Data Engineering Techniques (AID30010) — Unit IV

Learning Intelligent Data Patterns: Supervised and Unsupervised Learning

Structured Question Bank

PART B: LONG ANSWER QUESTIONS (5 marks / 10 marks)

Introduction to Data Mining

Q1: What is Data Mining? Explain its need and applications.

Answer:

1. Data Mining — Definition

- Data Mining is the **extraction of interesting, non-trivial, implicit, previously unknown, and potentially useful patterns or knowledge** from a huge amount of data
- Also called **Knowledge Discovery in Databases (KDD)**
- Other names: knowledge extraction, data archeology, information harvesting, business intelligence
- Data Mining = Exploration & analysis, by automatic or semi-automatic means, of large quantities of data to discover meaningful patterns
- It is NOT simple search/query processing or expert systems — patterns must be valid, novel, useful, and understandable

2. Need / Why Data Mining?

- The **explosive growth of data** — from terabytes to petabytes to zettabytes
- Major sources: Business (Web, e-commerce, transactions), Science (remote sensing, bioinformatics), Society (news, YouTube, digital cameras)
- **"We are drowning in data, but starving for knowledge!"**
- Computers have become cheaper and more powerful — analysis is now feasible
- Competitive pressure: organizations need insights from data for better decision-making and customized services

3. What Data Mining is NOT

- Simple lookup/query is NOT data mining
- Expert systems are NOT data mining
- The discovered patterns must be: **Valid, Novel, Potentially Useful, Understandable**

4. Data Mining Tasks — Two Main Types

- **Predictive Methods:** Use some variables to predict unknown or future values of other variables
  - Examples: Classification, Regression, Time Series Analysis, Prediction
- **Descriptive Methods:** Find human-interpretable patterns that describe the data
  - Examples: Clustering, Summarization, Association Rules, Sequence Discovery

5. Applications of Data Mining

- Direct marketing and Customer Relationship Management (CRM)
- Health industry (predicting patient outcomes)
- E-commerce (recommendation systems)
- Telecommunication and financial sector (fraud detection)
- Specialized forms: Text mining, web mining, audio/video mining, social network mining

---

Q2: Explain Data Mining Models and Tasks with a suitable diagram.

Answer:

1. Data Mining — Two Main Models

- **Predictive Model:** Makes predictions about unknown data using known results
- **Descriptive Model:** Identifies patterns or relationships in data for human interpretation

2. Predictive Tasks

Task	Description
Classification	Predicts categorical class labels (discrete)
Regression	Predicts continuous/numeric values
Time Series Analysis	Makes predictions over time
Prediction	General prediction of unknown values

3. Descriptive Tasks

Task	Description
Clustering	Groups similar data objects together
Summarization	Compact description of a subset of data
Association Rules	Finds items that frequently occur together
Sequence Discovery	Finds sequential patterns in data

4. Data Mining Process Flow

```
Input Data → Data Preprocessing → Data Mining → Post-processing → Information
                    ↓                                  ↓
          (Feature Selection,                (Filtering Patterns,
           Dimensionality Reduction,          Visualization,
           Normalization,                     Pattern Interpretation)
           Data Subsetting)
```

5. Important Points

- The same data can lead to different results depending on the mining task chosen
- Predictive tasks require labeled training data; descriptive tasks do not
- Selection of the right task depends on the business goal
- Data preprocessing is always required before mining
- Patterns discovered must meet the criteria: valid, novel, useful, understandable

---

Supervised, Semi-Supervised, and Unsupervised Learning

Q3: Define Supervised and Unsupervised Learning. Explain each with example.

