Savitribai Phule Pune University

Machine Learning

(Course Code : 410242) (Compulsory Subject)

Semester VII - Computer Engineering

Strictly as per the New Syllabus (2019 Course) of Savitribai Phule Pune University w.e.f. Academic Year 2022-2023

Prof. R. M. Baphana
Adjunct Faculty,
Government College of Engineering, Pune (C.O.E.P)
(Teaching M. Tech & Ph.D students)

Dr. Chaitanya S. Kulkarni
Ph.D Computer Engineering, M.Tech. Computer
H.O.D. and Associate Professor
Vidya Pratishthan's Kamalnayan Bajaj Institute of Engineering and Technology, Baramati, Dist. Pune: 413 133

Dr. Nilesh M. Patil
Ph.D. (Computer Engineering)
Associate Professor, Computer Engg. Department,
SVKM's D J Sanghvi College of Engineering, Mumbai

TECH-NEO PUBLICATIONS
Where Authors Inspire Innovation
A Sachin Shah Venture

P7-72

INDEX

In Sem

End Sem

❑❑❑

UNIT I

CHAPTER 1

Introduction to Machine Learning

Syllabus

Introduction to Machine Learning, Comparison of Machine learning with traditional programming, ML vs AI vs Data Science.

Types of learning: Supervised, Unsupervised, and semi-supervised, reinforcement learning techniques, Models of Machine learning: Geometric model, Probabilistic Models, Logical Models, Grouping and grading models, Parametric and non-parametric models.

Important Elements of Machine Learning- Data formats, Learnability, Statistical learning approaches

1.1 INTRODUCTION

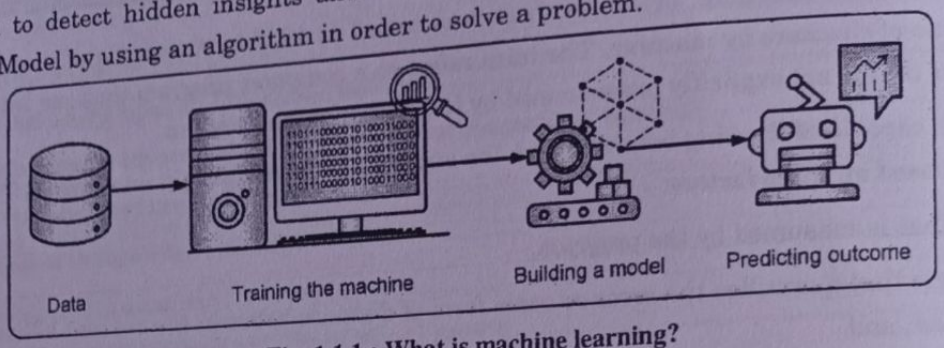
- The term "Machine Learning" or ML in short, was coined in 1959 by Arthur Samuel in the context of solving game of checkers by machine. The term refers to a computer program that can learn to produce a behaviour that is not explicitly programmed by the author of the program.
- Rather it is capable of showing behavior that the author may be completely unaware of. This behaviour is learned based on three factors:
  (1) Data that is consumed by the program,
  (2) A metric that quantifies the error or some form of distance between the current behavior and ideal behavior, and
  (3) A feedback mechanism that uses the quantified error to guide the program to produce better behavior in the subsequent events
- Machine learning is a subfield of AI and is, in many cases, the basis for AI technology. The goal of machine learning technology is to understand the structure of data and fit that data into specific models that are able to then be understood and used by humans for various applications throughout life.
- Traditional computer sciences are driven with algorithms that are human-created and managed, machine learning is driven by algorithms that the device itself can learn from and grow from. Beyond that, they are often built with a very specific purpose that enables them to specialize in specific areas of "knowledge" or capabilities.
- Everything from the face recognition capabilities in your phone to the self-driving technology in cars is derived from specialized forms of machine learning technology. It has become, and continues to become, a highly relevant and well researched part of our modern world.

1.1.1 What is Machine Learning ?

GQ. What is Machine Learning ?
GQ. What is the importance of Machine Learning ?

- Machine learning is a form of computer science technology whereby the machine itself has a complex range of "knowledge" that allows it to take certain data inputs and use complex statistical analysis strategies to create output values that fall within a specific range of knowledge, data, or information.
- "Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves."
- Machine learning devices essentially take data, and use to look for patterns and other pieces of specified information to create predictions or recommendations.
- The goal is for computers to learn how to use data and information to be able to learn automatically, rather than requiring humans to intervene or assist with the learning process.

- A Machine Learning process begins by feeding the machine lots of data, by using this data the machine is trained to detect hidden insights and trends. These insights are then used to build a Machine Learning Model by using an algorithm in order to solve a problem.



(1A7)**Fig. 1.1.1 : What is machine learning?**

1.1.2 Why Is Machine Learning Important ?

- Ever since the technical revolution, we've been generating an immeasurable amount of data. As per research, we generate around 2.5 quintillion bytes of data every single day! It is estimated that by 2020, 1.7MB of data will be created every second for every person on earth.
- With the availability of so much data, it is finally possible to build predictive models that can study and analyze complex data to find useful insights and deliver more accurate results
- Machine learning and data mining, a component of machine learning, are crucial tools in the process to glean insights from massive datasets held by companies and researchers today.

Here's a list of reasons why Machine Learning is so important :

(1) **Increase in Data Generation :** Due to excessive production of data, we need a method that can be used to structure, analyze and draw useful insights from data. This is where Machine Learning comes in. It uses data to solve problems and find solutions to the most complex tasks faced by organizations.

(2) **Improve Decision Making :** By making use of various algorithms, Machine Learning can be used to make better business decisions.

(3) **Uncover patterns & trends in data :** Finding hidden patterns and extracting key insights from data is the most essential part of Machine Learning. By building predictive models and using statistical techniques, Machine Learning allows you to dig beneath the surface and explore the data at a minute scale. Understanding data and extracting patterns manually will take days, whereas Machine Learning algorithms can perform such computations in less than a second.

(4) **Solve complex problems :** From detecting the genes linked to the deadly ALS disease to building self-driving cars, Machine Learning can be used to solve the most complex problems.

1.1.3 Machine Learning Definitions

- **Algorithm :** A Machine Learning algorithm is a set of rules and statistical techniques used to learn patterns from data and draw significant information from it. It is the logic behind a Machine Learning model. An example of a Machine Learning algorithm is the Linear Regression algorithm.

- **Mode l :** A model is the main component of Machine Learning. A model is trained by using a Machine Learning Algorithm. An algorithm maps all the decisions that a model is supposed to take based on the given input, in order to get the correct output.
- **Predictor Variable :** It is a feature(s) of the data that can be used to predict the output.
- **Response Variable :** It is the feature or the output variable that needs to be predicted by using the predictor variable(s).
- **Training Data :** The Machine Learning model is built using the training data. The training data helps the model to identify key trends and patterns essential to predict the output.
- **Testing Data :** After the model is trained, it must be tested to evaluate how accurately it can predict an outcome. This is done by the testing data set.

1.1.4 Machine Learning Process

GQ. What are the various steps in machine learning process?

The Machine Learning process involves building a Predictive model that can be used to find a solution for a Problem Statement. To understand the Machine Learning process let's assume that a problem that needs to be solved by using Machine Learning. The problem is to predict the occurrence of rain in your local area by using Machine Learning. The below steps are followed in a Machine Learning process :

- **Step 1 :** Define the objective of the Problem Statement
  - At this step, we must understand what exactly needs to be predicted. In this case, the objective is to predict the possibility of rain by studying weather conditions.
  - At this stage, it is also essential to take mental notes on what kind of data can be used to solve this problem or the type of approach you must follow to get to the solution.
- **Step 2 :** Data Gathering
  - At this stage, the questions are such as,
    - What kind of data is needed to solve this problem?
    - Is the data available?
    - How can I get the data?
  - Once you know the types of data that is required, you must understand how you can derive this data. Data collection can be done manually or by web scraping.
  - The data needed for weather forecasting includes measures such as humidity level, temperature, pressure, locality, whether or not you live in a hill station, etc. Such data must be collected and stored for analysis.
- **Step 3 :** Data Preparation
  - **The data collected is almost never** in the right format. There will be a lot of inconsistencies in the data set such as missing values, redundant variables, duplicate values, etc.

- Removing such inconsistencies is very essential because they might lead to wrongful computations and predictions. Therefore, at this stage, the data set can be scanned for any inconsistencies and can be fixed then and there.

Step 4 : Exploratory Data Analysis

- EDA or Exploratory Data Analysis is the brainstorming stage of Machine Learning. Data Exploration involves understanding the patterns and trends in the data. At this stage, all the useful insights are drawn and correlations between the variables are understood.
- For example, in the case of predicting rainfall, we know that there is a strong possibility of rain if the temperature has fallen low. Such correlations must be understood and mapped at this stage.

Step 5 : Building a Machine Learning Model

- All the insights and patterns derived during Data Exploration are used to build the Machine Learning Model.
- This stage always begins by splitting the data set into two parts, training data, and testing data. The training data will be used to build and analyze the model. The logic of the model is based on the Machine Learning Algorithm that is being implemented.
- In the case of predicting rainfall, since the output will be in the form of True (if it will rain tomorrow) or False (no rain tomorrow), we can use a Classification Algorithm such as Logistic Regression.
- Choosing the right algorithm depends on the type of problem to be solved, the data set and the level of complexity of the problem.

Step 6 : Model Evaluation & Optimization

- After building a model by using the training data set, it is finally time to put the model to a test. The testing data set is used to check the efficiency of the model and how accurately it can predict the outcome.
- Once the accuracy is calculated, any further improvements in the model can be implemented at this stage. Methods like parameter tuning and cross-validation can be used to improve the performance of the model.

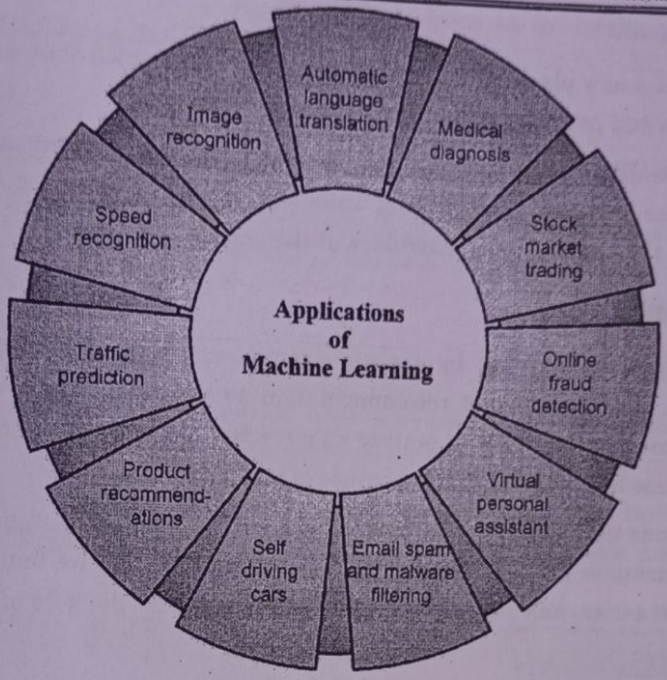
Step 7 : Predictions

- Once the model is evaluated and improved, it is finally used to make predictions. The final output can be a Categorical variable (e.g. True or False) or it can be a Continuous Quantity (e.g. the predicted value of a stock).
- In this case, for predicting the occurrence of rainfall, the output will be a categorical variable.

1.1.5 Applications of Machine Learning

GQ. State various applications of machine learning?

GQ. What are the various applications of machine learning in Mechanical Engineering?



(1A8)**Fig 1.1.2 : Applications of machine learning**

(1) Image Recognition

- Image recognition is one of the most common applications of machine learning. It is used to identify objects, persons, places, digital images, etc. The popular use case of image recognition and face detection is, Automatic friend tagging suggestion.
- Facebook provides us a feature of auto friend tagging suggestion. Whenever we upload a photo with our Facebook friends, then we automatically get a tagging suggestion with name, and the technology behind this is machine learning's face detection and recognition algorithm.
- It is based on the Facebook project named "Deep Face," which is responsible for face recognition and person identification in the picture.

