# UNIT 1

# INTRODUCTION TO MACHINE LEARNING

## QUESTIONS

## 1. Give the difference between supervised learning and unsupervised learning.

| Supervised learning | Unsupervised learning |
|---|---|
| Supervised learning algorithms are trained using labeled data. | Unsupervised learning algorithms are trained using unlabeled data. |
| Supervised learning model predicts the output. | Unsupervised learning model finds the hidden patterns in data. |
| In supervised learning, input data is provided to the model along with the output. | In unsupervised learning, only input data is provided to the model. |
| The goal of supervised learning is to train the model so that it can predict the output when it is given new data. | The goal of unsupervised learning is to find the hidden patterns and useful insights from the unknown dataset. |
| Supervised learning can be categorized in **Classification** and **Regression** problem s. | Unsupervised Learning can be classified in **Clustering** and **Associations** problem s. |
| It includes various algorithms such as Linear Regression, Logistic Regression, Support Vector Machine, Multi-class Classification, Decision tree, Bayesian Logic, etc. | It includes various algorithms such as Linear Regression, Logistic Regression, Support Vector Machine, Multi-class Classification, Decision tree, Bayesian Logic, etc. |

## 2. Describe application of machine learning in the field of healthcare.

**OR**

**Explain application of ML in healthcare sector.**

- Machine learning is used in healthcare to analyse medical images, predict patient outcomes and develop personalized treatment plans. It is also used in drug recovery and genomics research.
- Machine learning is used in medical imaging to analyze images from X rays, CT scans and MRIs. Algorithms can detect patterns and anomalies in the images, which can help with diagnosis and treatment planning.
- Machine learning can be used to analyze patient data and developer personalized treatment plants based on factors such as genetics, lifestyle and medical history.

## 3. Explain different tools and technology used in machine learning.

A. Python

- Python is a high level, general-purpose, object-oriented programming language. It has a huge number of libraries and framework: deep Python language comes with many libraries and frameworks that make coding easy this also saves a significant amount of time.
- The most popular libraries are NumPy, which is used for scientific calculations; SciPy for more advanced competitions; and scikit, for learning data mining and data analysis. Python is consistent when is anchor on simplicity which makes it most appropriate for machine learning.

B. R Programming

- R is a procedural programming language created by statisticians for statistics, specially for working with data. It is a language for statistical computing and data visualizations used widely by business analytics ,data analytics , data scientist and scientists.
- The R language has become popular because it lets researchers easily combine different machine learning techniques into a single program.

C. MATLAB

- MATLAB (matrix laboratory) makes machine learning easy with tools and functions for handling big data as well as apps to make machine learning accessible MATLAB is an ideal environment for applying machine learning data analytics.
- In MATLAB it takes fever lines of code and build a machine learning or deep learning model, without needing to be a specialist and techniques. MATLAB provide ideal environment for machine learning, through model training and deployment.

## 4. Describe basic concept of machine learning and its application.

- The basic concept of Machine Learning (ML) involves training machines or computer systems to learn from data and improve their performance on specific tasks without being explicitly programmed. ML algorithms use data inputs to automatically learn patterns, make predictions, or take actions. It involves steps such as data collection, preprocessing, model selection, training, evaluation, and prediction. ML finds applications in various domains, including image recognition, natural language processing, predictive analytics, healthcare, recommendation systems, and fraud detection.

- There are many application used in machine learning:
  1) Face Recognition
  2) Speech Recognition
  3) Healthcare
  4) Financial services
  5) Automatic language translation
  6) Traffic prediction
  7) Product Recommendation
  8) Weather Forecasting
  9) Stock market trading
  10) Online Fraud detection
  11) E-mail spam and Malware filtering
  12) Astrology

- Face Recognition:
- ➢ Face recognition task use for recognised other friends, family members and relative it also recognised photographs in different pose , hairstyle , makeup and without makeup and background colour.
- ➢ Face recognition system first decides structure of face and find midpoint of face after that based on pixel algorithm decide person's face.

- Healthcare:
- ➢ In healthcare system sensors are available in device this sensors use a data to excess of health of patient.
- ➢ Sensors provide real-time patient information like overall health condition, heartbeat, blood pressure and other parameters.
- ➢ Doctors use this information and predict disease of patient.

- Financial services:
- ➢ In financial sector identify all financial data and predict financial fraud by machine learning algorithm.
- ➢ This technology also used to identify opportunity for investment and trade.

**5. Define human learning. Give the types of human learning.**

- Human learning is the process of gaining information through observations.
- In our day to day life, we perform several activities. To do the activities in a proper way, we must have proper information related to that activity. Also, we always try to improve our activity by gaining more information or by previous experience related to that one. So, with more learning we can make our activity more efficient**.**

- **Types of human learning:**
  1. Learning under expert guidance.
  2. Learning guided by knowledge gain from experts.
  3. Learning by self.

6. **Describe different types of machine learning activities.**

- **Supervised Learning**:

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

➢ Some examples of Supervised learning are:

1) Predic realstate prize.
2) Find diseases and risk factoe.
3) Determine weather prediction.
4) Classifying bank transaction etc.

- **Unsupervised learning:**

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

➢ Example:

1) Creating customer group based on their behaviour.
2) Grouping eventary sale.
3) Customer who buy specific item might be impress or not.

- **Reinforecement Learning:**

➢ Reinforcement learning is a type of machine learning where an algorithm learns to make decisions by interacting with an environment, receiving rewards or penalties based on its actions. It learns through trial and error to maximize long-term rewards.

➢ Example:

1) Training robot to learn politic.
2) Controlling traffic like automatically.
3) Drive car automatically.

7. **Explain the concept of penalty and reward and reinforcement learning.**

- In reinforcement learning, penalties and rewards are used to guide the learning process of an algorithm.

  **Reward:** Rewards are positive feedback given to the algorithm when it takes actions that lead to desirable outcomes. Rewards encourage the algorithm to repeat those actions and learn strategies that maximize the total rewards received over time.

  **Penalty**: Penalties, also known as punishments, are negative feedback given to the algorithm when it takes actions that lead to undesirable outcomes. Penalties discourage the algorithm from repeating those actions and help it avoid making similar mistakes in the future.

**8. Explain the use of unsupervised learning in expression identification.**

- Unsupervised learning plays a crucial role in expression identification by allowing the algorithm to discover patterns and structures within facial image data without the need for labeled examples. Here's a brief explanation of its use:

**1.**Pattern Discovery: Unsupervised learning algorithms can identify hidden patterns in facial expressions by analyzing the data without prior knowledge of specific labels or classes. These algorithms can discover common features or combinations of features that correspond to different expressions.

**2.**Clustering: Unsupervised learning enables the algorithm to group similar facial expressions together based on their inherent similarities. By clustering similar patterns, the algorithm can identify different categories or types of expressions without explicit supervision.

**3.**Dimensionality Reduction: Unsupervised learning techniques such as dimensionality reduction can help simplify the complexity of facial image data by reducing the number of features while retaining essential information. This process aids in identifying relevant expression-related features and removing irrelevant noise from the data.

**9. Explain the use of Python in the field of machine learning.**

- Python is extensively used in the field of machine learning due to its robust libraries and user-friendly syntax. It offers a vast ecosystem of tools and frameworks, such as TensorFlow, PyTorch, and scikit-learn, that simplify tasks like data manipulation, model building, and evaluation.
- Python's simplicity and readability make it ideal for quick prototyping and experimentation. Its integration capabilities allow for seamless deployment of machine learning models on various platforms.
- Python's popularity, extensive community support, and rich resources make it the go-to language for machine learning practitioners and researchers.

## 10. Compare and contrast supervised learning and unsupervised learning.

**Supervised learning:**

- Definition: In supervised learning, the algorithm learns from labeled training data, where each data instance is paired with a corresponding target or output label.
- Goal: The goal is to learn a mapping between input features and the desired output labels to make predictions or classify new, unseen data.
- Training Process: During training, the algorithm adjusts its internal parameters to minimize the discrepancy between the predicted outputs and the true labels.
- Use of labels: Supervised learning requires labeled data, where the ground truth is known, to guide the learning process.
- Examples: Classification and regression are common supervised learning tasks.

**Unsupervised learning:**

- Definition: In unsupervised learning, the algorithm learns from unlabeled or partially labeled data, where no explicit output labels are provided.
- Goal: The goal is to discover patterns, structures, or relationships within the data without prior knowledge of the desired outcome.
- Training Process: Unsupervised learning algorithms explore the inherent structure in the data without explicit supervision, making use of clustering, dimensionality reduction, or anomaly detection techniques.
- Use of labels: Unsupervised learning does not rely on labeled data and does not require known output labels during training.
- Examples: Clustering, anomaly detection, and dimensionality reduction are common unsupervised learning tasks.

## 11. Describe the use of R in machine learning.

- R is a procedural programming language created by statisticians for statistics, specially for working with data .It is a language for statistical computing and data visualizations used widely by business analytics ,data analytics , data scientist and scientists.
- The R language has become popular because it lets researchers easily combine different machine learning techniques into a single program.

## 12. Give any 3 examples of supervised learning in industry 4.0.

**1.**Predictive Maintenance: Supervised learning is used to predict equipment failures and maintenance needs based on historical sensor data, enabling proactive maintenance scheduling and reducing unplanned downtime.
**2.**Quality control and Defect Detection: Supervised learning algorithms are employed to classify products as good or defective by learning from labeled examples, facilitating real-time defect detection and sorting during manufacturing processes.
**3.**Demand Forecasting and Inventory Management: Supervised learning is utilized to predict future demand for products by training models on historical sales data, aiding in optimized inventory management and ensuring sufficient stock levels while minimizing excess inventory costs.

**13. Define machine learning. Explain any 4 applications of machine learning in brief.**

- Machine Learning(ML) is a sub-field of Artificial Intelligence(AI) which systems to learn and make “prediction” baesd on the “historical data” or experiences.
- Machine Learning process is divided into three parts: Data inputs, abstraction, and generalization.

- Application of Machine Learning:

1. Face Recognition:

➢ Face recognition task use for recognised other friends, family members and relative it also recognised photographs in different pose , hairstyle , makeup and without makeup and background colour.
➢ Face recognition system first decides structure of face and find midpoint of face after that based on pixel algorithm decide person’s face.

2. Healthcare:
➢ In healthcare system sensors are available in device this sensors use a data to excess of health of patient.
➢ Sensors provide real-time patient information like overall health condition, heartbeat, blood pressure and other parameters.
➢ Doctors use this information and predict disease of patient.

3. Financial services:
➢ In financial sector identify all financial data and predict financial fraud by machine learning algorithm.
➢ This technology also used to identify opportunity for investment and trade.

4. Speech Recognition:

- While using Google, we get an option of "**Search by voice**," it comes under speech recognition, and it's a popular application of machine learning.
- Speech recognition is a process of converting voice instructions into text, and it is also known as "**Speech to text**", or "**Computer speech recognition**."

**14. Describe applications of unsupervised learning in detail.**

**1.**Clustering**:** Grouping similar data points together based on patterns.
**2.**Anamoly Detection: Identifying outliers or unusual instances in a dataset.
**3.**Dimensionality Reduction: Reducing the number of features while retaining important information.
**4.**Accosiation rule learning: Discovering relationships or associations among items in a dataset.
**5.**Generative Modeling: Creating new samples that resemble the training data.
**6.**Feature Learning: Learning useful representations or features from raw data.

# FUNDAMENTALS OF MACHINE LEARNING

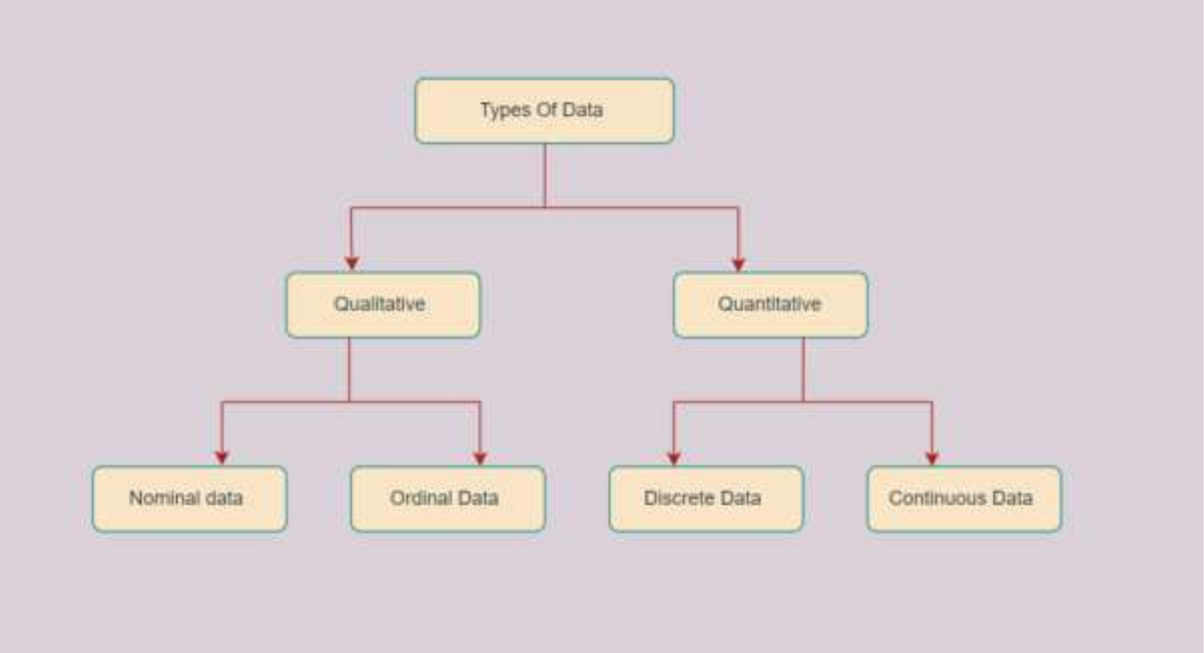
# ANSWER KEY OF CHAPTER – 2

## Q-1 Explain types of data with example. [03]

**Ans -There are two types of data: Qualitative and Quantitative data**, which are further classified into**:**

**The data is classified into four categories:**

- Nominal data.
- Ordinal data.
- Discrete data.
- Continuous data.



## Qualitative or Categorical Data

Qualitative or Categorical Data is data that can't be measured or counted in the form of numbers. These types of data are sorted by category, not by number. That's why it is also known as Categorical Data. These data consist of audio, images, symbols, or text. The gender of a person, i.e., male, female, or others, is qualitative data.

Example of qualitative data are :

- What language do you speak

- Favorite holiday destination

## The Qualitative data are further classified into two parts :

### Nominal Data

Nominal Data is used to label variables without any order or quantitative value. The color of hair can be considered nominal data, as one color can't be compared with another color.

**Examples of Nominal Data :**

- Colour of hair (Blonde, red, Brown, Black, etc.)
- Marital status (Single, Widowed, Married)

### Ordinal Data

Ordinal data have natural ordering where a number is present in some kind of order by their position on the scale. These data are used for observation like customer satisfaction, happiness, etc., but we can't do any arithmetical tasks on them.

**Examples of Ordinal Data :**

- Economic Status (High, Medium, and Low)
- Education Level (Higher, Secondary, Primary)

## Quantitative Data

Quantitative data can be expressed in numerical values, making it countable and including statistical data analysis. These kinds of data are also known as Numerical data.

**Examples of Quantitative Data :**

- Height or weight of a person or object
- Room Temperature

### The Quantitative data are further classified into two parts :

### Discrete Data

The term discrete means distinct or separate. The discrete data contain the values that fall under integers or whole numbers.

The discrete data are countable and have finite values; their subdivision is not possible. These data are represented mainly by a bar graph, number line, or frequency table.

### Examples of Discrete Data :

- Total numbers of students present in a class
- Numbers of employees in a company

### Continuous Data

Continuous data are in the form of fractional numbers. It can be the version of an android phone, the height of a person, the length of an object, etc. Continuous data represents information that can be divided into smaller levels. The continuous variable can take any value within a range.

### Examples of Continuous Data :

- Height of a person
- Speed of a vehicle

## Q-2 Write a short note on data Quality and Remediation. [07]

## Ans -Data Quality and Remediation

Data quality refers to the accuracy, completeness, consistency, and reliability of data. It is crucial for organizations to ensure that their data is of high quality to make informed decisions, support business operations, and maintain customer satisfaction. However, data quality issues can arise due to various factors such as data entry errors, system glitches, data integration problems, and outdated or inconsistent data sources.

Remediation is the process of identifying and resolving data quality issues to improve the overall quality and reliability of data. It involves a series of steps to assess, cleanse, and enhance data. Here is a brief overview of the key aspects of data quality and the remediation process:

**Data Assessment**: This step involves analyzing the current state of data to identify quality issues and their root causes. It includes evaluating data accuracy, completeness, consistency, integrity, and timeliness. Data profiling and data quality assessment tools can be used to gain insights into the data's characteristics and identify potential issues.

**Issue Identification**: Once the data assessment is complete, specific data quality issues are identified and documented. Common issues may include duplicate records, missing values, incorrect formatting, inconsistent data formats, or outliers. It is essential to prioritize the issues based on their impact on business processes and decision-making.

**Data Cleansing**: Data cleansing, also known as data scrubbing, involves the removal or correction of inaccurate, incomplete, or inconsistent data. Various techniques and tools are employed to cleanse the data, such as standardization, deduplication, validation, and normalization. Manual review and automated algorithms can be used to clean the data based on predefined rules and business requirements.

**Data Integration**: Data integration involves combining data from different sources or systems to create a unified view of the data. During the remediation process, integrating data from disparate sources may be necessary to ensure consistency and accuracy. Data integration techniques such as data mapping, transformation, and consolidation are employed to unify the data effectively.

**Data Enrichment**: Data enrichment involves enhancing the quality and value of data by adding additional information or attributes. It can include enriching data with external sources, performing data validation checks, and filling in missing data fields. Data enrichment techniques help improve data completeness, accuracy, and relevance.

**Monitoring and Maintenance**: Once the data quality issues have been remediated, ongoing monitoring and maintenance are crucial to sustaining data quality over time. Regular audits, data quality measurements, and feedback loops should be established to identify and address new data quality issues promptly.

Data quality remediation is an iterative process that requires continuous effort and commitment from organizations. By focusing on data quality, businesses can improve decision-making, enhance operational efficiency, and gain a competitive advantage in today's data-driven world.

## Q-3 Explain structure of box-plot. [03]

**Ans -** A box plot, also known as a box-and-whisker plot, is a graphical representation of a dataset that provides a visual summary of its distribution. It displays the minimum, first quartile (Q1), median (Q2), third quartile (Q3), and maximum values of a dataset.

Here's how the structure of a box plot is typically depicted:

- **Minimum (whisker):** The lowest value within 1.5 times the interquartile range (IQR) below the first quartile (Q1). It is represented by a vertical line extending downward from the box.
- **First Quartile (Q1):** The 25th percentile or the value below which 25% of the data falls. It marks the lower edge of the box.
- **Median (Q2):** The middle value of the dataset or the 50th percentile. It is represented by a horizontal line within the box.
- **Third Quartile (Q3):** The 75th percentile or the value below which 75% of the data falls. It marks the upper edge of the box.
- **Maximum (whisker):** The highest value within 1.5 times the IQR above the third quartile (Q3). It is represented by a vertical line extending upward from the box.
- **Outliers:** Data points that fall outside the whiskers, i.e., below the minimum or above the maximum. They are represented as individual data points or dots.

Box plots are useful for comparing distributions, identifying outliers, and gaining insights into the spread and central tendency of the data. They are commonly used in statistical analysis, data visualization, and exploratory data analysis.

## Q-4 Write a short note of feature subset selection method. [07]

**Ans -** Feature subset selection is a process in machine learning and data analysis that aims to identify and select a subset of relevant features from a larger set of available features. It involves choosing the most informative and discriminative features that contribute the most to the predictive accuracy or interpretability of a model. Here are a few key points about feature subset selection methods:

- **Importance of Feature Selection**: In many real-world scenarios, datasets may contain a large number of features, some of which may be irrelevant, redundant, or noisy. Feature selection helps in reducing the dimensionality of the data, improving computational efficiency, enhancing model interpretability, and mitigating the risk of overfitting.

- **Types of Feature Selection Methods:** There are various approaches to perform feature subset selection:

**a. Filter Methods:** These methods assess the relevance of features by examining their statistical properties or information gain independently of the learning algorithm. Examples include correlation-based feature selection, chi-square test, and mutual information.

**b. Wrapper Methods**: Wrapper methods evaluate feature subsets by incorporating the learning algorithm itself. They select subsets based on the performance of a specific learning algorithm using a defined evaluation criterion. Examples include recursive feature elimination (RFE) and forward/backward feature selection.

**c. Embedded Methods**: These methods integrate feature selection within the learning algorithm itself. They aim to find the most informative features during the training process. Examples include LASSO (Least Absolute Shrinkage and Selection Operator) and decision tree-based feature selection.

- **Evaluation Criteria:** When selecting a feature subset, it is crucial to have appropriate evaluation criteria. Common criteria include accuracy, area under the receiver operating characteristic curve (AUC-ROC), mean squared error (MSE), or other specific performance measures depending on the problem domain.
- **Trade-off between Subset Size and Performance:** Feature subset selection involves a trade-off between the number of selected features and the resulting model performance. Adding more features may improve the model's ability to capture complex relationships but can also increase the risk of overfitting. It is important to strike a balance between model complexity and generalization capability.
- **Iterative Process**: Feature subset selection is often an iterative process. Different methods and combinations of features need to be evaluated and compared to identify the optimal subset. Cross-validation techniques are commonly used to estimate the performance of different subsets and ensure generalizability.
- **Domain Knowledge:** Incorporating domain knowledge and expert insights can be valuable in guiding the feature selection process. Prior knowledge about the problem domain and the relationships between features can help in making informed decisions.

