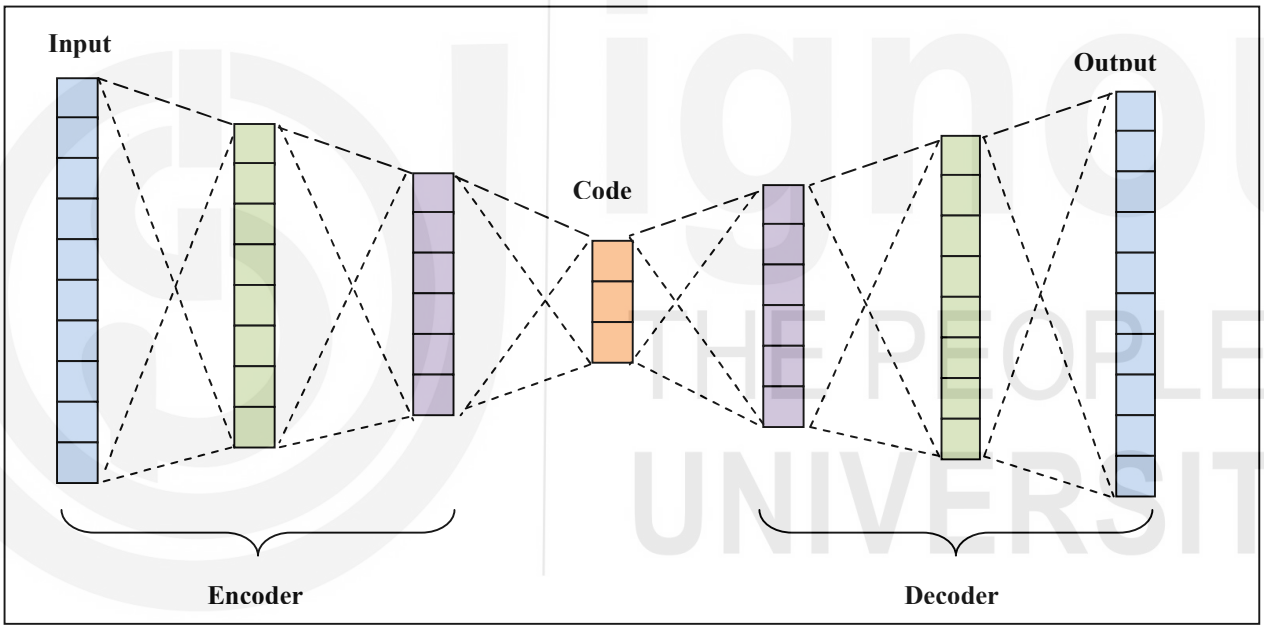
Researchers have transformed the original neuron into more complicated structures such as GRU units and LSTM Units to improve performance. Language translation, speech production, and text to speech synthesis have all employed LSTM units extensively in natural language processing.

Recursive Neural Networks : Another type of recurrent network is the recursive neural network, which is set up in a tree-like manner. As a result, they can simulate the hierarchical structures training dataset's.

They're frequently utilised in NLP applications like audio-to-text transcription and sentiment analysis because they're related to binary trees, contexts, and natural-language-based parsers. They are, however, typically much slower than Recurrent Networks.

Auto-Encoders (Auto Encoder Neural Networks) are a type of unsupervised technique that is used to reduce dimensionality and compress data. Their technique is to try and make the output equal to the input. They are attempting to recreate the data.

An encoder and a decoder are included in Auto-Encoders. The encoder receives the input and encodes it in a lower-dimensional latent space. Whereas, the decoder is used to decode that vector back to the original input.



Restricted Boltzmann Machines (RBM) are stochastic neural networks that can learn a probability distribution over their inputs and so have generative capabilities. They differ from other networks in that they only have input and hidden layers (no outputs).

They take the input and create a representation of it in the forward phase of the training. They rebuild the original input from the representation in the backward pass. (This is similar to autoencoders, but in a single network.)

Several RBMs are piled on top of each other to form a Deep Belief Network. They have the same appearance as Fully Connected layers, but they are trained differently.

Generative Adversarial Networks (GANs): Ian Goodfellow introduced Generative Adversarial Networks (GANs) in 2016, and they are built on a basic but elegant idea: You need to create data, such as photos. What exactly do you do?

You must construct two models. You teach the first one to make up fake data (generator) and the second one to tell the difference between actual and fake data (discriminator). And you turned them against one another.

The generator develops better and better at image production, as its ultimate purpose is to mislead the discriminator. As its purpose is to avoid being tricked, the discriminator improves its ability to identify fake from real images. As a result, we now have extremely realistic fake data from the discriminator.

Video games, astronomical imagery, interior design, and fashion are all examples of Generative Adversarial Networks at action. Essentially, you can utilise GANs if you have photos in your fields. Do you recall the movie Deep Fakes? That was all created by GANs.

Transformers are also very new, and they are mostly employed in language applications because recurrent networks are becoming obsolete. They are based on the concept of "attention," which instructs the network to focus on a certain data piece.

Instead of complicating LSTM units, you may use Attention mechanisms to assign varying weights to different regions of the input based on their importance. The attention mechanism is simply another weighted layer whose sole purpose is to change the weights such that some parts of the inputs are given greater weight than others.

In actuality, transformers are made up of stacked encoders (encoder layer), stacked decoders (decoder layer), including several attention layers (self- attentions and encoder-decoder attentions)

Graph Neural Networks: Deep Learning does not operate well with unstructured data in general. And there are many circumstances in which unstructured data is organised as a graph in the actual world. Consider social networks, chemical molecules, knowledge graphs, and location information.

Graph Neural Networks are used to model graph data. This implies they locate and convert the connections between nodes in a network into integers. As if it were an embedding. As a result, they can be utilized in any other machine learning model to perform tasks such as grouping, classifying, and so on.

Check Your Progress – 4

Q8 What is Deep Learning ?How Deep learning relates to AI & ML.

……
……

Q9 Briefly discuss the various algorithms of Reinforcement Learning.

……
……

9.6 ENSEMBLE METHODS

Ensemble learning is a general meta approach to machine learning that combines predictions from different models to improve predictive performance.

Although you can create an apparently infinite number of ensembles for any predictive modelling problem, the subject of ensemble learning is dominated by three methods. Bagging, stacking, and boosting. They are the three primary classes of ensemble learning methods, and it's essential to understand each one thoroughly.

- **Bagging Ensemble learning** is the process of fitting multiple decision trees to various samples of the same dataset and averaging the results.
- **Stacking Ensemble learning** is fitting multiple types of models to the same data and then using another model to learn how to combine the predictions in the best way possible.
- **Boosting Ensemble Learning** entails successively adding ensemble members that correct prior model predictions and produce a weighted average of the predictions.

Now let's discuss each of the learning method in some detail

(I) Bagging Ensemble learning Bagging ensemble learning involves fitting numerous decision trees to various samples of the same dataset, and then averaging the results of those tree fittings to provide a final prediction.

In most cases, this is accomplished by making use of a single machine learning method, which is nearly invariably an unpruned decision tree, and by training each model on a separate sample from the same training dataset. After then, straightforward statistical approaches such as voting or averaging are utilised in order to aggregate the predictions that were generated by each individual participant in the ensemble.

The manner in which each individual data sample is prepared to train members of the ensemble constitutes the most essential component of the technique. Every model receives its own unique, customised portion of the dataset to use for testing. Rows (examples) are selected at random from the dataset, and once selected, they are replaced.

When a row is selected, it is added back to the dataset that it was learned from, so that it can be selected once more from the same training dataset. This indicates that within a specific training dataset, a row of data may be selected 0 times, 1 times, or multiple times.

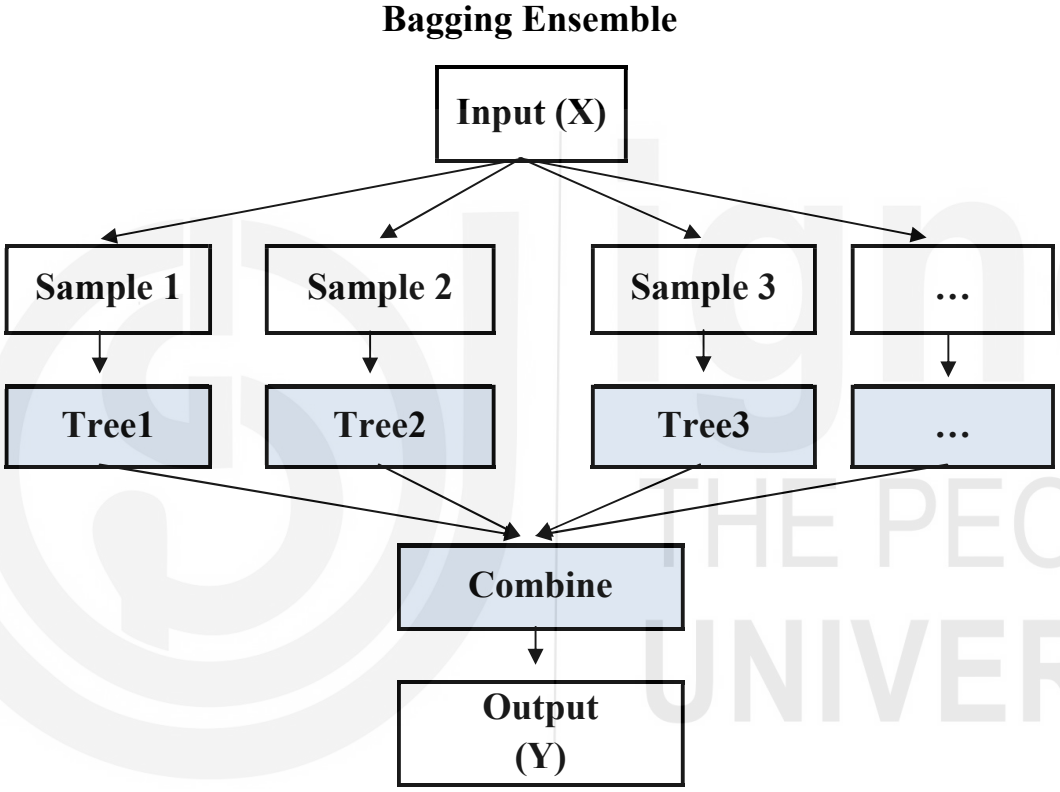
This type of sample is known as a bootstrap sample. In the field of statistics, this approach is a way for estimating the statistical value of a limited data sample. It is typically applied to somewhat limited data sets. You can get a better overall estimate of the desired quantity if you make a number of distinct bootstrap samples, estimate a statistical quantity, and then determine the average of the estimates. This is in comparison to the situation in which you would just estimate the quantity based on the dataset.

In the same way, several training datasets can be compiled, put to use in the process of estimating a predictive model, and then put to use in order to produce predictions. The majority of the time, it is preferable to take the average of the predictions made by all of the models rather than to fit a single model directly to the dataset used for training.

The following is a concise summary of the most important aspects of bagging:

- Take samples of the training dataset using bootstrapping.
- Unpruned decision trees fit on each sample.
- Voting or taking the average of all the predictions.

In a nutshell, bagging has an effect because it modifies the training data that is used to fit each individual member of the ensemble. This results in skillful but unique models.



Bagging ensemble learning

It is a comprehensive strategy that is simple to expand upon. For instance, additional alterations can be made to the dataset that was used for training, the method that was used to fit the training data can be modified, and the manner in which predictions are constructed can be altered.

Many popular ensemble algorithms are based on this approach, including:

- Bagged Decision Trees (canonical bagging)
- Random Forest
- Extra Trees

(II) Stacking Ensemble learning: Stacked Generalization, sometimes known as "stacking" due to its abbreviated form, is an ensemble strategy that searches for a diverse group of members by

varying the types of models that are fitted to the training data and utilising a model to aggregate predictions. It requires fitting of various kinds of models, applied to the same data, and then using another model to find out how to integrate the predictions in the best way possible. This process is known as model fitting.

There is a specific vocabulary for stacking. The individual models that comprise an ensemble are referred to as level-0 models, whereas the model that integrates all of the predictions is referred to as a level-1 model.

Although there are often only two levels of models applied, you are free to apply as many levels as you see fit. For instance, instead of a single level-1 model, we might have three or five level-1 models and a single level-2 model that integrates the forecasts of level-1 models to generate a prediction. This would allow us to make more accurate predictions.

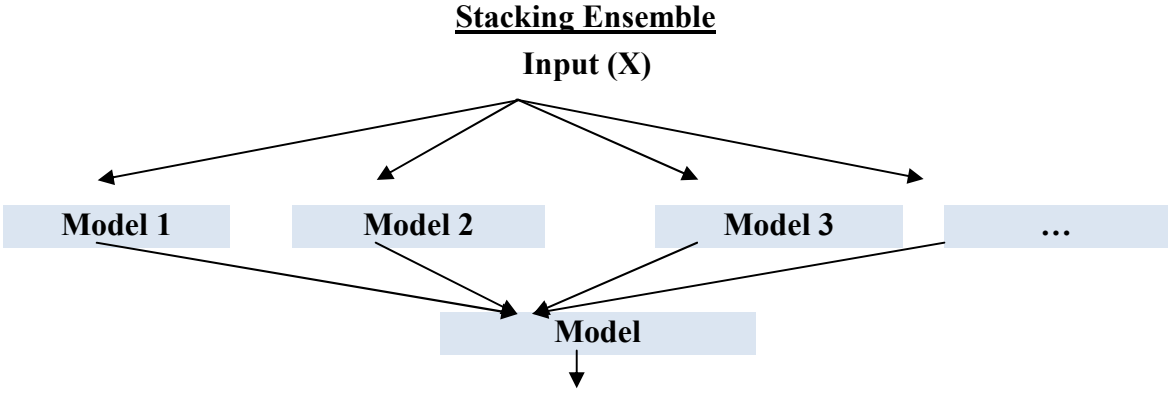
It is possible to integrate the predictions using any machine learning model, but the majority of users prefer linear models, such as linear regression for regression and logistic regression for binary classification. Because of this, it is more likely that the more difficult components of the model will be included in the lower-level ensemble member models, and that straightforward models will be used to learn how to apply the various predictions.

The key elements of stacking are summarized below:

- Unchanged training dataset.
- Separate machine learning algorithms for respective ensemble member.
- Machine learning model to learn how to combine predictions in the best way.

The diversity of the ensemble is a direct result of the many diverse machine learning models that serve as the ensemble's members.

As a consequence of this, it is recommended to make use of a variety of models that can be learnt or constructed in a wide variety of methods. Because of this, it is ensured that they will make separate assumptions, and as a consequence, it is less probable that their errors in prediction would be linked to one another.



Stacking Ensemble Learning

Many popular ensemble algorithms are based on this approach, including:

- Stacked Models (canonical stacking)
- Blending
- Super Ensemble

(III) Boosting Ensemble learning: Boosting is an ensemble strategy that aims to alter the training data so that it focuses on examples that earlier models that fit the training data got wrong. Boosting tries to do this by focusing on examples that prior models got wrong. In order for it to function, members are added to the ensemble one at a time, and when each new member is added, the predictions that were produced by the model that came before it are refined. A weighted average of the forecasts is what we get as a result.

The fact that boosting ensembles may correct errors in forecasts is the single most important advantage of using them. The models are calibrated and introduced to the ensemble one at a time, which means that the second model attempts to fix what the first model indicated, and so on and so forth.

The majority of the time, this is accomplished using weak learners, which are relatively straightforward decision trees that only make a single or a few decisions at a time. The forecasts of the weak learners are merged by simple voting or by average, but the importance of each learner's input is weighted according to how well they performed or how much they know. The objective is to create a "strong-learner" out of a number of "weak-learners," each of which was designed to accomplish a particular task.

Majority of the time the training dataset is left unchanged ; instead, the learning algorithm is adjusted to pay more or less attention to certain examples (rows of data) depending on how well they were predicted by ensemble members who were added earlier. For instance, a weight could be assigned to each row of data in order to demonstrate the level of focus that a learning algorithm must maintain on the model while it is doing so.

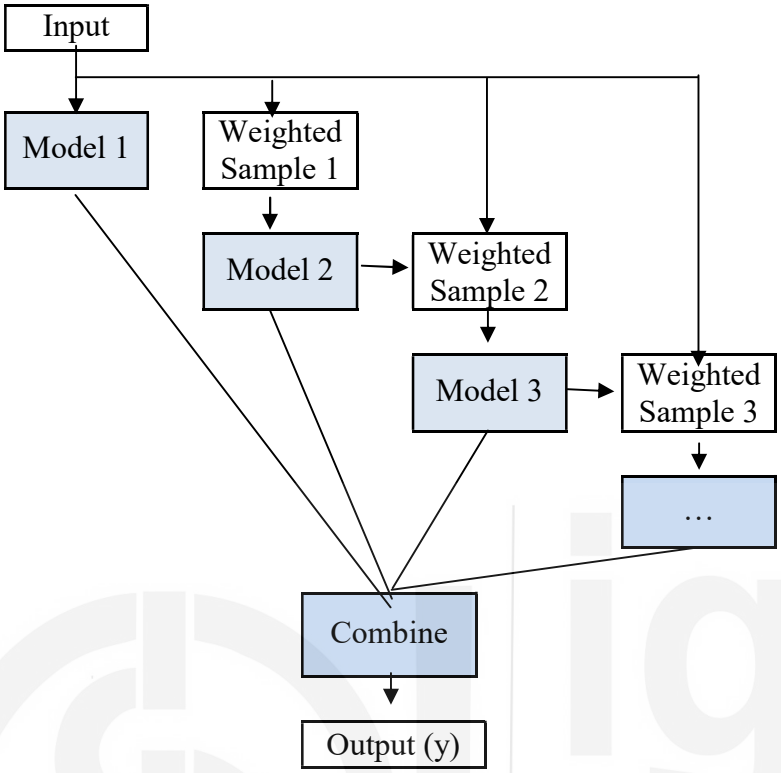
The key elements of boosting are summarized below:

- Give more weight to examples that are hard to guess when training.
- Add members of the ensemble one at a time to correct the predictions of earlier models.
- Use a weighted average of models to combine their predictions.

The idea of turning a group of weak learners into a group of strong learners was first thought of in theory, and many algorithms were tried but didn't work very well. Until the Adaptive Boosting (AdaBoost) algorithm was made, it wasn't clear that boosting was a good way to put together a group of methods.

Since AdaBoost, many boosting methods have been made, and some, like stochastic gradient boosting, may be among the best ways to use tabular (structured) data for classification and regression.

Boosting Ensemble



Boosting Ensemble Learning

To summarize, many popular ensemble algorithms are based on this approach, including:

- AdaBoost (canonical boosting)
- Gradient Boosting Machines
- Stochastic Gradient Boosting (XGBoost and similar)

This completes our tour of the standard ensemble learning techniques.

Check Your Progress – 5

Q10 What is Ensemble Learning ?

……
……

Q11 Briefly discuss the various Ensemble Methods.

……
……

9.7 SUMMARY

In this unit we discussed about the basic concepts of machine learning and also about the various Machine learning algorithms. The unit also covers the understanding of reinforcement learning and its related algorithms. There after we discussed the concept of Deep Learning and various techniques involved in Deep Learning. The unit finally discussed about the Ensemble Learning and its related methods. The unit

9.8 SOLUTIONS/ANSWERS

Check Your Progress - 1

Q1. How machine learning differs from Artificial intelligence ?

Solution: Refer to Section 9.2

Q2 Briefly discuss the major function or use of Machine learning algorithms **Solution:** Refer to Section 9.2

Check your progress - 2

Q3. Discuss the various phases of Machine Learning.

Solution: Refer to Section 9.3

Q4 When should we use machine learning ?

Solution: Refer to Section 9.3

Q5 Compare the concept of Classification, Regression and Clustering? List the algorithms in respective categories. 9.3

Solution: Refer to Section 9.3

Check Your Progress – 3

Q6 What is Reinforcement Learning ? List the various components involved in Reinforcement Learning.
Solution: Refer to Section 9.4
Q7 Briefly discuss the various algorithms of Reinforcement Learning.
Solution: Refer to Section 9.4

Check Your Progress – 4

Q8 What is Deep Learning ?How Deep learning relates to AI & ML.

Solution: Refer to Section 9.5

Q9 Briefly discuss the various algorithms of Reinforcement Learning.

Solution: Refer to Section 9.5

Check Your Progress – 5

Q10 What is Ensemble Learning ? 9.6

Solution: Refer to Section 9.6

Q11 Briefly discuss the various Ensemble Methods. 9.6

Solution: Refer to Section 9.6

9.9 FURTHER READINGS

- Prof. Ela Kumar, "Artificial Intelligence"Edition: First, Publisher: Dreamtech Press, (2020)ISBN: 9789389795134
- Machine learning an algorithm perspective, Stephen Marsland, 2nd Edition, CRC Press,, 2015.
- Machine Learning, Tom Mitchell, 1st Edition, McGraw- Hill, 1997.
- Machine Learning: The Art and Science of Algorithms that Make Sense of Data, Peter Flach, 1st Edition, Cambridge University Press, 2012.

UNIT 10 CLASSIFICATION

Structure

10.1 INTRODUCTION

What exactly does learning entail, anyway? What exactly is meant by "machine learning"? These are philosophical problems, but we won't be focusing too much on philosophy in this lesson; the whole focus will be on gaining a solid understanding of how things work in practise. In the subject of data mining, many of the ideas, such as classification and clustering, are being addressed, and so here in this Unit, we are going to once again investigate those concepts. Therefore, in order to achieve a better knowledge, the first step is to differentiate between the two fields of study known as data mining and machine learning.

It's possible that, at their core, data mining and machine learning are both about learning from data and improving one's decision-making. On the other hand, they approach things in a different manner. To get things started, let's start with the most important question, What exactly is the difference between Data Mining and Machine Learning?

What is data mining? Data mining is a subset of business analytics that involves exploring an existing huge dataset in order to discover previously unknown patterns, correlations, and anomalies that are present in the data. This process is referred to as "data exploration." It enables us to come up with wholly original ideas and perspectives.

What exactly is meant by "machine learning"? The field of artificial intelligence (AI) includes the subfield of machine learning . Machine learning involves computers performing analyses on large data sets, after which the computers "learn" patterns that will assist them in making predictions regarding additional data sets. It is not necessary for a person to interact with the computer for it to learn from the data; the initial programming and possibly some fine-tuning are all that are required.

It has come to our attention that there are a number of parallels between the two ideas, namely Data Mining and Machine Learning. These parallels include the following:

- Both are considered to be analytical processes;
- Both are effective at recognising patterns;
- Both focus on gaining knowledge from data in order to enhance decision-making capabilities;
- Both need a substantial quantity of information in order to be precise

Due to the mentioned similarities between the two, generally the people confuses the two concepts. So, to clearly demarcate the two concepts one should understand that What are the key differences between the two i.e. Data Mining and Machine Learning,?

The following are some of the most important distinctions between the two:

- Machine learning goes beyond what has happened in the past to make predictions about future events based on the pre-existing data. Data mining, on the other hand, consists of just looking for patterns that already exist in the data.

- At the beginning of the process of data mining, the 'rules' or patterns that will be used are unknown. In contrast, when it comes to machine learning, the computer is typically provided with some rules or variables to follow in order to comprehend the data and learn from it.

- The mining of data is a more manual process that is dependent on the involvement and choice-making of humans. With machine learning, on the other hand, once the foundational principles have been established, the process of information extraction, as well as "learning" and refining, is fully automated and does not require the participation of a human. To put it another way, the machine is able to improve its own level of intelligence.

- Finding patterns in an existing dataset (like a data warehouse) can be accomplished through the process of data mining. On the other hand, machine learning is trained on a data set referred to as a "training" data set, which teaches the computer how to make sense of data and then how to make predictions about fresh data sets.

The approaches to data mining problems are based on the type of information/ knowledge to be mined. We will emphasis on three different approaches: Classification, Clustering, and Association Rules.

The classification task puts data into groups or classes that have already been set up. The value of a user-specified goal attribute shows what type of thing a tuple is. Tuples are made up of one or more predicating attributes and one or more goal attributes. The task is to find some kind of relationship between the predicating attributes and the goal attribute, so that the information or knowledge found can be used to predict the class of new tuple (s).

The purpose of the clustering process is to create distinct classes from groups of tuples that share characteristic values. Clustering is the process of defining a mapping, using as input a database containing tuples and an integer value k, in such a way that the tuples are mapped to various clusters.

The idea entails increasing the degree of similarity within a class while decreasing the degree of similarity between classes. There is not an objective attribute in the clustering process. Clustering, on the other hand, is an example of an unsupervised classification, in contrast to classification, which is supervised by the aim attribute.

The goal of association rule mining is to find interesting connections between elements in a data set. Its initial use was for "market basket data." The rule is written as X➔Y, where X and Y are two sets of objects that do not intersect. Support and confidence are the two metrics for any rule. The aim is to identify, using rules with support and confidence above, minimum support and minimum confidence given the user-specified minimum support and minimum confidence.

The distance measure determines the distance between items or their dissimilarity. The following are the measures used in this unit:

$$\text{Euclidean distance: } dis(ti,tj) = \sqrt{\sum_{h=1}^{k} (t_{ih} - t_{jh})^2}$$

$$\text{Manhattan distance: } dis(ti,tj) = \sum_{h=1}^{k} |(t_{ih} - t_{jh})|$$

where ti and tj are tuples and h are the different attributes which can take values from 1 to k

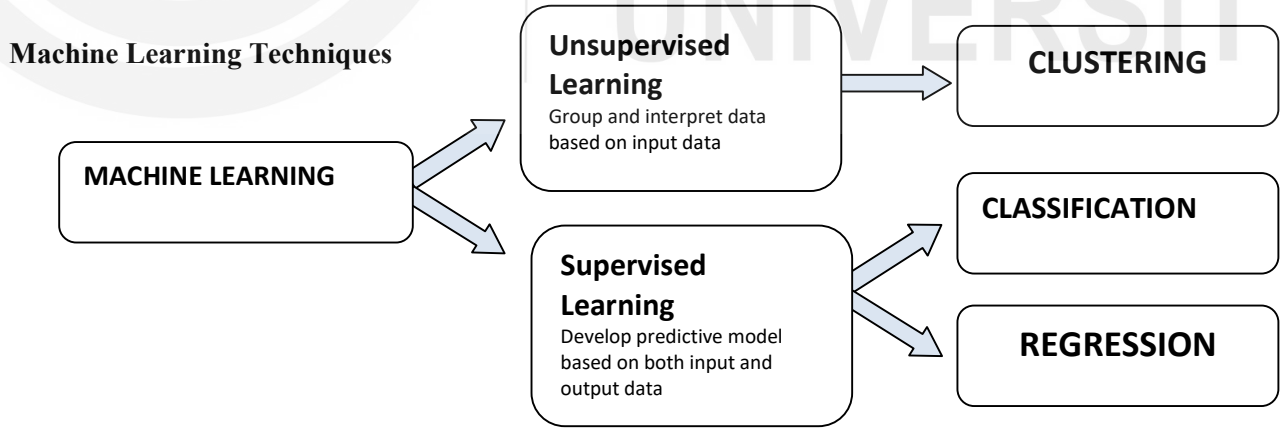
There are some clear differences between the two, though. But as businesses try to get better at predicting the future, machine learning and data mining may merge more in the future. For example, more businesses may want to use machine learning algorithms to improve their data mining analytics.

Machine learning algorithms use computational methods to "learn" information directly from data, without using an equation as a model. As more examples are available for learning, the algorithms get better and better at what they do.

Machine learning algorithms look for patterns in data that occur naturally. This gives you more information and helps you make better decisions and forecasts. They are used every day to make important decisions in diagnosing medical conditions, trading stocks, predicting energy load, and more. Machine learning is used by media sites to sort through millions of options and suggest songs or movies. It helps retailers figure out what their customers buy and how they buy it. With the rise of "big data," machine learning has become very important for solving problems in areas like:

• Computational finance, for applications such as credit scoring and algorithmic trading
• Face identification, motion detection, and object detection can all be accomplished through image processing and computer vision.
• Tumor detection, drug development, and DNA sequencing can all be accomplished through computational biology.
• Production of energy, for the sake of pricing and load forecasting • Automotive, aerospace, and manufacturing, for the purpose of predictive maintenance • Processing of natural languages

In general, Classical Machine Learning Algorithms can be put into two groups: Supervised Learning Algorithms, which use data that has been labelled, and Un-Supervised Learning Algorithms, which use data that has not been labelled and are used for Clustering. We will talk more about Clustering in Unit 15, which is part of Block 4 of this course.



