UNIT –I

INTRODUCTION MACHINE LEARNING

1) Describe about Machine Learning algorithms with their predictions. [L2][CO1] [12M]

## A) **Introduction to Machine Learning Algorithms**

Machine learning (ML) algorithms are computational methods that allow systems to learn from data and make predictions or decisions without explicit programming for each specific task. These algorithms are crucial in various applications, from predicting market trends to classifying images. Here's an overview of several prominent ML algorithms and their prediction mechanisms:

. Types of Machine Learning Algorithms

ML algorithms are typically divided into three categories:

- Supervised Learning: Uses labeled data to predict outcomes.
- Unsupervised Learning: Analyzes data without labeled responses.
- Reinforcement Learning: Focuses on making decisions based on rewards and penalties.

### 1. Linear Regression

- **Description**: Linear Regression models the relationship between a dependent variable (target) and one or more independent variables (features) by fitting a linear equation to the data. The model aims to minimize the sum of the squared differences between observed and predicted values.
- **Prediction**: This algorithm predicts continuous values. For example, it can forecast housing prices based on features like square footage and number of bedrooms.

### 2. Logistic Regression

- **Description**: Logistic Regression is used for binary classification tasks. It estimates the probability that an input belongs to a particular class using the logistic function, which maps predicted values to probabilities between 0 and 1.
- **Prediction**: It predicts categorical outcomes by providing probabilities. For instance, it can classify whether an email is spam or not.

### 3. Decision Trees

- **Description**: Decision Trees split the data into subsets based on feature values, creating a tree-like model of decisions. Each node represents a feature or attribute, and branches represent decision rules. The leaves of the tree represent class labels or continuous values.
- **Prediction**: Decision Trees are used for both classification and regression. For example, they can determine whether a customer will churn based on their behavior.

### 4. Random Forests

- **Description**: Random Forests are an ensemble method that builds multiple decision trees and aggregates their predictions. Each tree is trained on a random subset of the data with some features chosen randomly at each split, which helps to improve generalization and reduce overfitting.
- **Prediction**: It can handle both classification and regression. For instance, predicting loan defaults by combining the predictions from multiple decision trees.

### 5. Support Vector Machines (SVM)

- **Description**: SVM finds the optimal hyperplane that separates different classes in the feature space with the maximum margin. It can be extended to handle non-linear classification through kernel tricks.
- **Prediction**: Used for classification tasks, such as image classification or text categorization. It can distinguish between different categories of objects or documents.

### 6. K-Nearest Neighbors (KNN)

- **Description**: KNN is a non-parametric algorithm that classifies a data point based on the majority class among its 'k' nearest neighbors in the feature space. For regression, it predicts the value based on the average of the nearest neighbors.
- **Prediction**: It can be applied to both classification (e.g., recommending products) and regression tasks (e.g., predicting temperature).

### 7. Naive Bayes

- **Description**: Naive Bayes classifiers are based on Bayes' theorem with the assumption of independence between features. It is particularly effective for large datasets and text classification tasks.
- **Prediction**: Typically used for classification tasks, such as categorizing documents or spam filtering.

### 8. *Principal Component Analysis (PCA)*

- **Description**: PCA reduces the dimensionality of data by transforming it into a set of orthogonal components that capture the most variance.
- **Predictions**: By reducing the number of features while retaining the most important information, PCA simplifies the dataset. This can enhance the performance and efficiency of predictive models by removing noise and redundancy. For example, PCA can simplify high-dimensional data, making it easier for a regression model to identify important patterns.

### 9. K-Means Clustering

- **Description**: Groups data into clusters based on feature similarity.
- **Prediction:** The cluster assignments can be used as additional features in a supervised learning model. For example, clustering customers into groups based on purchase behavior, and then using these clusters to predict future purchases.

### 10. Neural Networks

- **Description**: Models composed of layers of interconnected nodes (neurons) that can learn complex patterns from data through training.
- **Prediction Type**: **Classification and Regression.**

**2)** Define basic concepts in Machine Learning. [L1][CO1] [12M]

### A) 1. Machine Learning

**Definition**: A field of artificial intelligence that enables systems to learn from data and improve their performance on tasks over time without being explicitly programmed for each specific task.

1. Data: Machine Learning algorithms learn from data. The data can be in the form of structured data (e.g., rows and columns in a spreadsheet) or unstructured data (e.g., text, images, audio). Data is typically split into a training set (used to train the model) and a test set (used to evaluate the model's performance).
2. Feature: A feature is a measurable attribute of the data that can be used to make predictions. For example, if we're trying to predict whether a loan will be approved, the features might include the applicant's income, credit score, and employment status.
3. Model: A model is a mathematical representation of the relationships between the features and the target variable (the variable we're trying to predict). The goal of Machine Learning is to find the best model that accurately predicts the target variable.

4. Training: During the training process, the Machine Learning algorithm adjusts the parameters of the model to minimize the error between the predicted output and the actual output. This is done by feeding the algorithm the training set of data and updating the model's parameters after each iteration.
5. Testing: After the model has been trained, it is tested on a separate test set of data to evaluate its performance. The performance is measured using various metrics such as accuracy, precision, recall, and F1 score.
6. Supervised learning: In supervised learning, the training data includes both the features and the target variable. The goal is to learn a mapping between the features and the target variable. Examples of supervised learning include classification (predicting a categorical variable) and regression (predicting a continuous variable).
7. Unsupervised learning: In unsupervised learning, the training data only includes the features. The goal is to discover patterns or relationships in the data. Examples of unsupervised learning include clustering (grouping similar data points together) and dimensionality reduction (reducing the number of features while preserving the most important information).
8. Reinforcement learning: In reinforcement learning, the algorithm learns through trial and error. The algorithm receives feedback in the form of rewards or penalties based on its actions. The goal is to learn a policy that maximizes the expected reward over time. Reinforcement learning is often used in robotics and game playing.

3) Discuss the Machine Learning techniques with neat diagrams . [L2][CO1] [12M]

A)

1. Regression: Regression is a technique used to predict continuous numerical values. Linear regression is a popular method where the model learns a linear relationship between the input features and the output.

2. Classification: Classification is a technique used to predict categorical values. The most popular algorithm for classification islogistic regression. Other popular algorithms include decision trees, random forests, and support vector machines.

3. Clustering: Clustering is a techni que used to group similar data points together based on their features. Clustering algorithms include k-means, hierarchical clustering, and DBSCAN. 4. Dimensionality reduction: Dimensionality reduction is a technique used to reduce the number of input features while preserving the most important information. Principal component analysis (PCA) is a popular method for dimensionality reduction.

5. Neural networks: Neural networks are a set of algorithms modeled after the human brain. They consist of layers of interconnected nodes and can be used for a variety of tasks such as image recognition, natural language processing, and speech recognition.

6. Deep learning: Deep learning is a subset of neural networks that uses multiple layers of nodes to learn increasingly complex representations of the data. Deep learning has been used to achieve state-of-the-art performance in tasks such as image recognition and natural language processing.

7. Reinforcement learning: Reinforcement learning is a technique used to teach machines to learn through trial and error. The algorithm receives feedback in the form of rewards or penalties based on its actions, and its goal is to learn a policy that maximizes the expected reward over time.

8. Ensemble learning: Ensemble learning is a technique that combines multiple models to improve the accuracy of predictions. Examples of ensemble methods include bagging, boosting, and stacking.

**REFOR DIAGRAMS IN CLASS WORK OR IN PREVIOUS PDF**

**4.)** Explain about Supervised Learning techniques. [L2][CO1] [12M]

A)

Supervised Learning is a type of Machine Learning where the algorithm learns to predict an output variable from input variables based on labeled data. In other words, the algorithm learns from examples where both the input and output variables are known. Here are some common techniques used in Supervised Learning:

1. Linear Regression: Linear regression is a technique used to predict a continuous output variable based on one or more input variables. The goal is to find a linear relationship between the input variables and the output variable. Linear regression can be used for both simple and multiple linear regression problems.
2. Logistic Regression: Logistic regression is a technique used to predict a categorical output variable based on one or more input variables. The goal is to find a relationship between the input variables and the probability of the output variable being in a certain category.
3. Decision Trees: Decision trees are a technique used to predict a categorical or continuous output variable based on one or more input variables. Decision trees are built by recursively splitting the data into subsets based on the values of the input variables until a stopping criterion is met.
4. Random Forests: Random forests are a type of ensemble learning technique that combines multiple decision trees to improve the accuracy of predictions. Random forests are built by constructing multiple decision trees on randomly selected subsets of the data and then averaging their predictions.
5. Support Vector Machines (SVMs): SVMs are a technique used to predict a categorical or continuous output variable based on one or more input variables. The goal is to find a hyperplane that separates the data into different classes with the largest possible margin.
6. Naive Bayes: Naive Bayes is a probabilistic technique used to predict a categorical output variable based on one or more input variables. Bayes assumes that the input variables are independent of each other and calculates the probability of the output variable based on the probabilities of the input variables.

(REFOR DIAGRAMS IN CLASS WORK)

5.) Explain the Un-Supervised Learning techniques. [L2][CO1] [12M]

A)

Unsupervised Learning is a type of Machine Learning where the algorithm learns to identify patterns and structures in the data without being explicitly trained on labelled data. In other words, the algorithm learns to find relationships and groupings in the data on its own. Here are some common techniques used in Unsupervised Learning:

1. Clustering: Clustering is a technique used to group similar data points together based on their features. The goal is to identify natural groupings in the data without prior knowledge of the group labels. Common clustering algorithms include k-means, hierarchical clustering, and DBSCAN.

2. Dimensionality Reduction: Dimensionality reduction is a technique used to reduce the number of input features while preserving the most important information. The goal is to simplify the data and make it easier to analyse. Common dimensionality reduction techniques include principal component analysis (PCA), t-distributed stochastic neighbour embedding (t-SNE), and autoencoders.

3. Anomaly Detection: Anomaly detection is a technique used to identify data points that are significantly different from the rest of the data. The goal is to detect unusual patterns or outliers in the data. Common anomaly detection techniques include density-based methods, distance- based methods, and clustering-based methods.

4. Association Rule Learning: Association rule learning is a technique used to identify relationships between variables in the data. The goal is to find patterns in the data such as frequent itemsets and association rules. Common association rule learning algorithms include Apriori and FPGrowth.

5. Generative Models: Generative models are a type of Unsupervised Learning technique that learns to generate new data samples that are similar to the input data. The goal is to learn the underlying structure of the data and use it to generate new samples. Common generative models include variational autoencoders, generative adversarial networks (GANs), and Boltzmann machines.

(REFOR DIAGRMS IN CLASS WORK)

6. a) What is the role of pre-processing of data in machine learning? Why it is needed? [L1][CO1] [6M]

A) Major steps of Data Preprocessing are:

1. Data Acquisition

2. Data Normalization/Cleaning
3. Data Formatting
4. Data Sampling
5. Data Scaling

### 1 . Data Acquisition

**Definition**: The process of collecting raw data from various sources. This is the first step in any machine learning project and is crucial for ensuring that you have the necessary data to work with.

**Processes Involved**:

- **Data Collection**: Gathering data from multiple sources such as databases, files, APIs, or web scraping.
- **Data Integration**: Combining data from different sources into a single dataset, ensuring that it is comprehensive and representative of the problem domain.

**Importance**: Proper data acquisition ensures that the dataset is relevant and sufficient for the analysis or model training. Inadequate or irrelevant data can lead to inaccurate or biased results.

### 2. Data Normalization/Cleaning

**Definition**: The process of preparing and cleaning data to ensure it is accurate, consistent, and usable for analysis. This step addresses issues such as missing values, duplicates, and inconsistencies.

**Processes Involved**:

- **Handling Missing Values**: Techniques like imputation (replacing missing values with mean, median, or mode) or removal of records with missing data.
- **Removing Duplicates**: Identifying and eliminating duplicate records to ensure data integrity.
- **Error Correction**: Identifying and correcting inaccuracies or anomalies in the data.
- **Data Cleaning**: Standardizing formats, correcting typos, and ensuring consistency in data entries.

**Importance**: Clean and normalized data improves the quality of the analysis or model training, leading to more accurate and reliable results.

### 3. Data Formatting

**Definition**: The process of converting data into a format that is suitable for analysis and modeling. This includes ensuring consistency in data types and structures.

**Processes Involved**:

- **Data Type Conversion**: Converting data into appropriate types (e.g., numeric, categorical, datetime).

- **Data Structuring**: Organizing data into a structured format, such as tables or matrices, that is compatible with machine learning algorithms.
- **Encoding Categorical Variables**: Transforming categorical data into numerical format using methods like one-hot encoding or label encoding.

**Importance**: Proper data formatting ensures that the data is compatible with machine learning algorithms and tools, facilitating smooth and effective analysis.

### 4. Data Sampling

**Definition**: The process of selecting a subset of data from the larger dataset for training, validation, and testing purposes. This step is crucial for managing large datasets and evaluating model performance.

**Processes Involved**:

- **Training Set**: Data used to train the model, allowing it to learn patterns and relationships.
- **Validation Set**: Data used to tune hyperparameters and validate the model's performance during training.
- **Test Set**: Data used to evaluate the final model's performance and generalizability.

**Importance**: Proper data sampling ensures that the model is trained and tested on representative subsets, helping to avoid overfitting and ensuring that the model generalizes well to unseen data.

### 5. Data Scaling

**Definition**: The process of transforming features so that they are on a similar scale. This is important for algorithms that are sensitive to the scale of input features.

**Processes Involved**:

- **Normalization**: Scaling features to a range (e.g., 0 to 1) using techniques like Min-Max scaling.
- **Standardization**: Transforming features to have zero mean and unit variance using techniques like Z-score normalization.

**Importance**: Data scaling ensures that features contribute equally to the model's training process, preventing features with larger scales from dominating the learning process and improving model performance.

b) Analyze Reinforcement Learning with neat diagram. [L4][CO1] [6M]

A) Reinforcement learning: Reinforcement learning is a technique used to teach machines to learn through trial and error. The algorithm receives feedback in the form of rewards or penalties based on its actions, and its goal is to learn a policy that maximizes the expected reward over time.

(REFOR DIAGRAM IN CLASSWORK)

*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.

Here are some common techniques used in Reinforcement Learning:

1.Q-Learning: A model-free reinforcement learning algorithm that seeks to find the best action to take given the current state.

2.Deep Q-Networks (DQN): Combines Q-Learning with deep neural networks, particularly useful in environments with large state spaces.

**3.Actor-Critic**: Combines both value-based and policy-based approaches. The "actor" updates the policy parameters, while the "critic" evaluates the policy by estimating the value function.

4.Policy Gradient: Policy Gradient is a technique that directly optimizes the policy by updating the parameters of the policy function based on the gradient of the expected reward.

7. a) Explain data processing and techniques used for data preprocessing. [L2][CO1] [6M]

A) As same as Answer 6.a)

b) Analyze the real world applications of ML. [L4][CO1] [6M]

A) **Here's your content with added two-line paragraphs about the algorithms used, highlighted in bold:**

---

1. *Healthcare*

• Disease Diagnosis: ML helps doctors analyze medical images (like X-rays) to find diseases early, such as cancer.

Convolutional Neural Networks (CNNs) are widely utilized for image analysis and feature extraction in medical imaging.

• Personalized Medicine: ML suggests tailored treatments for individuals based on their health data.

Decision Trees and Random Forests can be used to analyze patient data and suggest customized treatment plans.

2. *Finance*

• Fraud Detection: ML spots unusual patterns in transactions to prevent fraud.

Anomaly Detection techniques and Neural Networks are commonly employed to identify fraudulent activities in financial transactions.

• Stock Trading: ML predicts stock prices to help with buying and selling decisions.

Time Series Analysis and LSTM networks are effective for forecasting stock price movements based on historical data.

3. *Retail and E-Commerce*

• Product Recommendations: ML suggests products based on what you've looked at or bought before.

Collaborative Filtering and Matrix Factorization techniques are popular for generating personalized product recommendations.

• Inventory Management: ML predicts how much stock is needed to avoid running out or having too much.

Regression models and Time Series Forecasting are used to predict future inventory needs based on historical sales data.

4. *Transportation*

• Self-Driving Cars: ML helps autonomous vehicles navigate and make decisions.

Computer Vision and Reinforcement Learning are utilized to enable cars to interpret their surroundings and make driving decisions.

• Route Optimization: ML finds the best delivery routes to save time and fuel.

Graph Algorithms and Genetic Algorithms can optimize routes based on various factors like traffic and distance.

5. *Education*

• Personalized Learning: ML adjusts learning materials to fit each student's needs.

Adaptive Learning Algorithms are used to tailor educational content according to individual student performance and preferences.

• Automated Grading: ML grades assignments automatically, saving time for teachers.

Natural Language Processing (NLP) techniques are employed to analyze and grade written assignments efficiently.

6. *Telecommunications*

• Network Optimization: ML improves network performance and reduces downtime.

Clustering Algorithms and Time Series Forecasting are used to analyze network traffic and optimize performance.

• Customer Support: ML-powered chatbots assist with customer service inquiries.

NLP techniques and Decision Trees enable chatbots to understand and respond to customer queries effectively.

7. *Entertainment*

• Content Recommendations: ML suggests movies, shows, or music based on your preferences.

Collaborative Filtering and Deep Learning models are commonly used to recommend content tailored to individual user tastes.

• Content Creation: ML helps create and edit media, like generating music or video edits.

Generative Adversarial Networks (GANs) are used in creative applications for generating new content.

8. *Agriculture*

• Precision Farming: ML helps farmers decide the best times and methods for planting and harvesting.

Regression Models and Decision Trees assist in analyzing various environmental factors for optimal farming decisions.

• Pest Detection: ML identifies pests and diseases in crops early on.

Image Recognition algorithms, particularly CNNs, are used to detect and classify pests in agricultural fields.

8) Demonstrate the Probability theory. [L2][CO1] [12M]
A)

Probability theory is a branch of mathematics that deals with the study of random events and their likelihood of occurrence. It provides a framework for analysing uncertain situations and making predictions based on the available information. The fundamental concept of probability theory is the probability of an event, which is a measure of the likelihood of that event occurring. Probability is usually expressed as a number between 0 and 1, where 0 represents an impossible event and 1 represents a certain event. For example, the probability of flipping a fair coin and getting heads is 0.5, or 50%. Probability theory also includes concepts such as random variables, which are variables that can take on different values in a random manner, and probability distributions, which describe the probabilities of different outcomes of a random variable. There are many different types of probability distributions, including the normal distribution, the binomial distribution, and the Poisson distribution, each of which has its own properties and uses. In addition to these basic concepts, probability theory also includes

tools and techniques for analysing and manipulating probabilities, such as Bayes' theorem, which allows for the updating of probabilities based on new information, and hypothesis testing, which is used to test the validity of statistical claims based on sample data. Probability theory has a wide range of applications in many fields, including statistics, physics, engineering, economics, and finance. It is also an essential foundation for many machine learning and artificial intelligence techniques, such as Bayesian networks and probabilistic graphical models, which rely on probabilistic reasoning to make predictions and decisions

## 1. Basic Concepts of Probability Theory

### 1.1. Probability

**Definition**: Probability measures the likelihood of an event occurring. It is quantified as a number between 0 and 1.

- **0**: The event will not occur.
- **1**: The event will certainly occur.