Feature subset selection is a critical step in the machine learning pipeline, as it can significantly impact the performance and interpretability of models. By selecting the most relevant features, it enables more efficient and accurate learning, improves model understanding, and facilitates better decision-making based on the underlying data.

## Q-5 Give the difference between qualitative data and quantitative data. [03]

**Ans -**

| Qualitative data | Quantitative data |
|---|---|
| ○ This data provides information about the quality of an object or information which cannot be measured.<br>○ Types- nominal and ordinal.<br>○ Narratives often make use of adjective and descriptive word to refer on appearance, color, texture, etc.<br>○ Descriptive form.<br>○ Eg: Team is well prepared.<br>: River is peaceful. | ○ This data realates to information about the quantity of an object, hence it can be measured.<br>○ Types- interval and ratio.<br>○ Measures quantity such as length, size, amount, etc.<br>○ Numerical form.<br>○ Eg: Team has 7 players.<br>: The river is 25 miles long. |

## Q-6 Write a short note on dimensionality reduction. [07]

**Ans-** Dimensionality reduction is a process in machine learning and data analysis that aims to reduce the number of features or variables in a dataset while preserving the essential information. It is particularly useful when dealing with high-dimensional data, where the number of features is large relative to the number of observations. Here's a brief overview of dimensionality reduction:

- **Motivation:** High-dimensional data can pose challenges in terms of computational complexity, overfitting, and difficulties in visualization and interpretation. Dimensionality reduction helps to alleviate these issues by transforming the data into a lower-dimensional representation that captures the most important information.
- **Techniques:**

a. Principal Component Analysis (PCA): PCA is a widely used linear dimensionality reduction technique. It identifies the directions (principal components) along which the data exhibits the most variance and projects the data onto a reduced set of orthogonal axes. This allows for a lower-dimensional representation of the data while preserving the maximum variance.

b. Linear Discriminant Analysis (LDA): LDA is a dimensionality reduction technique that focuses on maximizing the class separability in supervised learning problems. It aims to find a projection that maximizes the between-class scatter while minimizing the within-class scatter.

c. t-SNE (t-Distributed Stochastic Neighbor Embedding): t-SNE is a nonlinear dimensionality reduction technique that is particularly useful for visualizing high-dimensional data. It maps the high-dimensional data onto a lower-dimensional space while preserving the local structure and pairwise similarities between data points.

d. Autoencoders: Autoencoders are neural network-based models that learn to compress the input data into a lower-dimensional representation and then reconstruct it. They can capture nonlinear relationships and are effective in unsupervised dimensionality reduction.

- **Benefits:**

a. Reduced Computational Complexity: By reducing the dimensionality of the data, dimensionality reduction techniques can significantly reduce the computational cost associated with training and inference.

b. Improved Model Performance: Dimensionality reduction can help in reducing overfitting by removing noise and irrelevant features, leading to improved generalization and predictive performance.

c. Data Visualization and Interpretability: By transforming high-dimensional data into a lower-dimensional space, dimensionality reduction facilitates data visualization and interpretation, as the reduced representation can be easily visualized and analyzed.

Trade-offs: Dimensionality reduction involves a trade-off between preserving the essential information and discarding some of the less informative or redundant features. It is crucial to strike a balance to avoid losing critical details or introducing distortions in the data.

Dimensionality reduction is a valuable tool in data preprocessing and analysis, enabling more efficient and effective machine learning and data exploration. It helps in managing high-dimensional data, improving model performance, and gaining insights into complex datasets.

## Q-7 Explain types of machine learning with example. [07]

**Ans-** Machine learning can be broadly classified into three types: supervised learning, unsupervised learning, and reinforcement learning. Each type has its own characteristics and applications. Let's explore them with examples:

**Supervised Learning**: Supervised learning involves training a model on labeled data, where the input features (X) and their corresponding output labels (Y) are provided. The model learns from the labeled examples to make predictions or classify new, unseen data points.

Examples:

Classification: Given a dataset of emails labeled as spam or non-spam, a supervised learning model can be trained to classify new emails as spam or non-spam.

Regression: Given historical data of house prices along with their features (e.g., size, location), a supervised learning model can be trained to predict the price of a new house based on its features.

**Unsupervised Learning:** Unsupervised learning involves training a model on unlabeled data, where the model learns patterns, structures, or relationships in the data without explicit labels or guidance.

Examples:

Clustering: Given a dataset of customer purchase behavior, an unsupervised learning model can group similar customers together based on their purchasing patterns.

Dimensionality Reduction: By applying unsupervised learning techniques such as Principal Component Analysis (PCA), a model can reduce the dimensionality of a dataset while preserving its essential characteristics.

Reinforcement Learning: Reinforcement learning involves training an agent to interact with an environment and learn optimal actions through a trial-and-error process. The agent learns by receiving feedback in the form of rewards or penalties for its actions.

**Examples:**

Game Playing: Reinforcement learning can be used to train an agent to play games such as chess or go, where the agent learns the best moves through repeated gameplay and receiving rewards or penalties based on the game outcomes.

Robotics: Reinforcement learning can be applied to train robotic systems to perform specific tasks by providing rewards for successful completion and penalties for failures.

### Q-8 Explain in details the different strategies of addressing missing data values. [07]

**Ans-** Dealing with missing data values is a crucial step in data preprocessing. There are various strategies to address missing data, each with its own advantages and considerations. Here are some commonly used approaches:

- **Complete Case Analysis**: In this strategy, any data point with missing values is completely removed from the dataset. This approach is simple and convenient, especially when the missing data is small in proportion. However, it can lead to a loss of valuable information if the missing data is systematic or occurs in patterns related to the outcome variable.
- **Mean/Mode/Median Imputation:** Missing values are replaced with the mean (for numerical data), mode (for categorical data), or median (for skewed data) of the available values for that variable. This strategy is quick and easy to implement, but it assumes that the missing values have the same statistical properties as the observed values. It can lead to biased estimates and underestimate the variability of the data.
- **Forward or Backward Fill:** Missing values are replaced with the last observed value (forward fill) or the next observed value (backward fill). This method is commonly used for time series data or datasets with a natural ordering. However, it may introduce autocorrelation in the data, especially if the missing values are not randomly distributed.
- **Hot Deck Imputation**: Missing values are imputed by randomly selecting a value from a similar observed record (donor record) in the dataset. The similarity can be determined based on variables such as distance, clustering, or matching criteria. This approach helps preserve the distribution and relationships in the data but requires careful selection of the donor record.
- **Multiple Imputation**: Multiple imputation involves creating multiple plausible imputed datasets based on the observed values and statistical

models. Each dataset is imputed separately, and the results are pooled to obtain the final estimates and measures of uncertainty. Multiple imputation takes into account the variability introduced by imputing missing values and provides more accurate estimates compared to single imputation methods.

- Model-based Imputation: Model-based imputation involves using statistical models to estimate missing values based on observed variables. Methods such as regression models, decision trees, or neural networks can be used to predict missing values. This approach captures the relationships between variables but requires a suitable model and assumes that the missingness mechanism is ignorable or correctly specified.
- **Domain-specific Imputation:** In some cases, domain knowledge or expert judgment can be used to impute missing values. This approach relies on subject matter expertise to estimate missing values based on logical or contextual reasoning specific to the data domain.

It's important to note that the choice of strategy depends on the nature of the missing data, the underlying assumptions, and the specific requirements of the analysis. No single strategy is universally optimal, and the selected approach should be carefully evaluated and validated based on the specific dataset and research question.

## Q-9 Describe machine learning activities in detail. [07]

**Ans-** Machine learning activities involve a series of steps and tasks to develop, train, evaluate, and deploy machine learning models. Here's a detailed description of the key activities involved in machine learning:

- **Problem Definition:** The first step in any machine learning project is to clearly define the problem statement or objective. This involves understanding the business or research goals, identifying the relevant variables, and determining the type of machine learning task (e.g., classification, regression, clustering) required to solve the problem.
- **Data Collection**: The next step is to gather the necessary data for the machine learning task. This may involve collecting data from various sources, such as databases, APIs, or external datasets. Care should be taken to ensure data quality, relevance, and representativeness for the problem at hand.
- **Data Preprocessing**: Once the data is collected, it often requires preprocessing to handle missing values, handle outliers, normalize or scale features, handle categorical variables, and address other data-

specific issues. This step helps in preparing the data for effective model training and analysis.

- **Feature Engineering:** Feature engineering involves transforming the raw data into a set of meaningful features that can effectively represent the underlying patterns and relationships in the data. This may involve techniques such as feature selection, dimensionality reduction, creating interaction terms, or deriving new features from existing ones. Well-crafted features can significantly impact the model's performance.
- **Model Selection and Training**: The selection of an appropriate machine learning algorithm depends on the problem type, available data, and desired performance metrics. Various algorithms, such as decision trees, support vector machines, neural networks, or ensemble methods, may be considered. The selected algorithm is trained on the labeled data to learn the underlying patterns and relationships between the features and the target variable.
- **Model Evaluation:** Once the model is trained, it needs to be evaluated to assess its performance and generalization capability. This involves using evaluation metrics such as accuracy, precision, recall, F1-score, or area under the curve (AUC-ROC) to measure the model's predictive accuracy. Cross-validation techniques and validation datasets can be used to estimate the model's performance on unseen data and to detect overfitting.
- **Model Tuning**: Model tuning aims to optimize the model's performance by fine-tuning the hyperparameters, such as learning rate, regularization, number of hidden layers, or feature selection thresholds. This can be done using techniques like grid search, random search, or Bayesian optimization to find the best combination of hyperparameters.
- **Model Deployment**: Once the model is trained and validated, it can be deployed for real-world use. This may involve integrating the model into an application, deploying it as a web service or API, or integrating it into an existing software infrastructure. Deployment considerations include scalability, real-time prediction, monitoring, and maintenance.
- **Model Monitoring and Maintenance:** Machine learning models require ongoing monitoring and maintenance to ensure their performance and relevance over time. This involves monitoring the model's performance on new data, retraining the model periodically with fresh data, and making necessary updates or re-evaluations as the problem or data distribution evolves.

Throughout these activities, it is essential to follow ethical considerations, ensure data privacy and security, and maintain transparency and interpretability of the models.

Machine learning activities require a combination of domain knowledge, data analysis skills, programming expertise, and critical thinking to effectively address complex problems and leverage the power of data-driven decision-making.

## Q-10 Write a short note on histogram. [04]

**Ans-** A histogram is a graphical representation that organizes and displays the distribution of a continuous variable. It provides a visual summary of the underlying data and helps understand its shape, central tendency, variability, and outliers. Here are some key points about histograms:

- **Representation:** A histogram consists of a series of bars, where each bar represents a specific range or bin of values. The height of each bar represents the frequency or count of data points falling within that bin. The width of the bars may vary, depending on the data and the desired level of detail.
- **Data Binning**: The range of values is divided into equal-sized intervals or bins. The number of bins influences the level of detail in the histogram and can impact the interpretation. Too few bins may oversimplify the distribution, while too many bins can result in noise or overfitting to small variations.
- **Visualizing Distribution**: Histograms are particularly useful for understanding the shape of the data distribution. Common distribution shapes include bell-shaped (normal distribution), skewed (positively or negatively), bimodal (two peaks), or uniform (no apparent pattern). These visual cues provide insights into the central tendency and spread of the data.
- **Identifying Central Tendency**: The histogram allows you to identify the central tendency of the data, such as the mean, median, or mode. The highest bar often corresponds to the mode, while the midpoint of the histogram represents the median.
- **Detecting Outliers:** Outliers, which are extreme values that deviate significantly from the majority of the data, can be identified on a histogram. Outliers appear as isolated bars far away from the main bulk of the distribution, indicating unusual or rare occurrences.

- **Data Skewness**: Histograms can reveal the skewness of the data, which is the asymmetry or lack of symmetry in the distribution. Positive skewness (right-skewed) indicates a longer tail on the right side, while negative skewness (left-skewed) indicates a longer tail on the left side. Symmetric distributions have zero skewness.
- **Interpretation:** Histograms provide an intuitive visual representation of the data, enabling quick insights and comparisons. They are widely used in exploratory data analysis, quality control, data visualization, and hypothesis testing.

In summary, histograms are a valuable tool for understanding the distribution, central tendency, variability, and outliers in continuous data. They provide a concise and visual summary of data characteristics, facilitating data-driven decision-making and further analysis.

## Q-11 Define outliers. Give one example. [03]

**Ans-** Outliers are data points that significantly deviate from the majority of the other data points in a dataset. They are observations that lie unusually far away from the central tendency of the data and can have a disproportionate impact on statistical analyses and modeling. Outliers can arise due to various reasons, such as measurement errors, data entry mistakes, natural variations, or rare events. Here's an example to illustrate outliers:

Suppose you have a dataset representing the ages of a group of individuals in a specific community. The ages of the individuals are as follows: 32, 34, 35, 33, 34, 30, 36, 38, 37, 31, 33, 40, 31, 33, 38, 45, 32, 33, 34.

In this dataset, the majority of the ages cluster around the mid-thirties, with values ranging from 30 to 40. However, there are two extreme values: 40 and 45. These values deviate significantly from the rest of the data and can be considered outliers. They represent individuals who are relatively older compared to the majority of the group.

Identifying outliers is important because they can affect statistical analyses, such as calculating the mean or standard deviation, and can impact the performance and accuracy of machine learning models. Proper handling of outliers, whether by removing, transforming, or treating them separately, is necessary to ensure robust and meaningful data analysis.

## Q-12 Find mean and median of the following data: 4, 5, 7, 8, 8, 9 ,11 ,12 ,14. [04]

**Ans-** To find the mean and median of the given dataset, we can follow these steps:

Arrange the data in ascending order: 4, 5, 7, 8, 8, 9, 11, 12, 14.

Mean Calculation: Add up all the numbers in the dataset and divide the sum by the total count of numbers.

Mean = (4 + 5 + 7 + 8 + 8 + 9 + 11 + 12 + 14) / 9 = 78 / 9 = 8.6667 (rounded to four decimal places)

Therefore, the mean of the given dataset is approximately 8.6667.

Median Calculation: To find the median, we locate the middle value in the ordered dataset. Since there are 9 numbers, the middle value will be the 5th value.

Median = 8

Therefore, the median of the given dataset is 8.

In summary: Mean = 8.6667 (rounded to four decimal places) Median = 8

## Q-13 Find standard deviation of the following data: 5, 10, 15, 20, 25, 29.[04]

**Ans-** To find the standard deviation of a dataset, we can follow these steps:

Calculate the mean of the dataset: Mean = (5 + 10 + 15 + 20 + 25 + 29) / 6 = 104 / 6 = 17.3333 (rounded to four decimal places)

Calculate the squared difference from the mean for each value in the dataset:

For 5: (5 - 17.3333)^2 = 154.1111

For 10: (10 - 17.3333)^2 = 53.5306

For 15: (15 - 17.3333)^2 = 5.5306

For 20: (20 - 17.3333)^2 = 7.1111

For 25: (25 - 17.3333)^2 = 59.5306

For 29: (29 - 17.3333)^2 = 135.1111

Calculate the average of the squared differences: Average = (154.1111 + 53.5306 + 5.5306 + 7.1111 + 59.5306 + 135.1111) / 6 = 414.9259 / 6 = 69.1543 (rounded to four decimal places)

Take the square root of the average to obtain the standard deviation: Standard Deviation = √69.1543 = 8.3139 (rounded to four decimal places)

Therefore, the standard deviation of the given dataset is approximately 8.3139.

## Q-14 Explain data pre-processing in brief. [04]

**Ans-** Data preprocessing is a crucial step in data analysis and machine learning. It involves transforming raw data into a clean, organized, and structured format that is suitable for further analysis. Data preprocessing includes several tasks to handle missing values, outliers, data normalization, feature scaling, and handling categorical variables. Here are some key steps involved in data preprocessing:

- **Data Cleaning**: This step involves handling missing data, outliers, and inconsistent or incorrect data entries. Missing data can be addressed through imputation or removal, depending on the extent and pattern of missingness. Outliers may be identified and treated, considering the context and requirements of the analysis.
- **Data Integration:** Data integration involves combining data from multiple sources into a single dataset. It includes resolving inconsistencies, standardizing variables, and ensuring compatibility and coherence across different data sources.
- **Data Transformation**: Data transformation aims to improve the distributional properties of the data and make it more suitable for analysis. It may involve transforming variables using mathematical functions, such as logarithmic or exponential transformations, to normalize skewed distributions. Non-linear transformations or encoding schemes may also be applied to improve the representation of the data.
- **Data Reduction**: Data reduction techniques are employed to reduce the dimensionality of the dataset while preserving relevant information. This helps in avoiding the curse of dimensionality, improving computational efficiency, and addressing issues related to overfitting. Techniques such as principal component analysis (PCA), feature selection, or feature extraction methods can be applied to reduce the number of variables.
- **Data Discretization**: Data discretization involves converting continuous variables into categorical or ordinal variables by creating bins or intervals. This can simplify the analysis, handle non-linear relationships, and reduce the impact of outliers. Discretization methods include equal-width or equal-frequency binning, clustering-based methods, or decision tree-based methods.

- **Data Normalization and Scaling**: Normalization and scaling techniques are used to bring the data within a specific range or distribution to facilitate fair comparisons and avoid dominance of certain variables. Common techniques include min-max scaling, z-score normalization, or robust scaling.
- **Handling Categorical Variables**: Categorical variables need to be appropriately encoded or transformed to numerical representations for analysis. This can be achieved through techniques such as one-hot encoding, label encoding, or ordinal encoding, depending on the nature of the categorical variable and the requirements of the analysis.

Data preprocessing plays a vital role in improving the quality and reliability of data analysis and machine learning models. It helps in addressing data quality issues, reducing bias, improving interpretability, and enabling accurate and meaningful insights from the data. The specific preprocessing steps applied depend on the characteristics of the data and the requirements of the analysis or modeling task.

## Q-15 Explain the factors which leads to the data quality issues. [03]

**Ans-** Data quality issues can arise from various factors throughout the data lifecycle. Understanding these factors is crucial for identifying and mitigating data quality problems. Here are some common factors that can lead to data quality issues:

- **Data Entry Errors**: Human errors during data entry can introduce mistakes such as typos, incorrect values, or missing information. These errors can propagate throughout the data and impact its quality.
- **Incomplete or Missing Data**: Data may be incomplete due to various reasons such as non-response, system failures, or data collection limitations. Missing data can introduce biases and affect the representativeness and integrity of the dataset.
- **Inconsistent Data:** Data inconsistencies can occur when different sources or systems use varying formats, standards, or definitions. Inconsistencies in variables, units of measurement, or coding schemes can hinder data integration and analysis.
- **Data Duplication**: Duplicate records or entries can distort the analysis results and affect the accuracy of statistical calculations. Duplication can arise from data integration processes, system errors, or merging of datasets.

- **Outliers and Anomalies**: Outliers are extreme values that deviate significantly from the majority of the data. They can arise due to measurement errors, data processing issues, or rare events. Outliers can impact statistical analysis and modeling outcomes.
- **Bias and Sampling Issues**: Bias can occur in the data collection process when certain groups or characteristics are overrepresented or underrepresented, leading to skewed results. Sampling errors, such as non-random sampling or selection bias, can affect the generalizability of the findings.
- **Data Integration Challenges**: When integrating data from multiple sources, differences in data formats, structures, or definitions can lead to data quality issues. Merging and aligning data can be complex, and inconsistencies may arise if not properly addressed.
- **Data Storage and Transfer Issues:** Data can get corrupted, altered, or lost during storage or transfer processes. Technical issues, such as hardware failures, software bugs, or network problems, can affect data integrity and quality.
- **Data Security and Privacy Concerns**: Data quality can be compromised if there are security breaches, unauthorized access, or data privacy violations. These issues can impact data accuracy, reliability, and compliance with privacy regulations.
- **Time-Related Issues**: Data quality can deteriorate over time due to outdated or obsolete information. Timeliness, relevance, and currency of the data need to be considered to ensure its quality.

## FML

## Ch: 3 Modeling and Evaluation

Q.1: **Describe K-fold cross validation method with example.[07]**

Ans: •Repeated Holdout Method is an iteration of the holdout method i.e it is the repeated execution of the holdout method.

•This method can be repeated — 'K' times/iterations.

•In this method, we employ random sampling of the dataset. The dataset is partitioned randomly and not on the basis of any formula.

**Example** – Consider a dataset, which is stratified into then training set and test set, randomly. We repeat the holdout method for 'K' iterations. Let us assume K=3.

•The shaded portions in the above iterations are the test sets and the unshaded portions are the training sets, which are obtained after the stratification of the dataset.

•In the first iteration 'ITERATION – 01', a classifier is constructed on the basis of the data items/example that belongs to the training set. The classifier after construction is applied to the test set. The result obtained is an error estimate, say 'E1'.

•In the second iteration 'ITERATION – 02', the first iteration is randomly arranged. A classifier is now constructed on the basis of training set data items/examples. The classifier after construction is applied to the test set. The result obtained is an error estimate, say 'E2'.

• In the third iteration 'ITERATION – 03', the second iteration is randomly arranged. A classifier is now constructed on the basis of training set data items/examples. The classifier after construction is applied to the test set. The result obtained is an error estimate, say 'E3'.

• The iterations are thus repeated 'K=3' times.