(2) Speech Recognition

- While using Google, we get an option of "Search by voice," it comes under speech recognition, and it's a popular application of machine learning.
- Speech recognition is a process of converting voice instructions into text, and it is also known as "Speech to text", or "Computer speech recognition."
- At present, machine learning algorithms are widely used by various applications of speech recognition. Google assistant, Siri, Cortana, and Alexa are using speech recognition technology to follow the voice instructions.

(3) Traffic prediction

- If we want to visit a new place, we take help of Google Maps, which shows us the correct path with the shortest route and predicts the traffic conditions.
- It predicts the traffic conditions such as whether traffic is cleared, slow-moving, or heavily congested with the help of two ways: Real Time location of the vehicle form Google Map app and sensors. Average time has taken on past days at the same time.

(4) Product recommendations

- Machine learning is widely used by various e-commerce and entertainment companies such as Amazon, Netflix, etc., for product recommendation to the user. Whenever we search for some product on Amazon, then we started getting an advertisement for the same product while internet surfing on the same browser and this is because of machine learning.
- Google understands the user interest using various machine learning algorithms and suggests the product as per customer interest. As similar, when we use Netflix, we find some recommendations for entertainment series, movies, etc., and this is also done with the help of machine learning.

(5) Email Spam and Malware Filtering

- Whenever we receive a new email, it is filtered automatically as important, normal, and spam. We always receive an important mail in our inbox with the important symbol and spam emails in our spam box, and the technology behind this is Machine learning.
- Some machine learning algorithms such as Multi-Layer Perceptron, Decision tree, and Naïve Bayes classifier are used for email spam filtering and malware detection.

(6) Virtual Personal Assistant

- We have various virtual personal assistants such as Google assistant, Alexa, Cortana, Siri. As the name suggests, they help us in finding the information using our voice instruction.
- These assistants can help us in various ways just by our voice instructions such as Play music, call someone, open an email, scheduling an appointment, etc. These virtual assistants use machine learning algorithms as an important part.
- These assistant record our voice instructions, send it over the server on a cloud, and decode it using ML algorithms and act accordingly.

(7) Online Fraud Detection

- Machine learning is making our online transaction safe and secure by detecting fraud transaction.
- Whenever we perform some online transaction, there may be various ways that a fraudulent transaction can take place such as fake accounts, fake ids, and steal money in the middle of a

- So, to detect this, Feed Forward Neural network helps us by checking whether it is a genuine transaction or a fraud transaction.

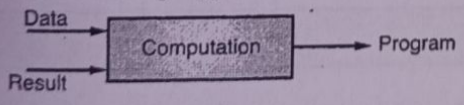
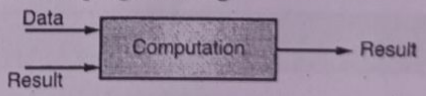
(8) Stock Market trading

- Machine learning is widely used in stock market trading.
- In the stock market, there is always a risk of up and downs in shares, so for this machine learning's long short term memory neural network is used for the prediction of stock market trends.

(9) Automatic Language Translation

- Nowadays, if we visit a new place and we are not aware of the language then it is not a problem at all, as for this also machine learning helps us by converting the text into our known languages.
- Google's GNMT (Google Neural Machine Translation) provide this feature, which is a Neural Machine Learning that translates the text into our familiar language, and it called as automatic translation.

1.2 COMPARISON OF MACHINE LEARNING WITH TRADITIONAL PROGRAMMING

Sr. No.	Machine Learning	Traditional Programming
1.	Machine learning is not a manual process here the algorithm automatically formulate the rules from the data.	Traditional programming is a manual process. – i.e. a person or programmer creates the program. Without anyone has to manually formulate or code rules.
2.	Machine learning approach  **Fig. A**	Traditional programming  **Fig.B**
3.	With a subset of Artificial Intelligence (AI), machine learning is motivated by human learning behaviour. Here we show the examples and machine figures out how to solve the problem by itself.	In T.P., we write down the exact steps required to solve the problem.
4.	A machine learning algorithm takes an input and output and gives some logic which can be used to work with new input to give an output.	A traditional algorithm takes some input and some logic in the form of code and gives the output.

1.3 ML VS AI VS DATA SCIENCE

- ML vs AI vs Data Science are interconnected but have different scopes. They follow different approaches and produce different results depending on the problem.
- Here we shall see how ML, AI and Data Science differ from each other.

Aspects	Machine Learning	Artificial Intelligence	Data Science
Job roles	Machine Learning, Engineer, Data Architect, Data scientist, Data Mining specialist, cloud Architect and cyber security Analyst, and more.	Machine learning Engineer, Data scientist, Business intelligence. Developer, Big Data Architect, Research Scientist.	Data engineer, Data scientist, Data Analyst, Data Architect, Database Administrator, Machine learning engineer, statistician, Business Analyst, Data and Analytics Manager
Skills	Statistics, Probability Data Modelling. Programming skills, Applying ML Libraries and algorithms. Software design python.	Mathematical and Algorithms skills Probability and statistics Knowledge, Expertise in programming Awareness about Advanced Signal Processing techniques well versed with unix Tools.	Programming skills. Statistics Machine Learning, Multi-variable calculus and linear Algorithm, Data visualisation and Communication Software Engineering Data Intuition.
Salary	1123 k/year Average base pay	14.3 lakhs per annum	1050 k/year Average base pay

1.4 TYPES OF LEARNING

With the constant advancements in artificial intelligence, the field has become too big to specialize in all together. There are countless problems that can be solved with countless methods. Knowledge of an experienced AI researcher specialized in one field may mostly be useless for another field. Understanding the nature of different machine learning problems is very important. Even though the list of machine learning problems is very long, these problems can be grouped into three different learning approaches:

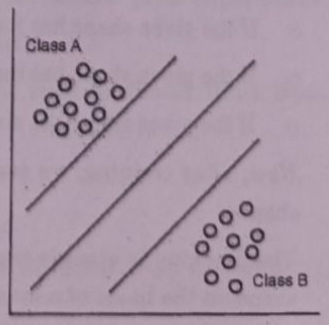
1. Supervised Learning;
2. Unsupervised Learning;
3. Reinforcement Learning.

Top machine learning approaches are categorized depending on the nature of their feedback mechanism for learning. Most of the machine learning problems may be addressed by adopting one of these approaches.

1.5 SUPERVISED LEARNING

GQ. What is supervised learning?
GQ. Explain supervised learning with the help of an example.
GQ. How supervised learning works?

- Learning that takes place based on a class of examples is referred to as supervised learning. It is learning based on labelled data. In short, while learning, the system has knowledge of a set of labelled data. This is one of the most common and frequently used learning methods
- The supervised learning method is comprised of a series of algorithms that build mathematical models of certain data sets that are capable of containing both inputs and the desired outputs for that particular machine.
- The data being inputted into the supervised learning method is known as training data, and essentially consists of training examples which contain one or more inputs and typically only one desired output. This output is known as a "supervisory signal."
- In the training examples for the supervised learning method, the training example is represented by an array, also known as a vector or a feature vector, and the training data is represented by a matrix.
- The algorithm uses the iterative optimization of an objective function to predict the output that will be associated with new inputs.
- Ideally, if the supervised learning algorithm is working properly, the machine will be able to correctly determine the output for the inputs that were not a part of the training data.
- Supervised learning uses classification and regression techniques to develop predictive models. Classification techniques predict categorical responses,
- Regression techniques predict continuous responses, for example, changes in temperature or fluctuations in power demand. Typical applications include electricity load forecasting and algorithmic trading.
- Let us begin by considering the simplest machine-learning task : supervised learning for classification. Let us take an example of classification of documents. In this particular case a learner learns based on the available documents and their classes. This is also referred to as labelled data.
- The program that can map the input documents to appropriate classes is called a classifier, because it assigns a class (i.e., document type) to an object (i.e., a document). The task of supervised learning is to construct a classifier given a set of classified training examples. A typical classification is depicted in Fig. 1.5.1.
- Fig. 1.5.1 represents a hyperplane that has been generated after learning, separating two classes - class A and class B in different parts. Each input point presents input–output instance from sample space. In case of document classification, these points are documents.

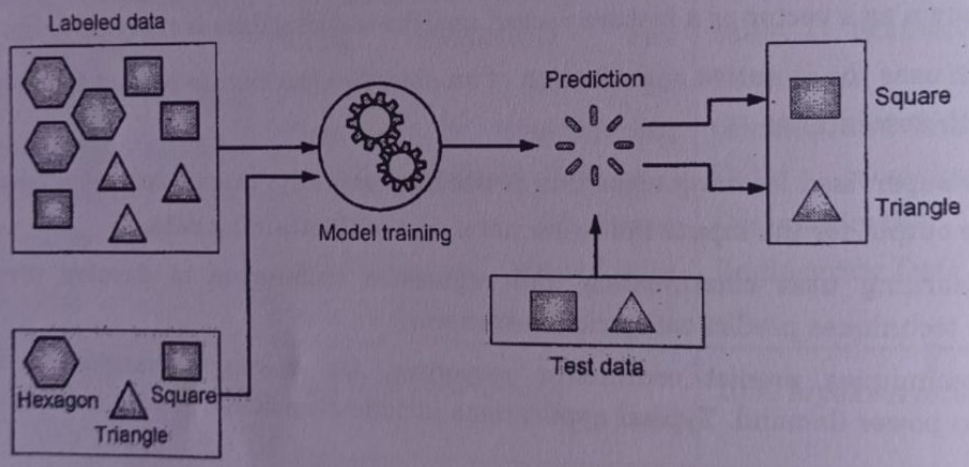


(1D1) **Fig. 1.5.1 : Supervised learning**

- Learning computes a separating line or hyperplane among documents. An unknown document type will be decided by its position with respect to a separator.
- There are a number of challenges in supervised classification such as generalization, selection of right data for learning, and dealing with variations. Labelled examples are used for training in case of supervised learning. The set of labelled examples provided to the learning algorithm is called the *training set*.
- Supervised learning is not just about classification, but it is the overall process that with guidelines maps to the most appropriate decision.

1.5.1 How Supervised Learning Works?

- In supervised learning, models are trained using labelled dataset, where the model learns about each type of data. Once the training process is completed, the model is tested on the basis of test data (a subset of the training set), and then it predicts the output.
- The working of Supervised learning can be easily understood by the below example and diagram (Fig. 1.5.2).



(102)**Fig. 1.5.2 How Supervised learning works?**

- Suppose we have a dataset of different types of shapes which includes square, rectangle, triangle, and Polygon. Now the first step is that we need to train the model for each shape.
  - If the given shape has four sides, and all the sides are equal, then it will be labelled as a **Square**.
  - If the given shape has three sides, then it will be labelled as a **triangle**.
  - If the given shape has six equal sides then it will be labelled as **hexagon**.
- Now, after training, we test our model using the test set, and the task of the model is to identify the shape.
- The machine is already trained on all types of shapes, and when it finds a new shape, it classifies the shape on the bases of a number of sides, and predicts the output.
- Following are the steps involved in Supervised Learning :
  - First Determine the type of training dataset

- Collect/Gather the labelled training data.
- Split the training dataset into training **dataset, test dataset, and validation dataset**.
- Determine the input features of the training dataset, which should have enough knowledge so that the model can accurately predict the output.
- Determine the suitable algorithm for the model, such as support vector machine, decision tree, etc.
- Execute the algorithm on the training dataset. Sometimes we need validation sets as the control parameters, which are the subset of training datasets.
- Evaluate the accuracy of the model by providing the test set. If the model predicts the correct output, which means our model is accurate.

- Supervised learning can be further divided into two types of problems: Regression and Classification.

Regression

Regression algorithms are used if there is a relationship between the input variable and the output variable. It is used for the prediction of continuous variables, such as Weather forecasting, Market Trends, etc. Below are some popular Regression algorithms which come under supervised learning :

- Linear Regression
- Regression Trees
- Non-Linear Regression
- Bayesian Linear Regression
- Polynomial Regression

Classification

Classification algorithms are used when the output variable is categorical, which means there are two classes such as Yes-No, Male-Female, True-false, etc.

- Random Forest
- Logistic Regression
- Decision Trees
- Support vector Machines

1.5.2 Advantages of Supervised Learning

(1) With the help of supervised learning, the model can predict the output on the basis of prior experiences.

