SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY:: PUTTUR
(AUTONOMOUS)
Siddharth Nagar, Narayanavanam Road – 517583

QUESTION BANK (DESCRIPTIVE)

Subject with Code: **Advanced Machine Learning(20CS0906)** Course & Branch: **B.Tech – CSM**

Regulation: **R20** Year &Sem: **III-B.Tech & II – Sem**

UNIT –I

INTRODUCTION

1	A	**Explain the working process of Machine Leaning and its Applications.**
		
		Working Process of Machine Learning
		
		Machine learning (ML) is a subset of artificial intelligence (AI) that enables systems to learn and improve from experience without being explicitly programmed. The working process of ML can be broadly divided into several key steps:
		
		1. **Data Collection**:
		  ○ Gathering relevant data from various sources. The data can be structured (like databases) or unstructured (like text, images).
		2. **Data Preprocessing**:
		  ○ Cleaning the data by handling missing values, removing duplicates, and correcting errors.
		  ○ Transforming the data into a suitable format for analysis. This may involve normalization, scaling, and encoding categorical variables.
		  ○ Splitting the data into training, validation, and test sets.
		3. **Feature Engineering**:
		  ○ Selecting the most relevant features (variables) that contribute to the predictive power of the model.
		  ○ Creating new features from the existing data to improve the model's performance.
		4. **Model Selection**:
		  ○ Choosing an appropriate machine learning algorithm based on the problem type (classification, regression, clustering, etc.).
		  ○ Common algorithms include decision trees, support vector machines, neural networks, and ensemble methods.
		5. **Model Training**:
		  ○ Feeding the training data into the chosen algorithm to learn the underlying patterns.
		  ○ The model adjusts its parameters to minimize the error or maximize the accuracy.
		6. **Model Evaluation**:
		  ○ Testing the trained model on the validation set to tune hyperparameters and prevent overfitting.
		  ○ Evaluating the model's performance using metrics such as accuracy, precision, recall, F1 score, and
			[L2][CO1]	**[6M]**

ROC-AUC.

7. **Model Testing**:
   - Assessing the model's performance on the test set to ensure it generalizes well to unseen data.
8. **Model Deployment**:
   - Integrating the model into a production environment where it can make real-time predictions or decisions.
   - Monitoring the model's performance and updating it as needed based on new data and feedback.
9. **Model Maintenance**:
   - Continuously monitoring the model's performance in the production environment.
   - Retraining or updating the model as more data becomes available or as the underlying data distribution changes.

Applications of Machine Learning

Machine learning has a wide range of applications across various industries. Some of the most prominent applications include:

1. **Healthcare**:
   - **Medical Diagnosis**: Predicting diseases from medical images, lab tests, and patient records.
   - **Personalized Medicine**: Tailoring treatment plans based on individual patient data.
2. **Finance**:
   - **Fraud Detection**: Identifying fraudulent transactions based on patterns in transaction data.
   - **Algorithmic Trading**: Making automated trading decisions based on market data analysis.
3. **Retail**:
   - **Recommendation Systems**: Suggesting products to customers based on their past behavior and preferences.
   - **Inventory Management**: Predicting demand to optimize stock levels and reduce waste.
4. **Marketing**:
   - **Customer Segmentation**: Grouping customers based on their behavior and preferences for targeted marketing.
   - **Sentiment Analysis**: Analyzing customer feedback and social media to gauge public sentiment towards products and brands.
5. **Transportation**:
   - **Autonomous Vehicles**: Enabling self-driving cars to

		navigate and make decisions in real-time. ◦ **Predictive Maintenance**: Forecasting when parts of vehicles will fail to perform maintenance proactively. 6. **Manufacturing**: ◦ **Quality Control**: Detecting defects in products using image recognition and other sensor data. ◦ **Process Optimization**: Improving manufacturing processes by analyzing data from production lines. 7. **Agriculture**: ◦ **Crop Prediction**: Predicting crop yields based on weather data, soil conditions, and farming practices. ◦ **Precision Farming**: Using sensors and data analysis to optimize the use of resources like water and fertilizers. 8. **Education**: ◦ **Personalized Learning**: Adapting educational content to the needs and progress of individual students. ◦ **Automated Grading**: Using natural language processing to grade essays and assignments. 9. **Entertainment**: ◦ **Content Recommendation**: Suggesting movies, music, and other content to users based on their viewing/listening history. ◦ **Game AI**: Creating intelligent behavior in non-player characters in video games.		
	B	**Analyze Well Posed Problems in machine learning with examples**. The formal definition of Well posed learning problem is, "A computer program is said to learn from Experience E when given a task T, and some performance measure P. If it performs on T with a performance measure P, then it upgrades with experience E. To break it down, the three important components of a well-posed learning problem are, Task Performance Measure Experience To understand the topic better let's have a look at a few classical examples, Learning to play Checkers: A computer might improve its performance as an ability to win at the class of tasks that are about playing checkers. The performance keeps improving through experience by playing against itself. To simplify, T -> Play the checkers game.	**[L4][CO1]**	**[6M]**

P -> Percentage of games won against the opponent.
E -> Playing practice games against itself.

Handwriting Recognition:
Handwriting recognition (HWR) is a technology that converts a user's handwritten letters or words into a computer-readable format (e.g., Unicode text).

Its applications are numerous, it is used in reading postal addresses, bank forms, etc.

T -> recognizing and classifying handwritten words from images.
P -> Percentage of correctly identified words.
E -> set of handwritten words with their classifications in a database.

A Robot Driving Learning Problem:
For a robot to drive on a four-lane highway it needs a human-like understanding of all the possibilities it might encounter.

With the use of sight scanners and advanced machine learning algorithms, it can be made possible.

T –> To drive on public four-lane highways using sight scanners.
P -> the average distance progressed before an error.
E -> the order of images and steering instructions noted down while observing a human driver.

A spam filtering for emails learning problem:
A spam filter is software that detects unsolicited and undesired email and prevents it from reaching the inbox of a user.

T -> Identifying whether or not an email is spam.
P -> The percentage of emails correctly categorized as spam or nonspam.
E -> Observing how you categorize emails as spam or nonspam.

Face Recognition Problem:
A facial recognition system device is capable of matching a human face from a digital image or a video frame against a database of faces.

It works by locating and measuring facial characteristics from a given image and is often used to verify users through ID verification services.

		T -> Predicting distinct sorts of faces. P -> Ability to anticipate the largest number of different sorts of faces. E -> train the system with as many datasets of varied facial photos as possible.		
2	A	**List out various applications of Machine Learning in real world.** Machine learning (ML) has found applications in a wide range of diverse fields, revolutionizing how tasks are automated, decisions are made, and insights are extracted. Here are examples of ML applications across various domains: **Healthcare:** • **Disease Diagnosis:** ML algorithms analyze medical images (X-rays, MRIs, CT scans) for early detection of diseases like cancer. • **Drug Discovery:** ML models help identify potential drug candidates and predict their efficacy. **Finance:** • **Credit Scoring:** ML algorithms assess creditworthiness by analyzing financial data. • **Algorithmic Trading:** ML models predict market trends and optimize trading strategies. **Marketing:** • **Customer Segmentation:** ML helps identify target audiences and personalize marketing campaigns. • **Recommendation Systems:** ML algorithms power personalized recommendations in e-commerce and streaming services. **Manufacturing:** • **Predictive Maintenance:** ML predicts equipment failures, optimizing maintenance schedules and reducing downtime. • **Quality Control:** ML identifies defects in manufacturing processes by analyzing sensor data. **Transportation:** • **Traffic Prediction:** ML models predict traffic patterns, helping optimize route planning. • **Autonomous Vehicles:** ML enables self-driving cars to recognize and respond to their environment. **Natural Language Processing (NLP):** • **Chatbots:** ML-driven chatbots provide automated customer support. • **Language Translation:** NLP models translate text between languages with high accuracy. **Education:** • **Personalized Learning:** ML tailors educational content based on individual student performance. • **Grading Automation:** ML automates the grading of assessments, saving time for educators. **Agriculture:** • **Crop Monitoring:** ML models analyze satellite	**[L1][CO1]**	**[6M]**

		imagery to monitor crop health and predict yields. • **Precision Farming:** ML guides farmers in optimizing resource usage for crop production. **Cybersecurity:** • **Anomaly Detection:** ML identifies unusual patterns in network traffic, helping detect cyber threats. • **Fraud Detection:** ML algorithms identify fraudulent activities in financial transactions. **Energy:** • **Demand Forecasting:** ML predicts energy consumption patterns for efficient resource allocation. • **Fault Detection:** ML helps identify and address faults in energy infrastructure. **Human Resources:** • **Resume Screening:** ML automates the initial screening of job applications. • **Employee Retention:** ML models predict employee turnover and help in retention strategies. **Environmental Science:** • **Climate Modeling:** ML assists in modeling and predicting climate patterns. • **Species Identification:** ML helps identify plant and animal species based on images. These examples illustrate the versatility of machine learning across different domains, demonstrating its ability to enhance efficiency, provide insights, and automate complex tasks in diverse fields.		
	B	**Explain the forms of Learning in Machine Learning.** Learning in the context of machine learning can be broadly categorized into several forms, each with its own characteristics and applications. Here are some key forms of learning in machine learning: **Supervised Learning:** • **Definition:** In supervised learning, the model is trained on a labeled dataset, where the input data is paired with corresponding output labels. The goal is to learn a mapping from inputs to outputs. • **Examples:** Classification and regression tasks. • **Applications:** Image recognition, spam detection, predicting house prices. **Unsupervised Learning:** • **Definition:** Unsupervised learning involves training a model on unlabeled data. The algorithm must find patterns, relationships, or structures within the data without explicit guidance. • **Examples:** Clustering, dimensionality reduction, and association. • **Applications:** Customer segmentation, anomaly detection, topic modeling. **Semi-Supervised Learning:**	**[L2][CO1]**	**[6M]**

- **Definition:** Semi-supervised learning combines elements of both supervised and unsupervised learning. The model is trained on a dataset containing both labeled and unlabeled examples.
- **Examples:** Training with a small labeled dataset and a large unlabeled dataset.
- **Applications:** Document categorization, speech recognition, fraud detection.

Reinforcement Learning:

- **Definition:** Reinforcement learning involves an agent that interacts with an environment and learns to make decisions by receiving feedback in the form of rewards or punishments.
- **Examples:** Game playing, robotic control, autonomous systems.
- **Applications:** AlphaGo, self-driving cars, robotic navigation.

Self-Supervised Learning:

- **Definition:** Self-supervised learning is a type of unsupervised learning where the model is trained to predict parts of the input data without explicit labels. It creates its own supervisory signal during training.
- **Examples:** Pre-training models on pretext tasks, such as predicting missing parts of an image.
- **Applications:** Natural language understanding, image representation learning.

Transfer Learning:

- **Definition:** Transfer learning involves pre-training a model on one task or domain and then fine-tuning it on a different but related task or domain.
- **Examples:** Using pre-trained neural network models for image classification and adapting them to a specific task.
- **Applications:** Image recognition, natural language processing, domain adaptation.

Meta-Learning:

- **Definition:** Meta-learning, or learning to learn, involves training a model to adapt quickly to new tasks with minimal examples by leveraging knowledge gained from previous tasks.
- **Examples:** Few-shot learning, learning to adapt across diverse tasks.
- **Applications:** Rapid adaptation to new tasks with limited data.

Ensemble Learning:

- **Definition:** Ensemble learning combines predictions from multiple models to improve overall performance. It can be applied in both supervised and unsupervised learning.
- **Examples:** Random Forest, boosting algorithms.
- **Applications:** Classification, regression, anomaly detection.

Understanding the different forms of learning is crucial for selecting the appropriate approach for a given machine learning task and domain. The choice of learning paradigm depends on factors such as the availability of labeled data, the nature of the problem, and the

		desired outcomes.		
3	A	**Differentiate Machine learning and Artificial Intelligence.**	**[L6][CO5]**	**[6M]**

ARTIFICIAL INTELLIGENCE	**MACHINE LEARNING**
1956 The terminology "Artificial Intelligence" was originally used by John McCarthy, who also hosted the first AI conference.	The terminology "Machine Learning" was first used in 1952 by IBM computer scientist Arthur Samuel, a pioneer in artificial intelligence and computer games.
AI stands for Artificial intelligence, where intelligence is defined as the ability to acquire and apply knowledge.	ML stands for Machine Learning which is defined as the acquisition of knowledge or skill
AI is the broader family consisting of ML and DL as its components.	Machine Learning is the subset of Artificial Intelligence.
The aim is to increase the chance of success and not accuracy.	The aim is to increase accuracy, but it does not care about; the success
AI is aiming to develop an intelligent system capable of performing a variety of complex jobs. decision-making	Machine learning is attempting to construct machines that can only accomplish the jobs for which they have been trained.
It works as a computer program that does smart work.	Here, the tasks systems machine takes data and learns from data.
The goal is to simulate natural intelligence to solve complex problems.	The goal is to learn from data on certain tasks to maximize the performance on that task.
AI has a very broad variety of applications.	The scope of machine learning is constrained.

	new things from data.
It is developing a system that mimics humans to solve problems.	It involves creating self-learning algorithms.
AI will go for finding the optimal solution.	ML will go for a solution whether it is optimal or not.
AI leads to intelligence or wisdom.	ML leads to knowledge.
AI is a broader family consisting of ML and DL as its components.	ML is a subset of AI.
Three broad categories of AI are : 1. Artificial Narrow Intelligence (ANI) 2. Artificial General Intelligence (AGI) 3. Artificial Super Intelligence (ASI)	Three broad categories of ML are : 1. Supervised Learning 2. Unsupervised Learning 3. Reinforcement Learning
AI can work with structured, semi-structured, and unstructured data.	ML can work with only structured and semi-structured data.
AI's key uses include- • Siri, customer service via chatbots • Expert Systems • Machine Translation like Google Translate • Intelligent humanoid robots such as Sophia, and so on.	The most common uses of machine learning- • Facebook's automatic friend suggestions • Google's search algorithms • Banking fraud analysis • Stock price forecast • Online recommender systems, and so on.
AI refers to the broad field of creating machines that can simulate human intelligence and perform tasks such as understanding natural language, recognizing images and sounds, making decisions, and solving complex problems.	ML is a subset of AI that involves training algorithms on data to make predictions, decisions, and recommendations.

AI is a broad concept that includes various methods for creating intelligent machines, including rule-based systems, expert systems, and machine learning algorithms. AI systems can be programmed to follow specific rules, make logical inferences, or learn from data using ML.	focuses on teaching machines how to learn from data without being explicitly programmed, using algorithms such as neural networks, decision trees, and clustering.
AI systems can be built using both structured and unstructured data, including text, images, video, and audio. AI algorithms can work with data in a variety of formats, and they can analyze and process data to extract meaningful insights.	In contrast, ML algorithms require large amounts of structured data to learn and improve their performance. The quality and quantity of the data used to train ML algorithms are critical factors in determining the accuracy and effectiveness of the system.
AI is a broader concept that encompasses many different applications, including robotics, natural language processing, speech recognition, and autonomous vehicles. AI systems can be used to solve complex problems in various fields, such as healthcare, finance, and transportation.	ML, on the other hand, is primarily used for pattern recognition, predictive modeling, and decision making in fields such as marketing, fraud detection, and credit scoring.
AI systems can be designed to work autonomously or with minimal human intervention, depending on the complexity of the task. AI systems can make decisions and take actions based on the data and rules provided to them.	In contrast, ML algorithms require human involvement to set up, train, and optimize the system. ML algorithms require the expertise of data scientists, engineers, and other professionals to design and implement the system.

B **Describe Types of Data in Machine Learning.** **[L2][CO1]** **[6M]**

In machine learning, data can be broadly categorized into two types based on its structure: structured data and unstructured data. Each type has its own characteristics, and the choice between structured and unstructured data often depends on the nature of the problem being addressed and the available data sources.

Structured Data:

- **Definition:** Structured data is highly organized and formatted in a specific way. It is typically represented in tabular form, with rows and columns, where each column corresponds to a specific attribute or feature.

- **Examples:** Databases, spreadsheets, CSV files, relational databases.
- **Characteristics:**
  - Well-defined and fixed schema.
  - Easy to query and analyze.
  - Suitable for traditional relational databases.
  - Commonly used in business applications, finance, and datasets with clear relationships.

Unstructured Data:

- **Definition:** Unstructured data lacks a predefined data model or structure. It doesn't fit neatly into tables or rows and may include text, images, audio, video, or other formats without a clear organization.
- **Examples:** Text documents, images, videos, audio recordings, social media posts.
- **Characteristics:**
  - No fixed schema; data may vary in format and size.
  - Challenging to analyze using traditional relational databases.
  - Requires advanced techniques for feature extraction and analysis.
  - Commonly found in natural language processing (NLP), computer vision, and multimedia applications.

Semi-Structured Data:

- **Definition:** Semi-structured data falls between structured and unstructured data. It may have some level of structure, but it doesn't conform to a rigid schema.
- **Examples:** JSON, XML, log files.
- **Characteristics:**
  - Contains some organizational elements (e.g., tags in XML).
  - Offers more flexibility than structured data.
  - Requires specialized tools for extraction and analysis.

Handling Diversity in Machine Learning:

- **Data Integration:** Combining structured and unstructured data for a comprehensive analysis.
- **Feature Engineering:** Extracting meaningful features from unstructured data (e.g., using text embeddings for NLP or deep learning models for image recognition).
- **Text Processing:** Techniques like natural language processing (NLP) can be applied to analyze and extract information from unstructured text data.
- **Image and Signal Processing:** Utilizing techniques such as computer vision or signal processing to extract features from unstructured data like images or audio.

In many real-world applications, a combination of structured and unstructured data is encountered. Leveraging the strengths of both types and employing appropriate preprocessing and feature extraction techniques is crucial for building effective machine learning models. Understanding the characteristics and challenges associated with each type of data is essential for making informed decisions throughout the

		machine learning pipeline.		
4	A	**In How many ways the data can be represented in Machine Learning.**	**[L1][CO5]**	**[6M]**

DATA REPRESENTATION

Data representation is a critical aspect of machine learning (ML) as it significantly influences the performance of models. Proper representation of data helps the machine learning algorithms to learn patterns, relationships, and features effectively. Here are some common methods of data representation in machine learning:

Numerical Representation:

- **Scalar Representation:** Single numerical values represent individual data points (e.g., temperature, age).
- **Vector Representation:** Arrays of numerical values, often used to represent features of an instance.
- **Matrix Representation:** Two-dimensional arrays are used when dealing with two or more features.

Categorical Representation:

- **One-Hot Encoding:** Converts categorical variables into binary vectors with only one element as 1 and the rest as 0.
- **Label Encoding:** Assigns a unique numerical label to each category.
- **Embeddings:** Learns a dense representation for categorical variables, especially useful in natural language processing (NLP) tasks.

Textual Representation:

- **Bag of Words (BoW):** Represents text as an unordered set of words, ignoring grammar and word order.
- **TF-IDF (Term Frequency-Inverse Document Frequency):** Weights words based on their frequency in a document relative to their frequency across all documents.
- **Word Embeddings:** Utilizes pre-trained word vectors or learns embeddings specific to the task using techniques like Word2Vec or GloVe.

Temporal Representation:

- **Time Series Data:** Represents data points collected over time.
- **Lagged Variables:** Includes past values of variables to capture temporal dependencies.
- **Temporal Embeddings:** Represents time-based patterns in continuous or categorical form.

Image Representation:

- **Pixel Values:** Represents images as matrices of pixel intensity values.
- **Feature Extraction:** Uses techniques like Convolutional Neural Networks (CNNs) to automatically extract hierarchical features from images.
- **Histograms of Oriented Gradients (HOG):** Represents the distribution of gradient orientations in an image.

Graph Representation:

- **Adjacency Matrix:** Represents relationships between

		entities in a graph. • **Node Embeddings:** Learns vector representations for nodes in a graph. • **Graph Neural Networks (GNNs):** Utilizes neural networks to process graph-structured data. **Audio Representation:** • **Spectrogram:** Represents audio signals in terms of their frequency content over time. • **Mel-Frequency Cepstral Coefficients (MFCC):** Represents features related to the human auditory system. • **Waveform Representation:** Directly uses the amplitude of the sound wave. **Spatial Representation:** • **Coordinate Systems:** Represents spatial data using coordinates (e.g., latitude and longitude). • **Geospatial Features:** Extracts features related to spatial patterns and relationships. Proper data representation is crucial for the success of machine learning models. The choice of representation depends on the nature of the data and the requirements of the specific machine learning task. It's often essential to preprocess and transform raw data into a suitable format before feeding it into machine learning algorithms.		
	B	**Compare structured , unstructured and semi structured data in machine learning**	**[L5][CO2]**	**[6M]**

5	**Explain about the three different types of machine learning techniques with neat diagrams.**	**[L2][CO3]**	**[12M]**

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

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

Supervised Machine Learning

Supervised learning is the types of machine learning in which machines are trained using well "labelled" training data, and on basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output.

In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns in the supervision of the teacher.

Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to **find a mapping function to map the input variable(x) with the output variable(y)**.

In the real-world, supervised learning can be used for **Risk Assessment,**

Image classification, Fraud Detection, spam filtering, etc.0:04/05:45

g Techniques

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:

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.

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.

Types of supervised Machine learning Algorithms:

Supervised learning can be further divided into two types of problems:

1. 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

2. 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.

Spam Filtering,

- Random Forest

- Decision Trees
- Logistic Regression
- Support vector Machines

Advantages of Supervised learning:

- With the help of supervised learning, the model can predict the output on the basis of prior experiences.
- In supervised learning, we can have an exact idea about the classes of objects.
- Supervised learning model helps us to solve various real-world problems such as **fraud detection, spam filtering**, etc.

Disadvantages of supervised learning:

- Supervised learning models are not suitable for handling the complex tasks.
- Supervised learning cannot predict the correct output if the test data is different from the training dataset.
- Training required lots of computation times.
- In supervised learning, we need enough knowledge about the classes of object.

Unsupervised Machine Learning

Unsupervised learning is a type of machine learning in which models are trained using unlabeled dataset and are allowed to act on that data without any supervision.

Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data. The goal of unsupervised learning is to **find the underlying structure of dataset, group that data according to similarities, and represent that dataset in a compressed format**.

Example: Suppose the unsupervised learning algorithm is given an input dataset containing images of different types of cats and dogs. The algorithm is never trained upon the given dataset, which means it does not have any idea about the features of the dataset. The task of the unsupervised learning algorithm is to identify the image features on their own. Unsupervised learning algorithm will perform this task by clustering the image dataset into the groups according to similarities between images.

Keep Watching

Why use Unsupervised Learning?

Below are some main reasons which describe the importance of Unsupervised Learning:

- Unsupervised learning is helpful for finding useful insights from the data.
- Unsupervised learning is much similar as a human learns to think by their own experiences, which makes it closer to the real AI.
- Unsupervised learning works on unlabeled and uncategorized data which make unsupervised learning more important.
- In real-world, we do not always have input data with the corresponding output so to solve such cases, we need unsupervised learning.

Working of Unsupervised Learning

Working of unsupervised learning can be understood by the below diagram:

Here, we have taken an unlabeled input data, which means it is not categorized and corresponding outputs are also not given. Now, this unlabeled input data is fed to the machine learning model in order to train it. Firstly, it will interpret the raw data to find the hidden patterns from the data and then will apply suitable algorithms such as k-means clustering, Decision tree, etc.

Once it applies the suitable algorithm, the algorithm divides the data objects into groups according to the similarities and difference between the objects.

Types of Unsupervised Learning Algorithm:

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

- **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.
- **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.

Unsupervised Learning algorithms:
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**
- **Apriority algorithm**

- **Singular value decomposition**

Advantages of Unsupervised Learning

- Unsupervised learning is used for more complex tasks as compared to supervised learning because, in unsupervised learning, we don't have labeled input data.
- Unsupervised learning is preferable as it is easy to get unlabeled data in comparison to labeled data.

Disadvantages of Unsupervised Learning

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

Reinforcement learning

It is an area of Machine Learning. It is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behaviour or path it should take in a specific situation. Reinforcement learning differs from supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of a training dataset, it is bound to learn from its experience.

Main points in Reinforcement learning –

- Input: The input should be an initial state from which the model will start
- Output: There are many possible outputs as there are a variety of solutions to a particular problem
- Training: The training is based upon the input, The model will return a state and the user will decide to reward or punish the model based on its output.
- The model keeps continues to learn.
- The best solution is decided based on the maximum reward.

6	A	**Illustrate the domain knowledge for the productive use of Machine learning**	**[L3][CO1]**	**[6M]**

DOMAIN KNOWLEDGE IN MACHINE LEARNING

Domain knowledge is a crucial component for the productive and effective use of machine learning (ML) in any specific field or industry. Having a deep understanding of the domain where ML is being applied can significantly enhance the quality of problem formulation, data preprocessing, model development, and result interpretation. Here are key aspects of domain knowledge for productive use of machine learning:

Problem Formulation:

- **Define Clear Objectives:** Understand the specific problems or challenges within the domain that machine learning can address.
- **Identify Key Metrics:** Define the metrics that matter most in the given domain to evaluate the success of the machine learning solution.

Data Understanding:

- **Data Relevance:** Recognize what data is critical for the problem at hand and understand its significance.
- **Data Sources and Quality:** Be aware of the sources of data, potential biases, and the quality of data available in the domain.

Feature Engineering:

- **Domain-specific Features:** Leverage knowledge about the domain to create relevant features that capture important aspects of the data.
- **Dimensionality Reduction:** Identify and reduce irrelevant or redundant features based on domain insights.

Model Selection and Customization:

- **Algorithm Selection:** Choose machine learning algorithms that align with the characteristics of the data and the goals of the domain.
- **Hyperparameter Tuning:** Adjust model parameters based on domain knowledge to improve performance.

Interpretability and Explainability:

- **Model Interpretation:** Understand the implications of model predictions in the context of the domain.
- **Explainability:** Choose models that offer transparency and interpretability, especially in fields where model decisions have significant consequences.

Addressing Domain-specific Challenges:

- **Accounting for Seasonality:** Understand and account for seasonal patterns if they are relevant to the domain.
- **Handling Imbalanced Data:** If the data is imbalanced, employ techniques that are suitable for the specific domain challenges.

Ethical Considerations:

- **Bias and Fairness:** Be aware of potential biases in the data and models, and take steps to mitigate them.
- **Legal and Ethical Implications:** Understand and adhere to ethical and legal standards specific to the domain.

Continuous Learning:

- **Stay Informed:** Keep up with advancements in both the domain and machine learning techniques relevant to the domain.
- **Adapt to Changes:** Be adaptable to changes in the domain and update machine learning models accordingly.

Collaboration with Domain Experts:

- **Team Collaboration:** Work closely with domain experts to ensure that the machine learning approach aligns with the real-world requirements.

- **Feedback Loop:** Establish a feedback loop with domain experts to continuously improve the model's performance.

Communication Skills:

- **Translate Insights:** Effectively communicate the insights derived from machine learning models to stakeholders who may not have a technical background.

B **Compare Data Mining Vs Machine Learning** [L6][CO1] [6M]

Factors	Data Mining	Machine Learning
Origin	Traditional databases with unstructured data.	It has an existing algorithm and data.
Meaning	Extracting information from a huge amount of data.	Introduce new Information from data as well as previous experience.
History	In 1930, it was known as knowledge discovery in databases(KDD).	The first program, i.e., Samuel's checker playing program, was established in 1950.
Responsibility	Data Mining is used to obtain the rules from the existing data.	Machine learning teaches the computer, how to learn and comprehend the rules.
Abstraction	Data mining abstract from the data warehouse.	Machine learning reads machine.
Applications	In compare to machine learning, data mining can produce outcomes on the lesser volume of data. It is also used in cluster analysis.	It needs a large amount of data to obtain accurate results. It has various applications, used in web search, spam filter, credit scoring, computer design, etc.
Nature	It involves human interference more towards the manual.	It is automated, once designed and implemented, there is no need for human effort.
Techniques involve	Data mining is more of research using a technique like a machine learning.	It is a self-learned and train system to do the task precisely.
Scope	Applied in the limited fields.	It can be used in a vast area.

7 A **Compare Supervised Learning and Unsupervised Learning.** [L6][CO1] [6M]

Supervised Learning	Unsupervised Learning
Supervised learning algorithms are trained using labeled data.	Unsupervised learning algorithms are trained using unlabeled data.
Supervised learning model takes direct feedback to check if it is predicting correct output or not.	Unsupervised learning model does not take any feedback.
Supervised learning model predicts the output.	Unsupervised learning model finds the hidden patterns in data.
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.

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.
Supervised learning needs supervision to train the model.	Unsupervised learning does not need any supervision to train the model.
Supervised learning can be categorized in **Classification** and **Regression** problems.	Unsupervised Learning can be classified in **Clustering** and **Associations** problems.
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.
Supervised learning model produces an accurate result.	Unsupervised learning model may give less accurate result as compared to supervised learning.
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.
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.

B **Analyze Reinforcement Learning with neat diagram..** [L4][CO1] [6M]

Reinforcement learning is an area of Machine Learning. It is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behaviour or path it should take in a specific situation. Reinforcement learning differs from supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of a training dataset, it is bound to learn from its experience.

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

The above image shows the robot, diamond, and fire. The goal of the robot is to get the reward that is the diamond and avoid the hurdles that are fired. The robot learns by trying all the possible paths and then choosing the path which gives him the reward with the least hurdles. Each right step will give the robot a reward and each wrong step will subtract the reward of the robot. The total reward will be calculated when it reaches the final reward that is the diamond.

Main points in Reinforcement learning –

- Input: The input should be an initial state from which the model will start
- Output: There are many possible outputs as there are a variety of solutions to a particular problem
- Training: The training is based upon the input, The model will return a state and the user will decide to reward or punish the model based on its output.
- The model keeps continues to learn.
- The best solution is decided based on the maximum reward.

Types of Reinforcement: There are two types of Reinforcement:

1. **Positive –**
   Positive Reinforcement is defined as when an event, occurs due to a particular behavior, increases the strength and the frequency of the behavior. In other words, it has a positive effect on behavior.
   Advantages of reinforcement learning are:
   - Maximizes Performance
   - Sustain Change for a long period of time
   - Too much Reinforcement can lead to an overload of states which can diminish the results
2. **Negative –**
   Negative Reinforcement is defined as strengthening of behavior because a negative condition is stopped or avoided.

Advantages of reinforcement learning:

- Increases Behavior
- Provide defiance to a minimum standard of performance
- It Only provides enough to meet up the minimum behavior

Various Practical applications of Reinforcement Learning –

- RL can be used in robotics for industrial automation.
- RL can be used in machine learning and data processing
- RL can be used to create training systems that provide custom instruction and materials according to the requirement of students.

RL can be used in large environments in the following situations:
A model of the environment is known, but an analytic solution is not available

8	**Discuss the Diversity of Data in Machine learning with suitable examples.**	**[L2][CO2]**	**[12M]**

The diversity of data in machine learning is a critical factor in developing robust and generalizable models. Diverse data ensures that machine learning models can learn to recognize patterns and make predictions across a wide range of scenarios, reducing bias and improving performance in real-world applications. Here are several dimensions of data diversity with suitable examples:

1. Feature Diversity

Feature diversity refers to the variety of input features (variables) used to train a machine learning model.

Example: Predicting House Prices

- **Features**: Number of bedrooms, square footage, location, age of the property, proximity to schools and public transportation, neighborhood crime rate, etc.
- **Diversity**: Incorporating features from various categories (e.g., property characteristics, location-based features, socio-economic factors) ensures a more comprehensive model.

2. Class Diversity

Class diversity involves having multiple classes or categories within the target variable, particularly important for classification problems.

Example: Image Classification

- **Classes**: Cats, dogs, birds, cars, airplanes, etc.
- **Diversity**: A diverse dataset with multiple classes helps the model distinguish between different types of objects, improving its generalization ability to new, unseen images.

3. Geographical Diversity

Geographical diversity refers to data collected from various geographical locations, ensuring that the model is not biased towards a specific region.

Example: Weather Prediction

- **Geographical Locations**: Data collected from different cities, countries, and climate zones.
- **Diversity**: This ensures the model can make accurate predictions

for various locations, accommodating different weather patterns and conditions.

4. Temporal Diversity

Temporal diversity includes data collected over different time periods, capturing temporal variations and trends.

Example: Stock Market Prediction

- **Time Periods**: Data from different years, months, and days.
- **Diversity**: Incorporating data from various time periods helps the model understand market trends, seasonal patterns, and economic cycles.

5. Demographic Diversity

Demographic diversity ensures that data represents various demographic groups, reducing biases related to age, gender, ethnicity, etc.

Example: Health Outcome Prediction

- **Demographics**: Age, gender, ethnicity, socio-economic status.
- **Diversity**: A diverse dataset in terms of demographics ensures the model can make accurate predictions for a broad population, improving fairness and equity in healthcare.

6. Modal Diversity

Modal diversity involves using data from multiple sources or modalities, such as text, images, audio, and sensor data.

Example: Autonomous Vehicles

- **Modalities**: Camera images, LiDAR, radar, GPS data, and accelerometer data.
- **Diversity**: Combining different types of data helps the model perceive and navigate the environment more accurately, improving safety and reliability.

7. Contextual Diversity

Contextual diversity refers to data representing various contexts or conditions under which observations are made.

Example: Speech Recognition

- **Contexts**: Different accents, background noises, speech speeds, and speaking environments (e.g., quiet rooms, noisy streets).
- **Diversity**: Ensuring contextual diversity helps the model recognize speech accurately in various real-world scenarios.

Importance and Challenges

- **Importance**: Diverse data helps build more robust, fair, and generalizable models that perform well across different conditions and populations.

- **Challenges**: Collecting diverse data can be challenging due to logistical, privacy, and cost constraints. Additionally, ensuring that the diversity does not introduce noise or irrelevant variability is crucial.

Strategies to Enhance Data Diversity

- **Data Augmentation**: Techniques like rotation, scaling, and flipping in image data to artificially increase diversity.
- **Synthetic Data Generation**: Creating synthetic data that mimics real-world diversity.
- **Transfer Learning**: Using pre-trained models on diverse datasets and fine-tuning them on specific tasks.
- **Crowdsourcing**: Leveraging platforms to collect data from a diverse group of contributors.

By addressing data diversity across these dimensions, machine learning practitioners can develop more accurate, reliable, and fair models suitable for a wide range of applications.

4o

9 **Analyze the Intelligent Machine Well Posed Problems and representation of data in machine learning with suitable examples.** **[L4][CO1]** **[12M]**

The concept of "well-posed problems" is fundamental in the field of artificial intelligence and machine learning. A well-posed problem is one that has a unique solution, is stable and continuous with respect to initial conditions, and the solution depends continuously on the input data. When we talk about well-posed problems in the context of intelligent machines, we are essentially addressing the clarity and feasibility of the tasks we expect these machines to perform.

Here are some key considerations when formulating well-posed problems for intelligent machines:

Clear Objectives: Clearly define the problem you want the intelligent machine to solve. Ambiguous or ill-defined problems can lead to ineffective solutions.

Observable and Measurable Outputs: Ensure that the outputs or results of the machine's task are observable and measurable. This allows for the evaluation of the machine's performance.

Appropriate Input Data: Provide the machine with relevant and sufficient input data. The quality and quantity of data play a crucial role in the learning and decision-making processes of intelligent machines.

Consistent and Reliable Data: Ensure that the data used for training and evaluation is consistent and reliable. Inconsistencies or biases in the data can lead to skewed results and unreliable predictions.

Well-Defined Constraints: Clearly specify any constraints or limitations associated with the problem. Constraints help in defining the boundaries within which the intelligent machine needs to operate.

Ethical Considerations: Consider ethical implications and potential biases in the problem formulation. Be aware of how the machine's decisions may impact individuals or groups and take steps to mitigate biases.

Scalability: Consider the scalability of the solution. Will the machine be able to handle an increase in data or complexity of the problem over time?

Interpretability and Explainability: Depending on the application, it may be crucial for the machine's decisions to be interpretable and explainable. This is especially important in critical domains like healthcare or finance.

Adaptability: Well-posed problems should allow for adaptability over time. Intelligent machines should be able to learn from new data and update their models to improve performance.

Iterative Improvement: Frame problems in a way that allows for iterative improvement. Intelligent machines can evolve and enhance their performance through continuous learning and feedback mechanisms.

By addressing these considerations, researchers and engineers can formulate well-posed problems that contribute to the development of intelligent machines capable of robust and reliable performance in various applications.

10	A	**Analyze the basic Linear Algebra in machine learning**	[L4][CO2]	[6M]

Linear Algebra for Machine Learning

Machine learning has a strong connection with mathematics. Each machine learning algorithm is based on the concepts of mathematics & also with the help of mathematics, one can choose the correct algorithm by considering training time, complexity, number of features, etc. ***Linear Algebra is an essential field of mathematics, which defines the study of vectors, matrices, planes, mapping, and lines required for linear transformation.***

The term Linear Algebra was initially introduced in the early 18^{th} century to find out the unknowns in Linear equations and solve the equation easily; hence it is an important branch of mathematics that helps study data. Also, no one can deny that Linear Algebra is undoubtedly the important and primary thing to process the applications of Machine Learning. It is also a prerequisite to start learning Machine Learning and data science.

Linear algebra plays a vital role and key foundation in machine learning***, and it enables ML algorithms to run on a huge number of datasets***.

The concepts of linear algebra are widely used in developing algorithms in machine learning. Although it is used almost in each concept of Machine learning, specifically, it can perform the following task:

Backward Skip 10sPlay VideoForward Skip 10s

- Optimization of data.

- Applicable in loss functions, regularisation, covariance matrices, Singular Value Decomposition (SVD), Matrix Operations, and support vector machine classification.
- Implementation of Linear Regression in Machine Learning.

Besides the above uses, linear algebra is also used in neural networks and the data science field.

Basic mathematics principles and concepts like Linear algebra are the foundation of Machine Learning and Deep Learning systems. To learn and understand Machine Learning or Data Science, one needs to be familiar with linear algebra and optimization theory. In this topic, we will explain all the Linear algebra concepts required for machine learning.

B **Explain the real world applications of ML.** **[L2][CO6]** **[6M]**

APPLICATIONS OF ML

Machine learning (ML) has found applications in a wide range of diverse fields, revolutionizing how tasks are automated, decisions are made, and insights are extracted. Here are examples of ML applications across various domains:

Healthcare:

- **Disease Diagnosis:** ML algorithms analyze medical images (X-rays, MRIs, CT scans) for early detection of diseases like cancer.
- **Drug Discovery:** ML models help identify potential drug candidates and predict their efficacy.

Finance:

- **Credit Scoring:** ML algorithms assess creditworthiness by analyzing financial data.
- **Algorithmic Trading:** ML models predict market trends and optimize trading strategies.

Marketing:

- **Customer Segmentation:** ML helps identify target audiences and personalize marketing campaigns.
- **Recommendation Systems:** ML algorithms power personalized recommendations in e-commerce and streaming services.

Manufacturing:

- **Predictive Maintenance:** ML predicts equipment failures, optimizing maintenance schedules and reducing downtime.
- **Quality Control:** ML identifies defects in manufacturing processes by analyzing sensor data.

Transportation:

- **Traffic Prediction:** ML models predict traffic patterns, helping optimize route planning.
- **Autonomous Vehicles:** ML enables self-driving cars to recognize and respond to their environment.

Natural Language Processing (NLP):

- **Chatbots:** ML-driven chatbots provide automated customer support.
- **Language Translation:** NLP models translate text between languages with high accuracy.

Education:

- **Personalized Learning:** ML tailors educational content based on individual student performance.
- **Grading Automation:** ML automates the grading of assessments, saving time for educators.

Agriculture:

- **Crop Monitoring:** ML models analyze satellite imagery to monitor crop health and predict yields.
- **Precision Farming:** ML guides farmers in optimizing resource usage for crop production.