**Formula**:

$$P(A) = \frac{\text{Number of favorable outcomes}}{\text{Total number of possible outcomes}}$$

### 1.2. Sample Space and Events

- **Sample Space (S)**: The set of all possible outcomes of an experiment.
- **Event (E)**: A subset of the sample space.

## 2. Probability Rules

### 2.1. Addition Rule

**Definition**: Used to find the probability of either of two events occurring.

- **For Mutually Exclusive Events (cannot occur together)**:
  $$P(A \cup B) = P(A) + P(B)$$
- **For General Events (not mutually exclusive)**:
  $$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

P(A∩B) is the probability that both events A and B occur.

### 2.2. Multiplication Rule

**Definition**: Used to find the probability of two events occurring together.

- **For Independent Events (events do not affect each other)**:
  $$P(A \cap B) = P(A) \times P(B)$$
- **For Dependent Events (one event affects the probability of another)**:
  $$P(A \cap B) = P(A) \times P(B|A)$$

Where $P(B|A)P(B|A)$P(B|A) is the probability of B given A.

### 3. Conditional Probability

**Definition**: The probability of an event occurring given that another event has already occurred.

**Formula**:

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

### 4. Bayes' Theorem

**Definition**: A method for updating probabilities based on new evidence.

**Formula**:

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

9 a) Differentiate the Bias and Variance tradeoff in Machine Learning. [L4][CO1] [6M]
A)

| Aspect | Bias | Variance |
|---|---|---|
| **Definition** | Error introduced by overly simplistic models. | Error introduced by overly complex models. |
| **Error Type** | Systematic error (consistent prediction errors). | Random error (fluctuates with different datasets). |
| **Training Error** | High (model cannot fit training data well). | Low (model fits training data very well). |
| **Test Error** | May be high due to underfitting. | May be high due to overfitting. |
| **Model Complexity** | Low (simple models like linear regression). | High (complex models like deep neural networks). |
| **Prediction Consistency** | Consistent but inaccurate predictions. | Predictions vary significantly with different datasets. |
| **Overfitting/Underfitting** | Underfitting (model is too simple). | Overfitting (model is too complex). |
| **Example** | Linear regression on a non-linear problem. | High-degree polynomial regression on a small dataset. |

b) Compare Machine Learning and Artificial Intelligence. [L4][CO1] [6M]

A)

| Sl.No. | ARTIFICIAL INTELLIGENCE | MACHINE LEARNING |
|---|---|---|
| 1. | 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. |
| 2. | 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 |
| 3. | AI is the broader family consisting of ML and DL as its components. | Machine Learning is the subset of Artificial Intelligence. |
| 4. | 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 |
| 5. | 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. |
| 6. | It works as a computer program that does smart work. | Here, the tasks systems machine takes data and learns from data. |
| 7. | 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. |
| 8. | AI has a very broad variety of applications. | The scope of machine learning is constrained. |

| | | |
|---|---|---|
| 9. | AI is decision-making. | ML allows systems to learn new things from data. |
| 10. | It is developing a system that mimics humans to solve problems. | It involves creating self-learning algorithms. |
| 11. | AI will go for finding the optimal solution. | ML will go for a solution whether it is optimal or not. |
| 12. | AI leads to intelligence or wisdom. | ML leads to knowledge. |
| 13. | AI is a broader family consisting of ML and DL as its components. | ML is a subset of AI. |
| 14. | Three broad categories of AI are :<br>1. Artificial Narrow Intelligence (ANI)<br>2. Artificial General Intelligence (AGI)<br>3. Artificial Super Intelligence (ASI) | Three broad categories of ML are :<br>1. Supervised Learning<br>2. Unsupervised Learning<br>3. Reinforcement Learning |

10.a) What is Machine learning? Explain the need of it. [L1][CO1] [6M]

A)

**Machine Learning (ML)** is a branch of artificial intelligence (AI) that focuses on the development of algorithms and statistical models that enable computers to perform tasks without explicit instructions. Instead, these systems learn from and make predictions or decisions based on data.

### Need for Machine Learning

Machine Learning addresses several crucial needs in modern technology and business, making it highly valuable. Here are some key reasons for its importance:

1. **Data-Driven Decision Making**
    - **Need**: Organizations generate vast amounts of data but need to extract actionable insights from it.
2. **Automation of Tasks**
    - **Need**: Many tasks are repetitive and time-consuming, requiring automation to improve efficiency.
3. **Personalization**
    - **Need**: Consumers expect personalized experiences and recommendations based on their preferences and behavior.
4. **Predictive Analysis**
    - **Need**: Businesses and organizations need to forecast future trends and outcomes to plan effectively.
5. **Handling Complex Data**
    - **Need**: Traditional methods struggle with large-scale or complex data sets that are high-dimensional and unstructured.
    - **ML Solution**: ML techniques, such as deep learning, can handle and analyze large volumes of complex data, including images, text, and speech.
6. **Improving Accuracy and Efficiency**
    - **Need**: Many processes require high accuracy and efficiency, which manual methods might not provide.

b) List out applications and some popular algorithms used in Machine Learning. Explain it. [L1][CO1] [6M]
A)

## Applications of Machine Learning

**1.Healthcare**

- **Application**: Disease Diagnosis
    - **Explanation**: ML algorithms analyze medical images and patient data to assist in diagnosing diseases like cancer, detecting abnormalities, and recommending treatment plans.

**2.Finance**

- **Application**: Fraud Detection
    - **Explanation**: ML models identify unusual patterns in financial transactions to detect and prevent fraudulent activities.

**3.E-Commerce**

- **Application**: Recommendation Systems
    - **Explanation**: ML algorithms suggest products to users based on their browsing history, past purchases, and preferences.

**4.Transportation**

- **Application**: Autonomous Vehicles

        - **Explanation**: ML enables self-driving cars to interpret sensory data, make decisions, and navigate safely on the roads.

**5.Marketing**

- **Application**: Customer Segmentation
    - **Explanation**: ML algorithms analyze customer data to segment users into different groups for targeted marketing strategies.

**6.Social Media**

- **Application**: Content Moderation
    - **Explanation**: ML models identify and filter inappropriate or harmful content on social media platforms.

## Popular Machine Learning Algorithms

**1.Linear Regression**

- **Description**: A statistical method used to model the relationship between a dependent variable and one or more independent variables by fitting a linear equation to the data.
- **Use Case**: Predicting house prices based on features such as size, location, and number of rooms.

**2.Decision Trees**

- **Description**: A model that makes decisions by splitting the data into subsets based on feature values, creating a tree-like structure of decisions.
- **Use Case**: Classifying customer churn based on historical data of customer behavior.

**3.Random Forest**

- **Description**: An ensemble method that combines multiple decision trees to improve accuracy and control overfitting.
- **Use Case**: Predicting credit risk by aggregating predictions from multiple decision trees.

**4.Support Vector Machines (SVM)**

- **Description**: A classification technique that finds the hyperplane that best separates different classes in the feature space.
- **Use Case**: Image classification tasks, such as detecting handwritten digits.

**5.K-Nearest Neighbors (KNN)**

- **Description**: A classification algorithm that assigns a class to a data point based on the majority class among its k-nearest neighbors.
- **Use Case**: Recommending products based on the preferences of similar users.

**6.Neural Networks**

- **Description**: Computational models inspired by the human brain, consisting of interconnected nodes (neurons) that process data in layers.
- **Use Case**: Image and speech recognition, such as recognizing objects in photos or transcribing spoken words into text.

**7.K-Means Clustering**

- **Description**: An unsupervised learning algorithm that partitions data into k clusters by minimizing the variance within each cluster.

- **Use Case**: Customer segmentation by grouping similar customers based on purchasing behavior.

**8.Principal Component Analysis (PCA)**

- **Description**: A dimensionality reduction technique that transforms data into a lower-dimensional space while retaining as much variance as possible.
- **Use Case**: Reducing the complexity of data for visualization and analysis in high-dimensional datasets.

## UNIT-2

1. Explain about machine learning classification and its usage. 12M

A) Definition:

Classification is a supervised machine learning task where an algorithm is trained to categorize data into predefined labels or classes. It operates by analyzing input data features and using learned patterns to classify new data points. The main goal is to predict the category of a given input based on past labeled data.

Process of Classification:

1. Data Collection: Initially, a labeled dataset is gathered, where each data point has input features and a target label.
2. Data Preprocessing: The data is cleaned, normalized, and sometimes transformed to improve model accuracy.
3. Model Training: A classification algorithm is trained on the dataset. During this process, the model learns patterns and correlations between the input features and output labels.
4. Evaluation: The model's performance is tested on unseen data to ensure it generalizes well, often using metrics like accuracy, precision, recall, and F1-score.
5. Prediction: Finally, the trained model is used to classify new, unlabeled data points.

*Types of Classification Tasks:*

- *Binary Classification*: Only two possible labels (e.g., "spam" or "not spam" in emails).
- *Multiclass Classification*: More than two labels (e.g., classifying types of animals like "dog," "cat," "bird").
- *Multilabel Classification*: Each data point can have multiple labels simultaneously (e.g., image classification where an image could contain both "dog" and "cat").

There are several types of machine learning classification algorithms, including:

1. Logistic Regression - a linear model that predicts the probability of an example belonging to a certain class.
2. Decision Trees - a non-parametric model that learns a hierarchical series of decisions based on the input features to predict the class label.
3. Random Forests - an ensemble learning method that uses multiple decision trees to improve classification accuracy.
4. Support Vector Machines (SVMs) - a powerful and flexible model that separates data points using a hyperplane in a high-dimensional space.

5. Neural Networks - a powerful and flexible model that can learn complex non-linear relationships between input features and class labels.

Machine learning classification has many applications, including:

1. Image Recognition - Classifying images based on their contents, such as identifying whether an image contains a cat or a dog.
2. Fraud Detection - Identifying fraudulent transactions based on transaction data and user behaviour patterns.
3. Sentiment Analysis - Analysing text data to determine the sentiment or emotion expressed, such as identifying whether a movie review is positive or negative.
4. Customer Segmentation - Dividing customers into groups based on their behaviour.

Advantages of Classification:*

- *Automation*: Automates decision-making processes in various industries.
- *Efficiency*: Speeds up tasks like spam detection, improving productivity.
- *Accuracy*: Well-trained models can achieve high accuracy, enhancing decision quality.

*Challenges:*

- *Data Quality*: Poor-quality data can reduce classification accuracy.
- *Overfitting*: Models may perform well on training data but poorly on new data if they memorize rather than generalize patterns.
- *Class Imbalance*: When one class is underrepresented, the model may struggle to predict it accurately.

## 2.) Explain Decision Tree Classification technique with an example. 12M

A)

o Decision Tree is a supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

o In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

o The decisions or the test are performed on the basis of features of the given dataset.

o It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.

o It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree- like structure.

o In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.

o Step-1: Begin the tree with the root node, says S, which contains the complete dataset.

o Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).
o Step-3: Divide the S into subsets that contains possible values for the best attributes.
o Step-4: Generate the decision tree node, which contains the best attribute.

o Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

**Refer Diagrams from the old PDF**

**3 a) Describe about Multivariate Tree prediction. [6M]**

**A)**

**Multivariate Tree Prediction**

**Definition:**

Multivariate Tree Prediction is an advanced decision tree approach in machine learning where multiple variables (or features) are used simultaneously at each decision node to make predictions. Unlike univariate trees, which split data based on a single feature, multivariate trees consider multiple features at once, allowing them to capture more complex relationships in the data.

**Structure and Working:**

**1. Splitting Criteria:** In a multivariate tree, each node evaluates a linear combination of multiple features rather than a single feature. This allows the tree to split data based on a weighted sum of feature values, enabling more nuanced decision boundaries.

**2. Building the Tree:** The algorithm iteratively finds the best combination of feature weights at each node that maximizes the separation between classes (in classification) or reduces error (in regression).

**3. Prediction:** Once built, the model uses the learned weights and splits to predict outcomes for new data points by traversing the tree from the root node to a leaf node.

**Advantages:**

**Higher Accuracy:** By considering multiple features, multivariate trees can capture complex patterns and dependencies, often leading to improved accuracy over univariate trees.

**Flexibility:** They work well with datasets where relationships between features are not easily separable by single-feature thresholds.

**Reduced Depth:** Multivariate trees often require fewer splits than univariate trees, resulting in shallower trees and potentially faster predictions.

**Applications:**

**Medical Diagnosis:** For predicting diseases where multiple symptoms are interrelated.

**Finance:** Risk assessment and credit scoring, where various financial indicators are combined.

**Marketing:** Customer segmentation based on multiple behavioral and demographic factors.

**Challenges:**

**Complexity:** Training multivariate trees can be computationally intensive, as finding the best feature combination requires more calculations.

**Overfitting:** Without careful tuning, they may overfit on training data, especially with smaller datasets.

**3 b) Describe about Univariate Tree prediction. [6M]**

**A) Univariate Tree Prediction**

**Definition:**

Univariate Tree Prediction is a type of decision tree where each decision node splits the data based on a single feature (variable) at a time. This means that each split in the tree only considers one feature's value to divide the data into subsets, resulting in simpler and more interpretable decision boundaries.

**Structure and Working:**

**1. Splitting Criteria:** At each node, the algorithm identifies the best feature and threshold value that optimally separates the data based on a single variable. This can be done using criteria such as Gini impurity, entropy (in classification), or mean squared error (in regression).

**2. Building the Tree:** The tree is built by recursively splitting the data at each node using the chosen single feature, creating branches that represent different paths based on the feature's values**.**

**3. Prediction:** For new data, predictions are made by traversing the tree from the root to a leaf node, following paths determined by the values of individual features at each split.

**Advantages:**

**Simplicity:** Univariate trees are easy to interpret since each split only depends on one feature, making it clear how decisions are made.

**Low Computational Cost:** Since they only consider one feature per split, univariate trees require less computation than multivariate trees.

**Visualization:** Univariate trees are easier to visualize and understand, making them suitable for explaining results to non-experts.

**Applications:**

**Customer Churn Prediction:** Identifying customers likely to leave based on a few key factors.

**Medical Testing:** Simple diagnostic tools based on individual symptoms or test results.

**Loan Approval:** Deciding loan approval based on single factors like income or credit score.

**Challenges:**

**Limited Decision Boundaries:** Univariate trees can struggle with complex data patterns that require interactions between features.

**Overfitting:** They can be prone to overfitting on noisy data unless pruned or regularized.

**4) Recognize the role of Pruning in machine learning. [L1][CO1] [12M]**

A) **Definition**:

Pruning is a technique used in machine learning, especially in decision tree algorithms, to reduce the size of the tree by removing branches that contribute little to the predictive power. The goal is to prevent overfitting by simplifying the model, which improves its ability to generalize to new data. Pruning helps in creating a balance between bias and variance, making the model both accurate and efficient.

**Role and Importance of Pruning:**

**1. Reduces Overfitting:** Pruning removes parts of the tree that capture noise or overly specific patterns from the training data, which might not be relevant for new, unseen data. By eliminating these complex branches, the model focuses on the underlying, general patterns.

**2. Improves Generalization:** A pruned model often performs better on test data, as it is less likely to have memorized training data idiosyncrasies. This leads to improved accuracy when the model encounters new examples.

**3. Increases Model Interpretability:** Pruning results in simpler trees, making the decision-making process easier to understand and interpret. In applications requiring transparency, such as finance or healthcare, this interpretability is highly beneficial.

**4. Reduces Complexity and Training Time:** With fewer nodes and branches, the tree becomes smaller and less complex, which can reduce computational cost during both training and inference.

**Types of Pruning Techniques:**

**1. Pre-Pruning (Early Stopping):** In pre-pruning, the tree-building process stops early if adding more branches does not significantly improve model performance. Criteria such as maximum tree depth, minimum samples per leaf, or a minimum gain threshold can trigger stopping. This method avoids growing a full tree but may risk underfitting if the tree stops too early.

**2. Post-Pruning (Cost Complexity Pruning):** In post-pruning, the tree is fully grown and then trimmed by removing branches that have low importance or predictive power. Cost complexity pruning (used in CART algorithms) assigns a penalty to complex trees, encouraging a trade-off between accuracy and simplicity. Post-pruning is often preferred as it allows for a more thorough evaluation before reducing complexity.

**Pruning Process:**

**1. Tree Growth:** A decision tree is grown to its full depth, which ensures that all training data is correctly classified but may overfit.

**2. Evaluation:** Each subtree or branch is evaluated, usually using a validation set or cross-validation to check its contribution to the overall accuracy.

**3. Branch Removal:** Subtrees or nodes that do not contribute significantly to accuracy or that increase error on the validation set are pruned.

**4. Final Pruned Tree:** The pruned tree is saved, which contains fewer branches and performs better on new data by avoiding overfitting.

**Metrics Used in Pruning:**

**Gini Impurity and Entropy:** In classification tasks, these measures help evaluate whether a split is beneficial. Nodes with little reduction in impurity or entropy are often pruned.

**Mean Squared Error (MSE):** In regression trees, nodes contributing minimally to error reduction may be pruned.

**Cross-Validation Score:** By comparing performance on validation data before and after pruning, we ensure that only branches reducing generalization error are retained.

**Applications of Pruning:**

**Finance:** In credit scoring models, pruning enhances model simplicity and transparency, making it easier to understand decisions.

**Healthcare:** Pruning can create simpler diagnostic models that are easier to interpret and validate.

**Marketing:** For customer segmentation, pruning can help build trees that are straightforward and effective without unnecessary complexity.

**Image Recognition:** Pruning simplifies complex decision boundaries, improving computational efficiency and model deployment in real-time applications.

**Challenges with Pruning:**

**Risk of Underfitting:** If over-pruned, the model may lose important information, leading to underfitting.

**Parameter Selection:** Finding the right criteria or threshold for pruning can be challenging and may require careful tuning.