Answer:

1. Supervised Learning — Definition

- Training data is **accompanied by labels** indicating the class of each observation
- The model learns the mapping from inputs to outputs using labeled examples
- New/unseen data is classified based on what was learned from the training set
- Also called **Predictive Learning**
- Goal: Infer a function from labeled training data

2. Methods under Supervised Learning

- **Classification:** Target attribute is finite/categorical (e.g., yes/no, spam/not spam)

    - A classifier assigns a class label to unseen examples
    - Example: Deciding if a patient needs ICU based on blood pressure, age, etc.
- **Regression:** Target attribute has infinite/continuous values
    - Fits a model to learn output as a function of input attributes
    - Example: Predicting house price
- **Time Series Analysis:** Making predictions over time

3. Unsupervised Learning — Definition

- Class labels of training data are **unknown** — no supervisor
- The aim is to find regularities, similarities, associations in the input on its own
- Also called **Descriptive Learning**
- Goal: Find hidden structure in unlabeled data

4. Methods under Unsupervised Learning

- **Clustering:** Groups data into clusters based on similarity
- **Association Rules:** Finds items that frequently appear together
- **Pattern Mining:** More general term for finding frequent patterns
- **Outlier Detection:** Finding data points that behave very differently from others

5. Differences — Supervised vs. Unsupervised

Feature	Supervised Learning	Unsupervised Learning
Labels	Present (labeled data)	Absent (unlabeled data)
Goal	Predict output for new data	Discover hidden structure
Examples	Classification, Regression	Clustering, Association
Algorithms	Naive Bayes, Decision Trees, SVM	K-Means, DBSCAN
Human involvement	High (labels needed)	Low

Q4: Define Semi-Supervised Learning. How is it different from supervised and unsupervised learning?

Answer:

1. Semi-Supervised Learning — Definition

- A class of machine learning tasks that make use of **both labeled and unlabeled data** for training

- Typically uses a **small amount of labeled data** with a **large amount of unlabeled data**
- Falls **between** unsupervised (no labels) and supervised (all labels) learning
- Useful when labeling data is expensive or time-consuming

2. Why Semi-Supervised Learning?

- Getting labeled data is costly — requires human experts
- Unlabeled data is cheap and abundant
- Semi-supervised combines both to get better results than using only labeled data
- Used widely in text classification, image recognition, web content classification

3. Position Among Learning Types

```
Unsupervised ←——— Semi-Supervised ———→ Supervised
(No labels)       (Some labels +       (All labels)
                   Lots of unlabeled)
```

4. Algorithms Used in Supervised Learning

Algorithm Type	Examples
Supervised	Naïve Bayes, Support Vector Machines, Random Forests, Decision Trees
Unsupervised	K-Means, Fuzzy Clustering, Hierarchical Clustering

5. Key Comparison

Type	Labels	Data Requirement	Example
Supervised	Fully labeled	Needs large labeled dataset	Email spam filter
Unsupervised	No labels	Only input data	Customer segmentation
Semi-Supervised	Partially labeled	Small labeled + large unlabeled	Google image search

---

Classification

Q5: What is Classification? Explain the Classification Process in detail.

Answer:

1. Classification — Definition

- Classification maps data into **predefined groups or classes**

- It is a form of supervised learning where a model is built from training data and then used to classify unknown data
- Used in: Pattern recognition, Prediction, Supervised learning
- The target attribute (class label) is **discrete or categorical**

2. Classification vs. Prediction

Feature	Classification	Prediction
Output Type	Categorical/Discrete labels	Continuous/Numeric values
Example	Spam/Not Spam	House price
Technique	Decision Tree, Naive Bayes	Regression
Known as	Supervised classification	Supervised regression

3. Applications of Classification

- Credit approval (approved/not approved)
- Target marketing (likely buyer/not)
- Medical diagnosis (high-risk/low-risk patient)
- Fraud detection (fraudulent/genuine transaction)

4. Classification Process — Two Steps

Step 1: Model Construction (Training Phase)

- Training set is used to build the classifier (model)
- Each tuple belongs to a predefined class as determined by the class label attribute
- Model is represented as classification rules, decision trees, or mathematical formulae
- Example Rule: `IF rank = 'professor' OR years > 6 THEN tenured = 'yes'`