(2) In supervised learning, we can have an exact idea about the classes of objects.

(3) Supervised learning model helps us to solve various real-world problems such as **fraud detection, spam filtering**, etc.

1.5.3 Disadvantages of Supervised Learning

(1) Supervised learning models are not suitable for handling the complex tasks.

(2) Supervised learning cannot predict the correct output if the test data is different from the training dataset.

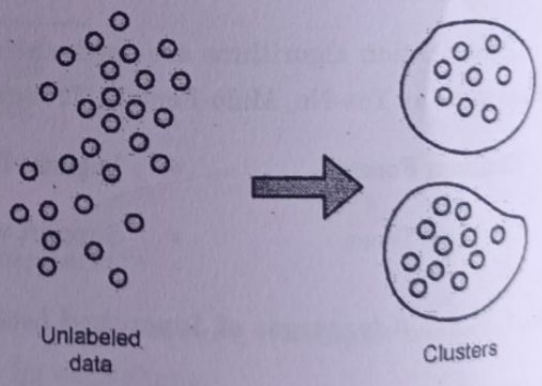
(3) Training required lots of computation times.

(4) In supervised learning, we need enough knowledge about the classes of object.

1.6 UNSUPERVISED LEARNING

GQ. What is Unsupervised Learning?
GQ. What are the types of unsupervised learning?
GQ. What are the advantages and disadvantages of unsupervised learning?

- Unsupervised learning refers to learning from unlabeled data. It is based more on similarity and differences than on anything else. In this type of learning, all similar items are clustered together in a particular class where the label of a class is not known.
- It is not possible to learn in a supervised way in the absence of properly labeled data. In these scenarios there is need to learn in an unsupervised way. Here the learning is based more on similarities and differences that are visible. These differences and similarities are mathematically represented in unsupervised learning.
- Given a large collection of objects, we often want to be able to understand these objects and visualize their relationships. For an example based on similarities, a kid can separate birds from other animals. It may use some property or similarity while separating, such as the birds have wings.
- The criterion in initial stages is the most visible aspects of those objects. Linnaeus devoted much of his life to arranging living organisms into a hierarchy of classes, with the goal of arranging similar organisms together at all levels of the hierarchy. Many unsupervised learning algorithms create similar hierarchical arrangements based on similarity-based mappings.
- The task of hierarchical clustering is to arrange a set of objects into a hierarchy such that similar objects are grouped together. Non-hierarchical clustering seeks to partition the data into some number of disjoint clusters. The process of clustering is depicted in Fig. 1.6.1.
- A learner is fed with a set of scattered points, and it generates two clusters with representative centroids after learning. Clusters show that points with similar properties and closeness are grouped together.



(1D3) **Fig. 1.6.1 : Unsupervised learning**

- Unsupervised learning is a set of algorithms where the only information being uploaded is inputs. The device itself, then, is responsible for grouping together and creating ideal outputs based on the data it discovers. Often, unsupervised learning algorithms have certain goals, but they are not controlled in any manner.
- Instead, the developers believe that they have created strong enough inputs to ultimately program the machine to create stronger results than they themselves possibly could. The idea here is that the machine is programmed to run flawlessly to the point where it can be intuitive and inventive in the most effective manner possible.

- The information in the algorithms being run by unsupervised learning methods is not labelled, classified, or categorized by humans. Instead, the unsupervised algorithm rejects responding to feedback in favour of identifying commonalities in the data. It then reacts based on the presence, or absence, of such commonalities in each new piece of data that is being inputted into the machine itself.
- It is used to draw inferences from datasets consisting of input data without labelled responses. Clustering is the most common unsupervised learning technique. It is used for exploratory data analysis to find hidden patterns or groupings in data. Applications for clustering include gene sequence analysis, market research, and object recognition.

1.6.1 Types of Unsupervised Learning Algorithm

The unsupervised learning algorithm can be further categorized into two types of problems:

1. Clustering
2. Association

(1) Clustering

Clustering is a method of grouping the objects into clusters such that objects with most similarities remains into a group and has less or no similarities with the objects of another group. Cluster analysis finds the commonalities between the data objects and categorizes them as per the presence and absence of those commonalities.

(2) Association

An association rule is an unsupervised learning method which is used for finding the relationships between variables in the large database. It determines the set of items that occurs together in the dataset. Association rule makes marketing strategy more effective. Such as people who buy X item (suppose a bread) are also tend to purchase Y (Butter/Jam) item. A typical example of Association rule is Market Basket Analysis.

Below is the list of some popular unsupervised learning algorithms :

- K-means clustering
- KNN (k-nearest neighbors)
- Hierarchal clustering
- Anomaly detection
- Neural Networks
- Principle Component Analysis
- Independent Component Analysis
- Apriori algorithm
- Singular value decomposition

1.6.2 Advantages of Unsupervised Learning

(1) Unsupervised learning is used for more complex tasks as compared to supervised learning because, in unsupervised learning, we don't have labeled input data.

(2) Unsupervised learning is preferable as it is easy to get unlabeled data in comparison to labeled data.

1.6.3 Disadvantages of Unsupervised Learning

(1) Unsupervised learning is intrinsically more difficult than supervised learning as it does not have corresponding output.

(2) The result of the unsupervised learning algorithm might be less accurate as input data is not labeled, and algorithms do not know the exact output in advance.

In practical scenarios there is always need to learn from both labeled and unlabeled data. Even while learning in an unsupervised way, there is the need to make the best use of labeled data available. This is referred to as semi supervised learning. Semi supervised learning is making the best use of two paradigms of learning - that is, learning based on similarity and learning based on inputs from a teacher. Semi supervised learning tries to get the best of both the worlds.

1.6.4 Difference between Supervised and Unsupervised Learning

GQ. What is the difference between supervised learning and unsupervised learning?

- Supervised and Unsupervised learning are the two techniques of machine learning. But both the techniques are used in different scenarios and with different datasets. Below the explanation of both learning methods along with their difference table is given.
- Supervised learning is a machine learning method in which models are trained using labeled data. In supervised learning, models need to find the mapping function to map the input variable (X) with the output variable (Y).

$$Y = f(X)$$

- Supervised learning needs supervision to train the model, which is similar to as a student learns things in the presence of a teacher. Supervised learning can be used for two types of problems: Classification and Regression.
- **Example :** Suppose we have an image of different types of fruits. The task of our supervised learning model is to identify the fruits and classify them accordingly. So to identify the image in supervised learning, we will give the input data as well as output for that, which means we will train the model by the shape, size, color, and taste of each fruit. Once the training is completed, we will test the model by giving the new set of fruit. The model will identify the fruit and predict the output using a suitable algorithm.
- Unsupervised learning is another machine learning method in which patterns inferred from the unlabeled input data. The goal of unsupervised learning is to find the structure and patterns from the input data. Unsupervised learning does not need any supervision. Instead, it finds patterns from the data by its own.
- Unsupervised learning can be used for two types of problems: Clustering and Association.
- **Example :** To understand the unsupervised learning, we will use the example given above. So unlike supervised learning, here we will not provide any supervision to the model. We will just provide the input dataset to the model and allow the model to find the patterns from the data. With the help of a suitable algorithm, the model will train itself and divide the fruits into different groups according to the most similar features between them.

- **The main differences between Supervised and Unsupervised learning are given following Table 1.6.1 :**

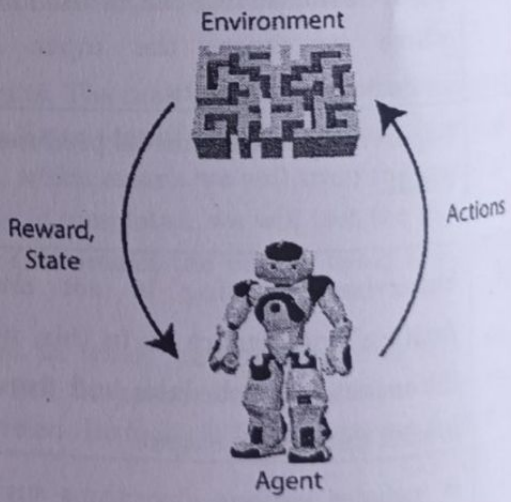
Table 1.6.1

Sr. No.	Supervised Learning	Unsupervised Learning
1.	Supervised learning algorithms are trained using labeled data.	Unsupervised learning algorithms are trained using unlabeled data.
2.	Supervised learning model takes direct feedback to check if it is predicting correct output or not.	Unsupervised learning model does not take any feedback.
3.	Supervised learning model predicts the output.	Unsupervised learning model finds the hidden patterns in data.
4.	In supervised learning, input data is provided to the model along with the output.	In unsupervised learning, only input data is provided to the model.
5.	The goal of supervised learning is to train the model so that it can predict the output when it is given new data.	The goal of unsupervised learning is to find the hidden patterns and useful insights from the unknown dataset.
6.	Supervised learning needs supervision to train the model.	Unsupervised learning does not need any supervision to train the model.
7.	Supervised learning can be categorized in Classification and Regression problems.	Unsupervised Learning can be classified in Clustering and Associations problems.
8.	Supervised learning can be used for those cases where we know the input as well as corresponding outputs.	Unsupervised learning can be used for those cases where we have only input data and no corresponding output data.
9.	Supervised learning model produces an accurate result.	Unsupervised learning model may give less accurate result as compared to supervised learning.
10.	Supervised learning is not close to true Artificial intelligence as in this, we first train the model for each data, and then only it can predict the correct output.	Unsupervised learning is more close to the true Artificial Intelligence as it learns similarly as a child learns daily routine things by his experiences.
11.	It includes various algorithms such as Linear Regression, Logistic Regression, Support Vector Machine, Multi-class Classification, Decision tree, Bayesian Logic, etc.	It includes various algorithms such as Clustering, KNN, and Apriori algorithm.

1.7 REINFORCEMENT LEARNING

GQ. What is Reinforcement Learning? Explain with an example.

- Reinforcement Learning is a feedback-based Machine learning technique in which an agent learns to behave in an environment by performing the actions and seeing the results of actions. For each good action, the agent gets positive feedback, and for each bad action, the agent gets negative feedback or penalty.
- In Reinforcement Learning, the agent learns automatically using feedbacks without any labeled data, unlike supervised learning.
- Since there is no labelled data, so the agent is bound to learn by its experience only.
- RL solves a specific type of problem where decision making is sequential, and the goal is long-term, such as game-playing, robotics, etc.
- The agent interacts with the environment and explores it by itself. The primary goal of an agent in reinforcement learning is to improve the performance by getting the maximum positive rewards.
- The agent learns with the process of hit and trial, and based on the experience, it learns to perform the task in a better way. Hence, we can say that "Reinforcement learning is a type of machine learning method where an intelligent agent (computer program) interacts with the environment and learns to act within that." How a Robotic dog learns the movement of his arms is an example of Reinforcement learning.
- It is a core part of Artificial intelligence, and all AI agent works on the concept of reinforcement learning. Here we do not need to pre-program the agent, as it learns from its own experience without any human intervention.
- **Example :** Suppose there is an AI agent present within a maze environment, and his goal is to find the diamond. The agent interacts with the environment by performing some actions, and based on those actions, the state of the agent gets changed, and it also receives a reward or penalty as feedback.
- The agent continues doing these three things (take action, change state/remain in the same state, and get feedback), and by doing these actions, he learns and explores the environment.



(1D4) **Fig. 1.7.1**

- The agent learns that what actions lead to positive feedback or rewards and what actions lead to negative feedback penalty. As a positive reward, the agent gets a positive point, and as a penalty, it gets a negative point.

- For machine learning, the environment is typically represented by an "MDP" or Markov Decision Process.
- These algorithms do not necessarily assume knowledge, but instead are used when exact models are infeasible. In other words, they are not quite as precise or exact, but they will still serve a strong method in various applications throughout different technology systems.
- The key features of Reinforcement Learning are mentioned below.
  - In RL, the agent is not instructed about the environment and what actions need to be taken.
  - It is based on the hit and trial process.
  - The agent takes the next action and changes states according to the feedback of the previous action.
  - The agent may get a delayed reward.
  - The environment is stochastic, and the agent needs to explore it to reach to get the maximum positive rewards.

1.7.1 Approaches to Implement Reinforcement Learning

GQ. What are the approaches for Reinforcement Learning?