**Computational Cost:** Pruning large trees post-hoc can be computationally intensive, especially with complex models or large datasets.

**Refer Diagrams In Previous PDF**

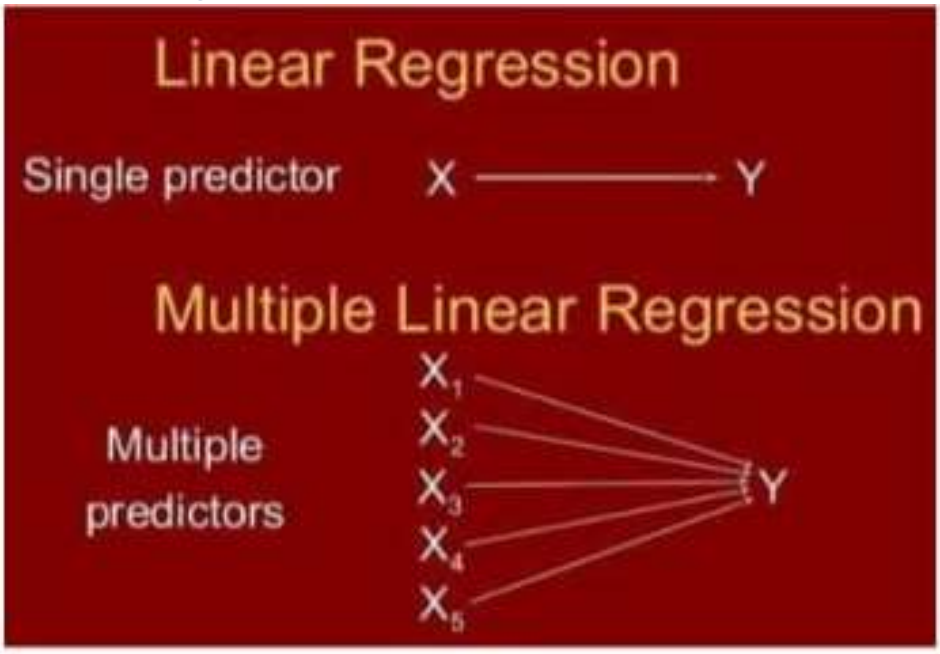
**5) Compare the Linear and Multiple Linear Regressions. 12M**

**A)**

| Aspect | Linear Regression | Multiple Linear Regression |
|---|---|---|
| **Definition** | A model that describes the relationship between a single independent variable and a dependent variable. | A model that describes the relationship between multiple independent variables and a dependent variable. |
| **Equation** | $Y=b0+b1X+\epsilon$ | $Y=b0+b1X1+b2X2+...+bnXn+\epsilon$ |
| **Model Complexity** | Simple, using only one predictor, resulting in straightforward visualization and interpretation. | More complex, using multiple predictors, which increases dimensions and the model's ability to capture variability. |

| | | |
|---|---|---|
| **Interpretation** | The coefficient b1b_1b1 shows the effect of a one-unit increase in XXX on YYY. | Each coefficient bnb_nbn represents the effect of XnX_nXn on YYY, while holding other variables constant. |
| **Visualization** | Can be easily visualized with a 2D scatter plot and a line of best fit. | Challenging to visualize beyond three predictors, often relying on statistical summaries and partial plots. |
| **Assumptions** | Linearity, independence, homoscedasticity, and normality of residuals. | Same as linear regression, with an added assumption of no multicollinearity among predictors. |
| **Applications** | Used for simple, one-variable models (e.g., predicting income based on years of experience). | Useful for complex models with multiple factors (e.g., predicting house prices based on size, location, and age). |
| **Strengths** | Easy to interpret and computationally efficient due to simplicity. | Captures more complex relationships between multiple predictors and the dependent variable. |
| **Weaknesses** | Limited in handling multi-factor relationships, often oversimplifies real-world data. | Prone to overfitting and multicollinearity, making careful predictor selection necessary. |
| **Evaluation Metrics** | R-squared and Mean Squared Error (MSE) help gauge fit and prediction accuracy. | R-squared, Adjusted R-squared (accounts for multiple predictors), and MSE provide a robust assessment. |
| **Example Use Cases** | Simple predictions, like estimating fuel consumption based on speed alone. | Complex predictions, such as estimating sales based on advertising spend, seasonality, and product pricing. |
| **Risk of Overfitting** | Lower risk due to fewer parameters. | Higher risk with many predictors, particularly if some have low significance or there is insufficient data. |

**6.) Explain the different Regression models.** [12M]




**Linear Regression is generally classified into two types:**

1. Simple Linear Regression
2. Multiple Linear Regression

1. Simple

In Simple Linear Regression, we try to find the relationship between a single independent variable (input) and a corresponding dependent variable (output). This can be expressed in the form of a straight line.

The same equation of a line can be re-written as:

$$Y = \beta_0 + \beta_1 X + \epsilon$$

- Y represents the output or dependent variable.
- β0 and β1 are two unknown constants that represent the intercept and coefficient (slope) respectively.
- ε (Epsilon) is the error term.

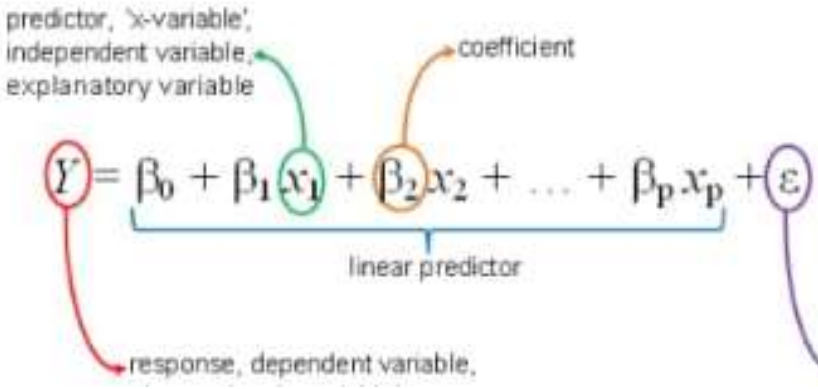
The following is a sample graph of a Simple Linear Regression Model :

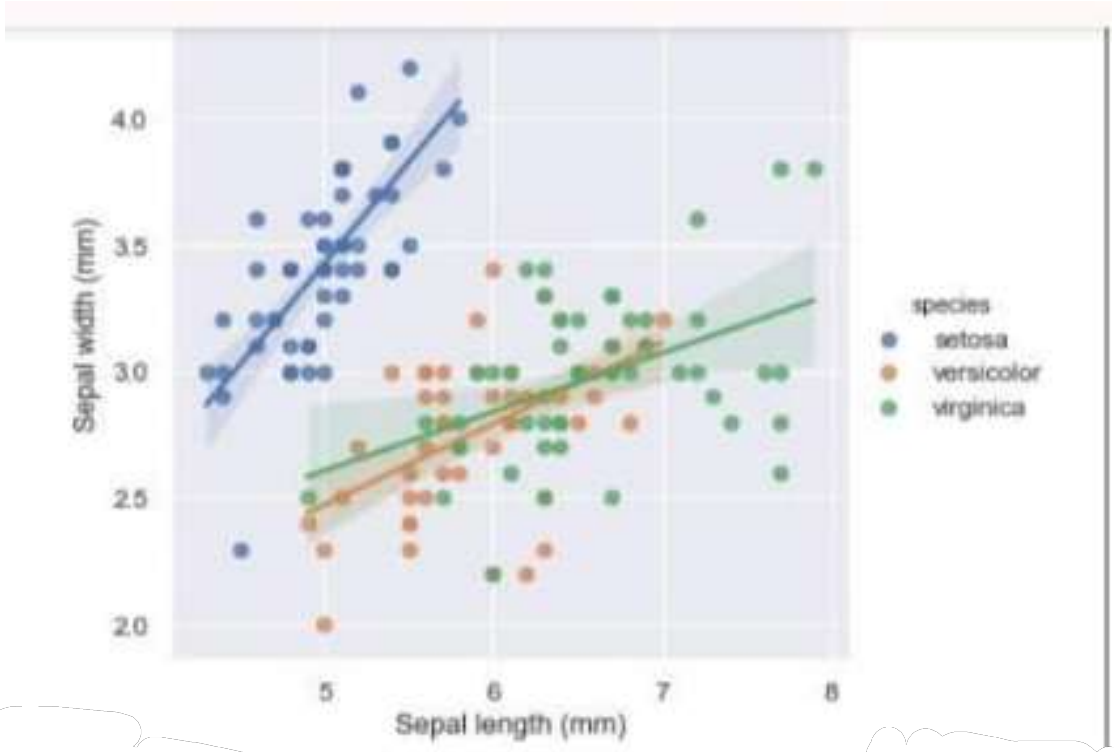

**Multiple Linear Regression**

In Multiple Linear Regression, we try to find the relationship between **2 or more independent variables (inputs)** and the corresponding dependent variable (output). The independent variables can be continuous or categorical.

The equation that describes how the predicted values of y is related to **p independent variables** is called as

**Multiple Linear Regression equation :**



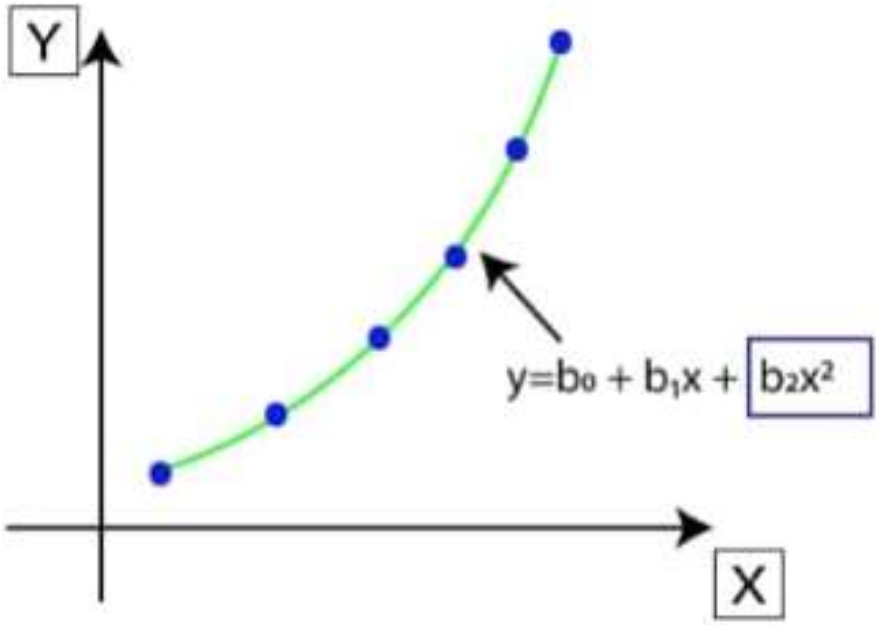



polynomial regression technique

Polynomial Regression is a type of regression which models the **non-linear dataset** using a linear model.

- It is similar to multiple linear regression, but it fits a non- linear curve between the value of x and corresponding conditional values of y.
- Suppose there is a dataset which consists of datapoints which are present in a non-linear fashion, so for such case, linear regression will not best fit to those data points. To cover such data points, we need Polynomial regression.

- **In Polynomial regression, the original features are transformed into polynomial features of given degree and then modelled using a linear model.** Which means the data points are best fitted using a polynomial line.



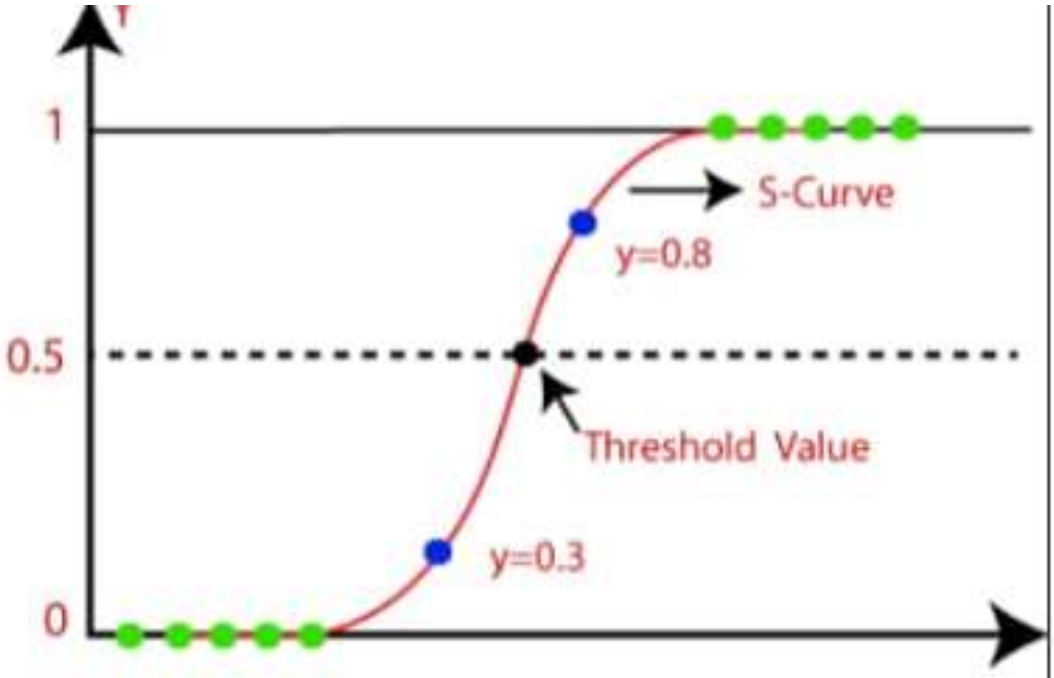

- The equation for polynomial regression also derived from linear regression equation that means Linear regression equation $Y = b_0 + b_1x$, is transformed into Polynomial regression equation $Y = b_0 + b_1x + b_2x^2 + b_3x^3 + \ldots + b_nx^n$.
- Here Y is the **predicted/target output, $b_0$, $b_1$,... $b_n$ are theregression coefficients**. x is our **independent/input variable**.
- The model is still linear as the coefficients are still linear with quadratic

**.ogistic 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. This sigmoid function is used to model the data in logistic regression. The function can be represented as:

$$f(x)= \frac{1}{1+e^{-x}}$$

- f(x)= Output between the 0 and 1 value.
- x= input to the function
- e= base of natural logarithm.




- The equation for logistic regression is:

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

re are three types of logistic regression:

- **Binary(0/1, pass/fail)**
- **Multi(cats, dogs, lions)**
- **Ordinal(low, medium, high)**

7.a) Express in detail about polynomial regression technique

[REFER 6Q – POLYNOMIAL REGRESSION]

7. b) Differentiate between classification and regression.

| Classification | Regression |
|---|---|
| Classification gives out discrete values. | Regression gives continuous values. |
| Given a group of data, this method helps group the data into different groups. | It uses the mapping function to map values to continuous output. |
| In classification, the nature of the predicted data is unordered. | Regression has ordered predicted data. |
| The mapping function is used to map values to pre–defined classes. | It attempts to find a best fit line. It tries to extrapolate the graph to find/predict the values. |
| Example include Decision tree, logistic regression. | Examples include Regression tree (Random forest), Linear regression |
| Classification is done by measuring the accuracy. | Regression is done using the root mean square error method. |

8. Describe about Multiple linear regression and MLR equations[12 M]

[REFER 6Q –MLR REGRESSION]

**Assumptions for Multiple Linear Regression**:

o A linear relationship should exist between the Target and predictor variables.

o The regression residuals must be normally distributed.

o MLR assumes little or no multicollinearity (correlation between the independent variable) in data.

**Implementation of Multiple Linear Regression**

1. Data Pre-processing Steps

2. Fitting the MLR model to the training set

3. Predicting the result of the test set

**9 Explain in details of types of Regression model in ML.[12M]**

[REFER 6Q]

**10 Explain about real world Applications of regression in machine learning.[12M]**

1. Predictive Modeling and Forecasting:

- Sales forecasting
- Demand forecasting
- Stock price prediction
- Weather forecasting

2. Marketing and Customer Analysis:

- Market response modeling
- Customer lifetime value prediction
- Market share analysis

3. Financial Analysis:

- Credit scoring and risk assessment
- Portfolio management
- Financial market analysis
- Fraud detection

4. Healthcare and Medical Research:

- Disease prediction and diagnosis

- Patient outcome prediction
- Drug effectiveness analysis
- Health risk assessment

5. Quality Control and Process Optimization:

- Manufacturing process optimization
- Product quality control
- Supply chain optimization
- Anomaly detection

6. Social Sciences and Behavioral Analysis:

- Social and economic impact analysis
- Opinion mining and sentiment analysis
- Education research
- Demographic analysis

7. Energy and Utilities:

- Energy consumption prediction
- Load forecasting
- Energy price modelling
- Renewable energy optimization

8. Sports Analytics:

- Player performance analysis
- Outcome prediction
- Team composition optimization
- In-game decision-making

9. Environmental Analysis:

- Climate modeling
- Pollution prediction
- Environmental impact assessment
- Natural resource management

10. Time Series Analysis:

- Trend analysis
- Seasonal pattern identification
- Economic indicator forecasting
- Stock market analysis

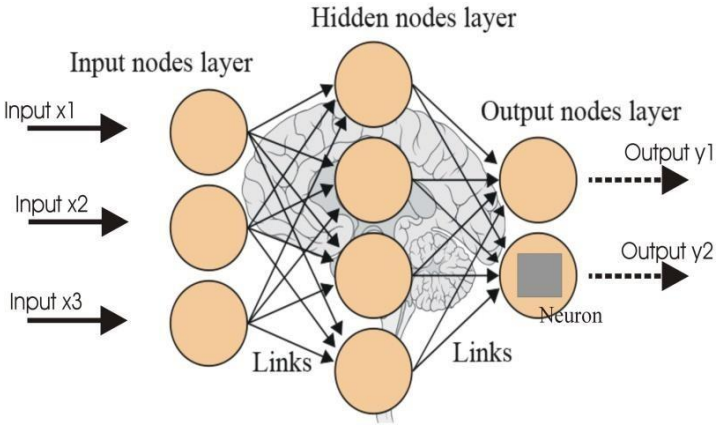
**Course Code:** 20CS0904 **R20**

## UNIT –III
## Learning Models and Decision Theory

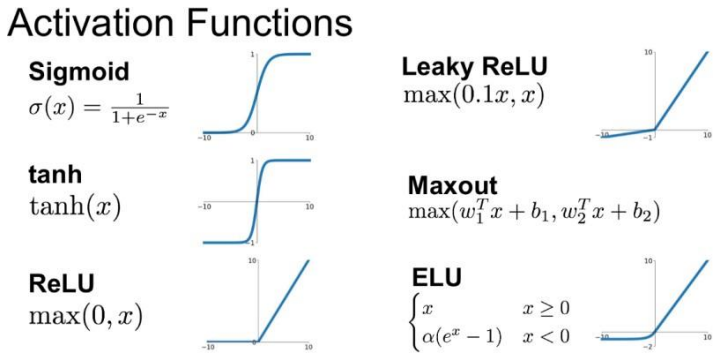
| 1 | A | Describe Artificial Neural Networks | [L1][CO3] | **[4M]** |
|---|---|---|---|---|

An Artificial Neural Network (ANN) is a computational model inspired by the human brain's neural structure. It consists of interconnected nodes (neurons) organized into layers. Information flows through these nodes, and the network adjusts the connection strengths (weights) during training to learn from data, enabling it to recognize patterns, make predictions, and solve various tasks in machine learning and artificial intelligence.