In this unit we will be discussing about the Supervised Learning Algorithms, which are mainly used for the classification purpose.

10.2 OBJECTIVES

After completing this unit you should be able to :

- Understand Supervised Learning
- Understand Un-Supervised Learning
- Understanding various Classification Algorithms

10.3 UNDERSTANDING OF SUPERVISED LEARNING

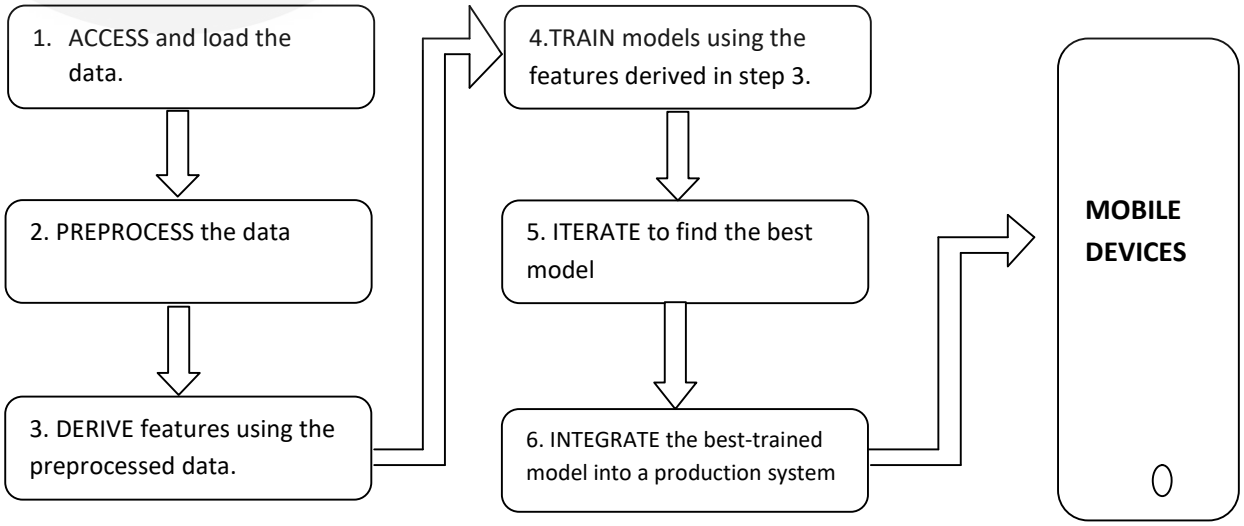
To use machine learning techniques effectively, you need to know how they work. You can't just use them without knowing how they work and expect to get good results. Different techniques work for different kinds of problems, but it's not always clear which techniques will work in a given situation. You need to know something about the different kinds of solutions.

Every workflow for machine learning starts with the following three questions:

- What kind of data do you have available to work with?
- What kinds of realisations are you hoping to arrive at as a result of it?
- In what ways and contexts will those realisations be utilised?

Your responses to these questions will assist you in determining whether supervised or unsupervised learning is best for you.

Workflow at a Glance



In an interesting way, supervised machine learning is like how humans and animals learn "concepts" or "categories." This is defined as "the search for and listing of attributes that can be used to tell exemplars from non-exemplars of different categories."

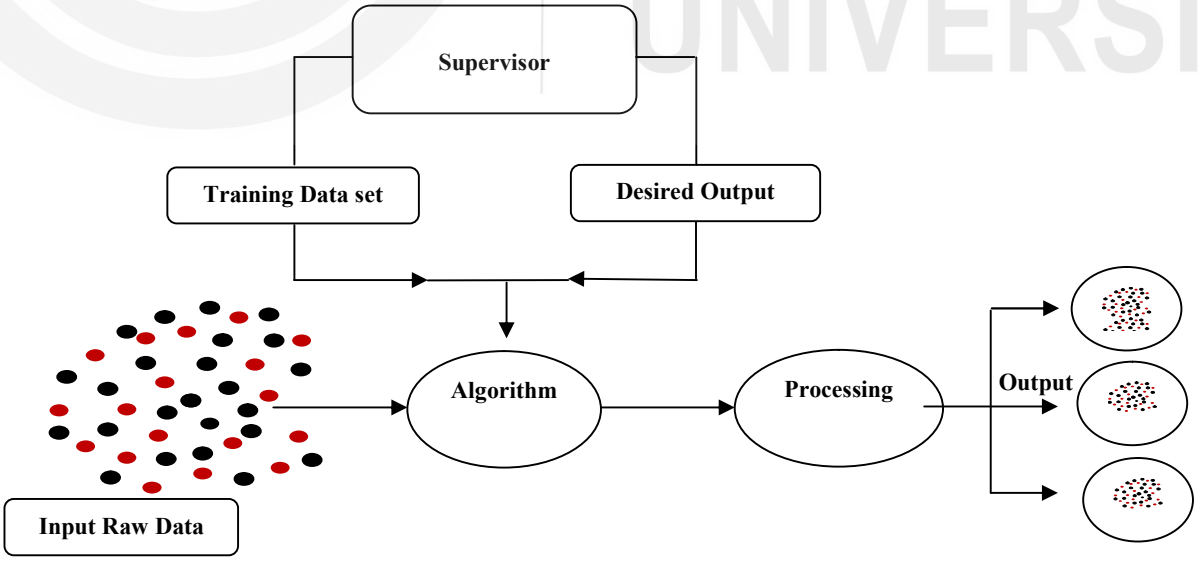
Technically, supervised learning means learning a function that gives an output for a given input based on a set of input-output pairs that have already been defined. It does this with the help of something called "training data," which is a set of examples for training.

In supervised learning, the data used for training is labelled. For example, every shoe is labelled as a shoe, and the same goes for every pair of socks. This way, the system knows the labels, and if it sees a new type of shoes, it will recognise them as "shoes" even if it wasn't told to do so.

In the example above, the picture of shoes and the word "shoes" are both inputs, and the word "shoes" is the output. After learning from hundreds or thousands of different pictures of shoes and the word "shoes" as well as the word "socks," our system will know what to do when given only a new picture of shoes (name: shoes).

supervised ML is often represented by the function y = f(x), where x is the input data and y is the output variable, which is a function of x that needs to be predicted. In training data, the example pair is usually made up of an input, which is usually a vector, and an output (a collection of features determining a sample). The output value we want, which we call a "supervisory signal" and whose meaning is clear from the name.

In fact, the goal of supervised machine learning is to build a model that can make predictions based on evidence even when there is uncertainty. A supervised learning algorithm uses a known set of input data and known responses to the data (output) to train a model to make reasonable predictions about the response to new data.



Example: Predicting heart attacks with the help of supervised learning: Let's say doctors want to know if someone will have a heart attack in the next year. They have information about the age,

weight, height, and blood pressure of past patients. They know if the patients who were there before had heart attacks within a year. So the problem is making a model out of the existing data that can tell if a new person will have a heart attack in the next year.

Following Steps are Involved in Supervised Learning, and they are self explanatory:

1. Determine the Type of Training Examples
2. Prepare/Gather the Training Data
3. Determine Relation Between Input Feature & Representing Learned Function
4. Select a Learning Algorithm
5. Run the Selected Algorithm on Training Data
6. Evaluate the Accuracy of the Learned Function Using Values from Test Set

There are some common issues which are generally faced when one applies the Supervised Learning, and they are listed below:

(i) Training and classifying require a lot of computer time, especially when big data is involved.

(ii) Overfitting: The model may learn so much from the noise in the data that instead of seeing it as a mistake, it can be seen as a learning concept.

(iii) A key difference between supervised and unsupervised learning is that if an input doesn't fit into any class, the model will add it to one of the existing ones instead of making a new one.

Lets discuss some of the Practical Applications of Supervised Machine Learning. For beginners at least, probably knowing ‘what does supervised learning achieve’ becomes equally or more important than simply knowing ‘what is supervised learning’.

A very large number of practical applications of the method can be outlined, but the following are some of the common ones

a) Detection of spam
b) Detection of fraudulent banking or other activities
c) Medical Diagnosis
d) Image recognition
e) Predictive maintenance

With increasing applications each day in all the fields, machine learning knowledge is an essential skill.

☞ Check Your Progress 1

1. Compare between Supervised and Un-Supervised Learning.

   ……

   ……

2. List the Steps Involved in Supervised Learning

   ……

   ……

3. What are the Common Issues Faced While Using Supervised Learning

   ……

   ……

10.4 INTRODUCTION TO CLASSIFICATION

Every supervised learning approach can be classified as either a regression or a classification method, depending on the nature of the data being analysed. The creation of predictive models is possible through supervised learning by utilising classification and regression methods. This can be performed by using these methods.

• **Classification techniques** : Classification methods make predictions about specific outcomes, such as whether an email is legitimate or spam or whether a tumour is malignant or benign. Classification models classify incoming data into categories. Applications such as medical imaging, speech recognition, and credit scoring are typical examples.

• **Regression techniques** : The techniques of regression can accurately forecast continuous reactions, such as shifts in temperature or variations in the amount of power required. Examples of typical applications are as follows: A few examples of applications are the predicting of stock prices, the recognition of handwriting, the forecasting of power load, and acoustic signal processing.

Note: *It's important to know whether a problem is a classification problem or a regression problem.*

> *• Can your information be tagged or put into groups? Use classification algorithms if your data can be put into clear groups or classes.*
>
> *• Are you working with a set of data? Use regression techniques if your answer is a real number, like TEMP. or the time until a piece of equipment fails to work.*

Before moving ahead lets understand some of the key terms, which will be frequently occurring in this course, they are listed below :

- **Classification** The process of organizing data into a predetermined number of categories is referred to as classification. Finding out which category or class a new collection of data falls under is the primary objective of a classification problem, which can be stated as follows, Data that is structured as well as data that is not structured can be utilized for classification:

  Structured data (data that is in a fixed field in a file or record is called "structured data"). A relational database (RDBMS) is where most structured data is kept.

  Unstructured data (unstructured data may have a natural structure, but it isn't set up in a way that can be predicted). There is no data model, and the data is stored in the format in which it was created. Rich media, text, social media activity, surveillance images, and so on, are all types of unstructured data.Following are some of the terminologies frequently encountered in machine learning – classification:

- **Classifier:** A classifier is an algorithm that puts the data you give it into a certain category. A classifier is an algorithm that does classification on a dataset.
- **Classification model:** A classification model looks at the output values to try to figure out what the values used for training mean. It will guess the class labels or categories of the new data.
- **Feature:** property of any object (real or virtual) that can be measured on its own is called a feature.
- **Classification predictive modeling** involves assigning a class label to input examples.

This section covers the following types of classification:

- **Binary classification** is a task for which there are only two possible outcomes. It means predicting which of the two classes will be correct. For example, dividing people into male and female.

Some popular algorithms that can be used to divide things into two groups are:

- Logistic Regression.
- k-Nearest Neighbors.
- Decision Trees.
- Support Vector Machine.
- Naive Bayes.

Major application areas of Binary Classification:

- Detection of Email spam.
- Prediction of Churn.
- Prediction of Purchase or Conversion(buy or not).

- **Multi-class classification.:**Multi-class classification means putting things into more than two groups. In multiclass classification, there is only one target label for each sample. This is done by predicting which of more than two classes the sample belongs to. An

animal, for instance, can either be a cat or a dog, but not both. Face classification, plant species classification, and optical character recognition are some of the examples.
Popular algorithms that can be used for multi-class classification include:

- k-Nearest Neighbors.
- Decision Trees.
- Naive Bayes.
- Random Forest.
- Gradient Boosting.

Binary classification algorithms can be changed to work for problems with more than two classes. This is done by fitting multiple binary classification models for each class vs. all other classes (called "one-vs-rest") or one model for each pair of classes (called one-vs-one).

- **One versus the Rest**: Fit one binary classification model for each class versus all of the other classes in the dataset.
- **One-versus-one**: Fit one binary classification model for each pair of classes using the one-on-one comparison method.

Binary classification techniques such as logistic regression and support vector machine are two examples of those that are capable of using these strategies for multi-class classification.

- **Multi-label classification:** Multi-label classification, also known as more than one class classification, is a classification task in which each sample is mapped to a collection of target labels. This classification task involves making predictions about one or more classes; say for example, a news story can be about Games, People, and a Location all together at the same time.

  Note : Classification algorithms used for binary or multi-class classification cannot be used directly for multi-label classification.

Specialized versions of standard classification algorithms can be used, so-called multi-label versions of the algorithms, including:

- Multi-label Decision Trees
- Multi-label Random Forests
- Multi-label Gradient Boosting

Another approach is to use a separate classification algorithm to predict the labels for each class.

• **An imbalanced classification** is a task in which the number of examples in each class is not the same. Examples includes Fraud detection, Outlier detection, Medical diagnostic tests.

These problems are modelled as two-way classification tasks, but you may need to use specialised methods to solve them.

The different types of classifications discussed above, have to deal with different type of learners, and **Learners in Classification Problems are categorized into following two types :**

1. **Lazy Learners:** Lazy Learner will first store the training dataset, and then it will wait until it is given the test dataset. In the case of the Lazy learner, classification is done based on the information in the training dataset that is most relevant to the question at hand. Less time is needed for training, but more time is needed for making predictions. K-NN algorithm and Case-based reasoning are two examples.
2. **Eager Learners:** Eager Learners use a training dataset to make a classification model before they get a test dataset. Eager learners, on the other hand, spend less time on training and more time on making predictions. Decision Trees, Naive Bayes, and ANN are some examples.

Examples of eager learners include the classification techniques of Bayesian classification, decision tree induction, rule-based classification, classification by back-propagation, support vector machines, and classification based on association rule mining. Eager learners, when presented with a collection of training pairs, will construct a generalisation model (also known as a classification model) before being presented with new tuples, also known as test tuples, to classify. One way to think about the learnt model is as one that is prepared and eager to categorise tuples that have not been seen before.

Imagine, on the other hand, in the lazy learner approach the learner is required to wait until the final moment, to develop a model for classification of the given test tuple, i.e. in the lazy approach the learner just stores the training tuple, given for classification. It does not do generalization until it is given a test tuple, after receiving the test tuple, it classifies the tuple based on how similar it is to the training tuples that it has previously stored. Lazy learning methods, on the other hand, do less work when a training pair is shown but more work when classifying or making a prediction. Because lazy learners keep the training tuples, which are also called "instances." So, they are also called "instance-based learners," even though this is how most people learn.

When classifying or making a prediction, lazy learners can take a lot of processing power. They need efficient ways to store information and can be done well on parallel hardware. They don't explain or show much about how the data is put together. Lazy learners, on the other hand, tend to be in favour of incremental learning. They can make models of complex decision spaces with hyper-polygonal shapes that other learning algorithms may not be able to do as well (such as hyper-rectangular shapes modeled by decision trees). The k-nearest neighbour classifier and the case-based reasoning classifier are both types of lazy learners.

☞ **Check Your Progress 2**

4. Compare between Multi Class and Multi Label Classification

……

……

5. Compare between structured and unstructured data

……

……

6. Compare between Lazy learners and Eager Learners algorithms for machine learning.

……

……

10.5 CLASSIFICATION ALGORITHMS

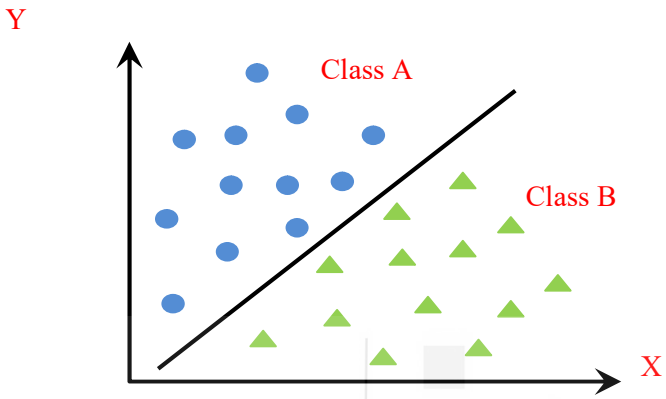
The Classification algorithm is a type of Supervised Learning that uses the training data to figure out the category of new observations. This method is used to figure out what kind of thing a new observation is. Classification is the process by which a computer programme learns from a set of data or observations and then sorts new observations into different classes or groups. "Yes" or "No," "0" or "1," "Spam" or "Not Spam," "Cat or Dog," and so on are all good examples. Classes are the same thing that have different names, like categories, objectives, and labels.

In classification, the output variable is not a value but a category, such as "Green or Blue," "Fruit or Animal," etc. This is different from regression, where the output variable is a value. Since the classification method is a supervised learning method, it needs data that has been labelled in order to work. This means that the implementation of the algorithm includes both the input and the output that go with it.

As the name suggests, classification algorithms do the job of predicting a label or putting a variable into a category (categorization). For example, classifying something as "socks" or "shoes" from our last example. Classification Predictive Algorithm is used every day in the spam detector in emails. It looks for features that help it decide if an email is spam or not spam.

The primary objective of the Classification algorithm is to determine the category of the dataset that is being provided, and these algorithms are primarily utilised to forecast the output for the data that is categorical in nature. A discrete output function, denoted by y, is mapped to an input variable, denoted by x, in a classification process. Therefore, y = function (x), where y denotes the categorical output. The best example of an ML classification algorithm is **Email Spam Detector**.

The diagram that follows can be used to have a better understanding of classification methods. There are two classes, Class A and Class B, depicted in the graphic that may be found below. These classes share characteristics that distinguish them from other classes but also distinguish them from one another.



The question arises as a result of the existence of a variety of algorithms under both supervised and unsupervised learning. How Should One Choose Which Algorithm to Employ? The task of selecting the appropriate machine learning algorithm can appear to be insurmountable because there are dozens of supervised and unsupervised machine learning algorithms, and each takes a unique approach to the learning process. There is no single approach that is superior or universally applicable. Finding the appropriate algorithm requires some amount of trial and error; even highly experienced data scientists are unable to determine whether or not an algorithm will work without first putting it to the test themselves. However, the choice of algorithm also depends on the quantity and nature of the data being worked with, as well as the insights that are desired from the data and the applications to which those insights will be put.

- **Choose supervised learning** if you need to train a model to make a prediction, for instance, the future value of a continuous variable, such as TEMP. or a stock price; use regression techniques and use classification techniques in situations such as identifying makes of cars from webcam video footage or identifying spams from emails; etc.

- **Choose unsupervised learning** if you need to investigate your data and want to train a model to find a decent internal representation, such as by dividing the data into clusters. This type of learning allows for more freedom in exploring and representing the data.

Note : The algorithm for Supervised Machine Learning can be broken down into two basic categories: regression algorithms and classification algorithms. We have been able to forecast the output for continuous values using the Regression methods; but, in order to predict the categorical values, we will need to use the Classification algorithms.

Let's take a closer look at the most commonly used algorithms for supervised machine learning.

Classification algorithms can be further divided into the mainly two categories, *Linear Models and Non Linear Models*, which includes various algorithms under them, the same are listed below :

- **Linear Models** : Involves Logistic Regression, Support Vector Machines
- **Non-linear Models :** Involves K-Nearest Neighbours, Kernel SVM, Naïve Bayes, Decision Tree Classification, Random Forest Classification

In order to build a classification model following steps are to be followed :

1. Start the classifier that will be utilised from scratch.
2. Train the classifier: In order to fit the model (training), all classifiers in scikit-learn use a function called fit(X, y) to do so. This method takes as input the train data X and the train label y.
3. Predict the target: Given an observation X that is not labelled, the predict(X) function returns the label that was predicted for the observation.
4. Conduct an analysis of the classification model.

EVALUATING A CLASSIFICATION MODEL: Now that we have a classification model, let's learn how to evaluate it. After we have finished developing our model, we need to assess how well it works, regardless of whether it is a regression or classification model. The following are some of the methods that can be used to evaluate a classification model:

1. Log Loss or Cross-Entropy Loss:

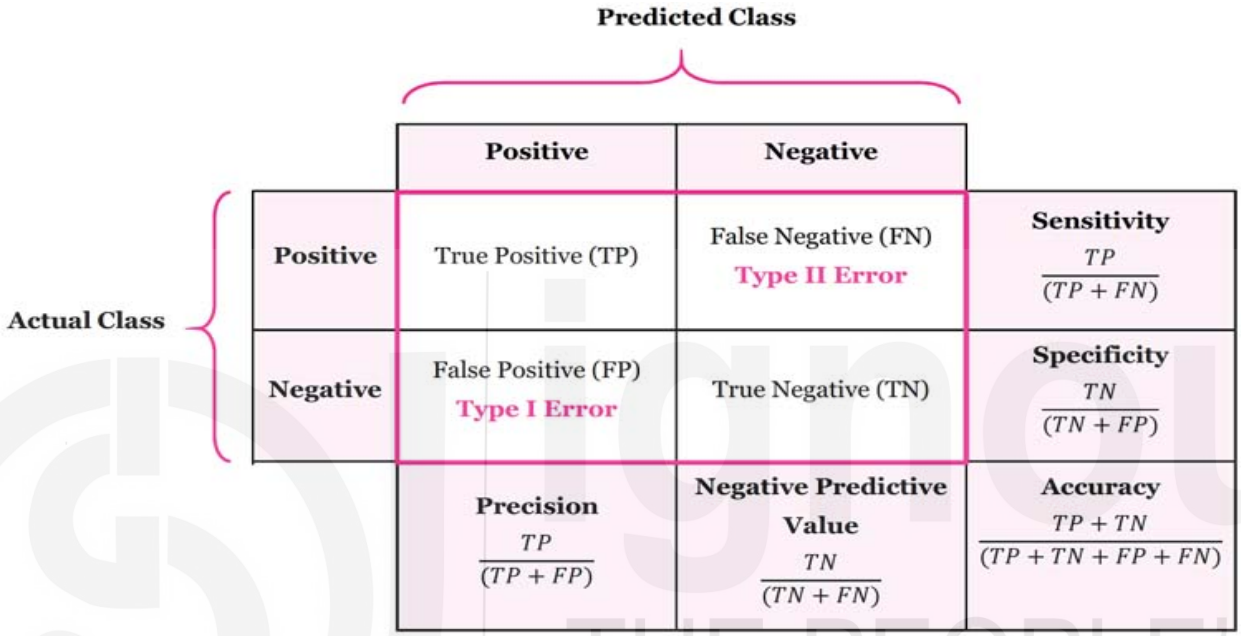
- It's used to measure how well a classifier works, which gives a probability value between 0 and 1.
- The value of log loss should be close to 0 for a good binary classification model.
- If the predicted value is different from the actual value, the value of log loss goes up.
- The model is more accurate when the log loss is lower.
- For binary classification, the cross-entropy is calculated by taking the actual output (y) and the expected output (p). The formula for cross-entropy is shown below.
  $-(y\log(p) + (1-y)\log(1-p))$

2. Confusion Matrix:

- The confusion matrix tells us how well the model works and gives us a matrix or table as Output.

- Sometimes, this kind of structure is called the error matrix.
- The matrix is a summary of the results of the predictions. It shows how many predictions were right and how many were wrong.

The matrix looks like as below table:



3. AUC-ROC curve:

- The letters AUC and ROC stand for "area under the curve" and "receiver operating characteristics curve," respectively.
- This graph shows how well the classification model works at several different thresholds.
- We use the AUC-ROC Curve to see how well the multi-class classification model is doing.
- The (TPR)TruePositiveRate and the (FPR)FalsePositiveRate are used to plot the ROC curve, with TPR on the Y-axis and FPR on the X-axis.

Classification algorithms have several applications, Following are some popular applications or use cases of Classification Algorithms:

- Detecting Email Spam
- Recognizing Speech
- Detection of Cancer tumor cells.
- Classifying Drugs
- Biometric Identification, etc.

□ Check Your Progress 3

4. List the classification algorithms under the categories of Linear and Non-Linear Models. Also Discuss the various methods used for evaluating a classification model

……

……

10.5.1 NAÏVE BAYES

This is an example of a statistical classification, which estimates the likelihood that a particular sample belongs to a particular group given the sample in question. The Bayes theorem provides the foundation for it. When used to big databases, the Bayesian classification demonstrates both improved accuracy and increased speed. In this section, we will talk about the most basic kind of Bayesian categorization.

"The effect of a given attribute value on a certain class is unaffected by the values of other attributes, i.e. both are independent," is one of the fundamental underlying assumptions that underpin the native Bayesian classification, which is the simplest form of Bayesian classification. Class conditional independence is another name for this basic assumption.

Let's go into greater depth about the naïve Bayesian classification, shall we? But before we get into it, let's take a moment to define the fundamental theorem that underpins this classification i.e. the Bayes Theorem.

Bayes Theorem: In order to understand this theorem firstly lets understand the meaning of the following symbols or assumptions, they are as follows :

- X is an example of a data set whose class needs to be determined.
- H refers to the hypothesis which states that the data sample X falls into the class C.
- P(H | X): The probability that the data sample X belongs to the class C is expressed by the formula P(H | X), where H is the hypothesis and X is the data sample. This formula represents the likelihood that the data sample X belongs to the class C. The likelihood that the condition H is true for the sample X is often referred to as the posterior probability.
- The prior probability of the H condition is denoted by the notation P(H), which is based on the training data.
- The posterior probability of the X sample is denoted by the symbol P(X | H), which assumes that H is correct.
- The prior probability on the sample X is denoted by the letter P(X).

Note: From the data sample X and the data used for training, we may get the parameters P(X), P(X | H), and P(H). Whereas, P(H | X) is the only variable that, by itself, may define the likelihood that X belongs to a class C; this probability, however, cannot be calculated. This purpose is served by Bayes' theorem in particular.

The Bayer's theorem states:

$$P(H \mid X) = \frac{P(X \mid H)\, P(H)}{P(X)}$$

Now after defining the Bayes theorem, let us explain the Bayesian classification with the help of an example.

i) Consider the sample having an n-dimensional feature vector. For our example, it is a 3-dimensional (Department, Age, Salary) vector with training data as given in the Figure 3.

ii) Assume that there are m classes C_1 to C_m. And an unknown sample X. The problem is to data mine which class X belongs to. As per Bayesian classification, the sample is assigned to the class, if the following holds:

$$P(C_i|X) > P(C_j|X) \text{ where j is from 1 to m but } j \neq i$$

In other words the class for the data sample X will be the class, which has the maximum probability for the unknown sample. **Please note:** The $P(C_i|X)$ will be found using:

$$P(C_i|X) = \frac{P(X|C_i)\,P(C_i)}{P(X)} \qquad (3)$$

In our example, we are trying to classify the following data:
X = (Department = "Personal", Age = "31 – 40" and Salary = "Medium_Range)

into two classes (based on position) C_1=_BOSS_ OR C_2=_ASSISTANT_.

iii) The value of P(X) is constant for all the classes, therefore, only $P(X|C_i)\,P(C_i)$ needs to be found to be maximum. Also, if the classes are equally, then,
$P(C_1)=P(C_2)=\ldots..P(C_n)$, then we only need to maximise $P(X|C_i)$.

How is $P(C_i)$ calculated?

$$P(C_i) = \frac{\text{Number of training samples for Class } C_i}{\text{Total Number of Training Samples}}$$

In our example,

$$P(C_1) = \frac{5}{11}$$

and

$$P(C_2) = \frac{6}{11}$$

So $P(C_1) \neq P(C_2)$

iv) $P(X|C_i)$ calculation may be computationally expensive if, there are large numbers of attributes. To simplify the evaluation, in the naïve Bayesian classification, we use the condition of class conditional independence, that is the values of attributes are independent of each other. In such a situation:

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i) \qquad \ldots.(4)$$

where x_k represent the single dimension or attribute.

The $P(x_k|C_i)$ can be calculated using mathematical function if it is continuous, otherwise, if it is categorical then, this probability can be calculated as:

$P(x_k|C_i)$= Number of training samples of class C_i having the value x_k for the attribute A_k
Number of training samples belonging to C_i

For our example, we have x_1 as Department= "_PERSONNEL_"

x_2 as Age="31 – 40" and

x_3 as Salary= "Medium_Range"

P (Department= "_PERSONNEL_" | Position = "_BOSS_") = 1/5

P (Department= "_PERSONNEL_" | Position = "_ASSISTANT_") = 2/6

P (Age="31 – 40" | Position = "_BOSS_") = 3/5

P (Age="31 – 40" | Position = "_ASSISTANT_") = 2/6

P (Salary= "Medium_Range"| Position = "_BOSS_") = 3/5

P (Salary= "Medium_Range" | Position = "_ASSISTANT_") = 3/6

Using the equation (4) we obtain:

P (X | Position = "_BOSS_") = 1/5 * 3/5 * 3/5

P (X | Position = "_ASSISTANT_") = 2/6 * 2/6 * 3/6

Thus, the probabilities:

P (X | Position = "_BOSS_") P(Position = "_BOSS_")

= (1/5 * 3/5 * 3/5) * 5/11

= 0.032727

P (X | Position = "_ASSISTANT_") P(Position = "_ASSISTANT_")

= (2/6 * 2/6 * 3/6) * 6/11

= 0.030303

Since, the first probability of the above two is higher, the sample data may be classified into the _BOSS_ position. Kindly check to see that you obtain the same result from the decision tree .