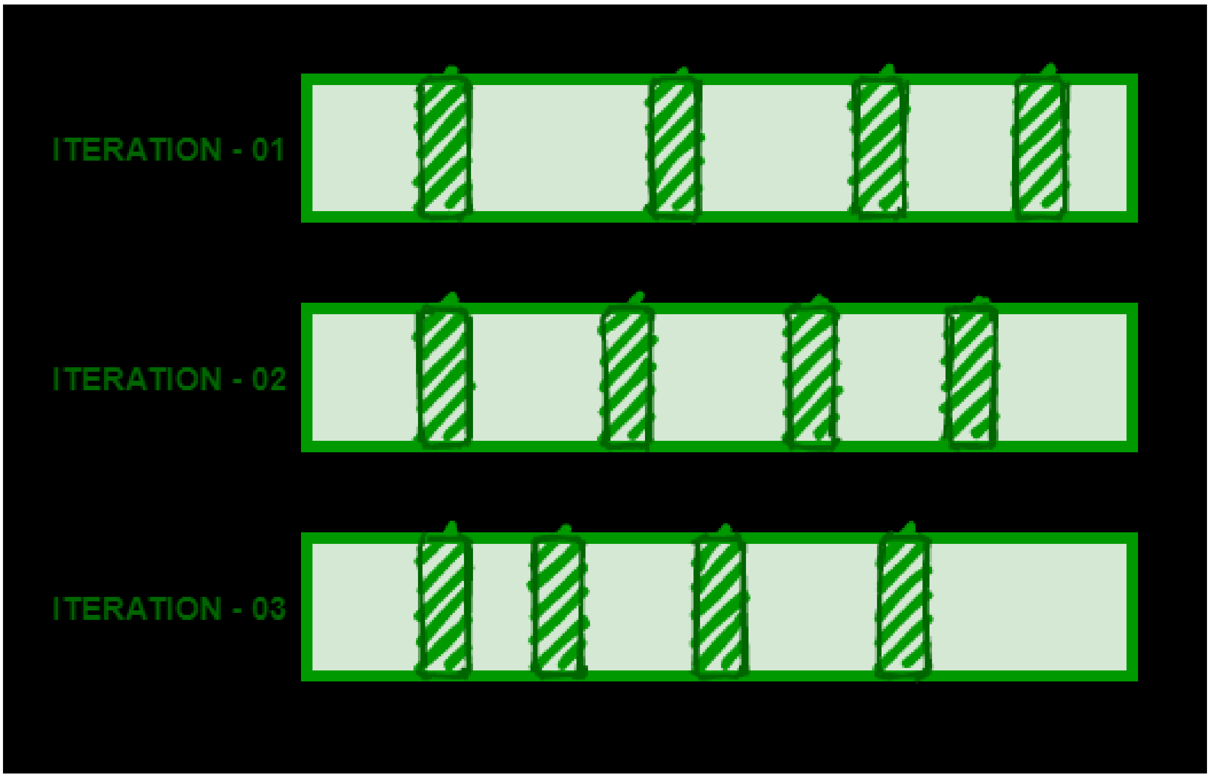
• To find the overall error estimate, we can use the formula –

**Problem**: Overlapping test set problem.

•Since we partition the dataset randomly into a training set and test

set, there are some data items/examples that could not be placed

in the training set at all.



Q.2: Consider the following confusion matrix of the win/loss prediction of cricket match. Calculate the accuracy, error rate, sensitivity, specificity, precision, recall and F-measure of the model.[07]

| | Actual win | Actual loss | |
|---|---|---|---|
| Predicted win | (TP) 80 | (FP) 6 | Total =86 |
| Predicted loss | (FN) 5 | (TN) 9 | Total=14 |
| | Total: 85 | Total: 15 | |

Ans: **Accuracy:**

⇨ Accuracy is the proportion of correct predictions out of the total predictions.

Total predictions = Sum of all values in the confusion matrix = 80 + 6 + 5 + 9 = 100 Correct predictions = Number of true positives + Number of true negatives = 80 + 9 = 89

Accuracy = (TP+TN)/(TP+TN+FP+FN)

= (9+80)/ (9+80+6+5)
= 89/100
=0.89(89% Accuracy of the model)

**Error rate:**

- Error Rate is the proportion of incorrect predictions out of the total predictions.

Error Rate = 1 – Accuracy = 1 - 0.89 = 0.11 or 11%

**sensitivity (True Positive Rate or Recall):**

- sensitivity measures the proportion of actual positive cases that are correctly identified as positive.

**Sensitivity=TP/(TP+FN)**

**=80/ (80+5)**
**=80/85**
**=0.941**

**Specificity:**

- Specificity measures the proportion of actual negative cases that are correctly identified as negative.

**Specifity = TN/TN+FP**
**=9/9+6**
**=0.6**

**Precision:**

- precision measures the proportion of predicted positive cases that are actually positive.

**Precision=TP/TP+FN**

**=80/80+5**

**=80/85**

**=0.94**

- **Recall:**

  Recall measures the proportion of actual positive cases that are correctly identified as positive.

  **Recall=TP/TP+FN**

**=80/80+5**

**=80/85**

**=0.94**

⇨ **F-measure:**

The F-measure combines precision and recall into a single metric. It is the harmonic mean of precision and recall.

**F-measure**=2(pre*recall)/(pre+ recall)

**=2(0.94*0.94)/(0.94+0.94)**

**=2(0.4512/1.88)**

**=8.33**

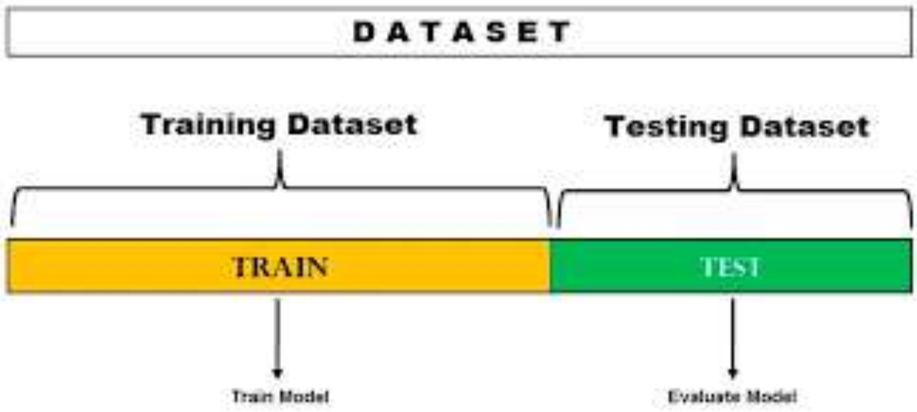
**To summarize:**

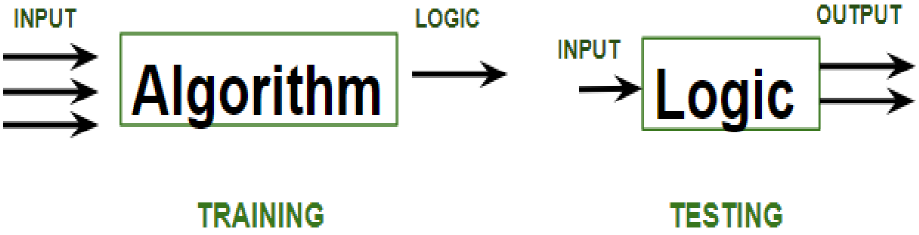
**Accuracy: 0.89 or 89%**
**Error Rate: 0.11 or 11%**
**Sensitivity/Recall: 0.941**
**Specificity: 0.6**
**Precision: 0.94**
**F-measure: 8.33**

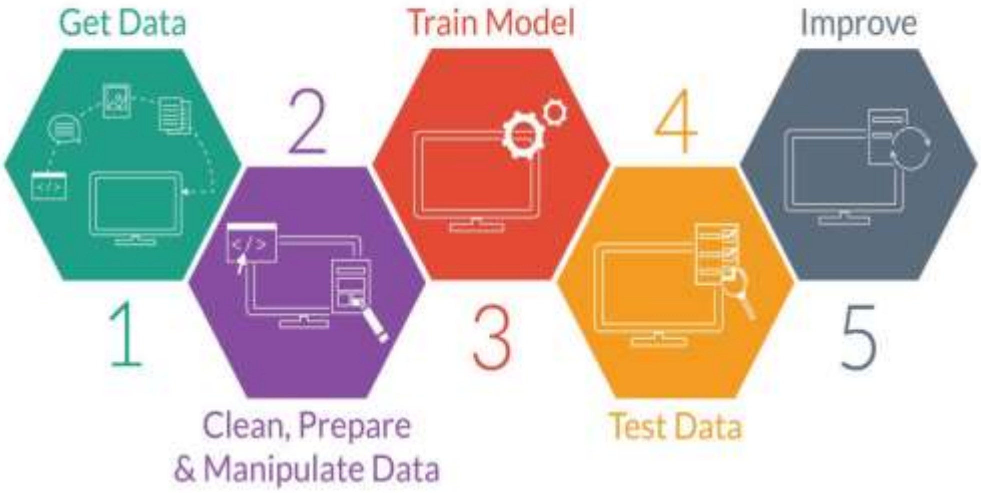
**Q.3 Explain Holdout method in detail.[07]**

**Ans**: Holdout Method is the simplest sort of method to evaluate a classifier. In this method, the data set (a collection of data items or examples) is separated into two sets, called the Training set and Test set.

A classifier performs function of assigning data items in a given collection to a target category or class.







**Example:**

⇨ E-mails in our inbox being classified into spam and non-spam. Classifier should be evaluated to find out, it's accuracy, error rate, and error estimates. It can be done using various methods. One of most primitive methods in evaluation of classifier is 'Holdout Method.

⇨ In the holdout method, data set is partitioned, such that – maximum data belongs to training set and remaining data belongs to test set.

**Example:**

⇨ A student 'gfg ' is coached by a teacher. Teacher teaches her all possible topics which might appear for exam. Hence, she tends to commit very less mistakes in exam, thus performing well.

⇨ If more training data are used to construct a classifier, it qualifies any data used from test set, to test it (classifier).

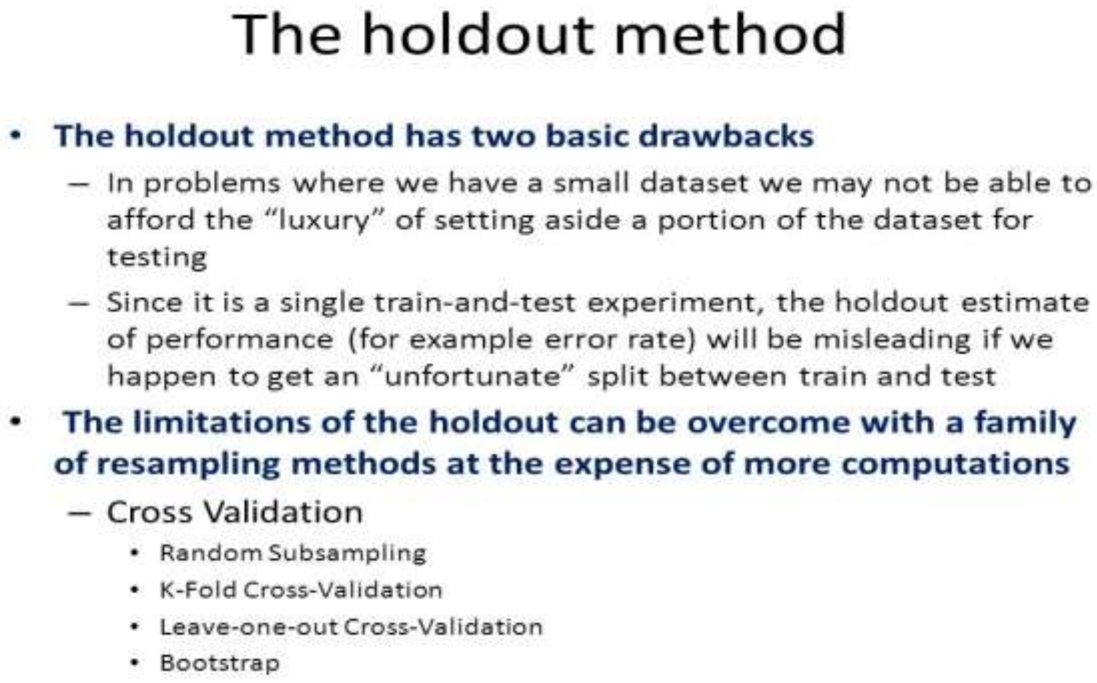
⇨ If more number of data items are present in test set, such that they are used to test classifier built using training set. We can observe more accurate evaluation of classifier with respect to its accuracy, error rate and estimation.

**Problem:**

⇨ During partitioning of whole data set into 2 parts i.e., training set and test set, if all data items belonging to class – GFG1, are placed in test set entirely, such that none of data items of class GFG1 are in training set. It is evident, that model/classifier built, is not trained using data items of class – GFG1.

**Solution:**

⇨ Stratification is a technique, using which data items belonging to class – GFG1 are divided and placed into two data sets i.e training set and test set, equally. Such that, model/classifier is trained by data items belonging to class -GFG1.

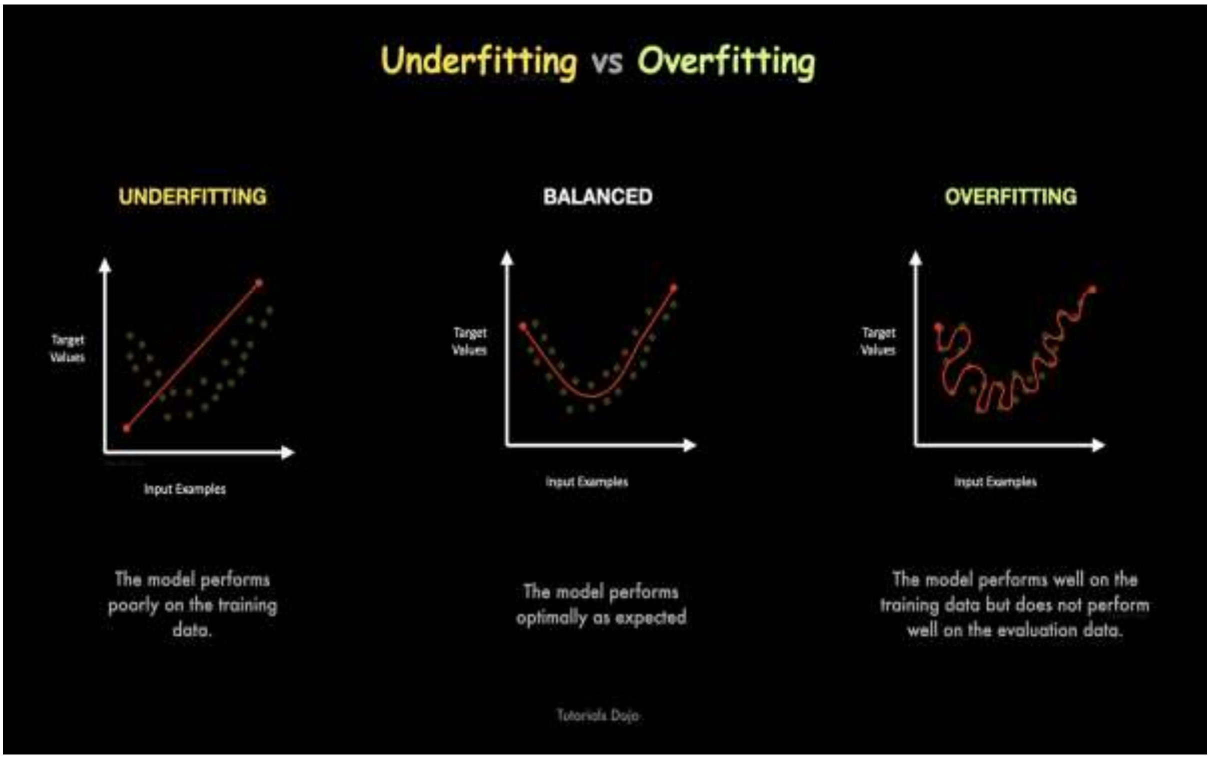


## The holdout method

- **The holdout method has two basic drawbacks**
  - In problems where we have a small dataset we may not be able to afford the "luxury" of setting aside a portion of the dataset for testing
  - Since it is a single train-and-test experiment, the holdout estimate of performance (for example error rate) will be misleading if we happen to get an "unfortunate" split between train and test
- **The limitations of the holdout can be overcome with a family of resampling methods at the expense of more computations**
  - Cross Validation
    - Random Subsampling
    - K-Fold Cross-Validation
    - Leave-one-out Cross-Validation
    - Bootstrap

Q.4: **Explain Overfitting and underfitting with suitable example [07]**

**Ans**: Overfitting and underfitting are two common problems in machine learning models that occur when the model fails to generalize well to unseen data.



The model performs poorly on the training data.

The model performs optimally as expected

The model performs well on the training data but does not perform well on the evaluation data.

⇨ **Underfitting:**
A statistical model or a machine learning algorithm is said to have underfitting when it cannot capture the underlying trend of the data, i.e., it only performs well on training data but performs poorly on testing data**.**

**Reasons for Overfitting are as follows:**

- High variance and low bias
- The model is too complex
- The size of the training data

**Techniques to reduce underfitting:**

- Increase model complexity
- Increase the number of features, performing feature engineering
- Remove noise from the data.

Increase the number of epochs or increase the duration of training to get better results

**Example:**

⇨ **Suppose we have a dataset of students with their corresponding ages and heights. We want to train a model to predict the height**

**of a student based on their age. We decide to use a polynomial regression model with increasing degrees.**

⇨ **Overfitting:**

A statistical model is said to be overfitted when the model does not make accurate predictions on testing data. When a model gets trained with so much data, it starts learning from the noise and inaccurate data entries in our data set.

**Reasons for Overfitting are as follows:**

- High variance and low bias
- The model is too complex
- The size of the training data

**Techniques to reduce overfitting:**

- Increase training data.
- Reduce model complexity.
- Early stopping during the training phase (have an eye over the loss over the training period as soon as loss begins to increase stop training).
- Ridge Regularization and Lasso Regularization
- Use dropout for neural networks to tackle overfitting.

**Example:**

⇨ Consider a classification problem where we have a dataset of emails labeled as "spam" or "not spam" based on their content features such as word frequencies and presence of certain keywords. We want to train a model to classify emails as spam or not spam.

**Q.5: Describe model parameter tuning in detail.[07]**

**Ans:** A Machine Learning model is defined as a mathematical model with a number of parameters that need to be learned from the data. By training a model with existing data, we are able to fit the model parameters**.**

The process of parameter tuning involves exploring different combinations of hyperparameter values and evaluating their impact on the model's performance. The goal is to find the set of hyperparameters that maximize the model's performance metrics, such as accuracy, precision, recall, or F1 score, depending on the specific task.

**Here is a step-by-step process for parameter tuning:**

**Define the range of hyperparameters:**

- ⇨ Start by identifying the hyperparameters that are relevant to your algorithm and problem. Determine a range of possible values for each hyperparameter. The range can be based on prior knowledge, intuition, or experimentation**.**

**Choose a search method:**

- ⇨ There are several methods available for searching the hyperparameter space, including grid search, random search, Bayesian optimization, and genetic algorithms.

**Split data into training and validation sets:**

- ⇨ Typically, the dataset is split into three parts: training set, validation set, and test set. The training set is used to train the model, the validation set is used to assess performance during the tuning process, and the test set is used for final evaluation after tuning is complete.

**Define a performance metric:**

- ⇨ Select an appropriate evaluation metric that reflects the goal of your machine learning task. For classification problems, common metrics include accuracy, precision, recall, F1 score, and area under the receiver operating characteristic curve (AUC-ROC). For regression problems, metrics like mean squared error (MSE) or mean absolute error (MAE) are commonly used.

**Evaluate and compare results:**

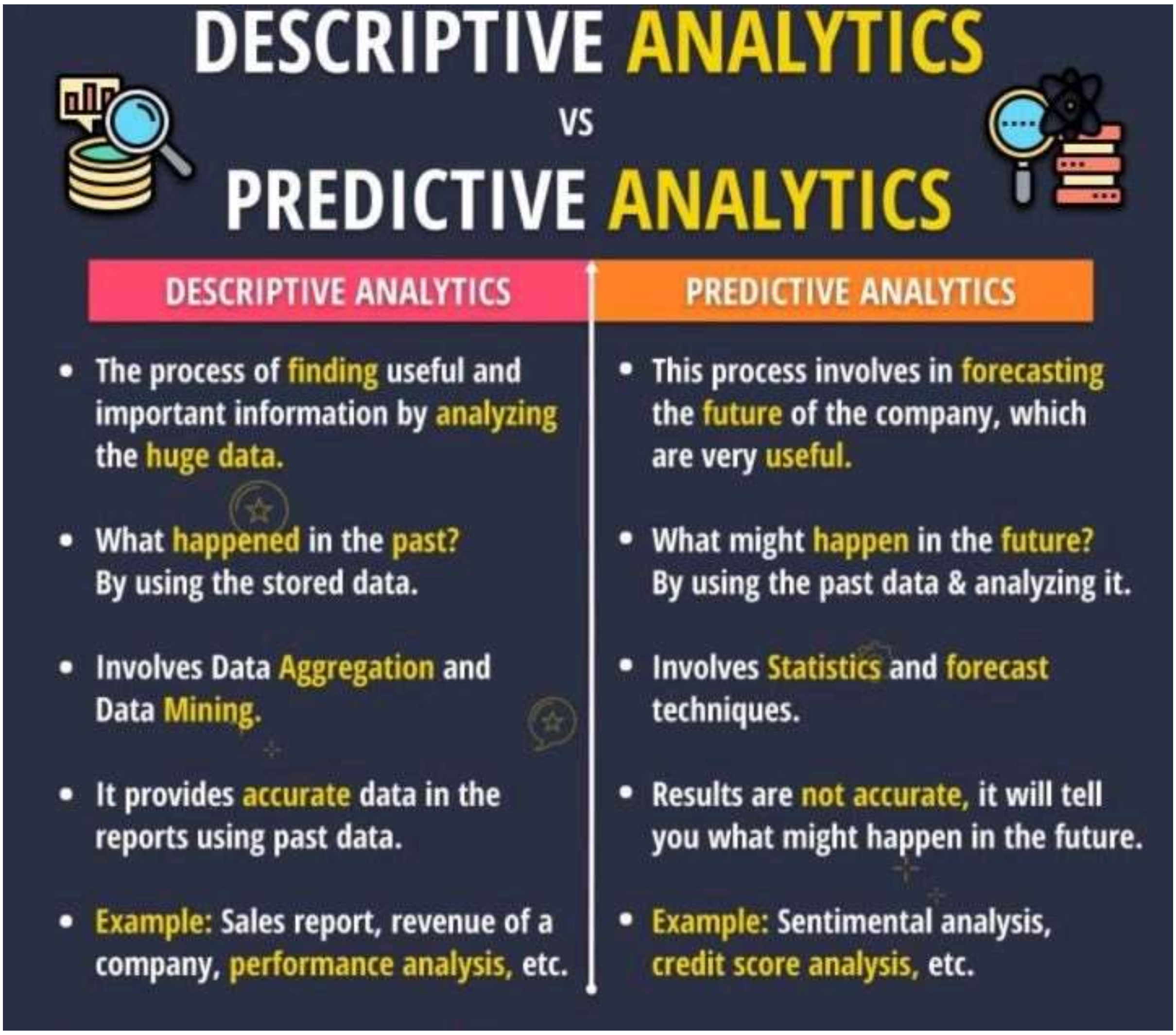
- ⇨ Analyse the performance of the model for each hyperparameter combination. Identify the combinations that yield the best performance according to the chosen evaluation metric. Visualize the results to gain insights into the relationships between hyperparameters and performance.