1. There are three layers in the network architecture: the input layer, the hidden layer (more than one), and the output layer. Because of the numerous layers are sometimes referred to as the MLP (Multi-Layer Perceptron).




2. It is possible to think of the hidden layer as a ‑distillation layer,‖ which extracts some of the most relevant patterns from the inputs and sends them on to the next layer for further analysis. It accelerates and improves the efficiency of the network by recognizing just the most important informationfrom the inputs and discarding the redundant information.
3. The activation function is important for two reasons: first, it allows you to turn on your computer. This model captures the presence of non-linear relationships between the inputs.It contributes to the conversion of the input into a more usable output.




4. Finding the ‑optimal values of W — weights‖ that minimize prediction error is critical to building a successful model. The ‑backpropagation algorithm‖ does this by converting ANN into

a learning algorithm by learning from mistakes.

5. The optimization approach uses a ―gradient descent‖ technique to quantify prediction errors. To find the optimum value for W, small adjustments in W are tried, and the impact on prediction errors is examined. Finally, those W values are chosen as ideal since further W changes do not reduce mistakes.

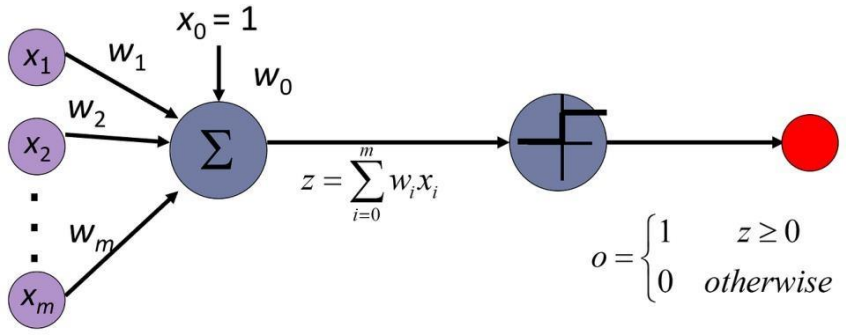
| | | | | |
|---|---|---|---|---|
| | B | Sketch the types of architectures of neural networks | [L2][CO3] | **[8M]** |

Three important types of neural networks:

1. Artificial Neural Networks (ANN)
2. Convolution Neural Networks (CNN)
3. Recurrent Neural Networks (RNN)

**Perceptron**

The perceptron is a fundamental type of neural network used for binary classification tasks. It consists of a single layer of artificial neurons (also known as perceptrons) that take input values, apply weights, and generate an output. The perceptron is typically used for linearly separable data, where it learns to classify inputs into two categories based on a decision boundary.

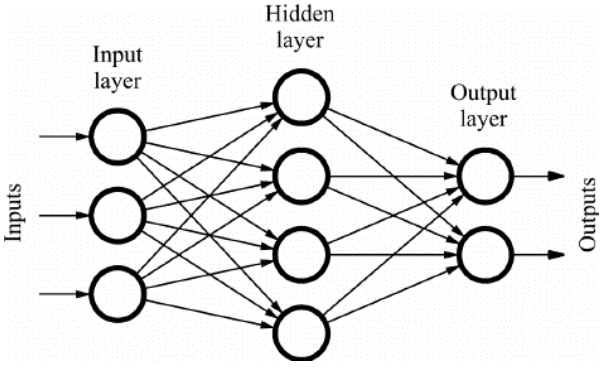



**1. Artificial Neural Networks (ANN)**

**Feed Forward Network**

The Feed Forward (FF) networks consist of multiple neurons and hidden layers which are connected to each other. These are called ―feed-forward‖ because the data flow in the forward direction only, and there is no backward propagation. Hidden layers might not be necessarily present in the network depending upon the application.

More the number of layers more can be the customization of the weights. And hence, more will be the ability of the network to learn. Weights are notupdated as there is no back propagation. The output of multiplication ofweights with the inputs is fed to the activation function which acts as athreshold value.



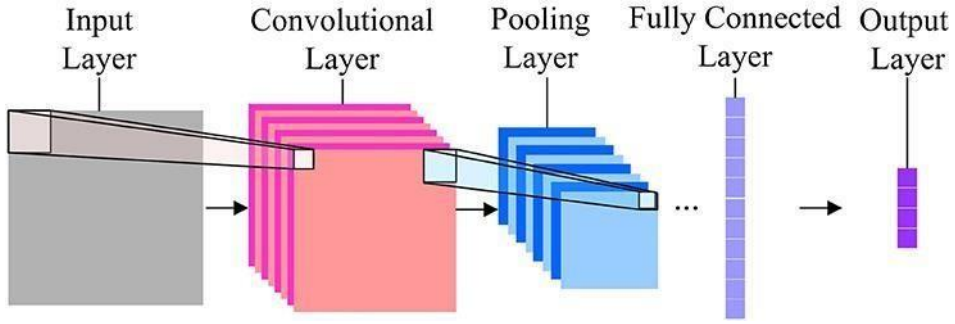
**2.Convolutional Neural Networks**

When it comes to image classification, the most used neural networks are Convolution Neural Networks (CNN). CNN contain multiple convolution layers which are responsible for the extraction of important features from the image. The earlier layers are responsible for low-level details and the later layers are responsible for more high-level features.

The Convolution operation uses a custom matrix, also called as filters, to convolute over the input image and produce maps. These filters are initialized randomly and then are updated via back propagation. One example of such a filter is the Canny Edge Detector, which is used to find the edges in any image.

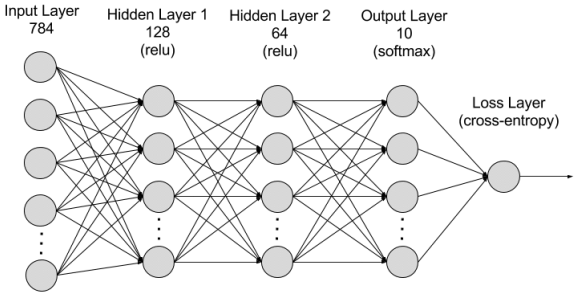
After the convolution layer, there is a pooling layer which is responsible for the aggregation of the maps produced from the convolutional layer. It can be Max Pooling, Min Pooling, etc. For regularization, CNNs also include an option for adding dropout layers which drop or make certain neurons inactive to reduce overfitting and quicker convergence.

CNNs use ReLU (Rectified Linear Unit) as activation functions in the hidden layers. As the last layer, the CNNs have a fully connected dense layer and the activation function mostly as Softmax for classification, and mostly ReLU for regression.




**b) Recurrent Neural Network (RNN)?**

Recurrent Neural Network(RNN) is a type of Neural Network where the output from the previous step is fed as input to the current step. In traditional neural networks, all the inputs and outputs are independent of each other. Still, in cases when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words. Thus RNN came into existence, which solved this issue with the help of a Hidden Layer. The main and most important feature of RNN is its Hidden state, which remembers some information about a sequence. The state is also referred to as Memory State since it remembers the previous input to the network. It uses the same parameters for each input as it performs




| | | | |
|---|---|---|---|
| **2** | What is multilayer perceptron? Explain in detail. | [L2][CO4] | **[12M]** |

A multilayer perceptron (MLP) Neural network belongs to the feedforward neural network. It is an Artificial Neural Network in which all nodes are interconnected with nodes of different layers.

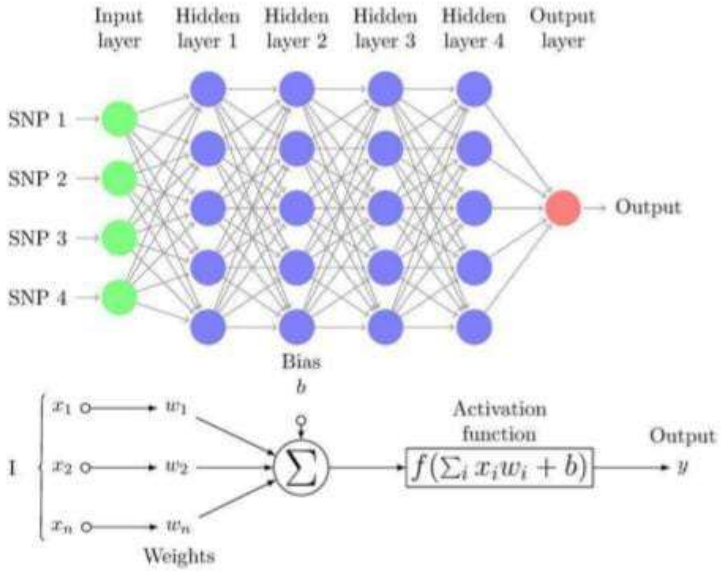
The word Perceptron was first defined by Frank Rosenblatt in his perceptron program. Perceptron is a basic unit of an artificial neural network that defines the artificial neuron in the neural network. It is a supervised learning algorithm that contains nodes' values, activation functions, inputs, and node weights to calculate the output.

The Multilayer Perceptron (MLP) Neural Network works only in the forward direction. All nodes are fully connected to the network. Each node passes its value to the coming node only in the forward direction. The MLP neural network uses a Back propagation algorithm to increase the accuracy of the training model.

**Working of Multilayer Perceptron Neural Network**

- The input node represents the feature of the dataset.
- Each input node passes the vector input value to the hidden layer.
- In the hidden layer, each edge has some weight multiplied by the input variable. All the production values from the hidden nodes are summed together. To generate the output
- The activation function is used in the hidden layer to identify the active nodes.
- The output is passed to the output layer.

- Calculate the difference between predicted and actual output at the output layer.
- The model uses back propagation after calculating the predicted output.




3. a) Explain single layer perceptron in detail

## Single Layer Perceptron (6 Marks)

A **Single Layer Perceptron** (SLP) is the simplest form of artificial neural network used for binary classification tasks. It consists of a single layer of output nodes connected directly to input features, with no hidden layers. Here's a detailed explanation:

### 1. **Structure**

- **Input Layer**: The perceptron receives inputs (features) from the dataset. For an input vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]$, each $x_i$ represents a feature.
- **Weights**: Each input feature is associated with a weight $w_i$. These weights determine the influence of each input on the output.
- **Bias**: A bias term $b$ is added to the weighted sum of inputs to shift the decision boundary away from the origin.

### 2. **Mathematical Representation**

The output of a single-layer perceptron can be represented mathematically as:

$$z = \sum_{i=1}^{n} w_i x_i + b$$

where $z$ is the weighted sum of inputs plus bias.

### 3. **Activation Function**

The perceptron uses an activation function to determine the final output. Commonly, a step function or a binary activation function is used:

$$y = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{if } z < 0 \end{cases}$$

Here, $y$ is the output of the perceptron. The activation function helps in making binary decisions based on the weighted sum of inputs.

### 4. **Learning Algorithm**

The perceptron learns the optimal weights and bias through a process known as the **Perceptron Learning Algorithm**. The steps are as follows:

1. **Initialization**: Set weights and bias to small random values.
2. **Training**: For each training sample:
   - Compute the output $y$.
   - Update the weights and bias based on the prediction error using the rule:
     $$w_i = w_i + \Delta w_i \quad \text{and} \quad b = b + \Delta b$$
     where:
     $$\Delta w_i = \eta(d - y)x_i \quad \text{and} \quad \Delta b = \eta(d - y)$$
     Here, $d$ is the desired output, and $\eta$ is the learning rate.
3. **Convergence**: Repeat until the weights converge (i.e., the output matches the desired output for all training samples).

### 5. **Limitations**

- **Linearly Separable Data**: The single-layer perceptron can only classify linearly separable data. If the data cannot be separated by a straight line (e.g., XOR problem), the perceptron fails to learn.
- **Capacity**: It has limited capacity due to the absence of hidden layers, making it inadequate for complex patterns.

### 6. **Applications**

Despite its limitations, the single-layer perceptron serves as a foundational concept in neural networks and can be applied in simple tasks such as:

- **Binary Classification**: Classifying linearly separable datasets (e.g., spam detection).
- **Logical Operations**: Implementing simple logical functions like AND and OR.

**3B) Explain multi-layer perceptron in detail**
**[REFER 2Q]**

## 4Q. Discuss the following terms a) Feed Forward Neural Networks b) Recurrent Neural Networks c) Convolutional Neural Networks

a) **A feedforward neural network**

It is one of the simplest types of artificial neural networks devised. In this network, the information moves in only one direction—forward—from the input nodes, through the hidden nodes (if any), and to the output nodes. There are no cycles or loops in the network. Feedforward neural networks were the first type of artificial neural network invented and are simpler than their counterparts like recurrent neural networks and convolutional neural networks.

**Architecture of Feedforward Neural Networks**

The architecture of a feedforward neural network consists of three types of layers: the input layer, hidden layers, and the output layer. Each layer is made up of units known as neurons, and the layers are interconnected by weights.

Input Layer: This layer consists of neurons that receive inputs and pass them on to the next layer. The number of neurons in the input layer is determined by the dimensions of the input data.

Hidden Layers:

These layers are not exposed to the input or output and can be considered as the computational engine of the neural network. Each hidden layer's neurons take the weighted sum of the outputs from the previous layer, apply an activation function, and pass the result to the next layer. The network can have zero or more hidden layers.

Output Layer: The final layer that produces the output for the given inputs. The number of neurons in the output layer depends on the number of possible outputs the network is designed to produce.

Each neuron in one layer is connected to every neuron in the next layer, making this a fully connected network. The strength of the connection between neurons is represented by weights, and learning in a neural network involves updating these weights based on the error of the output.

**Feedforward Neural Networks Work**

The working of a feedforward neural network involves two phases: the feedforward phase and the backpropagation phase.

**Feedforward Phase:**

In this phase, the input data is fed into the network, and it propagates forward through the network. At each hidden layer, the weighted sum of the inputs is calculated and passed through an activation function, which introduces non-linearity into the model. This process continues until the output layer is reached, and a prediction is made.

**Backpropagation Phase:**

Once a prediction is made, the error (difference between the predicted output and the actual output) is calculated. This error is then propagated back through the network, and the weights are adjusted to minimize this error. The process of adjusting weights is typically done using a gradient descent optimization algorithm.

**b) Recurrent Neural Network (RNN)?**

Recurrent Neural Network(RNN) is a type of Neural Network where the output from the previous step is fed as input to the current step. In traditional neural networks, all the inputs and outputs are independent of each other. Still, in cases when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words. Thus RNN came into existence, which solved this issue with the help of a Hidden Layer. The main and most important feature of RNN is its Hidden state, which remembers some information about a sequence. The state is also referred to as Memory State since it remembers the previous input to the network. It uses the same parameters for each input as it performs the same task on all the inputs or hidden layers to produce the output. This reduces the complexity of parameters, unlike other neural networks.

C) **Convolutional Neural Networks**

When it comes to image classification, the most used neural networks are Convolution Neural Networks (CNN). CNN contain multiple convolution layers which are responsible for the extraction of important features from the image. The earlier layers are responsible for low-level details and the later layers are responsible for more high-level features.

The Convolution operation uses a custom matrix, also called as filters, to convolute over the input image and produce maps. These filters are initialized randomly and then are updated via back propagation. One example of such a filter is the Canny Edge Detector, which is used to find the edges in any image.

After the convolution layer, there is a pooling layer which is responsible for the aggregation of the maps produced from the convolutional layer. It can be Max Pooling, Min Pooling, etc. For regularization, CNNs also include an option for adding dropout layers which drop or make certain neurons inactive to reduce overfitting and quicker convergence.

CNNs use ReLU (Rectified Linear Unit) as activation functions in the hidden layers. As the last layer, the CNNs have a fully connected dense layer and the activation function mostly as Softmax for classification, and mostly ReLU for regression.

**5 a) State and explain implementation of multilayer perceptron.**

**[REFER 2Q]**

**5.b) What are the advantages of multilayer perceptron?**

## Advantages of Multilayer Perceptrons (MLPs)

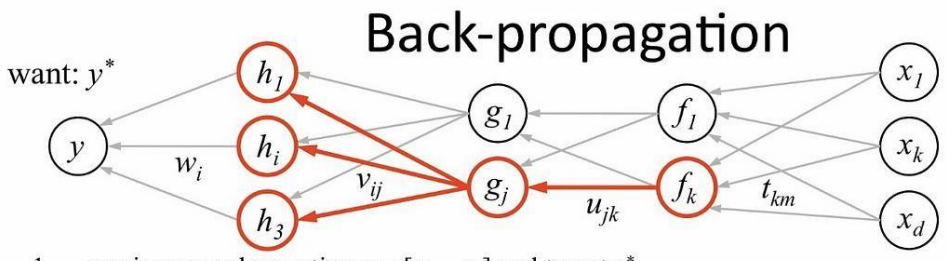
1. **Complex Pattern Learning**:
   - MLPs can model complex, non-linear relationships in data due to their multiple layers and non-linear activation functions, making them effective for various tasks.
2. **Universal Approximation**:
   - According to the universal approximation theorem, MLPs can approximate any continuous function with sufficient neurons in at least one hidden layer, ensuring flexibility in modeling.
3. **Automatic Feature Extraction**:
   - MLPs can automatically extract relevant features from raw data through multiple processing layers, reducing the need for manual feature engineering and enhancing efficiency.
4. **Good Generalization**:
   - With proper training and regularization techniques, MLPs can generalize well to unseen data, making them reliable for classification and regression tasks.
5. **Versatile Applications**:
   - MLPs are widely applicable across different domains, including image recognition, natural language processing, and time-series forecasting, showcasing their versatility.
6. **Integration Capabilities**:
   - MLPs can be easily integrated with other techniques, such as convolutional layers for images or recurrent layers for sequences, enhancing their performance in specialized tasks.