Cybersecurity:

- **Anomaly Detection:** ML identifies unusual patterns in network traffic, helping detect cyber threats.
- **Fraud Detection:** ML algorithms identify fraudulent activities in financial transactions.

Energy:

- **Demand Forecasting:** ML predicts energy consumption patterns for efficient resource allocation.
- **Fault Detection:** ML helps identify and address faults in energy infrastructure.

Human Resources:

- **Resume Screening:** ML automates the initial screening of job applications.
- **Employee Retention:** ML models predict employee turnover and help in retention strategies.

Environmental Science:

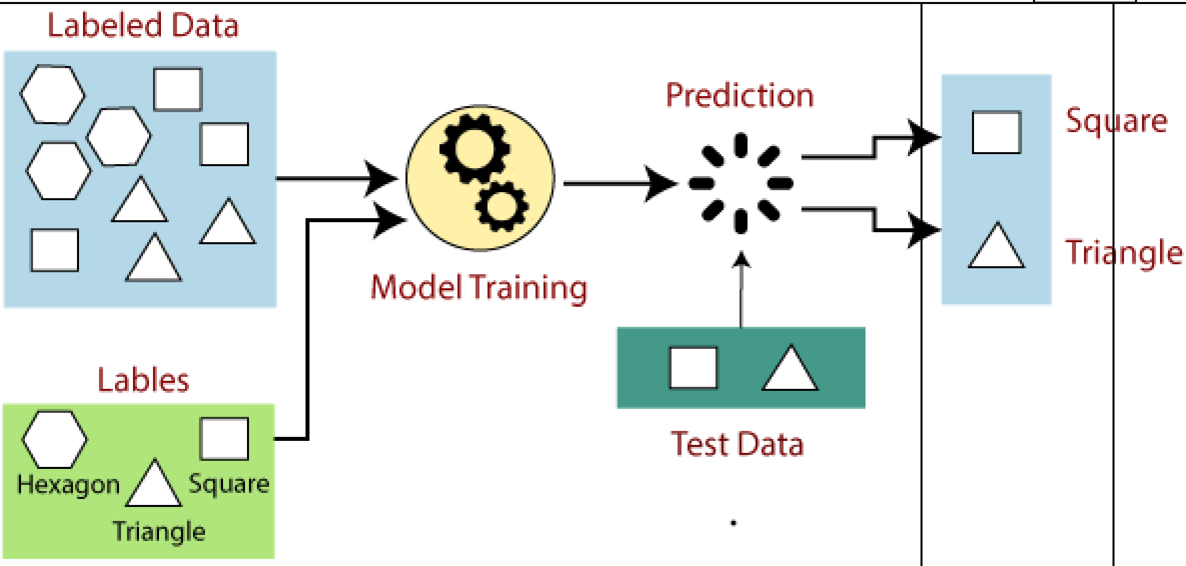
- **Climate Modeling:** ML assists in modeling and predicting climate patterns.
- **Species Identification:** ML helps identify plant and animal species based on images.

These examples illustrate the versatility of machine learning across different domains, demonstrating its ability to enhance efficiency, provide insights, and automate complex tasks in diverse fields.

UNIT-II

SUPERVISED LEARNING

1	a	**Explain about the Supervised learning with neat architecture and its techniques.** **Supervised Machine Learning** Supervised learning is the types of machine learning in which machines are trained using well "labelled" training data, and on basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output. In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns in the supervision of the teacher. Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to **find a mapping function to map the input variable(x) with the output variable(y)**. In the real-world, supervised learning can be used for **Risk Assessment, Image classification, Fraud Detection, spam filtering**, etc.0:04/05:45 g Techniques **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:	**[L2][CO2]**	**[8M]**



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.

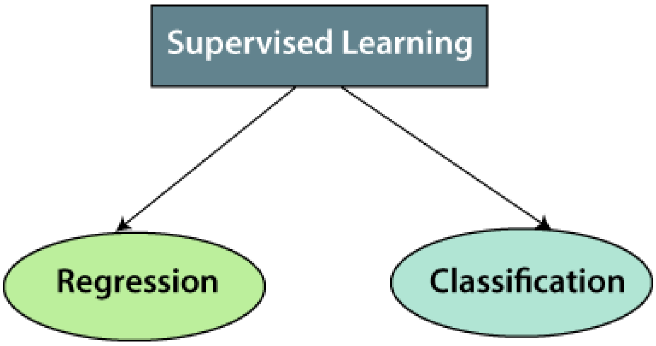
Steps Involved in Supervised Learning:

- 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.

Types of supervised Machine learning Algorithms:

Supervised learning can be further divided into two types of problems:



1. 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

2. 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.

Spam Filtering,

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

Advantages of Supervised learning:

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

- In supervised learning, we can have an exact idea about the classes of objects.
- Supervised learning model helps us to solve various real-world problems such as **fraud detection, spam filtering**, etc.

Disadvantages of supervised learning:

- Supervised learning models are not suitable for handling the complex tasks.
- Supervised learning cannot predict the correct output if the test data is different from the training dataset.
- Training required lots of computation times.
- In supervised learning, we need enough knowledge about the classes of object.

b **Differentiate Supervised Learning and Unsupervised Learning.** **[L4][CO5]** **[4M]**

Supervised Learning	**Unsupervised Learning**
Supervised learning algorithms are trained using labeled data.	Unsupervised learning algorithms are trained using unlabeled data.
Supervised learning model takes direct feedback to check if it is predicting correct output or not.	Unsupervised learning model does not take any feedback.
Supervised learning model predicts the output.	Unsupervised learning model finds the hidden patterns in data.
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.
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.
Supervised learning needs supervision to train the model.	Unsupervised learning does not need any supervision to train the model.
Supervised learning can be categorized in **Classification** and **Regression** problems.	Unsupervised Learning can be classified in **Clustering** and **Associations** problems.

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.
Supervised learning model produces an accurate result.	Unsupervised learning model may give less accurate result as compared to supervised learning.
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.
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.

2	a	**List out various Regression techniques in Machine Learning.**	**[L2][CO1]**	**[4M]**

LINEAR REGRESSION MODELS

- Regression modeling is a process of determining a relationship between one or more independent variables and one dependent or output variable.
- **Examples:**
  1. Predicting the height of a person given the age of the person.
  2. Predicting the price of the car given the car model, year of manufacturing, mileage, engine capacity, etc.

- Based on the type of functions used to represent the relationship between the dependent or output variable and independent variables, the regression models are categorized into four types. The regression models are,
  1. Simple Linear Regression
  2. Multiple Regression
  3. Polynomial Regression
  4. Logistic Regression

1. Simple Linear Regression

- Assume that there is only one independent variable x. If the relationship between x (independent variable) and y (dependent or output variable) is modeled by the relation,

$$y = a + bx$$

- then the regression model is called a linear regression model.

2. Multiple Regression

- Assume that there are multiple independent variables *say* x_1, x_2, ...x_n. If the relationship between independent variables x and dependent or output variable y is modeled by the relation,

$$y=a_0+a_1*x_1+a_2*x_2+.........+a_n*x_n$$

then the regression model is called a multiple regression model.

3. Polynomial regression

- Assume that there is only one independent variable x. If the relationship between independent variables x and dependent or output variable y is modeled by the relation,

$$y=a_0+a_1*x+a_2*x^2+........+a_n*x^n$$

- for some positive integer n>1, then we have a polynomial regression.

4. Logistic Regression

- Logistic regression is used when the dependent variable is binary (0/1, True/False, Yes/No) in nature.

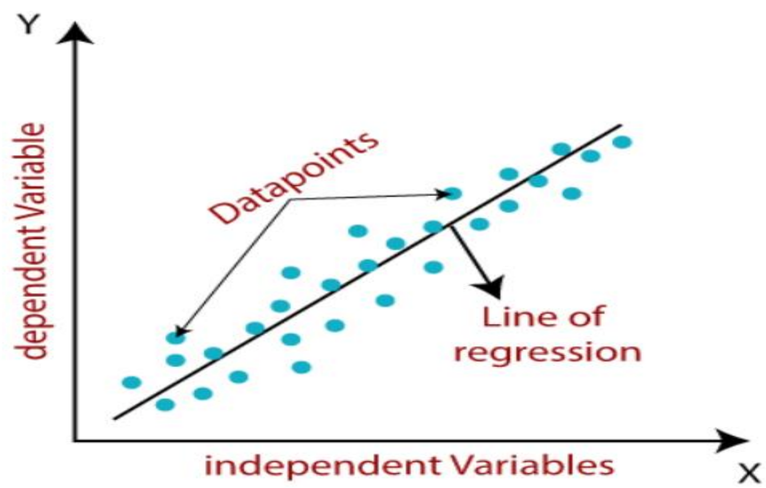
b	**Explain Linear models for Regression in Machine Learning.**	**[L2][CO1]**	**[8M]**

LINEAR REGRESSION

Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as **sales, salary, age, product price,** etc.

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

The linear regression model provides a sloped straight line representing the relationship between the variables. Consider the below image



Iathematically, we can represent a linear regression as:

$y = a_0 + a_1 x + \varepsilon$

Here,

Y= Dependent Variable (Target Variable)
X= Independent Variable (predictor Variable)
a0= intercept of the line (Gives an additional degree of freedom)
a1 = Linear regression coefficient (scale factor to each input value).
ε = random error

The values for x and y variables are training datasets for Linear Regression model representation.

Types of Linear Regression

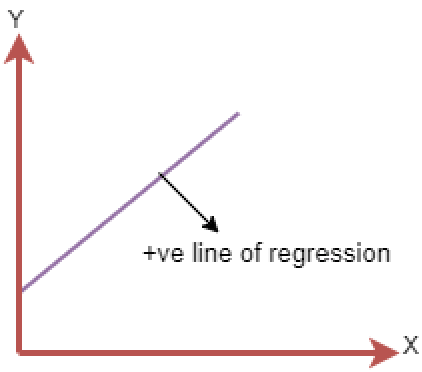
Linear regression can be further divided into two types of the algorithm:

Simple Linear Regression:
If a single independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Simple Linear Regression.
Multiple Linear regression:
If more than one independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Multiple Linear Regression.

Linear Regression Line

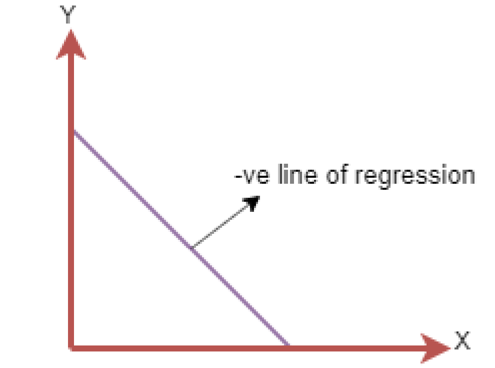
A linear line showing the relationship between the dependent and independent variables is called a **regression line**. A regression line can show two types of relationship:

- **Positive Linear Relationship:**
  If the dependent variable increases on the Y-axis and independent variable increases on X-axis, then such a relationship is termed as a Positive linear relationship.



The line equation will be: **$Y = a_0 + a_1 x$**

Negative Linear Relationship:
If the dependent variable decreases on the Y-axis and independent variable increases on the X-axis, then such a relationship is called a negative linear relationship.



The line of equation will be: $\mathbf{Y= -a_0+a_1x}$

Finding the best fit line:

When working with linear regression, our main goal is to find the best fit line that means the error between predicted values and actual values should be minimized. The best fit line will have the least error.

The different values for weights or the coefficient of lines (a_0, a_1) gives a different line of regression, so we need to calculate the best values for a_0 and a_1 to find the best fit line, so to calculate this we use cost fı

C

$$\text{MSE}= 1\frac{1}{N}\sum_{i=1}^{n}(y_i - (a_1x_i + a_0))^2$$

Where,

N=Total number of observation

Yi = Actual value

$(a1x_i+a_0)$= Predicted value.

Residuals: The distance between the actual value and predicted values is called residual. If the observed points ar far from the regression line, then the residual will be high, and so cost function will high. If the scatter points ar close to the regression line, then the residual will be small and hence the cost function.

Gradient Descent:

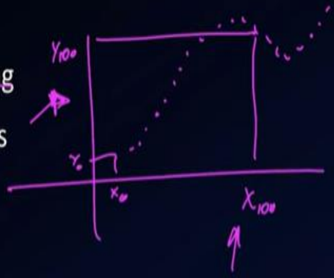
F
fı
p
F

- Gradient descent is used to minimize the MSE by calculating the gradient of the cost function.
- A regression model uses gradient descent to update the coefficients of the line by reducing the cost function.
- It is done by a random selection of values of coefficient and then iteratively update the values to reach the minimum cost function.

3	a	**Interpret the linear basis function models in supervised learning**	**[L4][CO2]**	**[6M]**
		Linear basis function models are a class of linear regression models that use a set of basis functions to transform the input variables before fitting a linear model. The basis functions are typically nonlinear functions applied to the input variables, allowing the model to capture nonlinear relationships between the inputs and the output.		

Basics

- Regression is part of Supervised Learning
- Training data set $\mathcal{D}$ with $\mathcal{N}$ observations
- $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N}$
- y = Output / Target / Label



$f(x) = y$

- Goal: learn a mapping from inputs X to outputs Y
- Make a prediction about y_* for a novel input x_*
- If these predictions are good, then we say that the model generalizes well
- Finding models that generalize well is the hardest part of machine learning

Function Fitting

1) Propose a deterministic parametric function $y = f(x, w)$
   w = Set of parameters
2) Estimate the parameters w by minimizing an error/loss/cost function $E(w)$
   => "Best fit of the data"
3) Prediction of novel inputs

Linear Basis Function Models

Scalar input X and a scalar output variable Y

$$f(x, \boldsymbol{w}) = w_0 + w_1 x$$

$$= \begin{bmatrix} w_0 & w_1 \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix} = \boldsymbol{w}^T \boldsymbol{x}$$

w_0 is the bias parameter / intercept

$f(x, w)$ is a linear function of the parameters w and a linear function of the input x

Linear Basis Function Models

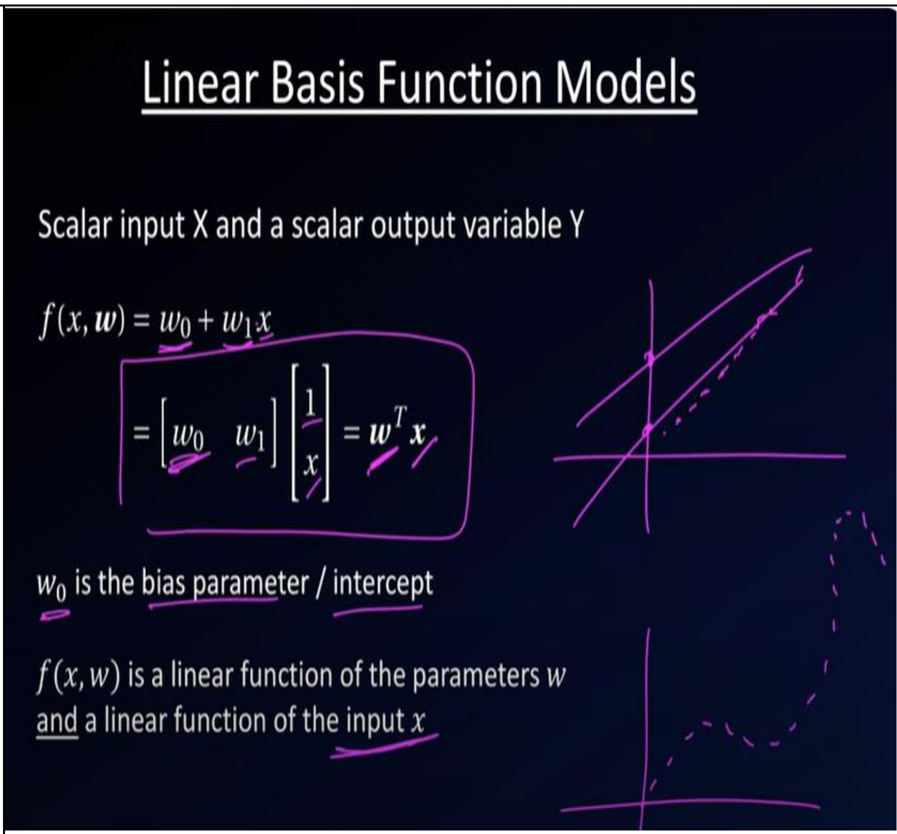
Scalar input X and a scalar output variable Y

$$f(x, \boldsymbol{w}) = w_0 + w_1 x$$

$$\boxed{= \begin{bmatrix} w_0 & w_1 \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix} = \boldsymbol{w}^T \boldsymbol{x}}$$

w_0 is the bias parameter / intercept

$f(x, w)$ is a linear function of the parameters w and a linear function of the input x

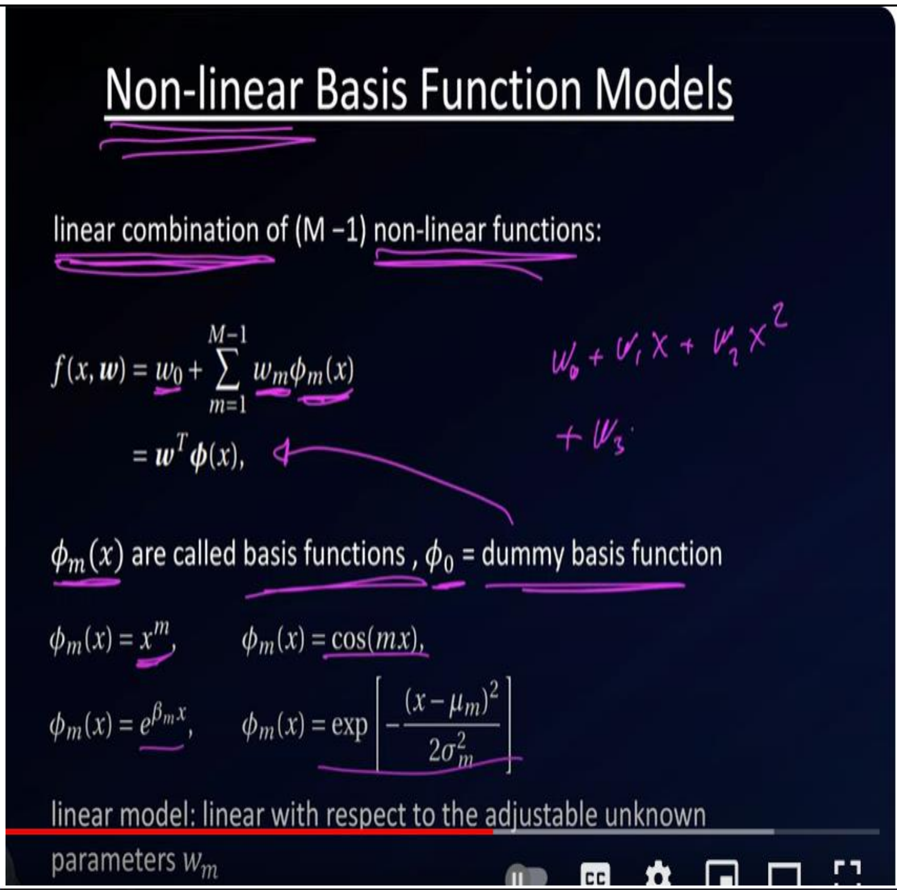


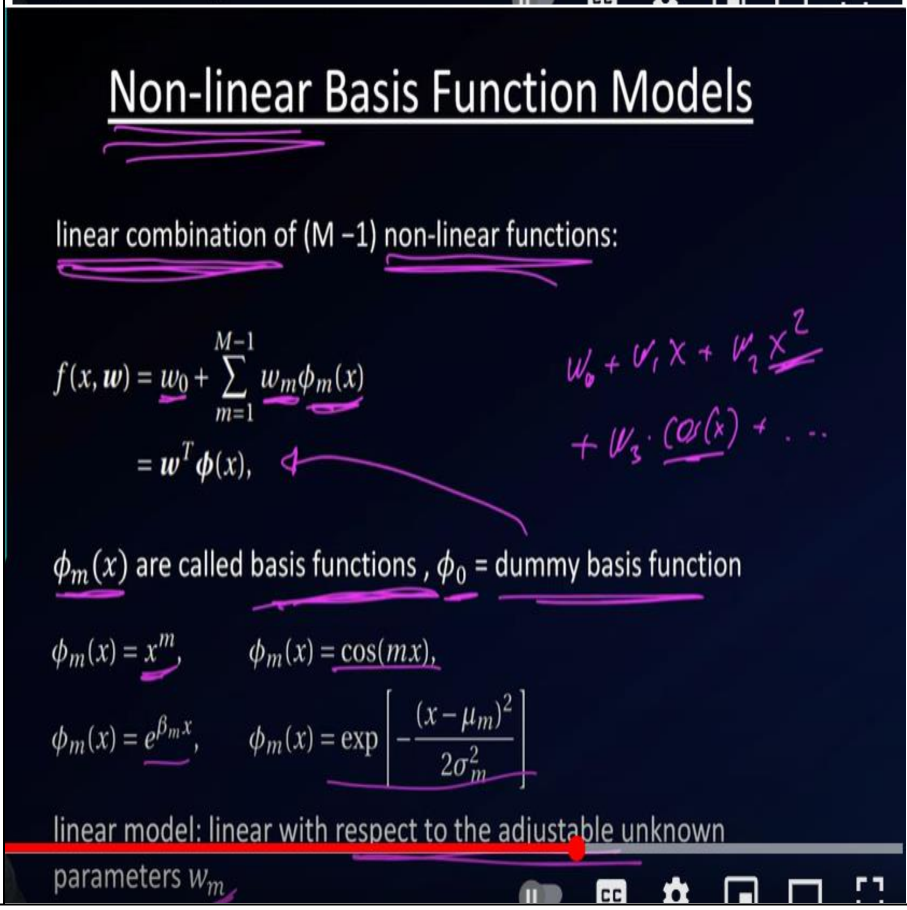
The general form of a linear basis function model is:

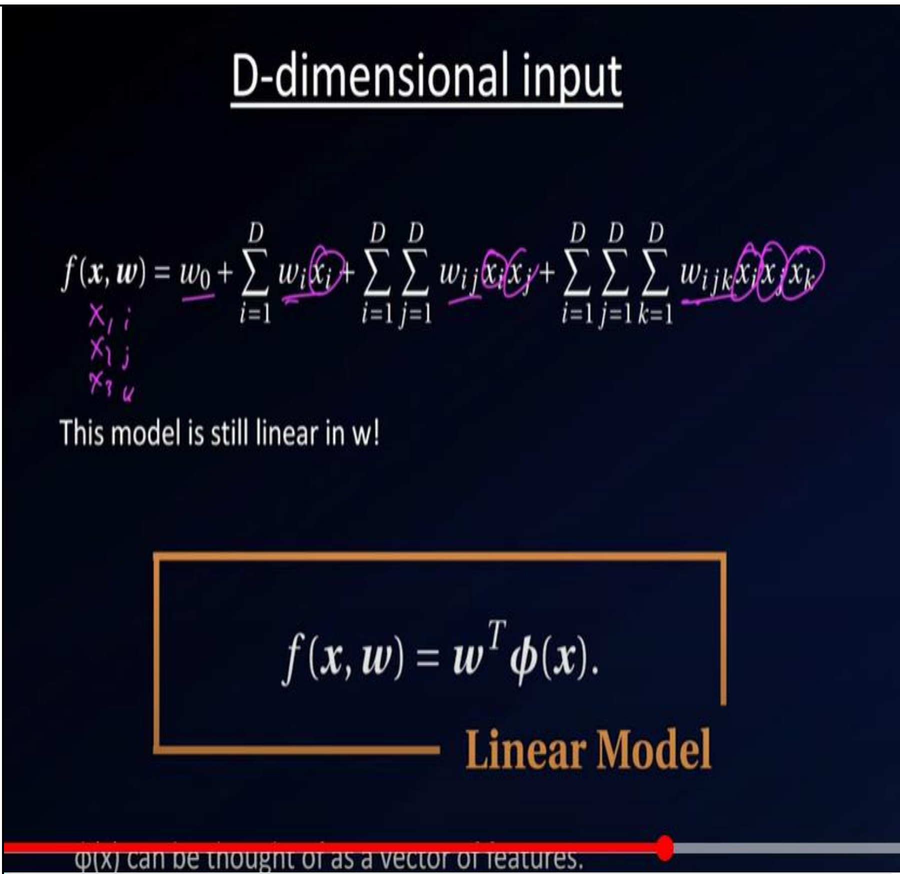
$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x})$$

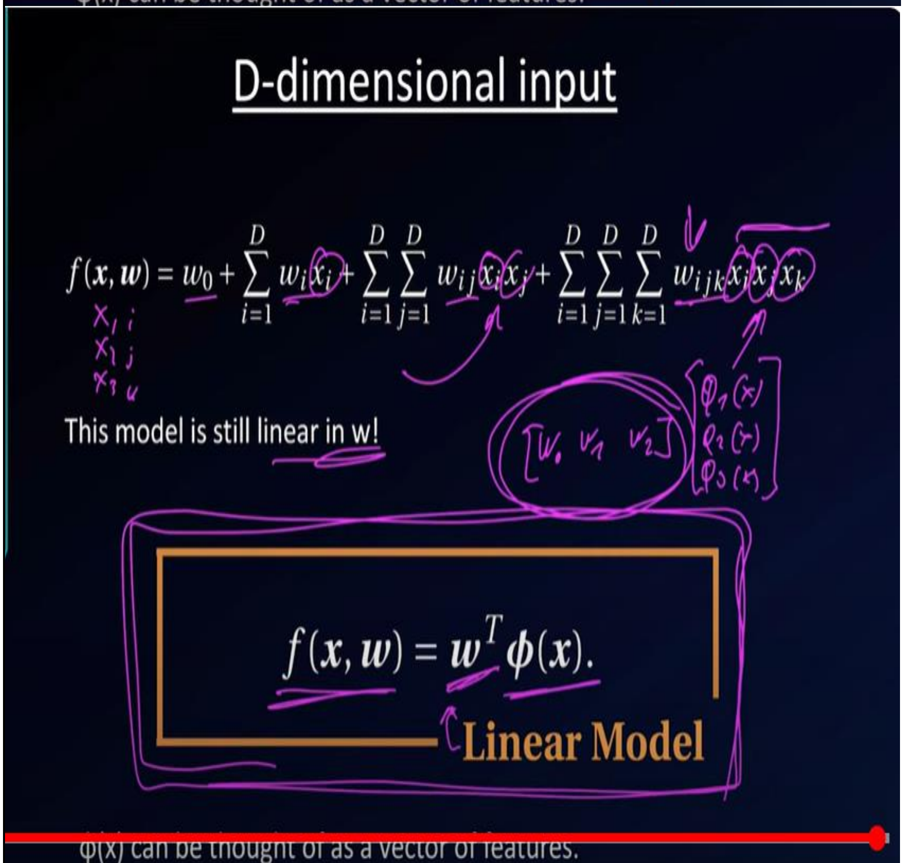
Where:

- $y(\mathbf{x}, \mathbf{w})$ is the predicted output.
- $\mathbf{x}$ is the input vector.
- $\mathbf{w}$ is the vector of parameters to be learned.
- w_0 is the bias term.
- $\phi_j(\mathbf{x})$ are the basis functions, which transform the input vector $\mathbf{x}$ into a higher-dimensional space. M is the number of basis functions.


Non-linear Basis Function Models
linear combination of (M −1) non-linear functions:
$f(x, \boldsymbol{w}) = w_0 + \sum_{m=1}^{M-1} w_m \phi_m(x)$
$= \boldsymbol{w}^T \boldsymbol{\phi}(x),$
$w_0 + w_1 x + w_2 x^2$
$+ w_3$
$\phi_m(x)$ are called basis functions , ϕ_0 = dummy basis function
$\phi_m(x) = x^m,$
$\phi_m(x) = \cos(mx),$
$\phi_m(x) = e^{\beta_m x},$
$\phi_m(x) = \exp\left[-\frac{(x-\mu_m)^2}{2\sigma_m^2}\right]$
linear model: linear with respect to the adjustable unknown
parameters w_m


Non-linear Basis Function Models
linear combination of (M −1) non-linear functions:
$f(x, \boldsymbol{w}) = w_0 + \sum_{m=1}^{M-1} w_m \phi_m(x)$
$= \boldsymbol{w}^T \boldsymbol{\phi}(x),$
$w_0 + w_1 x + w_2 x^2$
$+ w_3 \cdot \cos(x) + \ldots$
$\phi_m(x)$ are called basis functions , ϕ_0 = dummy basis function
$\phi_m(x) = x^m,$
$\phi_m(x) = \cos(mx),$
$\phi_m(x) = e^{\beta_m x},$
$\phi_m(x) = \exp\left[-\frac{(x-\mu_m)^2}{2\sigma_m^2}\right]$
linear model: linear with respect to the adjustable unknown
parameters w_m





Linear basis function models are "linear" in the sense that the model is linear in the parameters **w**, even though the transformation of the input variables through the basis functions can introduce nonlinearities in the input space. Commonly used basis functions include polynomial functions, Gaussian radial basis functions, sigmoidal functions, and piecewise linear functions.

.

b **Explain about Bias-variance decomposition techniques.** **[L2][CO2]** **[6M]**

The bias-variance decomposition is a useful theoretical tool for understanding a learning algorithm's performance characteristics. Certain algorithms have a large bias and a low variance by design, and vice versa. Bias-variance is a reducible error, in this article, we will be understanding the concept with ways to decompose the mean squared error. Following are the topics to be covered.

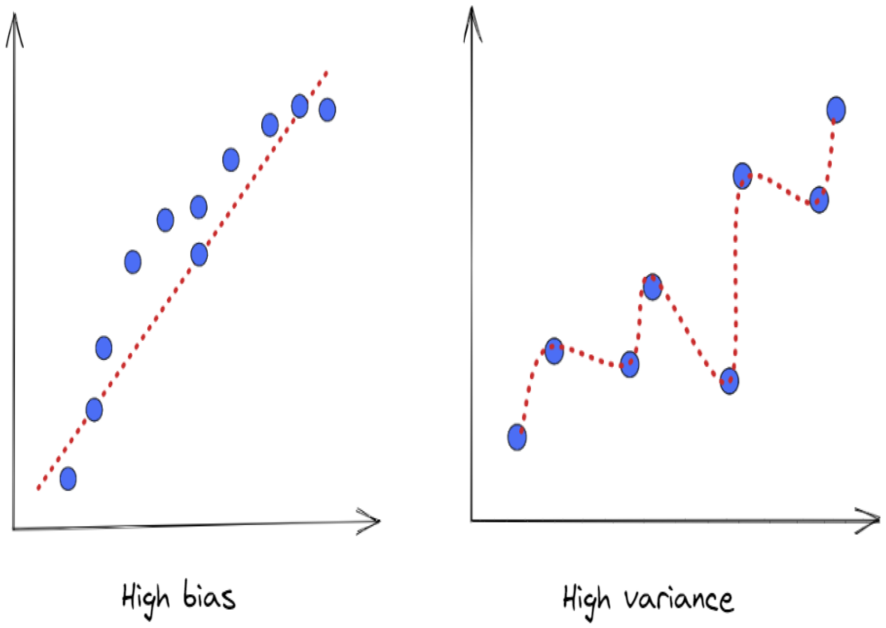
The bias is defined as the difference between the ML model's prediction of the values and the correct value. Biasing causes a substantial inaccuracy in both training and testing data. To prevent the problem of underfitting, it is advised that an algorithm be low biased at all times.

Are you looking for a complete repository of Python libraries used in data science, check out here**.**

The data predicted with high bias is in a straight-line format, which does not fit the data in the data set adequately. Underfitting of data is a term used to describe this type of fitting. This occurs when the theory is overly simplistic or linear in form.

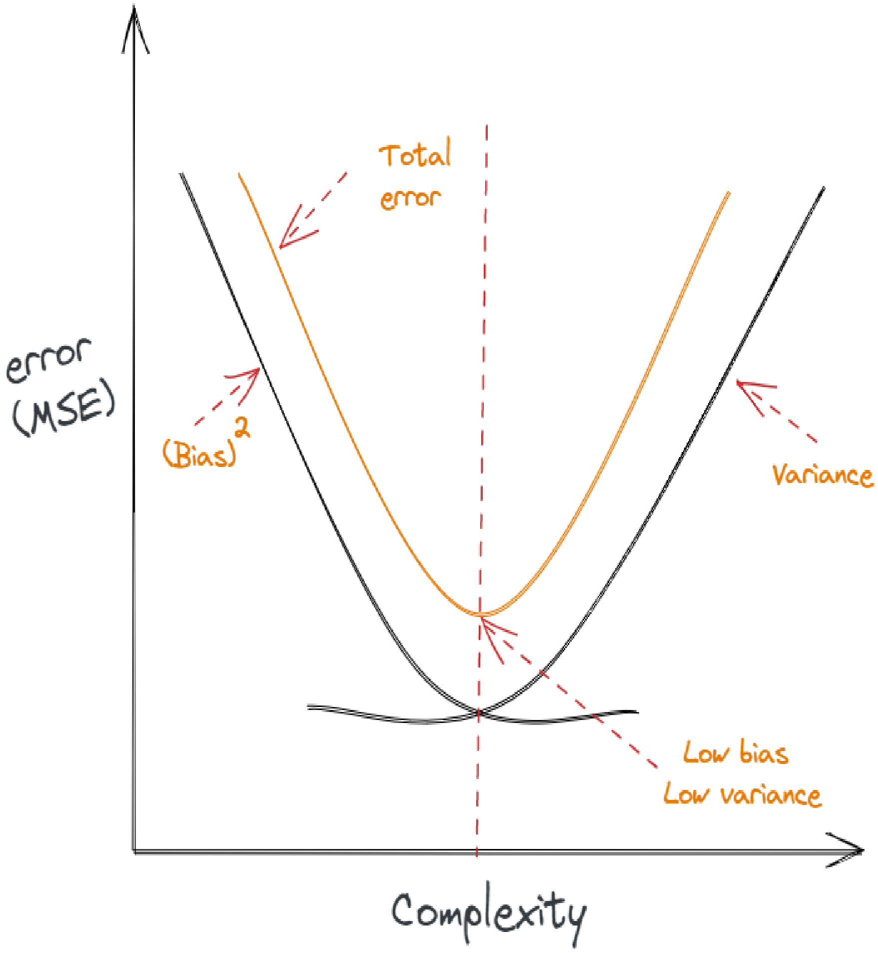
The variance of the model is the variability of model prediction for a particular data point, which tells us about the dispersion of the data. The model with high variance has a very complicated fit to the training data and so is unable to fit correctly on new data.

As a result, while such models perform well on training data, they have large error rates on test data. When a model has a large variance, this is referred to as Overfitting of Data. Variability should be reduced to a minimum while training a data model.



Bias and variance are negatively related, therefore it is essentially difficult to have an ML model with both a low bias and a low variance. When we alter the ML method to better match a specific data set, it results in reduced bias but increases variance. In this manner, the model will fit the data set while increasing the likelihood of incorrect predictions.

The same is true when developing a low variance model with a bigger bias. The model will not fully fit the data set, even though it will lower the probability of erroneous predictions. As a result, there is a delicate balance between biases and variance.



When to use bias-variance decomposition

Since bias and variance are connected to underfitting and overfitting, decomposing the loss into bias and variance helps us understand learning algorithms. Let's understand certain attributes.

- **Low Bias:** Tends to suggest fewer implications about the target function's shape.
- **High-Bias:** Suggests additional assumptions about the target function's shape.
- **Low Variance:** Suggests minor changes to the target function estimate when the training dataset changes.
- **High Variance:** Suggests that changes to the training dataset cause considerable variations in the target function estimate.

Theoretically, a model should have low bias and low variance but this is impossible to achieve. So, an optimal bias and variance are acceptable. Linear models have low variance but high bias and non-linear models have low bias but high variance.

How does this work?

The total error of a machine learning algorithm has three components: bias, variance and noise. So decomposition is the process of derivation

		of total error in this case we are taking Mean Squared Error (MSE). Total error = Bias2 + Variance + Noise		
4	a	**List out various common regression algorithms explain it.** There are several common regression algorithms used in machine learning and statistics. Here are some of the most popular ones: **Linear Regression**: A simple and widely used regression method that models the relationship between the dependent variable and one or more independent variables as a linear equation. **Ridge Regression**: A linear regression model that incorporates L2 regularization, which penalizes large coefficients, helping to reduce overfitting. **Lasso Regression**: Similar to Ridge Regression but uses L1 regularization, which tends to produce sparse coefficients (some coefficients are exactly zero), leading to feature selection. **ElasticNet Regression**: Combines L1 and L2 regularization to balance between Ridge and Lasso regression, providing a more flexible penalty term. **Decision Tree Regression**: Uses a decision tree to model the relationship between features and target variables by recursively partitioning the data into subsets based on feature values. **Random Forest Regression**: An ensemble method that uses multiple decision trees to improve prediction accuracy and reduce overfitting. **Gradient Boosting Regression**: Another ensemble method that builds models sequentially, each new model correcting errors made by the previous ones, leading to high prediction accuracy. **Support Vector Regression (SVR)**: An extension of Support Vector Machines (SVM) for regression, which uses the concept of maximizing the margin of the hyperplane to minimize the error. **K-Nearest Neighbors (KNN) Regression**: A simple and intuitive algorithm that predicts the value of a data point by averaging the values of its k nearest neighbors. These are just a few examples, and there are many other regression algorithms and variations, each with its strengths and weaknesses, suitable for different types of datasets and problems	**[L2][CO2]**	**[6M]**
	b	**Analyze Bayesian Linear Regression with simple example.** **THE BASICS OF BAYSICAN LINEAR REGRESSION:** Bayesian linear regression is a probabilistic approach to linear regression, where we treat model parameters as random variables with prior distributions. Unlike classical linear regression, which gives point estimates for the coefficients, Bayesian linear regression provides a full posterior distribution over the parameters, taking into account both the data and prior beliefs about the parameters.	**[L4][CO2]**	**[6M]**

Here's a brief overview of how Bayesian linear regression works:

1. **Model**: The model assumes that the target variable y is linearly related to the features X, with some Gaussian noise:
$y = X\beta + \epsilon$
where ϵ is a Gaussian noise term with mean 0 and variance σ^2.
2. **Prior**: We specify a prior distribution for the regression coefficients β. A common choice is a Gaussian prior:
$\beta \sim N(0, \Sigma_0)$
where Σ_0 is the covariance matrix of the prior.
3. **Likelihood**: Assuming the noise term ϵ is Gaussian, the likelihood of the data given the parameters is:
$p(y|X, \beta, \sigma^2) = N(y|X\beta, \sigma^2 I)$
where I is the identity matrix.
4. **Posterior**: Using Bayes' theorem, we can compute the posterior distribution of the parameters given the data:
$p(\beta|X, y, \sigma^2) \propto p(y|X, \beta, \sigma^2)p(\beta)$
This posterior distribution captures our updated beliefs about the parameters after seeing the data.

Prediction: To make predictions for new data points, we can use the posterior predictive distribution, which integrates over the uncertainty in the parameters:
$p(y_{\text{new}}|X_{\text{new}}, X, y) = \int p(y_{\text{new}}|X_{\text{new}}, \beta)p(\beta|X, y)d\beta$

In practice, the posterior distribution is often approximated using techniques like Markov chain Monte Carlo (MCMC) or variational inference. Bayesian linear regression allows us to not only make predictions but also quantify our uncertainty about those predictions, which can be useful in many applications.

Bayesian linear regression is a statistical technique used for modelling the relationship between a dependent variable and one or more independent variables. It extends the traditional linear regression by incorporating Bayesian inference methods.

In Bayesian linear regression, we assume a prior distribution for the regression coefficients and update this prior using the observed data to obtain the posterior distribution. This allows us to not only estimate the coefficients but also quantify the uncertainty associated with them.

Basics of Baysiana Linear regression are Probability Definitiions and Baysian statistics

The key steps in Bayesian linear regression are:

- Specify the model: Define the likelihood function, which describes the relationship between the dependent and independent variables, and specify a prior distribution for the regression coefficients.