There are mainly three ways to implement reinforcement-learning in ML, which are :

1. **Value-based :** The value-based approach is about to find the optimal value function, which is the maximum value at a state under any policy. Therefore, the agent expects the long-term return at any state(s) under policy π.
2. **Policy-based :** Policy-based approach is to find the optimal policy for the maximum future rewards without using the value function. In this approach, the agent tries to apply such a policy that the action performed in each step helps to maximize the future reward. The policy-based approach has mainly two types of policy :
   - **Deterministic :** The same action is produced by the policy (π) at any state.
   - **Stochastic :** In this policy, probability determines the produced action.
3. **Model-based :** In the model-based approach, a virtual model is created for the environment, and the agent explores that environment to learn it. There is no particular solution or algorithm for this approach because the model representation is different for each environment.

Here are important characteristics of reinforcement learning

- There is no supervisor, only a real number or reward signal
- Sequential decision making
- Time plays a crucial role in Reinforcement problems
- Feedback is always delayed, not instantaneous
- Agent's actions determine the subsequent data it receives

RL can be used in almost any application. It is a learning based on experience algorithm, a decision maker algorithm, an algorithm that learns autonomously, an optimization algorithm that over time learns to maximize its reward, the reward can be defined by the engineer to reach the objective of the problem.

1.7.2 Challenges of Reinforcement Learning

Here are the major challenges you will face while doing Reinforcement earning :

(1) Feature/reward design which should be very involved

(2) Parameters may affect the speed of learning.

(3) Realistic environments can have partial observability.

(4) Too much Reinforcement may lead to an overload of states which can diminish the results.

(5) Realistic environments can be non-stationary.

1.7.3 Applications of Reinforcement Learning

Here are applications of Reinforcement Learning :

(1) Robotics for industrial automation.

(2) Business strategy planning

(3) Machine learning and data processing

(4) Aircraft control and robot motion control

(5) It helps you to create training systems that provide custom instruction and materials according to the requirement of students.

1.7.4 Reinforcement Learning vs. Supervised Learning

GQ. What is the difference between Reinforcement Learning and Supervised Learning ?

Table 1.7.1

Parameters	Reinforcement Learning	Supervised Learning
Decision style	Reinforcement learning helps you to take your decisions sequentially.	In this method, a decision is made on the input given at the beginning.
Works on	Works on interacting with the environment.	Works on examples or given sample data.
Dependency on decision	In RL method learning decision is dependent. Therefore, you should give labels to all the dependent decisions.	Supervised learning the decisions which are independent of each other, so labels are given for every decision.
Best suited	Supports and work better in AI, where human interaction is prevalent.	It is mostly operated with an interactive software system or applications.
Example	Chess game	Object recognition

1.8 INTRODUCTION TO SEMI-SUPERVISED LEARNING

(1) Semi-Supervised learning is a type of Machine Learning algorithm that represents the intermediate ground between Supervised and Unsupervised learning algorithms. It uses the combination of labeled and unlabeled datasets during the training period.

(2) Before understanding the Semi-Supervised learning, you should know the main categories of Machine Learning algorithms. Machine Learning consists of three main categories: Supervised Learning, Unsupervised Learning, and Reinforcement Learning

(3) Further, the basic difference between Supervised and unsupervised learning is that supervised learning datasets consist of an output label training data associated with each tuple, and unsupervised datasets do not consist the same. Semi-supervised learning is an important category that lies between the Supervised and Unsupervised machine learning. Although Semi-supervised learning is the middle ground between supervised and unsupervised learning and operates on the data that consists of a few labels, it mostly consists of unlabeled data. As labels are costly, but for the corporate purpose, it may have few labels.

(4) The basic disadvantage of supervised learning is that it requires hand-labeling by ML specialists or data scientists, and it also requires a high cost to process. Further unsupervised learning also has a limited spectrum for its applications. To overcome these drawbacks of supervised learning and unsupervised learning algorithms, the concept of Semi-supervised learning is introduced. In this algorithm, training data is a combination of both labeled and unlabeled data. However, labeled data exists with a very small amount while it consists of a huge amount of unlabeled data. Initially, similar data is clustered along with an unsupervised learning algorithm, and further, it helps to label the unlabeled data into labeled data. It is why label data is a comparatively, more expensive acquisition than unlabeled data.

(5) We can imagine these algorithms with an example. Supervised learning is where a student is under the supervision of an instructor at home and college. Further, if that student is self-analyzing the same concept without any help from the instructor, it comes under unsupervised learning. Under semi-supervised learning, the student has to revise itself after analyzing the same concept under the guidance of an instructor at college.

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

1. **Continuity Assumption :** The algorithm assumes that the points which are closer to each other are more likely to have the same output label.
2. **Cluster Assumption :** The data can be divided into discrete clusters and points in the same cluster are more likely to share an output label.
3. **Manifold Assumption :** The data lie on a manifold of much lower dimension than the input space.

This assumption allows the use of distances and densities which are defined on a manifold.

Practical Applications of Semi Semi-Supervised learning :

1. **Speech Analysis :** Since labelling of audio files is very intensive task, semi-supervised learning is a natural approach to solve this problem.
2. **Internet Content Classification :** Labeling each webpage is impractical and unfeasible process and uses Semi-Supervised learning algorithms.
3. **Protein- Sequence Classification :** Since DNA strands are very large in size, hence here Semi-Supervised learning is must in this field.

1.9 MODELS OF MACHINE LEARNING

1.9.1 Geometric Models

- In Geometric models, features could be described as points in two dimensions (x- and y-axis) or a three-dimensional space (x, y, and z). Even when features are not intrinsically geometric, they could be modelled in a geometric manner (for example, temperature as a function of time can be modelled in two axes). In geometric models, there are two ways we could impose similarity.
- We could use geometric concepts like lines or planes to segment (classify) the instance space. These are called Linear models
- Alternatively, we can use the geometric notion of distance to represent similarity. In this case, if two points are close together, they have similar values for features and thus can be classed as similar. We call such models as Distance-based models

1.9.2 Probabilistic Models

- In contrast to **deterministic models**, where the relationship between quantities is already known, Probabilistic models are based on the assumption that relationship between quantities which is reasonably accurate but other components are also taken into consideration.
- Thus probabilistic models are statistical models, which give probability distributions to account for these components.
- Probabilistic models from the basis in other areas such as machine learning, artificial intelligence, and data analysis. Their formulation and solution rest on the two basic rules of probability theory, that is, the sum rule and product rule.
- **We mention an example :** if one lives in a cold climate, one knows that traffic tends to be more difficult when snow falls and covers the roads.
- We can go a step further and make a hypothesis : There will be a strong correlation between snowy weather and increased traffic incidents.
- Probabilistic models are used in a variety of disciplines, including statistical physics, quantum mechanics and theoretical computer science.

1.9.3 Logical Models

- Logical models use a logical expression to divide the instance space into segments and hence construct grouping models. A logical expression is an expression that returns a Boolean value, i.e., a True or False outcome. Once the data is grouped using a logical expression, the data is divided into homogeneous groupings for the problem we are trying to solve.
- There are two types of logical models: Tree models and Rule models.
    - Rule models consist of a collection of implications or IF-THEN rules. For tree-based models, the 'if-part' defines a segment and the 'then-part' defines the behaviour of the model for this segment. Rule models follow the same reasoning
    - Tree models can be seen as a particular type of rule model where the if- parts of the rules are organised in a tree structure. Both Tree models and Rule models use the same approach to supervised learning

1.9.4 Groping Models

- Tree models repeatedly split the instance space into smaller subsets.
- Trees are usually of limited depth and don't contain all the available features.
- Subsets at the leaves of the tree partition the instance space with some finite resolution.
- Instances filtered into the same leaf of the tree are treated the same, regardless of any features not in the tree that might be able to distinguish them.

1.9.5 Grading Model

- They don't use the notion of segment
- Forms one global model over instance space
- Grading models are (usually) able to distinguish between arbitrary instances, no matter how similar they are
- Resolution is, in theory, infinite, particularly when working in a Cartesian instance space.
- Support vector machines and other geometric classifiers are examples of grading models.
- They work in a Cartesian instance space.
- Exploit the minutest differences between instances

1.9.6 Groping and Grading Models

- The key difference between grouping and grading models is the way they handle the instance space

Grouping models

- Grouping models break up the instance space into groups or segments, the number of which is determined at training time.

- They have fixed resolution cannot distinguish instances beyond a resolution
- At the finest resolution grouping models assign the majority class to all instances that fall into the segment.
- Determine the right segments and label all the objects in that segment.

1.9.7 Groping versus Grading Models

- Some models combine the features of both grouping and grading models.
- Linear classifiers are a prime example of a grading model.
- Instances on a line or plane parallel to the decision boundary can't be distinguished by a linear model.
- There are infinitely many segments.

1.10 PARAMETRIC AND NON-PARAMETRIC MODELS

- Machine learning models can be parametric or non-parametric.
- Parametric models are those that require the application of some parameters before they can be used to make predictions.
- Non-parametric models do not rely on any specific parameter setting and hence they often produce more accurate results.

We mention the difference between Parametric and Non-Parametric Methods.

Sr. No.	Parametric Methods	Non-Parametric Methods
1.	Parametric methods use a fixed number of parameters to build the model.	Non-parametric methods use flexible number of parameters to build the model.
2.	Parametric analysis is for testing group means.	A non-parametric analysis is for testing medians.
3.	It is applicable only for variables.	It is applicable for both variable and Attribute.
4.	It always considers strong assumptions about data.	It generally considers fewer assumptions and data.
5.	Parametric methods require lesser data than Non-parametric methods.	Non-parametric methods require much more data than parametric methods.
6.	Parametric data handles intervals data or ration data.	Non-parametric methods handle original data.
7.	Parametric methods follow normal distribution.	There is no assumed distribution in non-parametric methods.

Sr. No.	Parametric Methods	Non-Parametric Methods
8.	The output generated by parametric methods can be easily affected by outliers.	The output generated cannot be seriously affected by outliers.
9.	Parametric methods function well in many situations but its performance is at peak (top) when the spread of each group is different.	Non-parametric Methods can perform well in many situations but its performance is at the top when the spread of each group is the same.
10.	Parametric Methods have more statistical power than Non-Parametric methods.	Non-parametric methods have less statistical power than parametric methods.
11.	As for as the computation is concerned, these methods are computationally faster than the Non-parametric methods. **Examples :** Logistic Regression, Naive Bayes Model etc.	As far as the computation is concerned, these methods are computationally slower than the parametric methods. **Examples :** KNN, Decision Tree Model, etc.

1.11 IMPORTANT ELEMENTS OF MACHINE LEARNING

1.11.1 Data Formats

- Each data format represents how the input data is represented in memory.
- Each machine learning application performs well for a particular data format and worse for others. Choosing the correct format is a major **optimisation technique.**
- There are four types of data formats ; which are commonly used.

(1) NHWC
(2) NCHW
(3) NCDHW
(4) NDHWC

Each letter in the formats denotes a particular aspect or dimension of the data :

(i) **N : Batch size :** is the number of images passed together as a group for inference.

(ii) **C : Channel :** is the number of data components that make a data point for the input data. It is 3 for opaque images and 4 for transparent images.

(iii) **W : Width :** is the width / measurement in x – axis of the input data.

(iv) **H : Height :** Is the height / measurement in y – axis of the input data.

(v) **D : Depth :** is the depth of the input data.

(1) NHWC

NHWC, denotes (Batch size, Height, Width, Channel). This implies that there is a 4D array where the first dimension represents batch size and accordingly. This 4D array is laid out in memory in row - major order. [commonly used data : images]

(2) NCHW

NCHW denotes (Batch Size, Channel, Height, Width). This means that there is a 4D array where the first dimension represents batch size and so on.

This 4D array is laid out in memory in row–major order. [Commonly used data – images]

(3) NCDHW

NCDHW denotes (Batch Size, Channel, Depth, Height, Width). This means that there is a 5D array where the first dimension represents batch size and so on. This 5D array is laid out in memory in row major order. [Commonly used data : Video]

(4) NDHWC

NDHWC denotes (Batch Size, Depth, Height, Width, Channel). This means there is a 5D array where the first dimension represents batch size and accordingly.