Naiive Bayes : Steps to perform naiive bayes algorithm

Step 1: Handling Data : Data is loaded from the CSV File and spread into training and tested assets.

Step 2: Summarizing the Data : Summarise the properties in the training data set to calculate the probabilities and make predictions.

Step 3: Making a Prediction : A particular prediction is made using a summarise of the data set to make a single prediction.

Step 4: Making all the Predictions : Generate prediction given a test data set and a summarise data set.

Step 5: Evaluate Accuracy : Accuracy of the prediction model for the test data set as a percentage correct out of them all the predictions made.

Step 6: Tying all Together : Finally, we tie to all steps together and form our own model of Naive Bayes Classifier.

With the help of the following example, you can see how Naive Bayes' Classifier works:

Example: Let's say we have a list of _WEATHER_ conditions and a target variable called "Play" that goes with it. So, using this set of data, we need to decide whether or not to play on a given day based on the _WEATHER_.
If it's _SUNNY_, should the Player play?
So, here are the steps we need to take to solve this problem:

1. Make frequency tables out of the given dataset.

2. Make a Likelihood table by figuring out how likely each feature is.

Use Bayes's theorem to figure out the posterior probability.

To figure this out, first look at the dataset given below:

	OUTLOOK	_PLAY_
0	_RAINY_	YES
1	_SUNNY_	YES
2	_OVERCAST_	YES
3	_OVERCAST_	YES
4	_SUNNY_	NO
5	_RAINY_	YES
6	_SUNNY_	YES
7	_OVERCAST_	YES
8	_RAINY_	NO
9	_SUNNY_	NO
10	_SUNNY_	YES
11	_RAINY_	NO
12	_OVERCAST_	YES
13	_OVERCAST_	YES

Frequency table for the _WEATHER_ Conditions:

WEATHER	NO	YES
OVERCAST	0	5
RAINY	2	2
SUNNY	2	3
TOTAL	4	10

Likelihood table _WEATHER_ condition:

WEATHER	NO	YES	
OVERCAST	0	5	5/14=0.35
RAINY	2	2	4/14=0.29
SUNNY	2	3	5/14=0.35
ALL	4/14 = 0.29	10/14 = 0.71	

Applying Bayes'theorem:

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

P(_SUNNY_|Yes)= 3/10= 0.3

P(_SUNNY_)= 0.35

P(Yes)=0.71

So P(Yes|_SUNNY_) = 0.3*0.71/0.35= **0.60**

P(No|_SUNNY_)= P(_SUNNY_|No)*P(No)/P(_SUNNY_)

P(_SUNNY_|NO)= 2/4=0.5

P(No)= 0.29

P(_SUNNY_)= 0.35

So P(No|_SUNNY_)= 0.5*0.29/0.35 = **0.41**

So as we can see from the above calculation that **P(Yes|_SUNNY_)>P(No|_SUNNY_)**

Hence on a _SUNNY_ day, Player can play the game.

Check Your Progress 4

8. Predicting a class label using naïve Bayesian classification. We wish to predict the class label of a tuple using naïve Bayesian classification, given the training data as shown in Table-1 Below. The data tuples are described by the attributes age, income, student, and credit rating.

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

The class label attribute known as "buys computer" can take on one of two distinct values—specifically, "yes" or "no." Let's say that C1 represents the class buying a computer and C2 represents the class deciding not to buy a computer. We are interested in classifying X as having the following characteristics: (age = youth, income = medium, student status = yes, credit rating = fair).

10.5.2 K-NEAREST NEIGHBOURS (K-NN)

This approach, places items in the class to which they are “closest” to their neighbour.It must determine distance between an item and a class. Classes are represented by centroid (Central value) and the individual points.One of the algorithms that is used is K-Nearest Neighbors.

We know that The classification task maps data into predefined groups or classes. Given database/dataset $D=\{t_1,t_2,\ldots,t_n\}$ and a set of classes $C=\{C_1,\ldots,C_m\}$, the classification Problem is to define a mapping $f:D\rightarrow C$ where each t_i is assigned to one class, that is, it divides database/dataset D into classes specified in the Set C.
A few very simple examples to elucidate classification could be:

- Teachers classify students’ marks data into a set of grades as A, B, C, D, or F.
- Classification of the height of a set of persons into the classes tall, medium or short.

The basic approaches to classification are:

- To create specific models by, evaluating training data, which is basically the old data, that has already been classified by using the domain of the experts' knowledge.
- Now applying the model developed to the new data.

Please note that in classification, the classes are predefined.

Some of the most common techniques used for classification may include the use of Decision Trees, K-NN etc. Most of these techniques are based on finding the distances or uses statistical methods.

The distance measure finds, the distance or dissimilarity between objects the measures that are used in this unit are as follows:

- Euclidean distance: $dis(t_i,t_j)= \sqrt{\sum_{h=1}^{k} (t_{ih} - t_{jh})^2}$
- Manhattan distance: $dis(t_i,t_j)= \sum_{h=1}^{k} |(t_{ih} - t_{jh})|$

where t_i and t_j are tuples and h are the different attributes which can take values from 1 to k

In this section, we look at the distance based classifier i.e. the k-nearest neighbor classifiers.

A test tuple is compared to training tuples that are used in the classification process that are similar to it. This is how nearest-neighbor classifiers work. There are n different characteristics that can be used to define the training tuples. Each tuple can be thought of as a point located in a space that has n dimensions. In this method, each and every one of the training tuples is preserved within an n-dimensional pattern space. A K-nearest-neighbor classifier searches the pattern space in order to find the k training tuples that are the most comparable to an unknown tuple. These k training tuples are referred to as the "k nearest neighbours" of the unknown tuple.

A distance metric, like Euclidean distance, is used to define "closeness."The Euclidean distance between two points or tuples, say, X1 = (x11, x12,..., x1n) and X2 = (x21, x22,..., x2n), is

$$dist(\boldsymbol{X_1}, \boldsymbol{X_2}) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}. \quad \textbf{(eq. 1)}$$

In other words, for each numeric attribute, we take the difference in value between the corresponding values of that property in tuple X1 and the values of that attribute in tuple X2 and then square this difference. Finally, we add up all of these differences. The final tally of all the accumulated distances is used to calculate the square root. In most cases, prior to making use of Equation, we first normalize the values of every property (eq. 1). This helps avoid attributes with initially high ranges (like income, for example) from outweighing attributes with originally lower ranges, which helps prevent unfairness (such as binary attributes). Min-max normalization, for example, can be used to change the value of a numeric attribute A from v to v' in the range [0, 1].

$$v' = \frac{v - min_A}{max_A - min_A},$$ **(eq. 2)**

Where, min_A and max_A are the minimum and maximum values of attribute A

For the purpose of k-nearest-neighbor classification, the unknown tuple is assigned to the class that has the highest frequency among its k closest neighbours. When k equals 1, the class of the training tuple that is assigned to the unknown tuple is the one that is most similar to the unknown tuple in pattern space. It is also possible to utilise nearest neighbour classifiers for prediction, which means that they can be used to deliver a real-valued forecast for a given unknown tuple. The result that the classifier produces in this scenario is the weighted average of the real-valued labels that are associated with the unknown tuple's k nearest neighbours.

Classification Using Distance (K-Nearest Neighbours) - Some of the basic points to be noted about this algorithm are:

- The training set includes *classes* along with other attributes. (Please refer to the training data given in the *Table* given below).
- The value of the K defines the number of *near items* (items that have less distance to the attributes of concern) that should be used from the given set of training data (just to remind you again, training data is already classified data). This is explained in point (2) of the following example.
- A new item is placed in the class in which the most number of close items are placed. (Please refer to point (3) in the following example).
- The value of K should be <= $\sqrt{Number\_of\_training\_items}$ However, in our example for limiting the size of the sample data, we have not followed this formula.

Example: Consider the following data, which tells us the person's class depending upon gender and height

Name	Gender	Height	Class
Sunita	F	1.6m	Short
Ram	M	2m	Tall
Namita	F	1.9m	Medium
Radha	F	1.88m	Medium
Jully	F	1.7m	Short
Arun	M	1.85m	Medium
Shelly	F	1.6m	Short
Avinash	M	1.7m	Short
Sachin	M	2.2m	Tall
Manoj	M	2.1m	Tall
Sangeeta	F	1.8m	Medium
Anirban	M	1.95m	Medium
Krishna	F	1.9m	Medium
Kavita	F	1.8m	Medium
Pooja	F	1.75m	Medium

1) You have to classify the tuple <Ram, M, 1.6> from the training data that is given to you.

2) Let us take only the **height** attribute for distance calculation and suppose K=5 then the following are the near five tuples to the data that is to be classified (using Manhattan distance as a measure on the height attribute).

Name	Gender	Height	Class
Sunita	F	1.6m	Short
Jully	F	1.7m	Short
Shelly	F	1.6m	Short
Avinash	M	1.7m	Short
Pooja	F	1.75m	Medium

3) On examination of the tuples above, we classify the tuple <Ram, M, 1.6> to *Short* class since most of the tuples above belongs to *Short* class.

Example- To classify whether a special paper tissue is Fine or not, we used data from a questionnaire survey (to get people's opinions) and objective testing with two properties (acid durability and strength). Here are four examples of training.

X1 = Acid_Durability_(seconds)	X2 = Strength(gram/Cm2)	Y = Classification
7	7	Poor
7	4	Poor
3	4	Fine
1	4	Fine

Now, the firm is producing a new kind of paper tissue that is successful in the laboratory and has the values X1 = 3 and X2 = 7 respectively. Can we make an educated judgement about the classification of this novel tissue without doing yet another expensive survey?

1. Find the value of the parameter K as the number of the nearest neighbours Suppose use K = 3
2. Determine the distance that separates the query-instance from each of the samples used for training

The coordinates of the query instance are (3, 7), and rather than computing the distance, we compute the square distance, which is a more efficient calculation (without square root)

X1 = Acid_Durability_(seconds)	X2 = Strength(gram/Cm2)	**Square Distance to query instance (3, 7)**
7	7	$(7-3)^2+(7-7)^2=16$
7	4	$(7-3)^2+(4-7)^2=25$
3	4	$(3-3)^2+(4-7)^2=9$
1	4	$(1-3)^2+(4-7)^2=13$

2. *Sort the distance and determine nearest neighbors based on the K-th minimum distance*

X1 = Acid_Durability_(seconds)	X2 = Strength (gram/Cm2)	Square Distance to query instance (3, 7)	**Rank minimum distance**	**Is it included in 3-Nearest neighbors?**
7	7	$(7-3)^2+(7-7)^2=16$	**3**	**Yes**
7	4	$(7-3)^2+(4-7)^2=25$	**4**	**No**
3	4	$(3-3)^2+(4-7)^2=9$	**1**	**Yes**
1	4	$(1-3)^2+(4-7)^2=13$	**2**	**Yes**

3. *Gather the category (Y) of the nearest neighbors.* Notice in the second row last scolumn that the category of nearest neighbor (Y) is not included because the rank of this data is more than 3 (=K).

X1 = Acid_Durability_(seconds)	X2 = Strength (gram/Cm2)	Square Distance to query instance (3, 7)	Rank minimum distance	Is it included in 3-Nearest neighbors?	**Y = Category of nearest Neighbor**
7	7	$(7-3)^2+(7-7)^2=16$	3	Yes	**Poor**
7	4	$(7-3)^2+(4-7)^2=25$	4	No	-
3	4	$(3-3)^2+(4-7)^2=9$	1	Yes	**Fine**
1	4	$(1-3)^2+(4-7)^2=13$	2	Yes	**Fine**

4. *Use simple majority of the category of nearest neighbors as the prediction value of the query instance*

Use the simple majority of the nearest neighbours as the query instance's prediction value.

Since 2<1 and we have 2 Fine and 1 Poor, So we can say that a new paper tissue that passed the lab test with X1 = 3 and X2 = 7 is in the **Fine** category.

"However, distance cannot be determined using qualities that are categorical, as opposed to quantitative, such as color." The preceding description operates under the presumption that all of the attributes that are used to describe the tuples are numeric. When dealing with categorical attributes, a straightforward way is to contrast the value of the attribute that corresponds to tuple X1 with the value that corresponds to tuple X2. If there is no difference between the two (for example, If tuple X1 and X2 both contain the blue color, then the difference between the two is

regarded as being equal to zero. If the two are distinct from one another (for example, if tuple X1 carries blue and tuple X2 carries red), then the comparison between them is counted as 1. It's possible that other ways will incorporate more complex systems for differentiating grades (such as in a scenario in which a higher difference score is provided, say, for blue and white than for blue and black).

"What about the missing values?" If the value of a certain attribute A is absent from either the tuple X1 or the tuple X2, we will, as a rule, assume the greatest feasible disparity between the two. Imagine for a moment that each of the traits has been mapped to the interval [0, 1]. When it comes to categorical attributes, the difference value is set to 1 if either one of the related values of A or both of them are absent. If A is a number and it is absent from both the tuple X1 and the tuple X2, then the difference is also assumed to be 1. If there is just one value that is absent and the other value (which we will refer to as v 0) is present and has been normalised, Consequently, we can either take the difference to be |1 – v' | or |0 – v' | (i.e., 1–v' or v'), depending on which of the two is larger.

Nearest-neighbor classifiers use comparisons based on distance to give each attribute an equal amount of weight. So, they can be less accurate if their attributes are noisy or don't make sense. But the method has been changed to include the weighting of attributes and the removal of noisy data tuples. How you measure distance can be very important. You could also use the city block distance or another way to measure distance.

Nearest neighbour classifiers can be very slow when classifying test tuples into groups. If D is a training database that contains |D| tuples and k is equal to one, then in order to classify a given test tuple, it must be compared to |D| training tuples. It is possible to reduce the total number of comparisons to O(log(|D|) by first putting the stored tuples in search trees and then performing the comparisons. The running time can be reduced to O(1) if parallel implementation is used, which is a constant that doesn't change no matter how big D is. You can also use partial distance calculations and change the stored tuples to cut down on the time it takes to classify. In the partial distance method, we use only some of the n attributes to figure out how far apart two things are. If this distance exceeds a specified threshold, the procedure aborts the execution of the current stored tuple and continues on to the next. Training tuples that aren't required are removed using the editing procedure. This strategy is also known as pruning or condensing. because it minimises the number of stored tuples.

Check Your Progress 5

9. Apply KNN classification algorithm to the following data and predict value for (10,7) for K = 3

Feature 1	Feature 2	Class
1	1	A
2	3	A
2	4	A
5	3	A
8	6	B
8	8	B
9	6	B
11	7	B

……

……

10.5.3 DECISION TREES

Given a data set D = $\{t_1,t_2 \ldots, t_n\}$ where $t_i=<t_{i1}, \ldots, t_{ih}>$, that is, each tuple is represented by h attributes, assume that, the database schema contains attributes as $\{A_1, A_2, \ldots, A_h\}$. Also, let us suppose that the classes are $C=\{C_1, \ldots., C_m\}$, then:

Decision or Classification Tree is a tree associated with D such that

- Each internal node is labeled with attribute, A_i
- Each arc is labeled with the predicate which can be applied to the attribute at the parent node.
- Each leaf node is labeled with a class, C_j

Basics steps in the Decision Tree are as follows:

- Building the tree by using the training set dataset/database.
- Applying the tree to the new dataset/database.

Decision Tree Induction is the process of learning about the classification using the inductive approach. During this process, we create a decision tree from the training data. This decision tree can, then be used, for making classifications. To define this we need to define the following.

Let us assume that we are given probabilities $p_1, p_2, .., p_s$ whose sum is 1. Let us also define the term Entropy, which is the measure of the amount of randomness or surprise or uncertainty. Thus our basic

goal in the classification process is that, the entropy for a classification should be zero, that, if no surprise then, entropy is equal to zero. Entropy is defined as:

$$H(p_1,p_2,\ldots p_s)= \sum_{i=1}^{s} (p_i * \log(1/p_i)) \qquad \ldots\ldots. \quad (1)$$

ID3 Algorithm for Classification

This algorithm creates a tree using the algorithm given below and tries to reduce the expected number of comparisons.

Algorithm: ID3 algorithm for creating decision tree from the given training data.
Input: The *training data* and the *attribute-list*.

Output: A decision tree.

Process: Step 1: Create a node N

Step 2: If all of the sample data belong to the same class, C, which means the probability is 1, then return N as a leaf node with the class C label.

Step 3: Return N as a leaf node if attribute-list is empty, and label it with the most common class with in training data; // majority voting

Step 4: Select *split-attribute*, which is the attribute in the *attribute-list* with the highest information gain;

Step 5: label node N with *split-attribute*;

Step 6: for each known value Ai, of *split-attribute* // partition the samples

Create a branch from node N for the condition: *split-attribute* = A_i;

// Now consider a partition and recursively create the decision tree:

Let xi be the set of data from training data that satisfies the condition:

split-attribute = A_i

if the set xi is empty then

attach a leaf labeled with the most common class in the prior set of training data;

else

attach the node returned after recursive call to the program with training data as xi and

new attribute list = present attribute-list – split-attribute;

End of Algorithm.

Please note: The algorithm given above, chooses the split attribute with the highest information gain, that is, calculated as follows:

$$\text{Gain (D,S)} = H(D) - \sum_{i=1}^{s} (P(D_i) * H(D_i)) \quad \text{.....(2)}$$

where S is new states $=\{D_1, D_2, D_3 \ldots D_S\}$ and H(D) finds the amount of order in that state
Consider the following data in which *Position* attribute acts as class

Department	Age	Salary	Position
PERSONNEL	31-40	Medium_Range	_BOSS_
PERSONNEL	21-30	Low_Range	_ASSISTANT_
PERSONNEL	31-40	Low_Range	_ASSISTANT_
MIS	21-30	Medium_Range	_ASSISTANT_
MIS	31-40	High_Range	_BOSS_
MIS	21-30	Medium_Range	_ASSISTANT_
MIS	41-50	High_Range	_BOSS_
ADMIN	31-40	Medium_Range	_BOSS_
ADMIN	31-40	Medium_Range	_ASSISTANT_
SECURITY	41-50	Medium_Range	_BOSS_
SECURITY	21-30	Low_Range	_ASSISTANT_

Figure 3: Sample data for classification

We are applying ID3 algorithm, on the above dataset as follows:
The initial entropy of the dataset using formula at (1) is

$$H(\text{initial}) = (6/11)\log(11/6) + (5/11)\log(11/5) = 0.29923$$
(_ASSISTANT_) (_BOSS_)

Now let us calculate gain for the departments using the formula at (2)

Gain(Department) = H(initial) – [P(_PERSONNEL_) * H(_MIS_) + P(_MIS_) * H(_PERSONNEL_) +
P(_ADMIN_) * H(_ADMIN_) + P(_SECURITY_) * H(_SECURITY_)]

= *0.29923*- { (3/11)[(1/3)log3 +(2/3)log(3/2)] + (4/11)[(2/4)log 2 + (2/4)log 2] +
(2/11)[(1/2)log 2 + (1/2)log 2] + (2/11)[(1/2)log 2 + (1/2)log 2] }
= 0.29923- 0.2943
= 0.0049

Similarly:

Gain(Age) = 0.29923- { (4/11)[(4/4)log(4/4)] + (5/11)[(3/5)log(5/3) + (2/5)log (5/2)] +
(2/11)[(2/2) log (2/2)] }
= 0.29923 - 0.1328
= 0.1664

Gain(Salary) = 0.29923 – { (3/11)[(3/3)log 3] + (6/11)[(3/6) log2 + (3/6)log 2] +
(2/11) [(2/2 log(2/2)) }
= 0.29923 – 0.164
= 0.1350

Since age has the maximum gain, so, this attribute is selected as the first splitting attribute. In age range 31-40, class is not defined while for other ranges it is defined.

So, we have to again calculate the spitting attribute for this age range (31-40). Now, the tuples that belong to this range are as follows:

Department	Salary	Position
PERSONNEL	Medium_Range	_BOSS_
PERSONNEL	Low_Range	_ASSISTANT_
MIS	High_Range	_BOSS_
ADMIN	Medium_Range	_BOSS_
ADMIN	Medium_Range	_ASSISTANT_

Again the initial entropy = (2/5)log(5/2) + (3/5)log(5/3) = 0.29922
(_ASSISTANT_) (_BOSS_)

Gain(Department)= 0.29922- { (2/5)[(1/2)log 2 + (1/2)log 2] +1/5[(1/1)log1] +
(2/5)[(1/2)log 2 + (1/2)log 2] }
=0.29922- 0.240
=0.05922
Gain (Salary) = 0.29922- { (1/5) [(1/1)log1] +(3/5)[(1/3)log 3 + (2/3)log(3/2)] +
(1/5) [(1/1)log1] }
= 0.29922- 0.1658
=0.13335
The Gain is maximum for salary attribute, so we take salary as the next splitting attribute. In middle range salary, class is not defined while for other ranges it is defined. So, we have to again calculate the spitting attribute for this middle range. Since only department is left, so, department will be the next splitting attribute. Now, the tuples that belong to this salary range are as follows:

Department	Position
PERSONNEL	_BOSS_
ADMIN	_BOSS_
ADMIN	_ASSISTANT_

Again in the _PERSONNEL_ department, all persons are _BOSS_, while, in the _ADMIN_ there is a tie between the classes. So, the person can be either _BOSS_ or _ASSISTANT_ in the _ADMIN_ department.

Now the decision tree will be as follows:

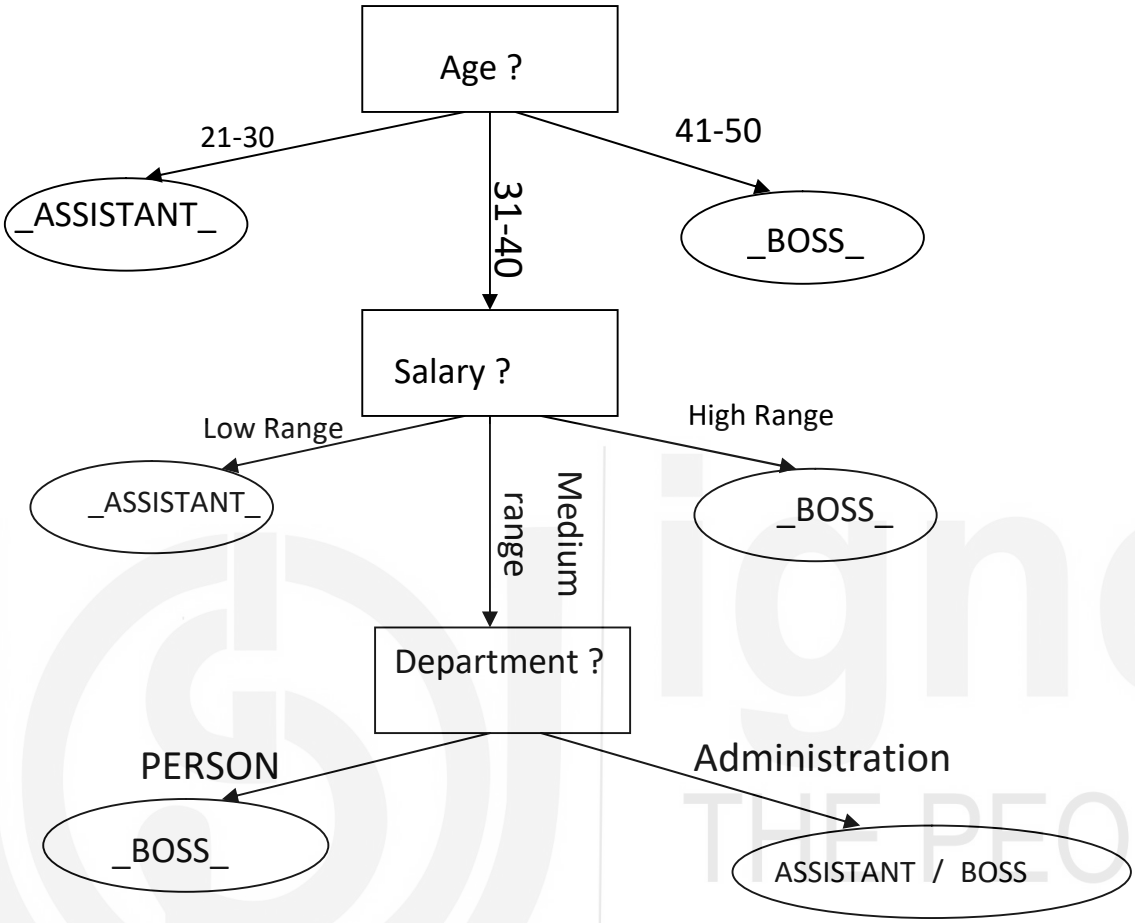


Figure 4: The decision tree using ID3 algorithm for the sample data

Now, we will take a new dataset and we will classify the class of each tuple by applying the decision tree that we have built above.

Steps of algorithm of decision tree

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

Example : Problem on Decision Tree - Consider whether a dataset based on which we will determine whether to play football or not.

OUTLOOK	TEMP.	HUMIDITY	WIND	PLAY FOOTBALL(YES/NO)
SUNNY	_HOT_	HIGH	WEAK	NO
SUNNY	_HOT_	HIGH	STRONG	NO
OVERCAST	_HOT_	HIGH	WEAK	YES
RAINY	_MILD_	HIGH	WEAK	YES
RAINY	_COOL_	NORMAL	WEAK	YES
RAINY	_COOL_	NORMAL	STRONG	NO
OVERCAST	_COOL_	NORMAL	STRONG	YES
SUNNY	_MILD_	HIGH	WEAK	NO
SUNNY	_COOL_	NORMAL	WEAK	YES
RAINY	_MILD_	NORMAL	WEAK	YES
SUNNY	_MILD_	NORMAL	STRONG	YES
OVERCAST	M	HIGH	STRONG	YES
OVERCAST	_HOT_	NORMAL	WEAK	YES
RAINY	_MILD_	HIGH	STRONG	NO

Here There are for independent variables to determine the dependent variable. The independent variables are _OUTLOOK_, TEMP., Humidity, and Wind. The dependent variable is whether to play football or not.

As the first step, we have to find the parent node for our decision tree. For that follow the steps:

Find the entropy of the class variable. E(S) = -[(9/14)log(9/14) + (5/14)log(5/14)] = 0.94

note: Here typically we will take log to base 2.Here total there are 14 yes/no. Out of which 9 yes and 5 no.Based on it we calculated probability above.

From the above data for _OUTLOOK_ we can arrive at the following table easily

		PLAY		
		YES	NO	TOTAL
	SUNNY	3	2	5
OUTLOOK	_OVERCAST_	4	0	4
	RAINY	2	3	5
				14

Now we have to calculate average weighted entropy. ie, we have found the total of weights of each feature multiplied by probabilities.

E(S, _OUTLOOK_) = (5/14)*E(3,2) + (4/14)*E(4,0) + (5/14)*E(2,3) = (5/14)(-(3/5)log(3/5)-(2/5)log(2/5))+ (4/14)(0) + (5/14)((2/5)log(2/5)-(3/5)log(3/5)) = 0.693

The next step is to find the information gain. It is the difference between parent entropy and average weighted entropy we found above.

IG(S, _OUTLOOK_) = 0.94 - 0.693 = 0.247

Similarly find Information gain for TEMP., Humidity, and Windy.

IG(S, TEMP.) = 0.940 - 0.911 = 0.029

IG(S, Humidity) = 0.940 - 0.788 = 0.152

IG(S, Windy) = 0.940 - 0.8932 = 0.048

Now select the feature having the largest entropy gain. Here it is _OUTLOOK_. So it forms the first node(root node) of our decision tree.