- Compute the posterior distribution: Use Bayes' theorem to update the prior distribution using the data, obtaining the posterior distribution of the regression coefficients.
- Predict outcomes: Use the posterior distribution to make predictions for new data points, incorporating uncertainty in the predictions.
- Evaluate the model: Assess the model's fit and predictive performance using appropriate metrics.

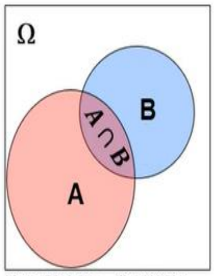
Product Rule:

$$P(B \cap A) = P(A|B)\,P(B)$$

$$P(A \cap B) = P(B|A)\,P(A)$$

It follows that:

$$P(B \cap A) = P(A \cap B)$$



Venn Diagram – illustrating intersection.

Therefore we combine two product rules, substitute:

$$P(A|B)\,P(B) = P(B|A)\,P(A)$$

We get Bayes' Theorem!

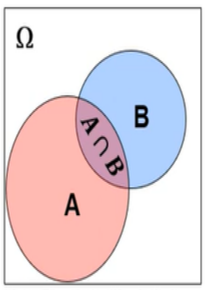
Bayesian Statistical Approaches:

- probabilities based on a degree of belief in an event
- updated as new information is available
- solve probability problems that we cannot use simple frequencies

Bayes' Theorem:

Make an easy adjustment and we get the popular form.

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$



Venn Diagram – illustrating intersection.

Observations:

1. We can get $P(A \mid B)$ from $P(B \mid A)$ as you will see this often comes in handy.
2. Each term is known as:

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

3. Prior should have no information from likelihood.
4. Evidence term is usually just a standardization to ensure closure.

Note: we got to Bayes' Theorem by fundamental frequentist approaches.

Bayes Theorem:

A common example of Bayes' Theorem for modeling the subsurface.

Model Updating with a New Data Source:

Posterior **Likelihood** **Prior**

$$P(\text{Model} \mid \text{New Data}) = \frac{P(\text{New Data} \mid \text{Model})\, P(\text{Model})}{P(\text{New Data})}$$

Evidence

Bayesian Model for Linear Regression:

We formulate a linear regression model, considering probability distributions instead of just the training data.

Response Feature Distribution

Conditional Expectation

$$y \sim N(\beta^T X, \sigma^2 I)$$

Conditional is Gaussian Distributed

Homoscedastic Variance

Now the response and model parameters are all random variables / distributions. We can also include the uncertainty term, σ^2, variance.

Bayesian Model for Linear Regression:

We formulate a linear regression model, considering probability distributions and prior before seeing the data instead of just the training data.

Posterior of Model Parameters

Likelihood of the Response Features Given the Model and Predictors Features.

Prior probability of the Model Parameters

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

Normalizing Constant Independent of the Model

Recall:

$$Posterior = \frac{Likelihood \cdot Prior}{Evidence}$$

$$P(\beta|y,X) = \frac{P(y,X|\beta)P(\beta)}{P(y,X)}$$

Likelihood, $P(y,X|\beta)$:

Conditional distribution of the response and predictor features given the model
- Data-driven
- As the number of sample data increases the likelihood overwhelms the prior distribution

Posterior, $P(y|\beta,X)$:

Conditional distribution of the model parameters given the response and predictor features
- Based on the data-driven likelihood and prior based on expert knowledge and belief
- As the number of data, $n \to \infty$, the model parameters, β, converge to ordinary least squares linear regression.

Solving for the posterior distribution

- Generally intractable for continuous features
- Requires a sampling approach, e.g. Markov chain Monte Carlo (McMC)

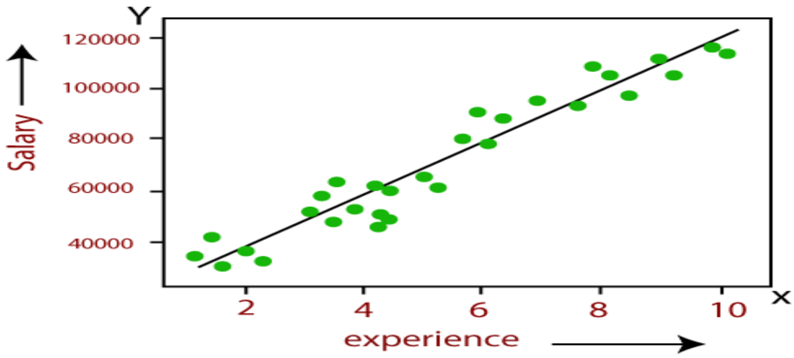
Bayesian linear regression has several advantages, including the ability to incorporate prior knowledge, handle small sample sizes, and provide uncertainty estimates for the coefficients and predictions. However, it can be computationally intensive, especially for models with many parameters.

5	**Summarize the following models.** **(i) Linear regression** **(ii) Logistic regression**	**[L2][CO1]**	**[12M]**

Linear Regression:

- Linear Regression is one of the most simple Machine learning algorithm that comes under Supervised Learning technique and used for solving regression problems.
- It is used for predicting the continuous dependent variable with the help of independent variables.
- The goal of the Linear regression is to find the best fit line that can accurately predict the output for the continuous dependent variable.
- If single independent variable is used for prediction then it is called Simple Linear Regression and if there are more than two independent variables then such regression is called as Multiple Linear Regression.

- By finding the best fit line, algorithm establish the relationship between dependent variable and independent variable. And the relationship should be of linear nature.
- The output for Linear regression should only be the continuous values such as price, age, salary, etc. The relationship between the dependent variable and independent variable can be shown in below image:



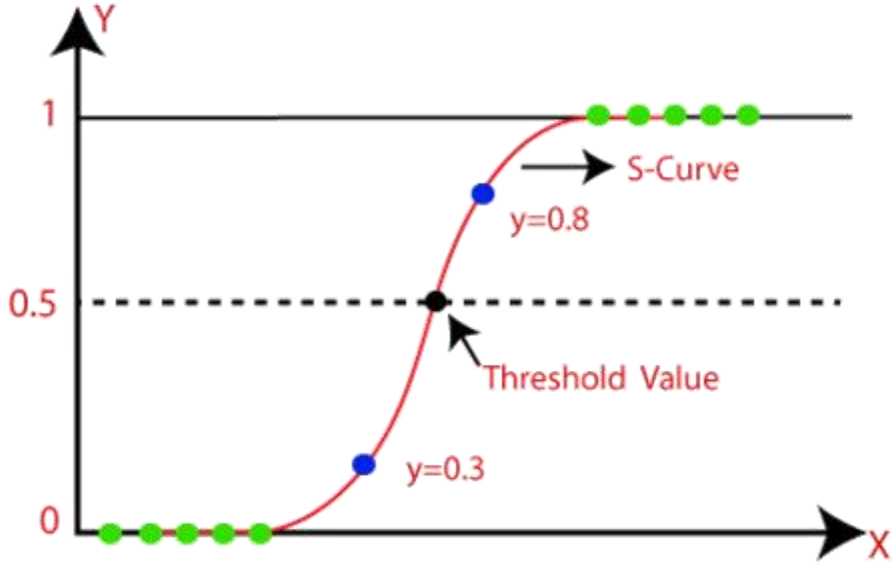
In above image the dependent variable is on Y-axis (salary) and independent variable is on x-axis(experience). The regression line can be written as:

$y= a_0+a_1x+ \varepsilon$

Where, a_0 and a_1 are the coefficients and ε is the error term.

Logistic Regression:

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



- The equation for logistic regression is:

$$\log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

6	a	**Organize how to Tackle Over fitting and Under fitting.**	[L4][CO3]	[6M]

Underfitting: A statistical model or a machine learning algorithm is said to have underfitting when it cannot capture the underlying trend of the data, i.e., it only performs well on training data but performs poorly on testing data. *(It's just like trying to fit undersized pants!)* Underfitting destroys the accuracy of our machine learning model. Its occurrence simply means that our model or the algorithm does not fit the data well enough. It usually happens when we have fewer data to build an accurate model and also when we try to build a linear model with fewer non-linear data. In such cases, the rules of the machine learning model are too easy and flexible to be applied to such minimal data and therefore the model will probably make a lot of wrong predictions. Underfitting can be avoided by using more data and also reducing the features by feature selection.

In a nutshell, Underfitting refers to a model that can neither performs well on the training data nor generalize to new data.

Reasons for Underfitting:

- High bias and low variance
- The size of the training dataset used is not enough.
- The model is too simple.
- Training data is not cleaned and also contains noise in it.

Techniques to reduce underfitting:

- Increase model complexity
- Increase the number of features, performing feature engineering
- Remove noise from the data.
- Increase the number of epochs or increase the duration of training to get better results.

Overfitting: A statistical model is said to be overfitted when the model does not make accurate predictions on testing data. When a model gets trained with so much data, it starts learning from the noise and inaccurate data entries in our data set. And when testing with test data results in High variance. Then the model does not categorize the data correctly, because of too many details and noise.

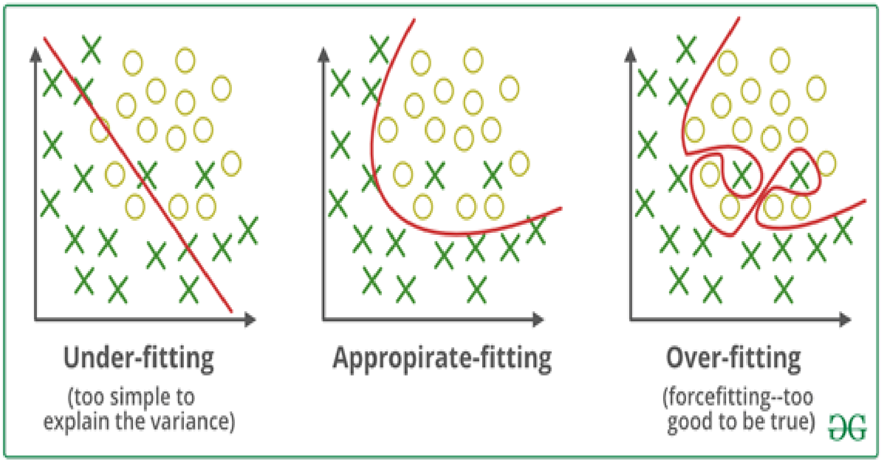
The causes of overfitting are the non-parametric and non-linear methods because these types of machine learning algorithms have more freedom in building the model based on the dataset and therefore they can really build unrealistic models. A solution to avoid overfitting is using a linear algorithm if we have linear data or using the parameters like the maximal depth if we are using decision trees.

In a nutshell, Overfitting is a problem where the evaluation of machine learning algorithms on training data is different from unseen data.

Reasons for Overfitting are as follows:

1. High variance and low bias
2. The model is too complex
3. The size of the training data

Examples:



Techniques to reduce overfitting:

- Increase training data.
- Reduce model complexity.
- Early stopping during the training phase (have an eye over the loss over the training period as soon as loss begins to

increase stop training).

- Ridge Regularization and Lasso Regularization Use dropout for neural networks to tackle overfitting

b **Compare Linear Regression and logistic regression in machine learning.** [L2][CO2] [6M]

Sl.No.	Linear Regression	Logistic Regression
1.	Linear Regression is a supervised regression model.	Logistic Regression is a supervised classification model.
2.	Equation of linear regression: **y = a0 + a1x1 + a2x2 + … + aixi** Here, **y** = response variable **xi** = ith predictor variable **ai** = average effect on y as xi increases by 1	Equation of logistic regression **y(x) = e(a0 + a1x1 + a2x2 + … + aixi) / (1 + e(a0 + a1x1 + a2x2 + … + aixi))** Here, **y** = response variable **xi** = ith predictor variable **ai** = average effect on y as xi increases by 1
3.	In Linear Regression, we predict the value by an integer number.	In Logistic Regression, we predict the value by 1 or 0.
4.	Here no activation function is used.	Here activation function is used to convert a linear regression equation to the logistic regression equation
5.	Here no threshold value is needed.	Here a threshold value is added.
6.	Here we calculate Root Mean Square Error(RMSE) to predict the next weight value.	Here we use precision to predict the next weight value.
7.	Here dependent variable should be numeric and the response variable is continuous to value.	Here the dependent variable consists of only two categories. Logistic regression estimates the odds outcome of the dependent variable given a set of quantitative or categorical independent variables.

8.	It is based on the least square estimation.	It is based on maximum likelihood estimation.		
9.	Here when we plot the training datasets, a straight line can be drawn that touches maximum plots.	Any change in the coefficient leads to a change in both the direction and the steepness of the logistic function. It means positive slopes result in an S-shaped curve and negative slopes result in a Z-shaped curve.		
10.	Linear regression is used to estimate the dependent variable in case of a change in independent variables. For example, predict the price of houses.	Whereas logistic regression is used to calculate the probability of an event. For example, classify if tissue is benign or malignant.		
11.	Linear regression assumes the normal or gaussian distribution of the dependent variable.	Logistic regression assumes the binomial distribution of the dependent variable. \		
12.	Applications of linear regression: • Financial risk assessment • Business insights • Market analysis	Applications of logistic regression: • Medicine • Credit scoring • Hotel Booking • Gaming • Text editing		

7	a	**Illustrate Multiple Linear regression in supervised learning.**	**[L3][CO3]**	**[6M]**

Multiple Linear Regression – Solved Example

- In linear regression model we have one dependent and one independent variable.
- Multiple regression model involves multiple predictors or independent variables and one dependent variable.
- This is an extension of the linear regression problem.

- The multiple regression of two variables x1 and x2 is given as follows:

$$y = f(x_1, x_2)$$

$$y = a_0 + a_1x_1 + a_2x_2$$

- In general, this is given for 'n' independent variables as:

$$y = f(x_1, x_2, \ldots . x_n)$$

$$y = a_0 + a_1x_1 + a_2x_2 + \ldots \ldots + a_nx_n + \varepsilon$$

- Here, $x_1, x_2, \ldots . x_n$ are predictor variables, y is the dependent variable, $(a_0, a_1, a_2, \ldots . a_n)$ are the coefficients of the regression equation and ε is the error term.

Play (k)

- Apply multiple regression for the values given in Table where weekly sales along with sales for products x1 and x2 are provided.
- Use matrix approach for finding multiple regression.

x1 Product 1 Sales	x2 Product 2 Sales	Y Weekly Sales
1	4	1
2	5	6
3	8	8
4	2	12

- Here, the matrices for Y and X are given as follows:

$$X = \begin{pmatrix} 1 & 1 & 4 \\ 1 & 2 & 5 \\ 1 & 3 & 8 \\ 1 & 4 & 2 \end{pmatrix} \text{ and } Y = \begin{pmatrix} 1 \\ 6 \\ 8 \\ 12 \end{pmatrix}$$

- The coefficient of the multiple regression equation is given as

$$a = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix}.$$

x1 Product 1 Sales	x2 Product 2 Sales	Y Weekly Sales
1	4	1
2	5	6
3	8	8
4	2	12

- The regression coefficient for multiple regression is calculated the same way as linear regression:

$$\hat{a} = ((X^T X)^{-1} X^T)Y$$

x1 Product 1 Sales	x2 Product 2 Sales	Y Weekly Sales
1	4	1
2	5	6
3	8	8
4	2	12

- The regression coefficient for multiple regression is calculated the same way as linear regression:

$$\hat{a} = ((X^T X)^{-1} X^T)Y$$

$$X^T X = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 4 & 5 & 8 & 2 \end{pmatrix} \begin{pmatrix} 1 & 1 & 4 \\ 1 & 2 & 5 \\ 1 & 3 & 8 \\ 1 & 4 & 2 \end{pmatrix} = \begin{pmatrix} 4 & 10 & 19 \\ 10 & 30 & 46 \\ 19 & 46 & 109 \end{pmatrix}$$

x1 Product 1 Sales	x2 Product 2 Sales	Y Weekly Sales
1	4	1
2	5	6
3	8	8
4	2	12

The regression coefficient for multiple regression is calculated the same way as linear regression:

x1 Product 1 Sales	x2 Product 2 Sales	Y Weekly Sales
1	4	1
2	5	6
3	8	8
4	2	12

$$\hat{a} = ((X^T X)^{-1} X^T) Y$$

$$(X^T X)^{-1} = \begin{pmatrix} 4 & 10 & 19 \\ 10 & 30 & 46 \\ 19 & 46 & 109 \end{pmatrix}^{-1} = \begin{pmatrix} 3.15 & -0.59 & -0.30 \\ -0.59 & 0.20 & 0.016 \\ -0.30 & 0.016 & 0.054 \end{pmatrix}$$

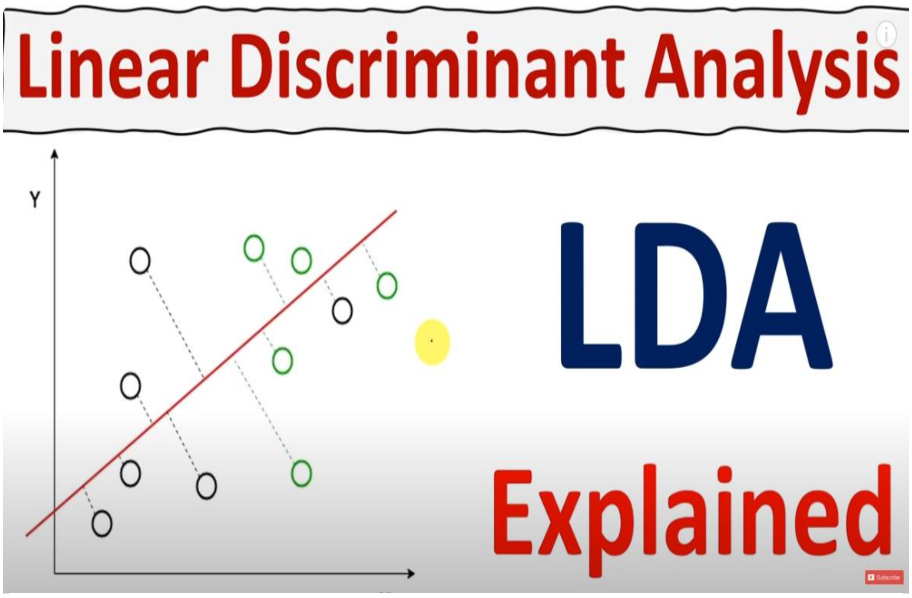
$$X^T X)^{-1} X^T = \begin{pmatrix} 3.15 & -0.59 & -0.30 \\ -0.59 & 0.20 & 0.016 \\ -0.30 & 0.016 & 0.054 \end{pmatrix} \times \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 4 & 5 & 8 & 2 \end{pmatrix} = \begin{pmatrix} 0.05 & 0.47 & -1.02 & 0.19 \\ -0.32 & -0.098 & 0.155 & 0.26 \\ -0.065 & 0.005 & 0.185 & -0.125 \end{pmatrix}$$

$$\hat{a} = ((X^T X)^{-1} X^T) Y = \begin{pmatrix} 0.05 & 0.47 & -1.02 & 0.19 \\ -0.32 & -0.098 & 0.155 & 0.26 \\ -0.065 & 0.005 & 0.185 & -0.125 \end{pmatrix} \times \begin{pmatrix} 1 \\ 6 \\ 8 \\ 12 \end{pmatrix} = \begin{pmatrix} -1.69 \\ 3.48 \\ -0.05 \end{pmatrix}$$

$$a_0 = -1.69$$
$$a_1 = 3.48$$
$$a_2 = -0.05$$

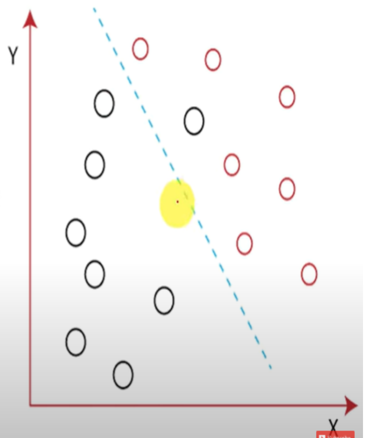
- $y = a_0 + a_1 x_1 + a_2 x_2$
- Hence, the constructed model is:
- $y = -1.69 + 3.48 x_1 - 0.05 x_2$

	b	**Explain about Linear Discriminant analysis**	**[L2][CO]2**	**[6M]**

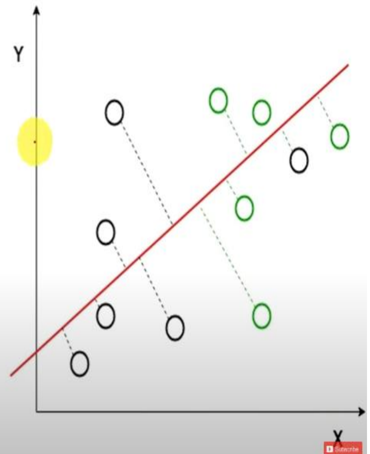


- Linear Discriminant Analysis or Normal Discriminant Analysis or Discriminant Function Analysis is a dimensionality reduction technique that is commonly used for supervised classification problems.
- It is used to project the features in higher dimension space into a lower dimension space.

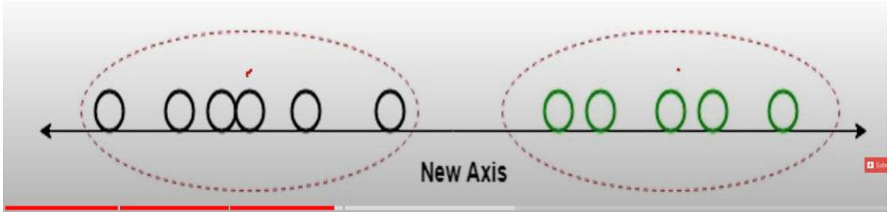
- Suppose we have two sets of data points belonging to two different classes that we want to classify.
- When the data points are plotted on the 2D plane, there's no straight line that can separate the two classes of the data points completely.



- Here, Linear Discriminant Analysis uses both the axes (X and Y) to create a new axis and projects data onto a new axis in a way to maximize the separation of the two categories and hence, reducing the 2D graph into a 1D graph.



- Two criteria are used by LDA to create a new axis:
  - –Maximize the distance between means of the two classes.
  - –Minimize the variation within each class.



1. Compute the class means of dependent variable $\mu_1 = \frac{1}{N_1}\sum_{x\in\omega_1} x$
2. Derive the covariance matrix of the class variable $S_1 = \sum_{x\in\omega_1}(x-\mu_1)(x-\mu_1)^T$
3. Compute the within class – scatter matrix (S1+S2) $S_w = S_1 + S_2$
4. Compute the between class scatter matrix $S_B = (\mu_1-\mu_2)(\mu_1-\mu_2)^T$
5. Compute the Eigen values and eigen vectors from the within class and between class scatter matrix

$$S_W^{-1} S_B w = \lambda w$$

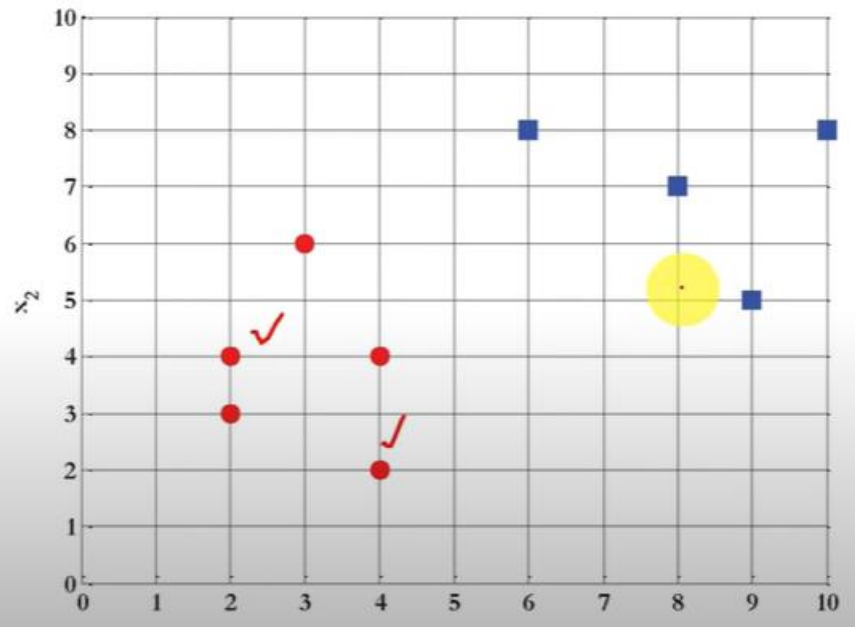
6. Sort the values of eigen values and select the top k values
7. Find the eigen vectors corresponds to the top k eigen vectors
8. Obtain the LDA by taking the dot product of eigen vectors and original data

$$\left(S_W^{-1} S_B - \lambda I\right)\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = 0$$

- Compute the Linear Discriminant projection for the following two dimensional dataset.

Samples for class ω_1 : $\mathbf{X}_1=(x_1,x_2)=\{(4,2),(2,4),(2,3),(3,6),(4,4)\}$

Sample for class ω_2 : $\mathbf{X}_2=(x_1,x_2)=\{(9,10),(6,8),(9,5),(8,7),(10,8)\}$



this is 2D

The classes mean are :

$$\mu_1 = \frac{1}{N_1}\sum_{x\in\omega_1} x = \frac{1}{5}\left[\begin{pmatrix}4\\2\end{pmatrix}+\begin{pmatrix}2\\4\end{pmatrix}+\begin{pmatrix}2\\3\end{pmatrix}+\begin{pmatrix}3\\6\end{pmatrix}+\begin{pmatrix}4\\4\end{pmatrix}\right]=\begin{pmatrix}3\\3.8\end{pmatrix}$$

$$\mu_2 = \frac{1}{N_2}\sum_{x\in\omega_2} x = \frac{1}{5}\left[\begin{pmatrix}9\\10\end{pmatrix}+\begin{pmatrix}6\\8\end{pmatrix}+\begin{pmatrix}9\\5\end{pmatrix}+\begin{pmatrix}8\\7\end{pmatrix}+\begin{pmatrix}10\\8\end{pmatrix}\right]=\begin{pmatrix}8.4\\7.6\end{pmatrix}$$

Covariance matrix of the first class:

$$S_1 = \frac{\sum_{x\in\omega_1}(x-\mu_1)(x-\mu_1)^T}{\text{N-1}} = \left[\begin{pmatrix}4\\2\end{pmatrix}-\begin{pmatrix}3\\3.8\end{pmatrix}\right]\left[\begin{pmatrix}4\\2\end{pmatrix}-\begin{pmatrix}3\\3.8\end{pmatrix}\right]^T + \left[\begin{pmatrix}2\\4\end{pmatrix}-\begin{pmatrix}3\\3.8\end{pmatrix}\right]\left[\begin{pmatrix}2\\4\end{pmatrix}-\begin{pmatrix}3\\3.8\end{pmatrix}\right]^T$$

$$+\left[\begin{pmatrix}2\\3\end{pmatrix}-\begin{pmatrix}3\\3.8\end{pmatrix}\right]\left[\begin{pmatrix}2\\3\end{pmatrix}-\begin{pmatrix}3\\3.8\end{pmatrix}\right]^T + \left[\begin{pmatrix}3\\6\end{pmatrix}-\begin{pmatrix}3\\3.8\end{pmatrix}\right]\left[\begin{pmatrix}3\\6\end{pmatrix}-\begin{pmatrix}3\\3.8\end{pmatrix}\right]^T + \left[\begin{pmatrix}4\\4\end{pmatrix}-\begin{pmatrix}3\\3.8\end{pmatrix}\right]\left[\begin{pmatrix}4\\4\end{pmatrix}-\begin{pmatrix}3\\3.8\end{pmatrix}\right]^T \Big/ \text{N-1}$$

$$= \begin{pmatrix}1 & -0.25\\ -0.25 & 2.2\end{pmatrix}$$

$$\begin{bmatrix}1\\-1.8\end{bmatrix}\begin{bmatrix}1 & -1.8\end{bmatrix} \Rightarrow \begin{bmatrix}1 & -1.8\\ -1.8 & 3.24\end{bmatrix}$$

Covariance matrix of the second class:

$$S_2 = \frac{\sum_{x\in\omega_2}(x-\mu_2)(x-\mu_2)^T}{\text{N-1}} = \left[\begin{pmatrix}9\\10\end{pmatrix}-\begin{pmatrix}8.4\\7.6\end{pmatrix}\right]\left[\begin{pmatrix}9\\10\end{pmatrix}-\begin{pmatrix}8.4\\7.6\end{pmatrix}\right]^T + \left[\begin{pmatrix}6\\8\end{pmatrix}-\begin{pmatrix}8.4\\7.6\end{pmatrix}\right]\left[\begin{pmatrix}6\\8\end{pmatrix}-\begin{pmatrix}8.4\\7.6\end{pmatrix}\right]^T$$

$$+\left[\begin{pmatrix}9\\5\end{pmatrix}-\begin{pmatrix}8.4\\7.6\end{pmatrix}\right]\left[\begin{pmatrix}9\\5\end{pmatrix}-\begin{pmatrix}8.4\\7.6\end{pmatrix}\right]^T + \left[\begin{pmatrix}8\\7\end{pmatrix}-\begin{pmatrix}8.4\\7.6\end{pmatrix}\right]\left[\begin{pmatrix}8\\7\end{pmatrix}-\begin{pmatrix}8.4\\7.6\end{pmatrix}\right]^T + \left[\begin{pmatrix}10\\8\end{pmatrix}-\begin{pmatrix}8.4\\7.6\end{pmatrix}\right]\left[\begin{pmatrix}10\\8\end{pmatrix}-\begin{pmatrix}8.4\\7.6\end{pmatrix}\right]^T \Big/ \text{N-1}$$

$$= \begin{pmatrix}2.3 & -0.05\\ -0.05 & 3.3\end{pmatrix} \checkmark$$

Within-class scatter matrix:

$$S_w = S_1 + S_2 = \begin{pmatrix}1 & -0.25\\ -0.25 & 2.2\end{pmatrix} + \begin{pmatrix}2.3 & -0.05\\ -0.05 & 3.3\end{pmatrix}$$

$$= \begin{pmatrix}3.3 & -0.3\\ -0.3 & 5.5\end{pmatrix}$$

$$S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$$

$$= \left[\begin{pmatrix} 3 \\ 3.8 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]\left[\begin{pmatrix} 3 \\ 3.8 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]^T$$

$$= \begin{pmatrix} -5.4 \\ -3.8 \end{pmatrix}\begin{pmatrix} -5.4 & -3.8 \end{pmatrix}$$

$$= \begin{pmatrix} 29.16 & 20.52 \\ 20.52 & 14.44 \end{pmatrix}$$

- Find Eigen Values

$$S_W^{-1} S_B w = \lambda w$$

$$\Rightarrow \left| S_W^{-1} S_B - \lambda I \right| = 0$$

$$\Rightarrow \left| \begin{pmatrix} 3.3 & -0.3 \\ -0.3 & 5.5 \end{pmatrix}^{-1} \begin{pmatrix} 29.16 & 20.52 \\ 20.52 & 14.44 \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right| = 0$$

$$\Rightarrow \left| \begin{pmatrix} 0.3045 & 0.0166 \\ 0.0166 & 0.1827 \end{pmatrix} \begin{pmatrix} 29.16 & 20.52 \\ 20.52 & 14.44 \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right| = 0$$

$$\Rightarrow \left| \begin{pmatrix} 9.2213 - \lambda & 6.489 \\ 4.2339 & 2.9794 - \lambda \end{pmatrix} \right|$$

$$= (9.2213 - \lambda)(2.9794 - \lambda) - 6.489 \times 4.2339 = 0$$

$$\Rightarrow \lambda^2 - 12.2007\lambda = 0 \Rightarrow \lambda(\lambda - 12.2007) = 0$$

$$\Rightarrow \lambda_1 = 0, \lambda_2 = 12.2007$$

- Find Eigen Vector

$$\left(S_W^{-1} S_B - \lambda I \right) \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = 0$$

$$w_1 = \begin{pmatrix} -0.5755 \\ 0.8178 \end{pmatrix} \qquad w_2 = \begin{pmatrix} 0.9088 \\ 0.4173 \end{pmatrix} = w^*$$

Or directly;

$$w^* = S_W^{-1}(\mu_1 - \mu_2) = \begin{pmatrix} 3.3 & -0.3 \\ -0.3 & 5.5 \end{pmatrix}^{-1} \left[\begin{pmatrix} 3 \\ 3.8 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix} \right]$$

$$= \begin{pmatrix} 0.3045 & 0.0166 \\ 0.0166 & 0.1827 \end{pmatrix} \begin{pmatrix} -5.4 \\ -3.8 \end{pmatrix}$$

$$= \begin{pmatrix} 0.9088 \\ 0.4173 \end{pmatrix}$$

- Obtain the LDA by taking the dot product of eigen vectors and original data

$[1, 2]$ $\begin{bmatrix} 0.9 \\ 0.4 \end{bmatrix}$ 2x1

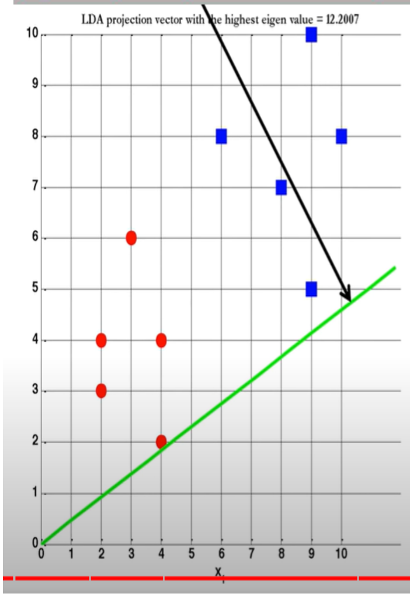
1x2 1x1

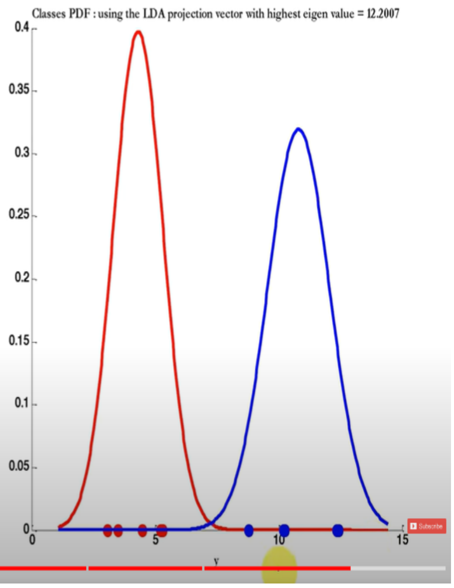
$$w_2 = \begin{pmatrix} 0.9088 \\ 0.4173 \end{pmatrix} = w^*$$

Samples for class ω_1 : $\mathbf{X}_1 = (x_1, x_2) = \{(4,2),(2,4),(2,3),(3,6),(4,4)\}$

Sample for class ω_2 : $\mathbf{X}_2 = (x_1, x_2) = \{(9,10),(6,8),(9,5),(8,7),(10,8)\}$

X1	4	2	2	3	4	9	6	9	8	10
X2	2	4	3	6	4	10	8	5	7	8
1st LD	4.46	3.48	3.06	5.2	5.3	12.35	8.8	10.2	10.19	12.42





8	**Discuss Simple Linear, polynomial Regression and regualization techniques in supervised learning.**	[L2][CO3]	[12M]

Simple Linear Regression:

Simple Linear Regression is a type of Regression algorithms that models the relationship between a dependent variable and a single independent variable. The relationship shown by a Simple Linear Regression model is linear or a sloped straight line, hence it is called Simple Linear Regression.

The key point in Simple Linear Regression is that the ***dependent variable must be a continuous/real value***. However, the independent variable can be measured on continuous or categorical values.

imple Linear regression algorithm has mainly two objectives:

- **Model the relationship between the two variables.** Such as the relationship between Income and expenditure, experience and Salary, etc.
- **Forecasting new observations.** Such as Weather forecasting according to temperature, Revenue of a company according to the investments in a year, etc.

he Simple Linear Regression model can be represented using the below equati

$$y= a_0+a_1x+ \varepsilon$$

Where,

a0= It is the intercept of the Regression line (can be obtained putting x=0)
a1= It is the slope of the regression line, which tells whether the line is increasing or decreasing.
ε = The error term. (For a good model it will be negligible)

Implementation of Simple Linear Regression Algorithm using Python

Problem Statement example for Simple Linear Regression:

Here we are taking a dataset that has two variables: salary (dependent variable) and experience (Independent variable). The goals of this problem is:

- **We want to find out if there is any correlation between these two variables**
- **We will find the best fit line for the dataset.**
- **How the dependent variable is changing by changing the independent variable.**

	3. Polynomial regression • Assume that there is only one independent variable x. If the relationship between independent variables x and dependent or output variable y is modeled by the relation, $$y=a_0+a_1*x+a_2*x^2+.........+a_n*x^n$$ • for some positive integer n>1, then we have a polynomial regression. Play (k) **Regularization techniques:** • Regularization is a technique used to reduce errors by fitting the function appropriately on the given training set and avoiding overfitting. The commonly used regularization techniques are : • Lasso Regularization – L1 Regularization • Ridge Regularization – L2 Regularization • Elastic Net Regularization – L1 and L2 Regularization **Lasso Regression** • A regression model which uses the L1 Regularization technique is called LASSO (Least Absolute Shrinkage and Selection Operator) regression. • Lasso Regression adds the "absolute value of magnitude" of the coefficient as a penalty term to the loss function(L). • Lasso regression also helps us achieve feature selection by penalizing the weights to approximately equal to zero if that feature does not serve any purpose in the model.		

Lasso Regression

$$\text{Cost} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\sum_{i=1}^{m}|w_i|$$

- *where,*
- m – *Number of Features*
- n – *Number of Examples*
- y_i – *Actual Target Value*

Ridge Regression

- A regression model that uses the **L2 regularization** technique is called **Ridge regression**.
- **Ridge regression** adds the "*squared magnitude*" of the coefficient as a penalty term to the loss function(L).

$$\text{Cost} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\sum_{i=1}^{m}w_i^2$$

Elastic Net Regression

- This model is a combination of L1 as well as L2 regularization.
- That implies that we add the absolute norm of the weights as well as the squared measure of the weights.
- With the help of an extra hyperparameter that controls the ratio of the L1 and L2 regularization.

$$\text{Cost} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\left((1-\alpha)\sum_{i=1}^{m}|w_i| + \alpha\sum_{i=1}^{m}w_i^2\right)$$

9	**Analyze three linear models for the classification in supervised learning.**	**[L4][CO3]**	**[12M]**

Linear models can also be used for classification tasks, where the goal is to predict the categorical label of an input data point. Some common linear models for classification include:

Logistic Regression: Despite its name, logistic regression is a linear model for binary classification. It models the probability of the binary outcome using a logistic function.

Linear Support Vector Machines (SVM): SVMs can be used for both binary and multiclass classification. In linear SVM, the algorithm finds the hyperplane that best separates the classes in the feature space.

Perceptron: The perceptron is a simple linear classifier used for binary classification. It learns a linear decision boundary separating the classes.
Linear Discriminant Analysis (LDA): LDA is a classification algorithm that models the distribution of the predictors separately in each class and then uses Bayes' theorem to estimate the probability of a data point belonging to each class.
Ridge Classifier: Similar to Ridge Regression, Ridge Classifier uses L2 regularization to penalize large coefficients in a linear model for classification.
ElasticNet Classifier: Similar to ElasticNet Regression, ElasticNet Classifier combines L1 and L2 regularization to balance between sparsity and smoothness in the coefficients.

Linear classifiers are often preferred for their simplicity, interpretability, and efficiency, especially when dealing with high-dimensional data. However, they may struggle with complex, non-linear relationships in the data, where more sophisticated models like decision trees or neural networks may be more appropriate.