**Evaluate the final model**:

- ⇨ Once the tuning process is complete, evaluate the final model on the test set, which provides an unbiased estimate of the model's performance on unseen data. This step ensures that the model's performance is not overly optimized for the validation set.

**Q.6: Give the difference between predictive model and descriptive model.[03]**

**Ans:**



DESCRIPTIVE ANALYTICS VS PREDICTIVE ANALYTICS

| DESCRIPTIVE ANALYTICS | PREDICTIVE ANALYTICS |
|---|---|
| The process of finding useful and important information by analyzing the huge data. | This process involves in forecasting the future of the company, which are very useful. |
| What happened in the past? By using the stored data. | What might happen in the future? By using the past data & analyzing it. |
| Involves Data Aggregation and Data Mining. | Involves Statistics and forecast techniques. |
| It provides accurate data in the reports using past data. | Results are not accurate, it will tell you what might happen in the future. |
| Example: Sales report, revenue of a company, performance analysis, etc. | Example: Sentimental analysis, credit score analysis, etc. |

**Q.7: Consider the following confusion matrix of the win/loss prediction of cricket match. Calculate the accuracy, error rate, sensitivity, specificity, precision, recall and F-measure of the model.[07]**

| | Actual win | Actual loss | |
|---|---|---|---|
| Predicted win | (TP) 70 | (FP) 10 | Total =86 |
| Predicted loss | (FN) 15 | (TN) 5 | Total=14 |
| | Total: 85 | Total: 15 | |

**Accuracy:**

- ⇨ Accuracy is the proportion of correct predictions out of the total predictions.

Total predictions = Sum of all values in the confusion matrix = 70 + 10+ 15 + 5 = 100 Correct predictions = Number of true positives + Number of true negatives = 70+5=75

Accuracy = (TP+TN)/(TP+TN+FP+FN)
= (70+5)/ (70+5+10+15)
= 75/100
=0.75(75% Accuracy of the model)

**Error rate:**

- ⇨ Error Rate is the proportion of incorrect predictions out of the total predictions**.**

Error Rate = 1 – Accuracy = 1 - 0.75 = 0.25 or 25%

**sensitivity (True Positive Rate or Recall):**

- ⇨ sensitivity measures the proportion of actual positive cases that are correctly identified as positive**.**

**Sensitivity=TP/(TP+FN)**

**=70/ (70+15)**
**=70/85**
**=0.823**

**Specificity:**

- ⇨ Specificity measures the proportion of actual negative cases that are correctly identified as negative.

**Specifity = TN/TN+FP**
**=5/5+10 =5/15 =0.33**

**Precision:**

- ⇨ precision measures the proportion of predicted positive cases that are actually positive.

**Precision=TP/TP+FN**

**=70/70+15**

**=70/85**

**=0.823**

**Recall:**

- Recall measures the proportion of actual positive cases that are correctly identified as positive.

**Recall=TP/TP+FN**

**=70/70+15**

**=70/85**

**=0.823**

**F-measure:**

- The F-measure combines precision and recall into a single metric. It is the harmonic mean of precision and recall.

**F-measure**=2(pre*recall)/(pre+ recall)

**=2(0.82*0.82)/(0.94+0.48)**

**=2(0.4512/1.42)**

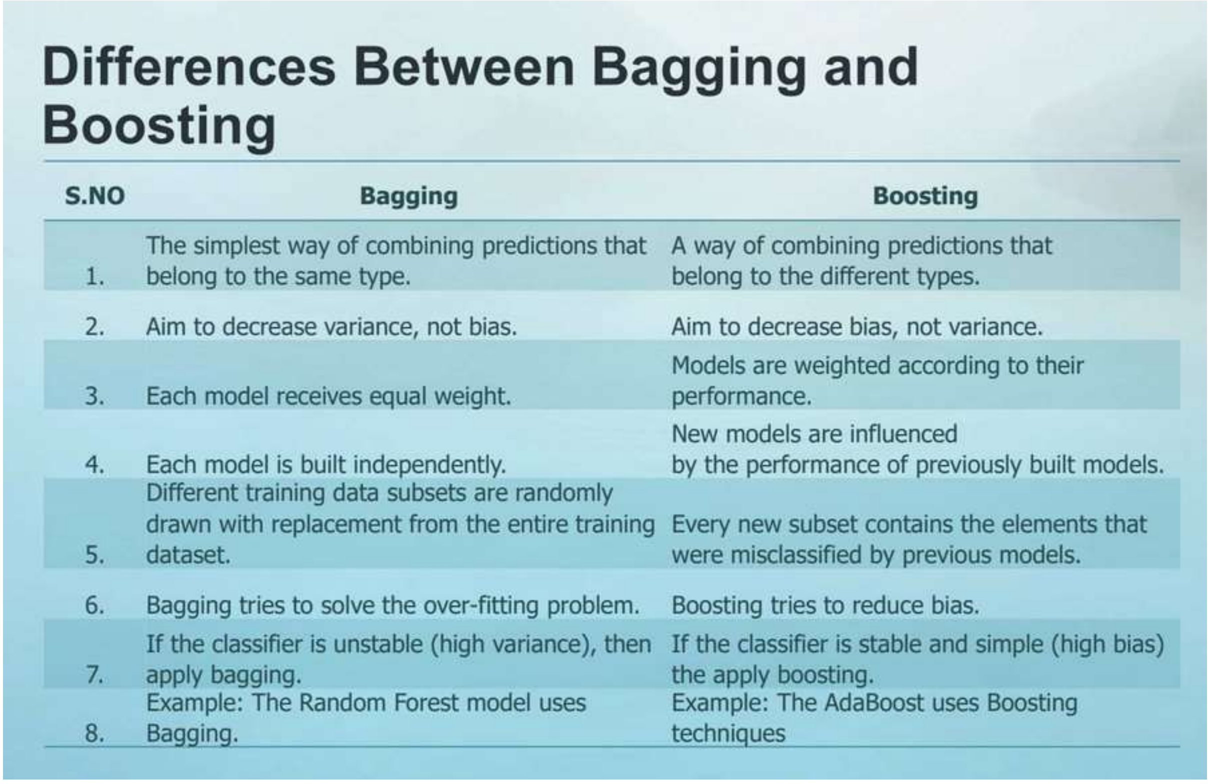
**=0.63**

**To summarize:**

**Accuracy: 0.75**
**Error Rate: 0.25**
**Sensitivity/Recall: 0.823**
**Specificity: 0.33**
**Precision: 0.823**
**F-measure: 0.63**

**Q.8: Give the difference between Bagging and Boosting.[03]**

**Ans:**



## Differences Between Bagging and Boosting

| S.NO | Bagging | Boosting |
|---|---|---|
| 1. | The simplest way of combining predictions that belong to the same type. | A way of combining predictions that belong to the different types. |
| 2. | Aim to decrease variance, not bias. | Aim to decrease bias, not variance. |
| 3. | Each model receives equal weight. | Models are weighted according to their performance. |
| 4. | Each model is built independently. | New models are influenced by the performance of previously built models. |
| 5. | Different training data subsets are randomly drawn with replacement from the entire training dataset. | Every new subset contains the elements that were misclassified by previous models. |
| 6. | Bagging tries to solve the over-fitting problem. | Boosting tries to reduce bias. |
| 7. | If the classifier is unstable (high variance), then apply bagging. | If the classifier is stable and simple (high bias) the apply boosting. |
| 8. | Example: The Random Forest model uses Bagging. | Example: The AdaBoost uses Boosting techniques |

**Q.9: while predicting malignancy of tumour of a set patients using a classification model, following are the data recorded:[07]**

**1.Correct predictions-15 malignant,75 benign**

**2.Incorrect predictions-4 malignant,6 benign**

**Create confusion matrix. Calculate the sensitively, specificity, precision, Recall and F-measure of the model.**

**Ans:**

| | Actual win | Actual loss | |
|---|---|---|---|
| Predicted win | (TP) 15 | (FP) 75 | Total =90 |
| Predicted loss | (FN) 4 | (TN) 6 | Total=10 |
| | Total: 19 | Total: 81 | |

**sensitivity (True Positive Rate or Recall):**

- ⇨ sensitivity measures the proportion of actual positive cases that are correctly identified as positive**.**

**Sensitivity=TP/(TP+FN)**

**=15/ (15+4)**
**=15/19**
**=0.78**

**Specificity:**

⇨ Specificity measures the proportion of actual negative cases that are correctly identified as negative.

**Specifity = TN/TN+FP**
**=6/6+75**
**=6/81=0.07**

**Precision:**

⇨ precision measures the proportion of predicted positive cases that are actually positive.

**Precision=TP/TP+FN**

**=15/15+75**

**=15/90**

**=0.14**

**Recall:**

⇨ Recall measures the proportion of actual positive cases that are correctly identified as positive.

**Recall=TP/TP+FN**
**=15/15+4**
**=15/19**
**=0.78**

**F-measure:**

⇨ The F-measure combines precision and recall into a single metric. It is the harmonic mean of precision and recall.

**F-measure**=2(pre*recall)/(pre+ recall)
**=2(0.16*0.78)/(0.16*0.78)**
**=0.26**

**To summarize:**
**Sensitivity/Recall: 0.78**
**Specificity: 0.07**
**Precision: 0.16**

**F-measure: 0.26**

## Q.10: Explain model underfitting in brief.[03]

Ans: Underfitting occurs when our machine learning model is not able to capture the underlying trend of the data. To avoid the overfitting in the model, the fed of training data can be stopped at an early stage, due to which the model may not learn enough from the training data. As a result, it may fail to find the best fit of the dominant trend in the data.

In the case of underfitting, the model is not able to learn enough from the training data, and hence it reduces the accuracy and produces unreliable predictions.

An underfitted model has high bias and low variance.

How to avoid underfitting:

- By increasing the training time of the model.
- By increasing the number of features.

High bias and low variance are good indicators of underfitting. Since this behavior can be seen while using the training dataset, underfitted models are usually easier to identify than overfitted ones.

**Reasons for Underfitting:**

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

## Q.11: Can the performance of a learning model be improved? If yes, explain how.[07]

**Ans:** Yes, the performance of a learning model can be improved through various techniques. Here are some common approaches to enhancing model performance:

**Increase the amount of training data:** Providing more diverse and representative data can help the model learn better and generalize well to unseen examples. More data allows the model to capture complex patterns and relationships, resulting in improved performance.

**Data preprocessing and feature engineering**: Properly preparing the data before training can significantly impact model performance. This includes techniques such as data cleaning, normalization, feature scaling, and handling missing values. Feature engineering involves creating new features or transforming existing ones to better represent the underlying patterns in the data.

**Select relevant features:** Sometimes, not all features are equally informative for the learning task. Feature selection techniques can help identify the most relevant features, reducing noise and improving model efficiency and generalization.

**Use appropriate model architecture**: Choosing the right model architecture for a given task is crucial. Complex problems may require more sophisticated models with deeper architectures, while simpler problems can be effectively solved with simpler models. Selecting the appropriate architecture, such as convolutional neural networks (CNNs) for image data or recurrent neural networks (RNNs) for sequential data, can significantly enhance performance.

**Hyperparameter tuning:** Models often have hyperparameters that need to be set before training, such as learning rate, regularization strength, or network size. Tuning these hyperparameters using techniques like grid search, random search, or Bayesian optimization can help find the optimal combination for improved performance.

**Regularization techniques:** Regularization methods, such as L1 and L2 regularization, dropout, or batch normalization, can prevent overfitting and improve the model's ability to generalize to unseen data.

**Model evaluation and iteration:** Evaluating the model's performance using appropriate metrics and validation techniques helps identify its weaknesses. Based on the evaluation results, adjustments can be made to the model or training process, such as increasing model capacity, adjusting hyperparameters, or acquiring additional data, leading to iterative improvements.

**Deploying on more powerful hardware**: Utilizing more powerful hardware, such as GPUs or TPUs, can significantly speed up the training process, enabling the exploration of larger models or more extensive hyperparameter search spaces.

Therefore, it's often necessary to experiment with different approaches and iterate to achieve the best possible performance.

**Q.12: Out of 200 emails, a classification model correctly predicted 150 emails and 30 ham emails. What is the error rate of the model?[03]**

**Ans:** To calculate the error rate of the classification model, we need to determine the number of incorrect predictions made by the model.

Given:

Total emails (n) = 200

Correctly predicted emails (c) = 150

Ham emails (actual negative) (a) = 30

The model's incorrect predictions can be calculated as the difference between the total emails and the correctly predicted emails:

Incorrect predictions = Total emails - Correctly predicted emails

= n - c

= 200 - 150

= 50

Now, the error rate of the model can be calculated as the ratio of incorrect predictions to the total number of emails:

Error rate = Incorrect predictions / Total emails

= 50 / 200

= 0.25

Therefore, the error rate of the model is 0.25, which can also be expressed as 25% or 25 out of 100.

**Q.13: Out of 200 emails, a classification model correctly predicted 150 emails and 30 ham emails. What is the accuracy of the model?[03]**

**Ans:** To calculate the accuracy of the classification model, we need to determine the number of correct predictions made by the model.

Given:

Total emails (n) = 200

Correctly predicted emails (c) = 150

The accuracy of the model can be calculated as the ratio of correct predictions to the total number of emails:

Accuracy = Correct predictions / Total emails

= c / n

= 150 / 200

= 0.75

Therefore, the accuracy of the model is 0.75, which can also be expressed as 75% or 75 out of 100.

**Q.14: Explain the main purpose of a descriptive model**.[03]

**Ans:** Descriptive models are used to gain insights, understand relationships, and provide a clear and concise representation of the data.

**Here are some of the main purposes of descriptive models:**

**Data Summarization:** Descriptive models are used to condense large or complex datasets into simpler forms, allowing for easier interpretation and understanding. These models provide summary statistics, such as mean, median, mode, variance, or frequency distributions, that provide an overview of the data's central tendencies, dispersion, and distribution.

**Pattern Recognition:** Descriptive models help in identifying and highlighting patterns, trends, or regularities within the data. They can reveal relationships between variables, detect anomalies or outliers, and uncover hidden structures or clusters.

**Visualization:** Descriptive models often include visual representations, such as charts, graphs, or plots, to present the data in a more intuitive and accessible manner. Visualizations aid in understanding the data, identifying patterns, and conveying information effectively.

**Data Exploration**: Descriptive models are valuable for exploring and getting familiar with a dataset. They allow analysts to examine the data from different angles, assess its quality, identify missing values or inconsistencies, and generate initial hypotheses or research questions.

**Communication:** Descriptive models provide a means to communicate findings and insights to a wider audience. By presenting a clear and concise summary of the data, they facilitate effective communication and enhance decision-making processes.

Overall, the main purpose of a descriptive model is to provide a comprehensive and informative overview of the data, enabling better understanding, exploration, and communication of the underlying **information.**

**Q.15: Define predictive models. Give one example.[03]**

**Ans:** Predictive modeling uses known results to create, process, and validate a model that can be used to forecast future outcomes. It is a tool used predictive analysis, a data mining technique that attempts to answer the question, "what might happen in the future?"

**Example:** Weather Forecast

Predictive modeling methods like a decision tree and linear regression forecast weather changes and natural calamities—thunderstorms, cyclones, and tsunamis. These models can ascertain the wind direction and wind speed of storms. Thus, these models are used to alert inhabitants of an area.

*Marketing and Retail Sector*

*When a business runs a marketing campaign, it uses predictive modeling techniques to anticipate campaign success. Predictive analysis also gauges target audience and future sales. In the retail sector, predictive analyses provide, forecasts based on which businesses decide the required inventory for each certain product. Projections help decide how much stock volume is required to meet future demands—pertaining to a particular product.*

**Q.16: Explain test data and training data in brief.[04]**

**Ans:** Test data and training data are two essential components in the development and evaluation of machine learning models. Here's a brief explanation of each:

**Training Data:**

Training data refers to the labeled dataset used to train a machine learning model. It consists of input data (features) and corresponding known output values (labels or targets). The training data is used to teach the model the patterns and relationships between the input and output variables. The model learns from this data by adjusting its internal parameters or weights to minimize the difference between its predicted output and the actual known output**.**

During the training phase, the model goes through multiple iterations, adjusting its parameters based on an optimization algorithm (e.g., gradient descent) to improve its performance. The goal is to enable the model to make accurate predictions on new, unseen data.

**Test Data:**

Test data, on the other hand, is a separate dataset that is used to evaluate the performance of a trained machine learning model. It is different from the training data and contains unseen examples that the model has not encountered during training. Test data helps assess how well the model generalizes to new, unknown data and measures its predictive accuracy.

The test data is typically labeled, meaning it includes the input features and the corresponding known output values. The model uses this unseen data to make predictions, and then the predicted outputs are compared to the actual known outputs. By comparing the predictions with the ground truth labels, various evaluation metrics (such as accuracy, precision, recall, etc.) are calculated to assess the model's performance.

Separating the test data from the training data is crucial to evaluate the model's ability to generalize. If the same data used for training is also used for testing, the model's performance might be overly optimistic, as it has already seen and learned from that data.

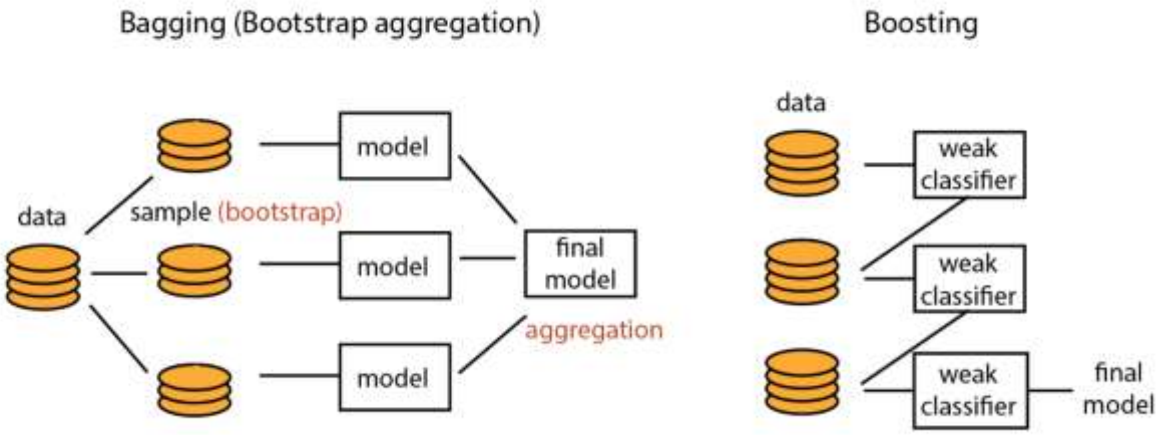
In summary, training data is used to teach the model and adjust its parameters, while test data is used to evaluate the model's performance on unseen examples. By splitting the data into training and test sets, we can assess how well the model generalizes and make reliable predictions on new, unseen data.

Dataset
Training Set
Test Set

Q.17**: Describe ensemble learning approach in detail.[07]**

**Ans:** Ensemble learning can be applied to both classification and regression problems. There are several popular techniques used in ensemble learning, including:

**Bagging (Bootstrap Aggregating):** Bagging involves creating multiple subsets of the training data by randomly sampling with replacement. Each subset is used to train a separate base model. The final prediction is obtained by averaging the predictions of all base models (for regression) or by majority voting (for classification). Examples of bagging-based ensemble methods include Random Forests.