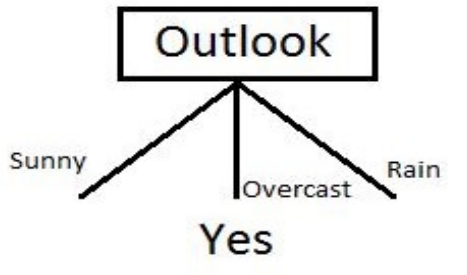
Now our data look as follows

OUTLOOK	TEMP.	HUMIDITY	WIND	PLAYED(YES/NO)
SUNNY	_HOT_	HIGH	WEAK	NO
SUNNY	_HOT_	HIGH	STRONG	NO
SUNNY	_MILD_	HIGH	WEAK	NO
SUNNY	_COOL_	NORMAL	WEAK	YES
SUNNY	_MILD_	NORMAL	STRONG	YES

OUTLOOK	TEMP.	HUMIDITY	WIND	PLAYED(YES/NO)
OVERCAST	_HOT_	HIGH	WEAK	YES
OVERCAST	_COOL_	NORMAL	STRONG	YES
OVERCAST	_MILD_	HIGH	STRONG	YES
OVERCAST	_HOT_	NORMAL	WEAK	YES

OUTLOOK	TEMP.	HUMIDITY	WIND	PLAYED(YES/NO)
RAIN	_MILD_	HIGH	WEAK	YES
RAIN	_COOL_	NORMAL	WEAK	YES
RAIN	_COOL_	NORMAL	STRONG	NO
RAIN	_MILD_	NORMAL	WEAK	YES
RAIN	_MILD_	HIGH	STRONG	NO

Since _OVERCAST_ contains only examples of class ‘Yes’ we can set it as yes. That means If _OUTLOOK_ is _OVERCAST_ football will be played. Now our decision tree looks as follows.



The next step is to find the next node in our decision tree. Now we will find one under _SUNNY_. We have to determine which of the following TEMP., Humidity or Wind has higher information gain.

OUTLOOK	TEMP.	HUMIDITY	WIND	PLAYED(YES/NO)
SUNNY	_HOT_	HIGH	WEAK	NO
SUNNY	_HOT_	HIGH	STRONG	NO
SUNNY	_MILD_	HIGH	WEAK	NO
SUNNY	_COOL_	NORMAL	WEAK	YES
SUNNY	_MILD_	NORMAL	STRONG	YES

Calculate parent entropy E(_SUNNY_)

E(_SUNNY_) = (-(3/5)log(3/5)-(2/5)log(2/5)) = 0.971.

Now Calculate the information gain of TEMP.. IG(_SUNNY_, TEMP.)

		PLAY		
		YES	NO	TOTAL
	HOT	0	2	2
TEMP.	_COOL_	1	1	2
	MILD	1	0	1
				5

E(_SUNNY_, TEMP.) = (2/5)*E(0,2) + (2/5)*E(1,1) + (1/5)*E(1,0)=2/5=0.4

Now calculate information gain.

IG(_SUNNY_, TEMP.) = 0.971–0.4 =0.571

Similarly we get

IG(_SUNNY_, Humidity) = 0.971

IG(_SUNNY_, Windy) = 0.020

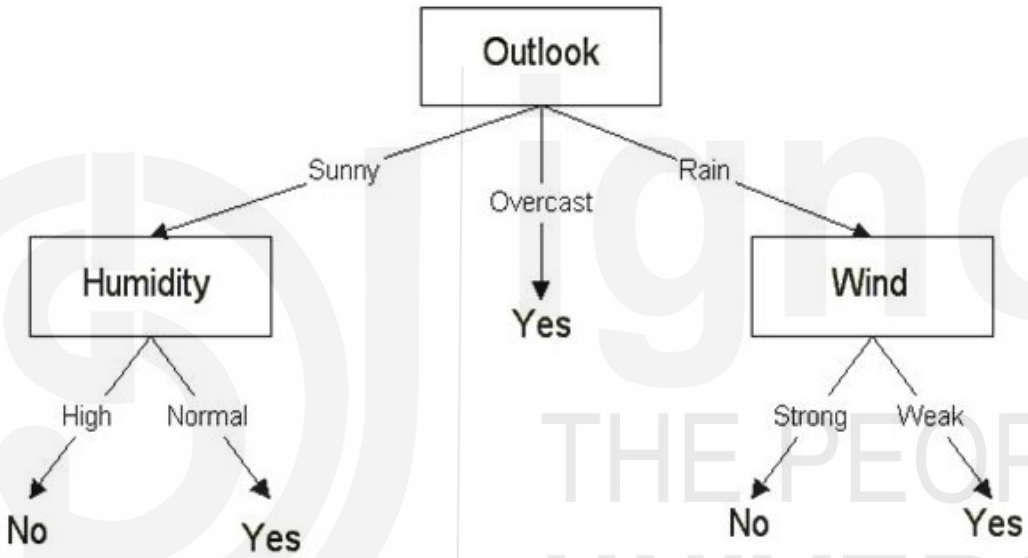
Here IG(_SUNNY_, Humidity) is the largest value. So Humidity is the node that comes under _SUNNY_.

		PLAY		
		YES	NO	TOTAL
	HIGH	0	3	3
HUMIDITY	NORMAL	2	0	2
				5

For humidity from the above table, we can say that play will occur if humidity is normal and will not occur if it is high. Similarly, find the nodes under _RAINY_.

Note: A branch with entropy more than 0 needs further splitting.

Finally, our decision tree will look as below:



10.5.4 LOGISTIC REGRESSION

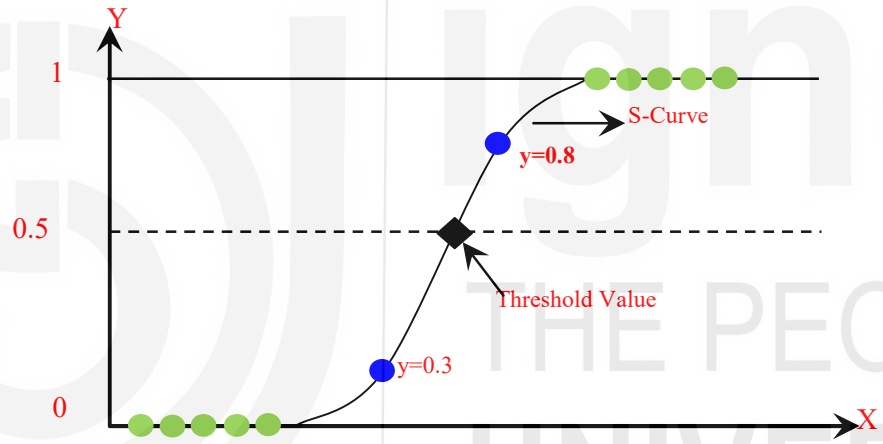
Logistic Regression in Machine Learning

- Logistic regression, which is part of the Supervised Learning method, is one of the most popular Machine Learning algorithms. It is used to predict the categorical dependent variable based on a set of independent variables.
- Logistic regression predicts the outcome of a dependent variable that has a "yes" or "no" answer. Because of this, the result must be a discrete or categorical value. It can be Yes or No, 0 or 1, true or false, etc., but instead of giving the exact value as 0 or 1, it gives the probabilistic values that lie between 0 and 1.

- Logistic Regression is a lot like Linear Regression, but the way they are used is different. Linear regression is used to solve regression problems, while logistic regression is used to solve classification problems.
- In logistic regression, we fit a "S"-shaped logistic function, which predicts two maximum values, instead of a regression line (0 or 1).
- The curve from the logistic function shows how likely something is, like whether the cells are cancerous or not, whether a mouse is overweight or not based on its weight, etc.

Logistic Regression is an important machine learning algorithm because it can use both continuous and discrete datasets to give probabilities and classify new data.

Logistic regression can be used to classify observations based on different types of data, and it is easy to figure out which variables are the most useful for classifying. The logistic function is shown in the picture below:



Note: Logistic regression is based on the idea of predictive modeling as regression, so that's why it's called "logistic regression." However, it's used to classify samples, so it's a part of the classification algorithm.

Logistic Function (Sigmoid Function):

- The math "function" called the "sigmoid function" turns predicted values into probabilities.
- The sigmoid function is a special case of the logistic function. It is usually written as (x) or sig (x). The formula for it is: $\sigma(x) = 1/(1+\exp(-x))$
- It turns any real number into a different number between 0 and 1.
- The value of the logistic regression must be between 0 and 1, and it can't be higher than that. Because of this, it forms a curve that looks like the letter "S." The Sigmoid function or the logistic function is the name for the curve in the shape of a S.

- The threshold value tells us how likely it is that either 0 or 1 will happen in logistic regression. For example, most values above the threshold are 1, and most values below it are 0.

Assumptions for Logistic Regression:

- The dependent variable must be a categorical one.
- The independent variable shouldn't be related to more than one other variable.

Types of Logistic Regression: Based on the categories, Logistic Regression can be divided into three types:

• **Binomial**: In binomial logistic regression, the dependent variables can only be either 0 or 1, Pass or Fail, etc.

• **Multinomial**: In multinomial Logistic regression, the dependent variable can be one of three or more types that are not in order, such as "cats," "dogs," or "sheep."

• **Ordinal:** In ordinal Logistic regression, the dependent variables can be ranked, such as "low," "medium," or "high."

Logistic Regression Equation From the Linear Regression equation, you can figure out what the Logistic Regression equation is. Here are the steps you need to take in math to get Logistic Regression equations:

- We know the equation of the straight line can be written as:

  $$y = b_0+b_1X_1+b_2X_2+\ldots\ldots+b_nX_n$$

- In Logistic Regression y can be between 0 and 1 only, so for this let's divide the above equation by (1-y): y/(1-y); 0 for y=0 and infinity for y=1
- But we need range between -[infinity] to +[infinity], then take logarithm of the equation it will become: $\mathbf{Log[y/(1-y)] = b_0+b_1X_1+b_2X_2+\ldots\ldots+b_nX_n}$

The above equation is the final equation for Logistic Regression.

The relationship between a numerical response and a numerical or categorical predictor is the subject of the statistical technique known as simple linear regression. While multiple regression looks at the relationship between a single numerical response and a number of different numerical and/or categorical predictors, single regression looks at the relationship between a single numerical response and a single. What should be done, however, when the predictors are odd (nonlinear, intricate dependence structure, and so on), or when the response is unusual (categorical, count data, and so on)? When this occurs, we deal with odds, which are another method of measuring the likelihood of an event and are frequently applied in the context of gambling (and logistic regression).

Odds For some event E is expressed as,

$$odds(E) = P(E)/P(Ec\ ) = P(E)/(1 - P(E))$$

Similarly, if we are told the odds of E are x to y then

$$odds(E) = x/\ y = \{x/(x + y)\}/\{y/(x + y)\}$$

which implies P(E) = x/(x + y), P(E c) = y/(x + y)

Logistic regression is a statistical approach for modelling a binary categorical variable using numerical and categorical predictors, and this idea of Odds is commonly employed in it. We suppose the outcome variable was generated by a binomial distribution, and we wish to build a model with p as the probability of success for a given collection of predictors. There are other alternatives, but the logit function is the most popular.

$$\text{Logit function: } \text{logit}(p) = \log\{p/(1-p)\}, \text{ for } 0 \le p \le 1$$

Example-1: In a survey of 250 customers of an auto dealership, the service department was asked if they would tell a friend about it. The number of people who said "yes" was 210, where "p" is the percentage of customers in the group from which the sample was taken who would answer "yes" to the question. Find the sample odds and sample proportion.

Solution: The number of customers who would respond Yes in an simple random sample (SRS) of size n has the binomial distribution with parameters n and p. The sample size of customers is n = 250, and the number who responded Yes is the count X = 210. Therefore, the sample proportion is p'=210/250 = 0.84

Since, Logistic regressions work with odds rather than proportions. We need to calculate the Odds, the odds are simply the ratio of the proportions for the two possible outcomes. If p' is the proportion for one outcome, then 1 – p' is the proportion for the second out

$$\text{odds} = p' / (1 - p')$$

A similar formula for the population odds is obtained by substituting p for p' in this expression

Odds of responding Yes. For the customer service data, the proportion of customers who would recommend the service in the sample of customers is p' = 0.84, so the proportion of customers who would not recommend the service department will be 1 – p' i.e. 1 – p' = 1 - 0.84 = 0.16

Therefore, the odds of recommending the service department are

$$\text{odds} = p'/1 - p' = 0.84/0.16 = 5.25$$

When people speak about odds, they often round to integers or fractions. If we round 5.25 to 5 = 5/1, we would say that the odds are approximately 5 to 1 that a customer would recommend the service to a friend. In a similar way, we could describe the odds that a customer would not recommend the service as 1 to 5.

Check Your Progress 6

Q1 Odds of drawing a heart. If you deal one card from a standard deck, the probability that the card is a heart is 13/52 = 1/4.

(a) Find the odds of drawing a heart.
(b) Find the odds of drawing a card that is not a heart.

10.5.5 SUPPORT VECTOR MACHINES

Support Vector Machine, also called Support Vector Classification, is a supervised and linear Machine Learning technique that is most often used to solve classification problems. In this section, we will take a look at Support Vector Machines, a new approach for categorising data that has a lot of potential and can be used for both linear and nonlinear datasets. A support vector machine, often known as an SVM, is a type of algorithm that transforms the primary training data into a new format that has a higher dimension by making use of a nonlinear mapping. It searches for the ideal linear separating hyperplane in this additional dimension. This hyperplane is referred to as a "decision boundary" since it separates the tuples of one class from those of another. A hyperplane can always be used to split data from two classes if the appropriate nonlinear mapping to a high enough dimension is used. This hyperplane is located by the SVM through the use of support vectors, also known as important training tuples, and margins (defined by the support vectors). We'll go into further detail about these fresh concepts in the next paragraphs.

While studying machine learning, one of the classifiers that we come across is called a Support Vector Machine, or SVM for short. One of the most common approaches for categorising data in the field of machine learning, which performs admirably on both small and large datasets. SVMs, which stands for support vector machines, can be utilised for both classification and regression jobs; however, their performance is superior when applied to classification scenarios. When they were first introduced in the 1990s, they quickly became quite popular, and even now, with only minor adjustments, they are the solution of choice when a high-performing algorithm is required.

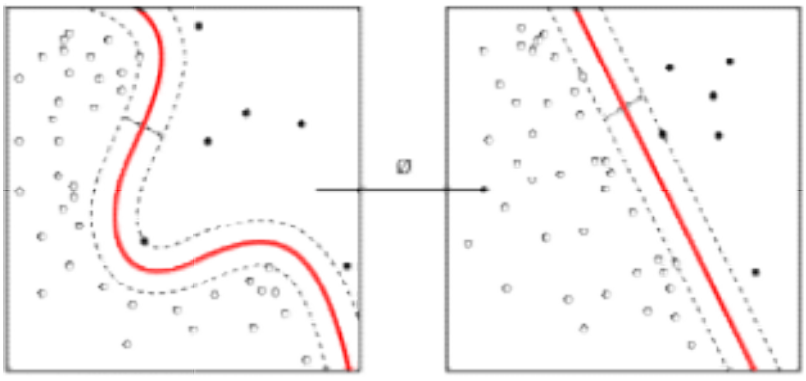


Fig1. Showing data into two groups

There are two main categories that can be applied to SVM:

• **Linear Support Vector Machine**: You can only use Linear SVM if the data can be completely separated into linear categories. The ability to separate a set of data points into two classes using just one straight line is what is meant when we talk about something being "completely linearly separable" (if 2D).

• **Non-Linear Support Vector Machine**: When the data isn't linearly separable, we can use Non-Linear SVM, which means that we apply advanced techniques like kernel tricks to categorise the data points that can't be divided into two classes using a straight line. This allows us to use Non-Linear SVM to classify the data points that aren't linearly separable (in 2D). We don't discover datapoints that are linearly separable in the majority of real-world applications, so we use the kernel approach to solve them instead.

Let's take a look of some SVM terminology.

- **Support Vectors:** The points on the hyperplane that are closest to the object in question are referred to as support vectors. The boundary between the two groups will be determined with the help of these data points.Infact these are the spots on the hyperplane that are closest to it. These data points will be used to define a separation line.
- **Margin:** The margin is the distance between the hyperplane and the nearest observations to the hyperplane (support vectors). A margin that is high is considered to be a favourable margin by SVM.In Short, It's the distance between the hyperplane and the hyperplane's nearest observations (support vectors). SVM considers a high margin to be a favourable margin.

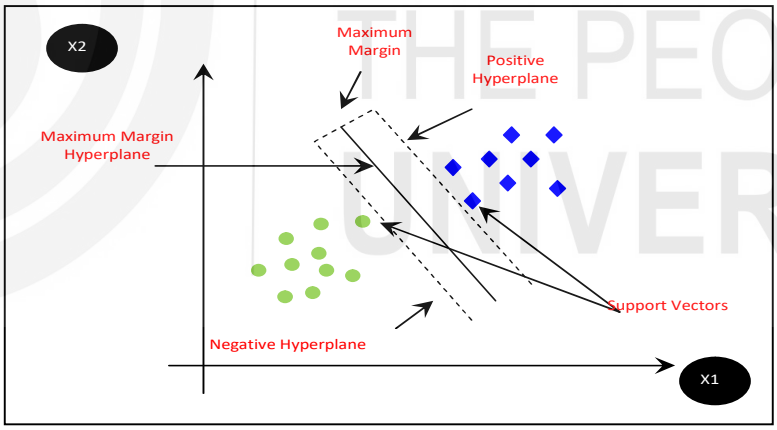


Fig2. Diagram of Support Vector Machine

Working of SVM

SVM is defined solely in terms of support vectors; we do not need to be concerned with any other observations because the margin is calculated based on the points that are support vectors that are closest to the hyperplane. This is in contrast to logistic regression, in which the classifier is defined over all of the points. Because of this, SVM is able to take use of some natural speedups.

To further understand how SVM operates, let's look at an example. Suppose we have a dataset that has two different classes (green and blue). It is necessary for us to decide whether the new data point should be classified as blue or green.

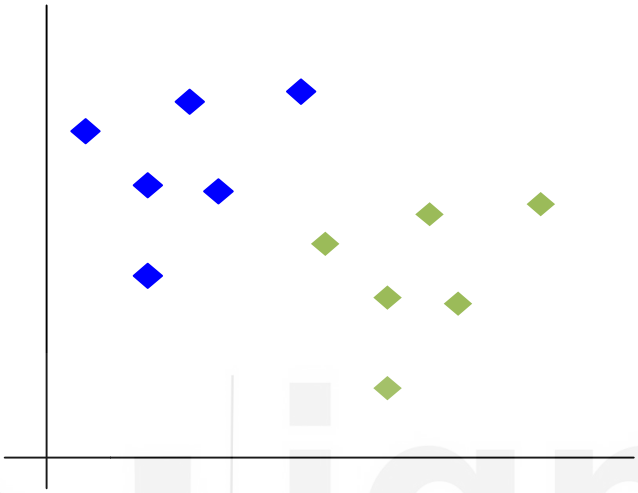


Fig 3. Dataset with two classes.

There are many ways to put these points into groups, but the question is which is the best and how do we find it?

NOTE: We call this decision boundary a "straight line" because we are plotting data points on a two-dimensional graph. If there are more dimensions, we call it a "hyperplane."

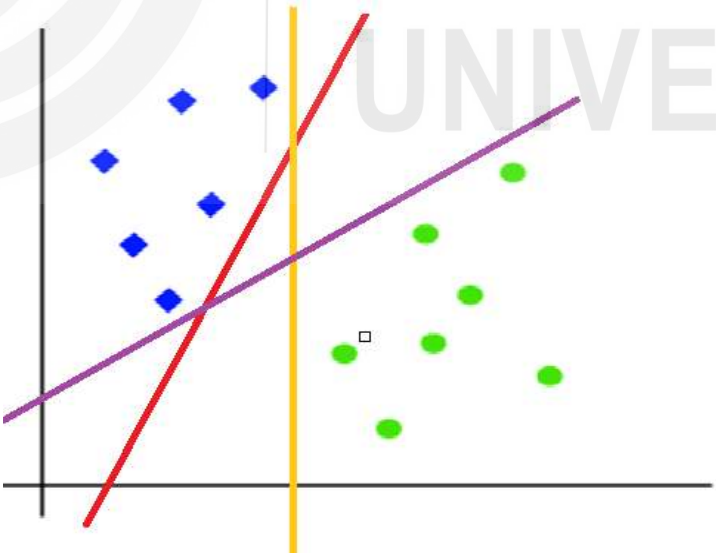


Fig4. Hyperplane

SVM is all about finding the hyperplane with the most space between the two classes, such hyperplane is the best hyperplane. This is done by finding many hyperplanes that best fit the labels and then picking the one that is farthest from the data points or has the biggest margin.i.e. The best hyperplane is the one with the greatest distance between the two classes, and this is what SVM is all about.

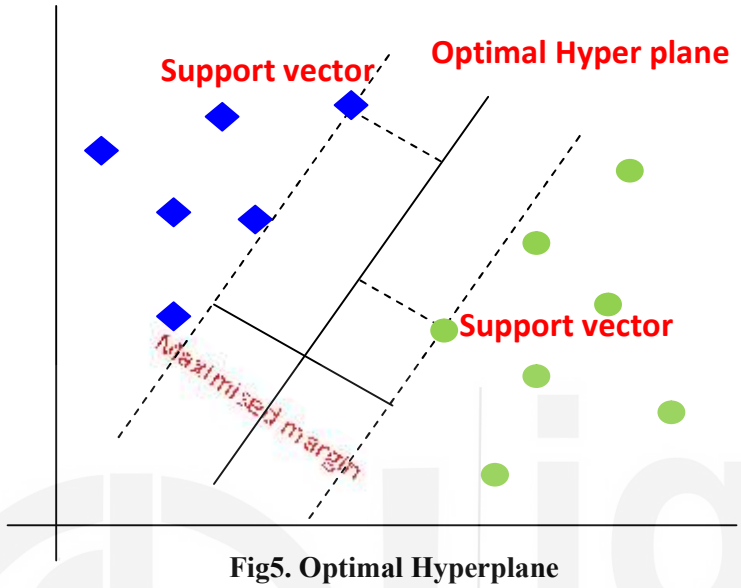


Fig5. Optimal Hyperplane

In geometry, a hyperplane is the name given to a subspace that is one dimension smaller than the ambient space. Despite the fact that this definition is accurate, it is not very clear. Instead of making use of it, we will concentrate on acquiring knowledge about lines in order to better understand what a hyperplane is. If you can recall the mathematics that you studied in high school, you presumably know that a line has an equation of the form, that the constant is called the slope, and that the y-axis is crossed by. If you can't remember those things, you should look them up. It is important to note, however, that the linear equation y = a x + b involves two variables. These variables are denoted by the letters y and x, but we are free to give them any name we like. This formula is valid for a wide range of possibilities for the value of, and we refer to the collection of those possibilities as a line.

Another notation for the equation of a line may be obtained if we define the two-dimensional vectors x=(x1,x2) and w=(a,-1) as follows: where w.x is the dot product of w and x.

$$\mathbf{w.x+b=0}$$

Now we need to locate a hyperplane : locating a hyperplane with the largest margin (a margin is a buffer zone around the hyperplane equation), and working toward having the largest margin while having the fewest points possible (known as support vectors).

"The goal is to maximise the minimum distance," to put it another way. for the sake of distance. If the point from the positive group is substituted in the hyperplane equation while generating predictions on the training data that was binary classified as positive and negative groups, we will get a value larger than 0. (zero), Mathematically, $w^T(\mathbf{\Phi(x)}) + \mathbf{b} > \mathbf{0}$ And predictions from the negative group in the hyperplane equation would give negative value as $w^T(\mathbf{\Phi(x)}) + \mathbf{b} < \mathbf{0}$. The indicators, on the other hand, were about

training data, which is how we're training our model. Give a positive sign for a positive class and a negative sign for a negative class.

However, if we properly predict a positive class (positive sign or greater than zero sign) as positive while testing this model on test data, then two positives equals positive and hence a greater than zero result. The same is true if we correctly forecast the negative group, because two negatives equal a positive.

However, if the model incorrectly identifies the positive group as a negative group, one plus and one minus equals a minus, resulting in a result that is less than zero. Thus summarising this we can say that The product of a predicted and actual label would be greater than 0 (zero) on correct prediction, otherwise less than zero.

$$y_n\left[w^T\phi(x) + b\right] = \begin{cases} \geq 0 \; if \; correct \\ < 0 \; if \; incorrect \end{cases}$$

CHECK YOUR PROGRESS-7

11. Suppose you are using a Linear SVM classifier with 2 class classification problem. Now you have been given the data in which some points are circled red that are representing support vectors. If you remove any one red points from the data. Does the decision boundary will change?

A) Yes
B) No

12. The effectiveness of an SVM depends upon:

A) Selection of Kernel
B) Kernel Parameters
C) Soft Margin Parameter C
D) All of the above

13. The SVM's are less effective when:

A) The data is linearly separable
B) The data is clean and ready to use
C) The data is noisy and contains overlapping points

10.11 SOLUTIONS/ANSWERS

□ Check Your Progress 1

1. Compare between Supervised and Un-Supervised Learning.

Solution : Refer to section 10.3

2. List the Steps Involved in Supervised Learning

Solution : Refer to section 10.3

3. What are the Common Issues Faced While Using Supervised Learning

Solution : Refer to section 10.3

□ Check Your Progress 2

4. Compare between Multi Class and Multi Label Classification

   Solution : Refer to section 10.4

5. Compare between structured and unstructured data

   Solution : Refer to section 10.4

6. Compare between Lazy learners and Eager Learners algorithms for machine learning.

   Solution : Refer to section 10.4

□ Check Your Progress 3

7. List the classification algorithms under the categories of Linear and Non-Linear Models. Also Discuss the various methods used for evaluating a classification model

   Solution : Refer to section 10.5

□ Check Your Progress 4

8. Using naive Bayesian classification, predict a class label. Given the training data in Table-1 below, we want to use naive Bayesian classification to predict the class label of a tuple. The characteristics age, income, student, and credit rating characterise the data tuples.

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Solution : Also Refer to section 10.5.1

The buys computer class label attribute has two unique values (yes and no). Let C1 represent the buys computer = yes class and C2 represent the buys computer = no class. The tuple we want to categorise is

X = (youthful age, medium income, student status, fair credit rating)

For i = 1, 2, we must maximise P(X|Ci)P(Ci). The prior probability for each class, P(Ci), can be calculated using the training tuples:

P(buys computer = yes) = 9/14 = 0.643
P(buys computer = no) = 5/14 = 0.357
To compute PX|Ci), for i = 1, 2, we compute the following conditional probabilities:
P(age = youth | buys computer = yes) = 2/9 = 0.222
P(age = youth | buys computer = no) = 3/5 = 0.600
P(income = medium | buys computer = yes) = 4/9 = 0.444
P(income = medium | buys computer = no) = 2/5 = 0.400
P(student = yes | buys computer = yes) = 6/9 = 0.667
P(student = yes | buys computer = no) = 1/5 = 0.200
P(credit rating = fair | buys computer = yes) = 6/9 = 0.667
P(credit rating = fair | buys computer = no) = 2/5 = 0.400
Using the above probabilities, we obtain

P(X|buys computer = yes) = P(age = youth | buys computer = yes) ×
P(income = medium | buys computer = yes) ×
P(student = yes | buys computer = yes) ×
P(credit rating = fair | buys computer = yes)
= 0.222×0.444×0.667×0.667 = 0.044.

Similarly, P(X|buys computer = no) = 0.600×0.400×0.200×0.400 = 0.019.

To find the class, Ci , that maximizes P(X|Ci)P(Ci), we compute
P(X|buys computer = yes)P(buys computer = yes) = 0.044×0.643 = 0.028
P(X|buys computer = no)P(buys computer = no) = 0.019×0.357 = 0.007

Therefore, the naïve Bayesian classifier predicts buys computer = yes for tuple X.

Check Your Progress 5

9. Apply KNN classification algorithm to the following data and predict value for (10,7) for K = 3

Feature 1	Feature 2	Class
1	1	A
2	3	A

2	4	A
5	3	A
8	6	B
8	8	B
9	6	B
11	7	B

Solution : Refer to section 10.5.2

□ Check Your Progress 6

10. Odds of drawing a heart. If you deal one card from a standard deck, the probability that the card is a heart **is 13/52 = 1**/4.

(a) Find the odds of drawing a heart.
(b) Find the odds of drawing a card that is not a heart.

Solution : Refer to section 10.5.4

□ Check Your Progress 7

11. Suppose you are using a Linear SVM classifier with 2 class classification problem. Now you have been given the data in which some points are circled red that are representing support vectors. If you remove any one red points from the data. Does the decision boundary will change?

A) Yes
B) No

Solution: A

12. The effectiveness of an SVM depends upon:

A) Selection of Kernel
B) Kernel Parameters
C) Soft Margin Parameter C
D) All of the above

Solution: D

The SVM effectiveness depends upon how you choose the basic 3 requirements mentioned above in such a way that it maximises your efficiency, reduces error and overfitting.