10 a **Compare Probabilistic Generative model and Discriminative models.** [L6][CO3] [6M]

Criteria	Generative Models	Discriminative Models
Application (Use cases)	Generate new data instances. e.g.- Predict next word in a sequence.	Discriminate between different kinds of data instances e.g.- Text Classification.
Model Performance	• Measured by likelihood. • Outliers affect the distribution significantly and results in bad model accuracy. • With correct distribution the generative models are more accurate.	• Measured by misclassification rate. • These models are robust against outliers and poor modeling.
Training time	Need less data to train as these models make stronger assumptions of conditional independence	Discriminative models take more time to train on substantial amount of training samples.
Training Data	Generative models can work with missing data and generalize well	Discriminative Models need sufficient training data for better generalization performance.
Intuitive	More elegant by having explanatory power	Relationships between variables are not explicit and visualizable (Black boxes)

b **List out the categorization of probabilistic models explain it.** [L2][CO3] [6M]

PROBABILISTIC MODELS

Probabilistic models are an essential component of machine learning, which aims to learn patterns from data and make predictions on new, unseen data. They are statistical models that capture the inherent uncertainty in data and incorporate it into their predictions. Probabilistic models are used in various applications such as image and speech recognition, natural language processing, and recommendation systems.

A Probabilistic model in machine learning is a mathematical representation of a real-world process that incorporates uncertain or random variables. The goal of probabilistic modeling is to estimate the probabilities of the possible outcomes of a system based on data or prior knowledge.

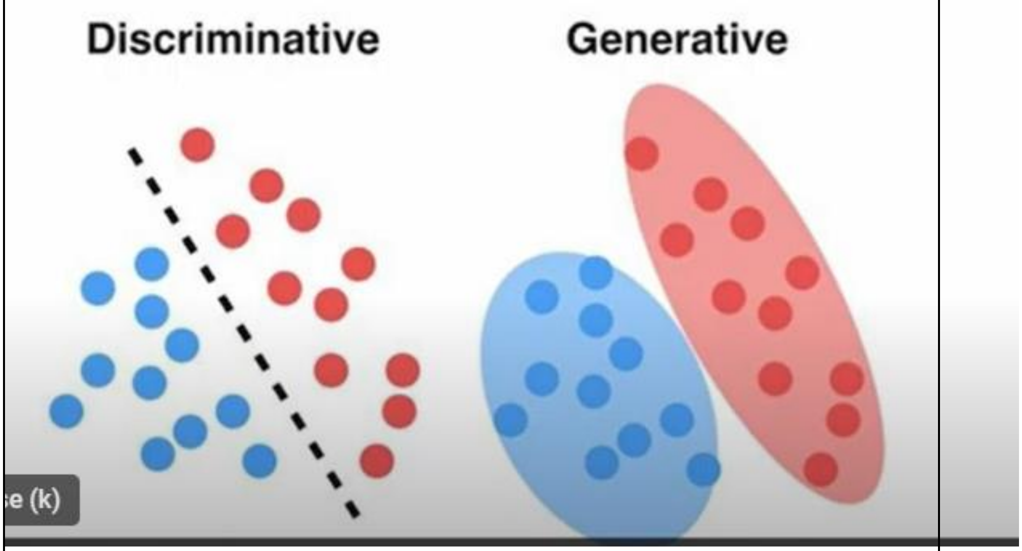
Probabilistic models are used in a variety of machine learning tasks such as classification, regression, clustering, and dimensionality reduction. Some popular probabilistic models include:

- **Gaussian Mixture Models (GMMs)**
- **Hidden Markov Models (HMMs)**
- **Bayesian Networks**
- **Markov Random Fields (MRFs)**

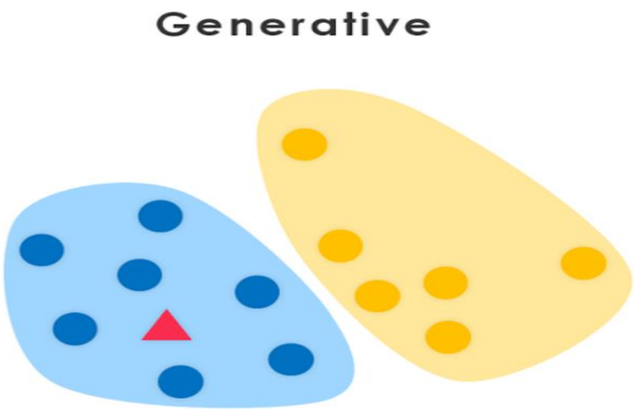
Categories Of Probabilistic Models
These models can be classified into the following categories:

- Generative models
- Discriminative models.
- Graphical models

Decoding Generative and Discriminative Models (analyticsvidhya.com)



PROBABILISTIC GENERATIVE MODELS:
Generative models aim to model the joint distribution of the input and output variables. These models generate new data based on the probability distribution of the original dataset. Generative models are powerful because they can generate new data that resembles the training data. They can be used for tasks such as image and speech synthesis, language translation, and text generation.



The Approach of Generative Models

In the case of generative models, to find the conditional probability **P(Y|X)**, they estimate the priorprobability **P(Y)** and likelihood probability **P(X|Y)** with the help of the training data and use the Bayes Theorem to calculate the posterior probability **P(Y |X):**

$$posterior = \frac{prior \times likelihood}{evidence} \Rightarrow P(Y|X) = \frac{P(Y) \cdot P(X|Y)}{P(X)}$$

Generative models are considered a class of statistical models that can generate new data instances. These models are used in unsupervised machine learning as a means to perform tasks such as:

- Probability and Likelihood estimation,
- Modeling data points
- To describe the phenomenon in data,
- To distinguish between classes based on these probabilities.

Probabilistic generative models are a class of models used in machine learning and statistics to model the joint distribution of observed and latent variables. These models are used for various tasks, including density estimation, clustering, and generating new samples from the learned distribution. Here are some common types of probabilistic generative models:

Gaussian Mixture Models (GMMs): GMMs assume that the data is generated by a mixture of several Gaussian distributions. Each component in the mixture represents a cluster in the data, and the model learns the parameters of these Gaussian distributions.

Hidden Markov Models (HMMs): HMMs are used to model sequential data, where the underlying process is assumed to be a Markov chain with unobserved (hidden) states. HMMs are widely used in speech recognition, natural language processing, and bioinformatics.

Latent Dirichlet Allocation (LDA): LDA is a topic model used for text analysis and other applications where data can be represented as a collection of discrete categories. LDA assumes that each document is a mixture of topics, and each word in the document is generated from one of these topics.

Variational Autoencoders (VAEs): VAEs are a type of autoencoder that learns to generate new data points by modeling the latent variables in the data. VAEs are trained to maximize the evidence lower bound (ELBO), which is a lower bound on the log-likelihood of the data.

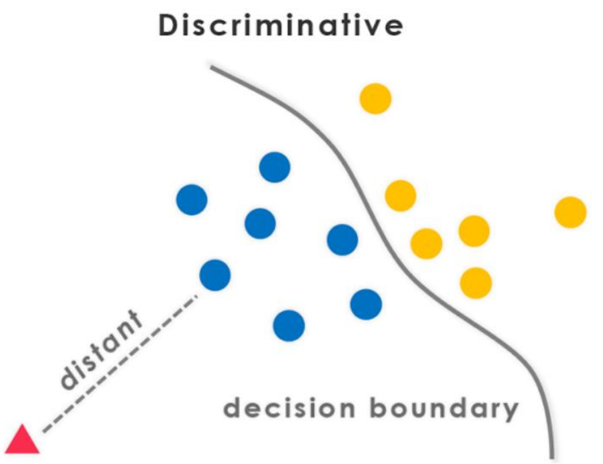
Generative Adversarial Networks (GANs): GANs consist of two neural networks, a generator and a discriminator, that are trained together in a competitive setting. The generator learns to generate realistic samples, while the discriminator learns to distinguish between real and generated samples.

These are just a few examples of probabilistic generative models, and there are many other variants and extensions used in different applications. These models provide a powerful framework for understanding and modeling complex data distributions.

PROBABILISTIC DISCRIMINATIVE MODELS

Discriminative models

The discriminative model aims to model the conditional distribution of the output variable given the input variable. They learn a decision boundary that separates the different classes of the output variable. Discriminative models are useful when the focus is on making accurate predictions rather than generating new data. They can be used for tasks such as image recognition, speech recognition, and sentiment analysis.



The Approach of Discriminative Models

In the case of discriminative models, to find the probability, they directly assume some functional form for **P(Y|X)** andthen estimate the parameters of **P(Y|X)** with the help of the training data.

The Mathematics of Discriminative Models

Training discriminative classifiers or discriminant analysis involves estimating a function **f: X -> Y**, or probability **P(Y|X)**

- Assume some functional form for the probability, such as **P(Y|X)**
- With the help of training data, we estimate the parameters of **P(Y|X)**

Probabilistic discriminative models are a class of models used in machine learning for classification tasks. Unlike generative models, which model the joint distribution of features and labels, discriminative models directly model the conditional distribution of labels given the features. This allows discriminative models to focus on learning the decision boundary between classes, rather than modeling the entire distribution of the data. Here are some common types of probabilistic discriminative models:

Logistic Regression: Logistic regression is a linear model for binary classification that models the probability of the binary outcome using a logistic function. It is a simple and widely used discriminative model.

Support Vector Machines (SVM): SVMs can be used for both binary and multiclass classification. In the context of probabilistic classification, SVMs can output class probabilities using methods such as Platt scaling or by using a modified SVM formulation that directly optimizes for probabilities.

Conditional Random Fields (CRFs): CRFs are a type of probabilistic graphical model used for structured prediction tasks, such as sequence labeling and image segmentation. CRFs model the conditional distribution of labels given the input features and capture dependencies between neighboring labels.

Neural Networks: While neural networks are often used as discriminative models for classification, they can also be used in a probabilistic framework. For example, softmax regression is a probabilistic version of the standard softmax activation function used in neural networks for classification.

Random Forests: Random forests are an ensemble learning method that constructs a multitude of decision trees during training and outputs the class that is the mode of the classes output by individual trees. While not inherently probabilistic, random forests can be combined with techniques like bootstrap aggregating (bagging) to estimate class probabilities.

These models are widely used in various applications due to their flexibility and ability to model complex decision boundaries.

Graphical models

These models use graphical representations to show the conditional dependence between variables. They are commonly used for tasks such as image recognition, natural language processing, and causal inference.

UNIT –III

UNSUPERVISED LEARNING

1		**Analyze the unsupervised learning and its techniques with suitable examples.** **Referred from first and second units**	**[L2][CO3]**	**[12M]**
2	a	**Explain the various Clustering algorithms.**	**[L2][CO3]**	**[8M]**

Types of Clustering

Broadly speaking, there are 2 types of clustering that can be performed to group similar data points:

Hard Clustering: In this type of clustering, each data point belongs to a cluster completely or not. For example, Let's say there are 4 data point and we have to cluster them into 2 clusters. So each data point will either belong to cluster 1 or cluster 2.

Data Points	Clusters
A	C1
B	C2
C	C2
D	C1

Soft Clustering: In this type of clustering, instead of assigning each data point into a separate cluster, a probability or likelihood of that point being that cluster is evaluated. For example, Let's say there are 4 data point and we have to cluster them into 2 clusters. So we will be evaluating a probability of a data point belonging to both clusters. This probability is calculated for all data points.

Types of Clustering Algorithms

Various types of clustering algorithms are:

- Centroid-based Clustering (Partitioning methods)
- Density-based Clustering (Model-based methods)
- Connectivity-based Clustering (Hierarchical clustering)
- Distribution-based Clustering

Types of Clustering Methods

The clustering methods are broadly divided into **Hard clustering** (datapoint belongs to only one group) and **Soft Clustering** (data points can belong to another group also). But there are also other various approaches of Clustering exist. Below are the main clustering methods used in Machine learning:

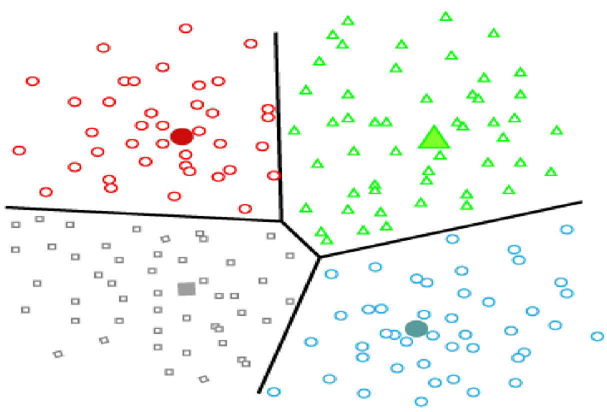
1. **Partitioning Clustering**

2. **Density-Based Clustering**
3. **Distribution Model-Based Clustering**
4. **Hierarchical Clustering**
5. **Fuzzy Clustering**

Partitioning Clustering

It is a type of clustering that divides the data into non-hierarchical groups. It is also known as the **centroid-based method**. The most common example of partitioning clustering is the **K-Means Clustering algorithm**.

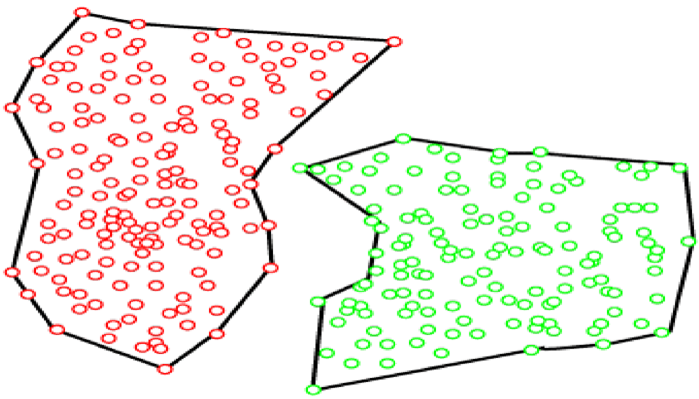
In this type, the dataset is divided into a set of k groups, where K is used to define the number of pre-defined groups. The cluster center is created in such a way that the distance between the data points of one cluster is minimum as compared to another cluster centroid.



Density-Based Clustering

The density-based clustering method connects the highly-dense areas into clusters, and the arbitrarily shaped distributions are formed as long as the dense region can be connected. This algorithm does it by identifying different clusters in the dataset and connects the areas of high densities into clusters. The dense areas in data space are divided from each other by sparser areas.

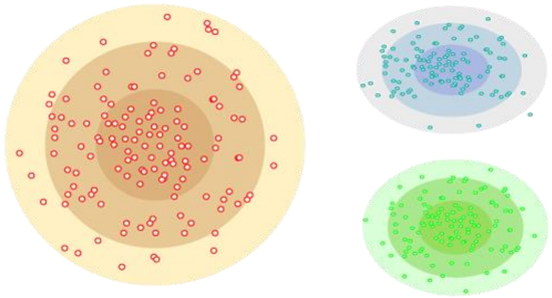
These algorithms can face difficulty in clustering the data points if the dataset has varying densities and high dimensions.



Distribution Model-Based Clustering

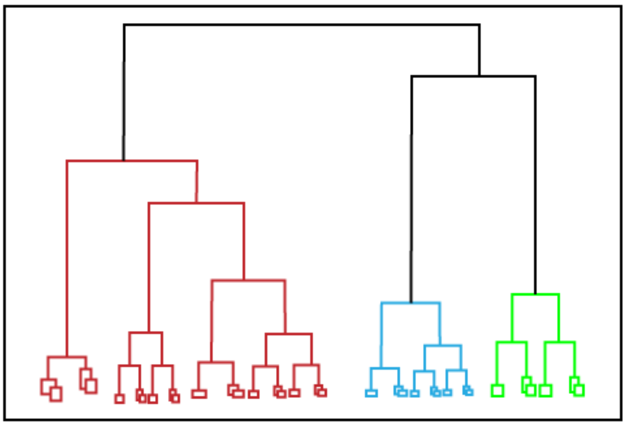
In the distribution model-based clustering method, the data is divided based on the probability of how a dataset belongs to a particular distribution. The grouping is done by assuming some distributions commonly **Gaussian Distribution**.

The example of this type is the **Expectation-Maximization Clustering algorithm** that uses Gaussian Mixture Models (GMM).



Hierarchical Clustering

Hierarchical clustering can be used as an alternative for the partitioned clustering as there is no requirement of pre-specifying the number of clusters to be created. In this technique, the dataset is divided into clusters to create a tree-like structure, which is also called a **dendrogram**. The observations or any number of clusters can be selected by cutting the tree at the correct level. The most common example of this method is the **Agglomerative Hierarchical algorithm**.



Fuzzy Clustering

Fuzzy clustering is a type of soft method in which a data object may belong to more than one group or cluster. Each dataset has a set of membership coefficients, which depend on the degree of membership to be in a cluster. **Fuzzy C-means algorithm** is the example of this type of clustering; it is sometimes also known as the Fuzzy k-means algorithm.

Here we are discussing mainly popular Clustering algorithms that are widely used in machine learning:

K-Means algorithm: The k-means algorithm is one of the most popular clustering algorithms. It classifies the dataset by dividing the samples into different clusters of equal variances. The number of clusters must be specified in this algorithm. It is fast with fewer computations required, with the linear complexity of **O(n).**

Mean-shift algorithm: Mean-shift algorithm tries to find the dense areas in the smooth density of data points. It is an example of a centroid-based model, that works on updating the candidates for centroid to be the center of the points within a given region.

DBSCAN Algorithm: It stands **for Density-Based Spatial Clustering of Applications with Noise**. It is an example of a density-based model similar to the mean-shift, but with some remarkable advantages. In this algorithm, the areas of high density are separated by the areas of low density. Because of this, the clusters can be found in any arbitrary shape.

Expectation-Maximization Clustering using GMM: This algorithm can be used as an alternative for the k-means algorithm or for those cases where K-means can be failed. In GMM, it is assumed that the data points are Gaussian distributed.

1. **Agglomerative Hierarchical algorithm:** The Agglomerative hierarchical algorithm performs the bottom-up hierarchical clustering. In this, each data point is treated as a single cluster at the outset and then successively merged. The cluster hierarchy can be represented as a tree-structure.
2. **Affinity Propagation:** It is different from other clustering algorithms as it does not require to specify the number of clusters. In this, each data point sends a message between the pair of data points until convergence. It has $O(N^2T)$ time complexity, which is the main drawback of this algorithm.

b **List out the various applications of clustering.** **[L1][CO3]** **[4M]**

Clustering algorithms have various applications across different domains. Here are some common applications of clustering:

- Customer Segmentation: Clustering is used to segment customers based on their purchasing behavior, demographics, or other attributes. This helps businesses understand customer segments and tailor marketing strategies, product recommendations, and customer support accordingly.
- Image Segmentation: Clustering is employed to partition images into meaningful regions or objects based on pixel intensities, colors, textures, or other visual features. It finds applications in computer vision, object recognition, and image processing tasks.
- Anomaly Detection: Clustering algorithms can be used to identify anomalies or outliers in datasets. By clustering normal data points

		together, any data point that does not belong to any cluster can be considered as an anomaly. This is useful in fraud detection, network intrusion detection, and detecting anomalies in sensor data. • Document Clustering: Clustering is utilized to group documents or texts based on their content or similarity. It aids in tasks like information retrieval, topic modeling, sentiment analysis, and document organization. • Recommender Systems: Clustering is used in collaborative filtering-based recommender systems to group users or items with similar preferences. This helps in making personalized recommendations by identifying clusters of users with similar tastes or clusters of items with similar characteristics. • Market Segmentation: Clustering assists in market research by segmenting markets based on customer preferences, behaviors, or demographics. This enables businesses to target specific market segments with tailored marketing campaigns and product offerings. • Gene Expression Analysis: Clustering is applied to gene expression data to identify groups of genes with similar expression patterns. This aids in understanding genetic relationships, gene function discovery, and studying diseases at a molecular level. • Image Compression: Clustering algorithms, such as vector quantization, are used in image compression techniques to group similar image patches and represent them with fewer bits. This helps in reducing the storage space required for images. • Social Network Analysis: Clustering can be used to identify communities or clusters of individuals with similar interests or social connections in social network data. It helps in understanding social relationships, influence analysis, and targeted advertising. • Traffic Pattern Analysis: Clustering algorithms can be used to analyze traffic patterns and identify groups of similar traffic flow patterns in transportation data. This aids in traffic management, route planning, and optimizing transportation systems. These are just a few examples of the wide range of applications where clustering algorithms can be employed. The suitability of clustering depends on the specific problem and the nature of the data being analyzed.		
3	a	**Illustrate the any one of latent variable models with suitable example.**	**[L3][CO3]**	**[6M]**

Latent Variable models

Latent variable models aim to model the probability distribution with latent variables.

Latent variables are a transformation of the data points into a **continuous lower-dimensional space**.

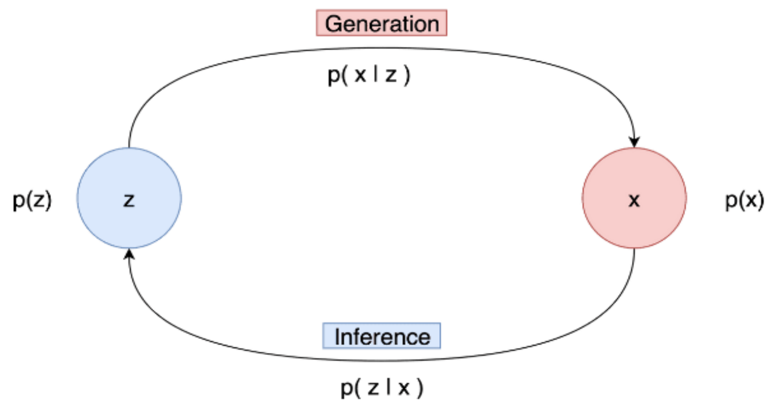
Intuitively, the latent variables will describe or "explain" the data in a simpler way.

In a stricter mathematical form, data points x that follow a probability distribution $p(x)$, are mapped into latent variables z that follow a distribution $p(z)$.

Given that idea, we can now define five basic terms:

- The **prior distribution** $p(z)$ that models the behaviour of the latent variables
- The **likelihood** $p(x|z)$ that defines how to map latent variables to the data points
- The **joint distribution** $p(x, z) = p(x|z)p(z)$, which is the multiplication of the likelihood and the prior and essentially describes our model.
- The **marginal distribution** $p(x)$ is the distribution of the original data and it is the ultimate goal of the model. The marginal distribution tells us how possible it is to generate a data point.
- The **posterior distribution** $p(z|x)$ which describes the latent variables that can be produced by a specific data point

Visually we can keep in mind the following diagram.

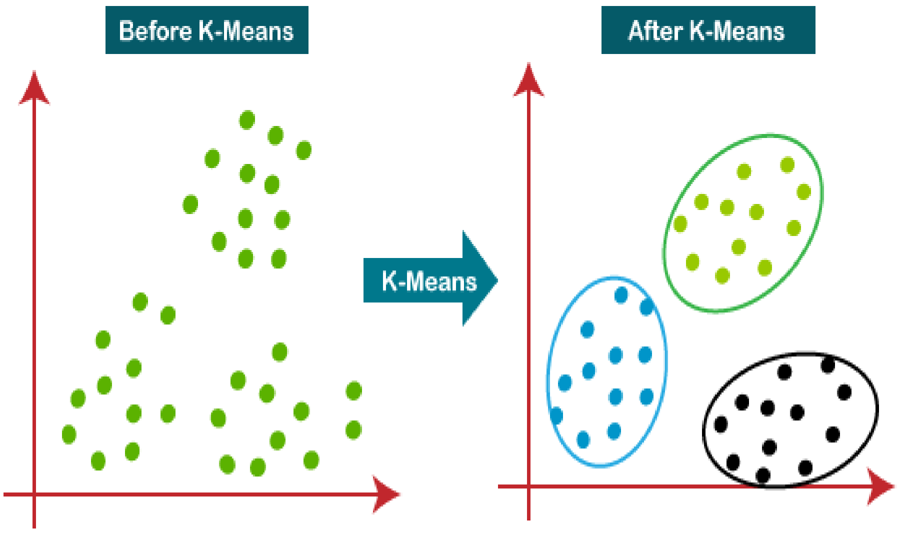


And here is the point where everything clicks together. If we assume that we somehow know the likelihood $p(x|z)$, the posterior $p(z|x)$, the marginal $p(x)$, and the prior $p(z)$ we can do the following:

		In machine learning, mixture models are a class of latent variable models that are used to represent complex distributions by combining simpler component distributions. Latent variable models involve unobserved variables (latent variables) that are used to capture hidden patterns or structure in the data. Let's consider an example of a mixture of Gaussian distributions, which is one of the most commonly used types of mixture models. In this case, the observed data is assumed to come from a combination of several Gaussian distributions. **Model Representation**: Latent Variables: We introduce a set of latent variables, often called "mixture indicators" or "cluster assignments," denoted as z. Each latent variable z corresponds to a specific component of the mixture. **Parameters**: We have a set of parameters for the mixture model, including the mixing proportions π and the parameters (mean and covariance) of each Gaussian component. **Data Generation:** **Sample Cluster**: For each data point, we first sample a latent variable z from a categorical distribution according to the mixing proportions π. This determines the component from which the data point will be generated. **Generate Data**: Given the selected component, we sample the data point x from the corresponding Gaussian distribution. **Model Inference:** Given observed data points x, the goal is to infer the latent variables z and the model parameters. Inference can be done using various techniques such as Expectation-Maximization (EM) algorithm, variational inference, or Markov chain Monte Carlo (MCMC) methods. **Model Learning**: The model parameters, including the mixing proportions π and the Gaussian parameters, are learned from the observed data using the chosen inference algorithm. The learning process involves iteratively updating the model parameters until convergence, maximizing the likelihood or posterior probability of the observed data. **Model Utilization**: Once the model is learned, it can be used for various tasks such as clustering, density estimation, anomaly detection, or generating new data points from the learned distribution. Mixture models are powerful tools in machine learning as they can capture complex data distributions by combining simpler components. They are widely used in various domains, including image analysis, natural language processing, recommendation systems, and many more.		
	b	**Explain applications of EM algorithm.** Applications of EM algorithm The primary aim of the EM algorithm is to estimate the missing data in the latent variables through observed data in datasets. The EM algorithm or latent variable model has a broad range of real-life applications in machine	**[L1][CO3]**	**[6M]**

		learning. These are as follows: ○ The EM algorithm is applicable in data clustering in machine learning. ○ It is often used in computer vision and NLP (Natural language processing). ○ It is used to estimate the value of the parameter in mixed models such as the **Gaussian Mixture Model**and quantitative genetics. ○ It is also used in psychometrics for estimating item parameters and latent abilities of item response theory models. ○ It is also applicable in the medical and healthcare industry, such as in image reconstruction and structural engineering. ○ It is used to determine the Gaussian density of a function.		
4	a	**Analyze the working principle of K-means Clustering.** K--Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on. It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties. It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training. It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters. The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm. The k-means clustering algorithm mainly performs two tasks: ○ Determines the best value for K center points or centroids by an iterative process. ○ Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster. Hence each cluster has datapoints with some commonalities, and it is away from other clusters.	**[L4][CO3]**	**[7M]**

The below diagram explains the working of the K-means Clustering Algorithm:



How does the K-Means Algorithm Work?

The working of the K-Means algorithm is explained in the below steps:

Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be other from the input dataset).

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step-4: Calculate the variance and place a new centroid of each cluster.

Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.

Step-7: The model is ready.

Consider any example for the explanation

b	**Give the different types of Clustering algorithms used in clustering.**	**[L2][CO3]**	**[5M]**

Here we are discussing mainly popular Clustering algorithms that are widely used in machine learning:

1. **K-Means algorithm:** The k-means algorithm is one of the most popular clustering algorithms. It classifies the dataset by dividing the samples into different clusters of equal variances. The number of clusters must be specified in this algorithm. It is fast with fewer computations required, with the linear complexity of **O(n).**
2. **Mean-shift algorithm:** Mean-shift algorithm tries to find the dense

areas in the smooth density of data points. It is an example of a centroid-based model, that works on updating the candidates for centroid to be the center of the points within a given region.

3. **DBSCAN Algorithm:** It stands **for Density-Based Spatial Clustering of Applications with Noise**. It is an example of a density-based model similar to the mean-shift, but with some remarkable advantages. In this algorithm, the areas of high density are separated by the areas of low density. Because of this, the clusters can be found in any arbitrary shape.
4. **Expectation-Maximization Clustering using GMM:** This algorithm can be used as an alternative for the k-means algorithm or for those cases where K-means can be failed. In GMM, it is assumed that the data points are Gaussian distributed.
5. **Agglomerative Hierarchical algorithm:** The Agglomerative hierarchical algorithm performs the bottom-up hierarchical clustering. In this, each data point is treated as a single cluster at the outset and then successively merged. The cluster hierarchy can be represented as a tree-structure.
6. **Affinity Propagation:** It is different from other clustering algorithms as it does not require to specify the number of clusters. In this, each data point sends a message between the pair of data points until convergence. It has O(N^2T) time complexity, which is the main drawback of this algorithm

5	a	**List out the various types of Cluster methods in unsupervised learning**.	**[L1][CO3]**	**[8M]**

Types of Clustering

Centroid-based clustering organizes the data into non-hierarchical clusters, in contrast to hierarchical clustering defined below. k-means is the most widely-used centroid-based clustering algorithm. Centroid-based algorithms are efficient but sensitive to initial conditions and outliers. This course focuses on k-means because it is an efficient, effective, and simple clustering algorithm.

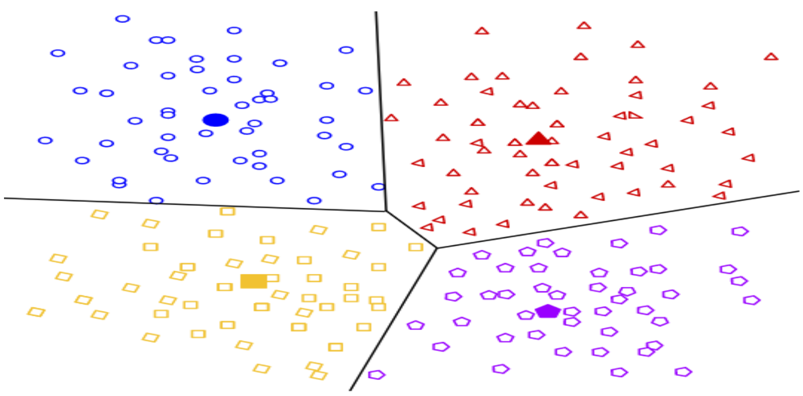


Figure 1: Example of centroid-based clustering.

Density-based Clustering

Density-based clustering connects areas of high example density into clusters. This allows for arbitrary-shaped distributions as long as dense areas can be connected. These algorithms have difficulty with data of varying densities and high dimensions. Further, by design, these algorithms do not assign outliers to clusters.

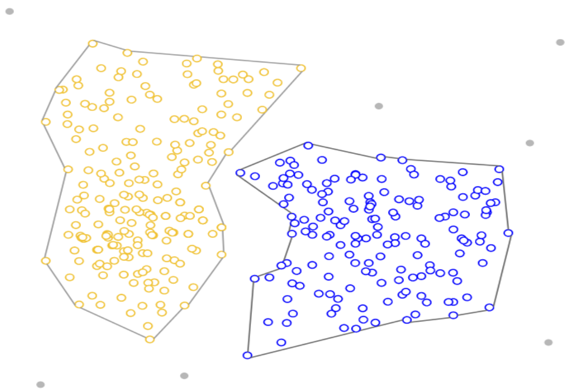


Figure 2: Example of density-based clustering.

Distribution-based Clustering

This clustering approach assumes data is composed of distributions, such as **Gaussian distributions**. In Figure 3, the distribution-based algorithm clusters data into three Gaussian distributions. As distance from the distribution's center increases, the probability that a point belongs to the distribution decreases. The bands show that decrease in probability. When you do not know the type of distribution in your data, you should use a different algorithm.

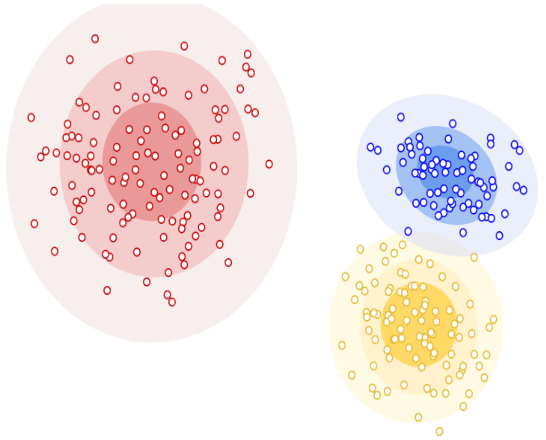


Figure 3: Example of distribution-based clustering.

Hierarchical Clustering

Hierarchical clustering creates a tree of clusters. Hierarchical clustering, not surprisingly, is well suited to hierarchical data, such as taxonomies. See *Comparison of 61 Sequenced Escherichia coli Genomes* by Oksana Lukjancenko, Trudy Wassenaar & Dave Ussery for an example. In addition, another advantage is that any number of clusters can be chosen by cutting the tree at the right level.

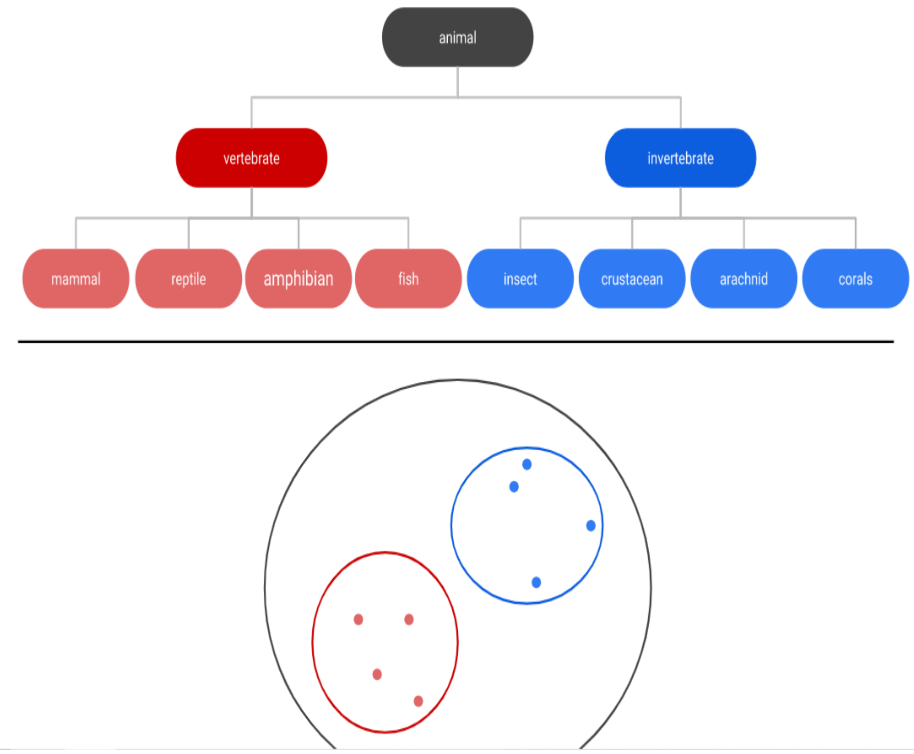
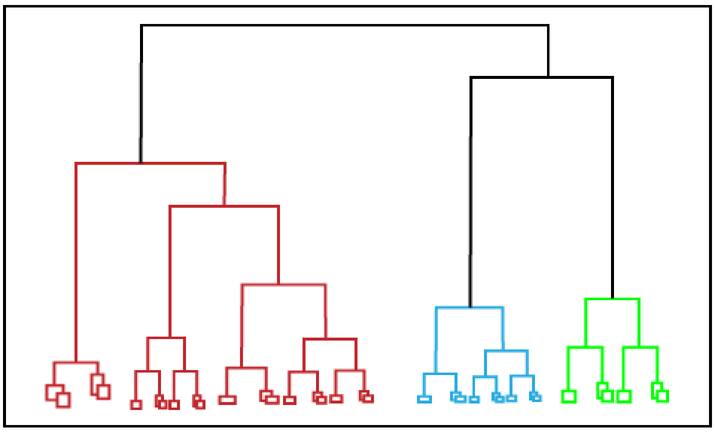


Figure 4: Example of a hierarchical tree clustering animals.

Hierarchical clustering can be used as an alternative for the partitioned clustering as there is no requirement of pre-specifying the number of clusters to be created. In this technique, the dataset is divided into clusters to create a tree-like structure, which is also called a **dendrogram**. The observations or any number of clusters can be selected by cutting the tree at the correct level. The most common example of this method is the **Agglomerative Hierarchical algorithm**.