**6 Explain back propagation algorithm with example?[12M]**

**Backpropagation** is the essence of neural network training. It is the method of fine-tuning the weights of a neural network based on the error rate obtained in the previous epoch (i.e., iteration). Proper tuning of the weights allows you to reduce error rates and make the model reliable by increasing its generalization.

Backpropagation in neural network is a short form for –backward propagation of errors.| It is a standard method of training artificial neural networks. This method helps calculate the gradient of a loss function with respect to all the weights in the network.

The main features of Backpropagation are the iterative, recursive and efficient method through which it calculates the updated weight to improve the network until it is not able to perform the task for which it is being trained. Derivatives of the activation function to be known at network design time are required to Backpropagation




1. receive new observation $\boldsymbol{x} = [x_1 \ldots x_d]$ and target $y^*$
2. **feed forward:** for each unit $g_j$ in each layer 1…L
   compute $g_j$ based on units $f_k$ from previous layer: $g_j = \sigma\left(u_{j0} + \sum_k u_{jk} f_k\right)$
3. get prediction $y$ and error $(y\text{-}y^*)$
4. **back-propagate error:** for each unit $g_j$ in each layer L…1

(a) compute error on $g_j$

$$\underbrace{\frac{\partial E}{\partial g_j}}_{\text{should } g_j \text{ be higher or lower?}} = \sum_i \underbrace{\sigma'(h_i) v_{ij}}_{\text{how } h_i \text{ will change as } g_j \text{ changes}} \underbrace{\frac{\partial E}{\partial h_i}}_{\text{was } h_i \text{ too high or too low?}}$$

(b) for each $u_{jk}$ that affects $g_j$

(i) compute error on $u_{jk}$

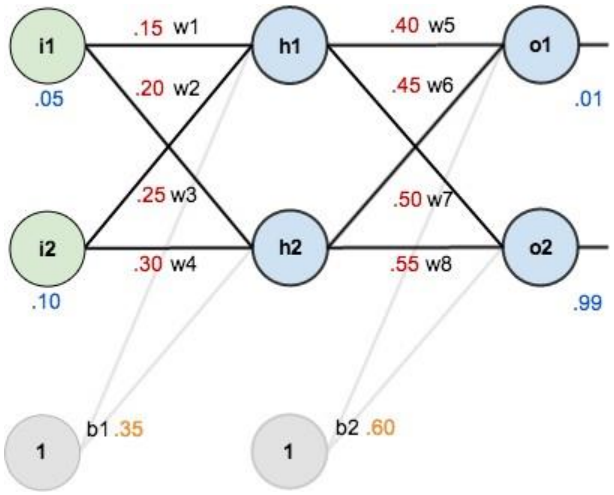
$$\underbrace{\frac{\partial E}{\partial u_{jk}}}_{\text{do we want } g_j \text{ to be higher/lower}} = \frac{\partial E}{\partial g_j} \underbrace{\sigma'(g_j) f_k}_{\text{how } g_j \text{ will change if } u_{jk} \text{ is higher/lower}}$$

(ii) update the weight

$$u_{jk} \leftarrow u_{jk} - \eta \frac{\partial E}{\partial u_{jk}}$$

.

**Example**




The goal of Backpropagation is to optimize the weights so that the neural network can learn how to correctly map arbitrary inputs to outputs.

Given inputs 0.05 and 0.10, we want the neural network to output 0.01 and 0.99.

**The Forward Pass**

To begin, let's see what the neural network currently predicts given the weights and biases above and inputs of 0.05 and 0.10. To do this we'll feed those inputs forward though the network.

We figure out the *total net input* to each hidden layer neuron, *squash* the total net input using an *activation function* (here we use the *logistic function*), then repeat the process with the output layer neurons.

Here's how we calculate the total net input for $h_1$:

$$net_{h1} = w_1 * i_1 + w_2 * i_2 + b_1 * 1$$

$$net_{h1} = 0.15 * 0.05 + 0.2 * 0.1 + 0.35 * 1 = 0.3775$$

$$out_{h1} = \frac{1}{1+e^{-net_{h1}}} = \frac{1}{1+e^{-0.3775}} = 0.593269992$$

$$net_{o1} = w_5 * out_{h1} + w_6 * out_{h2} + b_2 * 1$$

We repeat this process for the output layer neurons, using the output from the hidden layer neurons as inputs.

Here's the output for $o_1$:

$$net_{o1} = w_5 * out_{h1} + w_6 * out_{h2} + b_2 * 1$$
$$net_{o1} = 0.4 * 0.593269992 + 0.45 * 0.596884378 + 0.6 * 1 = 1.105905967$$
$$out_{o1} = \frac{1}{1+e^{-net_{o1}}} = \frac{1}{1+e^{-1.105905967}} = 0.75136507$$

And carrying out the same process for $o_2$ we get:

$$out_{o2} = 0.772928465$$

We can now calculate the error for each output neuron using the **squared error function** and sum them to get the total error:

$$E_{total} = \sum \frac{1}{2}(target - output)^2$$

**Some sources** refer to the target as the *ideal* and the output as the *actual*.

The $\frac{1}{2}$ is included so that exponent is cancelled when we differentiate later on. The result is eventually multiplied by a learning rate anyway so it doesn't matter that we introduce a constant here [**1**].

For example, the target output for $o_1$ is 0.01 but the neural network output 0.75136507, therefore its error is:

Repeating this process for $o_2$ (remembering that the target is 0.99) we get:

The total error for the neural network is the sum of these errors:

$$E_{total} = E_{o1} + E_{o2} = 0.274811083 + 0.023560026 = 0.298371109$$

**The Backwards Pass**

Our goal with backpropagation is to update each of the weights in the network so that they cause the actual output to be closer the target output, thereby minimizing the error for each output neuron and the network as a whole.

Output Layer

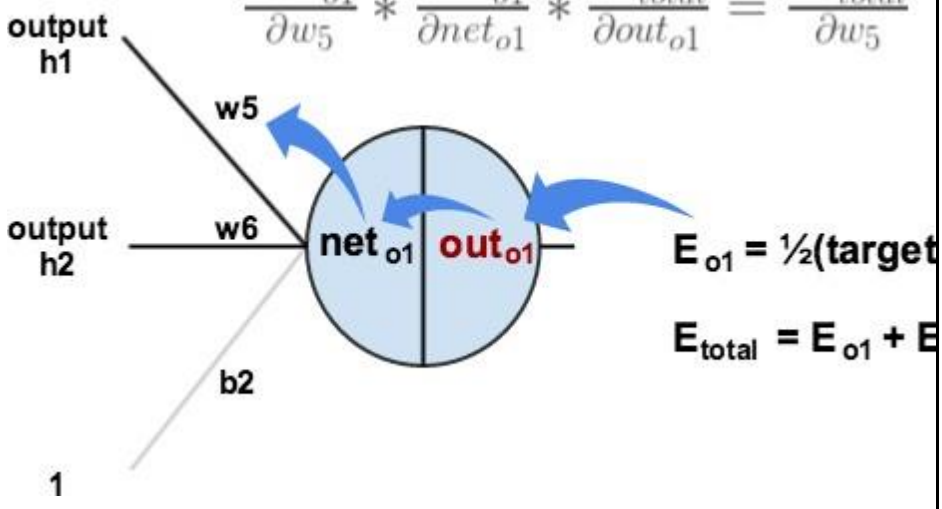
Consider $w_5$. We want to know how much a change in $w_5$ affects the total error, aka $\frac{\partial E_{total}}{\partial w_5}$.

$\frac{\partial E_{total}}{\partial w_5}$ is read as "the partial derivative of with respect to $w_5$". You can also say "the gradient with respect to $w_5$".

By applying the **chain rule** we know that:

Visually, here's what we're doing:

$$\frac{\partial net_{o1}}{\partial w_5} * \frac{\partial out_{o1}}{\partial net_{o1}} * \frac{\partial E_{total}}{\partial out_{o1}} = \frac{\partial E_{total}}{\partial w_5}$$




We need to figure out each piece in this equation.

First, how much does the total error change with respect to the output?

$E_{total} = \frac{1}{2}(target_{o1} - out_{o1})^2 + \frac{1}{2}(target_{o2} - out_{o2})^2$

$\frac{\partial E_{total}}{\partial out_{o1}} = 2 * \frac{1}{2}(target_{o1} - out_{o1})^{2-1} * -1 + 0$

$\frac{\partial E_{total}}{\partial out_{o1}} = -(target_{o1} - out_{o1}) = -(0.01 - 0.75136507) = 0.74136507$

$-(target - out)$ is sometimes expressed as $out - target$

When we take the partial derivative of the total error with respect to $out_{o1}$, the quantity $\frac{1}{2}(target_{o2} - out_{o2})^2$ becomes zero because does not affect it which means we're taking the derivative of a constant which is zero.

Next, how much does the output of $o_1$ change with respect to its total net input?

The partial **derivative of the logistic function** is the output multiplied by 1 minus the output:

$out_{o1} = \frac{1}{1+e^{-net_{o1}}}$

$\frac{\partial out_{o1}}{\partial net_{o1}} = out_{o1}(1 - out_{o1}) = 0.75136507(1 - 0.75136507) = 0.186815602$

Finally, how much does the total net input of $o1$ change with respect to $w_5$?

$net_{o1} = w_5 * out_{h1} + w_6 * out_{h2} + b_2 * 1$

$\frac{\partial net_{o1}}{\partial w_5} = 1 * out_{h1} * w_5^{(1-1)} + 0 + 0 = out_{h1} = 0.593269992$

Putting it all together:

You'll often see this calculation combined in the form of the **delta rule**:

$\frac{\partial E_{total}}{\partial w_5} = -(target_{o1} - out_{o1}) * out_{o1}(1 - out_{o1}) * out_{h1}$

Alternatively, we have $\frac{\partial E_{total}}{\partial out_{o1}}$ and $\frac{\partial out_{o1}}{\partial net_{o1}}$ which can be written as $\frac{\partial E_{total}}{\partial net_{o1}}$, aka $\delta_{o1}$ (the Greek letter delta) aka the *node delta*. We can use this to rewrite the calculation above:

$\delta_{o1} = -(target_{o1} - out_{o1}) * out_{o1}(1 - out_{o1})$

Therefore:

$\frac{\partial E_{total}}{\partial w_5} = \delta_{o1} out_{h1}$

Some sources extract the negative sign from $\delta$ so it would be written as:

## 7 A) Describe Bayesian decision classifier.

### Bayesian Decision Classifier

The Bayesian Decision Classifier is a probabilistic model used in machine learning and statistics for classification tasks. It is based on Bayes' theorem, which provides a way to update the probability estimate for a hypothesis as more evidence or information becomes available. Here's a detailed description suitable for a 6-mark answer:

1. **Foundation on Bayes' Theorem**:
   - The Bayesian classifier uses Bayes' theorem to calculate the posterior probability of each class given the input features. The theorem is mathematically expressed as:
     $$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)}$$
     where $P(C|X)$ is the posterior probability of class $C$ given the feature vector $X$, $P(X|C)$ is the likelihood of the features given the class, $P(C)$ is the prior probability of the class, and $P(X)$ is the evidence.
2. **Assumption of Independence**:
   - The Naive Bayes classifier, a popular variant of the Bayesian classifier, assumes that features are conditionally independent given the class label. This simplifies the computation of the likelihood $P(X|C)$:
     $$P(X|C) = \prod_{i=1}^{n} P(X_i|C)$$
     This assumption allows the model to handle high-dimensional data efficiently.
3. **Class Prediction**:
   - To classify a new instance, the Bayesian classifier computes the posterior probabilities for each class and assigns the instance to the class with the highest probability:
     $$\hat{C} = \arg\max_{C} P(C|X)$$
4. **Incorporation of Prior Knowledge**:
   - The model allows the integration of prior knowledge through the prior probabilities $P(C)$. This is particularly useful when dealing with imbalanced datasets, as it can influence the classification outcomes by reflecting the relative frequency of classes in the training data.
5. **Handling Continuous Features**:
   - For continuous features, the Bayesian classifier can utilize different distributions (e.g., Gaussian) to model the likelihood. For instance, if the features are assumed to be normally distributed, the likelihood can be computed using the probability density function of the Gaussian distribution.
6. **Advantages**:
   - Bayesian classifiers are computationally efficient, easy to implement, and perform well even with a small amount of training data. They are robust to irrelevant features and can provide probabilistic outputs, which are valuable for decision-making.

# 7 b) Explain linear discriminant analysis

## Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) is a statistical method used for classification and dimensionality reduction. It is particularly effective in situations where classes are linearly separable. Here's a detailed explanation suitable for an academic context:

1. **Objective:**
   - The primary goal of LDA is to find a linear combination of features that characterizes or separates two or more classes. It aims to project the features in such a way that the distance between the means of different classes is maximized while minimizing the variance within each class.
2. **Mathematical Foundation:**
   - LDA starts by calculating the means of each class and the overall mean of the dataset. For a dataset with $k$ classes, the within-class scatter matrix $S_W$ and the between-class scatter matrix $S_B$ are defined as follows:

$$S_W = \sum_{i=1}^{k} \sum_{x \in C_i} (x - \mu_i)(x - \mu_i)^T$$

$$S_B = \sum_{i=1}^{k} n_i (\mu_i - \mu)(\mu_i - \mu)^T$$

   - Here, $C_i$ represents class $i$, $n_i$ is the number of samples in class $i$, $\mu_i$ is the mean of class $i$, and $\mu$ is the overall mean of all classes.
3. **Eigenvalue Problem:**
   - The next step involves solving the generalized eigenvalue problem to find the optimal projection direction(s). This is done by maximizing the ratio of the determinant of the between-class scatter matrix to the determinant of the within-class scatter matrix:

$$\frac{|S_B|}{|S_W|}$$

   - The eigenvectors corresponding to the largest eigenvalues give the directions that best separate the classes.
4. **Dimensionality Reduction:**
   - LDA not only helps in classification but also reduces dimensionality by projecting the data onto a lower-dimensional space. For $k$ classes, LDA can project the data into a space of at most $k - 1$ dimensions, which can simplify the dataset while retaining discriminatory information.
5. **Classification:**
   - Once the optimal projection is obtained, new samples can be classified based on their distance to the class centroids in the reduced space. The classifier typically assigns a sample to the class with the nearest mean.
6. **Assumptions:**
   - LDA operates under several assumptions:
     - The features are normally distributed within each class.
     - The classes have the same covariance matrix (homoscedasticity).
     - The samples are statistically independent.
7. **Applications:**
   - LDA is widely used in various fields, including finance (for credit scoring), medical diagnostics (for disease classification), and image recognition.
8. **Advantages:**
   - LDA is computationally efficient, easy to interpret, and works well with small sample sizes. It is particularly useful when classes are well-separated and normally distributed.

## 8 Explain linear discriminant analysis with an example

## [REFER 7B FOR THEORY]

### Example of Linear Discriminant Analysis (LDA)

Let's consider a simple example to illustrate how Linear Discriminant Analysis works.

**Scenario**

Suppose we have a dataset of two classes: **Class A** and **Class B**. Each class has two features: **Feature 1** (height in cm) and **Feature 2** (weight in kg).

**Data**

Here's a small dataset:

| Class | Feature 1 (Height) | Feature 2 (Weight) |
|---|---|---|
| A | 150 | 50 |
| A | 160 | 60 |
| A | 155 | 55 |
| B | 170 | 70 |
| B | 180 | 80 |
| B | 175 | 75 |

**Steps in LDA**

1. **Calculate Means**:
   - Mean of Class A:
   $$\mu_A = \left(\frac{150+160+155}{3}, \frac{50+60+55}{3}\right) = (155, 55)$$
   - Mean of Class B:
   $$\mu_B = \left(\frac{170+180+175}{3}, \frac{70+80+75}{3}\right) = (175, 75)$$
2. **Calculate Within-Class and Between-Class Scatter**:
   - **Within-Class Scatter** ($S_W$) measures how much the data points in each class deviate from their class mean.
   - **Between-Class Scatter** ($S_B$) measures how much the class means deviate from the overall mean.
3. **Compute the Projection**:
   - LDA finds a linear combination of features (weights) that maximizes the separation between the classes based on the scatter matrices.
4. **Classification**:
   - Once the projection is calculated, new data points can be classified based on their proximity to the means of Class A and Class B in the reduced space.

**Visualization**

In a 2D space, you would plot the data points for Class A and Class B. The LDA will determine a line (or hyperplane in higher dimensions) that best separates these classes based on their means.

**New Data Point Classification**

Suppose you have a new observation with height = 165 cm and weight = 65 kg. You would project this point onto the LDA line and classify it as Class A or Class B based on its distance from the respective class means.

## 9.Analyze logistic regression and Bayesian logistic regression.

| Aspect | Logistic Regression | Bayesian Logistic Regression |
|---|---|---|
| Definition | A statistical method that models the probability of a binary outcome based on one or more predictor variables using a logistic function. | An extension of logistic regression that incorporates Bayesian principles, allowing for the estimation of probability distributions for the model parameters. |
| Model Interpretation | Estimates the odds of an event occurring by fitting a logistic function to the data. | Provides a probabilistic interpretation of model parameters, offering a distribution over the weights rather than point estimates. |
| Parameter Estimation | Parameters (coefficients) are estimated using Maximum Likelihood Estimation (MLE). | Parameters are estimated using Bayes' theorem, which incorporates prior beliefs and updates them with observed data. |
| Prior Information | Does not incorporate prior information about the parameters. | Allows the incorporation of prior distributions for parameters, reflecting prior beliefs or knowledge before observing the data. |
| Uncertainty Quantification | Provides point estimates for parameters, but does not quantify uncertainty directly. Confidence intervals can be calculated post hoc. | Directly quantifies uncertainty in parameter estimates through posterior distributions, allowing for credible intervals. |
| Handling Overfitting | Regularization techniques (like L1 or L2) can be used to prevent overfitting. | The use of priors can naturally incorporate regularization, especially with informative priors that penalize complex models. |
| Computation | Generally faster to compute, especially for large datasets, due to the MLE approach. | Often requires more computational resources, especially when using methods like Markov Chain Monte Carlo (MCMC) for posterior estimation. |
| Flexibility | Limited to logistic functions for binary outcomes; can be extended to multiclass via multinomial logistic regression. | Can incorporate various models and priors, providing flexibility in modeling complex relationships and accommodating uncertainty. |
| Application Areas | Widely used in fields such as medicine, finance, and social sciences for binary classification tasks. | Useful in situations where uncertainty is important, such as in medical diagnostics, risk assessment, and predictive modeling with small datasets. |

## 10 a) Describe about discriminant functions

## Discriminant Analysis

Discriminant Analysis (DA) is a statistical technique used to determine group membership based on a set of predictor variables. This method aims to assign each observation to a specific group or category based on the characteristics of the independent variables.