This 5D array is laid out in memory in row major order.

Commonly used data : Video

Software : Tensor flow

1.11.2 Learnability

- Learnability is a quality of products and interfaces that allows users to become familiar with them. It makes good use of all their features and capabilities.
- A very learnable product is sometimes said to be intuitive because the user can immediately grasp how to interact with the system.
- First time lerarnability refers to the degree of ease with which a user can learn a newly developed system without referring to its documentation, e.g., manuals, user guides or frequently asked questions (FAQ) lists.
- One elements of first – time learnability is discoverability, i.e. ; the degree of ease with which the user can find all the elements and features of the new system.
- Learnability over time is the capacity of a user to gain experience in working with a given system through repeated interaction.
- Comparatively simple systems with good learnability are said to have short or steep learning curves. It implies that most learning associated with the system happens quickly.
- More complex systems involve a longer learning curve.
- In software testing **learnability**, according to ISO 9126, is the **capability** of a software product to enable the user to learn how to use it.
- Lernability is considered as an aspect of **usability** and is of major **concern** in the **design** of complex software applications.

- In computational learning theory, **learnability** is the mathematical analysis of machine learning. It is also employed in **language acquisition** in arguments within linguistics.
- The skill of learnability confers a future value by making one agile (active). It is a **currency** that is rewarded with better employability and high growth prospects. Learning does not and with school or college

1.11.3 Statistical learning Approaches

- **Statistical learning theory** is a tramework for machine learning – drawing from the fields of statistics and functional analysis.
- Statistical learning theory deals with the statistical inference problem of finding a predictive function based on data.
- Statistical learning theory has led to successful applications in fields such as computer vision, speech recognition and bioinformatics.
- Statistical learning is a set of tool for understanding data.
- These tools come under two classes : Supervised learning and Unsupervised learning.
- Statistical learning is mathematical intensive which is based on the coefficient estimator and requires a good understanding of data.
- On the other hand, Machine Learning identifies patterns from the dataset through it the iterations which does not require much human effort.

Lexical Acquisition

- The role of statistical learning in language acquisition is well documented in **lexical acquisition.**
- One important contribution to infant's understanding of segmenting words from a speech is their ability to recognize statistical regularities of the speech, that is heard from their environment.

Statistical Algorithm

- Statistical algorithms **create a statistical model of the input data,** which is in most cases represented as a probabilistic tree data structure.
- Subsequences with a higher frequency are represented with shorter codes.
- The type of algorithm used is linear regression. It is the popular algorithm in machine learning and statistics.
- This model will assume a linear relationship between the input and the output variable.
- It is represented in the form of linear equation which has a set of inputs and a predictive output.

Types of statistical analysis

(i) Descriptive statistical analysis. (ii) Inferential statistical analysis,
(iii) Associational statistical analysis ; (iv) Predictive analysis,
(v) Prescriptive analysis (vi) Exploratory data analysis
(vii) Causal analysis (viii) Data collection

Chapter Ends ...

❑❑❑

UNIT II

CHAPTER 2

Feature Engineering

Syllabus

Concept of Feature, Preprocessing of data, Normalization and Scaling, Standardization, Managing Missing Values, Introduction to Dimensionality Reduction, Principal Component Analysis (PCA), Feature Extraction: Kernel PCA, Local Binary Pattern.

Introduction to Various Feature Selection Techniques; Sequential Forward Selection, Sequential Backward Selection Statistical Feature Engineering: count-based, Length, Mean, Median, Mode, etc., based Feature vector creation.

Multidimensional scaling, Matrix Factorization Techniques

2.1 CONCEPT OF FEATURE

GQ. Define Feature Engineering. Explain the four processes in feature engineering.

- Feature engineering is the pre-processing step of machine learning, which is used to transform raw data into features that can be used for creating a predictive model using Machine learning or statistical Modelling.
- Feature engineering is the pre-processing step of machine learning, which extracts features from raw data.
- It helps to represent an underlying problem to predictive models in a better way, which as a result, improve the accuracy of the model for unseen data.
- The predictive model contains predictor variables and an outcome variable, and while the feature engineering process selects the most useful predictor variables for the model.
- Generally, all machine learning algorithms take input data to generate the output. The input data remains in a tabular form consisting of rows (instances or observations) and columns (variable or attributes), and these attributes are often known as **features**.
- For example, an image is an instance in computer vision, but a line in the image could be the feature. Similarly, in NLP, a document can be an observation, and the word count could be the feature.
- So, we can say a feature is an attribute that impacts a problem or is useful for the problem.
- Feature engineering in ML contains mainly four processes: Feature Creation, Transformations, Feature Extraction, and Feature Selection.
- These processes are described as below :

(1) **Feature Creation :** Feature creation is finding the most useful variables to be used in a predictive model. The process is subjective, and it requires human creativity and intervention. The new features are created by mixing existing features using addition, subtraction, and ration, and these new features have great flexibility.

(2) **Transformations :** The transformation step of feature engineering involves adjusting the predictor variable to improve the accuracy and performance of the model. For example, it ensures that the model is flexible to take input of the variety of data; it ensures that all the variables are on the same scale, making the model easier to understand. It improves the model's accuracy and ensures that all the features are within the acceptable range to avoid any computational error.

(3) **Feature Extraction :** Feature extraction is an automated feature engineering process that generates new variables by extracting them from the raw data. The main aim of this step is to reduce the volume of data so that it can be easily used and managed for data modelling. Feature extraction methods include cluster analysis, text analytics, edge detection algorithms, and principal components analysis (PCA).

(4) Feature Selection : While developing the machine learning model, only a few variables in the dataset are useful for building the model, and the rest features are either redundant or irrelevant. If we input the dataset with all these redundant and irrelevant features, it may negatively impact and reduce the overall performance and accuracy of the model. Hence it is very important to identify and select the most appropriate features from the data and remove the irrelevant or less important features, which is done with the help of feature selection in machine learning. "Feature selection is a way of selecting the subset of the most relevant features from the original feature set by removing the redundant, irrelevant, or noisy features."

2.2 PREPROCESSING OF DATA

GQ. Define data preprocessing. Explain the steps involved in data preprocessing.

- Data preprocessing is the process of preparing raw data to be used on a machine learning model. It is the first and most important step in developing a machine learning model.
- When developing a machine learning project, we do not always come across clean and formatted data. And, before performing any operation on data, it must be cleaned and formatted. As a result, we use the data preprocessing task for this.
- Real-world data typically contains noise, missing values, and may be in an unusable format that cannot be used directly for machine learning models.
- Data preprocessing is a necessary task for cleaning the data and preparing it for a machine learning model, which improves the accuracy and efficiency of the machine learning model.
- It involves below steps :

(1) Get the Dataset
(2) Importing Libraries
(3) Importing the Datasets
(4) Handling Missing Data
(5) Encoding the Categorical Data
(6) Splitting the Dataset into the Training set and Test set
(7) Feature Scaling

(1) Get the Dataset

- The first thing we need to create a machine learning model is a dataset, because a machine learning model is entirely dependent on data. The dataset is the collection of data for a specific problem in the proper format.
- Datasets can be of various formats for various purposes. For example, if we want to create a machine learning model for business purposes, the dataset will be different from the dataset required for a liver patient. As a result, each dataset is distinct from the others. We usually save the dataset as a CSV file before using it in our code. However, there may be times when we need to use an HTML or xlsx file.

(2) Importing Libraries

- To perform data preprocessing with Python, we must first import some predefined Python libraries.
- These libraries are used to carry out specific tasks. For data preprocessing, we will use three specific libraries, which are: Numpy, Matplotlib and Pandas.

(3) Importing the Datasets

- We must now import the datasets that we have gathered for our machine learning project. However, before importing a dataset, we must make the current directory a working directory.
- To import the dataset, we will use the pandas library's read_csv() function, which reads a csv file and performs various operations on it. We can use this function to read a csv file both locally and via a URL.
- It is essential in machine learning to distinguish the feature matrix (independent variables) from the dataset.

(4) Handling Missing Data

- The next step in data preprocessing is to deal with missing data in the datasets. If our dataset contains some missing data, it may pose a significant challenge to our machine learning model.
- As a result, handling missing values in the dataset is required.
- There are primarily two approaches to dealing with missing data :
  - (i) **By removing the specific row** : The first method is commonly used to deal with null values. In this manner, we simply delete the specific row or column that contains null values. However, this method is inefficient, and removing data may result in information loss, resulting in an inaccurate output.
  - (ii) **By calculating the mean :** In this way, we will calculate the mean of that column or row which contains any missing value and will put it on the place of missing value. This strategy is useful for the features which have numeric data such as age, salary, year, etc. Here, we will use this approach.

(5) Encoding the Categorical Data

- Categorical data is data which has some categories such as Country. Because machine learning models are entirely based on mathematics and numbers, having a categorical variable in our dataset may cause problems when building the model.
- As a result, these categorical variables must be encoded into numbers. We can use OneHotEncoding or Label Encoding technique.

(6) Splitting the Dataset into the Training set and Test set

- In machine learning data preprocessing, we divide our dataset into a training set and a test set.
- This is an important step in data preprocessing because it allows us to improve the performance of our machine learning model.

- Assume we trained our machine learning model on one dataset and then tested it on another. It will then be difficult for our model to understand the correlations between the models.
- **Training Set :** A subset of dataset to train the machine learning model, and we already know the output.
- **Test set :** A subset of dataset to test the machine learning model, and by using the test set, model predicts the output.

(7) Feature Scaling

- Feature scaling is the final step in machine learning data preprocessing. It is a method for standardizing the independent variables of a dataset within a given range.
- In feature scaling, we place our variables in the same range and scale so that no one variable dominates the other.

2.3 NORMALIZATION AND SCALING

GQ. Explain the concept of scaling and normalization with its types.

- Scaling and normalization are so similar that they're often applied interchangeably, but they have different effects on the data.
- In both scaling and normalization, we are transforming the values of numeric variables so that the transformed data points have specific helpful properties. These properties can be exploited to create better features and models.
- In Scaling, we're changing the **range** of the distribution of the data, while in normalization, we're changing the **shape** of the distribution of the data.
- In scaling, we're transforming the data so that it fits within a specific scale, like 0-100 or 0-1.
- Usually, 0-1. You want to scale data especially when you're using methods based on measures of how far apart data points are.
- Normalization is a more radical transformation. The point of normalization is to change your observations so that they can be described as a normal distribution.

2.3.1 Types of Scaling

(1) Simple Feature Scaling : This method simply divides each value by the maximum value for that feature. The resultant values are in the range between zero(0) and one(1). Simple-feature scaling is the defacto scaling method used on image-data, when we scale images by dividing each image by 255 (maximum image pixel intensity). The formula for simple feature scaling is given as :

$$X_{new} = \frac{X_{old}}{X_{max}}$$

(2) **Min-max Scaling :** This scaler takes each value and subtracts the minimum and then divides by the range(max-min).The resultant values range between zero(0) and one(1). The formula for min-max scaling is given as :

$$X_{new} = \frac{X_{old} - X_{min}}{X_{max} - X_{min}}$$

2.3.2 Types of Normalization

(1) Z-Score or Standard Score

In this technique, values are normalized based on mean and standard deviation of the data A. The formula used is :

$$X_{new} = \frac{X_{old} - \mu_A}{\sigma_A}$$

σ_A, μ_A is the standard deviation and mean of A respectively.

Example : If mean salary is \$54,000 and standard deviation is \$16,000, then the z-score value of salary \$73,600 will be $\frac{73600 - 54000}{16000} = 1.225$

(2) Box-cox Transformation

- A Box-Cox transformation is a transformation of a non-normal dependent variable into a normal shape. The Box-Cox transformation is named after statisticians George Box and Sir David Roxbee Cox.
- At the heart of the box-cox normalization is an exponent *lambda* (λ), which varies from – 5 to 5. All values of λ are considered and the optimal value for your data is selected; The "*optimal value*" is the one which results in the best approximation of a normal distribution curve.

$$w_t = \begin{cases} \log(y_t); & \text{if } \lambda = 0 \\ \frac{(y_t^{\lambda} - 1)}{\lambda}; & \text{otherwise} \end{cases}$$

2.4 STANDARDIZATION

- Standardization is necessary when features of the input data set have wide ranges or are simply measured in different measurement units (such as pounds, metres, miles, etc.).
- For many machine learning models, these variations in the initial feature ranges are problematic. For models that compute distance, for instance, if one of the features has a wide range of values, the distance will be determined by this specific feature.
- **Example :** Let's say we have a 2-dimensional data set with the variables Height in Meters and Weight in Pounds, both of which have ranges of [1 to 2] Meters and [10 to 200] Pounds, respectively. No matter what distance-based model you run on this set of data, the Weight feature will take precedence over the Height feature and contribute more to the distance calculation simply because it contains larger values.