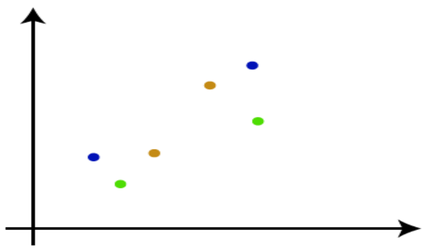


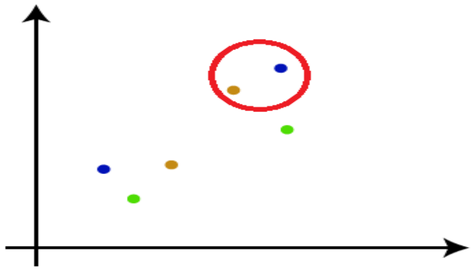
b **Infer the similarities and differences between average-link clustering and k-means?** [L4][CO3] [4M]

- Both average-link clustering and k-means are popular clustering algorithms, but they have some similarities and differences in terms of their approach and characteristics. Here's a comparison between the two:
- Similarities:
- Unsupervised Learning: Both average-link clustering and k-means

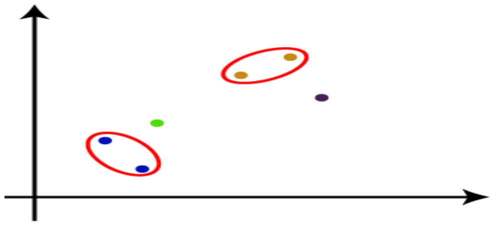
		are unsupervised learning algorithms, meaning they do not require labeled data for training. They discover patterns and groupings in the data without prior knowledge of the class labels. • Iterative Process: Both algorithms use an iterative process to refine their cluster assignments. They repeatedly update the cluster centroids or merge clusters until convergence or a stopping criterion is met. Differences: • Algorithm Type: Average-link clustering is a hierarchical clustering algorithm, whereas k-means is a centroid-based clustering algorithm. This fundamental difference affects how the clusters are formed and the overall approach to clustering. • Cluster Representation: Average-link clustering produces a hierarchical structure of clusters, often represented as a dendrogram. It captures the nested relationships between clusters and allows for fdifferent levels of granularity. In contrast, k-means produces non-overlapping, flat clusters, with each data point assigned to a single cluster. • Distance Metric: Average-link clustering typically uses a distance or dissimilarity metric, such as Euclidean distance or cosine similarity, to measure the similarity between clusters during the merging process. K-means, on the other hand, uses the distance between data points and the cluster centroids to assign points to the nearest centroid. • Number of Clusters: Average-link clustering does not require specifying the number of clusters in advance. The hierarchy can be cut at different levels to obtain different numbers of clusters. In contrast, k-means requires the user to specify the desired number of clusters (K) before running the algorithm. • Complexity: Average-link clustering can have higher computational complexity compared to k-means, especially for large datasets, as it needs to compute and update the pairwise distances between clusters in each iteration. K-means, on the other hand, has a lower computational complexity, making it more efficient for larger datasets. • Sensitivity to Initialization: K-means is sensitive to the initial placement of cluster centroids. Different initializations can result in different final cluster assignments and centroids. Average-link clustering is less sensitive to initialization because it operates on a hierarchical structure and merges clusters based on similarity		
6	a	**Generalize K-Means Clustering algorithm in Unsupervised Learning with simple example**. Generalized k-means clustering is an extension of the traditional k-means clustering algorithm that allows for more flexible and customizable clustering. While the standard k-means algorithm assigns data points to clusters based on their proximity to cluster centroids, generalized k-means clustering introduces additional parameters and distance metrics to accommodate various data types and cluster shapes. In traditional k-means clustering, each data point is assigned to the cluster with the nearest centroid, where the centroid is the mean vector of the data	**[L6][CO3]**	**[6M]**

		points in that cluster. The algorithm aims to minimize the sum of squared distances between the data points and their assigned centroids. However, this approach assumes that the clusters are spherical and that the data features are continuous and normally distributed. Generalized k-means clustering relaxes these assumptions and offers more flexibility. Here are a few key elements that can be customized in generalized k-means clustering: **Distance metrics**: Instead of relying solely on the Euclidean distance, generalized k-means allows for the use of other distance metrics that are more suitable for specific data types. For example, for categorical data, Hamming distance or Jaccard distance can be used. **Cluster shape**: Traditional k-means assumes that clusters are spherical and have equal variance. Generalized k-means allows for different cluster shapes, such as elliptical or arbitrary-shaped clusters. This is achieved by using a covariance matrix for each cluster and considering the Mahalanobis distance to measure the dissimilarity between data points and cluster centroids. **Weighting**: Generalized k-means allows for assigning different weights to different dimensions or features of the data. By assigning appropriate weights, certain dimensions can be emphasized or de-emphasized in the clustering process. **Constraints**: Generalized k-means can incorporate additional constraints into the clustering process. For example, constraints can be applied to enforce that certain data points must belong to specific clusters or that clusters must have a minimum number of data points. Overall, generalized k-means clustering offers more flexibility and adaptability to different data types and clustering scenarios. By customizing the distance metric, cluster shape, weighting, and constraints, it becomes possible to better model and analyze complex data sets in a way that suits the specific requirements of the problem at hand.		
	b	**Analyze the mixture of latent variable models.** Mixture of latent variable models (MLVMs) are a class of statistical models that combine aspects of mixture models and latent variable models. In these models, each observation is assumed to arise from one of several subpopulations (as in mixture models), and the subpopulation itself is characterized by unobserved latent variables (as in latent variable models). The key idea is that the observed data are assumed to be generated from a mixture distribution, where the mixing proportions and the parameters of the component distributions are determined by the values of the latent variables. The latent variables themselves are typically assumed to follow some distribution, such as a Gaussian distribution. MLVMs are used in a variety of applications, including clustering, classification, and density estimation. They are particularly useful when the underlying data may come from different distributions or when there are unobserved variables that affect the data generation process. One common example of an MLVM is the Gaussian mixture model (GMM), where each component in the mixture is assumed to follow a Gaussian distribution, and the latent variable indicates which component generated each observation. Another example is the latent Dirichlet allocation (LDA) model, which is used for topic modeling in text data,	**[L5][CO4]**	**[6M]**

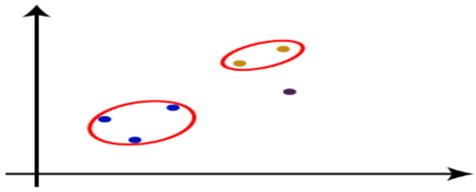
		where each document is generated from a mixture of topics, and the topics themselves are characterized by the distribution of words.		
7		**Describe the various types of Hierarchal Clustering techniques.** Hierarchical clustering is another unsupervised machine learning algorithm, which is used to group the unlabeled datasets into a cluster and also known as **hierarchical cluster analysis** or HCA. In this algorithm, we develop the hierarchy of clusters in the form of a tree, and this tree-shaped structure is known as the **dendrogram**. Sometimes the results of K-means clustering and hierarchical clustering may look similar, but they both differ depending on how they work. As there is no requirement to predetermine the number of clusters as we did in the K-Means algorithm. The hierarchical clustering technique has two approaches: 1. **Agglomerative:** Agglomerative is a **bottom-up** approach, in which the algorithm starts with taking all data points as single clusters and merging them until one cluster is left. 2. **Divisive:** Divisive algorithm is the reverse of the agglomerative algorithm as it is a **top-down approach.** Agglomerative Hierarchical clustering The agglomerative hierarchical clustering algorithm is a popular example of HCA. To group the datasets into clusters, it follows the **bottom-up approach**. It means, this algorithm considers each dataset as a single cluster at the beginning, and then start combining the closest pair of clusters together. It does this until all the clusters are merged into a single cluster that contains all the datasets. This hierarchy of clusters is represented in the form of the dendrogram. How the Agglomerative Hierarchical clustering Work? The working of the AHC algorithm can be explained using the below steps: ○ **Step-1:** Create each data point as a single cluster. Let's say there are N data points, so the number of clusters will also be N.  ○ **Step-2:** Take two closest data points or clusters and merge them to form one cluster. So, there will now be N-1 clusters.	[L2][CO4]	[12M]

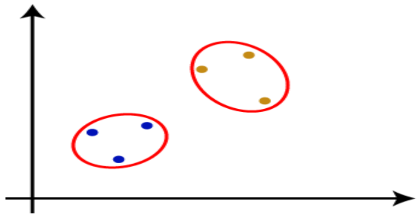


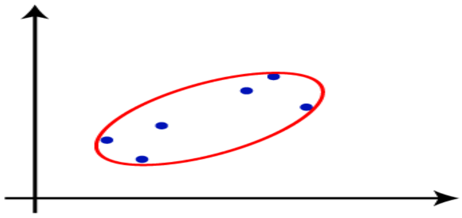
- **Step-3**: Again, take the two closest clusters and merge them together to form one cluster. There will be N-2 clusters.



- **Step-4:** Repeat Step 3 until only one cluster left. So, we will get the following clusters. Consider the below images:







- **Step-5:** Once all the clusters are combined into one big cluster, develop the dendrogram to divide the clusters as per the problem.

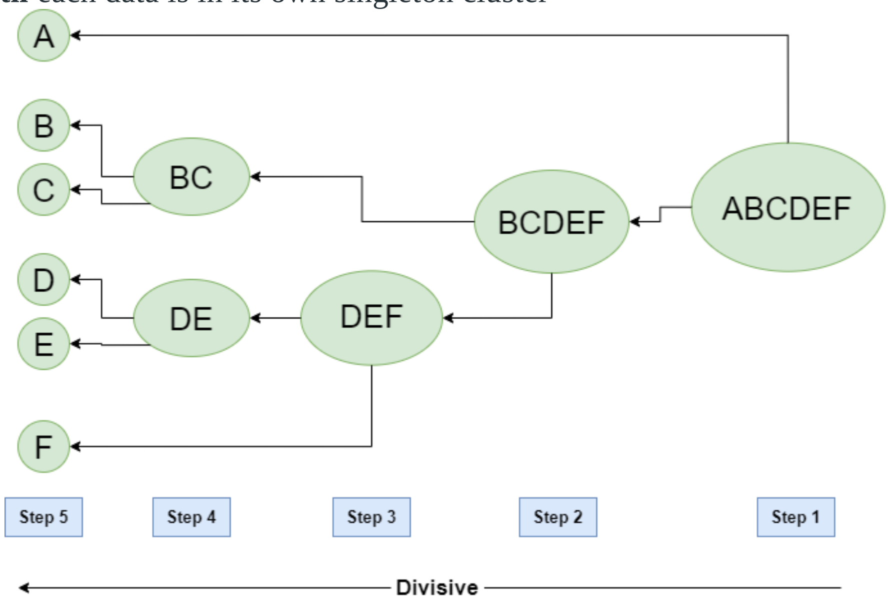
Hierarchical Divisive clustering

It is also known as a top-down approach. This algorithm also does not require to prespecify the number of clusters. Top-down clustering requires a method for splitting a cluster that contains the whole data and proceeds by splitting clusters recursively until individual data have been split into singleton clusters.

Algorithm :

given a dataset (d_1, d_2, d_3, ...d_N) of size N
at the top we have all data in one cluster
the cluster is split using a flat clustering method eg. K-Means etc
repeat
choose the best cluster among all the clusters to split
split that cluster by the flat clustering algorithm

until each data is in its own singleton cluster

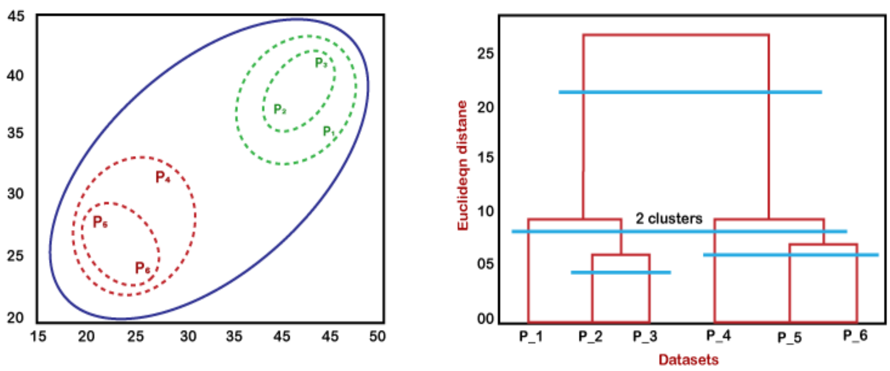


Hierarchical Divisive clustering

Woking of Dendrogram in Hierarchical clustering

The dendrogram is a tree-like structure that is mainly used to store each step as a memory that the HC algorithm performs. In the dendrogram plot, the Y-axis shows the Euclidean distances between the data points, and the x-axis shows all the data points of the given dataset.

The working of the dendrogram can be explained using the below diagram:



In the above diagram, the left part is showing how clusters are created in agglomerative clustering, and the right part is showing the corresponding dendrogram.

- As we have discussed above, firstly, the datapoints P2 and P3 combine together and form a cluster, correspondingly a dendrogram is created, which connects P2 and P3 with a rectangular shape. The hight is decided according to the Euclidean distance between the data points.
- In the next step, P5 and P6 form a cluster, and the corresponding dendrogram is created. It is higher than of previous, as the Euclidean distance between P5 and P6 is a little bit greater than the

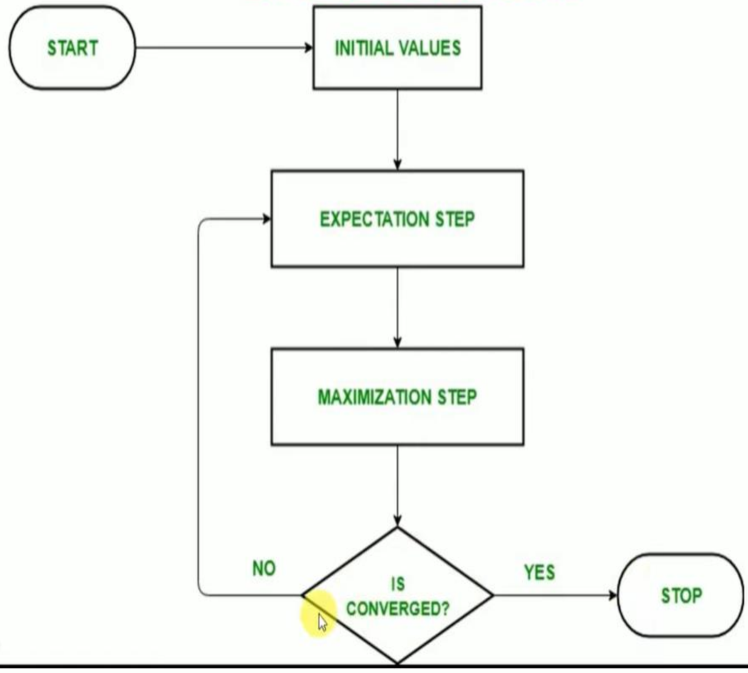
		P2 and P3. ○ Again, two new dendrograms are created that combine P1, P2, and P3 in one dendrogram, and P4, P5, and P6, in another dendrogram. ○ At last, the final dendrogram is created that combines all the data points together.		
8	a	**Analyze the Expectation-Maximization algorithm with simple Example** **EM ALGORITHM** • In the real-world applications of machine learning, it is very common that there are many relevant features available for learning but only a small subset of them are observable. • The ***Expectation-Maximization algorithm*** can be used for the latent variables (variables that are not directly observable and are actually inferred from the values of the other observed variables). • This algorithm is actually the base for many unsupervised clustering algorithms in the field of machine learning. **EM ALGORITHM** Let us understand the EM algorithm in detail. • Initially, a set of initial values of the parameters are considered. A set of incomplete observed data is given to the system with the assumption that the observed data comes from a specific model. • The next step is known as "Expectation" – step or *E-step*. In this step, we use the observed data in order to estimate or guess the values of the missing or incomplete data. It is basically used to update the variables. • The next step is known as "Maximization"-step or *M-step*. In this step, we use the complete data generated in the preceding "Expectation" – step in order to update the values of the parameters. It is basically used to update the hypothesis. • Now, in the fourth step, it is checked whether the values are converging or not, if yes, then stop otherwise repeat *step-2* and *step-3* i.e. "Expectation" – step and "Maximization" – step until the convergence occurs.	**[L4][CO4]**	**[6M]**

EM ALGORITHM

Algorithm:

1. Given a set of incomplete data, consider a set of starting parameters.
2. **Expectation step (E – step):** Using the observed available data of the dataset, estimate (guess) the values of the missing data.
3. **Maximization step (M – step):** Complete data generated after the expectation (E) step is used in order to update the parameters.
4. Repeat step 2 and step 3 until convergence.

EM ALGORITHM



Usage of EM algorithm –

- It can be used to fill the missing data in a sample.
- It can be used as the basis of unsupervised learning of clusters.
- It can be used for the purpose of estimating the parameters of Hidden Markov Model (HMM).
- It can be used for discovering the values of latent variables.

Advantages of EM algorithm –

- It is always guaranteed that likelihood will increase with each iteration.
- The E-step and M-step are often pretty easy for many problems in terms of implementation.
- Solutions to the M-steps often exist in the closed form.

Disadvantages of EM algorithm –

- It has slow convergence.
- It makes convergence to the local optima only.
- It requires both the probabilities, forward and backward (numerical optimization requires only forward probability).

Example

Let C1 and C2 be two coins.

Θ_1 be probability of getting head with C1

Θ_2 be probability of getting head with C2

Find value of Θ_1 and Θ_2 by tossing C1 and C2 for multiple times..

Toss for 5 times Choosing any of the coin randomly.

Each selected coin has to toss for 10 times.

B	H	T	T	T	H	H	T	H	T	H
A	H	H	H	H	T	H	H	H	H	H
A	H	T	H	H	H	H	H	T	H	H
B	H	T	H	T	T	T	H	H	T	T
A	T	H	H	H	T	H	H	H	T	H

$$\Theta_1 = \frac{\text{no of heads with C1}}{\text{Total no of flips using C1}}$$

$$\Theta_2 = \frac{\text{no of heads with C2}}{\text{Total no of flips using C2}}$$

If we know the coin labels the probability will be as follows:

Coin A	Coin B
	5 H, 5 T
9 H, 1 T	
8 H, 2 T	
	4 H, 6 T
7 H, 3 T	

$$\Theta_1 = 24/(24+6) = 0.8$$

$$\Theta_2 = 9/(9+11) = 0.45$$

If we don't know the identity of coin labels then we will assume or estimate the probabilities.

$\Theta_1 = 0.6$

$\Theta_2 = 0.5$

We have to use binomial distribution to find likelihood.

$$L(C)=\Theta^k (1- \Theta)^{n-k}$$

$$L(C)=\Theta^k (1- \Theta)^{n-k}$$

Likelihood For first coin Flips

$L(A)= 0.6^5 (1- 0.6)^{10-5} = 0.0007963$

$L(B)= 0.5^5 (1- 0.5)^{10-5} = 0.0009766$

P(A)=L(A)/L(A)+L(B) = 0.0007963/(0.0007963+0.0009766)=0.45

P(B)=L(B)/L(A)+L(B) = 0.0009766/(0.0007963+0.0009766) = 0.55

In similar fashion find probability of all coins with all flips. It will be as follows:

L(H): Likely no of heads L(T): Likely no of tails

											Iteration 1->:		Coin A		Coin B	
											P(A)	P(B)	L(H)	L(T)	L(H)	L(T)
B	H	T	T	T	H	H	T	H	T	H	0.45	0.55	2.2	2.2	2.8	2.8
A	H	H	H	H	T	H	H	H	H	H	0.80	0.20	7.2	0.8	1.8	0.2
A	H	T	H	H	H	H	H	T	H	H	0.73	0.27	5.9	1.5	2.1	0.5
B	H	T	H	T	T	T	H	H	T	T	0.35	0.65	1.4	2.1	2.6	3.9
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.65	0.35	4.5	1.9	2.5	1.1

For Coin A:

$\sum L(H) = 21.3$

$\sum L(T) = 8.6$

$\Theta_1 = 21.3/(21.3+8.6)$

$= 0.71$

For Coin B:

$\sum L(H) = 11.7$

$\sum L(T) = 8.4$

$\Theta_2 = 11.7/(11.7+8.4)$

$= 0.58$

These values of Θ_1 and Θ_2 will be sent to next iteration.

After 10 iteration:

$\Theta = 0.80$ and $\Theta = 0.52$

The process will be continued until you get stable value of Θ_1 and Θ_2.

In similar fashion find probability of all coins with all flips. It will be as follows:

L(H): Likely no of heads L(T): Likely no of tails

										Iteration 1->:		Coin A		Coin B	
										P(A)	P(B)	L(H)	L(T)	L(H)	L(T)
B	H	T	T	T	H	H	T	H	T	H	0.45	0.55	2.2	2.2	2.8	2.8
A	H	H	H	H	T	H	H	H	H	H	0.80	0.20	7.2	0.8	1.8	0.2
A	H	T	H	H	H	H	H	T	H	H	0.73	0.27	5.9	1.5	2.1	0.5
B	H	T	H	T	T	T	H	H	T	T	0.35	0.65	1.4	2.1	2.6	3.9
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.65	0.35	4.5	1.9	2.5	1.1

b **Explain about Supervised learning after clustering.** [L2][CO4] [6M]

After clustering, supervised learning can be applied in several ways depending on the specific goal of your analysis. Here are a few common approaches:

Label Propagation: If you have labels for a subset of your data, you can propagate these labels to the entire dataset based on the clusters. For example, if most of the data points in a cluster have a certain label, you can assign that label to all data points in that cluster.

Cluster as a Feature: You can treat the cluster assignments as additional features in your dataset and then use these features in a supervised learning model. This can sometimes improve the performance of the model, especially if the clusters capture useful information about the data.

Cluster-Specific Models: You can train a separate supervised learning model for each cluster. This allows you to capture the different characteristics of each cluster and can sometimes lead to better performance compared to a single model for the entire dataset.

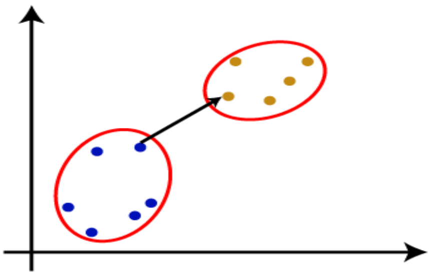
Cluster Ensemble: You can create an ensemble of supervised learning models, where each model is trained on a different cluster. This can help capture the heterogeneity of the data and improve the overall performance of the ensemble.

9	a	**Demonstrate linkage methods in Hierarchical Clustering**	[L2][CO4]	[6M]

Hierarchical clustering is a clustering algorithm that builds a hierarchy of clusters. Linkage methods are used in hierarchical clustering to determine how the distance between clusters is measured and how clusters are merged. Here, I will demonstrate three commonly used linkage methods: Single Linkage, Complete Linkage, and Average Linkage.

The **closest distance** between the two clusters is crucial for the hierarchical clustering. There are various ways to calculate the distance between two clusters, and these ways decide the rule for clustering. These measures are called **Linkage methods**. Some of the popular linkage methods are given below:

1. **Single Linkage:** It is the Shortest Distance between the closest points of the clusters. Consider the below image:



Single linkage, also known as the nearest-neighbor linkage, measures the distance between two clusters as the shortest distance between any two points in the two clusters.

Let's say we have the following data points and their pairwise distances:

A: (1, 1)
B: (2, 2)
C: (4, 4)
D: (6, 6)

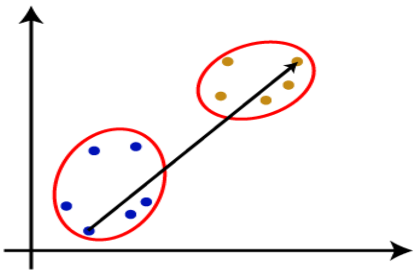
Initially, each data point is considered as a separate cluster.

Calculate the pairwise distances between all clusters:
Distance between AB: $\sqrt{(2-1)^2 + (2-1)^2} = \sqrt{2}$
Distance between AC: $\sqrt{(4-1)^2 + (4-1)^2} = \sqrt{18}$
Distance between AD: $\sqrt{(6-1)^2 + (6-1)^2} = \sqrt{50}$
Merge the two closest clusters (A and B) to form a new cluster AB.
Update the pairwise distances:
Distance between AB and C: $\sqrt{(4-2)^2 + (4-2)^2} = \sqrt{8}$
Distance between AB and D: $\sqrt{(6-2)^2 + (6-2)^2} = \sqrt{32}$
Merge the closest clusters (AB and C) to form a new cluster ABC.
Merge the last two remaining clusters (ABC and D) to obtain the final cluster ABCD.

The dendrogram representation of the clustering process would show the steps of merging clusters based on single linkage.

2. **Complete Linkage:** It is the farthest distance between the two

points of two different clusters. It is one of the popular linkage methods as it forms tighter clusters than single-linkage.



Complete linkage, also known as the farthest-neighbor linkage, measures the distance between two clusters as the maximum distance between any two points in the two clusters.

Using the same data points as before:

Calculate the pairwise distances between all clusters.
Merge the two clusters with the maximum distance between their points.
Update the pairwise distances.
Repeat steps 2 and 3 until all clusters are merged into a single cluster.

The dendrogram representation of the clustering process using complete linkage would show the steps of merging clusters based on the maximum distance.

3. **Average Linkage:** It is the linkage method in which the distance between each pair of datasets is added up and then divided by the total number of datasets to calculate the average distance between two clusters. It is also one of the most popular linkage methods.

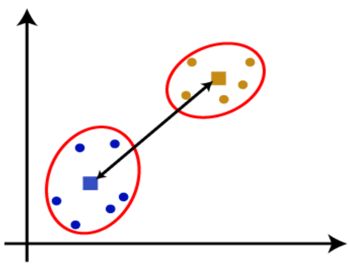
Average linkage measures the distance between two clusters as the average distance between all pairs of points from the two clusters.

Using the same data points as before:

Calculate the pairwise distances between all clusters.
Merge the two clusters with the minimum average distance between their points.
Update the pairwise distances.
Repeat steps 2 and 3 until all clusters are merged into a single cluster.

The dendrogram representation of the clustering process using average linkage would show the steps of merging clusters based on the average distance.

4. **Centroid Linkage:** It is the linkage method in which the distance between the centroid of the clusters is calculated. Consider the below image:



		From the above-given approaches, we can apply any of them according to the type of problem or business requirement. These are just examples to demonstrate the basic concepts of single linkage, complete linkage, and average linkage in hierarchical clustering. In practice, various other linkage methods and distance metrics can be used based on the specific requirements of the data and the clustering task.		
	b	**Compare Divisive and Agglomerative clustering.**	**[L6][CO4]**	**[6M]**

Difference between agglomerative clustering and Divisive clustering :

Parameters	Agglomerative Clustering	Divisive Clustering
Category	Bottom-up approach	Top-down approach
Approach	each data point starts in its own cluster, and the algorithm recursively merges the closest pairs of clusters until a single cluster containing all the data points is obtained.	all data points start in a single cluster, and the algorithm recursively splits the cluster into smaller sub-clusters until each data point is in its own cluster.
Complexity level	Agglomerative clustering is generally more computationally expensive, especially for large datasets as this approach requires the calculation of all pairwise distances between data points, which can be computationally expensive.	Comparatively less expensive as divisive clustering only requires the calculation of distances between sub-clusters, which can reduce the computational burden.
Outliers	Agglomerative clustering can handle outliers better than divisive clustering since outliers can be absorbed into larger clusters	divisive clustering may create sub-clusters around outliers, leading to suboptimal clustering results.
Interpretability	Agglomerative clustering tends to produce more interpretable results since the dendrogram shows the merging process of the clusters, and the user can choose the number of clusters based on the desired level of granularity.	divisive clustering can be more difficult to interpret since the dendrogram shows the splitting process of the clusters, and the user must choose a stopping criterion to determine the number of clusters.

10	**Summarize the following terms briefly** **i.K-means Clustering ii. Hierarchal Clustering** **Refer Q.No 4 and 7 Answers**	**[L2][CO4]**	**[12M]**

Course Code: 20CS0906 **R20**

UNIT-IV

DIMENTIONALITY REDUCTION

&

NONPARAMETRIC METHODS

<table>
<tr>
<td>1</td>
<td>a</td>
<td>

Explain about Dimensionality reduction and its techniques

The number of input features, variables, or columns present in a given dataset is known as dimensionality, and the process to reduce these features is called dimensionality reduction.

A dataset contains a huge number of input features in various cases, which makes the predictive modeling task more complicated. Because it is very difficult to visualize or make predictions for the training dataset with a high number of features, for such cases, dimensionality reduction techniques are required to use.

Dimensionality reduction technique can be defined as, *"It is a way of converting the higher dimensions dataset into lesser dimensions dataset ensuring that it provides similar information."* These techniques are widely used in machine learning for obtaining a better fit predictive model while solving the classification and regression problems.

It is commonly used in the fields that deal with high-dimensional data, such as **speech recognition, signal processing, bioinformatics, etc. It can also be used for data visualization, noise reduction, cluster analysis**, etc.

Approaches of Dimension Reduction

There are two ways to apply the dimension reduction technique, which are given below:

Feature Selection

Feature selection is the process of selecting the subset of the relevant features and leaving out the irrelevant features present in a dataset to build a model of high accuracy. In other words, it is a way of selecting the optimal features from the input dataset.

Three methods are used for the feature selection:

1. Filters Methods

In this method, the dataset is filtered, and a subset that contains only the relevant features is taken. Some common techniques of filters method are:

- **Correlation**
- **Chi-Square Test**
- **ANOVA**

</td>
<td>[L2][CO5]</td>
<td>[6M]</td>
</tr>
</table>

- **Information Gain, etc.**

2. Wrappers Methods

The wrapper method has the same goal as the filter method, but it takes a machine learning model for its evaluation. In this method, some features are fed to the ML model, and evaluate the performance. The performance decides whether to add those features or remove to increase the accuracy of the model. This method is more accurate than the filtering method but complex to work. Some common techniques of wrapper methods are:

- Forward Selection
- Backward Selection
- Bi-directional Elimination

3. Embedded Methods: Embedded methods check the different training iterations of the machine learning model and evaluate the importance of each feature. Some common techniques of Embedded methods are:

- **LASSO**
- **Elastic Net**
- **Ridge Regression, etc.**

Feature Extraction:
Feature extraction is the process of transforming the space containing many dimensions into space with fewer dimensions. This approach is useful when we want to keep the whole information but use fewer resources while processing the information.

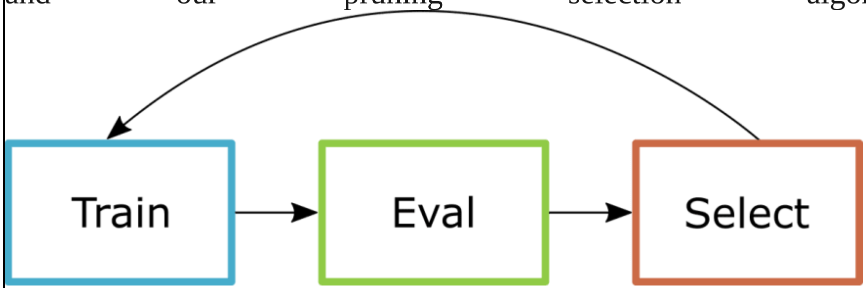
Some common feature extraction techniques are:

- Principal Component Analysis
- Linear Discriminant Analysis
- Kernel PCA
- Quadratic Discriminant Analysis

Factor Analysis
Factor analysis is a technique in which each variable is kept within a group according to the correlation with other variables, it means variables within a group can have a high correlation between themselves, but they have a low correlation with variables of other groups.

We can understand it by an example, such as if we have two variables Income and spend. These two variables have a high correlation, which means people with high income spends more, and vice versa. So, such variables are put into a group, and that group is known as the **factor**. The number of these factors will be reduced as compared to the original dimension of the dataset.

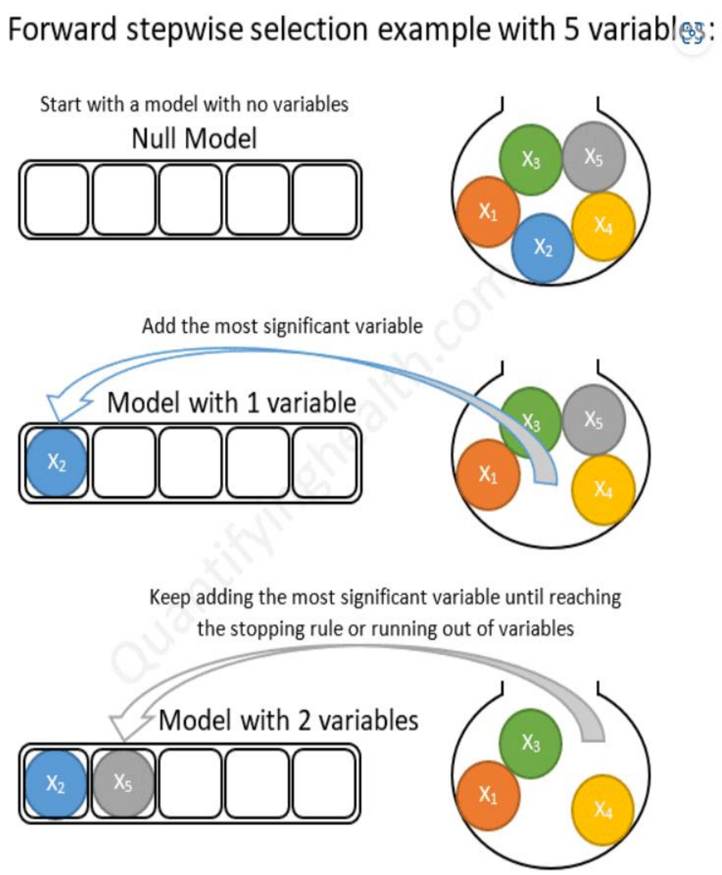
Auto-encoders

		One of the popular methods of dimensionality reduction is auto-encoder, which is a type of ANN or artificial neural network, and its main aim is to copy the inputs to their outputs. In this, the input is compressed into latent-space representation, and output is occurred using this representation. It has mainly two parts: ○ **Encoder:** The function of the encoder is to compress the input to form the latent-space representation. **Decoder:** The function of the decoder is to recreate the output from the latent-space representation.		
	b	**List out the categories of features subset selection and explain it.** **Subset selection** **Subset selection** is a feature selection algorithm that can variate between a **forward selection and a backward selection.** Both methods consist in finding a subset of the initial features that contain the least number of dimensions that most contribute to accuracy. A naive approach would be to try all the 2^n possible subset combinations but if the number of dimensions is too big it would take forever. Instead, based on a **heuristic** function (error function) we add or remove features. The performance of subset selection depends highly on the model we choose and our pruning selection algorithm.  **Forward selection** In forward selection we start with an empty set of features, for each feature that is not in the set we train the model with it and test its performance; we then select the feature with the **least amount of error**. We continue adding new features for the model to train until the error is low enough or until we have selected a proportion of the total features. **Forward Feature Selection is a feature selection technique that iteratively builds a model by adding one feature at a time, selecting the feature that maximizes model performance.** It starts with an empty set of features and adds the most predictive feature in each iteration until a stopping criterion is met. T his method is particularly useful when dealing with a large number of features, as it incrementally builds the model based on the most informative features. This process involves assessing new features, evaluating combinations of features, and selecting the optimal subset of features that best contribute to model accuracy. **Forward stepwise selection** (or **forward selection**) is a variable	**[L2][CO5]**	**[6M]**

selection method which:

1. **Begins** with a model that contains no variables (called the *Null Model*)
2. **Then** starts adding the most significant variables one after the other
3. **Until** a pre-specified stopping rule is reached or until all the variables under consideration are included in the model

Here's an example of forward selection with 5 variables:

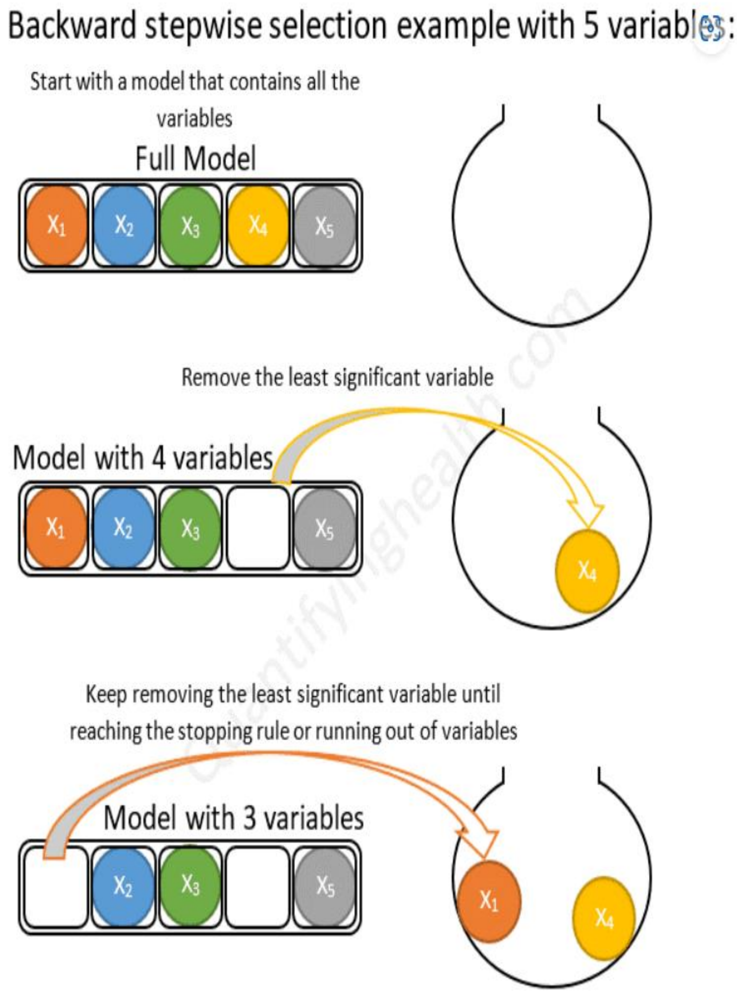


Backward stepwise

Backward stepwise selection (or **backward elimination**) is a variable selection method which:

1. **Begins** with a model that contains all variables under consideration (called the *Full Model*)
2. **Then** starts removing the least significant variables one after the other
3. **Until** a pre-specified stopping rule is reached or until no variable is left in the model

Here's an example of backward elimination with 5 variables:



2	a	**Discuss the Principle Component Analysis.**	[L2][CO5]	[6M]

Principal Component Analysis 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.

PCA generally tries to find the lower-dimensional surface to project the high-dimensional data.

PCA works by considering the variance of each attribute because the high attribute shows the good split between the classes, and hence it reduces the dimensionality. 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.

The PCA algorithm is based on some mathematical concepts such as:

- ariance and Covariance
- Eigenvalues and Eigen factors

Some common terms used in PCA algorithm:

- **Dimensionality:** It is the number of features or variables present in the given dataset. More easily, it is the number of columns present in the dataset.
- **Correlation:** It signifies that how strongly two variables are related to each other. Such as if one changes, the other variable

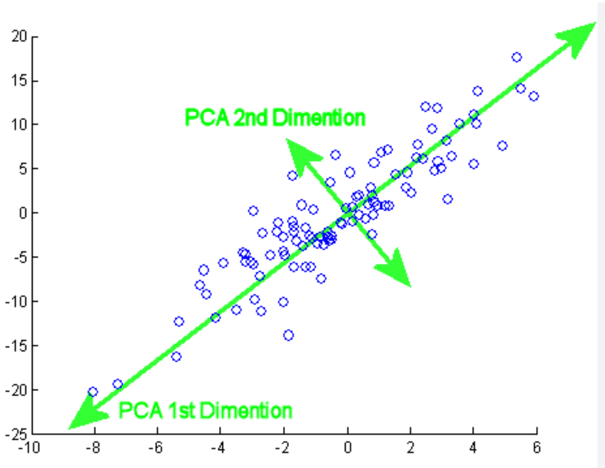
also gets changed. The correlation value ranges from -1 to +1. Here, -1 occurs if variables are inversely proportional to each other, and +1 indicates that variables are directly proportional to each other.

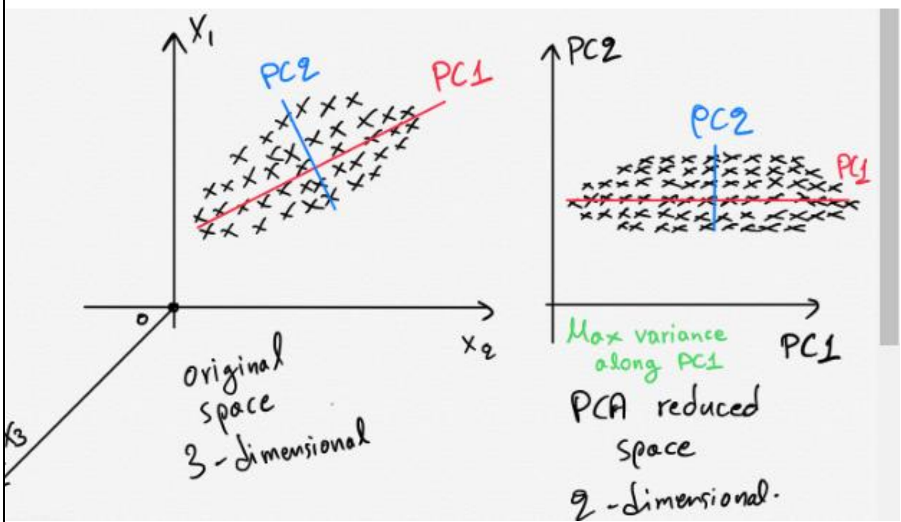
- **Orthogonal:** It defines that variables are not correlated to each other, and hence the correlation between the pair of variables is zero.
- **Eigenvectors:** If there is a square matrix M, and a non-zero vector v is given. Then v will be eigenvector if Av is the scalar multiple of v.
- **Covariance Matrix:** A matrix containing the covariance between the pair of variables is called the Covariance Matrix.

Principal Components in PCA

As described above, the transformed new features or the output of PCA are the Principal Components. The number of these PCs are either equal to or less than the original features present in the dataset. Some properties of these principal components are given below:

- The principal component must be the linear combination of the original features.
- These components are orthogonal, i.e., the correlation between a pair of variables is zero.
- The importance of each component decreases when going to 1 to n, it means the 1 PC has the most importance, and n PC will have the least importance.