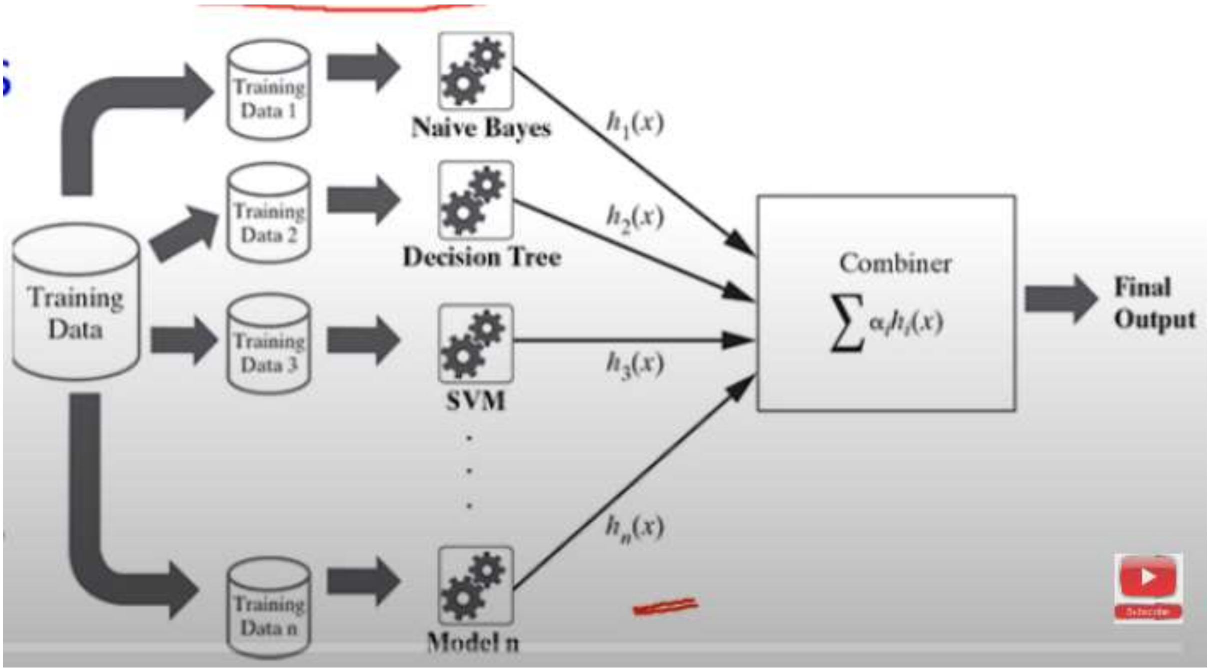


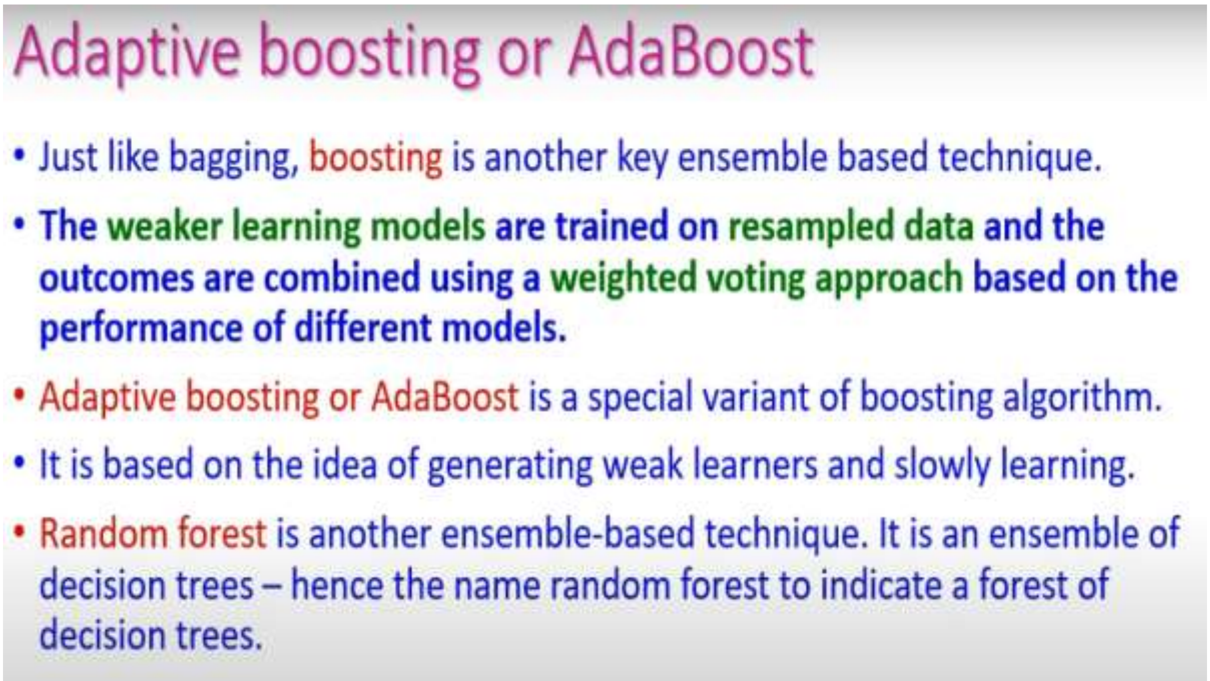
**Boosting**: Boosting is an iterative ensemble method where base models are trained sequentially, and each subsequent model focuses on correcting the mistakes of the previous models. During training, the instances that were misclassified by previous models are given higher weights, so the subsequent models can focus more on those difficult instances. The final prediction is obtained by combining the weighted predictions of all base models. Gradient Boosting and AdaBoost are popular boosting algorithms.

**Stacking**: Stacking involves training multiple base models on the same dataset and then using another model, called a meta-learner or a blender, to combine their predictions. The predictions of the base models serve as input features for the meta-learner, which learns to make the final prediction. Stacking allows for more complex combinations of base models and can potentially capture higher-level patterns and relationships in the data.

The benefits of ensemble learning include increased accuracy, improved generalization, and better resistance to overfitting. By combining diverse models that have different strengths and weaknesses, ensemble methods can reduce biases, increase robustness, and capture a wider range of patterns in the data.

It widely used in various machine learning applications.





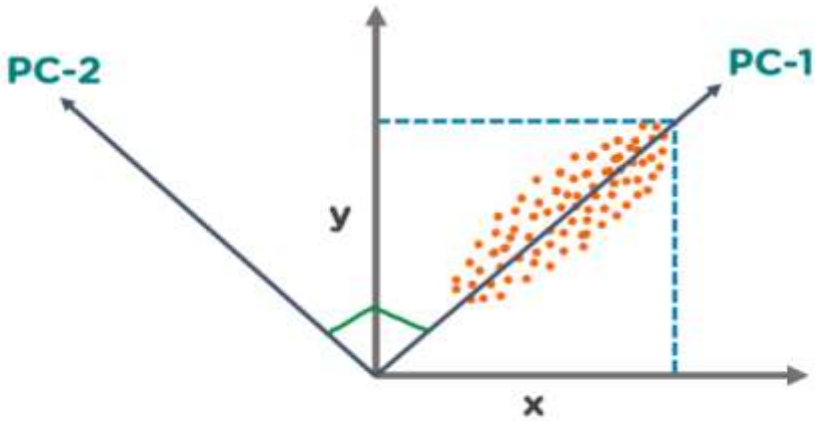
# Adaptive boosting or AdaBoost

- Just like bagging, boosting is another key ensemble based technique.
- **The weaker learning models are trained on resampled data and the outcomes are combined using a weighted voting approach based on the performance of different models.**
- Adaptive boosting or AdaBoost is a special variant of boosting algorithm.
- It is based on the idea of generating weak learners and slowly learning.
- Random forest is another ensemble-based technique. It is an ensemble of decision trees – hence the name random forest to indicate a forest of decision trees.

# CHAPTER 4: SUPERVISED LEARNING CLASSIFICATION

Q1] WRITE A SHORT NOTE ON PCA?

Ans. The Principal Component Analysis is a popular unsupervised learning technique for reducing the dimensionality of data. It increases interpretability yet, at the same time, it minimizes information loss. It helps to find the most significant features in a dataset and makes the data easy for plotting in 2D and 3D. PCA helps in finding a sequence of linear combinations of variables.



In the above figure, we have several points plotted on a 2-D plane. There are two principal components. PC1 is the primary principal component that explains the maximum variance in the data. PC2 is another principal component that is orthogonal to PC1.

Uses of PCA

PCA is a widely used technique in data analysis and has a variety of applications, including:

1. Data compression: PCA can be used to reduce the dimensionality of high-dimensional datasets, making them easier to store and analyze.
2. Feature extraction: PCA can be used to identify the most important features in a dataset, which can be used to build predictive models.
3. Visualization: PCA can be used to visualize high-dimensional data in two or three dimensions, making it easier to understand and interpret.

## Advantages of PCA

In terms of data analysis, PCA has a number of benefits, including:

1. Dimensionality reduction: By determining the most crucial features or components, PCA reduces the dimensionality of the data, which is one of its primary benefits. This can be helpful when the initial data contains a lot of variables and is therefore challenging to visualize or analyze.
2. Feature Extraction: PCA can also be used to derive new features or elements from the original data that might be more insightful or understandable than the original features. This is particularly helpful when the initial features are correlated or noisy.

### Disadvantages of PCA

1. Interpretability: Although principal component analysis (PCA) is effective at reducing the dimensionality of data and spotting patterns, the resulting principal components are not always simple to understand or describe in terms of the original features.
2. Information loss: PCA involves choosing a subset of the most crucial features or components in order to reduce the dimensionality of the data. While this can be helpful for streamlining the data and lowering noise, if crucial features are not included in the components chosen, information loss may also result.

**Q2] EXPLAIN IN DETAIL, THE PROCESS OF EVALUATING THE PERFORMANCE OF A CLASSIFICATION MODEL?**

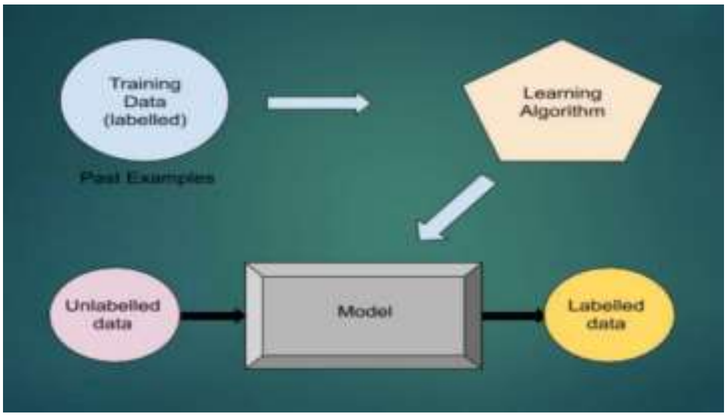
Ans. Classification model evaluation. Before starting out directly with classification let’s talk about ML tasks in general. Machine Learning tasks are mainly divided into three types

1. Supervised Learning — In Supervised learning, the model is first trained using a Training set

2. (it contains input-expected output pairs). This trained model can be later used to predict output for any unknown input.
3. Unsupervised Learning — In unsupervised learning, the model by itself tries to identify patterns in the training set.
4. Reinforcement Learning — This is an altogether different type. Better not to talk about it.

## Q3] DRAW AND DESCRIBE CLASSIFICATION MODEL IN DETAIL?

Ans.



The Classification algorithm is a Supervised Learning technique that is used to identify the category of new observations on the basis of training data. In Classification, a program learns from the given dataset or observations and then classifies new observation into a number of classes or groups. Such as, Yes or No, 0 or 1, Spam or Not Spam, cat or dog**,** etc. Classes can be called as targets/labels or categories.

Unlike regression, the output variable of Classification is a category, not a value, such as "Green or Blue", "fruit or animal", etc. Since the Classification algorithm is a Supervised learning technique, hence it takes labelled input data, which means it contains input with the corresponding output.

In classification algorithm, a discrete output function(y) is mapped to input variable(x).

1. y=f(x), where y = categorical output

The best example of an ML classification algorithm is **Email Spam Detector**.

## Q4] GIVE THE WEAKNESS OF LOGISTIC REGRESSION?

Ans. Logistic Regression is a classification algorithm used to find the probability of event success and event failure. It is used when the dependent variable is binary (0/1, True/False, Yes/No) in nature. It supports categorizing data into studying the relationship from a given set of labelled data.

The weakness is:

- Logistic regression fails to predict a continuous outcome.

- Logistic regression assumes linearity between the predicted (dependent) variable and the predictor (independent) variables.

- Logistic regression may not be accurate if the sample size is too small.

Q5] EXPLAIN THE APPLICATION OF SIMPLE LINEAR REGRESSION?

Ans. In simple linear regression, we aim to reveal the relationship between a single independent variable or you can say input, and a corresponding dependent variable or output. We can discuss this in a simple line as ***y = β0 +β1x+ε***

Here, Y speaks to the output or dependent variable, β0 and β1 are two obscure constants that speak to the intercept and coefficient that is slope separately, and the error term is ε Epsilon.

We can also discuss this in the form of a graph and here is a sample simple linear regression model graph. Thus, in this whole blog, you will get to learn so many new things about simple linear regression in detail.

- **Applications of Simple Linear Regression:**

**1.Marks scored by students based on number of hours studied (ideally)-** Here marks scored in exams are independent and the number of hours studied is independent.

**2.Predicting crop yields based on the amount of rainfall-** Yield is a dependent variable while the measure of precipitation is an independent variable.

**3.Predicting the Salary of a person based on years of experience-** Experience becomes independent while Salary turns into the dependent variable

## Q6] GIVE THE DIFFERENCE BETWEEN CLASSIFICATION AND REGRESSION?

Ans.

| 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. |

## Q7] DEFINE FOLLOWING TERMS: SUPPORT VECTORS, HYPERPLANE, MARGIN?

Ans. Support vector: The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N — the number of features) that distinctly classifies the data points.

Hyperplane: in Machine Learning, a hyperplane is a decision boundary that divides the input space into two or more regions, each corresponding to a different class or output label. In a 2D space, a hyperplane is a straight line that divides the space into two halves. In a 3D space, however, a hyperplane is a plane that divides the space into two halves. Meanwhile in higher-dimensional spaces, a hyperplane is a subspace of one dimension less than the input space.

Margin: In machine learning the margin of a single data point is defined to be the distance from the data point to a decision boundary. Note that there are many distances and decision boundaries that may be appropriate for certain datasets and goals.

## Q8] DEFINE CLASSIFICATION. EXPLAIN CLASSIFICATION LEARNING STEPS IN DETAIL USING FLOWCHART?

Ans.Classification is a supervised machine learning process of categorizing a given set of input data into classes based on one or more variables. Additionally, a classification problem can be performed on structured and unstructured data to accurately predict whether or not the data will fall into predetermined categories.

Classification in machine learning can require two or more categories of a given data set. Therefore, it generates a probability score to assign the data into a specific category, such as spam or not spam, yes or no, disease or no disease, red or green, male, or female, etc. The Classification algorithm is a Supervised Learning technique that is used to

identify the category of new observations on the basis of training data. In

Classification, a program learns from the given dataset or observations and then

classifies new observation into a number of classes or groups. Such as, Yes or No,

0 or 1, Spam or Not Spam, cat or dog, etc. Classes can be called as targets/labels

or categories.

Types of ML Classification Algorithms:

Classification Algorithms can be further divided into the Mainly two category:

- Linear Models

- Logistic Regression

- Support Vector Machines ( in Syllabus)

- Non-linear Models

- K-Nearest Neighbours ( in Syllabus)

- Kernel SVM

- Naïve Bayes

- Decision Tree Classification

- Random Forest Classification

## Q9] COMPARE AND CONTRAST BETWEEN SINGLE LINEAR REGRESSION AND MULTIPLE LINEAR REGRESSION?

Ans. Linear Regression vs. Multiple Regression: An Overview

Regression analysis is a common statistical method used in finance and investing Linear regression is one of the most common techniques of regression analysis. Multiple regression is a broader class of regressions that encompasses linear and nonlinear regressions with multiple explanatory variables.

Regression as a tool helps pool data together to help people and companies make informed decisions. There are different variables at play in regression, including a dependent variable—the main variable that you're trying to understand—and an independent variable—factors that may have an impact on the dependent variable.

Q10] Write and discuss KNN algorithms?

Ans. K-Nearest Neighbor (KNN) Algorithm for Machine Learning

o K-Nearest Neighbor is one of the simplest Machine Learning algorithms

based on Supervised Learning technique.

o K-NN algorithm assumes the similarity between the new case/data and

available cases and put the new case into the category that is most similar to

the available categories.

o K-NN algorithm stores all the available data and classifies a new data point

based on the similarity. This means when new data appears then it can be

easily classified into a well suite category by using K- NN algorithm.

o K-NN algorithm can be used for Regression as well as for Classification but

mostly it is used for the Classification problems.

o K-NN is a non-parametric algorithm, which means it does not make any

assumption on underlying data.

o It is also called a lazy learner algorithm because it does not learn from the

training set immediately instead it stores the dataset and at the time of

classification, it performs an action on the dataset.

o KNN algorithm at the training phase just stores the dataset and when it gets

new data, then it classifies that data into a category that is much like the

new data.

o Example: Suppose, we have an image of a creature that looks like cat

and dog, but we want to know either it is a cat or dog. So, for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

### Q11] DEFINE: SUPPORT AND CONFIDENCE?

Ans. Support

In data mining, support refers to the relative frequency of an item set in a dataset. For example, if an itemset occurs in 5% of the transactions in a dataset, it has a support of 5%. Support is often used as a threshold for identifying frequent item sets in a dataset, which can be used to generate association rules. For example, if we set the support threshold to 5%, then any itemset that occurs in more than 5% of the transactions in the dataset will be considered a frequent itemset.

The support of an itemset is the number of transactions in which the itemset appears, divided by the total number of transactions. For example, suppose we have a dataset of 1000 transactions, and the itemset {milk, bread} appears in 100 of those transactions. The support of the itemset {milk, bread} would be calculated as follows:

```
Support ({milk, bread}) = Number of transactions containing
                           {milk, bread} / Total number of
transactions
                        = 100 / 1000
                        = 10%
```

Confidence

In data mining, confidence is a measure of the reliability or support for a given association rule. It is defined as the proportion of cases in which the association rule holds true, or in other words, the percentage of times that the items in the antecedent (the “if” part of the rule) appear in the same transaction as the items in the consequent (the “then” part of the rule).

Confidence is a measure of the likelihood that an itemset will appear if another itemset appears. For example, suppose we have a dataset of 1000 transactions, and the itemset {milk, bread} appears in 100 of those transactions. The itemset {milk} appears in 200 of those transactions. The confidence of the rule “If a customer buys milk, they will also buy bread” would be calculated as follows:

```
Confidence ("If a customer buys milk, they will also buy bread")
```

```
= Number of transactions containing
 {milk, bread} / Number of transactions containing {milk}
= 100 / 200
= 50%
```

## Q12] EXPLAIN CLASSIFICATION STEPS IN DETAIL?

Ans. Classification in machine learning and statistics is a supervised learning approach in which the computer program learns from the data given to it and makes new observations or classifications. In this article, we will learn about classification in machine learning in detail. Moreover, if you want to go beyond this article and gain some hands-on experience of Machine learning under expert guidance of it.

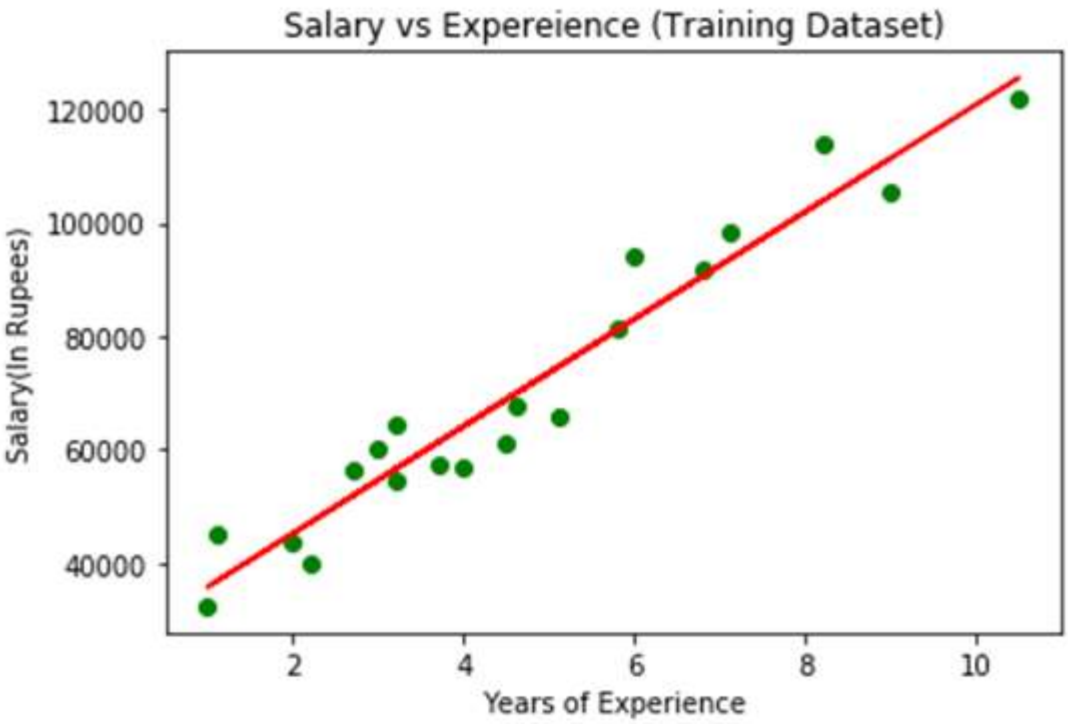
Classification Terminologies In Machine Learning

- Classifier – It is an algorithm that is used to map the input data to a specific category.
- Classification Model – The model predicts or draws a conclusion to the input data given for training, it will predict the class or category for the data.
- Feature – A feature is an individual measurable property of the phenomenon being observed.
- Binary Classification – It is a type of classification with two outcomes, for eg – either true or false.
- Multi-Class Classification – The classification with more than two classes, in multi-class classification each sample is assigned to one and only one label or target.
- Multi-label Classification – This is a type of classification where each sample is assigned to a set of labels or targets.
- Initialize – It is to assign the classifier to be used for the
- Train the Classifier – Each classifier in sci-kit learn uses the fit(X, y) method to fit the model for training the train X and train label y.
- Predict the Target – For an unlabeled observation X, the predict(X) method returns predicted label y.
- Evaluate – This basically means the evaluation of the model i.e classification report, accuracy score, etc.