So, standardization is the way to avoid this issue by transforming features to comparable scales.

- Z-score is one of the most popular methods to standardize data, and can be done by subtracting the mean and dividing by the standard deviation for each value of each feature.

$$z = \frac{\text{value} - \text{mean}}{\text{standard deviation}}$$

- Once the standardization is done, all the features will have a mean of zero, a standard deviation of one, and thus, the same scale.

2.5 MANAGING MISSING VALUES

GQ. Explain how the missing values are handled in data preprocessing.

Imagine that you are asked to analyze a dataset. You find that there are many tuples having no recorded value for several attributes such as customer income. So, the question arising here is how to fill in the missing values for this attribute. There are several methods as discussed here.

(1) **Ignore the tuple :** When the class label is missing, this technique is used. However, unless the tuple contains numerous attributes with missing values, this approach is not particularly useful.

(2) **Fill in the missing value manually :** This approach is effective on small data set with some missing values.

(3) **Use a global constant to fill in the missing value :** You can replace all missing attribute values with global constant, such as a label like "Unknown" or $-\infty$.

(4) **Use a measure of central tendency for attribute (e.g. the mean or median) to fill in the missing value :** For example, suppose customer average income is $25000, then you can use this value to replace missing value for income.

(5) **Use the attribute mean or median for all samples belonging to the same class as the given tuple :** For example, if you are classifying customers according to their credit_score, then you can replace the missing value with the mean income value for customers in the same credit_score category as that of the given tuple. If the data distribution for a given class is skewed, then use the median value.

(6) **Use the most probable value to fill in the missing value :** This can be determined using regression, Bayesian classification or decision-tree induction.

2.6 INTRODUCTION TO DIMENSIONALITY REDUCTION

- The number of input variables or features for a dataset is referred to as its dimensionality.
- Dimensionality reduction refers to techniques that reduce the number of input variables in a dataset.
- More input features often make a predictive modeling task more challenging to model, more generally referred to as the curse of dimensionality.

- High-dimensionality statistics and dimensionality reduction techniques are often used for data visualization.
- Nevertheless, these techniques can be used in applied machine learning to simplify a classification or regression dataset in order to better fit a predictive model.
- There are a number of advantages that makes dimensionality reduction important:
  (1) The model accuracy is improved when there is less data.
  (2) When dealing with fewer dimensions, it requires a lot less computing power and also since the dater is lesser, the algorithm can train faster.
  (3) Lesser data requires lesser storage space.
  (4) Lesser dimensions can work with algorithms that cannot be used with larger dimensions.
  (5) Lesser features come with the benefit of noise and redundant variables.
- Dimensionality reduction has two main components :
  (1) **Feature selection :** This is the process where the universal set of features or variables is used to extract a subset that can be used to model the problem. Feature selection is done as Filter or Wrapper or Embedded.
  (2) **Feature extraction :** This is used to reduce data that is in a higher-dimensional space to a lower-dimensional space. For example as to how features in 3 dimensions can be reduced to two dimensions for simplicity.
- Some of the dimension reduction techniques include :
  (1) **Principal Component Analysis (PCA) :** This method is commonly used with continuous data. It works under the condition that the variance in mapped data in the lower dimensional space needs to be at the peak when the data is mapped from a higher-dimensional space. I other words it projects data where variance increases and the features with the most variance become the principal components.
  (2) **Linear Discriminant Analysis (LDA) :** This project's data in such a way that the separability of the class is maximized. Points from the same class are projected closely together while those from different classes are spaced far apart.
  (3) **Generalized Discriminant Analysis (GDA) :** The GDA is quite an effective approach when it comes to extracting non-linear features.

2.7 PRINCIPAL COMPONENT ANALYSIS (PCA)

GQ. Explain how PCA helps in dimensionality reduction.

- Principal Component Analysis (PCA) is an unsupervised learning algorithm that is used for the dimensionality reduction in machine learning.
- It is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation. These new transformed features are called the **Principal Components**.

- It is one of the popular tools that is used for exploratory data analysis and predictive modeling.
- It is a technique to draw strong patterns from the given dataset by reducing the variances.
- Some real-world applications of PCA are image processing, movie recommendation system, optimizing the power allocation in various communication channels.
- It is a feature extraction technique, so it contains the important variables and drops the least important variable.

Steps in PCA

(1) **Standardize the dataset :** First, we need to standardize the dataset and for that, we need to calculate the mean and standard deviation for each feature. We use Z-score method to standardize the dataset.

(2) **Calculate the covariance matrix for the whole dataset :** The covariance matrix for the given dataset will be calculated as below

For population

$$\text{Cov}(x, y) = \frac{\sum (x_i - \bar{x}) * (y_i - \bar{y})}{N}$$

For sample

$$\text{Cov}(x, y) = \frac{\sum (x_i - \bar{x}) * (y_i - \bar{y})}{(N - 1)}$$

Since we standardize the dataset, so the mean for each feature is 0 and the standard deviation is 1.

For example, for a 3-dimensional data set with 3 variables x, y, and z, the covariance matrix is a 3×3 matrix of this from :

$$\begin{bmatrix} \text{Cov}(x, x) & \text{Cov}(x, y) & \text{Cov}(x, z) \\ \text{Cov}(y, x) & \text{Cov}(y, y) & \text{Cov}(y, z) \\ \text{Cov}(z, x) & \text{Cov}(z, y) & \text{Cov}(z, z) \end{bmatrix}$$

(3) **Calculating Eigen values and Eigen vectors**

An eigenvector is a nonzero vector that changes at most by a scalar factor when that linear transformation is applied to it. The corresponding eigenvalue is the factor by which the eigenvector is scaled.

Let A be a square matrix (in our case the covariance matrix), ν a vector and λ a scalar that satisfies $A\nu = \lambda\nu$, then λ is called eigenvalue associated with eigenvector ν of A.

Rearranging the above equation, $A\nu - \lambda\nu = 0$; $(A - \lambda I)\,\nu = 0$

Eigen vectors can be obtained by solving the $(A - \lambda I)\,\nu = 0$ equation for ν vector with different λ values.

(4) **Sort the eigenvectors from the highest eigenvalue to the lowest.** The eigenvector with the highest eigenvalue is the first principal component. Higher eigenvalues correspond to greater amounts of shared variance.

(5) **Select the number of principal components.** Select the top N eigenvectors (based on their eigenvalues) to become the N principal components. The optimal number of principal components is both subjective and problem-dependent. Usually, we look at the cumulative amount of shared variance explained by the combination of principal components and pick that number of components, which still significantly explains the shared variance.

(6) **Transform the original matrix :** Feature matrix * top k eigenvectors = Transformed Data

☞ **Advantages of PCA**

(1) **Easy to compute :** PCA is based on linear algebra, which is computationally easy to solve by computers.

(2) **Speeds up other machine learning algorithms :** Machine learning algorithms converge faster when trained on principal components instead of the original dataset.

(3) **Counteracts the issues of high-dimensional data :** High-dimensional data causes regression-based algorithms to overfit easily. By using PCA beforehand to lower the dimensions of the training dataset, we prevent the predictive algorithms from overfitting.

☞ **Disadvantages of PCA**

(1) **Low interpretability of principal components :** Principal components are linear combinations of the features from the original data, but they are not as easy to interpret. For example, it is difficult to tell which are the most important features in the dataset after computing principal components.

(2) **The trade-off between information loss and dimensionality reduction :** Although dimensionality reduction is useful, it comes at a cost. Information loss is a necessary part of PCA. Balancing the trade-off between dimensionality reduction and information loss is unfortunately a necessary compromise that we have to make when using PCA.

2.8 FEATURE EXTRACTION

GQ. Explain how kernel PCA and Local Binary Pattern helps in dimensionality reduction

- Feature extraction is a process of dimensionality reduction by which an initial set of raw data is reduced to more manageable groups for processing.
- A characteristic of these large data sets is a large number of variables that require a lot of computing resources to process.
- Feature extraction is the name for methods that select and /or combine variables into features, effectively reducing the amount of data that must be processed, while still accurately and completely describing the original data set.
- The feature extraction technique gives us new features which are a linear combination of the existing features. The new set of features will have different values as compared to the original feature values.
- **The main aim is that fewer features will be required to capture the same information.**

- We might think that choosing fewer features might lead to underfitting but in the case of the Feature Extraction technique, the extra data is generally noise.

2.8.1 Kernel PCA

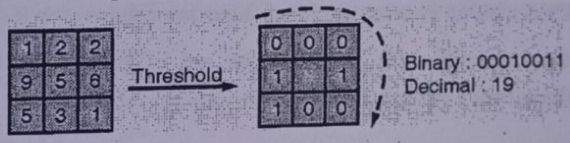
- Kernel PCA was developed in an effort to help with the classification of data whose decision boundaries are described by non-linear function.
- The idea is to go to a higher dimension space in which the decision boundary becomes linear.
- Here's an easy argument to understand the process. Suppose the decision boundary is described by a third order polynomial = $a + bx + cx^2 + dx^3$. Now, plotting this function in the usual $x - y$ plane will produce a wavy line, something similar to the made-up decision boundary in the right-hand-side picture above.
- Suppose instead we go to a higher dimensionality space in which the axes are x, x^2, x^3 and y. In this 4D space the third order polynomial becomes a linear function and the decision boundary becomes a hyperplane.
- So, the trick is to find a suitable transformation (up-scaling) of the dimensions to try and recover the linearity of the boundary. In this way the usual PCA decomposition is again suitable.
- This is all good but, as always, there's a catch. A generic non-linear combination of the original variables will have a huge number of new variables which rapidly blows up the computational complexity of the problem.
- However, we won't know the exact combination of non-linear terms we need, hence the large number of combinations that are in principle is required.
- Let's try and explain this issue with another simple example. Suppose we have only two wavelengths, call them λ_1 and λ_2. Now suppose we want to take a generic combination up to the second order of these two variables. The new variable set will then contain the following: $[\lambda_1,\lambda_2,\lambda_1\lambda_2,\lambda_1\lambda_1,\lambda_2\lambda_2]$. So, we went from 2 variables to 5, just by seeking a quadratic combination!
- Since one in general has tens or hundreds of wavelengths, and would like to consider higher order polynomials, you can get the idea of the large number of variables that would be required.
- Now fortunately there is a solution to this problem, which is commonly referred to as the **kernel trick.**
- Ok, let's call x the original set of n variables, let's call ϕ (x) the non-linear combination (mapping) of these variables into a m > nm > n dataset.
- Now we can compute the kernel function κ (x) = ϕ (x) ϕ^T(x). Note that the kernel function in practice is an array even though we are using a function (continuous) notation.
- Now, it turns out that the kernel function plays the same role as the covariance matrix did in linear PCA.
- This means that we can calculate the eigenvalues and eigenvectors of the kernel matrix and these are the new principal components of the m-dimensional space where we mapped our original variables into.
- The kernel trick is called this way because the kernel function (matrix) enables us to get to the

eigenvalues and eigenvector without actually calculating ϕ (x) explicitly. This is the step that would blow up the number of variables and we can circumvent it using the kernel trick.

- There are of course different choices for the kernel matrix. Common ones are the Gaussian kernel or the polynomial kernel.
- A polynomial kernel would be the right choice for decision boundaries that are polynomial in shape.
- A Gaussian kernel is a good choice whenever one wants to distinguish data points based on the distance from a common centre.
- Once we have the kernel, we follow the same procedure as for conventional PCA. Remember the kernel plays the same role as the covariance matrix in linear PCA, therefore we can calculate its eigenvalues and eigenvectors and stack them up to the selected number of components we want to keep.