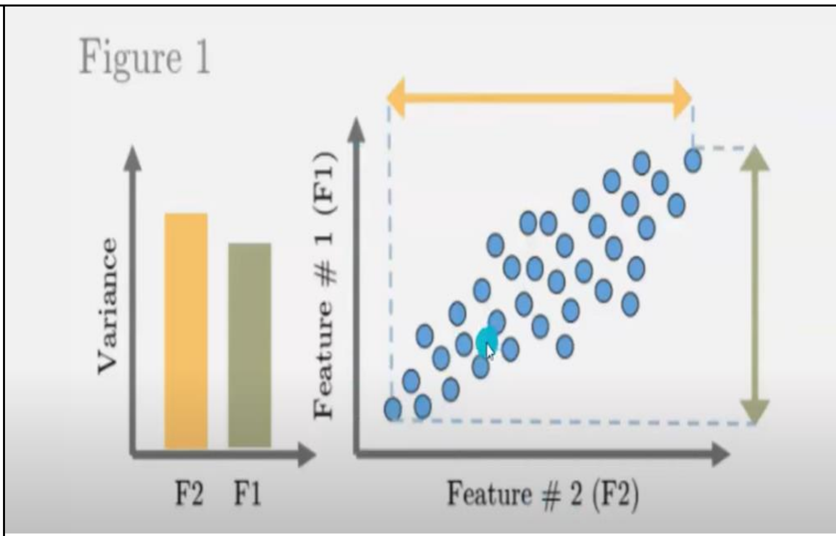
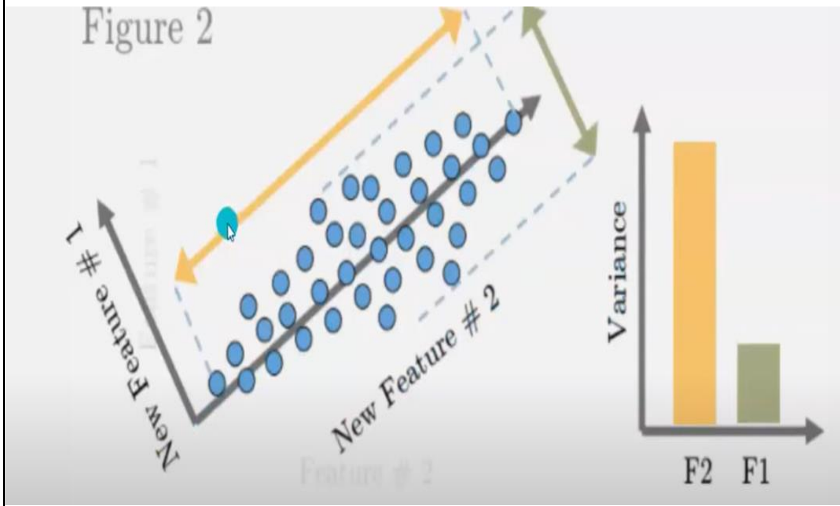
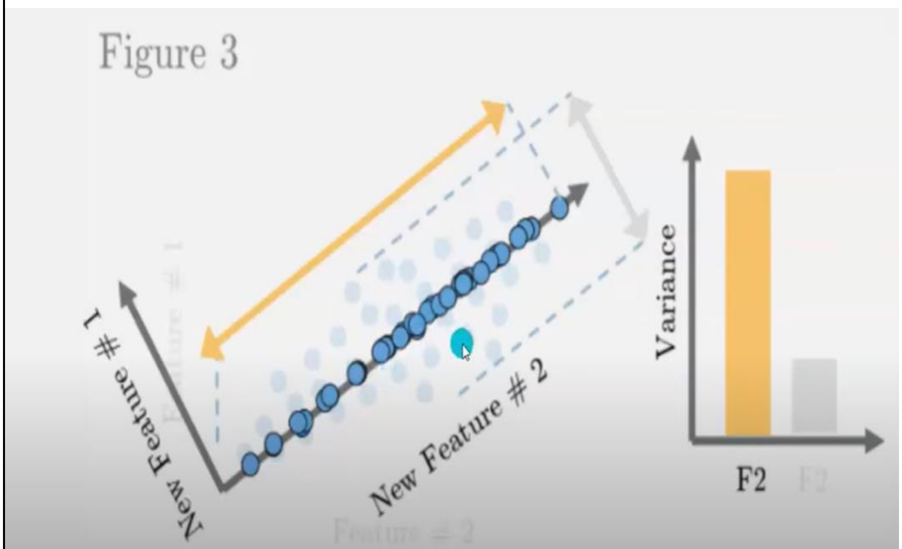




Steps for PCA algorithm

1. **Getting the dataset**
   Firstly, we need to take the input dataset and divide it into two subparts X and Y, where X is the training set, and Y is the validation set.
2. **Representing data into a structure**
   Now we will represent our dataset into a structure. Such as we will represent the two-dimensional matrix of independent variable X. Here each row corresponds to the data items, and the column corresponds to the Features. The number of columns is the dimensions of the dataset.
3. **Standardizing the data**
   In this step, we will standardize our dataset. Such as in a particular column, the features with high variance are more important compared to the features with lower variance. If the importance of features is independent of the variance of the feature, then we will divide each data item in a column with the standard deviation of the column. Here we will name the matrix as Z.
4. **Calculating the Covariance of Z**
   To calculate the covariance of Z, we will take the matrix Z, and will transpose it. After transpose, we will multiply it by Z. The output matrix will be the Covariance matrix of Z.
5. **Calculating the Eigen Values and Eigen Vectors**
   Now we need to calculate the eigenvalues and eigenvectors for the resultant covariance matrix Z. Eigenvectors or the covariance matrix are the directions of the axes with high information. And the coefficients of these eigenvectors are defined as the eigenvalues.
6. **Sorting the Eigen Vectors**
   In this step, we will take all the eigenvalues and will sort them in decreasing order, which means from largest to smallest. And simultaneously sort the eigenvectors accordingly in matrix P of eigenvalues. The resultant matrix will be named as P*.
7. **Calculating the new features Or Principal Components**
   Here we will calculate the new features. To do this, we will multiply the P* matrix to the Z. In the resultant matrix Z*, each observation is the linear combination of original features. Each column of the Z* matrix is independent of each other.
8. **Remove less or unimportant features from the new dataset.**
   The new feature set has occurred, so we will decide here what to keep and what to remove. It means, we will only keep the relevant or important features in the new dataset, and unimportant features will be removed out.

Example:

		  		
	b	**List out the Applications of PCA in machine learning.**	[L1][CO5]	[6M]
		Dimensionality Reduction: PCA is primarily used for reducing the dimensionality of datasets with many features (or variables) while preserving most of the variability present in the data. This is particularly useful in high-dimensional datasets where the number of features exceeds the number of observations. By reducing the dimensionality, PCA simplifies the dataset, making it easier to visualize, analyze, and model. **Data Visualization**: PCA is widely employed for data visualization		

purposes. It projects high-dimensional data onto a lower-dimensional space, typically two or three dimensions, making it possible to visualize complex datasets in a more comprehensible form. This is especially useful for exploring patterns, clusters, and relationships in the data.

Feature Extraction: PCA can be used to extract a smaller set of features (or principal components) that capture the most important information in the data. These extracted features can then be used as input for downstream machine learning algorithms. Feature extraction with PCA is particularly beneficial for improving the efficiency and performance of predictive models, especially when dealing with high-dimensional data.

Noise Reduction: PCA can help in reducing noise or redundant information present in the data by focusing on the most significant sources of variation. By retaining only the principal components that explain the majority of the variance in the data, PCA can effectively filter out noise and irrelevant features, leading to improved model performance.

Data Preprocessing: PCA is often used as a preprocessing step before applying other machine learning techniques. It can help in improving the performance and convergence of algorithms by reducing multicollinearity among features and mitigating the curse of dimensionality.

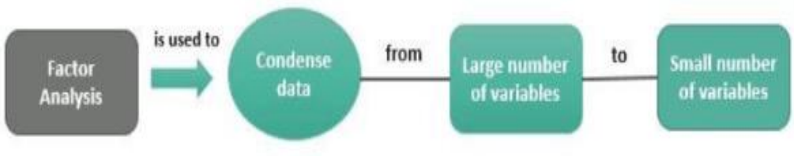
Image Processing: PCA finds applications in image processing tasks such as face recognition, image compression, and denoising. In image compression, for example, PCA can be used to transform high-dimensional image data into a lower-dimensional representation while preserving most of the important image features.

Signal Processing: PCA is utilized in signal processing for tasks such as noise reduction, feature extraction, and classification. It can help in identifying underlying patterns and structures in signals, making it easier to analyze and interpret complex signal data.

These are just a few examples of the many applications of PCA across various domains, highlighting its versatility and importance in data analysis and machine learning.

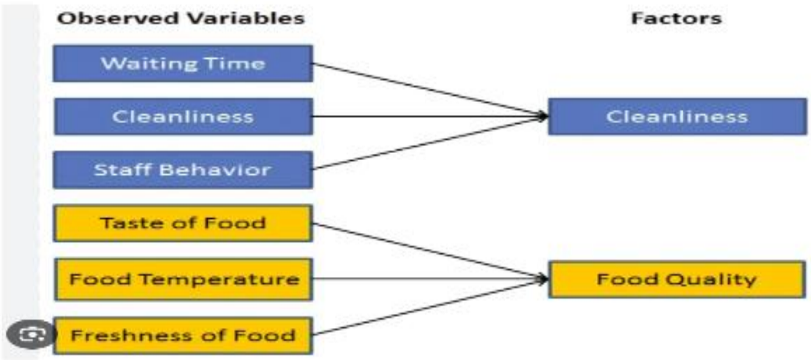
3	a	**Describe the Factor Analysis Technique.**	**[L2][CO5]**	**[6M]**

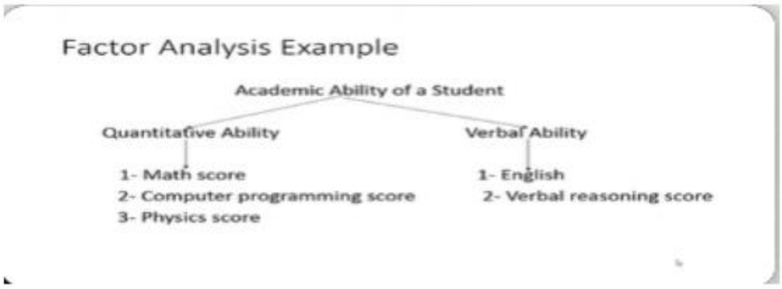
Factor Analytics is a special technique reducing the huge number of variables into a few numbers of factors is known as factoring of the data, and managing which data is to be present in sheet comes under factor analysis. It is completely a statistical approach that is also used to describe fluctuations among the observed and correlated variables in terms of a potentially lower number of unobserved variables called *factors.*



Factor analysis is a very effective tool for inspecting changeable relationships for complex concepts such as social status, economic status, dietary patterns, psychological scales, biology, psychometrics, personality theories, marketing, product management, operations research, finance, etc.

For example:





The main goal of factor analysis is to reduce the dimensionality of the data by identifying a smaller number of unobservable factors that account for the common variance among the observed variables. It assumes that each observed variable is influenced by a combination of these latent factors.

The process of factor analysis involves the following steps:

1. **Data Collection**: Collect a dataset consisting of a set of observed variables (also known as indicators) that are believed to be related to the underlying factors.
2. **Factor Extraction**: Use a statistical method, such as Principal Component Analysis (PCA) or Maximum Likelihood Estimation (MLE), to extract the initial set of factors. These factors are linear combinations of the observed variables that capture the most significant sources of variance in the data.
3. **Factor Rotation**: After extracting the factors, it is often useful to rotate them to achieve a more interpretable and meaningful solution. Orthogonal rotation methods (e.g., Varimax, Quartimax) or oblique rotation methods (e.g., Promax, Oblimin) can be applied to achieve different goals, such as maximizing factor loadings or allowing for correlations among factors.
4. **Factor Interpretation**: Once the factors are extracted and rotated, the researcher interprets the meaning of each factor based on the pattern of loadings. Loadings represent the strength and direction of the relationship between each observed variable and the underlying factor. Variables with high loadings on a factor are considered to be strongly associated with that factor.

Types of factor analysis:

Exploratory factor analysis (EFA) :

It is used to identify composite inter-relationships among items and group items that are the part of uniting concepts. The Analyst can't make any prior assumptions about the relationships among factors. It is also used to find the fundamental structure of a huge set of variables. It lessens the large data to a much smaller set of summary variables. It is almost similar to the **Confirmatory Factor Analysis(CFA).**

Similarities are:

- Evaluate the internal reliability of an amount.
- Examine the factors represented by item sets. They presume that the factors aren't correlated.
- Investigate the grade/class of each item.

Confirmatory factor analysis (CFA) :

It is a more complex(composite) approach that tests the theory that the items are associated with specific factors. Confirmatory Factor Analysis uses a properly structured equation model to test a measurement model whereby loading on the factors allows for the evaluation of relationships between observed variables and unobserved variables.

It is similar to the Exploratory Factor Analysis.
The main difference between the two is:

- Simply use Exploratory Factor Analysis to explore the pattern.
- Use Confirmatory Factor Analysis to perform hypothesis testing.

Multiple Factor Analysis :
This type of Factor Analysis is used when your variables are structured in changeable groups. For example, you may have a teenager's health questionnaire with several points like sleeping patterns, wrong addictions, psychological health, mobile phone addiction, or learning disabilities.
The Multiple Factor Analysis is performed in two steps which are:-

- Firstly, the Principal Component Analysis will perform on each and every section of the data. Further, this can give a useful eigenvalue, which is actually used to normalize the data sets for further use.
- The newly formed data sets are going to merge into a distinctive matrix and then global PCA is performed.

Generalized Procrustes Analysis (GPA) :
The Procrustes analysis is actually a suggested way to compare then the two approximate sets of configurations and shapes, which were originally developed to equivalent to the two solutions from Factor Analysis, this technique was actually used to extend the GP Analysis so that more than two shapes could be compared in many ways. The shapes are properly aligned to achieve the target shape. Mainly GPA (Generalized Procrustes Analysis) uses geometric transformations.
Geometric progressions are :

- Isotropic rescaling,
- Reflection,
- Rotation,
- Translation of matrices to compare the sets of data.

Eigenvalues
When factor analysis going to generate the factors, each and every factor has ab associated eigenvalue which will give the total variance explained by each factor.
Usually, the factors having eigenvalues greater than 1 are useful :
Percentage of variation explained by

F1 = Eigenvalue of Factor 1/No. of Variables

Percentage of variation explained by

F2 = Eigenvalue of Factor 2/No. of Variables

FactorLoadings
In addition, factors are created with equality; some factors have more weights some have low. In a simple example, imagine your car company says Maruti Suzuki is conducting a survey includes, using – telephonic survey, physical survey, google forms, etc. for customer satisfaction and the results show the following factor loadings:

VARIABLE	F1	F2	F3

Problem 1	0.985	0.111	-0.032
Problem 2	0.724	0.008	0.167
Problem 3	0.798	0.180	0.345

Here –
F1 – Factor 1
F2 – Factor 2
F3 – Factor 3

The factors that affect the question the most (and therefore have the highest factor loadings) are bolded. Factor loadings are similar to correlation coefficients in that they can vary from -1 to 1. The closer factors are to -1 or 1, the more they affect the variable.

b **List out the applications of Factor Analysis.** **[L1][CO5]** **[6M]**

Applications of Factor Analysis

- **Marketing**

Marketing strategies can significantly benefit from the statistical method of analysis. Companies can use these techniques to determine a correlation between different factors or variables of a marketing campaign.
Furthermore, it can build connections with consequent feedback and customer satisfaction. It ensures that you verify the efficacy of a marketing campaign and its impact on the target market.

- **Nutrition**

It can build a connection between the nutritional health of an individual and their diet. To establish that, this type analysis focuses on the dietary practices of a certain population. Moreover, the nutritional intake of an individual and their consequent health status has enabled nutritionists to determine the right quantity of nutrients one individual should consume within a specific time period.

Data Mining

In data mining, this analysis is as crucial as artificial intelligence. Factor analysis can classify a complex and vast dataset into filtered-out variables that have some connection with each other. It helps simplify the process of data mining.
Data scientists have always struggled with finding connections between different variables. But data mining has become much more advanced due to factor analysis.

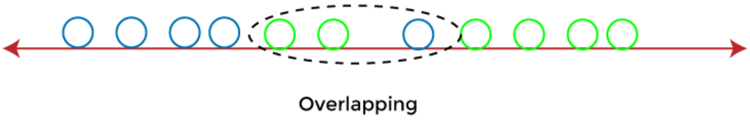
Machine Learning

Machine Learning and data mining techniques complement one another. Maybe this explains why there are tools and methodologies for machine learning to perform factor analysis.
Factor Analysis in machine learning is used to reduce the number of variables in a given dataset to obtain a more accurate and enhanced collection of observable factors. Multiple algorithms based on machine learning are used to work in this manner.

Automotive industry

The use of factor analysis in the automotive industry was mentioned as far back as 1997 in an article by Professor Emeritus Richard B. Darlington of Cornell University. He explained how a study could be

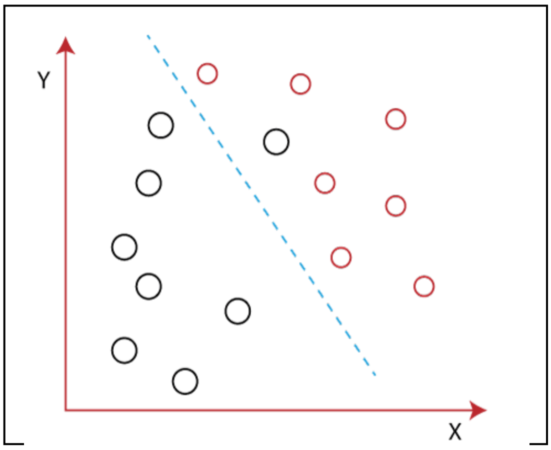
		used to identify all the variables that apply to the decision-making of purchasing a car—size, pricing, options, accessories, and more. The study could then be used to arrive at a few key variables that actually close a purchase decision. Automotive dealers can then tailor their offerings to cater to the market. **Investing** The key to a productive investment portfolio is diversification. To ensure a diverse portfolio, investment professionals use factor analysis to predict movement across a wide sector of industries and provide insights on factors that may be under the radar. For example, the average portfolio contains stocks of industries like technology and commodities. A look at the rise in stock prices of a related industry, like oil, will give investment professionals a good idea on what to sell and retain. **Human resources** There are many factors that go into a company's hiring process. With statistics, human resource professionals will be able to create a comfortable and productive working environment. Several variables can be compared and analyzed to see which combination in terms of the number of team members, varied skill sets, and contractual or in-house talent works, improving the overall functioning of the organization. **Restaurants** For restaurants, factor analysis can be used to understand demographics and target diners in the creation of menus. A fast-food restaurant opening next to a university campus will have to plan its menu differently than if it was placed in a high-end shopping location. Factors such as surrounding competition, foot-traffic, age-groups, and location all determine success. **Education** When hiring teachers and deciding on a curriculum for the school year, factor analysis plays a huge role. It is used to determine classroom sizes, staffing limits, salary distribution, and a wide range of other requirements necessary for the school year to run smoothly.		
4	a	**Explain Linear Discriminant Analysis?** Linear Discriminant Analysis (LDA) is one of the commonly used dimensionality reduction techniques in machine learning to solve more than two-class classification problems. It is also known as Normal Discriminant Analysis (NDA) or Discriminant Function Analysis (DFA). This can be used to project the features of higher dimensional space into lower-dimensional space in order to reduce resources and dimensional costs. In this topic, "Linear Discriminant Analysis (LDA) in machine learning" Linear Discriminant analysis is one of the most popular dimensionality reduction techniques used for supervised classification problems in machine learning. It is also considered a pre-processing step for modeling differences in ML and applications of pattern classification. Whenever there is a requirement to separate two or more classes having multiple features efficiently, the Linear Discriminant Analysis model is considered the most common technique to solve such classification problems. For e.g., if we have two classes with multiple features and need to separate them efficiently. When we classify them using a single feature, then it may show overlapping.	**[L2][CO5]**	**[8M]**



To overcome the overlapping issue in the classification process, we must increase the number of features regularly.

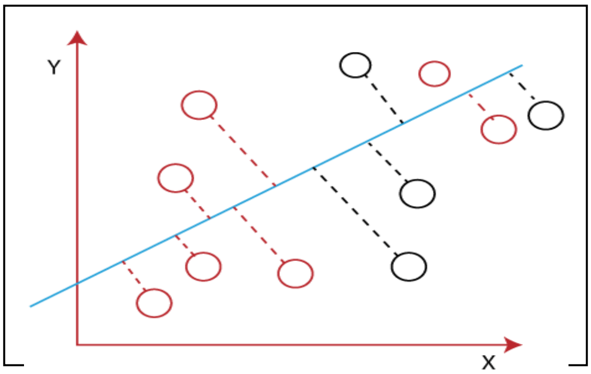
Example:

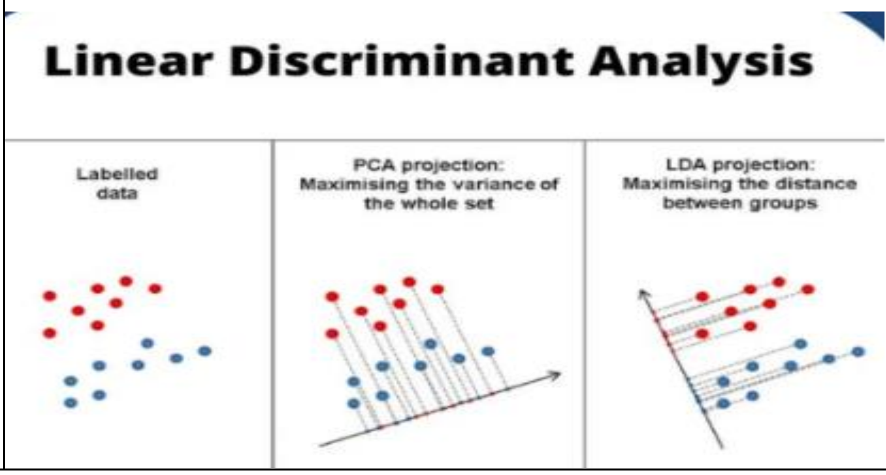
Let's assume we have to classify two different classes having two sets of data points in a 2-dimensional plane as shown below image:



Linear Discriminant analysis is used as a dimensionality reduction technique in machine learning, using which we can easily transform a 2-D and 3-D graph into a 1-dimensional plane.

Let's consider an example where we have two classes in a 2-D plane having an X-Y axis, and we need to classify them efficiently. As we have already seen in the above example that LDA enables us to draw a straight line that can completely separate the two classes of the data points. Here, LDA uses an X-Y axis to create a new axis by separating them using a straight line and projecting data onto a new axis.





		To create a new axis, Linear Discriminant Analysis uses the following criteria: ○ It maximizes the distance between means of two classes. ○ It minimizes the variance within the individual class. Using the above two conditions, LDA generates a new axis in such a way that it can maximize the distance between the means of the two classes and minimizes the variation within each class. In other words, we can say that the new axis will increase the separation between the data points of the two classes and plot them onto the new axis. Extension to Linear Discriminant Analysis (LDA) Linear Discriminant analysis is one of the most simple and effective methods to solve classification problems in machine learning. It has so many extensions and variations as follows: 1. Quadratic Discriminant Analysis (QDA): For multiple input variables, each class deploys its own estimate of variance. 2. Flexible Discriminant Analysis (FDA): it is used when there are non-linear groups of inputs are used, such as splines. Flexible Discriminant Analysis (FDA): This uses regularization in the estimate of the variance (actually covariance) and hence moderates the influence of different variables on LDA.		
	b	**Outline the various applications of Linear Discriminant Analysis?** **Applications of LDA** Some of the common real-world applications of Linear discriminant Analysis are given below: **FaceRecognition** Face recognition is the popular application of computer vision, where each face is represented as the combination of a number of pixel values. In this case, LDA is used to minimize the number of features to a manageable number before going through the classification process. It generates a new template in which each dimension consists of a linear combination of pixel values. If a linear combination is generated using Fisher's linear discriminant, then it is called Fisher's face. **Medical** In the medical field, LDA has a great application in classifying the patient disease on the basis of various parameters of patient health and the medical treatment which is going on. On such parameters, it classifies disease as mild, moderate, or severe. This classification helps the doctors in either increasing or decreasing the pace of the treatment. **Customer Identification** In customer identification, LDA is currently being applied. It means with the help of LDA; we can easily identify and select the features that can specify the group of customers who are likely to purchase a specific product in a shopping mall. This can be helpful when we want to identify a group of customers who mostly purchase a product in a shopping mall. **For Predictions** LDA can also be used for making predictions and so in decision making. For example, "will you buy this product" will give a predicted result of either one or two possible classes as a buying or not. **In Learning**	**[L1][CO5]**	**[4M]**

		Nowadays, robots are being trained for learning and talking to simulate human work, and it can also be considered a classification problem. In this case, LDA builds similar groups on the basis of different parameters, including pitches, frequencies, sound, tunes, etc.		
5	a	**Compare Multidimensionality scaling and Metric dimensionality scaling.** Multidimensional scaling (MDS) and Metric multidimensional scaling (MMDS) are both techniques used in data analysis to visualize and analyze the relationships between objects or entities based on their similarities or dissimilarities. However, there are some key differences between these two methods. Conceptual Difference: MDS: Multidimensional scaling is a general term that refers to a family of methods aimed at representing the structure of similarity or dissimilarity data in a lower-dimensional space. MDS attempts to preserve the original distances or dissimilarities between objects in the data. MMDS: Metric multidimensional scaling is a specific form of MDS that assumes the underlying distances or dissimilarities between objects are metric (i.e., satisfy the triangle inequality). It aims to find a low-dimensional representation that not only preserves the ordinal relationships between objects but also satisfies the triangle inequality. Mathematical Difference: MDS: MDS techniques, such as classical MDS or non-metric MDS, focus on finding a configuration of points in a lower-dimensional space that best approximates the pairwise dissimilarities between objects. It uses optimization algorithms to minimize the discrepancy between observed dissimilarities and distances in the reduced space. MMDS: MMDS, on the other hand, specifically deals with metric dissimilarities. It constructs a Euclidean distance matrix based on the dissimilarities and then applies classical MDS to obtain a low-dimensional representation that respects the metric properties of the data. Data Requirements: MDS: MDS can handle various types of dissimilarity measures, including ordinal, interval, or even non-metric dissimilarities. It is more flexible in terms of data requirements and can be applied to both metric and non-metric data. MMDS: MMDS assumes that the dissimilarity measures are metric, meaning they obey the triangle inequality. This assumption restricts its applicability to situations where the data can be represented by a metric space. Preserved Relationships: MDS: In MDS, the goal is to preserve the original pairwise dissimilarities or similarities as closely as possible in the lower-dimensional space. The emphasis is on preserving the ordinal relationships between objects. MMDS: MMDS aims to preserve the metric relationships between objects, in addition to the ordinal relationships. It ensures that the distances between objects in the reduced space conform to the triangle inequality.	[L5][CO5]	[6M]

	b	**List out the applications of MDS.** Multidimensional scaling (MDS) has various applications across different fields. Some of the common applications of MDS include: **Psychology and Cognitive Science**: MDS is widely used in psychology and cognitive science to understand and visualize how individuals perceive and organize information. It can be used to study mental representations of concepts, semantic relationships, and similarity judgments. **Marketing and Consumer Research**: MDS is used to analyze consumer preferences, brand positioning, and product mapping. By representing consumer perceptions in a lower-dimensional space, MDS helps identify market segments, understand product similarities, and optimize marketing strategies. **Social Sciences**: MDS is applied in social sciences, such as sociology and political science, to explore and map social structures and relationships. It helps understand social networks, analyze intergroup relations, and visualize social distance or similarity between individuals or groups. **Geographic Information Systems (GIS):** MDS is utilized in GIS applications to visualize and analyze spatial relationships. It can be used to create maps or visualizations of geographic data based on the perceived similarities or dissimilarities between locations, such as in crime mapping or transportation planning. **Image and Pattern Recognition**: MDS is employed in computer vision and pattern recognition tasks. It helps visualize and analyze similarities or dissimilarities between images or patterns, facilitating tasks like image retrieval, object recognition, and clustering. **Marketing Research**: MDS is used in marketing research to understand and visualize consumer preferences and perceptions. It helps businesses identify market segments, study brand associations, and analyze customer satisfaction. **Environmental Science**: MDS is applied in environmental science to analyze and visualize similarities or dissimilarities between ecological communities, habitats, or species. It aids in studying biodiversity, species distributions, and ecological relationships. **Human-Computer Interaction**: MDS is utilized in human-computer interaction (HCI) research to understand user preferences, usability evaluations, and interface design. It helps designers and researchers map user perceptions and preferences in a lower-dimensional space.	**[L1][CO5]**	**[6M]**
6		**State and explain various non-parametric estimation techniques?** Non-parametric Density Estimations: Similar inputs have similar outputs. These are also called instance-based or memory-based learning algorithms. There are 4 Non – parametric density estimation methods: • Histogram Estimator • Naive Estimator • Kernel Density Estimator (KDE) • KNN estimator (K – Nearest Neighbor Estimator)	**[L1][CO5]**	**[12M]**

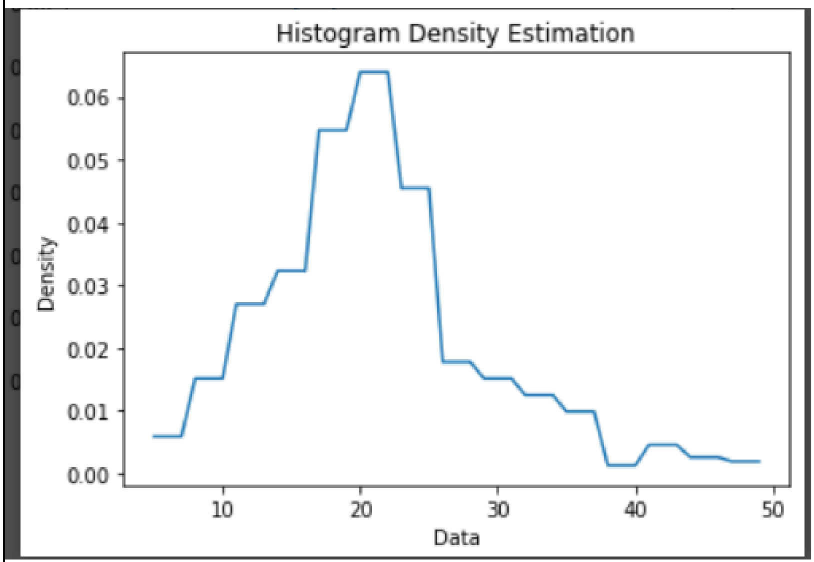
Histogram Estimator

It is the oldest and the most popular method used to estimate the density, where the input space is divided into equal-sized intervals called **bins**. Given the training set X = {xt}N t=1 an origin x0 and the bin width h, the histogram density estimator function is:

$$\hat{p}(x) = \frac{\#\{x^t \text{ in the same bin as } x\}}{Nh}$$

Histogram estimator

The density of a sample is dependent on the number of training samples present in that bin. In constructing the histogram of densities we choose the origin and the bin width, the position of origin affects the estimation near the boundaries.



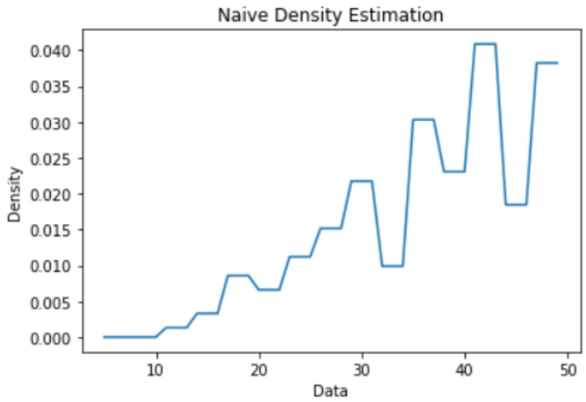
Naive Estimator

Unlike the Histogram estimator, the Naive estimator does not use the concept of origin. There is no assumption of choosing the origin. The density of the sample depends on the neighboring training samples. Given the training set X = {xt}Nt=1 and the bin width h, the Naive density estimator function is:

$$\hat{p}(x) = \frac{\#\{x - h/2 < x^t \le x + h/2\}}{Nh}$$

Naive estimator

The values in the range of h/2 to the left and right of the sample involve the density contribution.



Naive Density Estimation plot

Kernel Density Estimator (KDE)

Kernel estimator is used to smoothen the probability distribution function (pdf) and cumulative distribution function (CDF) graphics. The kernel is nothing but a weight. Gaussian Kernel is the most popular kernel:

$$K(u) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{u^2}{2}\right]$$

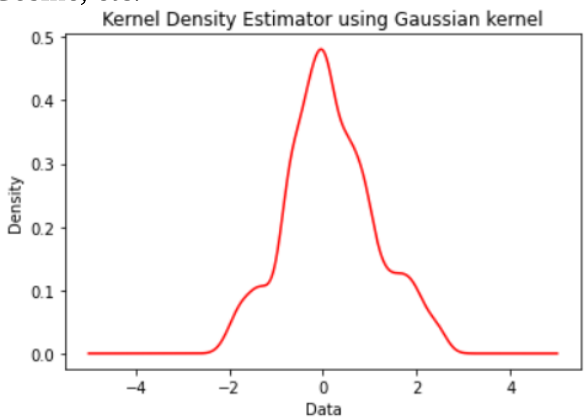
Gaussian kernel

The kernel estimator is also called Parzen Window:

$$\hat{p}(x) = \frac{1}{Nh} \sum_{t=1}^{N} K\left(\frac{x - x^t}{h}\right)$$

Kernel density estimator

As you can observe, as |x – xt| increases that means, the training sample is far away from the given sample, and the kernel value decreases. Hence we can say that the contribution of a farther sample is less when compared to the nearest training samples. There are many more kernels: Gaussian, Rectangular, Triangular, Biweight, Uniform, Cosine, etc.

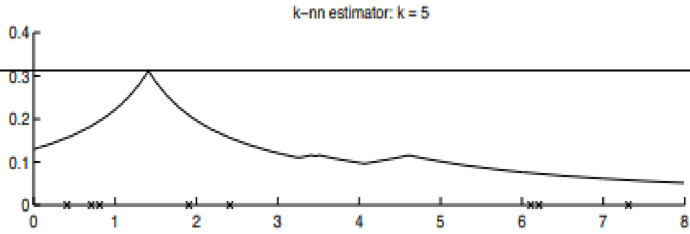


KDE plot using Gaussian Kernel

K – Nearest Neighbor Estimator (KNN Estimator)

Unlike the previous methods of fixing the bin width h, in this estimation, we fix the value of nearest neighbors k. The density of a sample depends on the value of k and the distance of the kth nearest neighbor from the sample. This is close enough to the Kernel estimation method. The K-NN density estimation is, where dk(x) is the Euclidean distance from the sample to its kth nearest neighbor.

<table>
<tr><td></td><td></td><td>



KNN Estimator

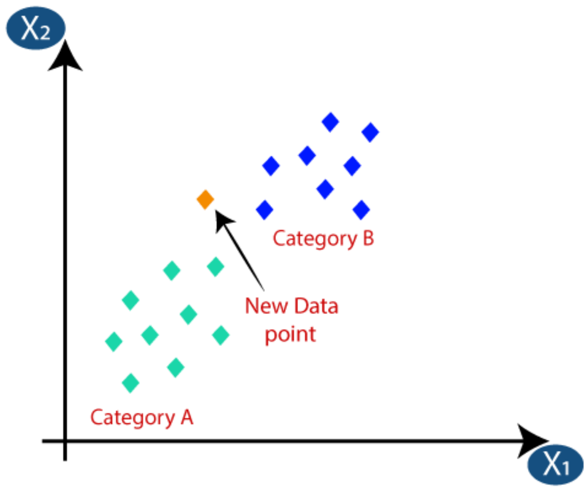
</td><td></td><td></td></tr>
<tr><td>7</td><td>a</td><td>

Analyze the K-Nearest Neighbor estimator with simple example.

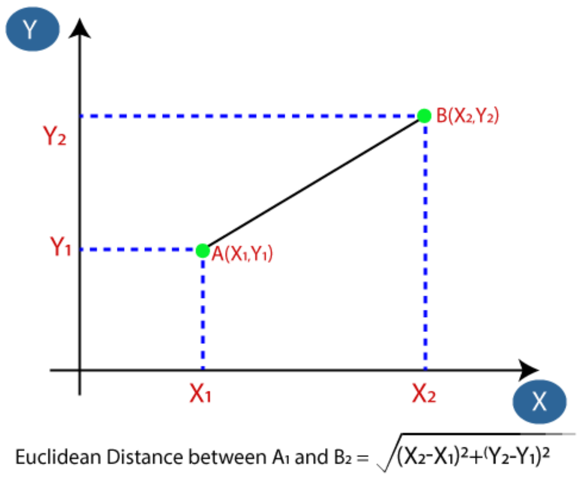
The K-NN working can be explained on the basis of the below algorithm:

- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Our model is ready.

Suppose we have a new data point and we need to put it in the required category. Consider the below image:



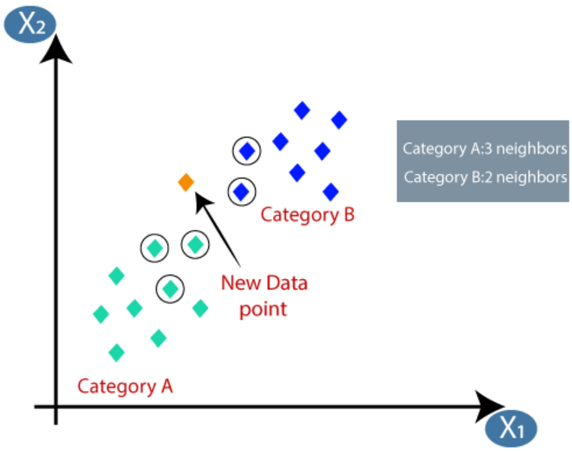
- Firstly, we will choose the number of neighbors, so we will choose the k=5.
- Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:



</td><td>[L4][CO5]</td><td>[6M]</td></tr>
</table>

- By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image:

- As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.



How to select the value of K in the K-NN Algorithm?

Below are some points to remember while selecting the value of K in the K-NN algorithm:

- There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
- A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
- Large values for K are good, but it may find some difficulties.

Advantages of KNN Algorithm:

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

Disadvantages of KNN Algorithm:

- Always needs to determine the value of K which may be complex some time.

The computation cost is high because of calculating the distance between the data points for all the training samples.

b	**Express the non-parametric classification techniques?**	**[L6][CO5]**	**[6M]**

Nonparametric classification techniques in machine learning are algorithms that do not make explicit assumptions about the functional form or distribution of the underlying data. These methods are flexible and can be powerful in situations where the data may not adhere to specific parametric assumptions. Here are some common nonparametric classification techniques:

K-Nearest Neighbors (KNN): KNN is a simple and intuitive algorithm that classifies a data point based on the majority class of its nearest neighbors. It does not assume any specific form for the decision boundary and can handle complex decision boundaries. KNN is often used for both binary and multiclass classification problems.

Decision Trees: Decision trees recursively partition the data based on different features to create a hierarchical structure of if-else rules. They can handle both categorical and numerical data and are capable of capturing non-linear relationships and interactions between features.

Random Forests: Random forests are an ensemble learning method that combines multiple decision trees. Each tree is trained on a different subset of the data using bootstrap sampling, and the final prediction is determined by aggregating the predictions of individual trees. Random forests can handle high-dimensional data and are robust against overfitting.

Support Vector Machines (SVM): SVMs find an optimal hyperplane that separates the data points of different classes with the largest margin. They can utilize kernel functions to implicitly map the data into higher-dimensional feature spaces, allowing them to capture complex decision boundaries. SVMs work well for both linear and non-linear classification problems.

Neural Networks: While neural networks are often associated with parametric models, they can also be considered nonparametric depending on their architecture. Deep neural networks with multiple hidden layers have the capacity to learn complex decision boundaries and patterns in the data, making them powerful nonparametric classifiers.