## Q13] EXPLAIN A SIMPLE LINEAR REGRESSION GRAPH USING A GRAPH EXPLAINING SLOPE AND INTERCEPT?

Ans. Simple linear regression is used to estimate the relationship between two Quantitative variables. You can use simple linear regression when you want to know:

1. How strong the relationship is between two variables (e.g., the relationship between rainfall and soil erosion).
2. The value of the dependent variable at a certain value of the independent variable (e.g., the amount of soil erosion at a certain level of rainfall).



## Q14] GIVE ANY THREE APPLICATIONS OF MULTIPLE LINEAR REGRESSION?

Ans. Three applications are:

**Real estate:**

You are a real estate employee who wants to create a model to help predict the best time to sell homes. You hope to sell homes at the maximum sales price, but multiple factors can affect the sales price. These variables include the age of the house, the value of other homes in the neighbourhood, quantitative measurements of the public school system regarding student performance and the number of nearby parks, among other factors.

You can build a prediction model off these four independent variables to predict the maximum sales price of homes. You can adjust the variables if any of these factors change in terms of their coefficient values.

**Business:**

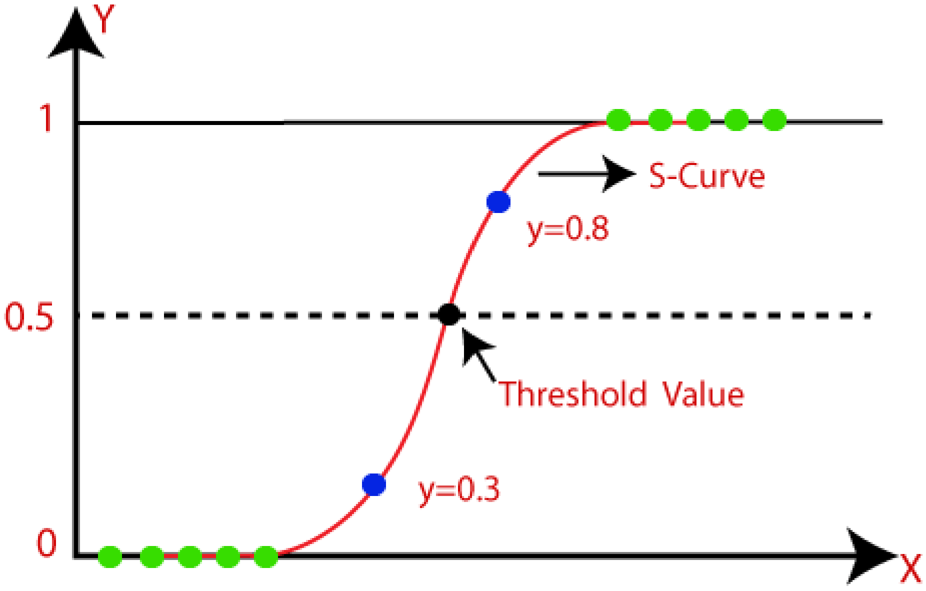
You own stock in a publicity thing and want to know if now is a good time to sell your stock. Several variables may affect the value of the stock price, including the company's profitability, the company's costs, the company's competition, and the company's assets. You can build a prediction model off these four independent variables to help decide whether to sell the stock immediately or continue holding the stock.

**Public health:**

You are an epidemiologist studying the spread of an infectious disease. You want to predict the future spread of this illness based upon current known infections. Multiple independent variables can affect the number of future infections, including the population size, population density, air temperature, asymptomatic carriers and whether the population has achieved herd immunity. You can conduct statistical modelling and multiple linear regression analysis on empirical data to predict an outcome accounting for potential changes in the coefficient values of the predictor variables.

## Q15] EXPLAIN LOGISTIC REGRESSION IN DETAIL?

- Ans. logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
- Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
- Logistic Regression is much like the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
- The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
- Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.
- Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification.



# Chapter : 5 ANSWER KEY

**1] Explain how the market basket Analysis used the concept of association analysis? [07]**

**ANS :** The Apriori algorithm uses frequent itemsets to generate association rules, and it is designed to work on the databases that contain transactions. With the help of these association rule, it determines how strongly or how weakly two objects are connected. This algorithm uses a breadth-first search and Hash Tree to calculate the itemset associations efficiently. It is the iterative process for finding the frequent itemsets from the large dataset.

This algorithm was given by the R. Agrawal and Srikant in the year 1994. It is mainly used for market basket analysis and helps to find those products that can be bought together. It can also be used in the healthcare field to find drug reactions for patients.

**Steps for Apriori Algorithm**

Below are the steps for the apriori algorithm:

**Step-1:** Determine the support of itemsets in the transactional database, and select the minimum support and confidence.

**Step-2:** Take all supports in the transaction with higher support value than the minimum or selected support value.

**Step-3:** Find all the rules of these subsets that have higher confidence value than the threshold or minimum confidence.

**Step-4:** Sort the rules as the decreasing order of lift.

**Example:** Suppose we have the following dataset that has various transactions, and from this dataset, we need to find the frequent itemsets and generate the association rules using the Apriori algorithm:

Given : Minimum support = 2,Minimum confidence = 50%

| TID | ITEMSET |
|---|---|
| T1 | A,B |
| T2 | B,D |
| T3 | B,C |
| T4 | A,B,D |
| T5 | A,C |
| T6 | B,C |
| T7 | A,C |
| T8 | A,B,C,E |
| T9 | A,B,C |

**Solution:**

**Step-1:** Calculating C1 and L1:

In the first step, we will create a table that contains support count (The frequency of each itemset individually in the dataset) of each itemset in the given dataset. This table is called the Candidate set or C1.

| ITEMSET | SUPPORT_COUNT |
|---|---|
| A | 6 |
| B | 7 |
| C | 5 |
| D | 2 |
| E | 1 |

Now, we will take out all the itemsets that have the greater support count that the Minimum Support (2). It will give us the table for the frequent itemset L1. Since all the itemsets have greater or equal support count than the minimum support, except the E, so E itemset will be removed.

| ITEMSET | SUPPORT_COUNT |
|---|---|
| A | 6 |
| B | 7 |
| C | 5 |
| D | 2 |

**Step-2:** Candidate Generation C2, and L2:

•In this step, we will generate C2 with the help of L1. In C2, we will create the pair of the itemsets of L1 in the form of subsets.

•After creating the subsets, we will again find the support count from the main transaction table of datasets, i.e., how many times these pairs have occurred together in the given dataset. So, we will get the below table for C2:

| ITEMSET | SUPPORT_COUNT |
|---|---|
| {A,B} | 4 |
| {A,C} | 4 |
| {A,D} | 1 |
| {B,C} | 4 |
| {B,D} | 2 |
| {C,D} | 0 |

Again, we need to compare the C2 Support count with the minimum support count, and after comparing, the itemset with less support count will be eliminated from the table C2. It will give us the below table for L2

| ITEMSET | SUPPORT_COUNT |
|---|---|
| {A,B} | 4 |
| {A,C} | 4 |
| {B,C} | 4 |
| {B,D} | 2 |

A B C D

**Step-3:** Candidate generation C3, and L3:

For C3, we will repeat the same two processes, but now we will form the C3 table with subsets of three itemsets together, and will calculate the support count from the dataset. It will give the below table:

| ITEMSET | SUPPORT_COUNT |
|---|---|
| {A,B,C} | 2 |
| {B,C,D} | 1 |
| {A,C,D} | 0 |
| {A,B,D} | 0 |

•Now we will create the L3 table. As we can see from the above C3 table, there is only one combination of itemset that has support count equal to the minimum support count. So, the L3 will have only one combination, i.e., {A, B, C}.

**Step-4:** Finding the association rules for the subsets:

To generate the association rules, first, we will create a new table with the possible rules from the occurred combination {A, B.C}. For all the rules, we will calculate the Confidence using formula sup( A ^B)/A. After calculating the confidence value for all rules, we will exclude the rules that have less confidence than the minimum threshold(50%).

Consider the below table:

| RULES | SUPPORT | CONFIDENCE |
|---|---|---|
| A^B->C | 2 | Sup{(A^B)^C}/Sup(A^B)=2/4=0.5=50% |
| B^C->A | 2 | Sup{(B^C)^A}/Sup(B^C)=2/4=0.5=50% |
| A^C->B | 2 | Sup{(A^C)^B}/Sup(A^C)=2/4=0.5=50% |
| C->A^B | 2 | Sup{(C^)(A^B)}/Sup(C)=2/5=0.4=40% |
| A->B^C | 2 | Sup{(A^)(B^C)}/Sup(A)=2/6=0.33=33.33% |
| B->B^C | 2 | Sup{(B^)(B^C)}/Sup(B)=2/7=0.28=28% |

As the given threshold or minimum confidence is 50%, so the first three rules A ^B → C, B^C → A, and A^C → B can be considered as the strong association rules for the given problem.

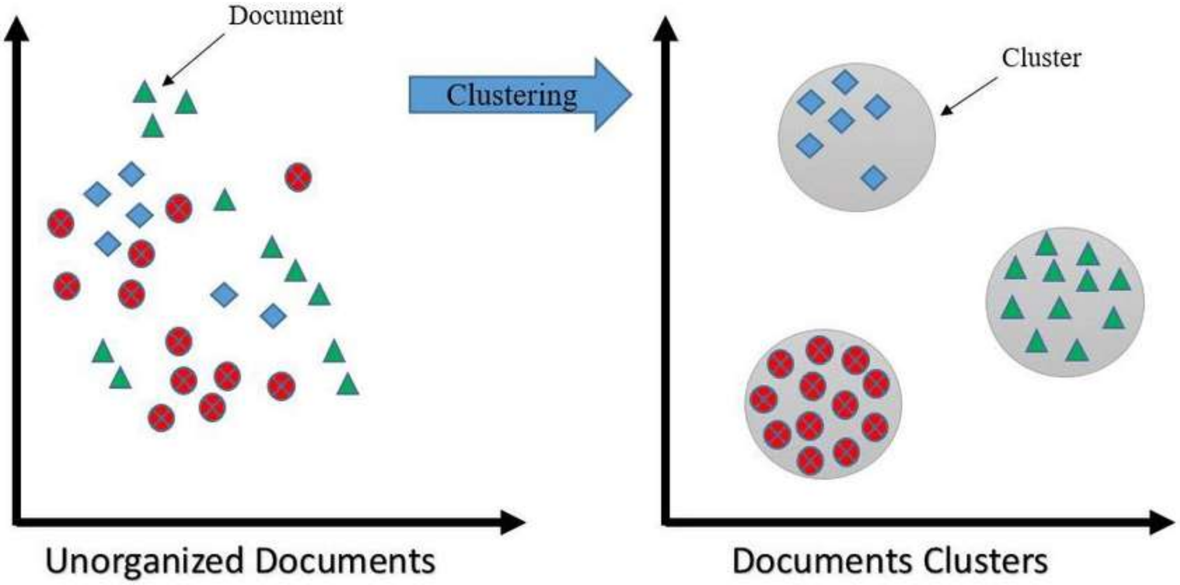
**2] Explain the k-mean clustering method with a step by step algorithm?[07]**

**ANS:** K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.

The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

The k-means clustering algorithm mainly performs two tasks:

Determines the best value for K center points or centroids by an iterative process. Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster. Hence each cluster has datapoints with some commonalities, and it is away from other clusters.



How does the K-Means Algorithm Work?

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

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

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

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

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

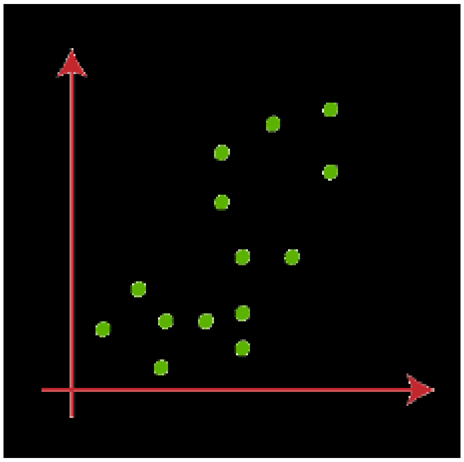
**Step-5:** Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

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

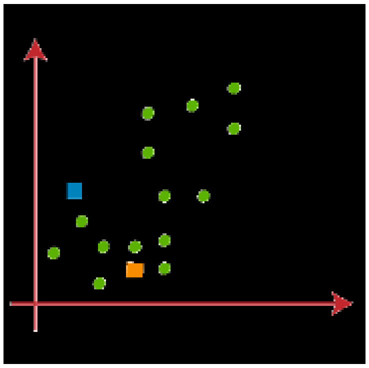
**Step-7:** The model is ready.

Suppose we have two variables M1 and M2. The x-y axis scatter plot of these two variables is given below:

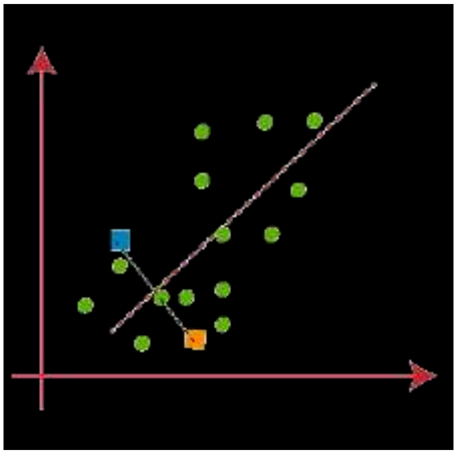
Let's take number k of clusters, i.e., K=2, to identify the dataset and to put them into different clusters. It means here we will try to group these datasets into two different clusters.



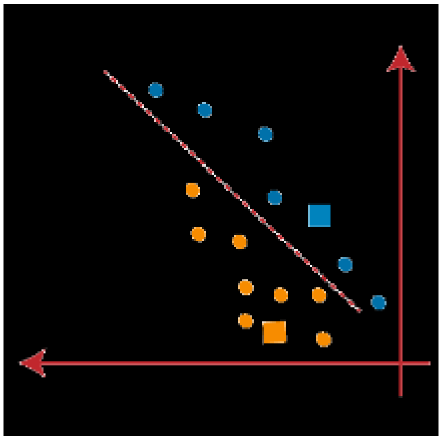
We need to choose some random k points or centroid to form the cluster. These points can be either the points from the dataset or any other point. So, here we are selecting the below two points as k points, which are not the part of our dataset. Consider the below image:



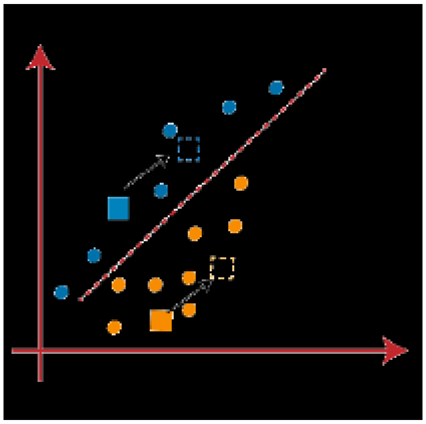
Now we will assign each data point of the scatter plot to its closest K-point or centroid. We will compute it by applying some mathematics that we have studied to calculate the distance between two points. So, we will draw a median between both the centroids. Consider the below image:



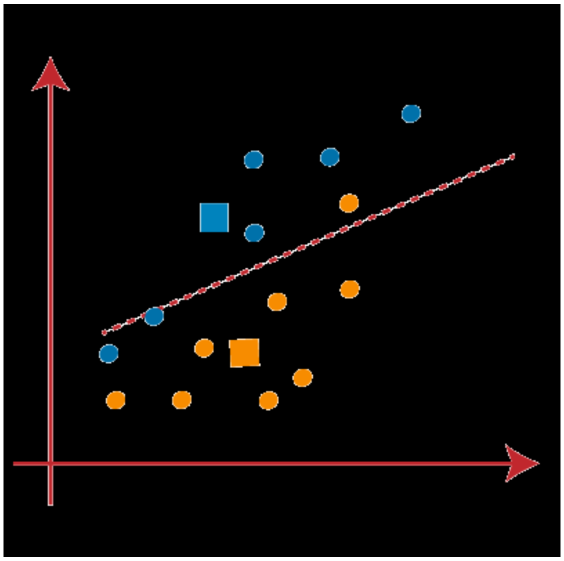
From the above image, it is clear that points left side of the line is near to the K1 or blue centroid, and points to the right of the line are close to the yellow centroid. Let's color them as blue and yellow for clear visualization.



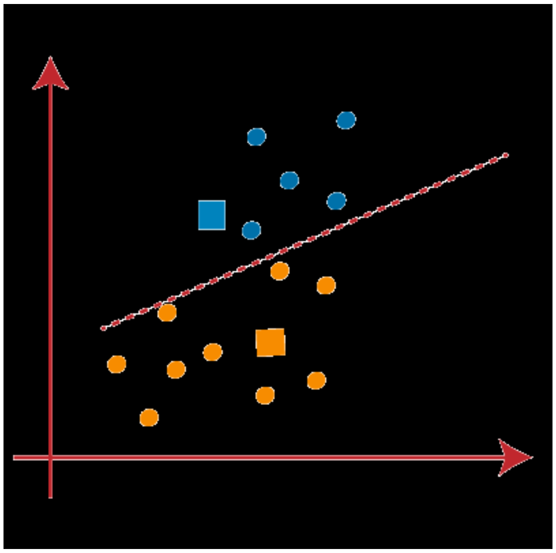
As we need to find the closest cluster, so we will repeat the process by choosing **a new centroid**. To choose the new centroids, we will compute the center of gravity of these centroids, and will find new centroids as below:



Next, we will reassign each datapoint to the new centroid. For this, we will repeat the same process of finding a median line. The median will be like below image:

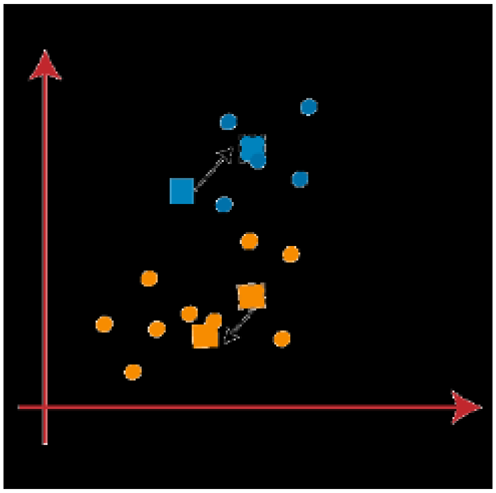


From the above image, we can see, one yellow point is on the left side of the line, and two blue points are right to the line. So, these three points will be assigned to new centroids.

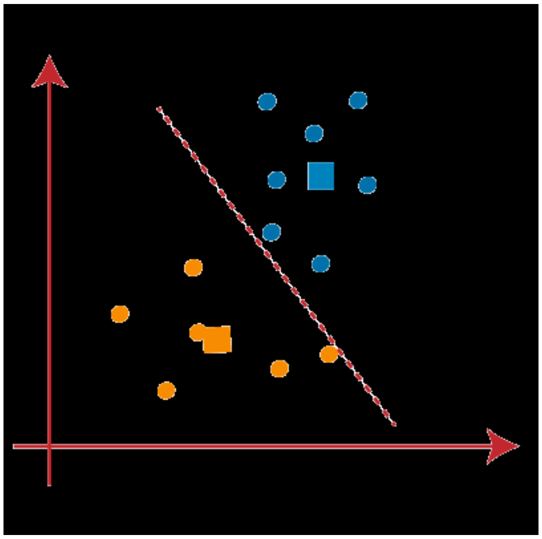


As reassignment has taken place, so we will again go to the step-4, which is finding new centroids or K-points.

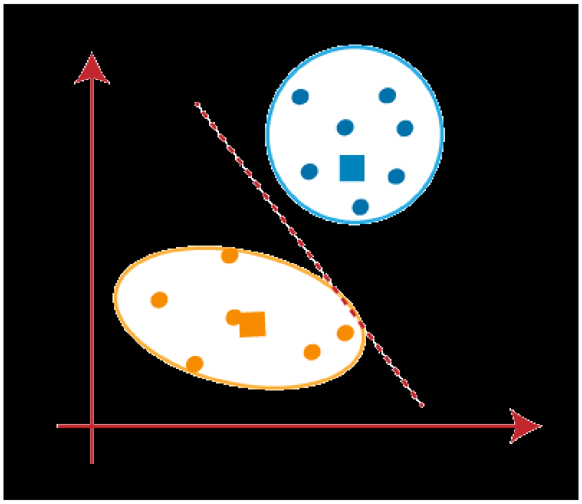
We will repeat the process by finding the center of gravity of centroids, so the new centroids will be as shown in the below image:



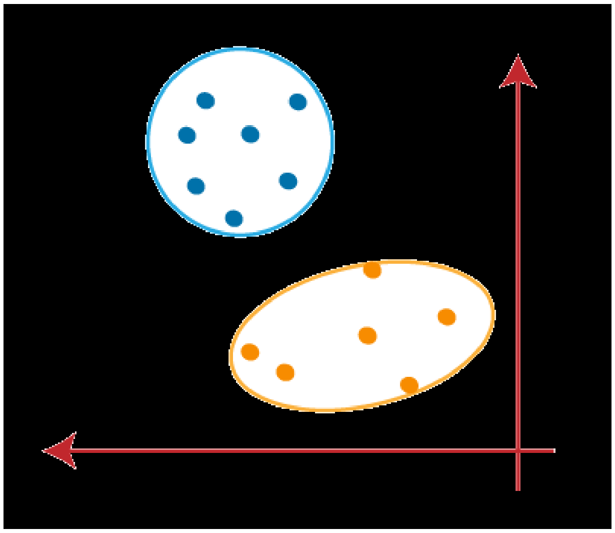
As we got the new centroids so again will draw the median line and reassign the data points. So, the image will be:



We can see in the above image; there are no dissimilar data points on either side of the line, which means our model is formed. Consider the below image:



As our model is ready, so we can now remove the assumed centroids, and the two final clusters will be as shown in the below image:



How to choose the value of "K number of clusters" in K-means Clustering?