13. The SVM's are less effective when:

A) The data is linearly separable
B) The data is clean and ready to use
C) The data is noisy and contains overlapping points

Solution: C

When the data has noise and overlapping points, there is a problem in drawing a clear hyperplane without misclassifying.

10.12 FURTHER READINGS

1. Machine learning an algorithm perspective, Stephen Marsland, 2nd Edition, CRC Press,, 2015.
2. Machine Learning, Tom Mitchell, 1st Edition, McGraw- Hill, 1997.
3. Machine Learning: The Art and Science of Algorithms that Make Senseof Data, Peter Flach, 1st Edition, Cambridge University Press, 2012.

UNIT 11 REGRESSION

11.0 INTRODUCTION

In 1908 British biologist Francis Galton investigated the relationships between two variables to study the hereditary growth of children. In his research he categorised parents into two categories on the height: 1st category of the parents belongs to the family length smaller than average length of than parents' length and 2nd category of parents belong to the parents having lengththan the average length. This "regression toward mediocrity" gave these statistical methods thereprimarily the term regression describes the relationship between variables.

Simple regression y=m*x + C describes the relationship between one independent and one dependent variable Where theueuse variable y varies with the value of x and thus a dependent variable,the value of variable x affected any variable hence is a independent variable and m is having some constant value.

Consider the following the parent-children's set

Parent	64.5	65.5	66.5	67.5	68.5	69.5	70.5	71.5	72.5
Children	65.8	66.7	67.2	67.6	68.2	68.9	69.5	69.9	72.2

The mean height of the children is 68.44 whereas the mean height for the parents is 68.5.

The linear equation for the parents and children is

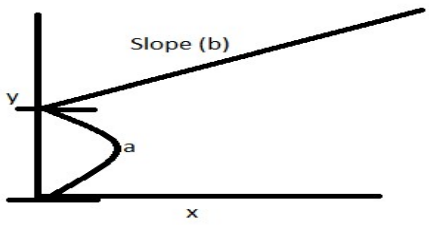
height_child = 21.52 + 0.69 * height_parent

Mathematically simple linear regression can be defined as y=bx+c+ϵ.Where b is the slope of the regression lin , x is the variable which can change the value of y but can't be affected by another variable. Whereas y is a variable which varies with a change in the n value of x. And the known as error value majors between the actual value and predicted value. Variable y is described ass dependent variable or response variable, and variable x is defined as an explanatory or predictor variable.

Regression is a supervised machine learning model which describes the relationships between the response variable and predictor variables.So, regression model is used when it is required to determine the value of one variable using another variable.

If the variable to be predicted is a single variable, then the regression equation will be y=a+bx

To determine the value of dependent variable y we need to determine the slope b and constant value and thus by substituting the different set of values for variable x we can get the different value of variabley.



when x=0 then y=a, which means when there is no independent variable then the predicted variable constant value.Suppose we are having multiple independent variables $x_1,x_2,x_3,x_4….,x_n$.Then the regressionequation will be $y=a+b_1x_1+b_2x_2+b_3x_3+…….box$.

The regression line is also calasthma best fit line because the regression line aims to fit all the points or will be minimum. Regression is a linear regression when there is one predictor variable, and we can apply a linear regression model. Themultiple linear regression model came into existence when the number of predictorsvaries are than then one in number. When the relationships between variable y and x are not linear,we can apply-linear regression model.

Following are the ways used by regression analysis to determine the relationships between the response variable and predictor variables:

- **Find the relationship**: It is required to determine the relationships between the predictor variable and response variables. If any change in the independent variable will result inin the age of dependent variable there is an exitstance of relationship.
- **Strength of relationships**: By changing the value of one variable how much another variable changedetermine the strength of relationships.
- **Formation of relationships**: If a change in the value of the dependent variable will result in a change in independent variable, then formulate a mathematical equation to represent the relationships between both variables.
- **Prediction**: After formulation of the mathematical equation find the predicted value.
- **Another independent variable:**Another independent variable which is having impact on dependent variable. If there exist, then formulate the mathematical equation using these variables also.

Uses of Regression

- In a business scenario when it is required to determine the impact of different-different independent variables to find the target value regression can be used.
- When we want to represent in a mathematical expressionform, or we want to model a problem to determine the impact of different variables.

- It is very easy to explain about the business logic with the help of regression. Business logics can be explained very easily to the person.
- When the target variable is normally distributed having some characteristics, regression is very effective.

Examples of Regression

Relationship between uploading a picture on Facebook page and number of likes by the friends.

Relationship between the height of the child and their parents' heights.

Relationship between the average food intake and weight gain.

Relationship between the numbers of hours studied and marks scored by the students.

Relationship between the product consumption by increasing the product price.

Terminologies used in Regression Analysis

- **Dependent Variable:** A variable used to predict the output. It is also called as the target variable.
- **Independent Variable:** The variable which is having an impact on dependent variable is called independent variable. There may be one or more independent variable. This variable is also called as predictor variable. For example, salary of an employee depends on age,qualification,experience. Here salary is a dependent variable and age,qualification and experience are independent variables.
- **Outlier:**Outlier is a value which effect out output, very high value or very low value will affect the result. In case of regression first we have to remove the outlier first.
- **Multicollinearity:**If two values in our dataset are corelated to each other than other variables, such a condition is called multicollinearity. Example: age and date of birth of are correlated to each other. So, we have to avoid one of them.
- **Underfitting and Overfitting:**Overfitting results when our machine learning model work well with the training data set but it does not work well with test data set. Underfitting results when our dataset does not perform well even with our training data set.

11.1 OBJECTIVES

After completing this unit you will be able to:

- Understand the Regression Algorithm
- Understand and apply Linear Regresssion
- Understand and apply Polynomial Regression
- Understand and apply Support Vector Regression

11.2 REGRESSION ALGORITHM

Following are various types of regression algorithms.

Linear regression: Linear regression algorithm comes into existence when there is only one dependent variable and independent variables can be be one or more in numbers. If there is a single independent variable, then it is called as simple linear regression. In linear regression the relationships between the dependent and independent variables are linear i.e. of type $y_i=a+b*x_i$; where y_i is a dependent variable and x_i is a independent variable. Variable b is the slope of the line and a is intercept with the axis. Example child height=a+b*(parent height)

Multiple Linear Regression: When there is only one dependent variable and more than one independent variables, then it results in multiple linear regression i.e. $y=a+bx_1+cx_2+dx_3$. ; example weight = a+b * (daily meal)+ c* (daily exercise)

Logistic regression: In logistic regression algorithm dependent variable is binary in nature (False/True). This algorithm is generally used under cases like testing of the medicines, to detect the bank fraud etc.We had already discussed the concept of logistic regression in unit no. 10 of this course.

Polynomial regression: Polynomial regression is described with the help of polynomial equation where the occurrence of independent variable is more than one. There is no linear relationships between the dependent and independent variables. It results in a curved line instead of a straight line i.e. $y=c+a*x+b*x^2$

Ecologic regression: Ecological regression algorithm is used when group data belongs to a group. Thus, data is divided into different groups and regression is performed on different groups. Ecologocal regression is mostly used in political research eg.party_votes %=.2+.5*(below_poverty_people_votes)

Ridge regression: It is a type of regularization. When data variables are highly correlated ridge regression is used. Using some constraints on regression coefficients, it is used to reduce the error and lower the bias. Mostly used in feature selection.

Lasso regression: Least absolute shrinkage and selection operator regression algorithm a penalty is assign the coefficients. Lasso regression uses shrinkage technique where data values are shrunk towards a mean.

Logic regression: In logic regression predictor variable and response variable both are binary in nature and applicable to both classification and regression problem.

Bayesian regression: Bayesian regression algorithm is based on Bayesian statistics. Random variables are used as a parameter to estimates. In this algorithm if the data is absent then some prior data is taken as an input.

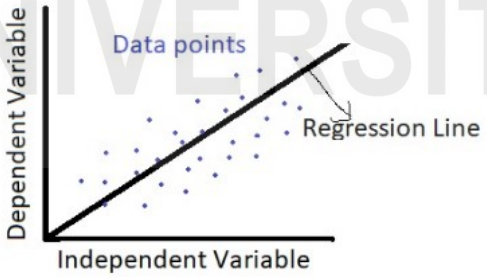
Quantile regression: This is used when the boundary of the quantile is of interest. Whenoverweight and underweight is considered for the health analysis it is consider as an quantile regression.

Cox regression: Cox regression algorithm is used when output of a variable depends on set of independent variables example patient_survival_after_surgery(Survived,Died)=(age,condition,BMI)

11.3 LINEAR REGRESSION

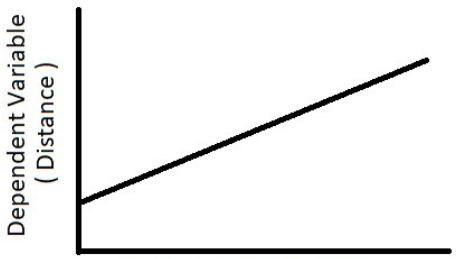
Linear regression is a mathematical method implemented where we want to find the response variable and predictor variables. When the relationships are linear then it is called as linear regression or otherwise it is called as a nonlinear regression. Linear regression makes prediction for continuous/real or numerical variables like age,salary,price etc.

As shown in figure x-axis represent independent variable and y-axis represent dependent variable. A Line with some slope is called linear regression line which shows the relationship between the independent and dependents variable and dots represents the point of the data sets, where some points lie on the line and some other points lie above and below of the line.

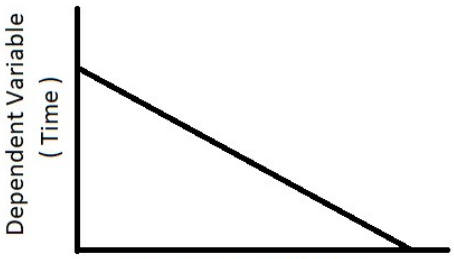


If DV is the dependent variable and IV is the independent variable, then the Positive Linear relationship results with the increases in dependent variable (DV)on the y-axis with respect to increase in value of independent variable (IV) on x-axis. For example, the distance traversed by the car increases when the speed of the car increases. Thus, the distance traversed by the car depends on the speed of the car.

And, the Negative Linear relationships result with the decrease in dependent variable (DV) on the y-axis with respect to the increase in independent variable (IV) on x-axis. For example, time taken by the car decreases with the increase in speed of car.



Positive Linear Relationship



Negative Linear Relationship

Now consider the following data points:

$(x_i,y_i)=\{(45,75),(48,80),(51,100),(37,70)\}$ for i =1,2,3 and 4.

$(x_1,y_1)=(20,41)$

$(x_2,y_2)=(30,83)$

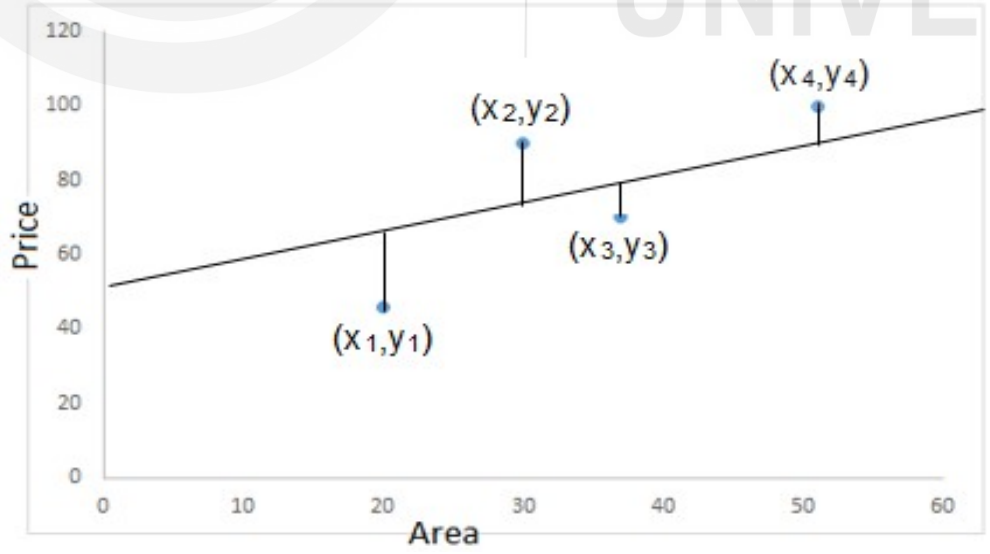
$(x_3,y_3)=(38,62)$

$(x_4,y_4)=(52,100)$

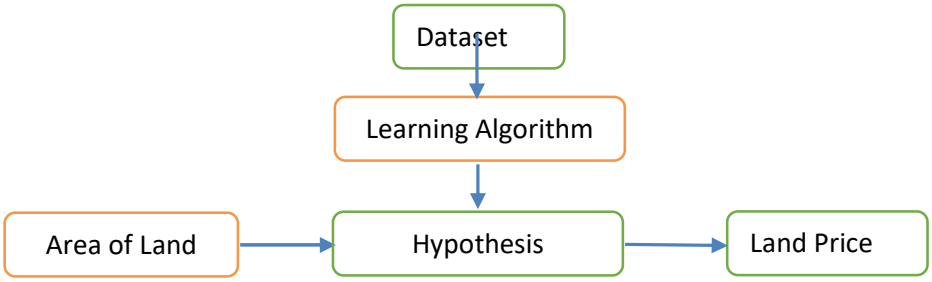
Where x=area given in meter square

And y= price of the land

Following graph draw the linear relationship between x and y



As shown in below diagram set of data is given as a input and learning algorithm will generate a output function conventionally known as a hypothesis (h). The role of the hypothesis function is to estimate the price by taking area as a input to the function. Mapping function h will map from area of land to price of land.



$h(y)=\Theta_0+\Theta_i x$, where h is a hypothesis of mapping from x to y.

we assume every point is described by our line on xy plane.

Total error=$\sum|\hat{y}^{(i)} - y^{(i)}|$

Where $\hat{y}^{(i)}$ is assumed data point and $y^{(i)}$ is the actual data point.

Average error=$\frac{1}{n}\sum_{i=1}^{n}|\hat{y}^{(i)} - y^{(i)}|$

But as we no error function is not differentiable for $-\infty<x<\infty$

So loss function will be

$J(\Theta)=\frac{1}{n}\sum_{i}^{n}|\hat{y}^{(i)} - y^{(i)}|^2$

$\hat{y}^{(i)}=h_\Theta(x^{(i)})=\Theta_1 x^{(i)}+\Theta_0$.

Now it is required to minimize our loss function($J(\Theta)$). A Gradient Descent approach will be used to minimize the loss function

Linear regression using least square method

Mathematical function is used to find the sum of squares (square of the distance of the points and the regression line) of all the data points. Least square method is a statistical method given by Carl Friedrich Gauss used to determine the best fit line or the regression line by minimizing the sum of squares. Least square method is used to find the line having minimum value of the sum of squares and this line is the best-fit regression line.

Regression line is y=m*x+c where

y= Estimated or predicted value (Dependent Variable)

x= Value of x for observation (Independent variable)

c= Intercept with the y-axis.

m= Slope of the line

Example :1

Consider the following set of data points (x,y), find the regression line for the given data points.

X	1	2	3	4	5
Y	3	4	2	4	5

Solution:

X	y	$(x - \bar{x})$	$(y - \bar{y})$	$(x - \bar{x})^2$	$(x - \bar{x})(y - \bar{y})$
1	3	-2	-0.6	4	1.2
2	4	-1	0.4	1	-0.4
3	2	0	-1.6	0	0
4	4	1	0.4	1	0.4
5	5	2	1.4	4	2.8
3	3.6	0	0	10	4

where $m=\frac{\sum(x-\bar{x})(y-\bar{y})}{\sum(x-\bar{x})^2}$

m= 4/10= 0.4

$\bar{x}$=mean of x=3

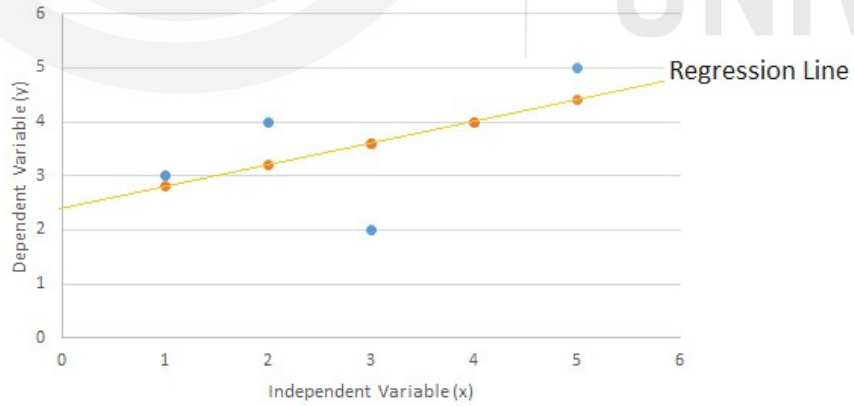
$\bar{y}$=mean of y=3.6

y=mx+c

m= 0.4

c=2.4

y=.4x+2.4



In the above figure blue points are the actual points and yellow points are the predicted points using least square method. Some points represented by blue color lie above the line while some other blue color points lie below the line. However some points represented by the yellow color lie on the line. All other points not lying on the line are the far away from the line with some distance. Thus, actual blue data

points and the predicted yellow data points contain some distance between them. This distance or difference between the data points represent an error.

Cost function is used to find the distance between the actual data point value lying other than the regression line and the predicted value of data points lying on the regression line. Cost function optimizes the regression coefficient or weights. It measures how a linear regression model is performing.
Difference between the actual value y on y-axis and predicted value $\hat{y}$ is (y-$\hat{y}$), and cost=(y-$\hat{y}$)^2
if there are n number of data points then the cost function will be

$$\text{cost} = \frac{1}{2n}\sum_{i=1}^{n}(y - \hat{y})$$

or

$$\text{cost} = \frac{1}{n}\sum|(y - \hat{y})|$$

Since cost function provide the error between the actual value and predicted value so minimizing the value of cost function will improve the prediction value. Higher the cost function value will degrade the performance.

Mean Squared Error (MSE): The average of squared of the distance measured between the actual data points lying other than the line and predicted data points lying on the line is called as a mean squared error. It is written as:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{n}\left(y_i - (a_1 x_i + a_0)\right)^2$$

Where N = total number of data points

y_i = Actual value

$(a_1 x_i + a_0)$ = Predicted value with slope a_1 and intercept a_0

Mean Absolute Error (MAE) is used to determine by calculation sum of all errors divided by the total number of errors in a group of predictions. While considering a group of data points their directions are not important. In other words, it is a mean of absolute differences among actual value and response value results where all individual deviations have even importance.

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{m}\left|\hat{y}^{(i)} - y^{(i)}\right|$$

Check your progress 1

1. What is regression? Define linear regression.

2. Describe about overfitting and underfitting.

-

3. State True or False

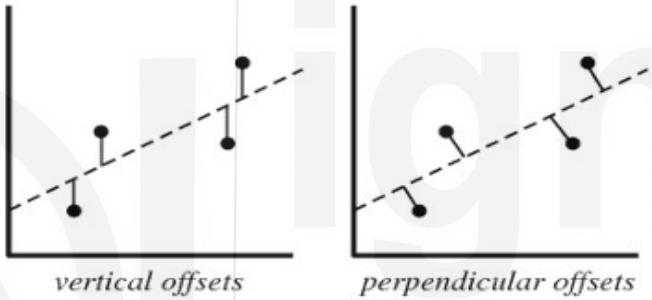
T	F

(a)To determine relationship between numeric variables Linear Regression is used.

(b) With the help of logistic regression, 0/1value attributes are predicted.

(c) In Linear Regression, Least Square Error is used to find the line fitted best.

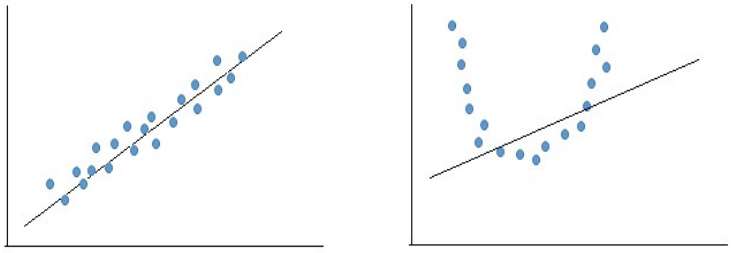
(d) If x-axis represent independent variable and y-axis represent dependent variable. Then vertical offset diagram shown below is used for least square line fit .



11.4 POLYNOMIAL REGRESSION ALGORITHM

Linear model can apply to data set having linear in nature, however if we have data set of nonlinear in nature then nonlinear model is to be applied.

As shown in figure all the data points are linear in nature. All points are close to the line, linear model regression model can be applied to the data sets. In figure 2 all the data points are nonlinear in nature so linear model cannot fit all the data points, only 2 or3 data points can be fitted to the linear model and all other points are far away from the line. Loss value for this graph will be very high and accuracy will be reduced.



$y_1=a_0+bx$ is equation of linear regression, with slope b where a_0 is the intercept with the x axis.

$$y_2 = a_0 + b_1x_1 + b_2x_2 + b_3x_3 \ldots\ldots\ldots\ldots b_nx_n = a_0 + \sum_{i=1}^{n}(b_ix_i)$$

where y_2 is a multiple regression equation with n independent variables.

Above two equations y_1 and y_2 are polynomial equations with degree 1.

Consider stock price S_p as a polynomial function of time.

$$S_p = a_0 + a_1T^1 + a_2T^2 + a_3T^3 + \varepsilon$$

Where S_p is a polynomial function and ε is an error and we need to find different values of a_0, a_1, a_2 *and* a_3 such that the difference between the value obtained from the above equation and from the model will be minimum.

Now we have data points (T_1,S_{p_1}), (T_2,S_{p_2}), (T_3,S_{p_3})……. (T_n,S_{p_n}).

Thus, the equation become

$$y_i = a_0 + a_1u_{i+}a_2v_{i+}a_3w_{i+}\varepsilon_i$$

Where $y_i = S_{p_i}$, $u_{i=}\mathrm{T}_i$, $v_{i=}\mathrm{T}_i^2$, $w_{i=}\mathrm{T}_i^3$

$$Y=\begin{bmatrix} S_{p_1} \\ S_{p_2} \\ . \\ . \\ . \\ S_{p_n} \end{bmatrix} \qquad X=\begin{bmatrix} 1 & \mathrm{T}_1 & \mathrm{T}_1^2 & \mathrm{T}_1^3 \\ 1 & \mathrm{T}_2 & \mathrm{T}_2^2 & \mathrm{T}_2^3 \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ 1 & \mathrm{T}_N & \mathrm{T}_N^2 & \mathrm{T}_N^3 \end{bmatrix}$$

Extending to the mth order polynomial it becomes

$$X=\begin{bmatrix} 1 & \mathrm{T}_1 & \mathrm{T}_1^2 & \mathrm{T}_1^m \\ 1 & \mathrm{T}_2 & \mathrm{T}_2^2 & \mathrm{T}_2^m \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ 1 & \mathrm{T}_N & \mathrm{T}_N^2 & \mathrm{T}_N^m \end{bmatrix}$$

For all m<<N , and

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = (\mathrm{X}^{\mathrm{T}}\mathrm{X})^{-1}\mathrm{X}^{\mathrm{T}}\mathrm{Y}$$ known as left inverse of matrix X

Squared Error (SE) is the error occurred between the predicted values and actual values used for polynomial regression line. It is written as:

$$S_r = \sum_{i=1}^{n} \left(y_i - \left(a_0 + a_1 x_i + a_2 x_i^2\right)\right)^2$$

Where n = total number of data points

y_i = Actual value

$\left(a_0 + a_1 x_i + a_2 x_i^2\right)$ = Predicted value

$$S_r = \sum_{i=1}^{n} \left(y_i - a_0 - a_1 x_i - a_2 x_i^2\right)^2 \qquad \ldots \text{(i)}$$

To minimize the error $\frac{\delta S_r}{\delta a_0} = 0, \frac{\delta S_r}{\delta a_1} = 0 \; and \; \frac{\delta S_r}{\delta a_2} = 0$ …(ii)

On solving equation (i) we will get

$$\frac{\delta S_r}{\delta a_0} = -2 \sum_{i=1}^{n} y_i - a_0 - a_1 x_i - a_2 x_i^2$$

Since $\frac{\delta S_r}{\delta a_0} = 0$

$$\Rightarrow -2 \sum_{i=1}^{n} y_i + \sum_{i=1}^{n} 2a_0 + \sum_{i=1}^{n} 2a_1 \, x_i + \sum_{i=1}^{n} 2a_2 \, x_i^2 = 0$$

$$n a_0 + a_1 \sum_{i=1}^{n} x_i + a_2 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} y_i \qquad \ldots . (iii)$$

Find $\frac{\delta S_r}{\delta a_1}$, on solving equation (i)

$$\frac{\delta S_r}{\delta a_1} = -2 x_i \sum_{i=1}^{n} y_i - a_0 - a_1 x_i - a_2 x_i^2$$

Since $\frac{\delta S_r}{\delta a_1} = 0$

$$\Rightarrow -2 \sum_{i=1}^{n} x_i y_i + 2a_0 \sum_{i=1}^{n} x_i + 2a_1 \sum_{i=1}^{n} x_i^2 + 2a_2 \sum_{i=1}^{n} x_i^3 = 0$$

$$\Rightarrow a_0 \sum_{i=1}^{n} x_i + a_1 \sum_{i=1}^{n} x_i^2 + a_2 \sum_{i=1}^{n} x_i^3 = \sum_{i=1}^{n} x_i y_i \qquad \ldots . (iv)$$

Find $\frac{\delta S_r}{\delta a_2}$, on solving equation (i)

$$\frac{\delta S_r}{\delta a_2} = -2 x_i^2 \sum_{i=1}^{n} y_i - a_0 - a_1 x_i - a_2 x_i^2$$

Since $\frac{\delta S_r}{\delta a_2} = 0$

$$\Rightarrow -2\sum_{i=1}^{n} x_i^2 y_i + 2a_0\sum_{i=1}^{n} x_i^2 + 2a_1\sum_{i=1}^{n} x_i^3 + 2a_2\sum_{i=1}^{n} x_i^4 = 0$$

$$\Rightarrow a_0\sum_{i=1}^{n} x_i^2 + a_1\sum_{i=1}^{n} x_i^3 + a_2\sum_{i=1}^{n} x_i^4 = \sum_{i=1}^{n} x_i^2 y_i \qquad \ldots.(v)$$

From equation (iii),(iv) and (v)

$$\begin{bmatrix} n & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^3 \end{bmatrix} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \sum x_i^2 y_i \end{bmatrix}$$

where value of i varies from 1 to n.

Example 2. Consider the following set of data points (x,y). Find the 2nd order polynomial $y=a_0 + a_1 x_i + a_2 x_i^2$, and using polynomial regression determine the value of y when x is 40.

X	40	10	-20	-88	-150	-170
Y	5.89	5.99	5.98	5.54	4.3	3.33

Solution. From the given data points (x,y):

x_i	y_i	x_i^2	x_i^3	x_i^4	$x_i y_i$	$x_i^2 y_i$
40	5.89	1600	64000	2560000	235.6	9424
10	5.99	100	1000	10000	59.9	599
-20	5.98	400	-8000	160000	-119.6	2392
-88	5.54	7744	-681472	59969536	-487.52	42901.8
-150	4.3	22500	-3375000	506250000	-645	96750
-170	3.33	28900	-4913000	835210000	-566.1	96237
$\sum_{i=1}^{n} x_i$ = -378	$\sum_{i=1}^{n} y_i$ = 31.03	$\sum_{i=1}^{n} x_i^2$ = 61244	$\sum_{i=1}^{n} x_i^3$ = -8912472	$\sum_{i=1}^{n} x_i^4$ = 1404159536	$\sum_{i=1}^{n} x_i y_i$ = -1522.7	$\sum_{i=1}^{n} x_i^2 y_i$ = 248304

$$\begin{bmatrix} 6 & -378 & 61244 \\ -378 & 61244 & -8912472 \\ 61244 & -8912472 & 1404159536 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 31.03 \\ -1522.7 \\ 248304 \end{bmatrix}$$

By solving above matrix the value a_0,a_1 and a_2 will be

a_0=6.07647

a_1=-0.00253987

a_2=-0.000104319

y= $6.07647 - 0.00253987\ x - 0.000104319\ x^2$

y(50)= 6.07647 − 0.00253987 X 50 - 0.000104319 X 2500

= 5.68

Check your progress 2

1. Define polynomial regression.

   ----------Write down the general equation for the polynomial curve fitting.