Discriminant analysis is a multivariate technique that divides two or more groups of observations (individuals) based on measured variables. It also assesses the impact of each parameter on the separation of groups. Additionally, the method allows for predicting or allocating new observations to previously defined groups using linear or quadratic functions for group assignment.

## Types of Discriminant Analysis

- **Linear Discriminant Analysis (LDA)**
- **Quadratic Discriminant Analysis (QDA)**

---

**Linear Discriminant Analysis (LDA)**

Linear Discriminant Analysis (LDA) is a supervised technique that predicts the class of the dependent variable using a linear combination of independent variables. LDA is based on the following assumptions:

- The independent variables are normally distributed (continuous and numerical).
- Each class has the same variance and covariance.

**Mathematical Formulation:**

1. **Discriminant Function**: The linear discriminant function can be represented as:

$$D(x) = w^T x + b$$

Where:

- $D(x)$ is the discriminant score for the observation $x$.
- $w$ is the weight vector (coefficients for each feature).
- $b$ is the bias term (intercept).
- $x$ is the feature vector.

2. **Weight Calculation**: The weights $w$ can be calculated using:

$$w = S_W^{-1}(\mu_1 - \mu_2)$$

Where:

- $S_W$ is the within-class scatter matrix.
- $\mu_1$ and $\mu_2$ are the means of the two classes.

3. **Decision Boundary**: The decision boundary is formed where the discriminant scores are equal for both classes:

$$D(x) = 0$$

LDA serves both classification and dimensionality reduction purposes, making it a versatile tool in statistical analysis.

---

**Quadratic Discriminant Analysis (QDA)**

Quadratic Discriminant Analysis (QDA) is a subtype of LDA that uses quadratic combinations of independent variables to predict the class of the dependent variable. While QDA maintains the assumption of normal distribution, it does not require the classes to have equal covariance. Consequently, QDA produces a quadratic decision boundary, allowing for more flexibility in modeling complex relationships between variables.

**Mathematical Formulation:**

1. **Quadratic Discriminant Function:** The quadratic discriminant function can be represented as:

$$D(x) = \frac{1}{2}\left(x^T S^{-1}\mu - \frac{1}{2}\mu^T S^{-1}\mu\right) - \frac{1}{2}\log|S|$$

   Where:

   - $D(x)$ is the discriminant score for the observation $x$.
   - $\mu$ is the mean vector of the class.
   - $S$ is the covariance matrix.
   - $|S|$ is the determinant of the covariance matrix.

2. **Decision Boundary:** The decision boundary is determined by the equation:

$$D_1(x) - D_2(x) = 0$$

   Where $D_1(x)$ and $D_2(x)$ are the discriminant functions for class 1 and class 2, respectively.

## 10.b) Differentiate between linear and nonlinear discriminant functions

| Aspect | Linear Discriminant Functions | Nonlinear Discriminant Functions |
|---|---|---|
| **Definition** | Linear discriminant functions create a linear decision boundary to separate classes based on a linear combination of features. | Nonlinear discriminant functions allow for complex, curved decision boundaries to separate classes based on nonlinear combinations of features. |
| **Decision Boundary** | The decision boundary is a straight line (in two dimensions) or a hyperplane (in higher dimensions). | The decision boundary can take various shapes, such as curves or surfaces, allowing for more flexibility in classification. |
| **Mathematical Representation** | Represented as: $D(x) = w^T x + b$ where $w$ is the weight vector and $b$ is the bias term. | Can be represented as: $D(x) = f(x)$ where $f(x)$ is a nonlinear function (e.g., polynomial, radial basis function). |
| **Assumptions** | Assumes that classes have the same variance and covariance (homoscedasticity) and that the feature distributions are normal. | Does not require the same variance and covariance among classes, allowing for more complex relationships. |
| **Computational Complexity** | Generally less computationally intensive and faster to compute due to the simpler model. | Often requires more complex calculations and can be computationally intensive, especially for high-dimensional data. |
| **Use Cases** | Suitable for linearly separable data, such as simple binary classification tasks or datasets where the relationship between features and classes is approximately linear. | Useful for complex datasets where relationships between features are not linear, such as image recognition, speech recognition, and other pattern recognition tasks. |
| **Interpretability** | Easier to interpret and visualize due to the linearity of the model. | More difficult to interpret because of the complexity and variability of the decision boundaries. |
| **Examples** | Linear Discriminant Analysis (LDA), Support Vector Machines with linear kernels. | Quadratic Discriminant Analysis (QDA), Support Vector Machines with nonlinear kernels (e.g., polynomial or radial basis functions), Neural Networks. |

## UNIT – 4

### 1) Explain Bayesian decision theory in detail. [L2][CO4] [12M]

Bayesian decision theory is a statistical framework for decision-making under uncertainty. It is based on the principles of Bayesian statistics, which involves updating prior beliefs with new data to obtain posterior beliefs. In the context of decision theory, Bayesian methods can be used to determine the optimal decision based on the available information.

The basic framework of Bayesian decision theory involves three components:
1. A set of possible decisions or actions that can be taken.

2. A set of possible states of the world, which are not directly observable but can be inferred from the available data.

3. A set of consequences or outcomes that result from each decision and state of the world.

The goal of Bayesian decision theory is to choose the decision that maximizes the expected utility, which is the sum of the utility of each outcome weighted by its probability.

To apply Bayesian decision theory, we need to specify a prior distribution over the possible states of the world and update this distribution with new data using Bayes' theorem. The resulting posterior distribution can be used to calculate the expected utility of each decision, which can then be compared to determine the optimal decision.

[REFER DIAGRAMS FROM OLD PDF]

For example, consider a medical diagnosis problem where a doctor needs to decide whether a patient has a particular disease or not. The doctor can order a diagnostic test, but the test is not perfect and can produce false positive or false negative results. The doctor can also choose to treat the patient or not treat the patient based on the test result. Using Bayesian decision theory, the doctor can specify a prior distribution over the probability of the patient having the disease based on prior knowledge and experience. The doctor can then update this distribution with the test results using Bayes' theorem to obtain the posterior distribution over the probability of the patient having the disease.

The doctor can then calculate the expected utility of each decision (e.g., treat the patient if the posterior probability is above a certain threshold) based on the posterior distribution and the utility of each outcome (e.g., curing the patient or causing harm). The decision with the highest expected utility is then chosen as the optimal decision.

## 2Q. Explain the Classification in Bayesian decision theory with example? [12M]

In Bayesian Decision Theory, decision-making under uncertainty can be classified into two main types: **Deterministic Decisions** and **Stochastic Decisions**. Each type has distinct characteristics based on how uncertainty and data influence decision-making.

### 1. Deterministic Decisions

In deterministic decision-making, the decision rule is fixed and relies solely on observed data, without incorporating uncertainty about the true state of the world. These decisions are straightforward and do not adapt to new information or changing conditions.

### Characteristics:

- The decision is based on predefined rules or thresholds.
- There is no need to update beliefs or probabilities with new data.
- It is often simpler and computationally less intensive.

**- Example:** Consider a factory producing electronic components where quality control inspects each component for defects. If the component meets the quality threshold, it is accepted; otherwise, it is rejected. The decision to accept or reject does not change based on any uncertainty about the component's true quality. The quality control process here is deterministic

because it follows a fixed rule that does not incorporate any probability of error or uncertainty.

**- Advantages:**

- Simple to implement and understand.

- Computationally efficient, as no additional calculations are needed for updating probabilities.

**- Disadvantages:**

- Lacks flexibility in changing or uncertain environments.

- Does not adapt when new information becomes available.

**2. Stochastic Decisions**

In contrast, stochastic decision-making takes uncertainty into account by updating decisions based on the **posterior probability distribution**. This means that the decision rule is adaptive and incorporates new data to account for the uncertain state of the world.

**- Characteristics:**

- Decisions are made using probabilities, often updated with new information.

- It considers both prior knowledge and observed data, resulting in a posterior probability.

- Stochastic decisions involve calculating expected outcomes based on probabilities, often using Bayesian inference.

**- Example:** A financial portfolio manager deciding which stocks to buy or sell illustrates stochastic decision-making. The manager uses Bayesian Decision Theory to estimate the expected return and risk of each stock based on

historical data, adjusting these estimates as new market data arrives. By calculating the posterior probability distribution of each stock's performance, the manager can make decisions (such as buying or selling) that consider both past performance and current market trends. This approach enables a more dynamic response to changing conditions in the market.

**- Advantages:**

- More flexible and adaptive, as it can incorporate and adjust based on new information.

- Can handle complex and uncertain environments more effectively.

**- Disadvantages:**

- Computationally intensive, requiring calculations of probabilities and expected outcomes.

- Requires sufficient data and prior information for reliable decision-making.

**3 Explain in detail about Expectation- Maximization algorithm with an example [L2][CO4] [12M]**

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.

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.

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.

### 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).

**[REFER EXAMPLE PROBLEM FROM THE CLASS NOTES OR ANY OTHER SOURCE]**

**4 Express discriminant functions [L2][CO4] [12M]**

Discriminant functions are a set of mathematical functions that are used to classify observations into different classes based on their measured features or variables. In pattern recognition and machine learning, discriminant functions are used to identify patterns or relationships in data and to make predictions about the class or category of a new observation.

In discriminant analysis, the discriminant function is derived from the features of the training data and is used to classify new observations into one of the pre- defined classes.

There are different types of discriminant functions, depending on the nature of the data and the problem at hand. Some common types of discriminant functions include:

1. **Linear Discriminant Function:**
    - **Purpose:** Finds a linear combination of features to separate classes with a linear boundary.
    - **Mathematical Form:** $g(x) = w^T x + w_0$, where $w$ is the weight vector and $w_0$ is the bias term.
    - **Example Application:** Linear Discriminant Analysis (LDA) is commonly used in two-class problems like binary medical diagnoses, where the data distributions are roughly Gaussian with equal covariance.
    - **Key Idea:** LDA maximizes the separation between classes by projecting data onto a line, minimizing within-class variance while maximizing between-class variance.
2. **Quadratic Discriminant Function:**
    - **Purpose:** Defines a quadratic boundary between classes and is useful when classes have different covariance structures.
    - **Mathematical Form:** $g(x) = x^T Ax + b^T x + c$, where $A$ is a matrix of quadratic terms, $b$ is a vector, and $c$ is a constant.
    - **Example Application:** Quadratic Discriminant Analysis (QDA) is commonly applied in finance for credit scoring, where different groups (e.g., low and high credit risk) have different distributions.
    - **Key Idea:** QDA allows for more flexibility than LDA by fitting quadratic curves, better handling cases where classes have different variances.
3. **Non-Parametric Discriminant Function:**
    - **Purpose:** Uses a flexible, non-linear boundary to separate classes without assuming a specific data distribution.
    - **Example Application:** K-Nearest Neighbors (KNN) classification uses a non-parametric approach, often in image recognition, where boundaries can be highly complex.
    - **Key Idea:** Non-parametric functions make fewer assumptions about data distribution and adapt well to complex patterns, although they may require more computational resources.

**Applications of Discriminant Functions:**

Discriminant functions are widely used in applications that require distinguishing between different classes, such as:

- **Image Recognition**: Separating objects or faces in images.
- **Speech Recognition**: Identifying spoken words or phonemes.
- **Natural Language Processing**: Classifying text or sentiment.
- **Bioinformatics**: Predicting protein functions or genetic mutations.

## 5. Explain about maximum likelihood estimation in detail. [L2][CO5] [12M]

**Maximum Likelihood Estimation (MLE)** is a statistical method used to estimate the parameters of a probabilistic model, allowing us to find the parameter values that make the observed data most probable. It's widely used in fields such as machine learning, econometrics, and biology to model distributions and make predictions based on observed data.

### Key Concepts of MLE:

1. **Likelihood Function:**
   - The likelihood function, $L(\theta|X)$, measures how likely it is to observe the given data $X$ under various possible parameter values $\theta$. In other words, it is a function of $\theta$ given a fixed dataset $X$.
   - For a set of independent observations $X = \{x_1, x_2, \ldots, x_n\}$, where each observation $x_i$ comes from a probability distribution parameterized by $\theta$, the likelihood function is the product of the individual probabilities:

$$L(\theta|X) = \prod_{i=1}^{n} P(x_i|\theta)$$

2. **Log-Likelihood Function:**
   - Since the likelihood function involves a product of probabilities, it can be complex to work with, especially for large datasets. Therefore, we often use the log-likelihood function, which simplifies calculations by converting the product into a sum:

$$\ln L(\theta|X) = \sum_{i=1}^{n} \ln P(x_i|\theta)$$

   - The log-likelihood function retains the maximum location since the logarithm is a monotonic transformation.

3. **Maximizing the Likelihood:**
   - The goal of MLE is to find the parameter value $\theta$ that maximizes the likelihood function. The value that maximizes $L(\theta|X)$ is called the **maximum likelihood estimate** of $\theta$.
   - This is often done by setting the derivative of the log-likelihood function with respect to $\theta$ to zero and solving for $\theta$, or using optimization algorithms if the derivative approach is complex.

## Steps in Maximum Likelihood Estimation:

1. **Define the Probability Model**: Choose a model that you believe describes your data well, such as a Gaussian distribution, binomial distribution, or exponential distribution.
2. **Write the Likelihood Function**: Using the model, express the likelihood function for the observed data in terms of the unknown parameter $\theta$.
3. **Log-Likelihood**: Take the logarithm of the likelihood function to simplify the expression, yielding the log-likelihood.
4. **Maximize the Log-Likelihood**: Differentiate the log-likelihood function with respect to $\theta$ and set it to zero to solve for $\theta$. If it's difficult to solve analytically, numerical methods or gradient-based optimization techniques are used.
5. **Verify the Maximum**: Ensure that the value found corresponds to a maximum by checking the second derivative or using other criteria.

## Example of MLE for a Gaussian Distribution:

Suppose we have a dataset $X = \{x_1, x_2, \ldots, x_n\}$ that we assume follows a Gaussian (normal) distribution with an unknown mean $\mu$ and variance $\sigma^2$. The probability density function of a normal distribution is:

$$P(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

1. **Likelihood Function**: The likelihood for $n$ independent observations is:

$$L(\mu, \sigma^2|X) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

2. **Log-Likelihood Function**:

$$\ln L(\mu, \sigma^2|X) = -\frac{n}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2$$

3. **Maximization:**
   - Differentiate the log-likelihood with respect to $\mu$ and $\sigma^2$, set the derivatives to zero, and solve to find:

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad \text{and} \quad \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu})^2$$

   - These are the MLE estimates for the mean and variance of a Gaussian distribution.

### Advantages of MLE:

1. **Consistency:** As the sample size increases, MLE estimates converge to the true parameter values.
2. **Efficiency:** MLE makes full use of the data by assuming the observed sample best represents the population.
3. **Flexibility:** It applies to many types of distributions and models, including linear regression and logistic regression.

### Disadvantages of MLE:

1. **Sensitivity to Outliers:** MLE can be affected by outliers, especially in small samples.
2. **Dependence on Model Assumptions:** MLE relies heavily on the assumed model being correct.
3. **Complex Computation:** For complex models or large datasets, MLE may require advanced optimization methods, increasing computational complexity.

---

## 6 State and explain the following

## a) Bernoulli density

## b) Multinomial density

## c) Gaussian density [L1][CO5] [12M]

a) The Bernoulli density is a probability distribution that models a binary outcome, where there are only two possible outcomes, typically represented as 0 and 1. The Bernoulli distribution is characterized by a single parameter p, which represents the probability of the outcome being 1. The probability mass function of the Bernoulli distribution is given by:

P(X=x) = p^x * (1-p)^(1-x)

where X is the binary random variable, x can take the values of 0 or 1, and p is the probability of X being 1.

For example, if we have a coin that is flipped and we are interested in the probability of getting heads, we can model this situation with a Bernoulli distribution, where p represents the probability of getting heads on a single flip.

If p=0.5, then the probability of getting heads is equal to the probability of getting tails, and the distribution is symmetric. If p is not equal to 0.5, then the distribution is skewed towards the more likely outcome.

[REFER GRAPHS FROM OLD PDF]

b) The multinomial density is a probability distribution that models outcomes with more than two possible categories. The multinomial distribution is characterized by a vector of probabilities p1, p2, ..., pk, where pi represents the probability of observing category i. The multinomial distribution is used when we have a fixed number of independent trials, and each trial can result in one of k possible outcomes. The probability mass function of the multinomial distribution is given by:

$P(X = (x1, x2, ..., xk)) = n! / (x1! x2! ... xk!) * p1^{x1} * p2^{x2} * ... * pk^{xk}$

where X is a vector of random variables representing the frequencies of each category, n is the total number of trials, and xi represents the number of times category i was observed.

The multinomial distribution satisfies the properties of a probability distribution, meaning that the sum of the probabilities over all possible outcomes equals 1.

For example, if we have a six-sided die that is rolled 10 times and we are interested in the probability of each outcome, we can model this situation with a multinomial distribution, where p1 = p2 = p3 = p4 = p5 = p6 = 1/6, and n=10. The probability of observing 3 ones, 2 twos, 1 three, 2 fours, 1 five, and 1 six is given by the multinomial distribution.

The expected value and variance of a multinomial distribution are given by:

$E(Xi) = n*pi$

$Var(Xi) = npi(1-pi)$

C) The Gaussian density is a continuous probability distribution that is also known as the normal distribution. It is widely used in statistical analysis and

machine learning due to its convenient mathematical properties and its ability to model awide variety of real-world phenomena. The Gaussian distribution is characterized by two parameters: the mean (μ) and the variance (σ^2). The probability density function (PDF) of the Gaussian distribution is given by: f(x) = (1 / (σ*sqrt(2π))) * exp(-(x-μ)^2 / (2σ^2)) where x is the random variable, μ is the mean, σ is the standard deviation, and π is the mathematical constant pi. The Gaussian distribution has a bell-shaped curve, with the peak of the curve at the mean μ. The standard deviation σ controls the spread of the curve. Larger values of σ result in a flatter and wider curve, while smaller values of σ result in a taller and narrower curve.

**[REFER GRAPHS FROM OLD PDF]**

**7 a) Examine about bias and variance [L3][CO4] [ 6M ]**

**[REFER GRAPHS FROM OLD PDF]**

Bias is simply defined as the differences between actual or expected values and the predicted values are known as error or bias error or error due to bias.

$$\text{Bias}(\hat{Y}) = E(\hat{Y}) - Y$$

- **Low Bias:** Low bias value means fewer assumptions are taken to build the target function. In this case, the model will closely match the training dataset.
- **High Bias:** High bias value means more assumptions are taken to build the target function. In this case, the model will not match the training dataset closely.