Step 2: Model Usage (Testing/Classification Phase)

- Estimate accuracy of the model using test set
- Known labels of test samples are compared with classified results
- **Accuracy Rate** = percentage of test samples correctly classified
- Test set must be **independent of training set** — otherwise **over-fitting** occurs
- If accuracy is acceptable → use model to classify unknown data

5. Important Points

- Training set and test set must be independent
- Model evaluation uses accuracy rate

- Over-fitting occurs when model performs well on training but poorly on new data
- Multiple classification algorithms exist: Decision Trees, Naïve Bayes, SVM, Random Forest
- Choice of algorithm depends on data type, size, and required accuracy

---

Q6: Explain Decision Tree Induction in detail with its algorithm and example.

Answer:

1. Decision Tree — Definition

- A tree-structured classifier where:
  - **Root Node:** Top-most special node (starting point)
  - **Internal Nodes:** Test attributes/decision attributes (conditions)
  - **Leaf Nodes (Terminal Nodes):** Class labels (final answers)
  - **Edges/Branches:** Outcomes for each value of the test attribute
- Decision tree can be **binary** (two branches) or **multi-class** (multiple branches)
- No attribute is repeated along any path in the tree

2. Decision Tree Terminologies

- **Root Node:** Starting point of the tree; represents the best splitting attribute
- **Internal Nodes:** Each contains a test condition on an attribute
- **Leaf Nodes:** Represent class labels (final prediction output)
- **Branches:** Possible outcomes/values for each tested attribute
- The decision tree is **not unique** — different orderings of attributes give different trees

3. Decision Tree Algorithm (Greedy/Top-Down Recursive)

- Tree is constructed in a **top-down, recursive, divide-and-conquer** manner
- At start, all training examples are at the root
- Attributes must be categorical (continuous values are discretized in advance)
- Examples are partitioned recursively based on selected attributes
- Test attribute is selected using **Information Gain** (highest gain attribute is chosen)

Stopping Conditions:

- All samples for a node belong to the same class
- No remaining attributes for further partitioning (majority voting used)
- No samples left

4. Information Gain — Formula

Step 1: Calculate Entropy (Expected Information) for Dataset D:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

where p_i = probability that a tuple belongs to class C_i

Step 2: Calculate Information needed after splitting on attribute A:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times I(D_j)$$

Step 3: Calculate Information Gain:

$$Gain(A) = Info(D) - Info_A(D)$$

→ Select the attribute with the **highest Gain** as the splitting attribute

5. Rule Extraction from Decision Tree

- One rule is created for each path from the root to a leaf
- Each attribute-value pair along the path forms a conjunction (AND condition)
- Rules are mutually exclusive and exhaustive
- Example rules from buys_computer dataset:
  - `IF age = young AND student = no → buys_computer = no`
  - `IF age = young AND student = yes → buys_computer = yes`
  - `IF age = mid-age → buys_computer = yes`
  - `IF age = old AND credit_rating = excellent → buys_computer = yes`
  - `IF age = old AND credit_rating = fair → buys_computer = no`

Q7: Solve the following: Calculate Information Gain and build a Decision Tree for the given dataset.

Answer: (Standard buys_computer dataset — Exam Solved Example)

Given Dataset (14 tuples):

- Class P: buys_computer = "yes" → 9 tuples
- Class N: buys_computer = "no" → 5 tuples

Step 1: Calculate Entropy of Whole Dataset

$$Info(D) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940$$

Step 2: Calculate Information for each attribute

For **age** attribute (split into: ⩽30, 31-40, >40):

Age	p_i	n_i	$I(p_i, n_i)$
⩽30	2	3	0.971
31-40	4	0	0
>40	3	2	0.971

$$Info_{age}(D) = \frac{5}{14}(0.971) + \frac{4}{14}(0) + \frac{5}{14}(0.971) = 0.694$$