   ----

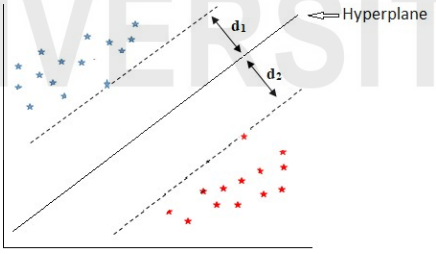
2. State True or False

   (a) A quadratic regression equation can be represented by $\hat{y} = b_0 + b_1x_1 + b_2x_2^2$, where $x_1$and x_2 are independent variables and y is one dependent variable.

   (b) Height of regression line is used to determine the intercept in multiple regression.

   (c) Multiple regression is used when dependent variable does not depend on more than one independent variable.

11.5 SUPPORT VECTOR REGRESSION



Support vector machine is used in solving both classification and regression problem. Consider a classification problem having two different categories as sown in figure.It is easy to separate these two categories by using a line between the two. There is a hyperplane between these two categories which will separate these two from each other. This hyperplane is used to divide the points into different categories lying opposite to the line. Other than hyperplane there are two marginal lines opposite to the hyperplane at a distance apart from the hyperplane. These two marginal lines are having a certain distance from the hyperplane so that all the points can be easily categorised.

Parallel to the hyperplane there are two parallel lines at a marginal distance from the hyperplane. Thus, we can say that there are three hyperplane ie., two line at a marginal distance are also hyperplane. These two marginal hyperplanes must pass through at least one of the closest datapoints. These data points are called **support vectors**. There can be more than one support vectors passes through the marginal hyperplane. These support vectors determine the marginal distance of these two lines from hyperplane

(ie., d_1 and d_2). There can be more than one marginal hyperplane for the given data set.We have to choose the marginal hyperplane so that the distance d_1 and d_2 will be the maximum distance $\max(d_1+d_2)$.

Considering the data points of the given graph of Fig a. It is not possible to divide the points into two categories by using a linear hyperplane. Thus, we need to convert this graph into three-dimensional graph. The SVM kernel convert the two-dimensional data points in 3- or 4-dimensional data points as shown in Fig b, the hyperplane divide the data points of three dimension into two separate categories.

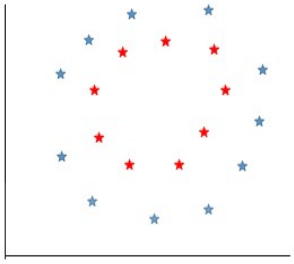


Fig a

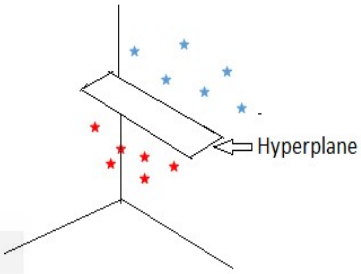


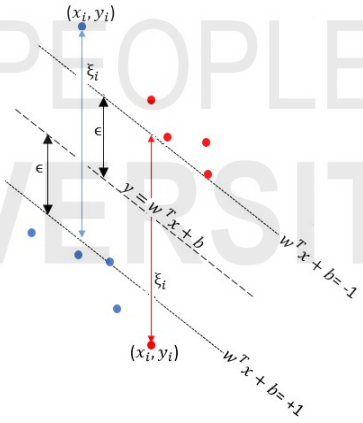
Fig b

Consider the following graph

As shown in figure equation of hyperplane is $y=w^Tx+b$, where b is constant having value zero since line is passes through the coordinate (0,0) and the slope of line m is -1.

$y=w^Tx$ (since b=0)

Any point that lies below the hyperplane (w^Tx) contribute to the positive value of x, and so is an example of the blue data points, while any data point that lie above the hyperplane (w^Tx) contribute to the negative value of x, and so is an example of the red data points.



For a given margin value M we can say for any value of x where $w^Tx \geq M$ lies on blue points, and for any value of x where $w^Tx \leq -M$ lies on red points. Now consider a point x_i^+ that lies at the positive margin of the hyperplane then $w^Tx_i^+=M$. Here x_i^+ is a support vector. On travelling to the opposite direction perpendicular to the positive margine we will reach a point closest to the negative margin of the hyperplane called as x_i^-.

If x_1 and x_2 are two negative and positive regression vector lies on marginal lines $w^Tx+b=-1$ and $w^Tx+b=+1$, the distance between marginal lines can be determined by

$$w^T(x_2 - x_1)=2$$

$$\frac{w^T(x_2-x_1)}{||w||} = \frac{2}{||w||}$$

We have to maximize $\frac{2}{||w||}$ subject to $w^T(x_2 - x_1) \geq +1$ and maximize $\frac{2}{||w||}$ subject to $w^T(x_2 - x_1) \leq -1$ for all i=1, 2…...n. In generalized form, $y_i * w^T x_i + b_i \geq +1$, and we need to minimize $\frac{||w||}{2}$ (reciprocal of $\frac{2}{||w||}$).

If all the data points are classified by the marginal line, then it will overfit the machine. And this is not happening in real scenario. It is not always possible that all the data point lies on the right side of the classification. As shown in figure one of the red data points lie below positive margin and one of the blue data points lie above negative margin of the hyperplane. These two data points lie in opposite plane area. If ξis the distance of the data point from respective marginal line, we need to find out the error $\xi_i$for such points.

$$y_i - (w^T x_i + b_i) \leq \epsilon + \xi_i \text{ for each } \xi_i \geq 0$$

$$(w^T x_i + b_i) - y_i \leq \epsilon + \xi_i$$

Error computed $= C_i \sum_{i=1}^{n} \xi_i$ where C_i is the number of error and ξ_i is the error value

Thus it is required to minimize $(w^*, b^*) = \frac{2}{||w||} + C_i \sum_{i=1}^{n} \xi_i$

Where * represent the optimal value.

Example 3. For the given points of two classes red and blue:

Blue: { (1,1), (2,1), (1,-1), (2,-1)}

Red : { (4,0), (5,1), (5,-1), (6,0)}

Ploat a graph for the red and blue categories. Find the support vectors and optimal separating line.

Solution.

Now first support vector SV_1 with x-coordinate 2 and y-coordinate 1 is represented by

$SV_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$

Similarly support vector $SV_2$with x-coordinate 2 and y-coordinate -1 and $SV_3$with x-coordinate 4 and y-coordinate 0 will be represented by

$SV_2 = \begin{pmatrix} 2 \\ -1 \end{pmatrix}$ and $SV_3 = \begin{pmatrix} 4 \\ 0 \end{pmatrix}$

Adding 1 as a input bias in support vector SV_1, SV_2 and SV_3

$\overline{SV_1} = \begin{pmatrix}2\\1\\1\end{pmatrix}$, $\overline{SV_2} = \begin{pmatrix}2\\-1\\1\end{pmatrix}$, and $\overline{SV_3} = \begin{pmatrix}4\\0\\1\end{pmatrix}$

To determine the value of α_1, $\alpha_2$and α_3 form the given linear equations we will assume that the support vector SV_1, SV_2 belong to the negative class and support vector SV_3belongs to the positive class.

$\alpha_1\overline{SV_1SV_1} + \alpha_2\overline{SV_1SV_2} + \alpha_3\overline{SV_1SV_3}$ = -1 (-ve class)

$\alpha_1\overline{SV_1SV_2} + \alpha_2\overline{SV_2SV_2} + \alpha_3\overline{SV_2SV_3}$ = -1 (-ve class)

$\alpha_1\overline{SV_1SV_3} + \alpha_2\overline{SV_2SV_3} + \alpha_3\overline{SV_3SV_3}$ = +1 (+ve class)

$\alpha_1\begin{pmatrix}2\\1\\1\end{pmatrix}\begin{pmatrix}2\\1\\1\end{pmatrix} + \alpha_2\begin{pmatrix}2\\1\\1\end{pmatrix}\begin{pmatrix}2\\-1\\1\end{pmatrix} + \alpha_3\begin{pmatrix}2\\1\\1\end{pmatrix}\begin{pmatrix}4\\0\\1\end{pmatrix}$ = -1

$\alpha_1\begin{pmatrix}2\\1\\1\end{pmatrix}\begin{pmatrix}2\\-1\\1\end{pmatrix} + \alpha_2\begin{pmatrix}2\\-1\\1\end{pmatrix}\begin{pmatrix}2\\-1\\1\end{pmatrix} + \alpha_3\begin{pmatrix}2\\-1\\1\end{pmatrix}\begin{pmatrix}4\\0\\1\end{pmatrix}$ = -1

$\alpha_1\begin{pmatrix}2\\1\\1\end{pmatrix}\begin{pmatrix}4\\0\\1\end{pmatrix} + \alpha_2\begin{pmatrix}2\\-1\\1\end{pmatrix}\begin{pmatrix}4\\0\\1\end{pmatrix} + \alpha_3\begin{pmatrix}4\\0\\1\end{pmatrix}\begin{pmatrix}4\\0\\1\end{pmatrix}$ = +1

After simplification of above three equations, we get

$6\alpha_1 + 2\alpha_2 + 9\alpha_3 = -1$

$4\alpha_1 + 6\alpha_2 + 9\alpha_3 = -1$

$9\alpha_1 + 9\alpha_2 + 17\alpha_3 = 1$

After simplification of above three equations, we get

$\alpha_1 = \alpha_2 = -3.25$ and $\alpha_3 = 3.5$

The hyperplane that discriminates the positive class from the negative class is given by

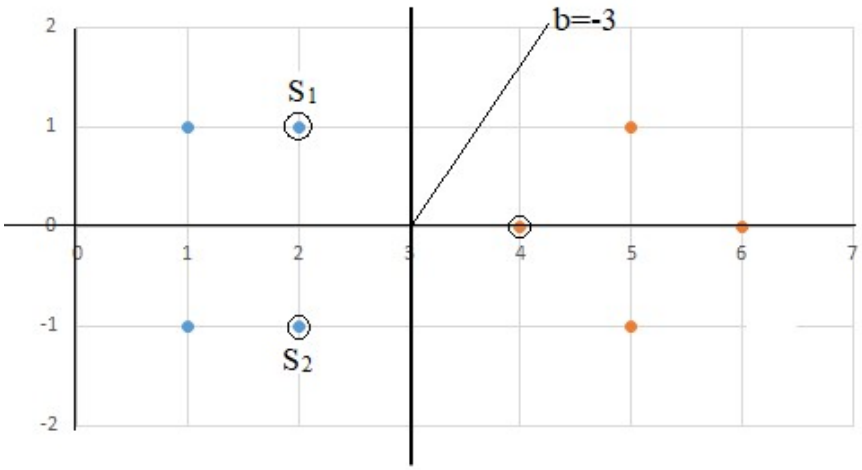
$\overline{w} = \sum_i \alpha_i \overline{SV_i}$

$\overline{w} = \alpha_1\begin{pmatrix}2\\1\\1\end{pmatrix} + \alpha_2\begin{pmatrix}2\\-1\\1\end{pmatrix} + \alpha_3\begin{pmatrix}4\\0\\1\end{pmatrix}$

$\overline{w} = (-3.25) * \begin{pmatrix}2\\1\\1\end{pmatrix} + (-3.25) * \begin{pmatrix}2\\-1\\1\end{pmatrix} + (3.5) * \begin{pmatrix}4\\0\\1\end{pmatrix} = \begin{pmatrix}1\\0\\-3\end{pmatrix}$

Hyperplane equation is y=wx+b

Where w= $\begin{pmatrix}2\\0\end{pmatrix}$ and b=-3 or b+3=0 is a line parallel to y-axis which separate both of the category red and blue.



Applications of Support Vector Regression

Used to solve supervised regression problems.

Can be used in both linear and non linear type of data.

Prediction of fire in forest during weather changes.

Prediction of electric power demand.

Check Your Progress 3

1. Define hyperplane.

   ---

2. Explain about support vector.

   ---

3. With the given set of points in two classes:

   Class A: $\binom{0}{1}\binom{0}{-1}\binom{1}{0}\binom{-1}{0}$

   Class B: $\binom{3}{1}\binom{6}{1}\binom{3}{-1}\binom{6}{-1}$

   Plot these two classes and find the line separating these two classes. Determine the margin and support vector of the two classes

11.6 SUMMARY

In this unit, we discussed about the concepts of regression – linear regression and nonlinear regression. We discussed about how to find relationship between response variable and predictor variable. Various terminologies used in regression are discussed with an example. Concept of dependent variable, independent variables and how to find the relationships are defined in this unit. Different types of regression also discussed in this unit.

This unit also focused on polynomial regression and how to plot a polynomial curve is also discussed. Concepts of overfitting and underfitting are also discussed in this unit.

In this unit support vector regression algorithm is discussed. Concept of hyperplane, marginal hyperplane and marginal distance are discussed with an example.

11.7 SOLUTIONS / ANSWERS

Check Your Progress 1

1. Regression is a supervised machine learning model which describes the relationships between response variable and predictor variables.So, regression model is used when it is required to determine the value of one variable using another variable.
   Mathematically simple linear regression can be defined as $y=bx+c+\epsilon$. Where b is the slope of the regression line, x is the variable which can change the value of y but can't be affected by another variable. Whereas y is a variable which varies with change in value of x. And ϵ is the known as an error value majored between the actual value and predicted value.
2. Overfitted results when it is unable to generalize well to new data. It results in high performance on trading data. Whereas underfitting results poor performance on training dataset
3. a. T
   b. T
   c. T
   d. T

Check Your Progress 2

1. Polynomial regression is a type of regression algorithm in which specifies the relationships between independent and dependent variable. But here the independent variables are of n^{th} degree polynomial.

2. General equation for the polynomial curve fitting

$$y = m_1x^1 + m_2x^2 + m_3x^3 + \dots\dots\dots + m_nx^n$$

$$y = \sum_{i=1}^{1=D} m_i x^i + C$$

3. a. T
   b. T
   c. F

Check Your Progress 3

1. Hyperplane is the line which categorise the data points into two categories.
2. Data points lying on two marginal hyperplanes are called as support vectors.
3. Now first support vector SV_1 with x-coordinate 1 and y-coordinate 0 is represented by

   $SV_1=\begin{pmatrix}1\\0\end{pmatrix}$

   Similarly support vector S_2 and S_3 will be represented by

   $SV_2=\begin{pmatrix}3\\1\end{pmatrix}$ and $SV_3=\begin{pmatrix}3\\-1\end{pmatrix}$

   Adding 1 as a input bias in support vector S_1, S_2 and S_3

   $\overline{SV_1} = \begin{pmatrix}1\\0\\1\end{pmatrix}$, $\overline{SV_2} = \begin{pmatrix}3\\1\\1\end{pmatrix}$, and $\overline{SV_3} = \begin{pmatrix}3\\-1\\1\end{pmatrix}$

   we need to find out 3 parameters α_1, $\alpha_2$and α_3 form the given linear equations by assuming that support vector S_1 ,S_2 belong to the negative class and support vector S_3belongs to the positive class.

   $\alpha_1\overline{SV_1SV_1} + \alpha_2\overline{SV_1SV_2} + \alpha_3\overline{SV_1SV_3}$ = -1 (Negative class)

   $\alpha_1\overline{SV_1SV_2} + \alpha_2\overline{SV_2SV_2} + \alpha_3\overline{SV_2SV_3}$ = +1 (Positive class)

   $\alpha_1\overline{SV_1SV_3} + \alpha_2\overline{SV_2SV_3} + \alpha_3\overline{SV_3SV_3}$ = +1 (Positive class)

   $$\alpha_1\begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix} + \alpha_2\begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}3\\1\\1\end{pmatrix} + \alpha_3\begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}3\\-1\\1\end{pmatrix} = -1$$

   $$\alpha_1\begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}3\\1\\1\end{pmatrix} + \alpha_2\begin{pmatrix}3\\1\\1\end{pmatrix}\begin{pmatrix}3\\1\\1\end{pmatrix} + \alpha_3\begin{pmatrix}3\\1\\1\end{pmatrix}\begin{pmatrix}3\\-1\\1\end{pmatrix} = +1$$

   $$\alpha_1\begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}3\\-1\\1\end{pmatrix} + \alpha_2\begin{pmatrix}3\\1\\1\end{pmatrix}\begin{pmatrix}3\\-1\\1\end{pmatrix} + \alpha_3\begin{pmatrix}3\\-1\\1\end{pmatrix}\begin{pmatrix}3\\-1\\1\end{pmatrix} = +1$$

   After simplification of above three equations, we get

   $2\alpha_1 + 4\alpha_2 + 4\alpha_3 = -1$

$4\alpha_1 + 11\alpha_2 + 9\alpha_3 = 1$

$4\alpha_1 + 9\alpha_2 + 11\alpha_3 = 1$

After simplification of above three equations, we get

$\alpha_1 = -3.5, \ \alpha_2 = \alpha_3 = .75$

The hyperplane that discriminates the positive class from the negative class is given by

$$\bar{w} = \sum_i \alpha_i \bar{S}_i$$

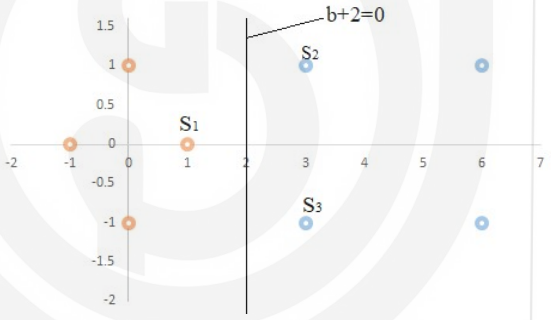
$$\bar{w} = \alpha_1 * \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + \alpha_2 * \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} + \alpha_3 * \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix}$$

$$\bar{w} = (-3.5) * \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + (7.5) * \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} + (7.5) * \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ -2 \end{pmatrix}$$

Hyperplane equation is y=wx+b

Where w= $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and

b=-2 or b+2=0 is a line parallel to y-axis which separate both classes.



11.8 FURTHER READINGS

1. Machine learning an algorithm perspective, Stephen Marshland, 2nd Edition, CRC Press, 2015.
2. Machine Learning, Tom Mitchell, 1st Edition, McGraw- Hill, 1997.
3. Machine Learning: The Art and Science of Algorithms that Make Sense of Data, Peter Flach, 1st Edition, Cambridge University Press, 2012.

UNIT 12 NEURAL NETWORKS AND DEEP LEARNING

Structure

12.1 INTRODUCTION

Jain et al. in 1996 mentioned in their work that a neuron is a unique biological cell that has the capability of information processing. Figure 1 describes a biological neuron's structure, consisting of a cell body and tree-like branches called axons and dendrites. The working of neurons is based on receiving the signals from other neurons through their dendrites, processing the alerts through their body, and finally passing the signals to other neurons via its axon. The synapse is responsible for connecting two neurons through an axon for the first neuron while the dendrite for the second neuron. A synapse can either enhance or reduce the learning capabilities' signal value. If the signals exceed a particular value, called a threshold, then the neuron fires, otherwise not fire.

Biological Neuron	**Artificial Neuron**
Cell Nucleus (Soma)	Node
Dendrites	Input
Synapse	Weights or interconnections
Axon	Output

Table1: Biological Neuron and Artificial Neuron

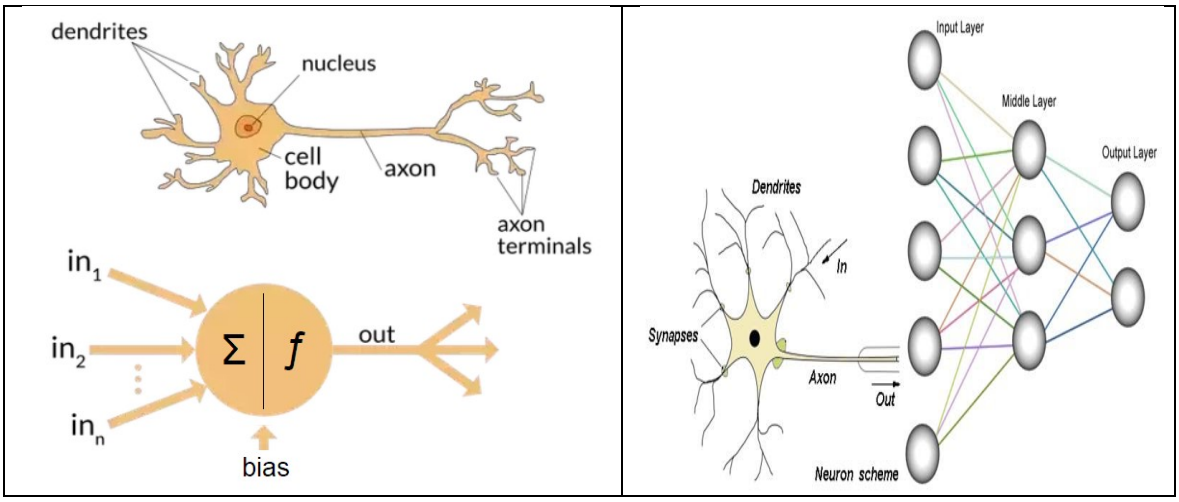


Figure 1: Biological Neuron and Artificial Neuron

12.2 OBJECTIVES

After completing this unit, you will be able to:

- Understand the concept of Neural Networks
- Understand Feed forward Neural networks
- Understand Back propagation Algorithm
- Understand the concept of Deep Learning

12.3 OVERVIEW OF ARTIFICIAL NEURAL NETWORKS

An artificial neural network (ANN) is like a computing system that simulates how the human brain analyzes information and processes it. It is the branch of artificial intelligence (AI) and solves problems that may be difficult or impossible to solve such issues for humans. In addition, ANNs have the potential for self-learning that provide better results if more data becomes available.

Artificial neurons consist of the following things:

- **Interconnecting model of neurons**. The neuron is the elementary component connected with other neurons to form the network.
- **Learning algorithm** to train the network. Various learning algorithms are available in the literature to train the model. Each layer consists of neurons, and

these neurons are connected to other layers. Weight is also assigned to each layer, and these weights are changed at each iteration for training purpose.

An artificial neural network (ANN) consists of an interconnected group of artificial neuronsthat process information through input, hidden and output layers and use a connectionist approach to computation. Neural networks use nonlinearstatisticaldata modeling tools to solve complex problems by finding the complex relationships between inputs and outputs. After getting this relation, we can predict the outcome or classify our problems.

ANN is similar to the biological neural networks as both perform the functions collectively and in parallel. Artificial Neural Network (ANN) is a general term used in various applications, such as weather predictions, pattern recognitions, recommendation systems, and regression problems.

Figure 2 describes three neurons that perform "AND" logical operations. In this case, the output neuron will fire if both input neurons are fired. The output neurons use a threshold value (T), T=3/2 in this case. If none or only one input neuron is fired, then the total input to the output becomes less than 1.5 and firing for output is not possible. Take another scenario where both input neurons are firing, and the total input becomes 1+1=2, which is greater than the threshold value of 1.5, then output neurons will fire. Similarly, we can perform the "OR" logical operation with the help of the same architecture but set the new threshold to 0.5. In this case, the output neurons will be fired if at least one input is fired.

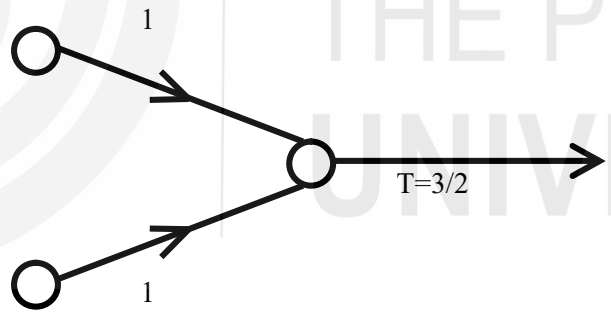


Figure 2: Three neurons diagram

Example-1 : Below is a diagram if a single artificial neuron (unit):

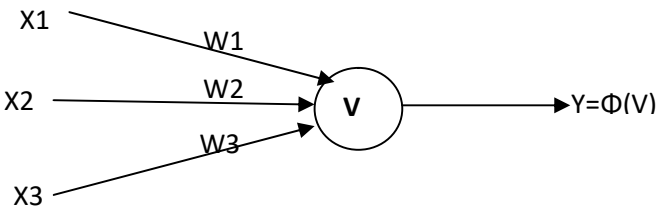


Figure A: Single unit with three inputs.

The node has three inputs x = (x1, x2, x3) that receive only binary signals (either 0 or 1). How many different input patterns this node can receive? What if the node had four inputs? Or Five inputs? Can you give a formula that computes the number of binary input patterns for a given number of inputs?

Answer - 1: For three inputs the number of combinations of 0 and 1 is 8:

x1 : 0 1 0 1 0 1 0 1
x2 : 0 0 1 1 0 0 1 1
x3 : 0 0 0 0 1 1 1 1

and for four inputs the number of combinations is 16:

x1 : 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
x2 : 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
x3 : 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
x4 : 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1

You may check that for five inputs the number of combinations will be 32. Note that $8 = 2^3$, $16 = 2^4$ and $32 = 2^5$ (for three, four and five inputs).

Thus, the formula for the number of binary input patterns is: 2^n, where n in the number of inputs.

☞ **Check Your Progress 1**

Question -1: Below is a diagram if a single artificial neuron (unit):

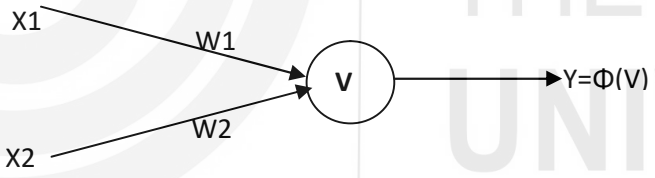


Figure A-1: Single unit with three inputs.

The node has three inputs x = (x1, x2) that receive only binary signals (either 0 or 1). How many different input patterns this node can receive?

……

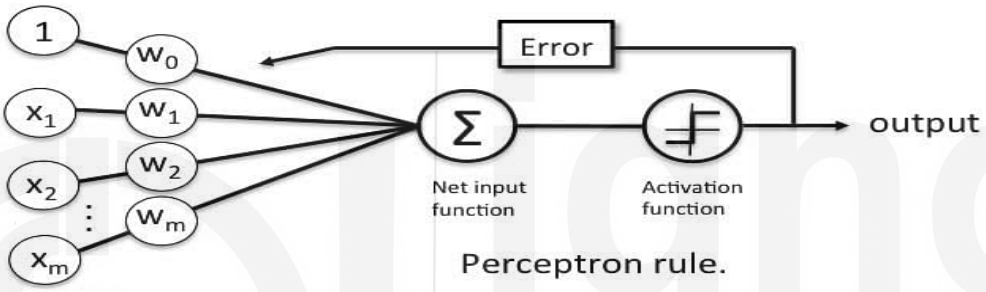
……

12.4 MULTILAYER FEEDFORWARD NEURAL NETWORKS WITH SIGMOID ACTIVATION FUNCTIONS

A multilayer feed forward neural network consists of the interconnection of various layers, named input, hidden layer, and output layer. The number of hidden layers is not fixed. It depends

upon the requirements and complexity of the problem. The simple neural network is one with a single input layer and an output layer is known as perceptrons. A Perceptron accepts inputs, moderates them with certain weight values, then applies the transformation function to output the final result. The word perceptron is used here because every connection has a certain weight, and through these connections, one layer is connected to the next layer.

The model's working is defined as follows: All inputs usually are multiplied by the weight, and this weighted sum is calculated. After it, this sum is applied to the activation function, and it is the output of an individual layer. This output becomes the input to the next layer. We have various activation functions, such as sigmoid, tanh, and Relu. After getting the output, the predicted output is compared with the actual output.



12.4.1 Neural Networks with Hidden Layers

Figure 3 describes the hidden layers of a neural network by adding more neurons in between the input and output layers. There may be a single hidden layer or multiple hidden layers.