**Variance** tells that how much a random variable is different from its expected value.

Variance refers to the changes in the model when using different portions of the training data set.

Simply stated, variance is the variability in the model prediction—how much the ML function can adjust depending on the given data set.

- Models with high bias will have low variance.
- Models with high variance will have a low bias.

Characteristics of a high variance model include:

- Noise in the data set
- Potential towards overfitting
- Complex models
- Trying to put all data points as close as possible

**b) Describe the Bernoulli density. Give an example [L1][CO4] [ 6M ]**

a) The Bernoulli density is a probability distribution that models a binary outcome, where there are only two possible outcomes, typically represented as 0 and 1. The Bernoulli distribution is characterized by a single parameter p, which represents the probability of the outcome being 1. The probability mass function of the Bernoulli distribution is given by:

P(X=x) = p^x * (1-p)^(1-x)

where X is the binary random variable, x can take the values of 0 or 1, and p is the probability of X being 1.

For example, if we have a coin that is flipped and we are interested in the probability of getting heads, we can model this situation with a Bernoulli distribution, where p represents the probability of getting heads on a single flip.

If p=0.5, then the probability of getting heads is equal to the probability of getting tails, and the distribution is symmetric. If p is not equal to 0.5, then the distribution is skewed towards the more likely outcome.

[REFER GRAPHS FROM OLD PDF]

### Example

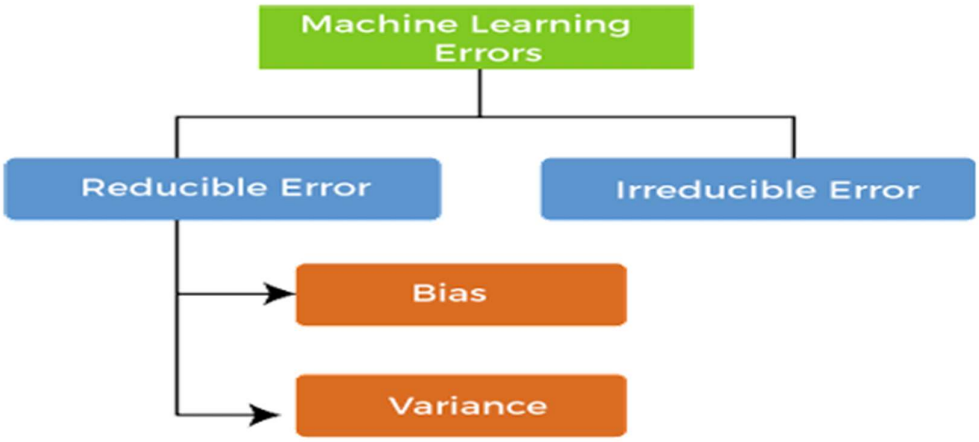
**Scenario**: Let's consider a simple example involving a light bulb.

Suppose we have a light bulb that has a probability $p = 0.9$ of working (being in good condition) when switched on. Conversely, the probability of the light bulb being faulty (not working) is $1 - p = 0.1$.

**Application of Bernoulli Distribution**

1. **Define the Random Variable:**
   - Let $X$ represent the state of the light bulb.
   - $X = 1$ if the light bulb is working (success).
   - $X = 0$ if the light bulb is not working (failure).
2. **Calculate the Probabilities:**
   - The probability that the light bulb is working:
     $$P(X = 1) = p = 0.9$$
   - The probability that the light bulb is not working:
     $$P(X = 0) = 1 - p = 0.1$$
3. **Interpret the Results:**
   - There is a 90% chance that the light bulb will work when switched on and a 10% chance that it will not work.

## 8 ) Explain with example of bias and variance [L3][CO5] [12M]




### Bias and Variance:

Bias and variance are two important concepts in statistics and machine learning that help in understanding the performance of predictive models. They are

critical to understanding the trade-offs involved in model complexity and accuracy.

1. Bias

Bias refers to the error introduced by approximating a real-world problem (which may be complex) with a simplified model. It represents the assumptions made by the model to make the target function easier to learn. High bias can cause an algorithm to miss the relevant relations between features and target outputs, leading to underfitting.

- Example of High Bias: Consider a linear regression model trying to fit a quadratic function. Here, the model is too simple (linear) to capture the underlying relationship (quadratic), resulting in a significant error. The predictions are systematically off from the actual values.

2. Variance

Variance refers to the error introduced by the model's sensitivity to small fluctuations in the training set. High variance can cause an algorithm to model the random noise in the training data, leading to overfitting. This means that while the model performs well on the training data, it performs poorly on unseen data.

- Example of High Variance: Consider a high-degree polynomial regression model trying to fit a dataset with a quadratic relationship. The model might perfectly fit the training data, creating a very complex curve that zigzags through the data points. However, when tested on new data, it may perform poorly, as it has captured noise instead of the underlying trend.

Bias-Variance Trade-off

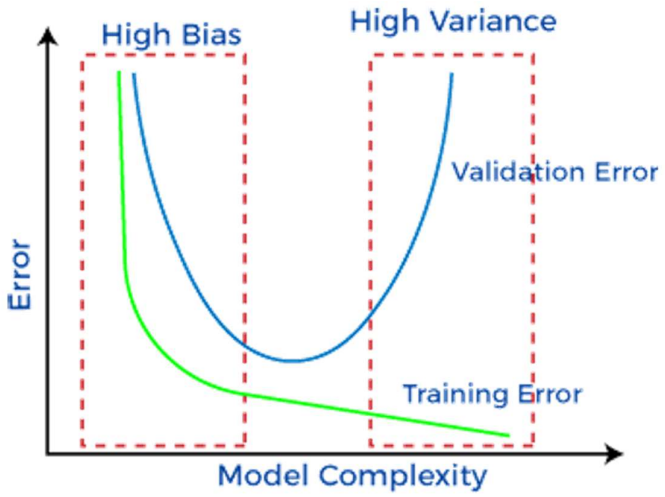
The key to building a good predictive model is to find a balance between bias and variance, known as the bias-variance trade-off. The goal is to minimize the total error, which consists of three components:

- Irreducible error: The noise in the data that cannot be reduced.
- Bias error: Error due to the assumptions made by the model.
- Variance error: Error due to the model's sensitivity to fluctuations in the training set.

Example Scenario

Scenario: Predicting house prices based on square footage.

1. High Bias Model: A simple linear regression model assumes a linear relationship between square footage and price. It might predict prices that are consistently lower than the actual values for larger houses, indicating underfitting.
2. High Variance Model: A complex polynomial regression model fits every data point closely, including outliers and noise in the dataset. While it might perform well on training data, it could fail to generalize to new houses, leading to high error in predictions.




**9 a) Define bias and variance dilemma. Explain in detail [L1][CO5] [ 6M ]**

Bias and Variance Dilemma

The bias and variance dilemma is a fundamental concept in machine learning and statistics that describes the trade-off between two types of errors that affect the performance of predictive models: bias and variance. Understanding this dilemma is crucial for building models that generalize well to unseen data.

1. Bias

- Definition: Bias refers to the error due to overly simplistic assumptions in the learning algorithm. It measures how far off the average prediction of

the model is from the actual values. High bias can lead to underfitting, where the model fails to capture the underlying trends in the data.

- Example: Consider a linear model used to predict a quadratic relationship. The linear model assumes a straight-line relationship, which results in systematic errors in predictions, particularly at the extremes of the dataset.

2. Variance

- Definition: Variance measures how much the model's predictions change when it is trained on different subsets of the training data. High variance can lead to overfitting, where the model learns the noise in the training data rather than the actual signal. This means the model performs well on the training data but poorly on new, unseen data.
- Example: A high-degree polynomial regression model can fit all the training points closely, resulting in a complex model that captures noise and outliers. While it may show high accuracy on training data, it fails to generalize to new data, leading to poor predictions.

3. The Dilemma

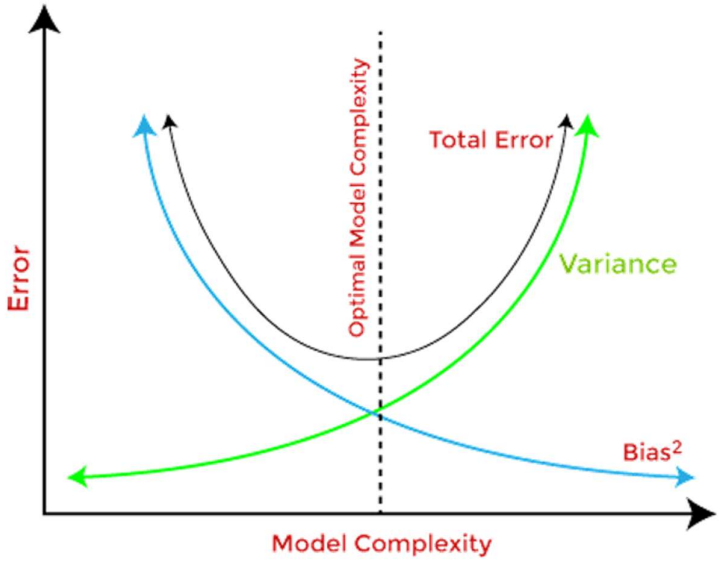
The dilemma arises because increasing model complexity typically reduces bias but increases variance, while reducing complexity increases bias but decreases variance. The goal is to find a balance between these two errors to minimize the overall prediction error on unseen data.

- Underfitting (High Bias): A model that is too simple fails to capture important relationships in the data, leading to high training and test errors.
- Overfitting (High Variance): A model that is too complex captures noise in the training data, resulting in low training error but high test error due to its inability to generalize.

Bias-Variance Trade-Off :

While building the machine learning model, it is really important to take care of bias and variance in order to avoid overfitting and underfitting in the model. If the model is very simple with fewer parameters, it may have low variance and high bias. Whereas, if the model has a large number of parameters, it will have high variance and low bias. So, it is required to make a balance between bias

and variance errors, and this balance between the bias error and variance error is known as the Bias-Variance trade-off.




## 9 b) What is estimator? explain briefly [L1][CO5] [ 6M ]

**Estimator**

An **estimator** is a statistical method or formula used to infer the value of an unknown parameter in a population based on observed data from a sample. Estimators play a crucial role in statistical inference, allowing us to make conclusions about population parameters without having to measure every individual in the population.

**Types of Estimators**

1. **Point Estimator**:
    - A point estimator provides a single value estimate for a population parameter. For example, the sample mean (x¯\bar{x}x¯) is a point estimator of the population mean (μ\muμ).
    - **Example**: If you want to estimate the average height of students in a university, you could measure the heights of a random sample of students and calculate the mean height. This mean serves as a point estimate of the true average height.

2. **Interval Estimator**:
    - An interval estimator provides a range of values within which the population parameter is believed to lie, often accompanied by a confidence level. For example, a confidence interval for the population mean is an interval that estimates the range of the mean based on sample data.
    - **Example**: Continuing from the previous example, you might compute a 95% confidence interval for the average height, indicating that you are 95% confident that the true average height lies within this interval.

**Properties of Good Estimators**

1. **Unbiasedness**: An estimator is unbiased if its expected value equals the true parameter value. This means that, on average, it hits the target.
2. **Consistency**: An estimator is consistent if, as the sample size increases, it converges in probability to the true parameter value.
3. **Efficiency**: An estimator is efficient if it has the smallest possible variance among all unbiased estimators for a given sample size.

## 10) Analyze and explain various model selection procedures

Model selection is an essential phase in the development of powerful and precise predictive models in the field of machine learning. Model selection is the process of deciding which algorithm and model architecture is best suited for a particular task or dataset.

In machine learning, the process of selecting the top model or algorithm from a list of potential models to address a certain issue is referred to as model selection. It entails assessing and contrasting various models according to how well they function and choosing the one that reaches the highest level of accuracy or prediction power.

• Problem formulation: Clearly express the issue at hand, including the kind of predictions or task that you'd like the model to carry out (for example, classification, regression, or clustering).

• Candidate model selection: Pick a group of models that are appropriate for the issue at hand. These models can include straightforward methods like decision trees or linear regression as well as more sophisticated ones like deep neural networks, random forests, or support vector machines.

• Performance evaluation: Establish measures for measuring how well each model performs. Common measurements include area under the receiver's operating characteristic curve (AUC-ROC), recall, F1-score, mean squared error, and accuracy, precision, and recall. The type of problem and the particular requirements will determine which metrics are used.

• Training and evaluation: Each candidate model should be trained using a subset of the available data (the training set), and its performance should be assessed using a different subset (the validation set or via cross- validation). The established evaluation measures are used to gauge the model's effectiveness.

• Model comparison: Evaluate the performance of various models and determine which one performs best on the validation set. Take into account elements like data handling capabilities, interpretability, computational difficulty, and accuracy.

• Hyperparameter tuning: Before training, many models require that certain hyperparameters, such as the learning rate, regularization strength, or the number of layers that are hidden in a neural network, be configured. Use methods like grid search, random search, and Bayesian optimization to identify these hyperparameters' ideal values.

• Final model selection: After the models have been analyzed and fine- tuned, pick the model that performs the best. Then, this model can be used to make predictions based on fresh, unforeseen data

## UNIT-5

## MULTIVARIATE METHODS

**1) Identify and explain about multivariate methods. 12M**

A) Multivariate methods are statistical techniques used to analyze and model relationships between multiple variables simultaneously. These methods are used when there are multiple dependent variables and independent variables, and the goal is to understand the relationships between them.

**Some of the commonly used multivariate methods are:**

**1.Principal Component Analysis (PCA):**

PCA is a technique used to identify patterns in high-dimensional data. It involves finding a set of orthogonal variables (principal components) that capture the maximum amount of variance in the data. PCA can be used to reduce the dimensionality of the data, to visualize the data in a lower-dimensional space, and to identify the most important variables.

**Application**: Often used in exploratory data analysis and for reducing the dimensionality of data while retaining as much variance as possible.

**2. Factor Analysis:** Factor analysis is a technique used to identify underlying factors that explain the covariance between a set of observed variables. It involves extracting a set of latent variables (factors) that account for the observed correlations between the variables. Factor analysis can be used to reduce the dimensionality of the data, to identify the most important variables, and to identify the underlying structure of the data.

Application: Common in psychology and market research to identify underlying structures in data.

**3. Cluster Analysis:** Cluster analysis is a technique used to identify groups (clusters) of similar objects based on their similarity or distance. It involves grouping the objects into clusters such that the objects within each cluster are more similar to each other than to objects in other clusters. Cluster analysis can be used for exploratory data analysis, data visualization, and data mining.

**Application:** Used in market segmentation, social network analysis, and image processing.

**4. Discriminant Analysis:** Discriminant analysis is a technique used to classify objects into predefined categories based on their measurements on multiple variables. It involves finding a set of discriminant functions that can separate the objects into different groups. Discriminant analysis can be used for classification, prediction, and feature selection.

**Application:** Commonly applied in medical diagnosis and risk assessment to classify observations into predefined categories.

**5. Canonical Correlation Analysis**: Canonical correlation analysis is a technique used to identify the correlations between two sets of variables. It involves finding a set of canonical variables (linear combinations of the original variables) that maximize the correlation between the two sets of variables. Canonical correlation analysis can be used to identify the relationships between different variables, to identify the most important variables, and to reduce the dimensionality of the data.

**Application**: Useful in psychology and ecology to explore the relationships between two sets of measurements.

**6. Multivariate Regression Analysis:**

- Description: This method extends simple linear regression to include multiple independent variables to predict a single dependent variable.
- Application: Used in fields like economics and social sciences to understand how various factors affect a specific outcome.

**2) What is parameter estimation? Explain in detail. 12M**

A) Parameter estimation is a fundamental concept in statistics that involves using sample data to infer the values of parameters in a statistical model. This process is crucial for making predictions, understanding relationships between variables, and testing hypotheses. Here's a detailed explanation of parameter estimation, including types, methods, and applications.

**Key Concepts**

1. **Parameters**: These are numerical characteristics of a population, such as the mean, variance, or proportion. In a statistical model, parameters help define the model's structure and behaviour.
2. **Estimation**: This refers to the process of using sample data to calculate estimates of population parameters. The aim is to get as close to the true population values as possible.

**Types of Parameter Estimation**

1. **Point Estimation**:
   - **Definition**: Provides a single value (point estimate) as an estimate of a parameter.
   - **Example**: Using the sample mean ($x¯\bar{x}x¯$) to estimate the population mean ($\mu\muμ$).
2. **Interval Estimation**:
   - **Definition**: Provides a range of values (confidence interval) within which the parameter is expected to lie.
   - **Example**: A 95% confidence interval for the population mean might be (10, 15), suggesting that we are 95% confident that the true mean falls within this range.

**Methods of Parameter Estimation**

1. **Maximum Likelihood Estimation (MLE)**:
   - **Concept**: MLE estimates parameters by maximizing the likelihood function, which measures how likely it is to observe the given sample data for different parameter values.
   - **Application**: Commonly used for a wide range of models, including logistic regression and other generalized linear models.
2. **Method of Moments**:
   - **Concept**: This method involves equating sample moments (like sample mean and variance) to theoretical moments derived from the probability distribution to estimate parameters.
   - **Application**: Often used in simpler models or when MLE is difficult to compute.
3. **Bayesian Estimation**:
   - **Concept**: This approach incorporates prior beliefs about the parameters (prior distribution) along with the observed data (likelihood) to form a posterior distribution.
   - **Application**: Useful in cases where prior knowledge is available, and it can provide a full distribution of parameter estimates rather than a single value.
4. **Least Squares Estimation**:

- **Concept**: Primarily used in regression analysis, this method estimates parameters by minimizing the sum of the squares of the residuals (the differences between observed and predicted values).
- **Application**: Widely used in linear regression.

**Evaluating Estimates**

- **Bias**: An estimator is considered unbiased if its expected value equals the true parameter value. Bias can lead to systematic errors in estimation.
- **Consistency**: An estimator is consistent if, as the sample size increases, it converges in probability to the true parameter value.
- **Efficiency**: An efficient estimator has the smallest variance among all unbiased estimators, meaning it provides more precise estimates.

**Applications of Parameter Estimation**

1. **Predictive Modeling**: Estimating parameters helps build models that can predict future observations or outcomes based on past data.
2. **Quality Control**: In manufacturing, parameter estimation is used to monitor processes and ensure that they remain within specified limits.
3. **Medical Research**: Estimation techniques are essential for analyzing clinical trial data and making inferences about treatments or interventions.
4. **Economics**: Econometric models rely heavily on parameter estimation to understand relationships between economic variables.