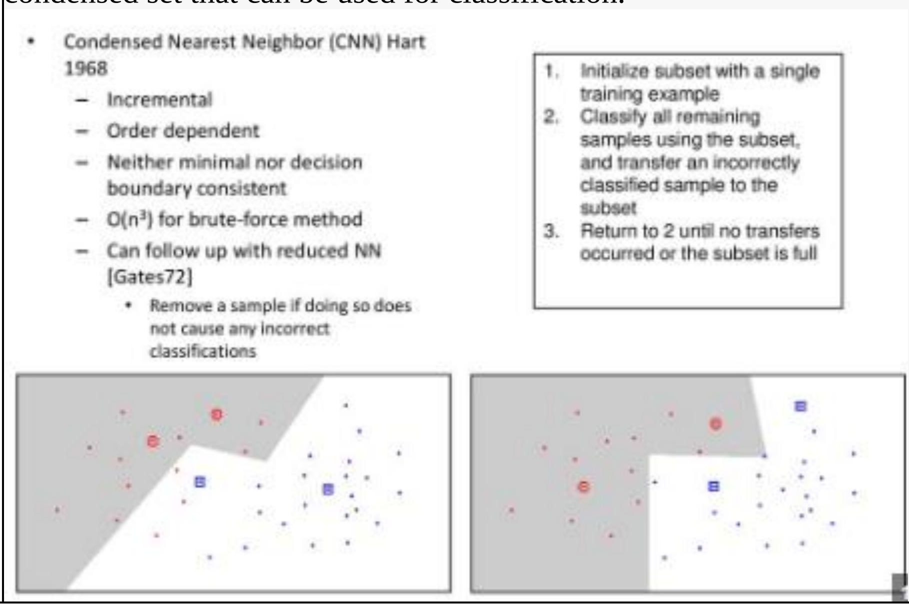
Gaussian Processes (GPs): GPs are a probabilistic approach to nonparametric classification. They model the underlying data distribution as a Gaussian process, which provides a flexible and expressive framework to make predictions. GPs can handle small to moderate-sized datasets and can provide uncertainty estimates for predictions.

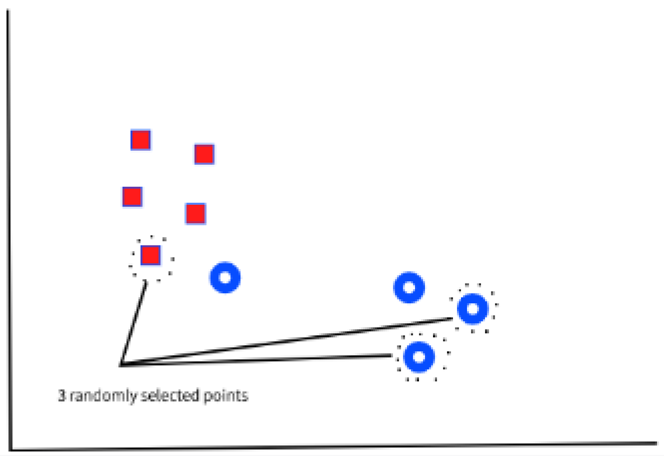
These are just a few examples of nonparametric classification techniques in machine learning. Each method has its own strengths and weaknesses, and the choice of algorithm depends on the specific problem and the characteristics of the dataset at hand.

Regenerate response

8	a	**Illustrate Condensed Nearest Neighbour(CNN)**	**[L3][CO5]**	**[6M]**

Condensed Nearest Neighbour (CNN) is a nonparametric classification technique that aims to reduce the size of the training dataset while maintaining its representativeness. It is a type of **instance-based learning algorithm** that focuses on selecting a subset of informative instances (prototypes) from the original training set to create a condensed set that can be used for classification.





The CNN algorithm follows these main steps:

Initialization: The algorithm starts with an empty set of prototypes.

Iterative process: The algorithm iteratively selects instances from the original training set and adds them to the prototype set if they are misclassified. Initially, the first misclassified instance is added to the prototype set.

Nearest Neighbor Classification: At each iteration, the misclassified instances are tested against the prototypes using a nearest neighbor classification rule. If an instance is misclassified, it is added to the prototype set.

Termination: The iterative process continues until no more misclassified instances are found or until a convergence criterion is met.

The CNN algorithm has several advantages:

Reduction of computational complexity: By selecting a condensed set of prototypes, the algorithm reduces the computational burden of classification since it only requires comparing new instances to a smaller set of prototypes instead of the entire training set.

Improved generalization: The condensed set of prototypes represents the most informative instances from the original training set. By focusing on these instances, CNN can potentially improve generalization performance and reduce overfitting.

Interpretability: The condensed set of prototypes can provide insights into the characteristics of the underlying data, as they represent the most relevant instances for classification.

However, CNN also has some limitations:

Sensitivity to initial selection: The algorithm's performance can depend on the initial selection of the first misclassified instance. Different initial instances may lead to different prototype sets and, consequently, different classification results.

Sensitivity to noisy or irrelevant instances: CNN may select noisy or irrelevant instances as prototypes, which can negatively impact classification performance.

Computational overhead during training: While CNN reduces the computational complexity during classification, the process of selecting prototypes can be computationally expensive, especially for large datasets.

		Overall, Condensed Nearest Neighbor is a useful technique for reducing the size of the training dataset while preserving classification accuracy, particularly in situations where computational efficiency and interpretability are important factors.		
	b	**Differentiate Exploratory and Confirmatory factor analysis.** **Exploratory Factor Analysis (EFA)** and Confirmatory Factor Analysis (CFA) are both techniques used in psychometrics and statistics to analyze the underlying factor structure of a set of observed variables. However, they differ in their objectives and approaches: **Exploratory Factor Analysis (EFA):** o **Objective**: EFA is used to explore and discover the latent factors that explain the relationships among a set of observed variables. It aims to identify the underlying structure and dimensions of the data. o **Hypotheses:** EFA does not rely on predefined hypotheses about the number of factors or their relationships. It allows for an open exploration of the data to uncover patterns and identify the most interpretable factor structure. o **Model Specification**: EFA is more flexible in terms of model specification. It does not require a priori specification of the factor structure and allows for the estimation of cross-loadings (variables that load on multiple factors). o **Model Fit**: EFA does not provide formal measures of model fit since it is an exploratory technique. Instead, researchers typically rely on subjective judgments, such as the interpretability of the factors and the amount of variance explained. o **Data Usage**: EFA can be used as an initial step to understand the structure of the data, generate hypotheses, and guide the development of measurement instruments or further research. **Confirmatory Factor Analysis (CFA):** o **Objective**: CFA is used to test and confirm a specific hypothesized factor structure that is derived from theory or prior research. It aims to assess how well the observed data fit the predefined factor model. o **Hypotheses**: CFA relies on specific a priori hypotheses about the number of factors, their relationships, and the loading patterns of variables on those factors. o **Model Specification**: CFA requires researchers to specify the factor structure in advance. It involves specifying the factor loadings, factor correlations, and potential measurement errors. o **Model Fit**: CFA provides formal measures of model fit, such as chi-square test, comparative fit index (CFI), root mean square error of approximation (RMSEA), etc. These measures evaluate how well the observed data fit the hypothesized factor model. o **Data Usage**: CFA is typically used to assess the adequacy of a hypothesized factor structure, validate measurement instruments, and test specific theoretical constructs. In summary, EFA is an exploratory technique used for data exploration and **hypothesis generation**, whereas CFA is a confirmatory technique used for **hypothesis testing** and **model validation**. EFA allows for more flexibility in model specification and does not require predefined hypotheses, while CFA relies on specific hypotheses and requires prior specification of the factor structure.	**[L5][CO5]**	**[6M]**

9	a	**Distinguish between parametric and non-parametric classifications.**	**[L4][CO5]**	**[6M]**

Difference between Parametric and Non-Parametric Methods are as follows:

Parametric Methods	Non-Parametric Methods
Parametric Methods uses a fixed number of parameters to build the model.	Non-Parametric Methods use the flexible number of parameters to build the model.
Parametric analysis is to test group means.	A non-parametric analysis is to test medians.
It is applicable only for variables.	It is applicable for both – Variable and Attribute.
It always considers strong assumptions about data.	It generally fewer assumptions about data.
Parametric Methods require lesser data than Non-Parametric Methods.	Non-Parametric Methods requires much more data than Parametric Methods.
Parametric methods assumed to be a normal distribution.	There is no assumed distribution in non-parametric methods.
Parametric data handles – Intervals data or ratio data.	But non-parametric methods handle original data.
Here when we use parametric methods then the result or outputs generated can be easily affected by outliers.	When we use non-parametric methods then the result or outputs generated cannot be seriously affected by outliers.
Parametric Methods can perform well in many situations but its performance is at peak (top) when the spread of each group is different.	Similarly, Non-Parametric Methods can perform well in many situations but its performance is at peak (top) when the spread of each group is the same.
Parametric methods have more statistical power than Non-Parametric methods.	Non-parametric methods have less statistical power than Parametric methods.
As far as the computation is considered these methods are computationally faster than	As far as the computation is considered these methods are computationally slower than the

		<table><tr><td>the Non-Parametric methods.</td><td>Parametric methods.</td></tr><tr><td>Examples: Logistic Regression, Naïve Bayes Model, etc.</td><td>Examples: KNN, Decision Tree Model, etc.</td></tr></table>		
	b	**Define and Explain about non-parametric methods?** Algorithms that do not make strong assumptions about the form of the mapping function are called nonparametric machine learning algorithms. By not making assumptions, they are free to learn any functional form from the training data. Nonparametric methods are good when you have a lot of data and no prior knowledge, and when you don't want to worry too much about choosing just the right features . Nonparametric methods seek to best fit the training data in constructing the mapping function, whilst maintaining some ability to generalize to unseen data. As such, they are able to fit a large number of functional forms. Some more examples of popular nonparametric machine learning algorithms are: • k-Nearest Neighbors • Decision Trees like CART and C4.5 • Support Vector Machines Benefits of Nonparametric Machine Learning Algorithms: **Flexibility**: Capable of fitting a large number of functional forms. **Power**: No assumptions (or weak assumptions) about the underlying function. **Performance**: Can result in higher performance models for prediction. Limitations of Nonparametric Machine Learning Algorithms: **More data**: Require a lot more training data to estimate the mapping function. **Slower**: A lot slower to train as they often have far more parameters to train. **Overfitting**: More of a risk to overfit the training data and it is harder to explain why specific predictions are made.	**[L2][CO5]**	**[6M]**
10	a	**Explain in detail about the various dimensionality reduction techniques** The number of input features, variables, or columns present in a given dataset is known as dimensionality, and the process to reduce these features is called dimensionality reduction. A dataset contains a huge number of input features in various cases, which makes the predictive modeling task more complicated. Because it is very difficult to visualize or make predictions for the training dataset	**[L1][CO5]**	**[6M]**

with a high number of features, for such cases, dimensionality reduction techniques are required to use.

Dimensionality reduction technique can be defined as, *"It is a way of converting the higher dimensions dataset into lesser dimensions dataset ensuring that it provides similar information."* These techniques are widely used in machine learning for obtaining a better fit predictive model while solving the classification and regression problems.

It is commonly used in the fields that deal with high-dimensional data, such as **speech recognition, signal processing, bioinformatics, etc. It can also be used for data visualization, noise reduction, cluster analysis**, etc.

Approaches of Dimension Reduction

There are two ways to apply the dimension reduction technique, which are given below:

Feature Selection

Feature selection is the process of selecting the subset of the relevant features and leaving out the irrelevant features present in a dataset to build a model of high accuracy. In other words, it is a way of selecting the optimal features from the input dataset.

Three methods are used for the feature selection:

1. Filters Methods

In this method, the dataset is filtered, and a subset that contains only the relevant features is taken. Some common techniques of filters method are:

- **Correlation**
- **Chi-Square Test**
- **ANOVA**
- **Information Gain, etc.**

2. Wrappers Methods

The wrapper method has the same goal as the filter method, but it takes a machine learning model for its evaluation. In this method, some features are fed to the ML model, and evaluate the performance. The performance decides whether to add those features or remove to increase the accuracy of the model. This method is more accurate than the filtering method but complex to work. Some common techniques of wrapper methods are:

- Forward Selection((explain with example available in previous question) refer 10 (b) answer

- Backward Selection(explain with example available in previous question) refer 10(b) answer
- Bi-directional Elimination

3. Embedded Methods: Embedded methods check the different training iterations of the machine learning model and evaluate the importance of each feature. Some common techniques of Embedded methods are:

- **LASSO**
- **Elastic Net**
- **Ridge Regression, etc.**

Feature Extraction:
Feature extraction is the process of transforming the space containing many dimensions into space with fewer dimensions. This approach is useful when we want to keep the whole information but use fewer resources while processing the information.

Some common feature extraction techniques are:

- Principal Component Analysis
- Linear Discriminant Analysis
- Kernel PCA
- Quadratic Discriminant Analysis

Factor Analysis
Factor analysis is a technique in which each variable is kept within a group according to the correlation with other variables, it means variables within a group can have a high correlation between themselves, but they have a low correlation with variables of other groups.

We can understand it by an example, such as if we have two variables Income and spend. These two variables have a high correlation, which means people with high income spends more, and vice versa. So, such variables are put into a group, and that group is known as the **factor**. The number of these factors will be reduced as compared to the original dimension of the dataset.

Auto-encoders

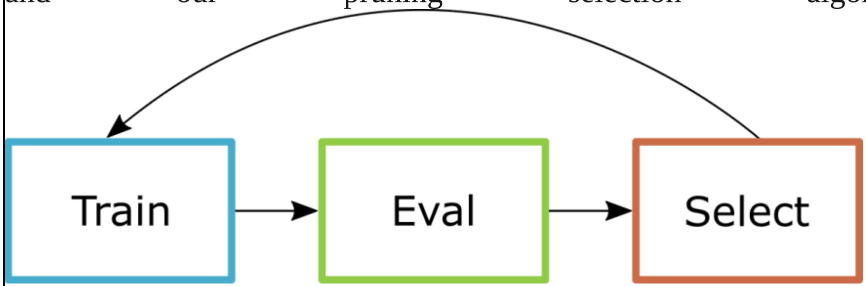
One of the popular methods of dimensionality reduction is auto-encoder, which is a type of ANN or artificial neural network, and its main aim is to copy the inputs to their outputs. In this, the input is compressed into latent-space representation, and output is occurred using this representation. It has mainly two parts:

- **Encoder:** The function of the encoder is to compress the input to form the latent-space representation
- **Decoder:** The function of the decoder is to recreate the output from the latent-space representation.

	b	**Describe Forward selection and Backward selection in subset selection**	[L2][CO5]	[12M]

Subset selection

Subset selection is a feature selection algorithm that can variate between a **forward selection and a backward selection.** Both methods consist in finding a subset of the initial features that contain the least number of dimensions that most contribute to accuracy. A naive approach would be to try all the 2^n possible subset combinations but if the number of dimensions is too big it would take forever. Instead, based on a **heuristic** function (error function) we add or remove features. The performance of subset selection depends highly on the model we choose and our pruning selection algorithm.



Forward selection

In forward selection we start with an empty set of features, for each feature that is not in the set we train the model with it and test its performance; we then select the feature with the **least amount of error**. We continue adding new features for the model to train until the error is low enough or until we have selected a proportion of the total features.

Forward Feature Selection is a feature selection technique that iteratively builds a model by adding one feature at a time, selecting the feature that maximizes model performance.

It starts with an empty set of features and adds the most predictive feature in each iteration until a stopping criterion is met. T

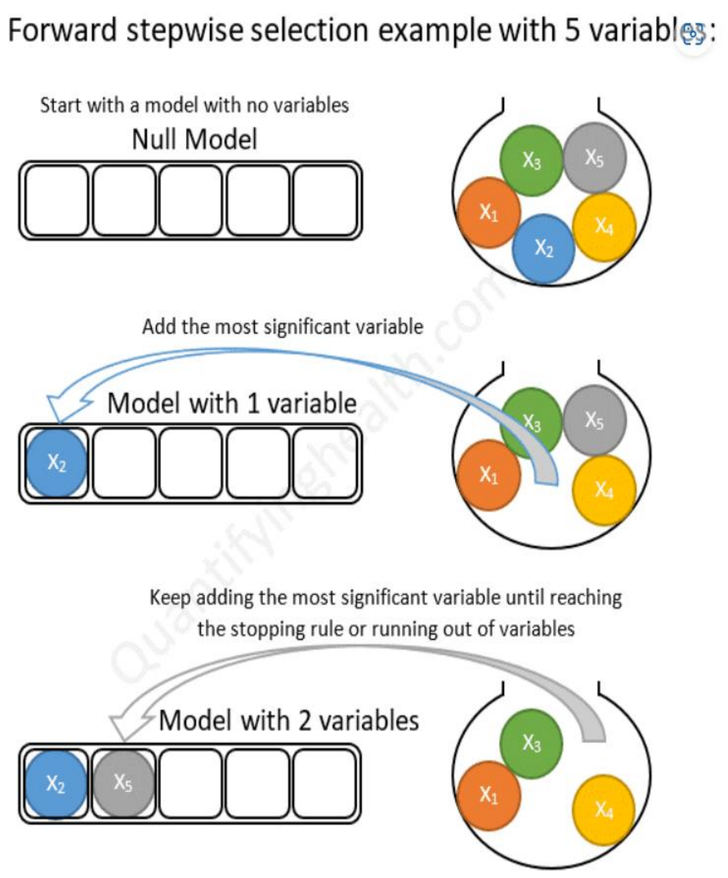
his method is particularly useful when dealing with a large number of features, as it incrementally builds the model based on the most informative features.

This process involves assessing new features, evaluating combinations of features, and selecting the optimal subset of features that best contribute to model accuracy.

Forward stepwise selection (or **forward selection**) is a variable selection method which:

4. **Begins** with a model that contains no variables (called the *Null Model*)
5. **Then** starts adding the most significant variables one after the other
6. **Until** a pre-specified stopping rule is reached or until all the variables under consideration are included in the model

Here’s an example of forward selection with 5 variables:

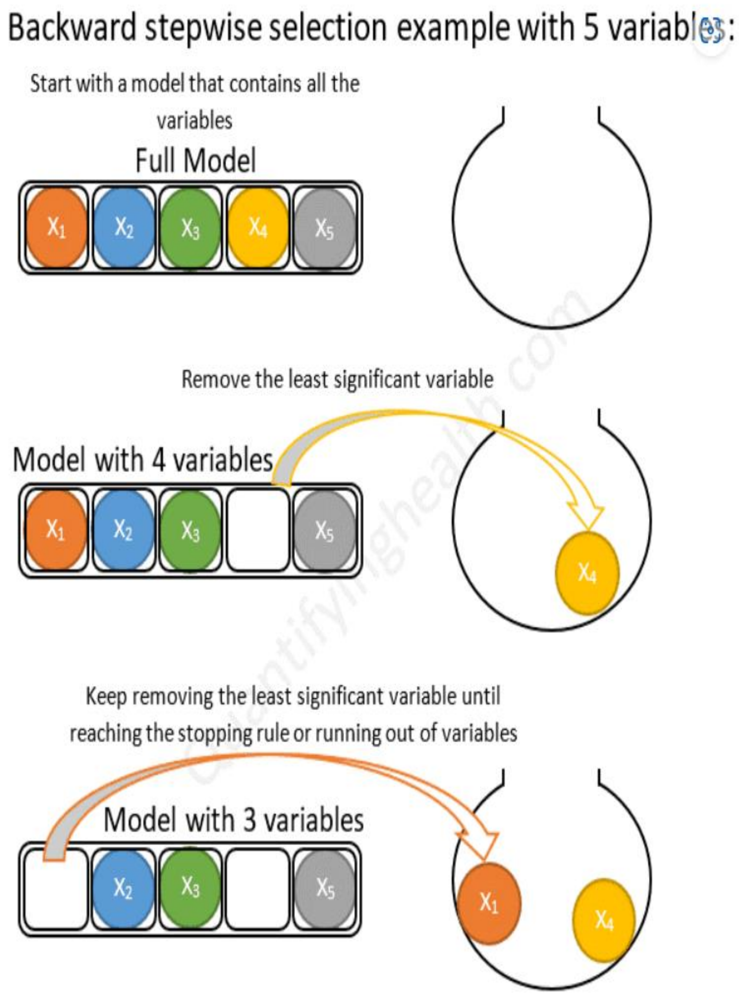


Backward stepwise

Backward stepwise selection (or **backward elimination**) is a variable selection method which:

4. **Begins** with a model that contains all variables under consideration (called the *Full Model*)
5. **Then** starts removing the least significant variables one after the other
6. **Until** a pre-specified stopping rule is reached or until no variable is left in the model

Here's an example of backward elimination with 5 variables:


Backward stepwise selection example with 5 variables:
Start with a model that contains all the variables
Full Model
X1 X2 X3 X4 X5
Remove the least significant variable
Model with 4 variables
X1 X2 X3 X5
X4
Keep removing the least significant variable until reaching the stopping rule or running out of variables
Model with 3 variables
X2 X3 X5
X1 X4

UNIT –V

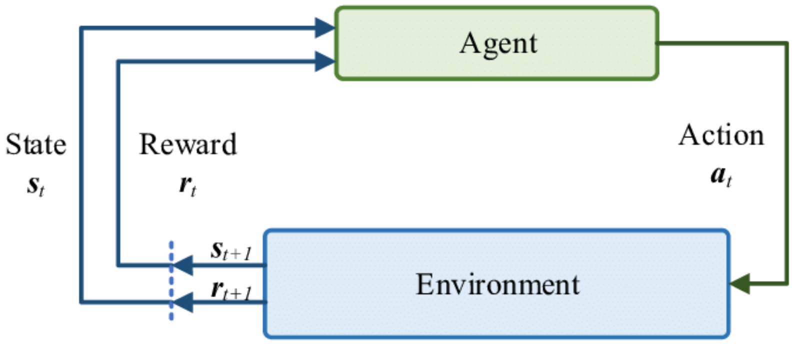
REINFORCEMENT LEARNING

1	a	**Define and explain about the Reinforcement learning.**	[L2][CO6]	[6M]

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 labelled data, unlike supervised learning.
- Since there is no labelled data, so the agent is bound to learn by its experience only.

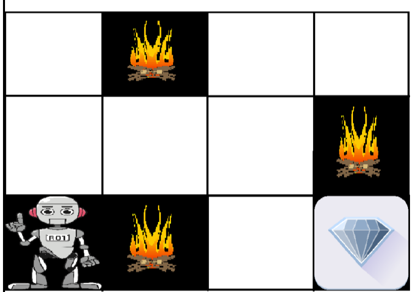
- "Reinforcement learning is a type of machine learning method where an intelligent agent (computer program) interacts with the environment and learns to act within that."



Reinforcement learning uses algorithms that learn from outcomes and decide which action to take next. After each action, the algorithm receives feedback that helps it determine whether the choice it made was correct, neutral or incorrect. It is a good technique to use for automated systems that have to make a lot of small decisions without human guidance.

Example:

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



The above image shows the robot, diamond, and fire. The goal of the robot is to get the reward that is the diamond and avoid the hurdles that are fired. The robot learns by trying all the possible paths and then choosing the path which gives him the reward with the least hurdles. Each right step will give the robot a reward and each wrong step will subtract the reward of the robot. The total reward will be calculated when it reaches the final reward that is the diamond.

Main points in Reinforcement learning –

Input: The input should be an initial state from which the model will start

Output: There are many possible outputs as there are a variety of solutions to a particular problem

Training: The training is based upon the input, The model will return a state and the user will decide to reward or punish the model based on its output.

- The model keeps continues to learn.
- The best solution is decided based on the maximum reward.

b **Compare unsupervised learning and Reinforcement learning.** **[L4][CO6]** **[6M]**

Criteria	**Unsupervised ML**	**Reinforcement ML**
Definition	Trained using unlabelled data without any guidance.	Works on interacting with the environment
Type of data	Unlabelled data	No – predefined data
Type of problems	Association and Clustering	Exploitation or Exploration
Supervision	No supervision	No supervision
Algorithms	K – Means, C – Means, Apriori	Q – Learning, SARSA
Aim	Discover underlying patterns	Learn a series of action
Application	Recommendation System, Anomaly Detection	Self Driving Cars, Gaming, Healthcare

2 a **Explain various types of reinforcement learning techniques.** **[L2][CO6]** **[6M]**

Types of Reinforcement:

There are two types of Reinforcement:

1. **Positive:** Positive Reinforcement is defined as when an event, occurs due to a particular behavior, increases the strength and the frequency of the behavior. In other words, it has a positive effect on behavior.

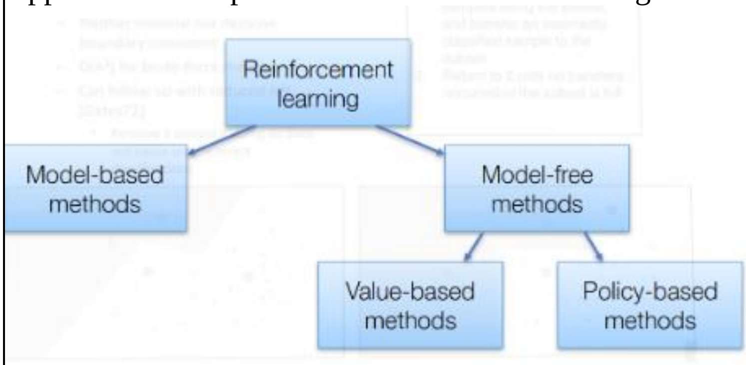
Advantages of reinforcement learning are:

- Maximizes Performance
- Sustain Change for a long period of time

- Too much Reinforcement can lead to an overload of states which can diminish the results

2. **Negative:** Negative Reinforcement is defined as strengthening of behavior because a negative condition is stopped or avoided.
   Advantages of reinforcement learning:
   - Increases Behavior
   - Provide defiance to a minimum standard of performance
   - It Only provides enough to meet up the minimum behaviour

Approaches to implement 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.

- **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.		
	b	**List out the advantages and disadvantages of Reinforcement Learning**. **Advantages of Reinforcement Learning**: 1. Flexibility and Adaptability: Reinforcement learning allows agents to adapt to changing environments and learn optimal strategies without explicitly programmed rules. It can handle complex and dynamic scenarios where traditional rule-based approaches may fail. 2. Learning from Experience: Reinforcement learning agents learn by interacting with the environment and receiving feedback in the form of rewards or punishments. This experiential learning enables agents to discover optimal policies by exploring different actions and observing their consequences. 3. Handling Uncertainty: Reinforcement learning is capable of dealing with uncertain and partially observable environments. Agents can learn to make decisions based on probabilistic models, effectively managing uncertainty and making near-optimal decisions. 4. Generalization: Reinforcement learning algorithms can generalize knowledge learned from one task or environment to new, unseen situations. This ability to transfer knowledge allows agents to apply learned policies to similar problems, reducing the need for retraining from scratch. 5. Autonomous Decision Making: Reinforcement learning enables autonomous decision making without the need for human intervention. This is particularly useful in domains where human expertise is limited or costly to acquire. **Disadvantages of Reinforcement Learning**: 1.**High Sample Complexity**: Reinforcement learning often requires a large number of interactions with the environment to achieve good performance. The agent must explore and gather sufficient data to learn effective policies, which can be time-consuming and inefficient in domains with lengthy feedback cycles or high-dimensional state spaces. 2.**Exploration-Exploitation Trade-off**: Finding an optimal policy requires a balance between exploration (trying new actions to learn) and exploitation (taking the best-known actions to maximize rewards). Striking the right balance can be challenging, as excessive exploration can hinder performance, while exploitation alone may lead to suboptimal solutions. 3.**Reward Design**: Designing suitable reward functions that guide the learning process is a crucial aspect of reinforcement learning. The reward signal should effectively capture the desired behaviour and provide clear guidance to the agent. However, designing appropriate reward functions can be complex and subjective, leading to biases or unintended consequences 4**. Lack of Safety**: Reinforcement learning agents typically optimize for a specific objective without considering potential risks or safety	**[L1][CO6]**	**[6M]**

		concerns. If the reward signal is not carefully defined, agents may discover unintended ways to achieve high rewards that are not aligned with human values or safety requirements. 5.**Limited Explain ability**: Reinforcement learning models often lack interpretability, making it challenging to understand and explain the decision-making process. This limitation can hinder trust and acceptance, especially in critical applications where explanations are crucial, such as healthcare or finance		
3	a	**List the applications and various elements of RL and explain it** RL has numerous applications across various domains. Here are some notable applications of reinforcement learning: 1. **Game Playing**: RL has been highly successful in game-playing scenarios. For instance, AlphaGo, developed by Deep Mind, used RL to defeat world champions in the board game Go. RL has also been applied to games like chess, poker, and video games, achieving remarkable results. 2. **Robotics**: RL enables robots to learn tasks and behaviours autonomously. Robots can learn to grasp objects, walk, navigate through environments, and perform complex tasks using reinforcement learning algorithms. 3. **Autonomous Vehicles**: Reinforcement learning can be employed to train autonomous vehicles to make decisions in dynamic and uncertain environments. RL helps in tasks like lane following, collision avoidance, and efficient route planning. 4. **Resource Management**: RL can optimize resource allocation in various domains, such as energy management, traffic signal control, and inventory management. It learns to make decisions that maximize efficiency, minimize costs, or optimize performance based on feedback and rewards. 5. **Recommendation Systems**: Reinforcement learning can enhance recommendation systems by learning user preferences and making personalized recommendations. By incorporating user feedback and reinforcement signals, RL algorithms can adapt and improve the recommendations over time. 6. **Healthcare:** RL can assist in optimizing treatment plans and personalized medicine. It can learn from patient data and clinical trials to suggest appropriate interventions, drug dosages, and treatment schedules. 7. **Finance**: RL can be applied to algorithmic trading, portfolio management, and risk analysis. RL algorithms can learn to make trading decisions by analysing market data, optimizing portfolios, and adapting to changing market conditions. 8. **Industrial Control Systems**: Reinforcement learning can optimize complex industrial processes by learning control policies that maximize efficiency, reduce downtime, and minimize resource consumption. It has applications in areas like manufacturing, power systems, and chemical processes. 9. **Natural Language Processing**: RL algorithms have been used in natural language processing tasks such as dialogue systems, machine translation, and text generation. RL can improve the performance of language models by learning to generate coherent and contextually appropriate responses. 10. **Education**: Reinforcement learning can be employed in	**[L2][CO6]**	**[6M]**

adaptive learning systems and intelligent tutoring systems. It can adapt the learning experience based on the student's progress, providing personalized feedback and optimizing learning outcomes.

Elements of Reinforcement Learning

Reinforcement learning elements are as follows:

- Policy
- Reward function
- Value function
- Model of the environment

Policy: Policy defines the learning agent behavior for given time period. It is a mapping from perceived states of the environment to actions to be taken when in those states.

Reward function: Reward function is used to define a goal in a reinforcement learning problem.A reward function is a function that provides a numerical score based on the state of the environment

Value function: Value functions specify what is good in the long run. The value of a state is the total amount of reward an agent can expect to accumulate over the future, starting from that state.

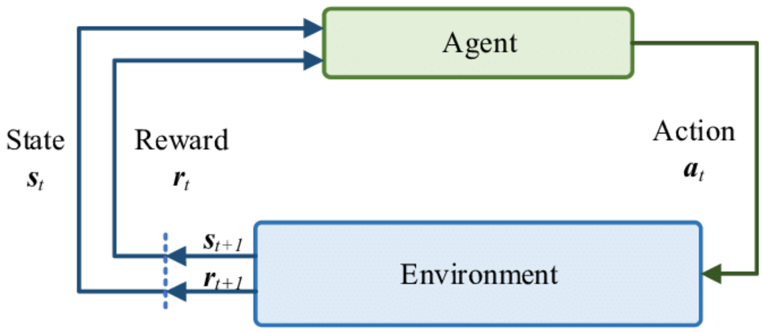
Model of the environment: Models are used for planning.

b **Differentiate the Reinforcement learning and Supervised learning.** **[L4][CO6]** **[6M]**

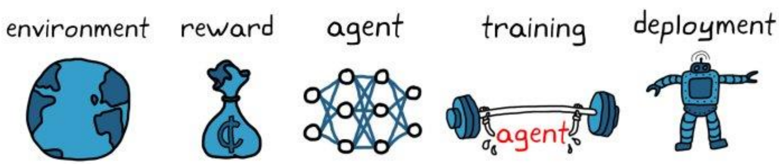
Criteria	**Supervised ML**	**Reinforcement ML**
Definition	Learns by using labelled data	Works on interacting wi environment
Type of data	Labelled data	No – predefined data
Type of problems	Regression and classification	Exploitation or Explorat
Supervision	Extra Supervision	No supervision
Algorithms	Linear Regression, Logistic Regression, SVM, KNN etc.	Q – Learning, SARSA
Aim	Calculate outcomes	Learn a series of action
Application	Risk Evaluation,	Self Driving Cars, Gami

			Forecast Sales		
4		**Analyze the working process of Reinforcement learning**		**[L4][CO6]**	**[12M]**

- 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 labelled data, unlike supervised learning.
- Since there is no labelled data, so the agent is bound to learn by its experience only.
- "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."



Agent – is the sole decision-maker and learner
Environment – a physical world where an agent learns and decides the actions to be performed
Action – a list of action which an agent can perform
State – the current situation of the agent in the environment
Reward – For each selected action by agent, the environment gives a reward. It's usually a scalar value and nothing but feedback from the environment
Policy – the agent prepares strategy (decision-making) to map situations to actions.
Value Function – The value of state shows up the reward achieved starting from the state until the policy is executed
Model – Every RL agent doesn't use a model of its environment. The agent's view maps state-action pairs probability distributions over the states



Elements of Reinforcement Learning

Reinforcement learning elements are as follows:

1. Policy
2. Reward function

3. Value function
4. Model of the environment

Policy: Policy defines the learning agent behavior for given time period. It is a mapping from perceived states of the environment to actions to be taken when in those states.
Reward function: Reward function is used to define a goal in a reinforcement learning problem.A reward function is a function that provides a numerical score based on the state of the environment
Value function: Value functions specify what is good in the long run. The value of a state is the total amount of reward an agent can expect to accumulate over the future, starting from that state.

Approaches to implement Reinforcement Learning

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

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.

1. **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.
2. **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.

represent the agent state:

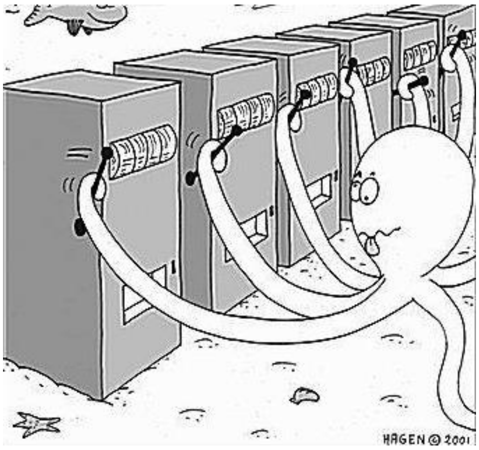
We can represent the agent state using the **Markov State** that contains all the required information from the history. The State St is Markov state if it follows the given condition:

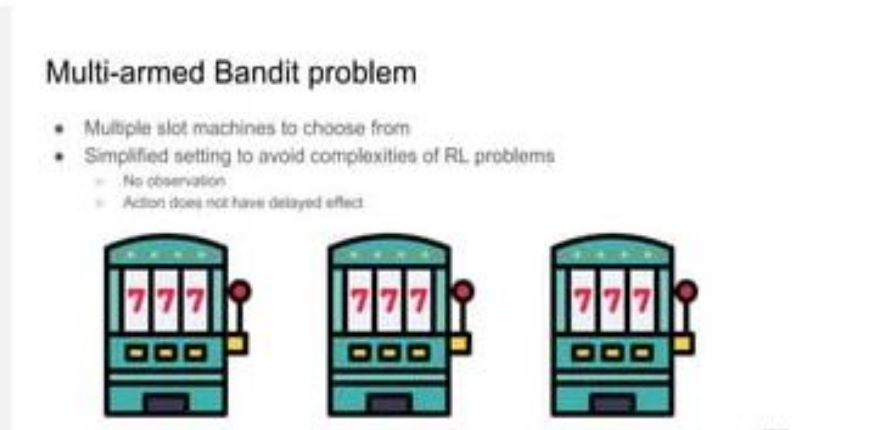
$P[S_t+1 \mid S_t] = P[S_t +1 \mid S_1,......, S_t]$

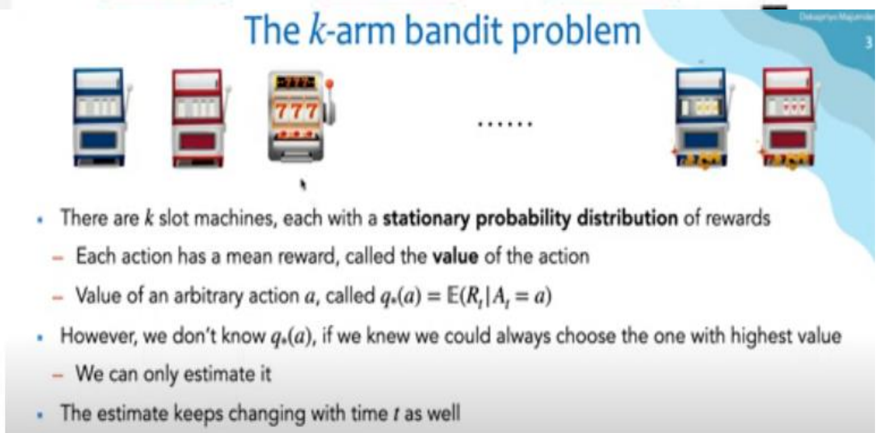
Markov Decision Process or MDP, is used to **formalize the reinforcement learning problems**. If the environment is completely observable, then its dynamic can be modeled as a **Markov Process**

		Markov Property: It says that "If the agent is present in the current state S1, performs an action a1 and move to the state s2, then the state transition from s1 to s2 only depends on the current state and future action and states do not depend on past actions, rewards, or states."		
5	a	**Explain in detail about Single State Case: K-Armed Bandit problem**	[L2][CO6]	[6M]

A bandit is defined as someone who steals your money. A one-armed bandit is a simple slot machine wherein you insert a coin into the machine, pull a lever, and get an immediate reward. But why is it called a bandit? It turns out all casinos configure these slot machines in such a way that all gamblers end up losing money!
A multi-armed bandit is a complicated slot machine wherein instead of 1 , there are several levers which a gambler can pull, with each lever giving a different return. The probability distribution for the reward corresponding to each lever is different and is unknown to the gambler.



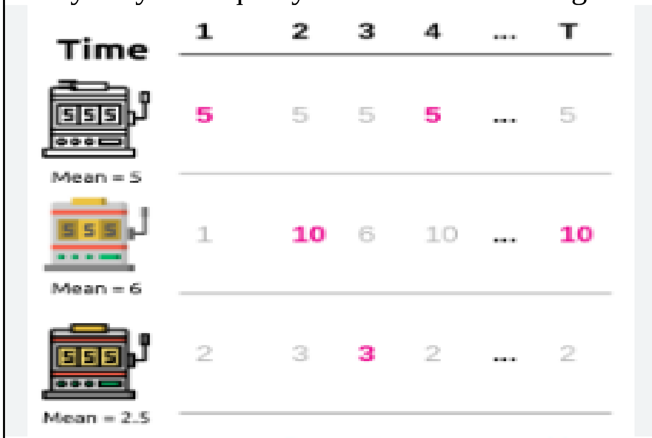


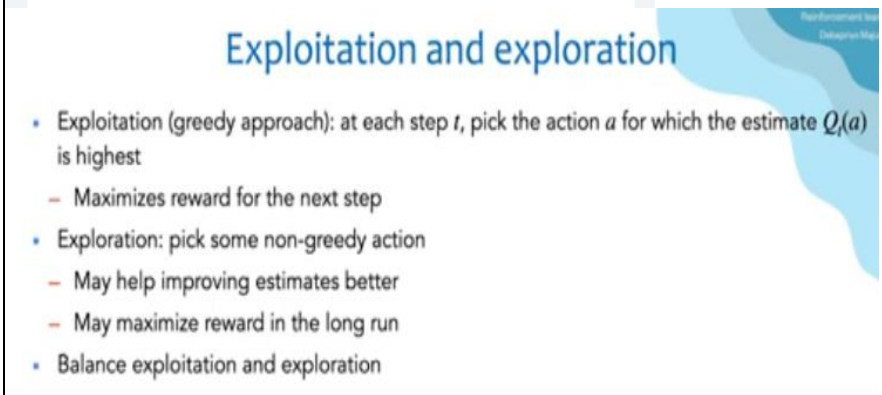


The task is to identify which lever to pull in order to get maximum reward after a given set of trials. This problem statement is like a single step Markov decision process. Each arm chosen is equivalent to an action, which then leads to an immediate reward.