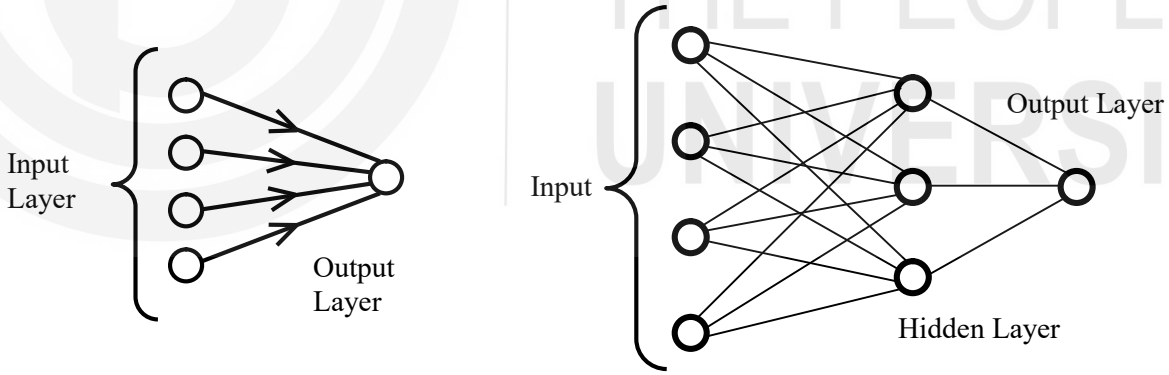


Figure 3: Neural network with a hidden layer

Data/ input is labeled in the input layer using *x value*with 1*, 2, 3, …, m as the subscript*, while neurons in the hidden layer are labeled as *h* with subscripts *1, 2, 3, …, n*... this n and m may be different as the hidden layer neurons, and the input neurons may have different values. Also, as several hidden layers may be multiple, the *first hidden layer has superscript 1, while the second hidden layer has superscript 2, and so on*. Output is labeled as *y* with a *hat i.e.,*$\hat{y}$.

The input data/ features with m dimension represented as ($x_1, x_2, \ldots, x_m$). You may say that a feature is nothing, but it is only a dependent variable that significantly influences a specific outcome/ dependent variable.

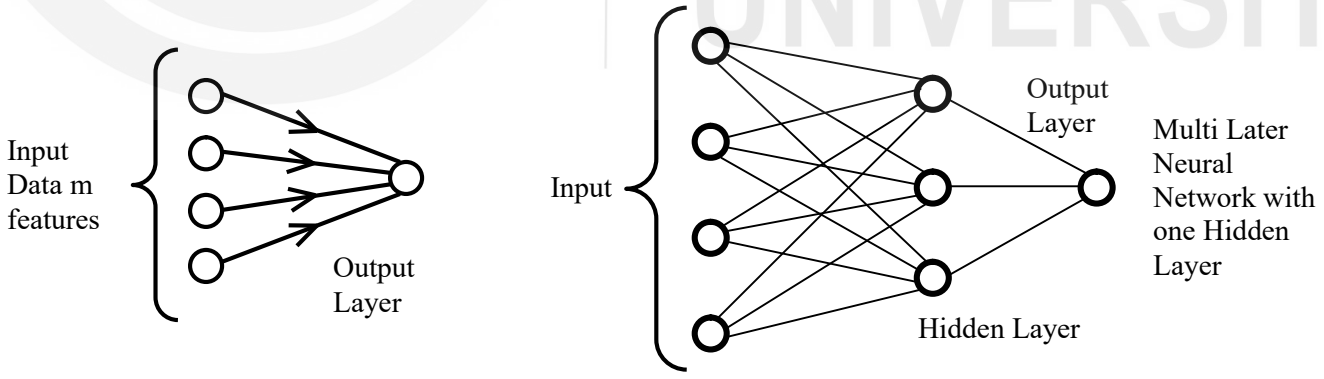
Now, we multiply m features ($x_1, x_2, \ldots, x_m$) with ($w_1, w_2, \ldots, w_m$) as a weight matrix, and then the sum is computed by adding these multiplicative terms. Finally, we define it as a **dot product**:

$$\mathbf{w.x} = w_1 x_1 + w_2 x_2 + \ldots + w_m x_m = \sum_{i=1}^{m} w_i x_i$$

There is the following important observation:

1. For ***m features***, we need precisely ***m*** weights to compute a dot product
2. Further, the exact computation is performed at the hidden layer. With n hidden neurons at the hidden layer, you need n number of consequences ($w_1, w_2, \ldots, w_n$) to find out the dot products
3. With one hidden layer, the output is defined as h: ($h_1, h_2, \ldots, h_n$)
4. Now imagine that you are working on a single-layer perceptron where you may consider hidden output h: ($h_1, h_2, \ldots, h_n$) as input data and perform dot product with weights ($w_1, w_2, \ldots, w_n$) to get your final output y_hat i.e.,$\hat{y}$.

Now, refereeing the above steps, you can understand the working of the multiple layers model and how it works; When you train the model networks on more extensive datasets with many input features, this process will consume a lot of computing resources. Therefore, deep learning was not popular in the early days as limited computing resources were available. However, when better configuration hardware is available, deep learning takes the attention of researchers. The procedure for forwarding the input features to the hidden layer and the hidden layer to the output layer is shown below in Figure 4.



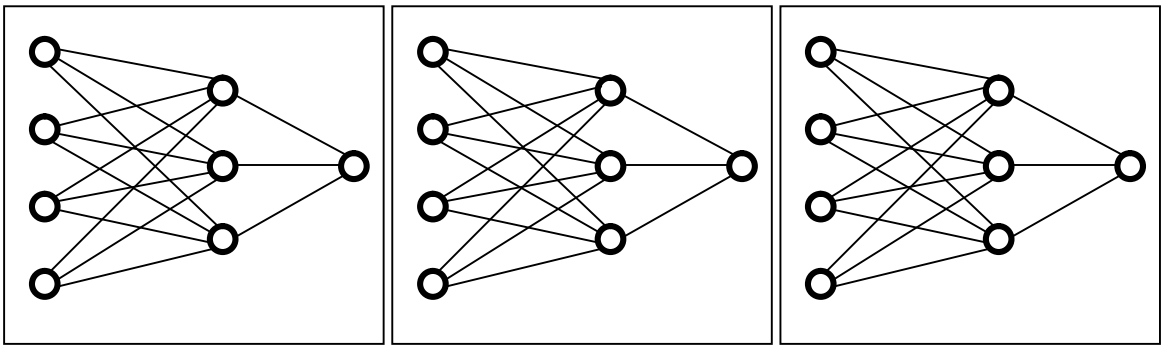


Figure 4: Neural network with different weights

Now you can understand the exact working how multiple layers work.

Example – 2 Consider the unit shown below.

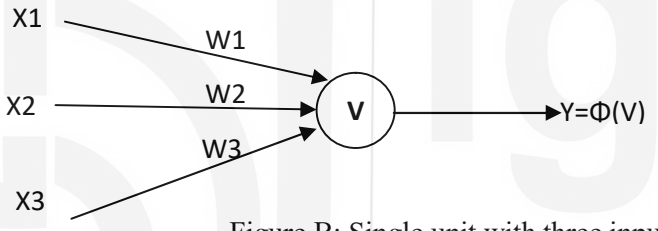


Figure B: Single unit with three inputs.

Suppose that the weights corresponding to the three inputs have the following values:

$w_1 = 2$; $w_2 = -4$; $w_3 = 1$

and the activation of the unit is given by the step-function:

$\Phi(V) = 1$ for $V >= 0$ and $\Phi(V) = 0$ Otherwise

Calculate what will be the output value y of the unit for each of the following input patterns:

Pattern	P_1	P_2	P_3	P_4
X_1	1	0	1	1
X_2	0	1	0	1
X_3	0	1	1	1

Answer:

To find the output value y for each pattern we have to:

a) Calculate the weighted sum: $v = \sum_i wi\ xi = w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3$

b) Apply the activation function to v, the calculations for each input pattern are:

$P1 : v = 2 \cdot 1 - 4 \cdot 0 + 1 \cdot 0 = 2 , (2 > 0) , y = \phi(2) = 1$

$P2 : v = 2 \cdot 0 - 4 \cdot 1 + 1 \cdot 1 = -3 , (-3 < 0) , y = \phi(-3) = 0$

$P3 : v = 2 \cdot 1 - 4 \cdot 0 + 1 \cdot 1 = 3 , (3 > 0) , y = \phi(3) = 1$

$P4 : v = 2 \cdot 1 - 4 \cdot 1 + 1 \cdot 1 = -1 , (-1 < 0) , y = \phi(-1) = 0$

Example - 3: Logical operators (i.e. NOT, AND, OR, XOR, etc) are the building blocks of any computational device. Logical functions return only two possible values, true or false, based on the truth or false values of their arguments. For example, operator AND returns true only when all its arguments are true, otherwise (if any of the arguments is false) it returns false. If we denote truth by 1 and false by 0, then logical function AND can be represented by the following table:

x1 :	**0**	**0**	**1**	**1**
x2 :	**0**	**1**	**0**	**1**
x1 AND x2 :	**0**	**0**	**0**	**1**

This function can be implemented by a single-unit with two inputs:

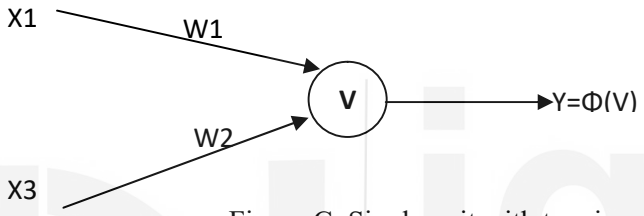


Figure C: Single unit with two inputs.

if the weights are w1 = 1 and w2 = 1 and the activation of the unit is given by the step-function:

$\Phi(V) = 1$ for $V>=2$ and $\Phi(V) = 0$ Otherwise

Note that the threshold level is 2 ($v \geq 2$).

a) Test how the neural AND function works.

Answer (a):

P1 : $v = 1 \cdot 0 + 1 \cdot 0 = 0$, $(0 < 2)$, $y = \phi(0) = 0$

P2 : $v = 1 \cdot 1 + 1 \cdot 0 = 1$, $(1 < 2)$, $y = \phi(1) = 0$

P3 : $v = 1 \cdot 0 + 1 \cdot 1 = 1$, $(1 < 2)$, $y = \phi(1) = 0$

P4 : $v = 1 \cdot 1 + 1 \cdot 1 = 2$, $(2 = 2)$, $y = \phi(2) = 1$

b) Suggest how to change either the weights or the threshold level of this single unitto implement the logical OR function (true when at least one of the arguments is true):

x1 :	**0**	**0**	**1**	**1**
x2 :	**0**	**1**	**0**	**1**
x1 OR x2 :	**0**	**1**	**1**	**1**

Answer(b): One solution is to increase the weights of the unit: w1 = 2 and w2 = 2:

P1 : $v = 2 \cdot 0 + 2 \cdot 0 = 0$, $(0 < 2)$, $y = \phi(0) = 0$

P2 : $v = 2 \cdot 1 + 2 \cdot 0 = 2$, $(2 = 2)$, $y = \phi(2) = 1$

P3 : $v = 2 \cdot 0 + 2 \cdot 1 = 2$, $(2 = 2)$, $y = \phi(2) = 1$

P4 : $v = 2 \cdot 1 + 2 \cdot 1 = 4$, $(4 > 2)$, $y = \phi(4) = 1$

Alternatively, we could reduce the threshold to 1:

$\Phi(V) = 1$ for $V>=1$ and $\Phi(V) = 0$ Otherwise

c) The XOR function (exclusive or) returns true only when one of the arguments is true and another is false. Otherwise, it returns always false. This can be represented by the following table:

x1 :	**0**	**0**	**1**	**1**
x2 :	**0**	**1**	**0**	**1**
x1 XOR x2 :	**0**	**1**	**1**	**0**

Do you think it is possible to implement this function using a single unit? A network of several units?

Answer(c): This is a difficult question, and it puzzled scientists for some time because it is impossible to implement the XOR function neither by a single unit nor by a single-layer feed-forward network (single-layer perceptron). This was known as the XOR problem. The solution was found using a feed-forward network with a hidden layer. The XOR network uses two hidden nodes and one output node.

☞ Check Your Progress 2

Question-2 : Consider the unit shown below.

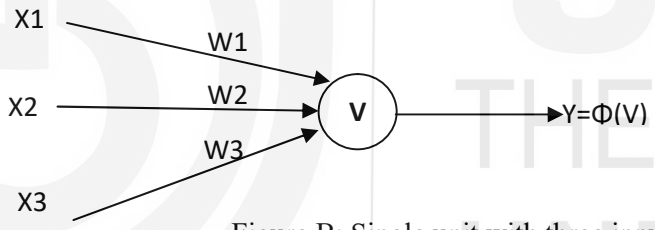


Figure B: Single unit with three inputs.

Suppose that the weights corresponding to the three inputs have the following values:

$w_1 = 1$; $w_2 = -1$; $w_3 = 2$

and the activation of the unit is given by the step-function:

$\Phi(V) = 1$ for $V>=1$ and $\Phi(V) = 0$ Otherwise

Calculate what will be the output value y of the unit for each of the following input patterns:

Pattern	**P_1**	**P_2**	**P_3**	**P_4**
X_1	1	0	1	1
X_2	0	1	0	1
X_3	0	1	1	1

……

……

Question - 3: NAND, NOR) are the universal building blocks of any computational device. Logical functions return only two possible values, true or false, based on the truth or false values of their arguments. For example, operator NAND returns False only when all its arguments are True, otherwise (if

any of the arguments is false) it returns false. If we denote truth by 1 and false by 0, then logical function NAND can be represented by the following table:

x1 :	0	0	1	1
x2 :	0	1	0	1
x1 NAND x2 :	1	1	1	0

This function can be implemented by a single unit with two inputs:

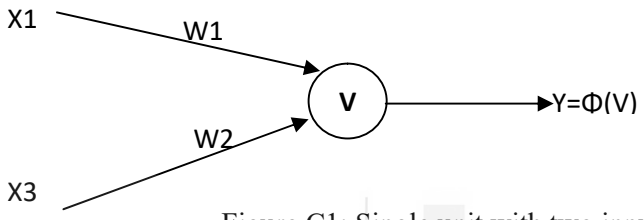


Figure C1: Single unit with two inputs.

if the weights are w1 = 1 and w2 = 1 and the activation of the unit is given by the step-function:

$\Phi(V) = 1$ for $V >= 2$ and $\Phi(V) = 0$ Otherwise

Note that the threshold level is 2 ($v \geq 2$).

a) Test how the neural NAND function works.
b) Suggest how to change either the weights or the threshold level of this single unit in order to implement the logical NOR function (true when at least one of the arguments is true):

x1 :	0	0	1	1
x2 :	0	1	0	1
x1 NOR x2 :	1	0	0	0

……

……

12.5 SIGMOID NEURONS: AN INTRODUCTION

So far, we have paid attention to a neural network model, how it works and what is the role of hidden layers. But now, we are required to emphasize on activation functions and their role in neural networks. The activation function is a mathematical function that decides the threshold value for a neuron, it may be linear or nonlinear.The purpose of an activation function is to add non-linearity to the neural network. If you have a linear activation function, then the number of hidden layers does matter, and the final output remains a linear combination of the input data. However, this linearity cannot help solving complex problems like patterns separated by curves where nonlinear activation is required.

Moreover, the activation function does not have a helpful derivative as its derivative is 0 everywhere. Therefore, it doesn't work for **Backpropagation**, a fundamental and valuable concept in multilayer perceptron.

Now, as we've covered the essential concepts, let's go over the most popular neural networks activation functions.

Binary Step Function: Binary step function depends on a threshold value that decides whether a neuron should be activated or not. The input fed to the activation function is compared to a certain threshold; if the input is greater than it, then the neuron is activated, else it is deactivated, meaning that its output is not passed on to the next hidden layer.

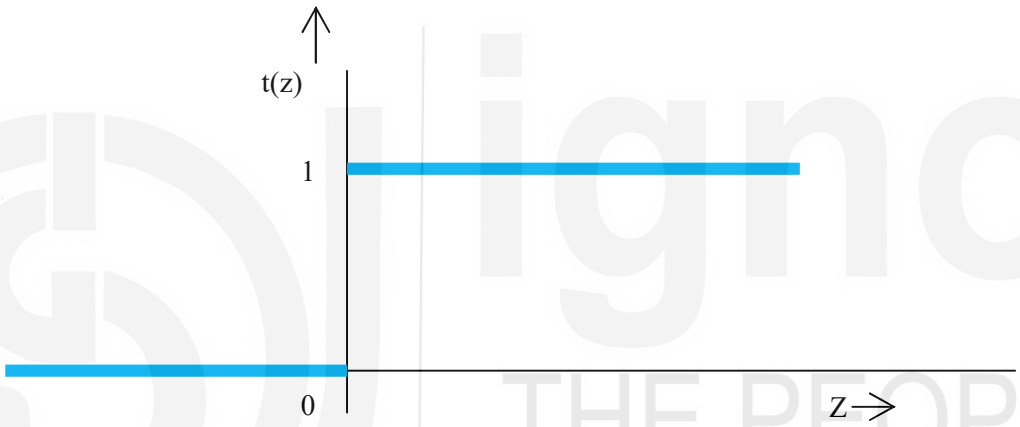


Figure 6: Sigmoidal Function

Mathematically it can be represented as:

$$F(x) = 0 \text{ for all } x<0$$
$$F(x) = 1 \text{ for all } x >=0$$

Here are some of the limitations of binary step function:

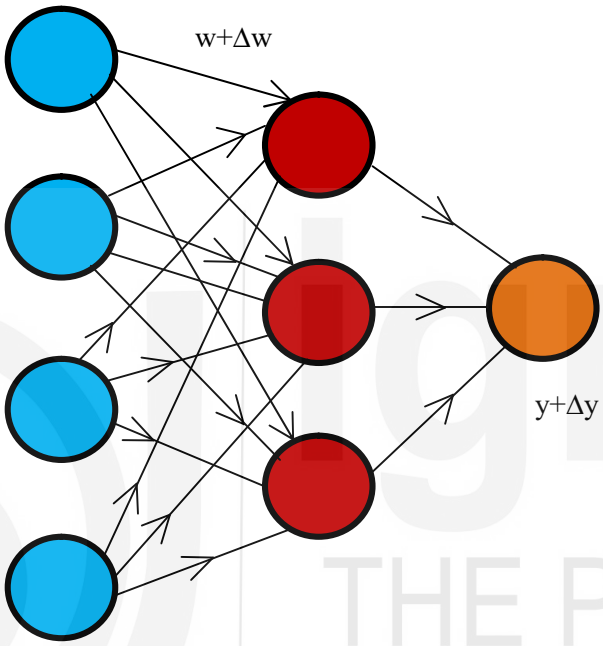
- It cannot provide multi-value outputs—for example, it cannot be used for multi-class classification problems.
- The gradient of the step function is zero, which causes a hindrance in the backpropagation process.

The idea of step function/Activation will be clear from this paragraph. For example, we have a perceptron with an activation function that isn't very "stable" as a relationship candidate.

For example, say some person has bipolar issues. One day ($z < 0$), s/he behaves with no responses as s/he is quiet, and on the second day ($z \geq 0$), s/he changes the mood and becomes very talkative, and speaks non-stop in front of you. There is s no transition for the spirit, and you

don't know the behavior when s/he will be quiet or talking. In such cases, we have a nonlinear step function that helps.

So, minor changes in the weight of the input layer of our model may activate the neuron by flipping from 0 to 1, which impacts the working of the hidden layer's working, and then the outcome may affect. Therefore, we want a model that enhances our exiting neural network by adjusting the weights. However, it is not possible by a linear activation function. If we don't have such activation functions, this task cannot be accomplished by simply changing the weights.



So, we need to say goodbye to the perceptron model with this linear activation function.

We are finding a new activation function that accomplishes our task for our neural network through the sigmoid function. We are changing only one thing: the activation function, and it meets our recruitments, which are sudden changes in the mood. Now, we define the learning Function by

$$Z = \sum_{i-1}^{m} w_i x_i + bias$$

$$\text{Sigmoidal function is: } \sigma(z) = \frac{1}{1 + e^{-z}}$$

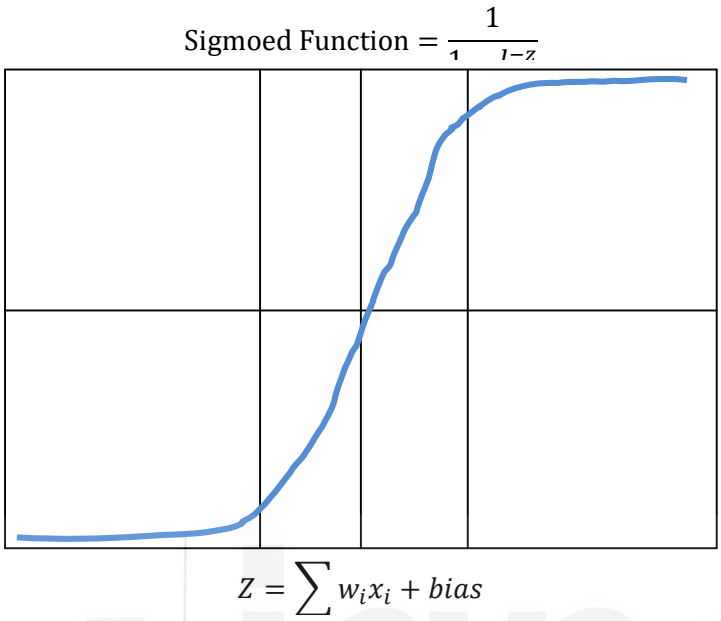


Figure 7: Sigmoidal Function

$\sigma(z)$ The function is called the sigmoid function. First, the value, Z, is computed then the sigmoid function is applied to$Z$. However, it looks very abstract or strange to you how it works. Those who don't have good knowledge of mathematics need not worry. **Figure 7** explains its curve and its derivative. Here are some observations mentioned:

1. The output of the Sigmoid Function produces the same results as produced by the linear step function;the output remains between 0 and 1. The curve marks 0.5 at *z=0*, for which we can make a straightforward rule that if the sigmoid neuron's output becomes more than or equal to 0.5, then its output one; otherwise, output o given for smaller values.
2. The sigmoid function should be continuous. It means that partial derivative, that is, *σ(z) / (1-σ(z)),* which is differentiable everywhere on the curve.

3. If z is a significant negative value, then the output is approximately 0; if z is a significant positive value, the output is given by around 1

The sigmoid activation function introduces non-linearity, which is the essential part, into our model. The meaning of this non-linearity is that the output is found out by the dot product of some inputs x (x_1, x_2, …, x_m), weights w (w_1, w_2, …, wm) plus bias, and then apply sigmoid function, cannot be represented linearly. The idea is that the nonlinear activation function allows us to classify nonlinear decision boundaries in our data.

We use hidden layers in our model by replacing perceptron with sigmoid activation function neurons. Now, the question arises what the requirement for hidden layers is? Are these useful? The answer is in yes. Hidden layers help us handle complex problems that single-layer neurons cannot solve.

Hidden layers twist the problem so that it can rewrite the problem and provide easy solutions to complex problems, pattern recognition problems. For example, figure 8 explains a classic textbook problem, recognition of handwritten digits, that can help you understand the workings of hidden layers and how they work.

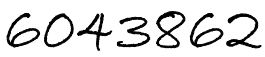


Figure 8: Digits in dataset MNIST

The digits in figure 8 is taken from a well-known dataset called MNIST. It has 70,000 examples of numbers that were written by a human. A picture of 28x28 pixels represents every digit. Therefore, this value is 28*28=784 pixels. Every pixel takes a deal between 0 and 255 (RGB color code). Zero manners the coloration is white and 255 manners the shade black.

Now, think about that computer that can really "see" a digit like a human see—the answer is no. Therefore, we need proper training to recognize these digits. The computer can't understand an image as a human can see. For this purpose, it can be interpreted to analyze how the pixel numbers are working to represent an image. Here, we dissect an image into an array defined by 784 numbers as appearing in each collection [0, 0, 180, …, … 77, 0, 0, 0], and after that, we need to feed the array into our model.

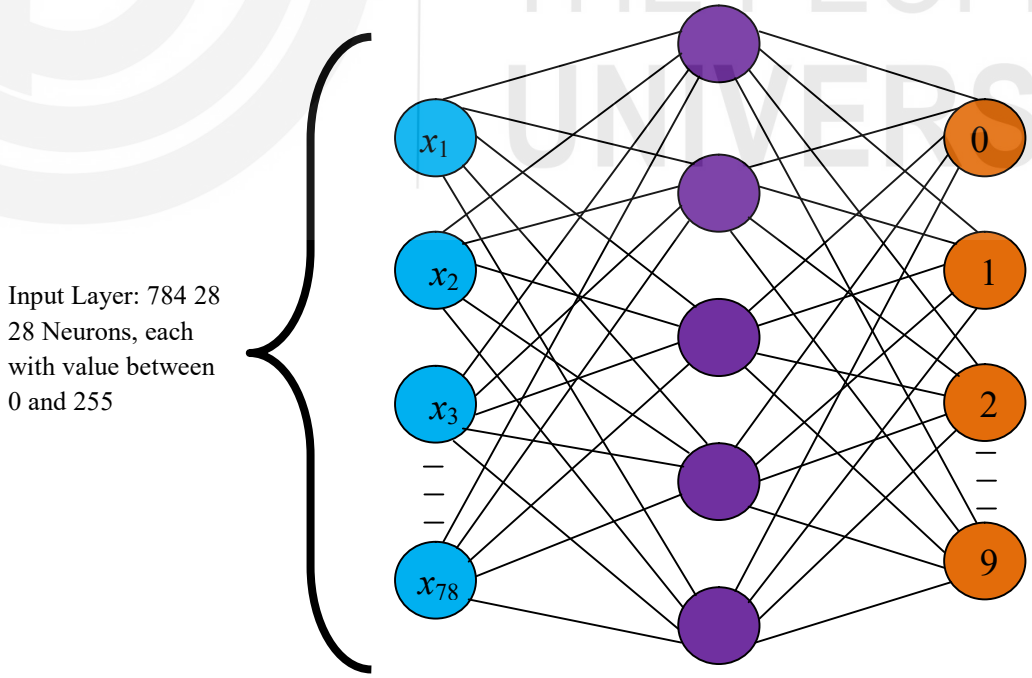


Figure 9: Neural Network with 28 * 28 pixel values

We set up a neural network, figure 9, for the problem mentioned above. It consists of 784 neurons for input layers with 28x28 pixel values. So, you may consider a total of 16 hidden neurons and ten output neurons. The ten output neurons returning in the form of an array will have different values to classify any digit from 0 to 9. So, for example, if the neural network finds the handwritten number is a zero, then the output array of [1, 0, 0, 0, 0, 0, 0, 0, 0, 0] would be returned, the first output of the array would be fired a zero, while rest of neurons at output layer would be set at 0. Similarly, take another example, If the neural network gets that the handwritten digit is a 5, then the array sequence would [0, 0, 0, 0, 0, 1, 0, 0, 0, 0] with six digits one while rest of the values will be 0's. Now, you can easily find out the sequence for any other number.

☞ **Check Your Progress 3**

Question-4 Discuss the utility of Sigmoid function in neural networks. Compare Sigmoid function with the Binary Step function.

12.6 BACK PROPAGATION ALGORITHM

The Backpropagation algorithm is a supervised learning algorithm for training the neural network model. **This algorithm** was first introduced in the 1960s, it was not popular, and in 1989 it gets popularized by Rumelhart, Hinton, and Williams, who have used this concept in a paper titled *"Learning representations by back-propagating errors.". It* is one of the most fundamental building blocks of any neural network.**, if you have multiple layers in the** neural network. Then**it is used to adjust the weight in the backward direction.**

When designing a neural network, we initially need to initialize the weights and biases with some random values. We initially gave some random values for weight and bias, but our model, through the backpropagation algorithm, will adjust these values and get the output if the difference between our actual output and predicted output is a large, more significant error.

This algorithm trains the neural network model based on chain rule method. In simple terms, you can say that after every forward pass through a network, the backpropagation algorithm works to perform a backward pass to adjust the weights and biased parameters of the model. It repeatedly adjusts the weights and biases of all the edges among all the layers so that Error i.e., the difference between predicted output and real output, should be minimum.

In other words, you can conclude that Backpropagation is **used to minimize the error/ cost function by repeatedly adjusting the network's weights and biases.** The level of adjustment is calculated by a method called the gradients of the error function concerning weight and biased parameters. In short, we are changing the weight and bias parameters so that Error becomes very small. Below Figure 10 explains the working of Backpropagation algorithm.