**3.) Explain multivariate normal distribution in detail.12M**

A) Multivariate normal distribution, also known as multivariate Gaussian distribution, is a probability distribution that describes the joint distribution of a set of random variables that are correlated with each other.

- It is a generalization of the univariate normal distribution to higher dimensions.
- The multivariate normal distribution is defined by two parameters: the mean vector μ and the covariance matrix Σ.
- The mean vector μ is a p-dimensional vector that represents the expected value of each of the p random variables.
- The covariance matrix Σ is a p×p matrix that represents the degree of correlation between each pair of the p random variables.

The probability density function (pdf) of the multivariate normal distribution is given by:

$f(x) = (1/((2\pi)^{p/2} |\Sigma|^{1/2})) * \exp(-1/2(x-\mu)^T\Sigma^{-1}(x-\mu))$ where x is a p-dimensional vector, $|\Sigma|$ is the determinant of the covariance matrix Σ, and $(\cdot)^T$ denotes the transpose of a matrix or vector. The pdf of the multivariate normal distribution has several important properties:

1. It is symmetric around the mean vector μ.
2. It has a peak at the mean vector μ.
3. It has an elliptical shape, with the shape determined by the covariance matrix Σ.

4. The parameters μ and Σ uniquely determine the distribution.One of the most important applications of the multivariate normal distribution is in statistical inference and data analysis. It is commonly used to model the joint distribution of a set of continuous random variables.

**Applications**

**1.Statistical Inference**: The multivariate normal distribution is used in hypothesis testing, confidence interval estimation, and regression analysis involving multiple variables.

**2.Machine Learning**: Many algorithms, such as Gaussian mixture models and linear discriminant analysis, assume that the data follows a multivariate normal distribution.

**3.Finance**: In portfolio theory, the returns of multiple assets are often modeled as a multivariate normal distribution to assess risks and correlations.

**4.Quality Control**: Multivariate normal distributions are used in control charts and process optimization in manufacturing settings.

**4 a) List the features of multivariate normal distribution.6M**

A) The features of the multivariate normal distribution are:

1**. Symmetry:** The distribution is symmetric around its mean vector.

2. **Elliptical shape:** The shape of the distribution is determined by the covariance matrix, which reflects the degree of correlation among the random variables.

3. **Peak at the mean:** The distribution has a maximum at the mean vector, indicating that it is most likely that the random variables take values near the mean.

4**. Uniquely determined by mean and covariance:** The multivariate normal distribution is completely determined by its mean vector and covariance matrix.

5. **Different values of covariance matrix produce different shapes: When** the covariance matrix is diagonal, the distribution is a product of independent univariate normal distributions. When the covariance matrix is non-diagonal, the distribution is elliptical and the correlation between variables must be taken into account.

6. **Conditional and marginal distributions are also normal:** If a subset of the variables is fixed, the conditional distribution of the remaining variables is also normal. Similarly, the marginal distribution of any subset of the variables is also normal.

7. **Independence of Variables**: If variables are independent, their joint distribution is a product of their individual distributions.

8. **Linear Transformations**: Linear combinations of multivariate normal variables yield another multivariate normal variable.

b) Discuss the applications of multivariate normal distribution.6M

A) The multivariate normal distribution has a wide range of applications in various fields, some of which are:

1. Finance: In finance, the multivariate normal distribution is used to model asset returns and portfolio optimization. It is also used in risk management and asset pricing models.

2. Engineering: In engineering, the multivariate normal distribution is used to model the behavior of complex systems, such as electronic circuits, control systems, and manufacturing processes.

3. Social sciences: In the social sciences, the multivariate normal distribution is used to analyze data from surveys and experiments, and to model relationships between variables in fields such as psychology and sociology.

4. Medical research: In medical research, the multivariate normal distribution is used to model the distribution of patient characteristics, such as age, sex, and disease severity, and to analyze the relationships between these characteristics and health outcomes.

5. Image processing: In image processing, the multivariate normal distribution is used to model the distribution of pixel values in images, and to segment and classify images based on their features.

6. Machine learning: In machine learning, the multivariate normal distribution is used as a building block for many models, such as Gaussian mixture models, Bayesian networks, and Hidden Markov models.

**5) Estimate and explain tuning complexity.12M**

A) Tuning complexity refers to the challenges and considerations involved in optimizing hyperparameters in machine learning models. Proper tuning is crucial for achieving the best performance from a model. Here's a detailed explanation of what tuning complexity entails:

**1. Definition of Tuning Complexity**

Tuning complexity arises from the need to adjust hyperparameters—parameters that govern the training process and structure of machine learning algorithms. These hyperparameters can significantly affect model performance, and finding the optimal values often involves a complex, resource-intensive process.

**2. Factors Contributing to Tuning Complexity**

- **Number of Hyperparameters**: As the number of hyperparameters increases, the search space for optimal values grows exponentially, making it more challenging to explore effectively.
- **Interactions Between Hyperparameters**: Hyperparameters can interact in non-linear ways, meaning the effect of one hyperparameter on model performance may depend on the values of others. This complicates the tuning process because optimizing one hyperparameter without considering others may not yield the best overall performance.
- **Model Complexity**: More complex models (like deep neural networks) often have many hyperparameters (e.g., learning rate, batch size, number of layers) that need careful tuning, which increases complexity.
- **Data Size and Dimensionality**: Larger datasets and higher-dimensional feature spaces can lead to longer training times, making hyperparameter tuning more time-consuming.

- **Performance Evaluation**: Determining the best hyperparameter settings typically requires evaluating model performance through methods like cross-validation, which itself can be computationally intensive.

**3. Methods for Hyperparameter Tuning**

- **Grid Search**: This exhaustive method evaluates all combinations of a predefined set of hyperparameter values. While comprehensive, it can be inefficient, especially with many hyperparameters.
- **Random Search**: This method samples random combinations of hyperparameters. It can be more efficient than grid search, particularly for high-dimensional spaces, as it allows exploration of a broader search area.
- **Bayesian Optimization**: This probabilistic model-based approach builds a surrogate model to find the optimal hyperparameters more efficiently. It balances exploration and exploitation, making it suitable for complex tuning tasks.
- **Gradient-Based Optimization**: Techniques like Hyperband or Successive Halving leverage early stopping to discard poorly performing configurations, thus focusing resources on promising candidates.

**4. Challenges in Tuning Complexity**

- **Computational Resources**: Tuning can be computationally expensive, requiring significant time and resources, particularly with large datasets and complex models.
- **Overfitting Risk**: If hyperparameters are tuned too closely to the training data, the model may overfit, leading to poor generalization on unseen data.
- **Interpretability**: With many hyperparameters, understanding how changes affect model performance can become difficult, complicating the tuning process.

**5. Strategies to Manage Tuning Complexity**

- **Prior Knowledge**: Utilizing domain knowledge to set initial ranges for hyperparameters can reduce the search space and improve tuning efficiency.
- **Automated Tuning Tools**: Libraries like Optuna or Hyperopt can automate the tuning process, using advanced techniques to find optimal hyperparameter settings.
- **Regularization Techniques**: Applying regularization can help manage model complexity, making tuning more straightforward.

**6 a) Analyze some features of multivariate normal distribution.6M**

**A)** The features of the multivariate normal distribution are:

1**. Symmetry:** The distribution is symmetric around its mean vector.

2. **Elliptical shape:** The shape of the distribution is determined by the covariance matrix, which reflects the degree of correlation among the random variables.

3. **Peak at the mean:** The distribution has a maximum at the mean vector, indicating that it is most likely that the random variables take values near the mean.

4**. Uniquely determined by mean and covariance:** The multivariate normal distribution is completely determined by its mean vector and covariance matrix.

5. **Different values of covariance matrix produce different shapes: When** the covariance matrix is diagonal, the distribution is a product of independent univariate normal distributions. When the covariance matrix is non-diagonal, the distribution is elliptical and the correlation between variables must be taken into account.

6. **Conditional and marginal distributions are also normal:** If a subset of the variables is fixed, the conditional distribution of the remaining variables is also normal. Similarly, the marginal distribution of any subset of the variables is also normal.

7. **Independence of Variables**: If variables are independent, their joint distribution is a product of their individual distributions.

8. **Linear Transformations**: Linear combinations of multivariate normal variables yield another multivariate normal variable.

**b) Determine few parameter estimation techniques.6M**

A) Here are some commonly used parameter estimation techniques:

1. Maximum Likelihood Estimation (MLE): It is a widely used method for estimating the parameters of a statistical model. The maximum likelihood estimate is the value of the parameter that maximizes the likelihood function.

2. Bayesian Estimation: It is a method of estimating the parameters of a statistical model by incorporating prior knowledge about the parameters.

3. Least Squares Estim ation (LSE): It is a method for estimating the parameters of a model by minimizing the sum of the squared residuals.

4. Maximum A Posteriori (MAP) Estimation: This method estimates the parameters of a model by maximizing the posterior probability of the parameters given the data and prior information.

5. Expectation-Maximization (EM) Algorithm: Expectation maximization the process that is used for clustering the data sample. It works on the concept of, starting with the random theory and randomly classified data along with the execution of below mentioned steps.

Step-1(“E”): In this step, Classification of current data using the theory that is currently being used is done.

Step-2(“M”): In this step, With the help of current classification of data, theory for that is generated. Thus EM means, Expected classification for each sample is generated used step-1 and theory is generated using step-2.

6.Kernel Density Estimation (KDE): It is a non-parametric method for estimating the probability density function of a random variable.

7.Generalized Method of Moments (GMM): It is a method for estimating the parameters of a model by matching the theoretical moments of the R Course Code: 20CS0904 20 model with the empirical moments of the data.

**7) Explain in detail about clustering and types of clustering.12M**

A) Clustering is the task of dividing the unlabeled data or data points into different clusters such that similar data points fall in the same cluster than those which differ from the others Types of Clustering Broadly speaking, clustering can be divided into two subgroups:

• Hard Clustering: In this, each input data point either belongs to a cluster completely or not.

• Soft Clustering: In this, instead of putting each input data point into a separate cluster, a probability or likelihood of that data point being in those clusters is assigned.

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

**1.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 centre is created in such a way that the distance between the data points of one cluster is minimum as compared to another cluster centroid.

**2.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 varying densities and high dimensions.

**3.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).

**4.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.

**5.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.

**Refer Diagram Through old PDF**

**8 a) Determine how multivariate regression is implemented.6M**

A)Multivariate regression is a technique used to measure the degree to which the various independent variable and various dependent variables are linearly related to each other. The relation is said to be linear due to the correlation between the variables. Once the multivariate regression is applied to the dataset, this method is then used to predict the behavior of the response variable based on its corresponding predictor variables.

Steps to achieve multivariate regression

Step 1: Select the features First, you need to select that one feature that drives the multivariate regression. This is the feature that is highly responsible for the change in your dependent variable.

Step 2: Normalize the feature Now that we have our selected features, it is time to scale them in a certain range (preferably 0-1) so that analysing them gets a bit easy. To change the value of each feature, we can use:

Step 3: Select loss function and formulate a hypothesis A formulated hypothesis is nothing but a predicted value of the response variable and is denoted by h(x). A loss function is a calculated loss when the hypothesis predicts a wrong value. A cost function is a cost handled for those wrongly predicting hypotheses.

Step 4: Minimize the cost and loss function Both cost function and loss function are dependent on each other. Hence, in order to minimize both of them, minimization algorithms can be run over the datasets. These algorithms then adjust the parameters of the hypothesis. One of the minimization algorithms that can be used is the gradient descent algorithm.

Step 5: Test the hypothesis The formulated hypothesis is then tested with a test set to check its accuracy and correctness.

b) Illustrate the uses of multivariate regression.6M

A) 1. Economics and Finance:

• Macroeconomic Modeling: Multivariate regression is frequently used in economics to model the relationships between various economic indicators such as GDP, inflation, unemployment, and interest rates.

• Financial Analysis: In finance, multivariate regression can be applied to analyze the factors influencing stock prices, bond yields, or other financial instruments.

2.Marketing and Business:

• Market Research: Companies use multivariate regression to understand the impact of multiple marketing variables (e.g., advertising expenditure, product features, pricing) on sales or market share.
• Customer Behavior Analysis: It helps analyze the factors affecting customer behavior, such as purchasing decisions, satisfaction, and loyalty.

3.Environmental Science:

• Climate Modeling: Multivariate regression is used to model the relationships between various climate variables, such as temperature, precipitation, and atmospheric pressure.

• Environmental Impact Assessments: Researchers use multivariate regression to assess the impact of multiple factors on environmental outcomes, such as air or water quality.

4. Medicine and Healthcare:

• Clinical Research: In medical studies, multivariate regression is employed to analyze the impact of multiple variables on health outcomes, taking into account factors like age, gender, and treatment regimens.

• Epidemiology: Researchers use multivariate regression to study the relationships between multiple risk factors and the occurrence of diseases.

5. Social Sciences:

• Education Research: Multivariate regression is used to analyze the impact of various factors (e.g., teaching methods, socio-economic status) on academic achievement.

• Psychology Studies: Psychologists use multivariate regression to examine the relationships between multiple variables, such as personality traits, environmental factors, and mental health outcomes.

6. Quality Control and Manufacturing:

• Process Optimization: In manufacturing, multivariate regression can be applied to optimize production processes by analyzing the impact of multiple variables on product quality.

• Quality Assurance: It is used to assess the influence of various factors on the quality of manufactured goods.

7. Sports Analytics:

• Performance Analysis: Multivariate regression can be used in sports analytics to understand the factors influencing team or individual performance, considering variables like player skills, coaching strategies, and environmental conditions.

8. Public Policy and Government:

• Policy Evaluation: Governments use multivariate regression to evaluate the impact of policies by considering various factors affecting social and economic outcomes.

• Social Welfare Programs: It helps analyze the effectiveness of social programs by examining the influence of multiple variables on outcomes like poverty rates or educational attainment.

9.) Explain in detail about a) Agglomerative Clustering

b) Hierarchical Clustering 6M

A) **a) Agglomerative Clustering**

Agglomerative clustering is a bottom-up approach to clustering, where each data point starts as its own cluster, and pairs of clusters are merged as one moves up the hierarchy. This method is widely used due to its simplicity and effectiveness in discovering the structure of data.

**Process**

1. **Initialization**: Each data point is considered as a separate cluster.
2. **Distance Calculation**: Calculate the distance (or similarity) between each pair of clusters. Common distance metrics include:
    - Euclidean distance
    - Manhattan distance
    - Cosine similarity
3. **Merge Clusters**: Identify the two closest clusters and merge them into a single cluster.
4. **Update Distance Matrix**: Recalculate distances between the newly formed cluster and all other clusters.
5. **Repeat**: Continue merging clusters until a stopping criterion is met, such as:
    - A predefined number of clusters is reached.
    - A certain distance threshold is exceeded.

**Dendrogram Visualization**
Agglomerative clustering can be visualized using a dendrogram, a tree-like diagram that illustrates the arrangement of clusters. The dendrogram shows the hierarchy of clusters and the distances at which merges occur.

**Applications**

- Market segmentation
- Document clustering
- Gene expression analysis

**Advantages**

- Does not require the number of clusters to be specified in advance.
- Provides a detailed hierarchy of clusters.

**Disadvantages**

- Computationally intensive for large datasets, as it involves calculating and updating a distance matrix.
- Sensitive to noise and outliers.

**b) Hierarchical Clustering**
Hierarchical clustering is a broader category that includes both agglomerative and divisive clustering methods. It builds a hierarchy of clusters either by merging smaller clusters into larger ones (agglomerative) or by splitting larger clusters into smaller ones (divisive).

**Types of Hierarchical Clustering**

1. **Agglomerative Hierarchical Clustering**: As described above, this is the most common form and starts with individual data points and merges them into larger clusters.

2. **Divisive Hierarchical Clustering**:
    - **Process**: Starts with all data points in one cluster and recursively splits them into smaller clusters.
    - **Algorithm**:
        1. Start with a single cluster containing all data points.
        2. Identify the cluster with the maximum dissimilarity.
        3. Split this cluster into two subclusters.
        4. Repeat until each data point is its own cluster or a stopping criterion is met.

**Dendrogram Representation**
Similar to agglomerative clustering, divisive clustering can also be represented using a dendrogram. This visual representation helps in understanding the relationships between clusters and determining the optimal number of clusters by observing where the structure changes significantly.

**Applications**

- Bioinformatics (e.g., phylogenetic trees)
- Social network analysis
- Image segmentation

**Advantages**

- Provides a comprehensive view of data structure and relationships.
- Does not require a predetermined number of clusters.
- Allows for varying levels of granularity in clustering.

**Disadvantages**

- Both agglomerative and divisive methods can be computationally expensive, especially for large datasets.
- The choice of linkage criteria can significantly influence results, leading to different cluster structures.

**10a) What is a Parameter? Describe parameter estimation method in detail.6M**

A) **Refer 2 question answer**

**10b) How can estimate the minimum mean square error estimation.6M**

**A) Minimum Mean Square Error (MMSE) estimation is a statistical approach used to estimate an unknown parameter or function by minimizing the expected value of the square of the error between the estimated and true values. Here's a detailed explanation of how to estimate MMSE.**

**1. Understanding MMSE Estimation**

**The MMSE estimator aims to minimize the expected squared difference between the estimated value θ^\hat{\theta}θ^ and the true value θ\thetaθ:**

**MMSE=E[(θ^−θ)2**

**Where:**

- **θ^\hat{\theta}θ^ is the estimated value.**
- **θ\thetaθ is the true parameter value.**
- **E\mathbb{E}E denotes the expected value.**

**2. Conditions for MMSE Estimation**

**To derive the MMSE estimator, we typically need to assume:**

- **The underlying probability distribution of the parameter.**
- **The distribution of the data given the parameter.**

**3. Minimize MSE: Differentiate with respect to g(X)g(X)g(X) and set to zero:**

$$g(X)=E[\theta|X]$$

**Thus, the MMSE estimator is the conditional expectation of θ\thetaθ given X.**

**4.Example:**

**5. Advantages of MMSE Estimation**

- **Optimality: MMSE estimators minimize the expected squared error, leading to efficient estimates.**
- **Incorporation of Prior Information: Bayesian approaches leverage prior knowledge about the parameter, leading to improved estimates when data is limited.**

**6. Applications**

- **MMSE estimation is widely used in fields such as signal processing, control systems, and economics, where estimating parameters with minimal error is critical.**