2.8.2 Local Binary Pattern (LBP)

- *Local Binary Pattern (LBP)* is a very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number.
- The LBP feature vector, in its simplest form, is created in the following manner :



(1B1)

(i) Divide the examined window into cells (e.g. 16x16 pixels for each cell).

(ii) For each pixel in a cell, compare the pixel to each of its 8 neighbors (on its left-top, left-middle, left-bottom, right-top, etc.). Follow the pixels along a circle, i.e. clockwise or counterclockwise.

(iii) In the above step, the neighbours considered can be changed by varying the radius of the circle around the pixel, R and the quantization of the angular space P.

(iv) Where the center pixel's value is greater than the neighbor's value, write "0". Otherwise, write "1". This gives an 8-digit binary number (which is usually converted to decimal for convenience).

(v) Compute the histogram, over the cell, of the frequency of each "number" occurring (i.e., each combination of which pixels are smaller and which are greater than the centre). This histogram can be seen as a 256-dimensional feature vector.

(vi) Optionally normalize the histogram.

(vii) Concatenate (normalized) histograms of all cells. This gives a feature vector for the entire window.

- The feature vector can now then be processed using some machine-learning algorithm to classify images. Such classifiers are often used for face recognition or texture analysis.

2.9 INTRODUCTION TO VARIOUS FEATURE SELECTION TECHNIQUES

GQ. What is feature selection? Explain different feature selection algorithms.
GQ. Explain forward and backward feature selection process.

- Feature Selection is the method of reducing the input variable to your model by using only relevant data and getting rid of noise in data.
- It is the process of automatically choosing relevant features for your machine learning model based on the type of problem you are trying to solve.
- We do this by including or excluding important features without changing them.
- It helps in cutting down the noise in our data and reducing the size of our input data.

Benefits of Feature Selection

(1) Using unnecessary feature variables for the prediction can deteriorate the performance of a predictive model. Thus, feature selection helps in improving the model performance.

(2) Algorithms like linear regression and logistic regression must avoid using correlated features. Using feature selection methods thus leads to a better fit of these models.

(3) It is an excellent practice to work with a minimum set of predictive modeling features as they significantly reduce the algorithm's complexity and computational costs.

Feature Selection Algorithms

Feature selection algorithms can be classified into three major categories :

(1) Filter methods (2) Wrapper Methods (3) Intrinsic Methods

(1) Filter methods

- Filter methods for feature selection are usually pre-processing techniques that independently consider each feature in the dataset.
- It implements its model on each feature and then evaluates which can then be used to analyze its impact on a predictive model.
- Such methods include information gain, entropy, consistency-based feature selection, correlation matrix, etc.
- For basic guidance, we can refer to the following table for defining correlation coefficients.

Feature/Response	Continuous	Categorical
Continuous	Pearson's correlation	LDA
Categorical	Anova	Chi-Square

- **Pearson's Correlation :** It is used as a measure for quantifying linear dependence between two continuous variables X and Y. Its value varies from – 1 to + 1. Pearson's correlation is given as :

$$\rho_{X,Y} = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y}$$

- **LDA :** Linear discriminant analysis is used to find a linear combination of features that characterizes or separates two or more classes (or levels) of a categorical variable.
- **ANOVA :** ANOVA stands for Analysis of variance. It is similar to LDA except for the fact that it is operated using one or more categorical independent features and one continuous dependent feature. It provides a statistical test of whether the means of several groups are equal or not.
- **Chi-Square :** It is a is a statistical test applied to the groups of categorical features to evaluate the likelihood of correlation or association between them using their frequency distribution.
- One thing that should be kept in mind is that filter methods do not remove multicollinearity. So, we must deal with multicollinearity of features as well before training models for your data.
- The filter methods are popular for feature selection methods because of their generic behavior.
- However, they are proven disadvantageous as they do not consider the nature of a predictive model and typically reduce its accuracy.

(2) Wrapper Methods

- The wrapper methods aim to create a subset of features from the given dataset that results in the best performance of a predictive model.
- In other words, it tests every subset of the available variables for the model's accuracy.
- There are two kinds of wrapper methods for feature selection, greedy and non-greedy.
- The greedy search approach involves following a path that heads towards achieving the best results at the given time. This approach results in locally best results. An example of a greedy search method is the Recursive Feature Elimination (RFE) method.
- On the other hand, the non-greedy approach involves assessing all the previous feature subsets and can lead to a path that results in the overall best performance. Genetic Algorithms (GA) and Simulated Annealing (SA) are examples of non-greedy wrapper methods.

(3) Intrinsic Methods

- This method combines the qualities of both the Filter and Wrapper method to create the best subset.
- In these methods, the feature selection algorithm is blended as part of the learning algorithm, thus having its own built-in feature selection methods.
- It means if you are using these algorithms, you don't need to worry about using a feature selection method explicitly.
- These methods are fast and easy to implement as no external algorithm is required to filter features.

- Examples of intrinsic methods for feature selection are :

  (i) **Rule-and-Tree-based algorithms :** The basic idea behind the mathematical structure of these algorithms is to split the dataset into different sets based on a feature variable in a manner that results in a homogenous spread in the resulting subsets. Thus, a feature variable that didn't lead to a split is automatically considered redundant by the model.

  (ii) **Multivariate adaptive regression spline (MARS) models :** The MARS algorithms create new feature variables from the existing ones in the dataset. These features are then added to a linear model in sequence. If the algorithm does not use a few features to create the MARS features, they are considered irrelevant and automatically ignored.

  (iii) **Regularization models :** These models assign weights to features in a model to improve the fit quality. The lasso regularization method implements consequences that can be narrowed down to absolute zero, indicating you should remove the feature from the predictive model's equation.

2.9.1 Forward Feature Selection

- Forward feature selection is an iterative method in which we start with having no feature in the model.
- In each iteration, we keep adding the feature which best improves our model till an addition of a new variable does not improve the performance of the model.

2.9.2 Backward Feature Selection

- In backward feature selection, also called backward elimination, we start with all the features and removes the least significant feature at each iteration which improves the performance of the model.
- We repeat this until no improvement is observed on removal of features.
- Below are some main steps which are used to apply backward elimination process:

  Step 1 : Firstly, we need to select a significance level to stay in the model. (SL=0.05)

  Step 2 : Fit the complete model with all possible predictors/independent variables.

  Step 3 : Choose the predictor which has the highest P-value, such that.

  If P-value > SL, go to step 4.

  Else Finish, and Our model is ready.

  Step 4 : Remove that predictor.

  Step 5 : Rebuild and fit the model with the remaining variables.

2.10 STATISTICAL FEATURE ENGINEERING

GQ Explain different statistical measures in feature engineering with suitable examples.

- It is essential to have an overall picture of the data, if data preprocessing is to be made successful.
- Statistical description of data is useful in identifying the properties of the data and highlight which data value should be treated at noise or outliers.

- Following are the basic statistical description of data :

2.10.1 Measures of Central Tendency

- A measure of central tendency is a number used to represent the center or middle of a set of data values.
- The mean, median, mode and midrange are commonly used measures of central tendency.

(i) Mean

- The mean, or average, of n numbers is the sum of the numbers divided by n.
- The mean is denoted by $\bar{x}$ and is read as "x-bar".
- For the data set $x_1, x_2, \ldots, x_n$, the mean is

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{x_1 + x_2 + \ldots + x_n}{n}$$

- Sometimes, weights are associated with the value x_i. The weights reflect the importance, significance or frequency of occurrence to their respective values. The weighted arithmetic mean or the weighted average is computed as

$$\bar{x} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i} = \frac{w_1 x_1 + w_2 x_2 + \ldots + w_n x_n}{w_1 + w_2 + \ldots + w_n}$$

- Mean has one limitation; it is highly sensitive to outliers. Under such condition, median would be a better measure of central tendency.

(ii) Median

- The median of n numbers is the middle number when numbers are written in order.
- If n is even, the median is the mean of the two middle numbers.
- When we have large number of observations, the median is expensive to compute.
- In such case, we can approximate the median of the entire data set by interpolation using the formula :

$$\text{median} = L_1 + \left(\frac{\frac{n}{2} - (\sum \text{freq})_1}{\text{freq}_{\text{median}}} \right) \text{width}$$

where,

L_1 is the lower boundary of the median interval,

n is the number of values in the entire data set,

$(\sum \text{freq})_1$ is the sum of frequencies of all of the intervals that are lower than the median interval

$Freq_{median}$ is the frequency of the median interval and

width is the width of the median interval.

(iii) Mode

The mode of n numbers is the number or numbers that occur most frequently.

There may be one mode, no mode or more than one mode.

For unimodal numeric data that are asymmetrical, we have the following empirical relation :

mean – median ≈ 3 × (mean – median)

(iv) Midrange

It is the average of the largest and smallest values in the set.

(v) Size, Count

Size or count is the number of data points in a data set.

Size = n = count(x_i) where i = 1...n.

(vi) Length

Length defines the number of rows present in the dataset. In Python, you can **use the built-in len() method.**

Ex. 2.10.1 : The data set below gives the waiting time (in minutes) of several people having the oil changed in their car at an auto mechanics shop. 22, 18, 25, 21, 28, 26, 20, 28, 20. Find the mean, median, mode and the midrange of the data set.

☑ **Soln. : Data set :** 22, 18, 25, 21, 28, 26, 20, 28, 20

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{x_1 + x_2 + \ldots + x_n}{n}$$

$$= \frac{22 + 18 + 25 + 21 + 28 + 26 + 20 + 28 + 20}{9} = 23.11$$

To find median, arrange the values in order.

18, 20, 20, 21, 22, 25, 26, 28, 28

There are total 9 values i.e. n is odd. Thus, the median is the middle value.

Median = 22

Mode is the number or numbers that occur most frequently. Here, 20 and 28 are repeated twice. Thus data set is bimodal with values 20 and 28.

$$\text{Midrange} = \frac{\text{largest value} + \text{smallest value}}{2} = \frac{28 + 18}{2} = 23$$

2.10.2 Dispersion of Data

- A measure of dispersion is a statistic that tells you how dispersed, or spread out, data values are.
- The measures include range, quantiles, quartiles, percentiles, the interquartile range, and the five-number summary displayed as a boxplot, variance, and standard deviation.

(i) Quartiles

- Quartiles are values that divide your data into quarters.
- However, quartiles are not shaped like pizza slices; instead they divide your data into four segments according to which the numbers fall on the number line.
- The four quarters that divide a data set into quartiles are:
  (a) The lowest 25% of numbers. Also called the 1st quartile (Q_1) or 25th percentile.
  (b) The next lowest 25% of numbers (up to the median). Also called the 2nd quartile (Q_2) or 50th percentile.
  (c) The second highest 25% of numbers (above the median). Also called the 3rd quartile (Q_3) or 75th percentile.
  (d) The highest 25% of numbers. Also called the 4th quartile (Q_4) or 100th percentile.
- As quartiles divide numbers up aecording to where their position is on the number line, you have to put the numbers in order before you can figure out where the quartiles are.

(ii) Interquartile Range (IQR)

- Interquartile range is defined asthe difference between the upper and lower quartile values in a set of data.
- It is commonly referred to as IQR and is used as a measure of spread and variability in a data set.

$$IQR = Q_3 - Q_1$$

(iii) Five Number Summary

- The five number summary gives you a rough idea about what your data set looks like.
- It includes five items: the minimum value, the first quartile (Q_1), the median, the third quartile (Q_3), the maximum value.
- In order for the five numbers to exist, your data set must meet these two requirements :
  (a) Your data must be **univariate**. In other words, the data must be a single variable. For example, this list of weights is one variable: 120, 100, 130, 145. If you have a list of ages and you want to compare the ages to weights, it becomes bivariate data (two variables). For example: age 1 (25 pounds), 5 (60 pounds), 15 (129 pounds). The matching pairs makes it impossible to find a five number summary.
  (b) Your data must be **ordinal**, **interval**, or **ratio**.

Steps to Find a Five-Number Summary

Step 1 : Put your numbers in ascending order (from smallest to largest).

For example, consider the data set in order as: 1, 2, 5, 6, 7, 9, 12, 15, 18, 19, 27.

Step 2 : Find the minimum and maximum for your data set.

In the example in step 1, the minimum (the smallest number) is 1 and the maximum (the larges number) is 27.

Step 3 : Find the median. The median is the middle number.