The performance of the K-means clustering algorithm depends upon highly efficient clusters that it forms. But choosing the optimal number of clusters is a big task. There are some different ways to find the optimal number of clusters, but here we are discussing the most appropriate method to find the number of clusters or value of K. The method is given below:

**Elbow Method**

The Elbow method is one of the most popular ways to find the optimal number of clusters. This method uses the concept of WCSS value. WCSS stands for Within

Cluster Sum of Squares, which defines the total variations within a cluster. The formula to calculate the value of WCSS (for 3 clusters) is given below:

**WCSS= ∑Pi in Cluster1 distance(Pi C1)2 +∑Pi in Cluster2distance(Pi C2)2+∑Pi in CLuster3 distance(Pi C3)2**

In the above formula of WCSS,

∑Pi in Cluster1 distance(Pi C1)2: It is the sum of the square of the distances between each data point and its centroid within a cluster1 and the same for the other two terms.

To measure the distance between data points and centroid, we can use any method such as Euclidean distance or Manhattan distance.

To find the optimal value of clusters, the elbow method follows the below steps:

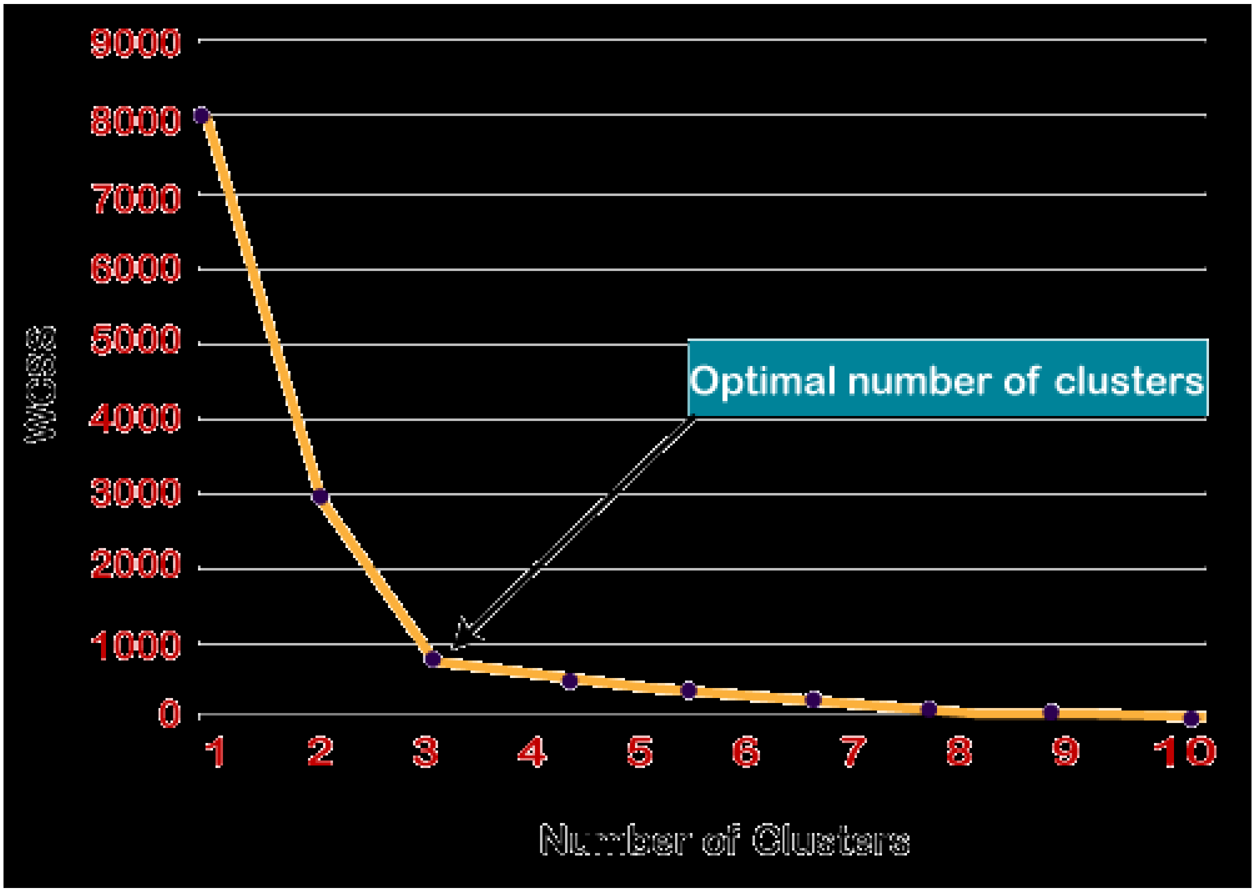
It executes the K-means clustering on a given dataset for different K values (ranges from 1-10).

For each value of K, calculates the WCSS value.

Plots a curve between calculated WCSS values and the number of clusters K.

The sharp point of bend or a point of the plot looks like an arm, then that point is considered as the best value of K.

Since the graph shows the sharp bend, which looks like an elbow, hence it is known as the elbow method. The graph for the elbow method looks like the below image:



**3] Describe the concept of clustering using appropriate real  world example?[07]**

**ANS:** Cluster analysis is a technique used in machine learning that attempts to find clusters of observations within a dataset.

The goal of cluster analysis is to find clusters such that the observations within each cluster are quite similar to each other, while observations in different clusters are quite different from each other.

**Example1:**

**RETAIL MARKETING**

Retail companies often use clustering to identify groups of households that are similar to each other.

For example, a retail company may collect the following information on households:

- Household income
- Household size
- Head of household Occupation
- Distance from nearest urban area

They can then feed these variables into a clustering algorithm to perhaps identify the following clusters:

- Cluster 1: Small family, high spenders
- Cluster 2: Larger family, high spenders
- Cluster 3: Small family, low spenders
- Cluster 4: Large family, low spenders

The company can then send personalized advertisements or sales letters to each household based on how likely they are to respond to specific types of advertisements.

**Example2:**

**<u>Email Marketing</u>**

Many businesses use cluster analysis to identify consumers who are similar to each other so they can tailor their emails sent to consumers in such a way that maximizes their revenue.

For example, a business may collect the following information about consumers:

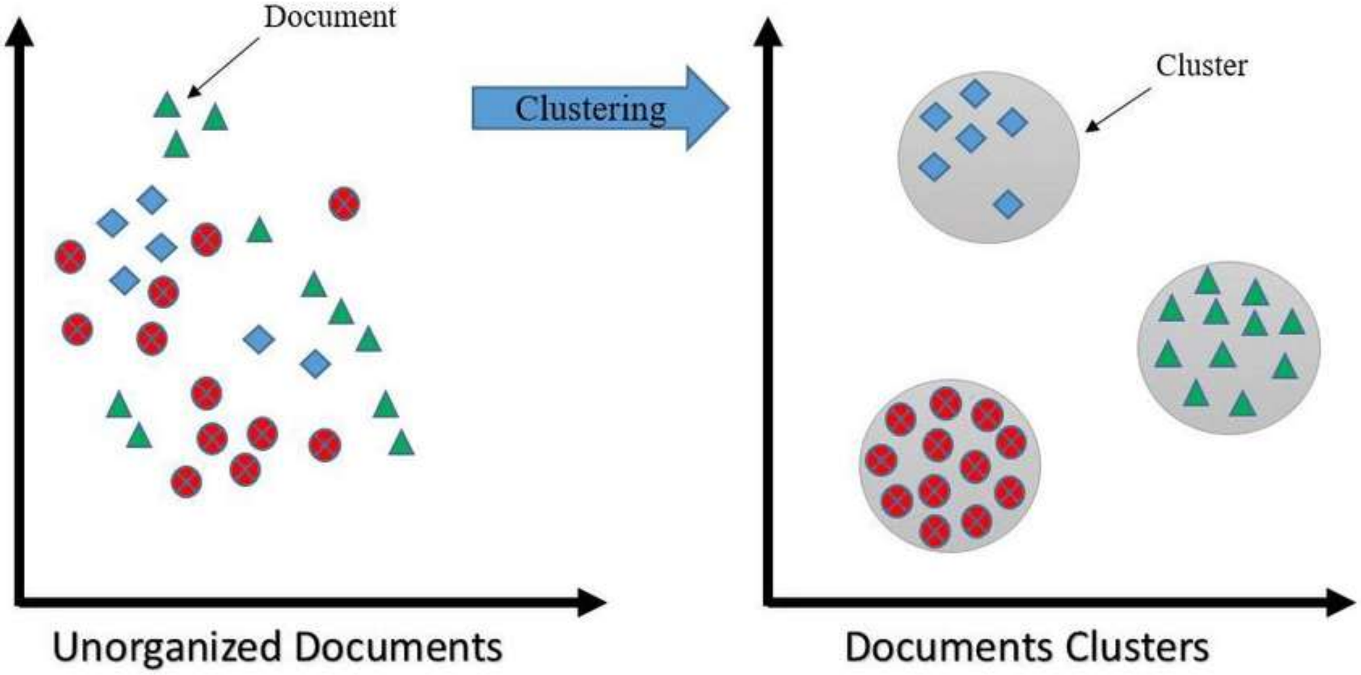
- Percentage of emails opened
- Number of clicks per email

- Time spent viewing email

Using these metrics, a business can perform cluster analysis to identify consumers who use email in similar ways and tailor the types of emails and frequency of emails they send to different clusters of customers.

**4] Draw the diagram which shows the clustering process?[04]**

**ANS:**



**5] State pros and cons of k means clustering algorithm?[05]**

**ANS : <u>Pros:</u>**

**Simple:** It is easy to implement k-means and identify unknown groups of data from complex data sets. The results are presented in an easy and simple manner.

**Flexible:** K-means algorithm can easily adjust to the changes. If there are any problems, adjusting the cluster segment will allow changes to easily occur on the algorithm.

**Suitable in a large dataset:** K-means is suitable for a large number of datasets and it's computed much faster than the smaller dataset. It can also produce higher clusters.

**Efficient:** The algorithm used is good at segmenting the large data set. Its efficiency depends on the shape of the clusters. K-means works well in hyper-spherical clusters.

**Time complexity:** K-means segmentation is linear in the number of data objects thus increasing execution time. It doesn't take more time in classifying similar characteristics in data like hierarchical algorithms.

**Easy to interpret:** The results are easy to interpret. It generates cluster descriptions in a form minimized to ease understanding of the data.

**<u>Cons:</u>**

**Order of values:** The way in which data is ordered in building the algorithm affects the final results of the data set.

**Sensitivity to scale:** Changing or rescaling the dataset either through normalization or standardization will completely change the final results.

**Handle numerical data:** K-means algorithm can be performed in numerical data only.

**Specify K-values:** For K-means clustering to be effective, you have to specify the number of clusters (K) at the beginning of the algorithm.

**Prediction issues:** It is difficult to predict the k-values or the number of clusters. It is also difficult to compare the quality of the produced clusters.

## 6] Give the difference between classification and clustering?[04]

**ANS:**

| Criteria | Classification | Clustering |
|---|---|---|
| Purpose | The main goal of classification is to assign predefined labels or categories to data instances based on their features . | Clustering aims to group similar data instances together based on their intrinsic similarities,without prior knowledge of the class labels |
| Supervision | It is a supervised learning task, which means the training data used to build the classification model includes both the input features and their corresponding output labels. | It is an unsupervised learning task, meaning there are no predefined labels or target outputs in the training data. |
| Output | The output of a classification model is a discrete class label or a probability distribution over the class labels. It assigns each data instance to a specific category based on the learned patterns. | The output of clustering is a set of clusters, where each cluster represents a group of similar data instances. |
| Training | Classification models require labeled training data, where each instance is associated with a known class label. | Clustering algorithms do not require labeled data for training. They analyze the similarity between data instances based on the given features and group them accordingly. |
| Application | Classification is commonly used for tasks such as spam filtering, sentiment analysis, image recognition, and | Clustering is employed in various fields like customer segmentation, anomaly detection, |

| | | |
|---|---|---|
| | predicting customer churn | document organization, and image segmentation. |

## 7] Write and explain Apriori algorithm with example?

**ANS:** The Apriori algorithm uses frequent itemsets to generate association rules, and it is designed to work on the databases that contain transactions. With the help of these association rule, it determines how strongly or how weakly two objects are connected. This algorithm uses a breadth-first search and Hash Tree to calculate the itemset associations efficiently. It is the iterative process for finding the frequent itemsets from the large dataset.

This algorithm was given by the R. Agrawal and Srikant in the year 1994. It is mainly used for market basket analysis and helps to find those products that can be bought together. It can also be used in the healthcare field to find drug reactions for patients.

**Steps for Apriori Algorithm**

Below are the steps for the apriori algorithm:

**Step-1:** Determine the support of itemsets in the transactional database, and select the minimum support and confidence.

**Step-2:** Take all supports in the transaction with higher support value than the minimum or selected support value.

**Step-3:** Find all the rules of these subsets that have higher confidence value than the threshold or minimum confidence.

**Step-4:** Sort the rules as the decreasing order of lift.

**Example:** Suppose we have the following dataset that has various transactions, and from this dataset, we need to find the frequent itemsets and generate the association rules using the Apriori algorithm:

Given : Minimum support = 2,Minimum confidence = 50%

| TID | ITEMSET |
|---|---|
| T1 | A,B |
| T2 | B,D |
| T3 | B,C |
| T4 | A,B,D |
| T5 | A,C |
| T6 | B,C |
| T7 | A,C |
| T8 | A,B,C,E |
| T9 | A,B,C |

**Solution:**

**Step-1:** Calculating C1 and L1:

In the first step, we will create a table that contains support count (The frequency of each itemset individually in the dataset) of each itemset in the given dataset. This table is called the Candidate set or C1.

| ITEMSET | SUPPORT_COUNT |
|---|---|
| A | 6 |
| B | 7 |
| C | 5 |
| D | 2 |
| E | 1 |

Now, we will take out all the itemsets that have the greater support count that the Minimum Support (2). It will give us the table for the frequent itemset L1. Since all the itemsets have greater or equal support count than the minimum support, except the E, so E itemset will be removed.

| ITEMSET | SUPPORT_COUNT |
|---|---|
| A | 6 |
| B | 7 |
| C | 5 |
| D | 2 |

**Step-2:** Candidate Generation C2, and L2:

•In this step, we will generate C2 with the help of L1. In C2, we will create the pair of the itemsets of L1 in the form of subsets.

•After creating the subsets, we will again find the support count from the main transaction table of datasets, i.e., how many times these pairs have occurred together in the given dataset. So, we will get the below table for C2:

| ITEMSET | SUPPORT_COUNT |
|---|---|
| {A,B} | 4 |
| {A,C} | 4 |
| {A,D} | 1 |
| {B,C} | 4 |
| {B,D} | 2 |
| {C,D} | 0 |

Again, we need to compare the C2 Support count with the minimum support count, and after comparing, the itemset with less support count will be eliminated from the table C2. It will give us the below table for L2

| ITEMSET | SUPPORT_COUNT |
|---|---|
| {A,B} | 4 |
| {A,C} | 4 |
| {B,C} | 4 |
| {B,D} | 2 |

A B C D

**Step-3:** Candidate generation C3, and L3:

For C3, we will repeat the same two processes, but now we will form the C3 table with subsets of three itemsets together, and will calculate the support count from the dataset. It will give the below table:

| ITEMSET | SUPPORT_COUNT |
|---|---|
| {A,B,C} | 2 |
| {B,C,D} | 1 |
| {A,C,D} | 0 |
| {A,B,D} | 0 |

•Now we will create the L3 table. As we can see from the above C3 table, there is only one combination of itemset that has support count equal to the minimum support count. So, the L3 will have only one combination, i.e., {A, B, C}.

**Step-4:** Finding the association rules for the subsets:

To generate the association rules, first, we will create a new table with the possible rules from the occurred combination {A, B.C}. For all the rules, we will calculate the Confidence using formula sup( A ^B)/A. After calculating the confidence value for all rules, we will exclude the rules that have less confidence than the minimum threshold(50%).

Consider the below table:

| RULES | SUPPORT | CONFIDENCE |
|---|---|---|
| A^B->C | 2 | Sup{(A^B)^C}/Sup(A^B)=2/4=0.5=50% |
| B^C->A | 2 | Sup{(B^C)^A}/Sup(B^C)=2/4=0.5=50% |
| A^C->B | 2 | Sup{(A^C)^B}/Sup(A^C)=2/4=0.5=50% |
| C->A^B | 2 | Sup{(C^)(A^B)}/Sup(C)=2/5=0.4=40% |
| A->B^C | 2 | Sup{(A^)(B^C)}/Sup(A)=2/6=0.33=33.33% |
| B->B^C | 2 | Sup{(B^)(B^C)}/Sup(B)=2/7=0.28=28% |

As the given threshold or minimum confidence is 50%, so the first three rules A ^B → C, B^C → A, and A^C → B can be considered as the strong association rules for the given problem.

**8] Write a program to cluster a set of points using k means training and test data must be provided explicitly?[07]**

**ANS:**

```
  import numpy as np
  from sklearn.cluster import KMeans

  def cluster_points(train_data, test_data, num_clusters):
   # Training phase
   kmeans = KMeans(n_clusters=num_clusters, random_state=0)
   kmeans.fit(train_data)

   # Test phase
   train_labels = kmeans.predict(train_data)
   test_labels = kmeans.predict(test_data)

  return train_labels, test_labels

# Example usage
train_data = np.array([[1, 2], [1.5, 1.8], [5, 8], [8, 8], [1, 0.6], [9, 11]])
test_data = np.array([[2, 2], [6, 9], [4, 5], [10, 10]])
```

```
num_clusters = 2

train_labels, test_labels = cluster_points(train_data, test_data, num_clusters)


print("Training labels:", train_labels)

print("Test labels:", test_labels)
```

**9] Generate large itemsets and association rules using Apriori algorithm on the following data set with minimum support value and minimum confidence value set as 50% and 75% respectively .[07]**

| **TID** | **Item Purchase** |
|---|---|
| **T1** | **Cheese , Milk , Cookies** |
| **T2** | **Butter , Milk , Bread** |
| **T3** | **Cheese , Milk , Butter , Bread** |
| **T4** | **Butter , Bread** |

**ANS:** Minimum Support=50%   Minimum confidence=75%

C1

| ITEMSET | SUPPORT_COUNT |
|---|---|
| Cheese | 2 |
| Milk | 3 |
| Cookies | 1 |
| Bread | 3 |
| Butter | 3 |

Cookies has less minimum support so it can be removed from C1

| ITEMSET | SUPPORT_COUNT |
|---|---|

| | |
|---|---|
| Cheese | 2 |
| Milk | 3 |
| Bread | 3 |
| Butter | 3 |

C2

| ITEMSET | SUPPORT_COUNT |
|---|---|
| {Cheese,Milk} | 2 |
| {Cheese,Bread} | 1 |
| {Cheese,Butter} | 1 |
| {Bread,Butter} | 3 |
| {Milk,Bread} | 2 |
| {Milk,Butter} | 2 |

{Cheese,Bread}and{Cheese,Butter} has less minimum support so it can be removed from C2

| ITEMSET | SUPPORT_COUNT |
|---|---|
| {Cheese,Milk} | 2 |
| {Bread,Butter} | 3 |
| {Milk,Bread} | 2 |
| {Milk,Butter} | 2 |

C3

| ITEMSET | SUPPORT_COUNT |
|---|---|
| {Cheese,Milk,Bread} | 1 |
| {Cheese,Milk,Butter} | 1 |
| {Bread,Butter,Milk} | 2 |

{Cheese,Milk,Bread} and {Cheese,Milk,Butter} has less minimum support so it can be removed from C3

| ITEMSET | SUPPORT_COUNT |
|---|---|
| {Bread,Milk,Butter} | 2 |

Minimum Confidence = 75%

| | | |
|---|---|---|
| Bread^Milk->Butter | 2 | Sup{(Bread^Milk)^Butter}/Sup(Bread^Milk)=2/2=1=100% |
| Milk^Butter->Bread | 2 | Sup{(Milk^Butter)^Bread}/Sup(Milk^Butter)=2/2=1=100% |
| Bread^Butter->Milk | 2 | Sup{(Bread^Butter)^Milk}/Sup(Bread^Butter)=2/3=0.6=60% |
| Butter->Bread^Milk | 2 | Sup{(Butter^)(Bread^Milk)}/Sup(Butter)=2/3=0.6=60% |
| Bread->Milk^Butter | 2 | Sup{(Bread^)(Milk^Butter)}/Sup(Bread)=2/3=0.6=60% |
| Milk->Milk^Butter | 2 | Sup{(Milk^)(Milk^Butter)}/Sup(Milk)=2/3=0.6=60% |

**10] During a research work, you found 7 observations as described with the data points below. You want to create 3 clusters from these observations using k-means algorithm . After first iteration, the clusters C1,C2,C3 has following observations:[07]**

**C1 : {(2,2),(4,4),(6,6)}**

**C2 : {(0,4),(4,0)}**

**C3 : {(5,5),(9,9)}**

**If you want to run a second iteration then what will be the cluster centroids ?**

**ANS: To determine the cluster centroids in the second iteration of the k-means algorithm, we need to calculate the mean of the data points in each cluster. Let's calculate the cluster centroids based on the given information from the first iteration:**

**Cluster C1: {(2,2), (4,4), (6,6)}**

**Cluster C2: {(0,4), (4,0)}**

**Cluster C3: {(5,5), (9,9)}**

**Cluster C1 centroid:**

**x-coordinate mean = (2 + 4 + 6) / 3 = 4**

**y-coordinate mean = (2 + 4 + 6) / 3 = 4**

**C1 centroid = (4, 4)**

**Cluster C2 centroid:**

**x-coordinate mean = (0 + 4) / 2 = 2**

**y-coordinate mean = (4 + 0) / 2 = 2**

**C2 centroid = (2, 2)**

**Cluster C3 centroid:**

**x-coordinate mean = (5 + 9) / 2 = 7**

**y-coordinate mean = (5 + 9) / 2 = 7**

**C3 centroid = (7, 7)**

**Therefore, the cluster centroids after the second iteration would be:**

**C1 centroid = (4, 4)**

**C2 centroid = (2, 2)**

**C3 centroid = (7, 7)**

# FML

## CHAPTER – 6 Python Libraries for machine learning

### Q-1 Comparing and contract NumPy with pandas.

**Ans**: NumPy and pandas are both popular libraries in Python used for data manipulation and analysis. While they have some similarities, they also have distinct features and purposes. Let's compare and contrast NumPy and pandas in different aspects:

**Purpose:**

NumPy: It provides powerful tools for efficient numerical computations and array operations. It is particularly useful for mathematical and scientific calculations.