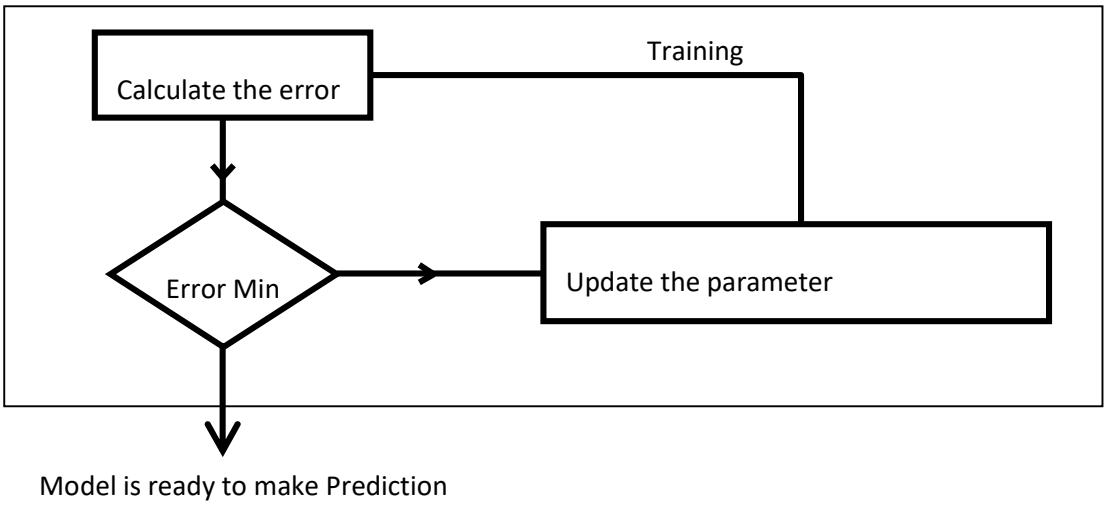


Figure 10: Backpropagation Model

The steps of backpropagation model are given below:

1. First, some random value 'W'is initialized as the weights and propagated forward accordingly.
2. Then, find out the Error after reducing that Error by propagating backward and increasing the value weight 'W.'
3. After that, observe the Error and whether it has been increased. If supplementing, then don't increase the value of 'W.'
4. Once again propagated backward and, at this time, decreased the value of 'W'.
5. Now, notice the Error, and check it has been reduced or not.

The weights that minimize the cost/ error reported. The detailed working is given by:

- Calculate the Error – The difference between the output produced by the model and the actual.
- Minimize the Error – need to check the Error, whether it is minimum or not.
- Tune the parameters – If the error value is substantial, the weights and biases must be updated. After reporting that significant Error again, this process will be repeated until the Error becomes very small.
- Check whether the model is ready for prediction – if an Error becomes significantly less, you can give some inputs to your model to get the output.

Now we learned about the need of backpropagation model and the meaning of training the model.

Now, we understand how the weight values are adjusted to reduce the Error. We are to determine whether an increment or decrement in the weight is required. After knowing it, we can keep updating the weights in that direction to reduce the Error. After some time, you will get to the

exact point where the Error is also increased if weights are updated further. We need to stop at that moment, and it becomes the final weight value.

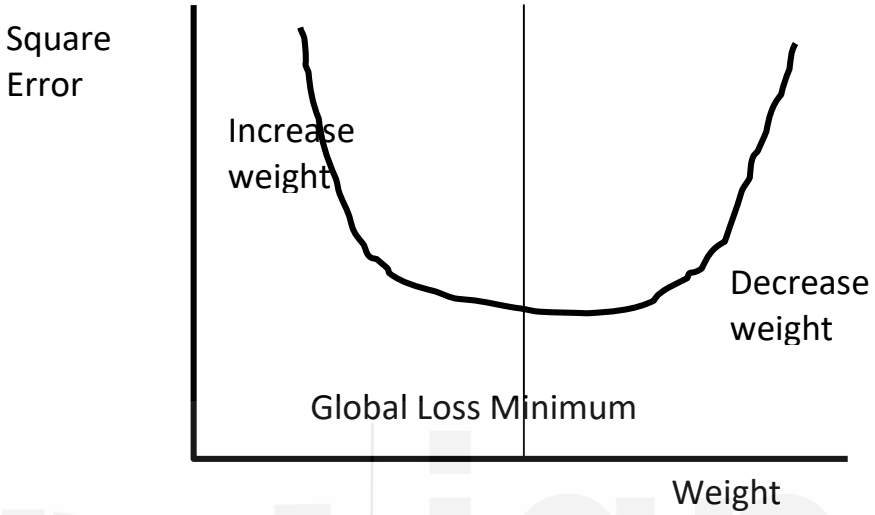


Figure 12: Error Calculation

Backpropagation Algorithm:

Initially, initialize the network weights and take small random values for it

do

for every training example, say we are terming as ex

prediction_ouput = neural_net_output_produced (network, ex) /*//it is used for forward pass*

actual output = teacher output(ex)

compute the Error part by (prediction output-actual output)

compute Delta w{h}} for all the mentioned weights, from the hidden layer to the output layer /*// it is used for the backward pass*

compute Delta w{i}} for all weights, from input layer to hidden layer *// It is backward pass continue step*

need to update network weights accordingly *// input layer does not modify*

till all examples are correctly classified or after meeting another stopping criterion

12.6.1 How does Backpropagation work?

Now you may consider below Neural Network for a better understanding:

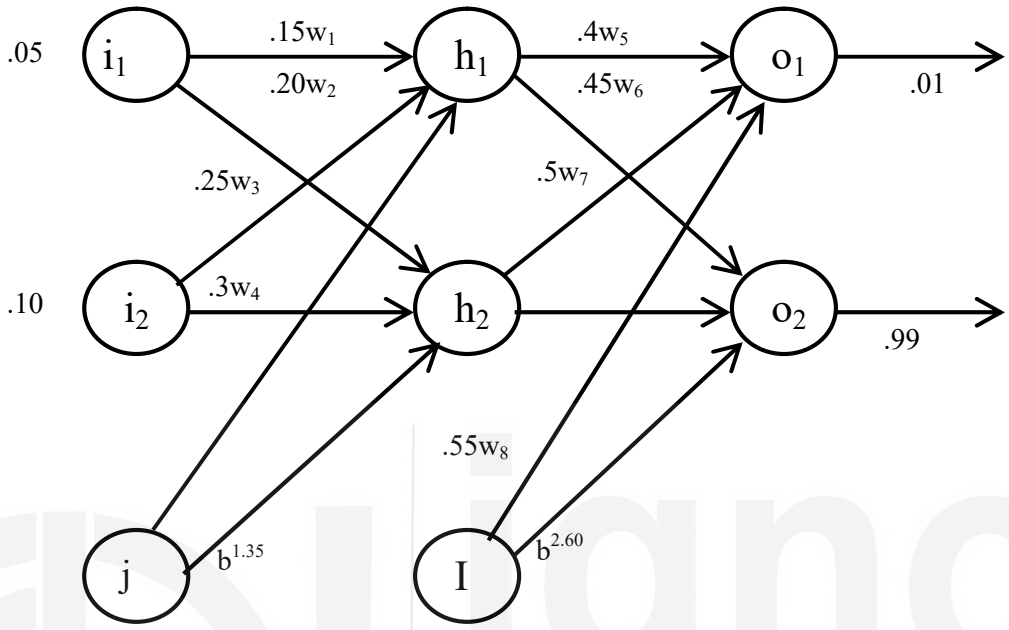


Figure 13: Neural Network Example

This network contains:

1. Three input layers
2. Two layersof hidden neurons
3. Two neurons at the output layer.

The following steps are used in the Backpropagation:

Step1: We need to use forward propagation

Step 2: After that, we have to follow backward propagation

Step 3: We put all the values to calculate the updated weight

Step 1: We use forward propagation

We start the working with forwarding propagation

$O_{1\text{Output}}$

$$net\ 0_1 = w_5 \times out\ h_1 + w_6 \times out\ h_2 + b_2 \times 1 \rightarrow .4 \times .5932 + .45 \times .5968 + .6 \times 1 = 1.1019$$

$$Out\ 0_1 = \frac{1}{1 + l^{-net\ 0_1}} \rightarrow \frac{1}{1 + l^{-1.1059}} = 0.7523$$

$$Out\ 0_2 = 0.7629$$

We also repeat the process for the output layer neurons. Hidden layer neurons outputs become the inputs.

Error for $0^1 : E0_1 = \sum \frac{1}{2}(target - output)^2 = \frac{1}{2}(.01 - 0.7513)^2 = 0.2748$

Error for $0_2 : E0_2 = 0.02356$

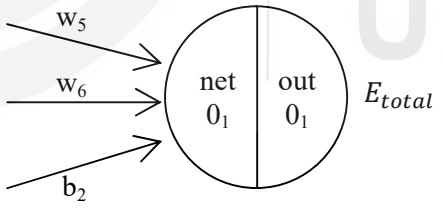
$$Total\ Error : E\ total = E0_1 + E0_2 = 0.2748 + 0.02356$$

Step – 2: Follow backward propagation

Now, we use Backpropagation to reduce the errors by adjusting weight and biases.

Now, we are checking for W_5

$$\frac{SEtotal}{Sw_5} = \frac{SEtotal}{Sout\ 0_1} \times \frac{Sout\ 0_1}{Snet\ 0_1} \times \frac{Snet\ 0_1}{Sw_5}$$



After applying the Backpropagation, we find a total change in errors regarding output-1 : O_1 and output-2 : O_2.

$$E_{total} = \frac{1}{2}(target\ 0_1 - out\ 0_2)^2 + \frac{1}{2}(target\ 0_2 - out\ o_2)^2$$

$$\frac{SE_{total}}{Sout\ 0_1} = -(target\ 0_1 - out\ 0_1) = -(0.01 - .7513) = 0.74136$$

Now, we need to propagate backward to find the changes in O1 concerning its total net input.

$$out\ o1 = 1/1 + e^{-net}$$

$$\frac{\delta out\ 01}{\delta net\ 01} = out\ o1\ (1 - out\ o1) = 0.75136507\ (1 - 0.75136507) = 0.186815602$$

Now, we check the total changes in O1 concerning weight W5.

$$net\ 0_1 = w_5 \times out\ h_1 + w_6 \times out\ h_2 + b_2 \times 1$$

$$\frac{\delta net\ 0_1}{\delta w_5} = 1 * out\ h_1 w_5^{(1-1)} + 0 + 0 = .593269$$

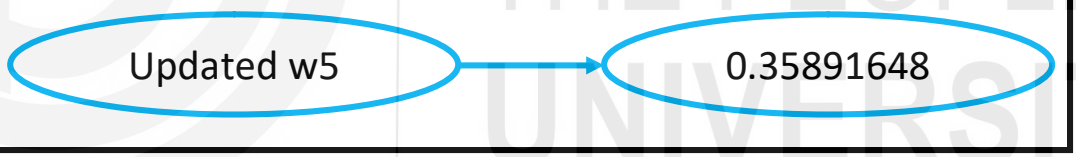
Step 3: We need to put all the values for calculating the updated weight
If we put all the values together, then:

$$\frac{\boldsymbol{\delta Etotal}}{\boldsymbol{\delta w_5}} = \frac{\boldsymbol{\delta Etotal}}{\boldsymbol{\delta out\ o1}} * \frac{\boldsymbol{\delta out\ o1}}{\boldsymbol{\delta net\ o1}} * \frac{\boldsymbol{\delta net\ o1}}{\boldsymbol{\delta w_5}} \rightarrow \mathbf{0.082167041}$$

Now, find out the updated value of weight W5:

$$W_5^+ = W_5 - n\frac{\delta Etotal}{\delta w_5}$$

$$W_5^+ = .4 - .5 * 0.082167041$$



- Similarly, we calculate the weight for others also.
- After that, we need to propagate forward and compute the output. After that, recalculate the Error.
- If a computed error is tiny, we need to stop. Otherwise, we further need to propagate backward and adjust the weight accordingly.
- We keep this process will continue till the Error is significantly less quantity.

☞ Check Your Progress 4

Question 5: Write Back Propagation algorithm, and showcase its execution on a neural network of your choice (make suitable assumptions if any)

……
……..

12.7 FEED FORWARD NETWORKS FOR CLASSIFICATION AND REGRESSION

Feed forward neural network is used for various problems, including classification , regression, and pattern encoding. In the first case, the web returns a value called z=f(w,x), which is very close to the target value y. While in the second case, the target becomes the input itself $v(x, y, f(w, x))$.To deal with multiclassification, we can use either of the techniques.

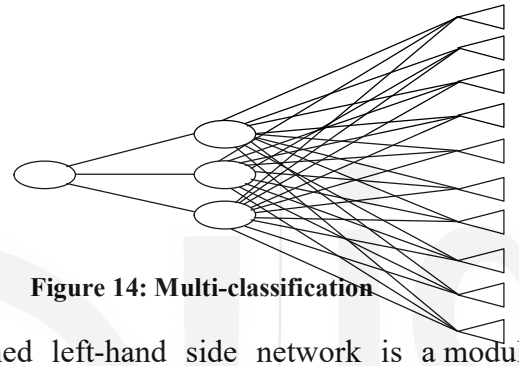
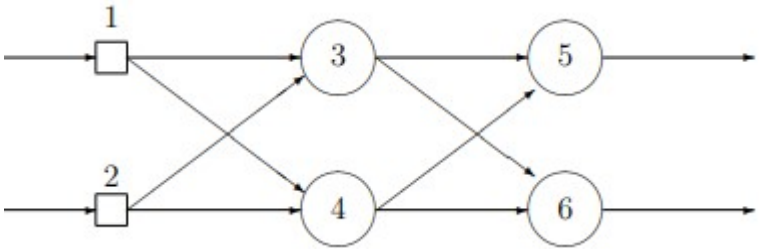


Figure 14: Multi-classification

The above-mentioned left-hand side network is a modular architecture. Here, every class connects with three distinct hidden neurons. While mentioned right-hand side network defines a fully connected network, which is used for a richer classification process. The left-side network is advantageous as it is modular and supports the classifiers' gradual construction. Whenever we feel to add a new class, the fully connected network requires further training, while the modular network only involves training for a new module. The same issue also holds for regression. However, it is worth mentioning that the output neurons are typically linear in regression tasks since there is no need to approximate any code.

As we have mentioned, a Neural network (NN) has several hidden layers; every layer consists of multiple neurons/ nodes. Each node connects with input layer connections and output layer connections. Also, we have mentioned that every connection is assigned a different weight, and finally output layer. Before giving the data into the NN, the dataset should be normalized and then processed. Training a neural network means adjusting the weights so that errors should be minimum. After introducing the NN, we can apply new data for classification or regression purposes.

Example - 4: The following diagram represents a feed-forward neural network with one hidden layer:



A weight on connection between nodes i and j is denoted by wij , such as w13 is the weight on the connection between nodes 1 and 3. The following table lists all the weights in the network:

w13 = -2, w23 = 3 ; w14=4,w24=-1 ; w35=1,w45=-1 ; w36 = -1, w46 = 1

Each of the nodes 3, 4, 5 and 6 uses the following activation function:

$\Phi(V) = 1$ for $V>=0$ and $\Phi(V) = 0$ Otherwise

Where, v denotes the weighted sum of a node. Each of the input nodes (1 and 2) can only receive binary values (either 0 or 1). Calculate the output of the network (y5 and y6) for each of the input patterns:

Pattern :	P1	P2	P3	P4
Node 1 :	0	1	0	1
Node 2 :	0	0	1	1

Answer: In order to find the output of the network it is necessary to calculate weighted sums of hidden nodes 3 and 4:

$v_3 = w_{13}x_1 + w_{23}x_2$, $v_4 = w_{14}x_1 + w_{24}x_2$

Then find the outputs from hidden nodes using activation function ϕ:

$y_3 = \phi(v_3)$, $y4 = \phi(v_4)$.

Use the outputs of the hidden nodes y3 and y4 as the input values to the output layer (nodes 5 and 6), and find weighted sums of output nodes 5 and 6:

v5 = w35y3 + w45y4 , v6 = w36y3 + w46y4 .

Finally, find the outputs from nodes 5 and 6 (also using ϕ):

y5 = ϕ(v5) , y6 = ϕ(v6) .

The output pattern will be (y5, y6). Perform this calculation for each input pattern:

P1: Input pattern (0, 0)

v3 = $-2 \cdot 0 + 3 \cdot 0 = 0$, y3 = $\phi(0) = 1$
v4 = $4 \cdot 0 - 1 \cdot 0 = 0$, y4 = $\phi(0) = 1$
v5 = $1 \cdot 1 - 1 \cdot 1 = 0$, y5 = $\phi(0) = 1$
v6 = $-1 \cdot 1 + 1 \cdot 1 = 0$, y6 = $\phi(0) = 1$

The output of the network is (1, 1)

P2: Input pattern (1, 0)

$v3 = -2 \cdot 1 + 3 \cdot 0 = -2,\ y3 = \phi(-2) = 0$

$v4 = 4 \cdot 1 - 1 \cdot 0 = 4,\ y4 = \phi(4) = 1$

$v5 = 1 \cdot 0 - 1 \cdot 1 = -1,\ y5 = \phi(-1) = 0$

$v6 = -1 \cdot 0 + 1 \cdot 1 = 1,\ y6 = \phi(1) = 1$

The output of the network is (0, 1).

P3: Input pattern (0, 1)

$v3 = -2 \cdot 0 + 3 \cdot 1 = 3,\ y3 = \phi(3) = 1$

$v4 = 4 \cdot 0 - 1 \cdot 1 = -1,\ y4 = \phi(-1) = 0$

$v5 = 1 \cdot 1 - 1 \cdot 0 = 1,\ y5 = \phi(1) = 1$

$v6 = -1 \cdot 1 + 1 \cdot 0 = -1,\ y6 = \phi(-1) = 0$

The output of the network is (1, 0).

P4: Input pattern (1, 1)

$v3 = -2 \cdot 1 + 3 \cdot 1 = 1,\ y3 = \phi(1) = 1$

$v4 = 4 \cdot 1 - 1 \cdot 1 = 3,\ y4 = \phi(3) = 1$

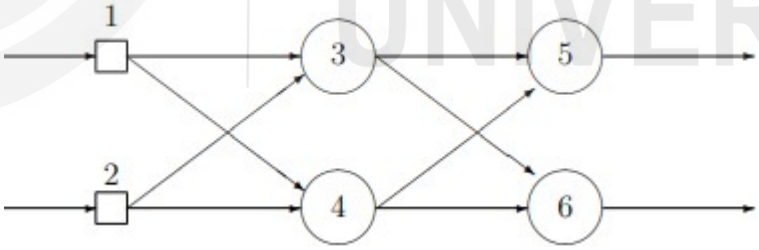
$v5 = 1 \cdot 1 - 1 \cdot 1 = 0,\ y5 = \phi(0) = 1$

$v6 = -1 \cdot 1 + 1 \cdot 1 = 0,\ y6 = \phi(0) = 1$

The output of the network is (1, 1).

☞ **Check Your Progress 5**

Question-6 The following diagram represents a feed-forward neural network with one hidden layer:



A weight on connection between nodes i and j is denoted by wij , such as w23 is the weight on the connection between nodes 2 and 3. The following table lists all the weights in the network:

w13 = -3, w23 = 2 ; w14=3,w24= - 2 ; w35=4,w45= - 3 ; w36 = -2, w46 = 2

Each of the nodes 3, 4, 5 and 6 uses the following activation function:

$\Phi(V) = 1$ for $V >= 1$ and $\Phi(V) = 0$ Otherwise

Where, v denotes the weighted sum of a node. Each of the input nodes (1 and 2) can only receive binary values (either 0 or 1). Calculate the output of the network (y5 and y6) for each of the input patterns:

Pattern :	P1	P2	P3	P4
Node 1 :	0	1	0	1
Node 2 :	0	0	1	1

.....

.....

12.8 DEEP LEARNING

Deep learning is a subset of artificial intelligence, commonly called AI, that tells us the workings of the human brain to process data and patterns defining for decision making. Deep learning has capable of learning unsupervised from unstructured data or unlabeled data. Deep learning is further classified as an AI function that is used to simulate the workings of the human brain in processing data to detect objects, recognize speech, translate languages, and make decisions.

12.8.1 How Deep Learning Works

Initially, there was a limitation of computing resources, and the concept of deep learning was not so popular. Once these resources were available, deep learning took the attention of the researchers. Deep Learning can handle all forms of data from all world regions. This data is available in massive amounts, termed big data, and is taken from various sources, including social media, search engines, different e-platforms, and others multimedia sources. Big data is accessible through multiple fintech applications such as cloud computing.

However, this data is so vast and primarily considered unstructured that it could take decades or centuries for humans to understand or find meaningful decisions. As mentioned earlier, the deep learning model's work is similar to the multilayer perceptron models. We have various models, such as convolution neural network (CNN) and long short term Model (LSTM). The exact working of CNN and LSTM is out of scope, but you can refer to the working of multilayer perception to understand the working of the deep learning model.

12.8.2 Deep Learning vs. Machine Learning

One of the popular AI techniques available for processing big data is machine learning.

For example, if a digital payments company wants to find out the occurrence of fraud in their system, such a company might use machine learning tools. The used algorithm, built on the machine learning technique, will process all transactions on the digital platform, try to find out the patterns, and mention the detection of anomalies by the pattern.

Deep learning, termed a subset of machine learning algorithms, uses a hierarchical structure of neural networks model to forward the same process as the machine learning algorithm used. The neural networks, like the human brain, connect like a web. The program built for machine learning linearly uses data, while deep learning systems enable a nonlinear approach to process the data.

12.8.3 A Deep Learning Example

For example, the fraud detection system mentioned above can be solved through deep learning. However, as a machine learning system starts working with parameters, such as transactions of dollars an account holder sends or receives, the deep-learning method can also use the same parameters, but it works on a neural network.

As we have mentioned earlier, each layer of the neural network takes the inputs from the previous layer; for example, the input layer has the parameters like sender information, data from social media, a credit score of the customer, using IP address, and others and passed the output to next layer for decision making. The final layer, the output layer, decides whether fraud has been detected or not.

Deep learning, a prevalent technology, is used across all major industries for various tasks, including decision making. Other examples may include commercial apps for image recognition, apps for a recommendation system, and medical research tools for exploring the possibility of reusing drugs.

☞ Check Your Progress 6

Question-7 Compare between Deep Learning and Machine Learning

……

……

12.9 SUMMARY

In this unit we learned about the fundamental concepts of Neural networks, and various concepts related to area of neural networks and deep learning, this includes the understanding of the activation function, back propagation algorithm, feed forward networks and many more. In this unit the concepts are simplified with the help of the numerical, which will help you map the theoretical concepts of neural networks with that of their implementation part.

12.10 SOLUTIONS/ANSWERS

☞ Check Your Progress 1

Question -1 : Below is a diagram if a single artificial neuron (unit):

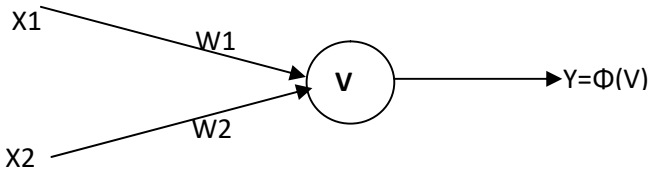


Figure A-1: Single unit with three inputs.

The node has three inputs x = (x1, x2) that receive only binary signals (either 0 or 1). How many different input patterns this node can receive?

Solution:**Refer to Section 12.3**

☞ Check Your Progress 2

Question-2 : Consider the unit shown below.

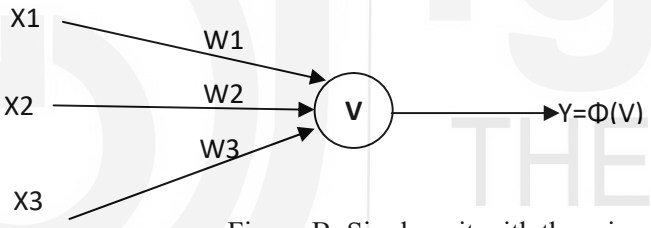


Figure B: Single unit with three inputs.

Suppose that the weights corresponding to the three inputs have the following values:

$w_1 = 1$; $w_2 = -1$; $w_3 = 2$

and the activation of the unit is given by the step-function:

$\Phi(V) = 1$ for $V >= 1$ and $\Phi(V) = 0$ Otherwise

Calculate what will be the output value y of the unit for each of the following input patterns:

Pattern	P_1	P_2	P_3	P_4
X_1	1	0	1	1
X_2	0	1	0	1
X_3	0	1	1	1

Solution: Refer to section 12.4

Question - 3: NAND, NOR) are the universal building blocks of any computational device. Logical functions return only two possible values, true or false, based on the truth or false values of their arguments. For example, operator NAND returns False only when all its arguments are True, otherwise (if any of the arguments is false) it returns false. If we denote truth by 1 and false by 0, then logical function NAND can be represented by the following table:

x1 :	**0**	**0**	**1**	**1**
x2 :	**0**	**1**	**0**	**1**
x1 NAND x2 :	**1**	**1**	**1**	**0**

This function can be implemented by a single unit with two inputs:

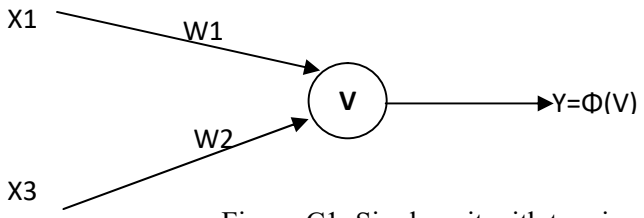


Figure C1: Single unit with two inputs.

if the weights are w1 = 1 and w2 = 1 and the activation of the unit is given by the step-function:

$\Phi(V) = 1$ for $V >= 2$ and $\Phi(V) = 0$ Otherwise

Note that the threshold level is 2 ($v \geq 2$).

a) Test how the neural NAND function works.
b) Suggest how to change either the weights or the threshold level of this single unit in order to implement the logical NOR function (true when at least one of the arguments is true):

x1 :	**0**	**0**	**1**	**1**
x2 :	**0**	**1**	**0**	**1**
x1 NOR x2 :	**1**	**0**	**0**	**0**

Solution: Refer to section 12.4

☞ Check Your Progress 3

Question-4 Discuss the utility of Sigmoid function in neural networks. Compare Sigmoid function with the Binary Step function.

Solution: Refer to Section 12.5

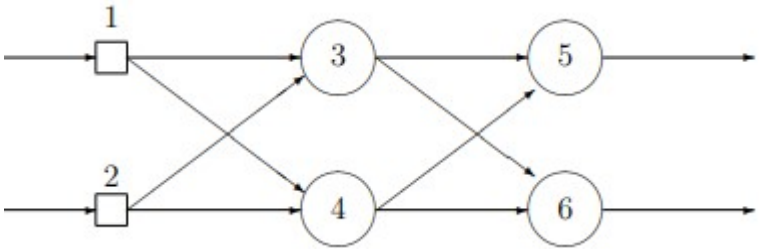
☞ Check Your Progress 4

Question 5: Write Back Propagation algorithm, and showcase its execution on a neural network of your choice (make suitable assumptions if any)

Solution: Refer to Section 12.6

☞ Check Your Progress 5

Question-6 The following diagram represents a feed-forward neural network with one hidden layer:



A weight on connection between nodes i and j is denoted by wij , such as w23 is the weight on the connection between nodes 2 and 3. The following table lists all the weights in the network:

w13 = -3, w23 = 2 ; w14=3,w24= - 2 ; w35=4,w45= - 3 ; w36 = -2, w46 = 2

Each of the nodes 3, 4, 5 and 6 uses the following activation function:

$\Phi(V) = 1$ for $V>=1$ and $\Phi(V) = 0$ Otherwise

Where, v denotes the weighted sum of a node. Each of the input nodes (1 and 2) can only receive binary values (either 0 or 1). Calculate the output of the network (y5 and y6) for each of the input patterns:

Pattern :	P1	P2	P3	P4
Node 1 :	0	1	0	1
Node 2 :	0	0	1	1

Solution: Refer to Section 12.7

☞ Check Your Progress 6

Question-7 Compare between Deep Learning and Machine Learning

Solution: Refer to Section 12.8

12.11 FURTHER READINGS

1) Dr K Uma Rao, "Artificial Intelligence and Neural Networks", Pearson Education (January 2011)

2) Tariq Rashid, "Make Your Own Neural Network: A Gentle Journey Through the Mathematics of Neural Networks, and Making Your Own Using the Python Computer Language",

3) Russell J. Stuart, Norvig Peter," Artificial Intelligence" , Pearson: A Modern Approach Paperback – January 2015.

4) F. AcarSavaci, Artificial Intelligence and Neural Networks, Springer; 2006th edition (18 July 2006).

5) Vladimir Golovko (Editor), Akira Imada (Editor), "Neural Networks and Artificial Intelligence: 8th International Conference, ICNNAI 2014"

6) ToshinoriMunakata, "Fundamentals of the New Artificial Intelligence" Springer; 2nd ed. 2008 edition (February 2008)