Step 4 : Place parentheses around the numbers above and below the median. (This is not technically necessary, but it makes Q_1 and Q_3 easier to find).

(1, 2, 5, 6, 7), 9, (12, 15, 18, 19, 27).

Step 5 : Find Q_1 and Q_3. Q_1 can be thought of as a median in the lower half of the data, and Q_3 can be thought of as a median for the upper half of data.

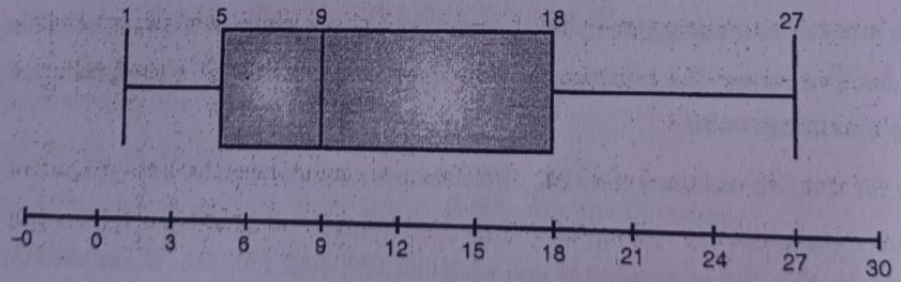
(1, 2, 5, 6, 7), 9, (12,15,18,19,27).

Step 6 : Write down your summary found in the above steps.

minimum = 1, Q_1= 5, median = 9, Q_3 = 18 and maximum = 27.

(iv) Boxplot

- A boxplot (or whisker plot) is defined as a graphical method of displaying variation in a set of data.
- A boxplot incorporates the five-summary as follows:
  (a) The ends of the box are at the quartiles and the box length is the interquartile range.
  (b) The median is marked by a line within a box.
  (c) Two lines (called whiskers) outside the box extend to the minimum and maximum values in the data set.
- The boxplot for five-number summary example above is as given below.



(184)**Fig. 2.10.1 : Boxplot Example**

Boxplots can be computed in *O(n logn)* time.

(v) Outlier

It is a value higher or lower than 1.5 × IQR **(Inter-Quartile Range)**

(vi) Variance and Standard Deviation

- Variance and standard deviation are measures of data dispersion. They indicate how spread out a data distribution is.
- For the data set $x_1, x_2, \ldots, x_n$, the variance is calculated as

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 = \left(\frac{1}{n}\sum_{i=1}^{n} x_i^2\right) - (\bar{x})^2$$

  where $\bar{x}$ is the mean value of the observation
- The standard deviation, σ, of the observations is the square root of the variance σ^2.
- A low standard deviation indicates that the data observations tend to be very close to the mean, while a high standard deviation indicates that the data observations are spread out over a large range of values.
- When all observations have the same value, $\sigma = 0$. Otherwise, $\sigma > 0$.

2.11 MULTIDIMENSIONAL SCALING

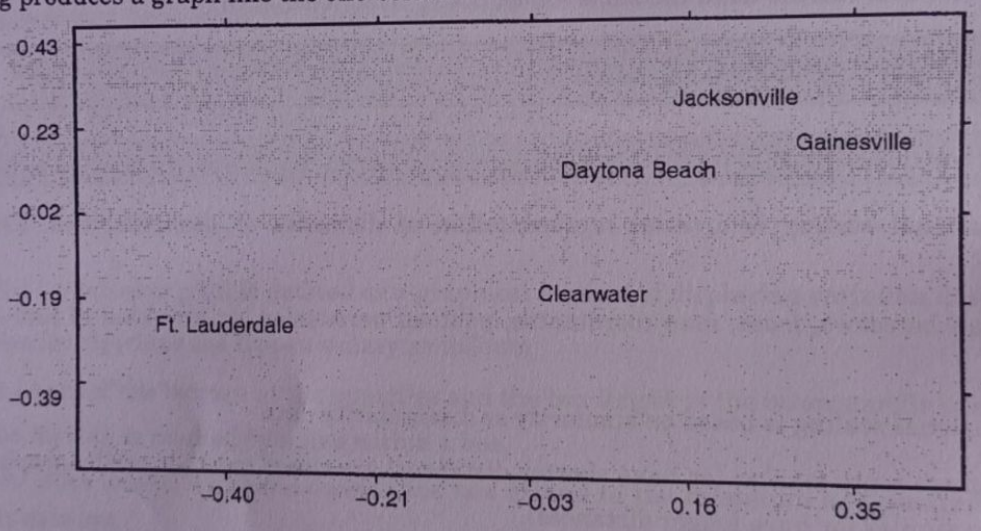
GQ. Explain the concept of multidimensional scaling.

- Multidimensional scaling is a visual representation of distances or dissimilarities between sets of objects.
- "Objects" can be colors, faces, map coordinates, political persuasion, or any kind of real or conceptual stimuli.
- Multidimensional scaling is based on similarity or dissimilarity data.
- Objects that are more similar (or have shorter distances) are closer together on the graph than objects that are less similar (or have longer distances).
- As well as interpreting dissimilarities as distances on a graph, MDS can also serve as a dimension reduction technique for high-dimensional data.
- The term scaling comes from psychometrics, where abstract concepts ("objects") are assigned numbers according to a rule. For example, you may want to quantify a person's attitude to global warming. You could assign a "1" to "doesn't believe in global warming", a 10 to "firmly believes in global warming" and a scale of 2 to 9 for attitudes in between. You can also think of "scaling" as the fact that you're essentially scaling down the data (i.e., making it simpler by creating lower-dimensional data).
- Data that is scaled down in dimension keeps similar properties. For example, two data points that are close together in high-dimensional space will also be close together in low-dimensional space.
- The "multidimensional" part is due to the fact that you aren't limited to two dimensional graphs or data.
- Three-dimensional, four-dimensional and higher plots are possible.

- Multidimensional scaling uses a square, symmetric matrix for input. The matrix shows relationship between items.
- For a simple example, let's say you had a set of cities in Florida and their distances :

CITY	Clearwater	Daytona Beach	Ft. Lauderdale	Gainesville	Jacksonville
Clearwater	0	159	247	131	197
Daytona Beach	159	0	230	97	89
Ft. Lauderdale	247	230	0	309	317
Gainesville	131	97	309	0	68
Jacksonville	197	89	317	68	0

- The scaling produces a graph like the one below.



- The very simple example above shows cities and distances, which are easy to visualize as a map. However, multidimensional scaling can work on "theoretically" mapped data as well.

Basic steps

(1) **Assign a number of points to coordinates in N-dimensional space :** N-dimensional space could be 2-dimensional, 3-dimensional, or higher spaces (at least, theoretically, because 4-dimensional spaces and above are difficult to model). The orientation of the coordinate axes is arbitrary and is mostly the researcher's choice. For maps like the one in the simple example above, axes that represent north/south and east/west make the most sense.

(2) **Calculate Euclidean distances for all pairs of points :** The Euclidean distance is the straight-line distance between two points x and y in Euclidean space. It's calculated using the Pythagorean theorem ($c^2 = a^2 + b^2$), although it becomes somewhat more complicated for N-dimensional space. This results in the similarity matrix.

(3) **Compare the similarity matrix with the original input matrix by evaluating the stress function :** Stress is a goodness-of-fit measure, based on differences between predicted and actual distances. Fits close to zero are excellent, while anything over 0.2 should be considered "poor".

(4) **Adjust coordinates, if necessary, to minimize stress.**

Types of MDS

(1) **Metric MDS :** Metric MDS already has the input matrix in the form of distances (i.e. actual distances between cities) and therefore the distances have meaning in the input matrix and create a map of actual physical locations from those distances.

(2) **Non-metric MDS :** In non-metric MDS, the distances are just a representation of the rankings (i.e., high as in 7 or low as in 1) and they do not have any meaning on their own but they are needed to create the map using Euclidean geometry and the map then just shows the similarity in rankings represented by distances between coordinates on the map.

2.12 MATRIX FACTORIZATION TECHNIQUE

GQ. Explain the concept of matrix factorization in recommender system.

- Matrix factorization is a class of collaborative filtering algorithms used in recommender systems.
- Matrix factorization algorithms work by decomposing the user-item interaction matrix into the product of two lower dimensionality rectangular matrices.
- The idea behind matrix factorization is to represent users and items in a lower dimensional latent space.
- Let $R \in \mathbb{R}^{m \times n}$ denote the interaction matrix with m users and n items, and the values of R represent explicit ratings.
- The user-item interaction will be factorized into a user latent matrix $P \in \mathbb{R}^{m \times k}$ and an item latent matrix $Q \in \mathbb{R}^{n \times k}$, where $k << m, n$ is the latent factor size.
- Let p_u denote the u^{th} row of P and q_i denote the i^{th} row of Q. For a given item i, the elements of q_i measure the extent to which the item possesses those characteristics such as the genres and languages of a movie. For a given user u, the elements of p_u measure the extent of interest the user has in items' corresponding characteristics.
- These latent factors might measure obvious dimensions as mentioned in those examples or are completely uninterpretable. The predicted ratings can be estimated by

$$\hat{R} = PQ^T$$

Where $\hat{R} \in \mathbb{R}^{m \times n}$ is the predicted rating matrix which has the same shape as R.

- One major problem of this prediction rule is that users/items biases cannot be modelled. For example, some users tend to give higher ratings or some items always get lower ratings due to poorer quality.

These biases are commonplace in real-world applications. To capture these biases, user specific and item specific bias terms are introduced. Specifically, the predicted rating user gives to item is calculated by

$$\hat{R}_{ui} = p_u q_i + b_u + b_i$$

- Then, we train the matrix factorization model by minimizing the mean squared error between predicted rating scores and real rating scores. The objective function is defined as follows :

$$\underset{p, Q, b}{\text{argmin}} \sum_{(u, i,) \in k} ||R_{ui} - \hat{R}_{ui}||^2 + \lambda (||P||_F^2 + ||Q||_F^2 + b_u^2 + b_i^2)$$

where λ denotes the regularization rate. The regularizing term

$$\lambda (||P||_F^2 + ||Q||_F^2 + b_u^2 + b_i^2)$$

is used to avoid over-fitting by penalizing the magnitude of the parameters. The (u, i) pairs for which R_{ui} is known are stored in the set $k = \{(u, i) | R_{ui} \text{ is known}\}$. The model parameters can be learned with an optimization algorithm, such as Stochastic Gradient Descent and Adam.

- An intuitive illustration of the matrix factorization model is shown below:

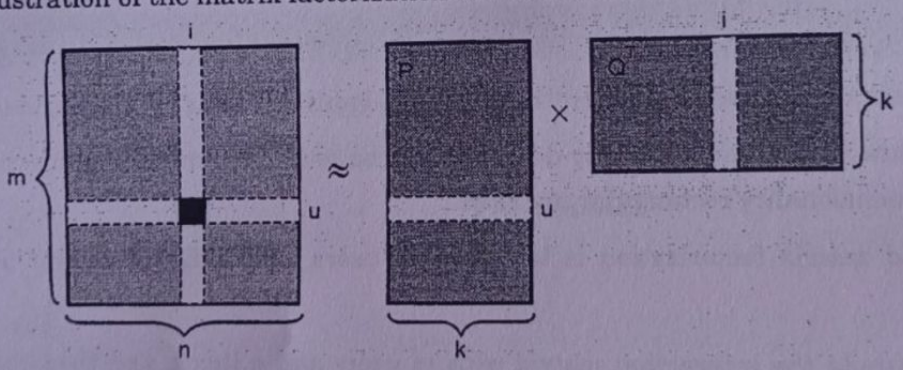


Fig. 2.12.1 : Illustration of matrix factorization model

Descriptive Questions

Q. 1 What is feature engineering ? Explain the four different processes in feature engineering.

Q. 2 What is data preprocessing ? Explain the steps involved in data preprocessing.

Q. 3 How do you handle missing data in dataset ?

Q. 4 Explain different types of feature selection methods.

Q. 5 Explain different statistical measures in feature engineering with suitable examples.

Q. 6 Write a short note on : Principal Component Analysis (PCA)

Q. 7 Write a short note on : Multidimensional Scaling

Q. 8 Explain the concept of Matrix Factorization.

Chapter Ends ...