Pandas: It is designed for data manipulation and analysis. It provides easy-to-use data structures and data analysis tools, making it convenient for working with structured and tabular data.

**Data Structures:**

NumPy: It introduces the nd array (n-dimensional array) object, which is a homogeneous collection of elements with a fixed size. It supports arrays of numerical data and provides efficient operations for array manipulation and mathematical computations.

Pandas: It introduces two primary data structures - Series and Data Frame. A Series is a one-dimensional labeled array that can hold any data type. A Data Frame is a two-dimensional table-like structure with labeled rows and columns, similar to a spreadsheet or a SQL table.

**Functionality:**

NumPy: It provides a wide range of mathematical functions and operations for arrays. It includes functions for basic array operations, linear algebra, Fourier transforms, random number generation, and more. NumPy is also the foundation for many other scientific and data-related Python libraries.

Pandas: It offers a rich set of data manipulation and analysis tools built on top of NumPy. Pandas allows for data indexing, selection, filtering, grouping, reshaping, merging, and more. It provides functions for data cleaning, handling missing values, and data visualization. Pandas also integrates well with other Python libraries for data analysis and visualization.

**Performance:**

NumPy: It is highly optimized for numerical computations and array operations. NumPy arrays are stored in a contiguous block of memory, allowing for efficient memory access and faster execution of mathematical operations. It is widely used in scientific computing and performance-critical applications.

Pandas: While Pandas leverages the NumPy array underneath, it introduces additional data structures and operations that may have a slight performance overhead compared to pure NumPy. However, Pandas provides a high-level and convenient interface for data analysis tasks, which often outweighs the minor performance differences.

**Q-2 State features of matplotlib.**

**Ans**: Here are some key features of Matplotlib:

1) Plot Types: Matplotlib supports various plot types, including line plots, scatter plots, bar plots, histograms, pie charts, box plots, 3D plots, and more. It allows you to visualize different types of data in a visually appealing manner.

2) Customization: Matplotlib offers extensive customization options to control the appearance of plots. You can customize the figure size, title, axis labels, ticks, colors, line styles, markers, transparency, fonts, and more

3) Subplots and Layouts: Matplotlib enables you to create multiple subplots within a single figure. You can arrange subplots in various configurations, such as grids or overlapping layouts.

4) Annotations and Text: You can add annotations, text, and labels to your plots using Matplotlib. Annotations can include arrows, text boxes, and other graphical elements to highlight specific data points or provide additional information

5) Support for LaTeX: Matplotlib supports LaTeX typesetting for mathematical expressions and symbols. This feature allows you to include complex mathematical equations and symbols directly in your plot titles, axis labels, annotations, and text.

6) Saving and Exporting: Matplotlib allows you to save your plots in various image formats, such as PNG, JPEG, PDF, SVG, and more. You can also save plots as vector graphics for high-quality and scalable output.

7) Integration with NumPy and pandas: Matplotlib integrates seamlessly with NumPy and pandas, enabling you to plot data directly from NumPy arrays or pandas Data Frames/Series.

8) Backends and Interactive Plots: Matplotlib supports different backends, allowing you to display plots in various environments, such as Jupyter notebooks, interactive GUIs, web applications, and more.

**Q-3 How to load dataset using NumPy? Explain**

**Ans:** NumPy itself does not have built-in functionality for loading datasets directly. However, you can use NumPy in conjunction with other Python libraries, such as pandas, scikit-learn, or numpy.loadtxt, to load datasets into NumPy arrays.

Here are a few common methods for loading datasets using NumPy:

Loading from CSV or Text Files:

Using numpy.loadtxt: This function can be used to load data from text files. You can specify the file path, delimiter, data type, and other parameters. Here's an example:

python

```
import numpy as np


data = np.loadtxt('data.csv', delimiter=',')
```

Using pandas and converting to NumPy array: pandas provides powerful tools for data manipulation, including loading data from various file formats. You can load the data using pandas and then convert it to a NumPy array. Here's an example:

python

```
import pandas as pd

import numpy as np


data_frame = pd.read_csv('data.csv')

data = data_frame.to_numpy()
```

Loading from NumPy-specific file formats:

Using numpy.load: If you have saved your dataset in a NumPy-specific .npy or .npz format, you can use the numpy.load function to load the data directly into a NumPy array. Here's an example:

python

```
import numpy as np


data = np.load('data.npy')
```

Using numpy.fromfile: This function allows you to load data from binary files. You need to specify the file path, data type, and other parameters. Here's an example:

python

Copy code

```
import numpy as np


data = np.fromfile('data.bin', dtype=np.float32)
```

Loading from libraries specifically designed for datasets:

Using scikit-learn or other data-specific libraries: Libraries like scikit-learn provide built-in functions to load specific datasets used for machine learning or data analysis tasks. These functions often return NumPy arrays. Here's an example:

python

```
from sklearn.datasets import load_iris


data = load_iris().data
```

NumPy's flexibility allows you to work with various data loading methods and integrate it seamlessly with other data processing and analysis libraries in Python.

**Q-4 How to plot a vertical lines and horizontal line using matplotlib? Give the code.**

**Ans:** To plot vertical and horizontal lines using Matplotlib, you can use the ‘axvline()’ and ‘axhline()’ functions, respectively. Here's an example code snippet that demonstrates how to plot a vertical line and a horizontal line:

```
import matplotlib.pyplot as plt

# Create a figure and axes
fig, ax = plt.subplots()

# Plot a vertical line
vertical_line_pos = 2.5
ax.axvline(x=vertical_line_pos, color='r', linestyle='--')

# Plot a horizontal line
horizontal_line_pos = 1.8
ax.axhline(y=horizontal_line_pos, color='g', linestyle=':')

# Set x and y limits for better visibility
ax.set_xlim(0, 5)
ax.set_ylim(0, 3)

# Add labels and title
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_title('Vertical and Horizontal Lines')
```

# Display the plot

plt.show()

To plot vertical and horizontal lines using Matplotlib, you can use the axvline() and axhline() functions, respectively. Here's an example code snippet that demonstrates how to plot a vertical line and a horizontal line:

In this code, we create a figure and axes using plt.subplots(). Then, we use axvline() to plot a vertical line at the specified x-coordinate (vertical_line_pos) with a red color and a dashed line style. Similarly, axhline() is used to plot a horizontal line at the specified y-coordinate (horizontal_line_pos) with a green color and a dotted line style.

We set the x and y limits using ax.set_xlim() and ax.set_ylim() to ensure the lines are visible in the plot. Then, we add labels to the x and y axes and provide a title for the plot using ax.set_xlabel(), ax.set_ylabel(), and ax.set_title() functions.

Finally, we display the plot using plt.shw().

**Q-5 Give the application of scikit learn.**

**Ans:** Scikit-learn is applied in various machine learning tasks, including:

**Classification:** Predicting categorical labels, such as classifying emails as spam or not spam.

**Regression:** Predicting continuous values, like predicting housing prices based on features.

**Clustering:** Identifying groups or patterns in unlabeled data, such as customer segmentation.

**Dimensionality Reduction:** Reducing the number of input features while preserving important information.

**Model Selection:** Comparing and selecting the best model for a given problem using cross-validation.

**Pre-processing:** Handling missing data, scaling features, and encoding categorical variables.

**Model Evaluation:** Assessing the performance of machine learning models using metrics like accuracy and precision.

**Hyperparameter Tuning:** Optimizing model performance by tuning hyperparameters using techniques like grid search.

**Q-6 State the features of NumPy.**

**Ans:** The features of NumPy are as follows:

1) N-dimensional arrays: NumPy introduces the ndarray object for efficient storage and manipulation of homogeneous arrays with multiple dimensions.

2) Array operations: NumPy provides a wide range of mathematical and logical operations for arrays, including element-wise operations, linear algebra, Fourier transforms, and random number generation.

3) Broadcasting: NumPy allows arrays with different shapes to be combined and operated upon, enabling efficient computation without the need for explicit loops.

4) Indexing and slicing: NumPy offers powerful indexing and slicing capabilities to access and manipulate specific elements or subsets of array data.

5) Integration with other libraries: NumPy integrates seamlessly with other scientific computing libraries in Python, serving as the foundation for many scientific and data-related operations.

6) Efficiency: NumPy is highly optimized for numerical computations and provides efficient storage and execution of array operations, making it suitable for performance-critical applications.

7) Memory management: NumPy provides flexible memory management options, allowing for efficient handling of large datasets and minimizing memory overhead.

8) Interoperability: NumPy supports interoperability with other data structures and libraries, enabling seamless data exchange and integration in scientific computing workflows.

**Q-7 Explain the data type used in Scikit Learn.**

**Ans:** Scikit-learn primarily uses NumPy arrays to represent data. The data types in scikit-learn are similar to those in NumPy:

1) int: Integer data type for whole numbers.

2) float: Floating-point data type for decimal numbers.

3) bool: Boolean data type for True or False values.

4) str: String data type for text or character data.

5) object: General data type for storing Python objects.

6) Specialized data types like datetime64 for dates and times.

Scikit-learn expects input data to be in a specific format, typically as a two-dimensional array or Data Frame. Features are represented as columns, and the target variable is usually a separate array or column.

Scikit-learn provides utilities to handle different data types, such as encoding categorical variables and scaling numerical variables, ensuring the data is properly prepared for machine learning models.

**Q-8 Explain use of pandas in machine learning.**

**Ans:** Pandas is a powerful library in Python used for data manipulation and analysis. In the context of machine learning, pandas offers several key functionalities that facilitate the data pre-processing and exploratory analysis stages. Here's a brief explanation of the uses of pandas in machine learning:

1) Data Loading: Pandas provides efficient tools to read data from various file formats such as CSV, Excel, SQL databases, and more. It allows you to load data into a Data Frame, a two-dimensional data structure that can store and manipulate labeled data.

2) Data Exploration: Pandas offers functions for inspecting and exploring the dataset. You can examine the data structure, check for missing values, summarize statistics, visualize distributions, and explore relationships between variables.

3) Data Pre-processing: Pandas provides flexible methods to clean and transform data. It allows you to handle missing values, perform data imputation, handle outliers, and apply various transformations like scaling, normalization, and encoding categorical variables.

4) Feature Engineering: Pandas enables feature engineering by creating new features or transforming existing ones. You can perform mathematical operations on columns, extract information from dates, apply text processing techniques, and create interaction features.

5) Data Integration and Merging: Pandas allows you to merge, join, and concatenate datasets. This is useful when working with multiple data sources or combining data from different tables or files.

6) Data Subset Selection: Pandas offers powerful indexing and slicing capabilities to extract specific subsets of data based on conditions, column

names, or row indices. This is crucial for creating training and testing datasets for machine learning models.

7) Data Export: Once the data pre-processing is done, pandas allows you to export the processed data to different file formats for further analysis or to be used as input for machine learning models.

**Q-9 Give the application of matplotlib.**

**Ans:** Matplotlib is a versatile Python library used for data visualization. It offers a wide range of plotting functions and features, making it useful in various applications, including:

1) Data Exploration: Matplotlib helps visualize data distributions, patterns, and relationships to gain insights and make data-driven decisions.

2) Data Presentation: Matplotlib enables the creation of professional-quality plots and charts for presentations, reports, and publications.

3) Trend Analysis: Matplotlib can be used to plot time series data, visualize trends, and analyze patterns over time.

4) Comparisons: Matplotlib facilitates the comparison of multiple datasets or different scenarios through various plot types, such as bar charts, line plots, and scatter plots.

5) Model Evaluation: Matplotlib aids in visualizing evaluation metrics, performance curves, and confusion matrices to assess the performance of machine learning models.

6) Geographic Visualization: Matplotlib supports the creation of maps, geospatial plots, and visualizations of spatial data.

7) Interactive Visualization: Matplotlib can be combined with interactive frameworks like Jupyter notebooks and other libraries to create interactive and dynamic visualizations.

8)Educational Purposes: Matplotlib is widely used in educational settings to teach data visualization concepts and techniques.

**Q-10 Give difference between pandas and NumPy.**

**Ans:**

| PANDAS | NUMPY |
|---|---|
| When we have to work on **Tabular data**, we prefer the p*anda's* module. | When we have to work on **Numerical data**, we prefer the n*umpy* module. |
| The powerful tools of pandas are **Data frame and Series.** | Whereas the powerful tool of *numpy* is **Arrays.** |
| *Pandas* consume **more memory**. | *Numpy* is **memory efficient.** |
| *Pandas* has a better performance when a number of rows is **500K or more.** | *Numpy* has a better performance when number of rows is **50K or less.** |
| Indexing of the *panda's* series is **very slow** as compared to *numpy* arrays. | Indexing of *numpy* Arrays is **very fast**. |
| *Pandas* offer a have2d table object called **Data Frame.** | *Numpy* is capable of providing **multi-dimensional arrays.** |
| It was developed by Wes McKinney and was released in 2008. | It was developed by Travis Oliphant and was released in 2005. |
| It is used in a lot of organizations like Kaidee, Trivago, Abeja Inc., and a lot more. | It is being used in organizations like Walmart Tokopedia, Instacart, and many more. |
| It has a higher industry application. | It has a lower industry application. |

**Q-11 How can we create 2D array in NumPy?**

**Ans:** To create a 2D array in NumPy, you can use the numpy.array() function by passing a nested list or a tuple of lists as the input. Each inner list represents a row in the 2D array. Here's an example:

python

Copy code

```
import numpy as np

# Create a 2D array using a nested list
my_array = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])

print(my_array)
```

Output:

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
```

In this example, we create a 2D array called my_array using the numpy.array() function. The nested list [[1, 2, 3], [4, 5, 6], [7, 8, 9]] represents three rows in the array. Each inner list corresponds to a row, and the elements within each inner list represent the values in that row.

The resulting output displays the created 2D array:

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
```

You can access individual elements in the 2D array using indexing. For example, my_array[0, 1] would give you the element at the first row, second column, which is 2 in this case.

**Q-12 Explain features of scikit learn.**

**Ans:** Some key features of scikit-learn:

1) Supervised Learning: Scikit-learn offers a variety of algorithms for supervised learning tasks, including regression and classification.

2) Unsupervised Learning: Scikit-learn provides algorithms for unsupervised learning tasks such as clustering and dimensionality reduction.

3) Data Pre-processing: Scikit-learn offers tools for handling missing data, scaling features, encoding categorical variables, and more.

4) Model Evaluation: Scikit-learn provides evaluation metrics and techniques for assessing the performance of machine learning models.

5) Model Selection: Scikit-learn offers methods for selecting the best model and tuning hyperparameters using techniques like grid search and cross-validation.

6) Pipelines: Scikit-learn supports the creation of data processing pipelines, simplifying the workflow and ensuring consistency.

7) Integration: Scikit-learn integrates seamlessly with other Python libraries, such as NumPy and pandas, for efficient data manipulation and analysis.

8) Scalability: Scikit-learn is designed to handle large datasets efficiently and offers optimized implementations of algorithms for scalability.

**Q-13 How can we plot a horizontal line in matplotlib?**

**Ans:** Matplotlib is a popular python library used for plotting, It provides an object-oriented API to render GUI plots

Plotting a horizontal line is fairly simple, Using **axhline()**
The axhline() function in pyplot module of matplotlib library is used to add a horizontal line across the axis.
***Syntax:*** *matplotlib.pyplot.axhline(y, color, xmin, xmax, linestyle)*

EXAMPLE:

```
# importing library

import matplotlib.pyplot as plt

# specifying horizontal line type

plt.axhline(y = 0.5, color = 'r', linestyle = '-')
```

# rendering the plot

plt.show()

## Q-14 Explain the steps to import csv file in pandas.

**Ans:** CSV files are the “comma separated values”, these values are separated by commas, this file can be view like as excel file. In Python, Pandas is the most important library coming to data science. We need to deal with huge datasets while analyzing the data, which usually can get in CSV file format.

Let’s see the different ways to import csv file in Pandas.

**Method #1:** Using read_csv() method.

```
# importing pandas module
import pandas as pd

# making data frame
df = pd.read_csv("https://media.geeksforgeeks.org/wp-content/uploads/nba.csv")

df.head(10)
```

**Method #2:** Using `csv` module.
One can directly import the csv files using `csv` module.

```
# import the module csv
import csv
import pandas as pd

# open the csv file
with open(r"C:\Users\Admin\Downloads\nba.csv") as csv_file:

    # read the csv file
    csv_reader = csv.reader(csv_file, delimiter=',')

    # now we can use this csv files into the pandas
    df = pd.DataFrame([csv_reader], index=None)
    df.head()

# iterating values of first column
for val in list(df[1]):
    print(val)
```

**Q-15 Explain any three applications of pandas.**

**Ans:** Pandas is a powerful library in Python that offers a wide range of functionalities for data manipulation and analysis. Here are three applications of pandas:

1) Data Cleaning and Pre-processing: Pandas provides powerful tools for handling missing data, removing duplicates, transforming data, and handling outliers. It allows users to clean and pre-process datasets by performing operations such as data imputation, data normalization, and feature scaling.

2) Exploratory Data Analysis (EDA): Pandas simplifies the process of exploring and understanding datasets. It enables users to perform tasks such as summarizing data, calculating descriptive statistics, grouping and aggregating data, and visualizing data distributions. Pandas integrates well with visualization libraries, allowing for the creation of insightful plots and charts.

3) Data Manipulation and Transformation: Pandas offers flexible and efficient methods for manipulating and transforming data. It allows users to filter rows based on conditions, select specific columns, create new columns based on computations, merge datasets, reshape data structures, and perform complex data transformations. This makes pandas valuable for tasks such as data wrangling, data reshaping, and data integration.

**Q-16 Write a python program to load the iris data from given csv file into a data frame print the shape of the data, type of the data and first 3 rows using scikit learn.**

**Ans: Python Code:**

```
import pandas as pd

data = pd.read_csv("iris.csv")

print("Shape of the data:")

print(data.shape)

print("\nData Type:")

print(type(data))

print("\nFirst 3 rows:")

print(data.head(3))
```

**Output:**

```
Shape of the data:
(150, 6)

Data Type:
<class 'pandas.core.frame.DataFrame'>

First 3 rows:
   Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
Species
0   1            5.1           3.5            1.4           0.2  Iris-setosa
1   2            4.9           3.0            1.4           0.2  Iris-setosa
2   3            4.7           3.2            1.3           0.2  Iris-setosa
```

**Q-17 How can you split data into training and test data using pandas? Give the code.**

**Ans:** To split data into training and test datasets using pandas, you can use the train_test_split() function from scikit-learn. Here's an example of how to do it:

```
import pandas as pd
from sklearn.model_selection import train_test_split

# Load the dataset into a pandas Data Frame
data = pd.read_csv('dataset.csv')

# Split the data into features (X) and target variable (y)
X = data.drop('target_variable', axis=1)  # Replace 'target_variable' with the name of your target column
y = data['target_variable']

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In this code, we first import the pandas library and the train_test_split() function from scikit-learn.

Next, we load the dataset from the CSV file into a pandas DataFrame named data. You need to replace 'dataset.csv' with the path and name of your actual dataset file.

Then, we split the data into features (X) and the target variable (y). You should replace 'target_variable' with the name of your target variable column.

Finally, we use the train_test_split() function to split the data into training and test sets. The test_size parameter specifies the proportion of the data to be used for testing (e.g., test_size=0.2 means 20% of the data will be used for testing). The random_state parameter sets the seed for random shuffling of the data before splitting, ensuring reproducibility.

The resulting training and test datasets are stored in X_train, X_test, y_train, and y_test, respectively.