There are infinite ways to build multi-armed bandit agents. Pure-exploration agents are completely random. They focus on exploration and never exploit any of the data they have gathered.

As the name suggests, pure-exploitation agents would always choose the best possible solution since they already have all the data to exploit. Being paradoxical by nature, this makes them possible in theory only and equally bad as the random agents.





There are three most popular MAB agents that are neither completely random nor impossible to deploy in practice.

Epsilon-greedy

Epsilon-greedy multi-armed bandits take care of the balance between exploration and exploitation by adding the exploration value (epsilon) to the formula. In case epsilon equals 0.3, the agent will explore random possibilities 30% of the time and focus on exploiting the best average outcome the other 70% of time.

A decay parameter is also included and it reduces epsilon over time. When constructing the agent, you may decide to remove epsilon from the equation after a certain amount of time or actions taken. This will cause the agent to focus solely on exploitation of the data it already gathered and remove random tests from the equation.

Upper confidence bound

These multi-armed bandits are quite similar to the epsilon-greedy agents. However, the key difference between the two is an additional parameter included when building upper confidence bound bandits.

A variable is included in the equation that forces the bandit to focus on the least-explored possibilities from time to time. For example, if you have options A, B, C, and D, and option D has only been chosen ten times, while the rest have been selected hundreds of times, the bandit will purposefully select D to explore the outcomes.

In essence, upper confidence bound agents sacrifice some of the resources to avoid a huge yet quite improbable mistake of never exploring the best possible outcome.

	b	**List out the various applications of Armed bandit problem explain it.**	**[L1][CO6]**	**[6M]**

The Armed Bandit Problem, also known as the Multi-Armed Bandit (MAB) problem, is a classic problem in probability theory and statistics, with extensive applications in various fields. The problem involves a scenario where a gambler must choose between multiple slot machines (bandits), each with an unknown probability of payout, to maximize their total reward over a series of trials. Here are various applications of the MAB problem along with explanations:

1. Online Advertising

In online advertising, the goal is to determine which ads (among several options) to show to users to maximize click-through rates or conversions. Each ad represents an "arm" of the bandit, and the algorithm learns over time which ads perform best.

2. Clinical Trials

In clinical trials, the MAB problem can be used to allocate patients to different treatments (arms) in a way that maximizes the overall patient benefit. This involves dynamically adjusting the allocation as more information about the effectiveness of each treatment is gathered.

3. Recommender Systems

Recommender systems, such as those used by Netflix, Amazon, or Spotify, use MAB algorithms to determine which items (movies, products, songs) to recommend to users. The goal is to recommend items that maximize user satisfaction or engagement.

4. A/B Testing

In A/B testing, businesses often need to compare multiple versions of a webpage, email, or product feature. MAB algorithms can optimize the testing process by dynamically adjusting the traffic towards the best-performing variant.

5. Portfolio Selection

In finance, portfolio selection involves choosing a set of assets to invest in. The MAB problem helps in dynamically adjusting the portfolio by learning which assets provide the best returns while balancing exploration and exploitation.

6. Dynamic Pricing

In e-commerce, dynamic pricing strategies can benefit from MAB algorithms by adjusting prices in real-time based on customer behavior and market conditions to maximize revenue or profit.

7. Adaptive Routing

In networking, adaptive routing algorithms can use MAB methods to optimize the routing of data packets through a network, choosing paths that minimize latency and maximize throughput.

8. Game Playing

In game playing, MAB algorithms are used to dynamically adjust strategies based on the performance of different moves or actions, such as in computer Go or poker.

9. Resource Allocation

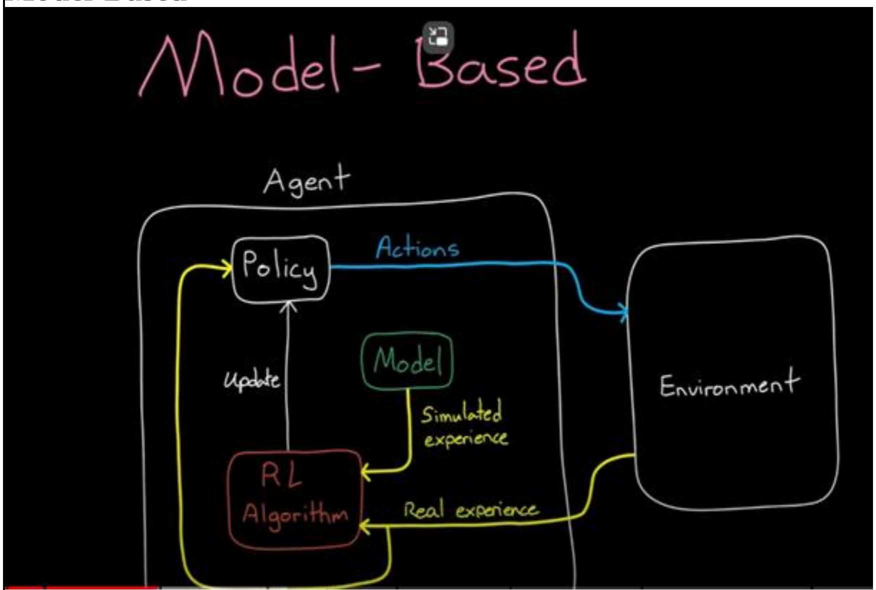
In various operational settings, such as allocating computing resources in cloud environments, MAB algorithms help in optimizing the allocation to different tasks to improve overall system performance.

10. Personalized Learning

In educational technologies, MAB can be used to personalize learning experiences by selecting the most effective teaching materials or activities for individual students to maximize learning outcomes.

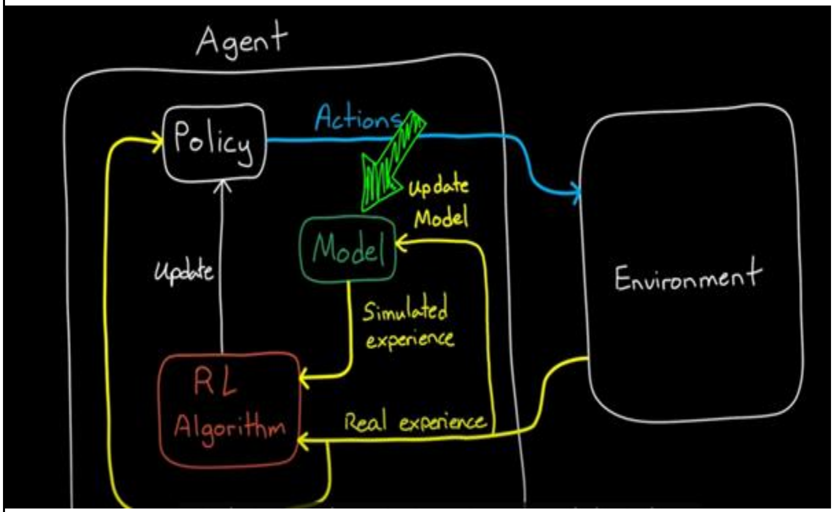
6	a	**Explain Model-Based Learning with an example.**	**[L2][CO6]**	**[6M]**

Model-Based

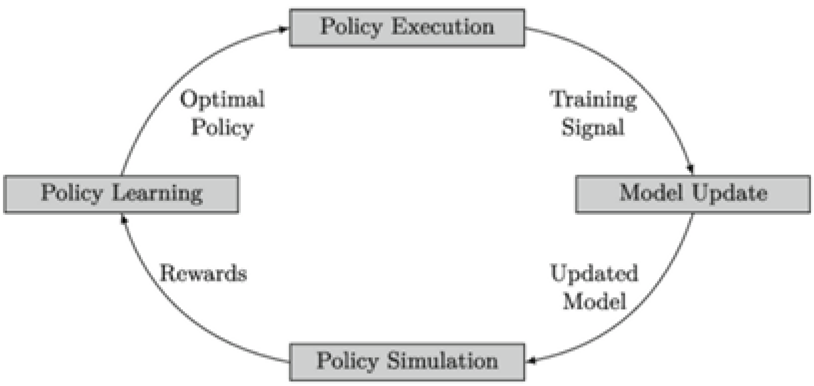


- In model-based RL the agent tries to understand how the environment is generating outcomes and rewards. The idea is to understand how the environment produces the outcomes that it produces to develop a 'model' that can simulate the environment.
- This model is used to simulate possible future state s' and outcomes, allowing the agent to plan and make decisions based on these simulations.
- Here the agent can estimate the reward of the action

beforehand without interacting with the environment as it now has a model or a simulator that behaves like the environment.



- Ultimately the model learns the transition probability (probability of going from one state to another state and then to another state) and which transition produces good rewards.
- For example, consider an agent interacting with a computer chess. Here the agent can try to learn that if I move a particular piece on the chess board what could be the response of my opponent. Based on its interaction, the agent will try to build a model that would have learned all the strategies and nuances of playing a game of chess from start to finish.
- Example Dynamic Programming Policy Evaluation



A flow diagram of model-based RL

Advantages of Model-Based Learning
Faster predictions: Model-based learning is typically faster than instance-based learning
because the model is already created and can be used to make predictions quickly.

More accurate predictions: Model-based learning can often make more accurate predictions

		than instance-based learning because the model is trained on a large dataset and can generalize to new data. Better understanding of data Model-based learning allows you to gain a better understanding of the relationships between input and output variables. This can help identify which variables are most important in making predictions. **Generalization**: Models can capture underlying patterns and make predictions on unseen data. **Efficiency**: Once the model is trained, making predictions for new instances is usually fast. **Interpretability**: Depending on the model, it is possible to gain insights into the relationships between features and predictions. **Disadvantages of Model-Based Learning** Requires a large dataset: model-based learning requires a large dataset to train the model. This can be a disadvantage if you have a small dataset. **Requires expert knowledge:** Model-based learning requires expert knowledge of statistical algorithms and mathematical modeling. This can be a disadvantage if you don't have the expertise to create the model. **Overfitting**: Models can become too complex and fit noise in the training data. May struggle with complex or non-linear relationships without appropriate model choices. May require careful feature engineering to perform well.		
	b	**Interpret the Applications of Model based Learning.** Model-based learning is a branch of machine learning where a model is constructed to understand the underlying process that generates the data. This model is then used to make predictions or decisions. Here are various applications of model-based learning, along with interpretations of how it is used in each context: **1. Robotics** **Application**: In robotics, model-based learning is used to create models of the robot's environment and its own dynamics. **Interpretation**: By understanding how its actions affect its state and the environment, a robot can plan and execute more effective actions. For instance, a robot vacuum can build a map of a room to clean more efficiently. **2. Autonomous Vehicles** **Application**: Autonomous vehicles use model-based learning to understand and predict the behavior of other vehicles and pedestrians. **Interpretation**: By modeling the dynamics of the vehicle and the road environment, these systems can make more informed decisions about navigation, collision avoidance, and path planning. **3. Healthcare** **Application**: In healthcare, model-based learning can be used to	**[L5][CO6]**	**[6M]**

create models of disease progression and patient response to treatments. **Interpretation**: This helps in personalizing treatment plans, predicting disease outcomes, and optimizing resource allocation. For example, a model can predict the progression of chronic diseases like diabetes based on patient data.

4. Natural Language Processing (NLP)

Application: In NLP, model-based approaches can be used to understand and generate human language. **Interpretation**: Models such as Hidden Markov Models (HMMs) or more advanced neural networks can capture the probabilistic structure of language, aiding in tasks like speech recognition, machine translation, and text generation.

5. Economic Modeling

Application: Model-based learning is extensively used in economics to model and predict market trends, consumer behavior, and economic outcomes. **Interpretation**: These models help in understanding complex economic systems and making policy decisions. For instance, a model might predict the impact of interest rate changes on inflation and employment.

6. Environmental Science

Application: Environmental scientists use model-based learning to predict climate change impacts, weather patterns, and ecological dynamics. **Interpretation**: By modeling the interactions between different environmental factors, these models can provide valuable insights for conservation efforts, disaster preparedness, and policy-making.

7. Manufacturing

Application: In manufacturing, model-based learning helps in process optimization and predictive maintenance. **Interpretation**: Models can predict when a machine is likely to fail, allowing for preemptive maintenance, or optimize production processes to improve efficiency and reduce waste.

8. Finance

Application: Financial markets use model-based learning for risk management, algorithmic trading, and credit scoring. **Interpretation**: By modeling the relationships between various financial instruments and market conditions, these models can predict price movements, assess credit risk, and optimize trading strategies.

9. Game AI

Application: In the development of artificial intelligence for games, model-based learning helps in creating intelligent and adaptive behaviors in non-player characters (NPCs). **Interpretation**: By modeling the game environment and possible actions, game AI can

		make strategic decisions, providing a more challenging and realistic gaming experience. **10. Energy Systems** **Application**: Model-based learning is used in energy systems for demand forecasting, grid management, and renewable energy integration. **Interpretation**: Models predict energy demand and supply, allowing for better management of resources and integration of renewable energy sources into the grid. **11. Supply Chain Optimization** **Application**: In supply chain management, model-based learning helps in optimizing logistics, inventory management, and demand forecasting. **Interpretation**: By modeling the supply chain dynamics, businesses can reduce costs, improve delivery times, and better match supply with demand.		
7		**Illustrate about Temporal Difference Learning(TDL) and its applications.** **Temporal difference learning** Temporal Difference Learning in reinforcement learning is an unsupervised learning technique that is very commonly used in it for the **purpose of predicting the total reward expected over the future.** They can, however, be **used to predict other quantities** as well. It is essentially a way to learn how to predict a quantity that is dependent on the future values of a given signal. It is a method that is used to compute the long-term utility of a pattern of behaviour from a series of intermediate rewards. Essentially, **Temporal Difference Learning (TD Learning) focuses on predicting a variable's future value in a sequence of states**. Temporal difference learning was a major breakthrough in solving the problem of reward prediction. You could say that it employs a mathematical trick that allows it to replace complicated reasoning with a simple learning procedure that can be used to generate the very same results. The trick is that rather than attempting to calculate the total future reward, temporal difference learning just attempts to predict the combination of immediate reward and its own reward prediction at the next moment in time. Now when the next moment comes and brings fresh information with it, the **new prediction is compared with the expected prediction**. If these two predictions are different from each other, the Temporal Difference Learning algorithm will	**[L3][CO6]**	**[12M]**

calculate how different the predictions are from each other and make **use of this temporal difference to adjust the old prediction toward the new prediction.**

*TEMPORAL DIFFERENCE LEARNING:

→ takes advantage of both monte carlo (mc) ideas and dynamic Programming (DP)

– Combination of mc and DP

* monte Carlo Ideas: (mc)

learns directly from raw experience i.e without model

– There is no predefined model

* Dynamic Programming (DP):

It estimates based on past of learning rather than waiting for the final outcome

→ Relation between TD, DP and mc methods are always cyclic

more about temporal difference Learning:

– it is a model free learning

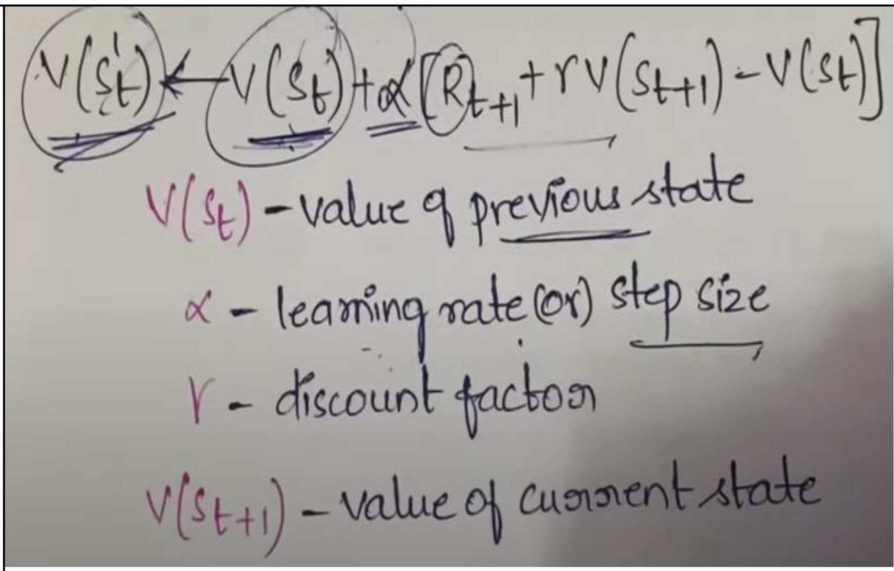
– it has 2 important properties:

– it has 2 important properties:

1. It does not require the model to be known in advance
2. It also can be applied for non-episodic tasks

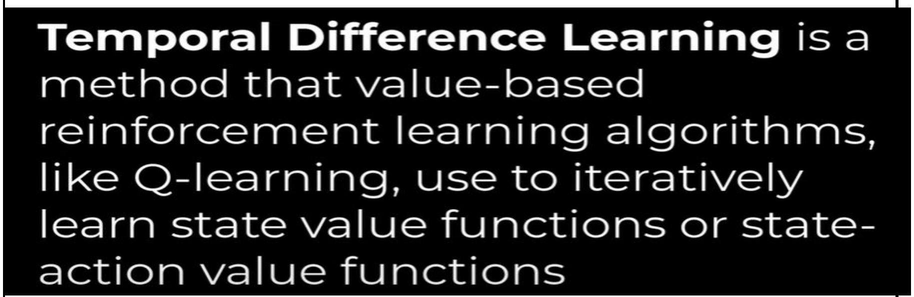
– Temporal difference learning uses an update rule for updat the value of a state

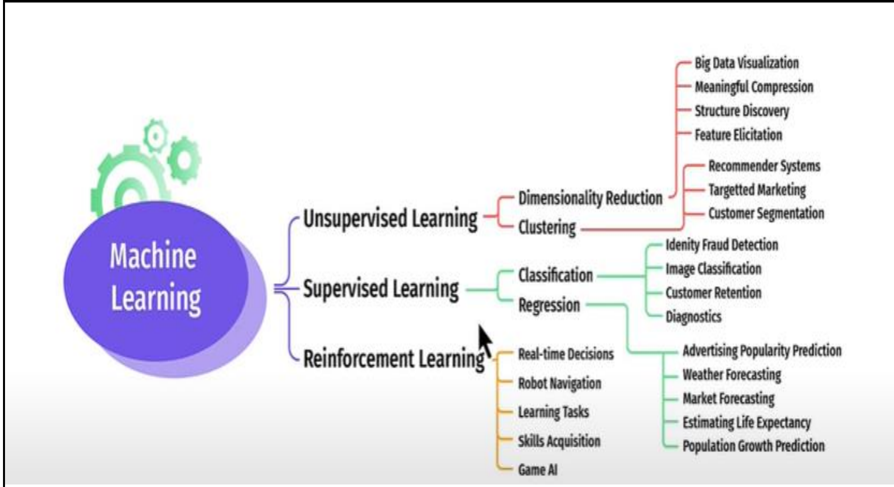
$$V(S_t) \leftarrow V(S_t) + \alpha[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$$

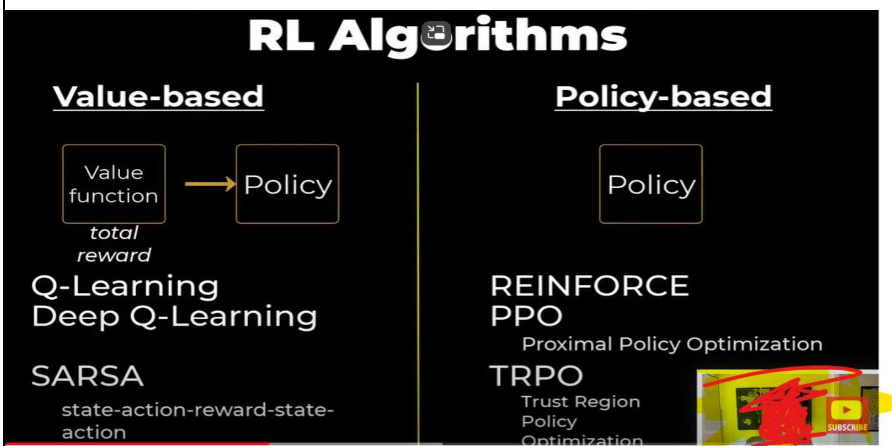


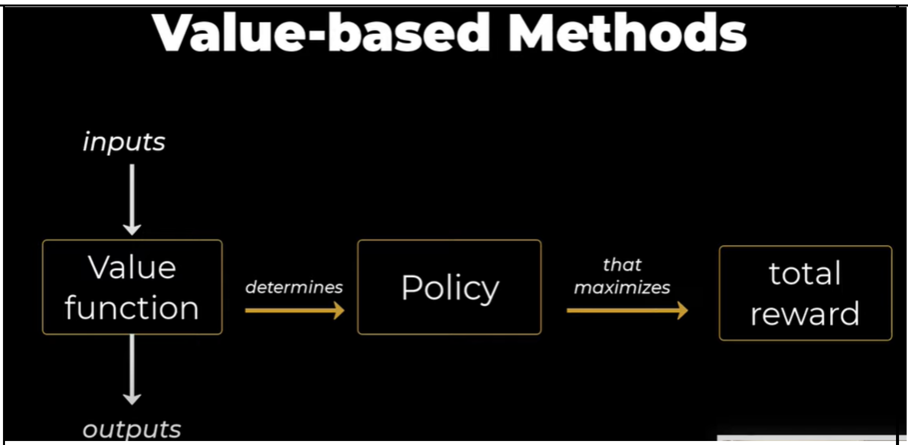
Episodic-non sequential

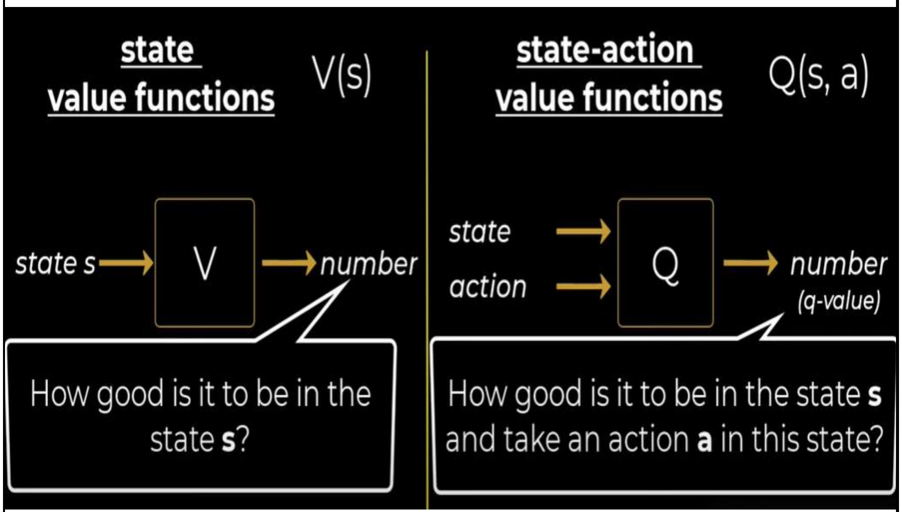
Non Episodic-sequential

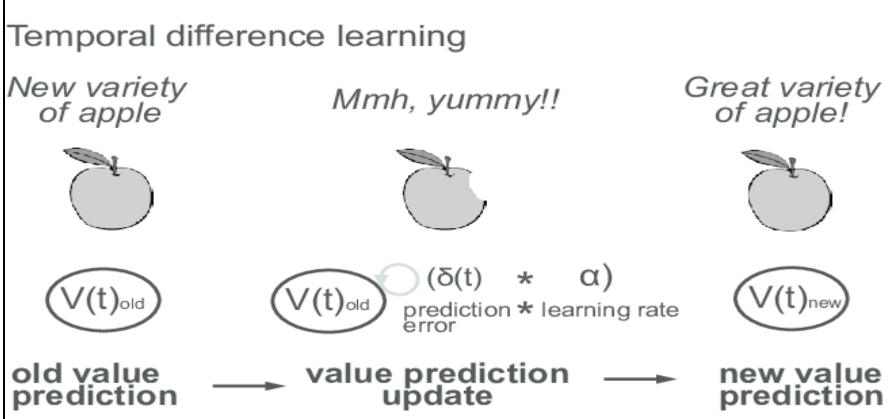












The temporal difference algorithm always aims to bring the expected prediction and the new prediction together, thus matching expectations with reality and gradually increasing the accuracy of the entire chain of prediction.

Temporal Difference Learning aims to predict a combination of the immediate reward and its own reward prediction at the next moment in time.

In TD Learning, the training signal for a prediction is a future prediction. This method is a combination of the Monte Carlo (MC) method and the Dynamic Programming (DP) method. Monte Carlo methods adjust their estimates only after the final outcome is known, but temporal difference methods tend to adjust predictions to match later, more accurate, predictions for the future, much before the final outcome is clear and know. This is essentially a type of bootstrapping.

Temporal difference learning in machine learning got its name from the way it uses changes, or differences, in predictions over successive time steps for the purpose of driving the learning process. The prediction at any particular time step gets updated to bring it nearer to the prediction of the same quantity at the next time step.

Applications:

1. **Game Playing**:
   - TD learning has been applied extensively in game playing scenarios, such as training agents to play board games like chess, Go, or video games. The agent learns to make decisions based on temporal differences between predicted and actual rewards, improving its strategy over time.
2. **Robotics and Control**:
   - In robotics, TD learning can be used for tasks like robot navigation, where the agent learns to navigate a complex environment by receiving rewards based on its actions. It helps the robot learn optimal paths and avoid obstacles.
3. **Finance and Stock Trading**:
   - TD learning is used in financial applications, especially in algorithmic trading. Agents learn to make buy/sell decisions based on temporal differences in stock prices and rewards, aiming to maximize profits over time.
4. **Recommendation Systems**:
   - TD learning is applied in recommendation systems to personalize recommendations for users. Agents learn from users' interactions (e.g., clicks, purchases) and adjust recommendations based on temporal differences in user preferences and feedback.
5. **Dynamic Pricing**:
   - TD learning is used in dynamic pricing strategies, where agents learn to set optimal prices based on temporal differences in demand, market conditions, and competitor pricing. This helps businesses maximize revenue and adapt pricing strategies in real-time.
6. **Healthcare**:
   - In healthcare, TD learning can be applied for personalized treatment recommendations. Agents learn from patient data and outcomes to adjust

treatment plans based on temporal differences in patient responses and medical conditions.

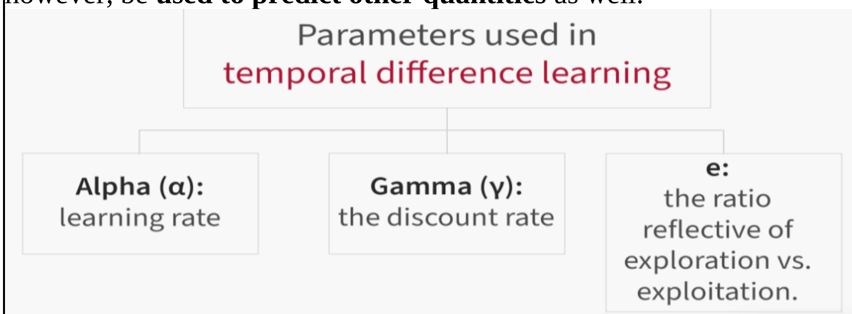
7. **Natural Language Processing (NLP)**:
   - TD learning is used in NLP tasks such as language modeling and machine translation. Agents learn to generate and understand sequences of words based on temporal differences in language patterns and context.
8. **Anomaly Detection**:
   - TD learning can be applied in anomaly detection systems where agents learn normal patterns of behavior and detect anomalies based on temporal differences in data distributions. This is useful for fraud detection, network intrusion detection, etc.

These applications demonstrate the versatility of temporal difference learning across various domains, from gaming and robotics to finance, healthcare, and natural language processing.

8	a	**Define TD and Describe various parameters used in Temporal Difference Learning.**	**[L2][CO6]**	**[6M]**

Temporal Difference Learning is an unsupervised learning technique that is very commonly used in reinforcement learning for the purpose of predicting the total reward expected over the future. Temporal Difference Learning (TD Learning) focuses on predicting a variable's future value in a sequence of states.

Temporal Difference Learning in reinforcement learning is an unsupervised learning technique that is very commonly used in it for the **purpose of predicting the total reward expected over the future.** They can, however, be **used to predict other quantities** as well.



Gamma (γ): the discount rate. A value between 0 and 1. The higher the value the less you are discounting.

Lambda (λ): the credit assignment variable. A value between 0 and 1. The higher the value the more credit you can assign to further back states and actions.

Alpha (α): the learning rate. How much of the error should we accept and therefore adjust our estimates towards. A value between 0 and 1. A higher value adjusts aggressively, accepting more of the error while a smaller one adjusts conservatively but may make more conservative moves towards the actual values.

Delta (δ): a change or difference in value.

	b	**List out the advantages, disadvantages of Temporal difference learning.**	**[L2][CO6]**	**[6M]**

Advantages of Temporal Difference Learning:

1. **Sample Efficiency**:
   - **Advantage**: TD learning can learn directly from raw experience without waiting for the end of an episode. This makes it more sample-efficient compared to Monte Carlo methods which need complete episodes to update the value function.
2. **Online Learning Capability**:
   - **Advantage**: TD methods can update their estimates at every time step, making them suitable for real-time applications where decisions need to be made continuously.
3. **Bootstrapping**:
   - **Advantage**: By bootstrapping, TD methods can effectively propagate value estimates through the state space more quickly than methods that rely on complete returns, such as Monte Carlo.
4. **Combines Model-Free and Dynamic Programming Approaches**:
   - **Advantage**: TD learning incorporates elements of both model-free reinforcement learning and dynamic programming, making it versatile and powerful in a wide range of applications.
5. **Convergence**:
   - **Advantage**: Under certain conditions, TD methods have been proven to converge to the optimal value function, especially when combined with appropriate policies and learning rates.

Disadvantages of Temporal Difference Learning:

1. **Bias-Variance Trade-off**:
   - **Disadvantage**: TD methods suffer from a bias-variance trade-off. The bootstrapping approach introduces bias, while Monte Carlo methods have higher variance but are unbiased.
2. **Stability and Convergence Issues**:
   - **Disadvantage**: The convergence of TD methods can be sensitive to the choice of learning rate and policy. Poor parameter settings can lead to instability and divergence.
3. **Complexity in Function Approximation**:
   - **Disadvantage**: When combined with function approximation techniques (like neural networks), TD learning can become complex and prone to instability, known as the deadly triad (bootstrapping, off-policy learning, and function approximation).
4. **Requires Tuning**:
   - **Disadvantage**: TD learning methods require careful tuning of hyperparameters such as the learning rate (α\alphaα) and the discount factor (γ\gammaγ), which can be challenging and time-consuming.
5. **Delayed Rewards**:
   - **Disadvantage**: TD methods might struggle with environments where rewards are significantly delayed, as the bootstrapped updates may not

		adequately capture the long-term dependencies.		
9	a	**Describe Exploration and Exploitation strategies in Machine Learning** Exploitation and Exploration in Machine Learning ***Exploitation and exploration are the key concepts in Reinforcement Learning, which help the agent to build online decision making in a better way.*** Reinforcement learning is a machine learning method in which an intelligent agent (computer program) learns to interact with the environment and take actions to maximize rewards in a specific situation. This ML method is currently being used in so many industries such as automobile, healthcare, medicine, education, etc. As in Reinforcement learning, the agent is not aware of the different states, actions for each state, associate rewards, and transition to the next state, but it learns it by exploring the environment. However, the knowledge of an agent about the state, actions, rewards, and resulting states is partial, and this results in **Exploration-Exploitation Dilemma**. In this topic, "**Exploitation and Exploration in Machine Learning**," we will discuss both these terms in detail with suitable examples. But before starting the topic, let's first understand reinforcement learning in ML. What are Exploration and Exploitation in Reinforcement Learning Before going to a brief description of exploration and exploitation in machine learning, let's first understand these terms in simple words. In reinforcement learning, whenever agents get a situation in which they have to make a difficult choice between whether to continue the same work or explore something new at a specific time, then, this situation results in Exploration-Exploitation Dilemma because the knowledge of an agent about the state, actions, rewards and resulting states is always partial. Now we will discuss exploitation and exploration in technical terms. Exploitation in Reinforcement Learning Exploitation is defined as a greedy approach in which agents try to get more rewards by using estimated value but not the actual value. So, in this technique, ***agents make the best decision based on current information.*** Exploration in Reinforcement Learning Unlike exploitation, in exploration techniques, agents primarily focus on improving their knowledge about each action instead of getting more rewards so that they can get long-term benefits. So, in this technique, ***agents work on gathering more information to make the best overall decision.***	**[L2][CO6]**	**[6M]**

Examples of Exploitation and Exploration in Machine Learning

Let's understand exploitation and exploration with some interesting real-world examples.

Coal mining:

Let's suppose people A and B are digging in a coal mine in the hope of getting a diamond inside it. Person B got success in finding the diamond before person A and walks off happily. After seeing him, person A gets a bit greedy and thinks he too might get success in finding diamond at the same place where person B was digging coal. This action performed by person A is called **greedy action**, and this policy is known as **a greedy policy.** But person A was unknown because a bigger diamond was buried in that place where he was initially digging the coal, and this greedy policy would fail in this situation.

In this example, person A only got knowledge of the place where person B was digging but had no knowledge of what lies beyond that depth. But in the actual scenario, the diamond can also be buried in the same place where he was digging initially or some completely another place. Hence, with this partial knowledge about getting more rewards, our reinforcement learning agent will be in a dilemma on whether to exploit the partial knowledge to receive some rewards or it should explore unknown actions which could result in many rewards.

However, both these techniques are not feasible simultaneously, but this issue can be resolved by using Epsilon Greedy Policy (Explained below).

There are a few other examples of Exploitation and Exploration in Machine Learning as follows:

Example 1: Let's say we have a scenario of online restaurant selection for food orders, where you have two options to select the restaurant. In the first option, you can choose your favorite restaurant from where you ordered food in the past; this is called **exploitation** because here, you only know information about a specific restaurant. And for other options, you can try a new restaurant to explore new varieties and tastes of food, and it is called exploration. However, food quality might be better in the first option, but it is also possible that it is more delicious in another restaurant.

Example 2: Suppose there is a game-playing platform where you can play chess with robots. To win this game, you have two choices either play the move that you believe is best, and for the other choice, you can play an experimental move. However, you are playing the best possible move, but who knows new move might be more strategic to win this game. Here, the first choice is called exploitation, where you know about your game strategy, and the second choice is called exploration, where you are exploring your knowledge and playing a new move to win the game.

	b	**Assess in detail about partially observables states in Reinforcement learning**.	**[L5][CO6]**	**[6M]**

		A partially observable system is one in which the entire state of the system is not fully visible to an external sensor. In a partially observable system the observer may utilize a memory system in order to add information to the observer's understanding of the system. **A fully observed state** means that there is no hidden information. Clear examples of this are chess and Go because both players have all the information. The fact that both these games are deterministic doesn't matter. A game where the state changes are stochastic can still be fully observable. Games like poker, where both players can observe their own hand but not their opponents' are called partially observable. Other examples of this can be real time strategy games like Starcraft where you can only see in the line of sight of your units. An example of a partially observable system would be a card game in which some of the cards are discarded into a pile face down. In this case the observer is only able to view their own cards and potentially those of the dealer. They are not able to view the face-down (used) cards, nor the cards that will be dealt at some stage in the future. A memory system can be used to remember the previously dealt cards that are now on the used pile. This adds to the total sum of knowledge that the observer can use to make decisions A **partially observable Markov decision process (POMDP)** is a combination of a regular Markov Decision Process to model system dynamics with a hidden Markov model that connects unobservable system states probabilistically to observations. $P=(S,A,T,R,\Omega,O,\gamma)$, Where $S=\{s1,s2,\ldots,sn\}$ is a set of partially observable states, $A=\{a1,a2,\ldots,am\}$ is a set of actions, T a set of conditional transition probabilities $T(s'\vert s,a)$ for the state transition $s\rightarrow s'$ conditioned on the taken action. $R:S\times A\rightarrow R$ is the reward function, $\Omega=\{o1,o2,\ldots,ok\}$ is a set of observations, O is a set of observation probabilities $O(o\vert s',a)$ conditioned on the reached state and the taken action, and $\gamma\in[0,1]$ is the discount factor.		
10	a	**Explain Generalization process in Model Based Learning.** In model-based learning, the generalization process refers to the ability of the learned model to make accurate predictions or simulate the behavior of the environment beyond the specific experiences it has encountered during training. Generalization allows the model to make informed decisions in novel situations and generalize its knowledge to unseen states and actions. The generalization process in model-based learning typically involves the following steps: 1. Training the Model: During the training phase, the model-based learning algorithm collects data by interacting with the environment. It observes the states, actions, and resulting rewards and uses this data to learn the dynamics of the environment. The learned model captures the transition probabilities and the expected rewards associated with different state-action pairs. 2. Model Evaluation: Once the model is trained, it needs to be evaluated to assess its predictive accuracy and generalization	**[L2][CO6]**	**[6M]**

		capabilities. The model can be tested by comparing its predictions against actual observations from the environment. This evaluation helps identify the areas where the model may require further improvement. 3. Generalization Testing: To assess the generalization capabilities of the learned model, it is exposed to novel situations or unseen states and actions that were not encountered during training. The model is used to simulate the environment's dynamics and predict the outcomes of actions in these new situations. 4. Assessing Performance: The performance of the learned model in generalization testing is evaluated by comparing its predictions or simulated outcomes with the actual observed outcomes. Metrics such as prediction accuracy, error rates, or reward accumulation can be used to quantify the model's generalization performance. 5. Iterative Refinement: If the model's generalization performance is not satisfactory, iterative refinement techniques can be applied. These techniques involve updating the model parameters, adjusting the learning algorithm, or collecting additional training data to improve the model's accuracy and generalization capabilities. By going through the generalization process, a model-based learning algorithm aims to develop a learned model that can accurately simulate the environment's dynamics, predict outcomes, and make informed decisions in novel situations. Generalization is crucial for the model to effectively transfer its learned knowledge to real-world scenarios beyond the specific training experiences.		
	b	**Difference between Model based learning and Model free learning**	**[L1][CO6]**	**[6M]**

Model-free	Model-based
1. Model free algorithms e.g. MC Control, SARSA, Q-learning rely on real samples from the environment and do not use generated predictions of next state and next reward to alter behaviour.	1. Model based algorithms like DP use the model's predictions of the next state and reward in order to calculate optimal actions.
2. Model-free approaches are based on habitual conditions and learn through trial-and-error methods.	2. Model-based approaches are well suited for goal-directed decisions and learn through planning.
3. Most state of the art algorithms use model-free RL due to availability of simulators that are able to generate huge amounts of data.	3. Model-based methods are beneficial in applications where we have strict restrictions on the sample complexity.
4. The consequences of actions are predicted by past experiences in case of model-free approach.	4. The consequences of actions are predicted by the structure of the world in case of model-based approach.
5. The values and parameters of Model-free approach change slowly over time due to iterative updating	5. Model-based approaches update its values and parameters very fast
6. Extensive experience is required by model-free approaches	6. Computational requirements are high in case of model-based approaches
7. Strong convergence is guaranteed in case of model-free model.	7. Strong convergence is not guaranteed in case of model-based models.

Prepared by: **Dr.R.M.Mallika, CSE, SIETK**