Step 3: Calculate Gain

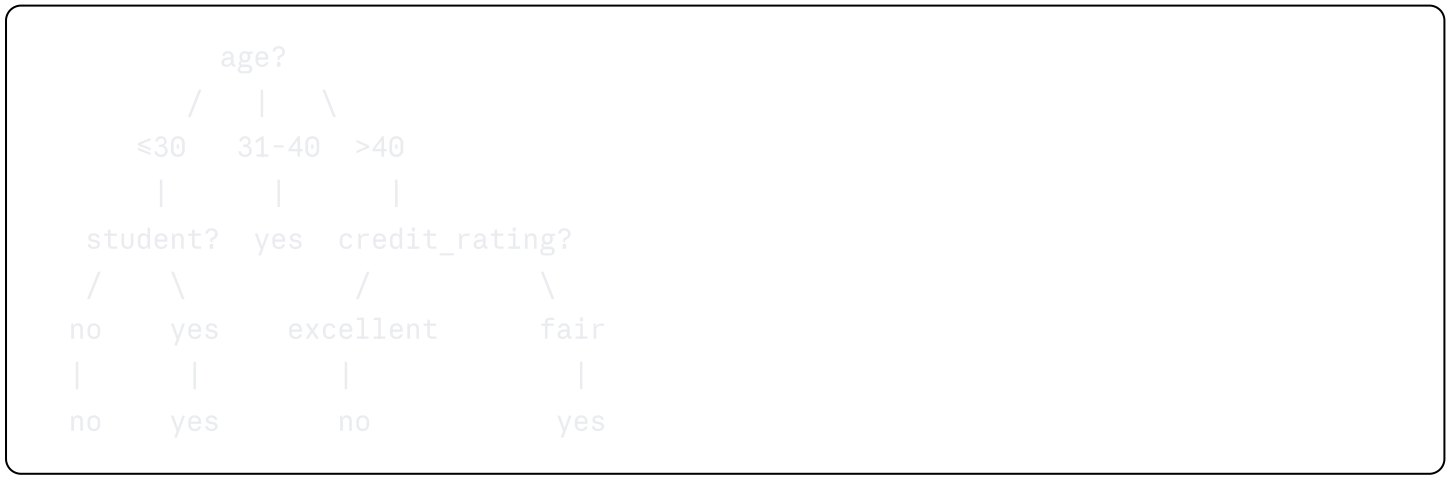
$$Gain(age) = 0.940 - 0.694 = 0.246$$

$$Gain(income) = 0.029, \quad Gain(student) = 0.151, \quad Gain(credit\_rating) = 0.048$$

Step 4: Select Root

- Gain(age) > all others → **age is selected as root node**

Final Decision Tree:



5. Key Points to Remember

- Always calculate entropy first for the full dataset
- Calculate Info for each attribute separately

- Gain = Info(D) − Info_A(D)
- Attribute with **maximum Gain** becomes root node
- Repeat process recursively for each branch until stopping condition is met

---

Q8: Explain the Bayesian Classification in detail with the Naïve Bayes theorem and example.

Answer:

1. Bayesian Classification — Definition

- A **statistical classifier** that performs probabilistic prediction — predicts class membership probabilities
- Based on **Bayes' Theorem**
- A simple Bayesian classifier (Naïve Bayes) has comparable performance with decision trees
- **Incremental:** Each training example can increase/decrease the probability of a hypothesis
- **Standard:** Provides an optimal decision-making standard against which other methods are measured

2. Probability Basics

$$P(\text{event}) = \frac{\text{Number of ways it can happen}}{\text{Total number of outcomes}}$$

Key Terms:

- **P(H)** — Prior probability: initial probability of hypothesis H
- **P(X)** — Probability that sample X is observed
- **P(X|H)** — Likelihood: probability of observing X given H holds
- **P(H|X)** — Posterior probability: probability H holds given observed X

3. Bayes' Theorem

$$P(H|X) = \frac{P(X|H) \cdot P(H)}{P(X)}$$

Informally:

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

→ Predict X belongs to class C_i **iff P(C_i|X) is the highest** among all classes

4. Naïve Bayes Classifier

- Since P(X) is constant for all classes, maximize: $\mathbf{P(C_i|X) = P(X|C_i) \times P(C_i)}$
- Assumption: **Class Conditional Independence** — attributes are independent of each other given the class
- This "naïve" assumption makes calculation much simpler

Steps:

1. Calculate prior probability $P(C_i)$ for each class
2. Calculate conditional probability $P(x_j|C_i)$ for each attribute value
3. Calculate $P(X|C_i)$ = product of all $P(x_j|C_i)$
4. Calculate $P(X|C_i) \times P(C_i)$ for each class
5. Assign X to the class with the **highest** $P(X|C_i) \times P(C_i)$

5. Naïve Bayes — Solved Example

Test sample: X = (age ⩽ 30, income = medium, student = yes, credit_rating = fair)

Prior Probabilities:

- P(buys_computer = "yes") = 9/14 = 0.643
- P(buys_computer = "no") = 5/14 = 0.357

Conditional Probabilities for "yes" class:

- P(age ⩽ 30 | yes) = 2/9 = 0.222
- P(income = medium | yes) = 4/9 = 0.444
- P(student = yes | yes) = 6/9 = 0.667
- P(credit_rating = fair | yes) = 6/9 = 0.667

Conditional Probabilities for "no" class:

- P(age ⩽ 30 | no) = 3/5 = 0.6
- P(income = medium | no) = 2/5 = 0.4
- P(student = yes | no) = 1/5 = 0.2
- P(credit_rating = fair | no) = 2/5 = 0.4

Calculating $P(X|C_i)$:

- P(X | yes) = 0.222 × 0.444 × 0.667 × 0.667 = **0.044**
- P(X | no) = 0.6 × 0.4 × 0.2 × 0.4 = **0.019**

Final Score:

- P(X|yes) × P(yes) = 0.044 × 0.643 = **0.028**

- $P(X|no) \times P(no) = 0.019 \times 0.357 = \mathbf{0.007}$

0.028 > 0.007 → X belongs to class "buys_computer = yes"

Q9: What are the Advantages and Disadvantages of Naïve Bayesian Classification?

Answer:

1. Advantages of Naïve Bayes

- **Easy to implement** — simple probability calculations
- **Fast** — both training and prediction are quick
- **Good results in most cases** — especially text classification (spam filtering)
- **Incremental** — can update model as new training data arrives without retraining from scratch
- Works well even with **small amounts of training data**

2. Disadvantages of Naïve Bayes

- **Class conditional independence assumption** — in reality, attributes are often correlated → loss of accuracy
- **Zero frequency problem** — if a category was never seen in training, it gets probability 0 (solved by Laplace smoothing)
- Not suitable for tasks where **attribute dependencies** are important
- Can be outperformed by more advanced models like Random Forest or SVM
- Not ideal for **high-dimensional numerical data**

3. Comparison: Decision Tree vs. Naive Bayes

Feature	Decision Tree	Naïve Bayes
Type	Rule-based	Probabilistic
Interpretability	High (visual tree)	Medium
Handles missing data	Yes	Yes
Assumption	None	Feature independence
Speed	Moderate	Fast
Accuracy	Good for structured data	Good for text data

4. When to Use Naïve Bayes

- Text classification (spam detection, sentiment analysis)

- Real-time prediction applications
- Multi-class problems
- When training data is limited

5. Key Exam Points

- Naïve = assumes features are independent given class label
- It is a probabilistic classifier, not rule-based
- Posterior probability = Likelihood × Prior / Evidence
- Predict the class with highest posterior probability
- Naïve Bayes has comparable performance to Decision Trees in many cases

---

Clustering Techniques

Q10: What is Clustering? Explain its basic concepts, types, and desirable properties.

Answer:

1. Clustering — Definition

- Clustering is the task of **dividing data points into groups (clusters) such that:**
  - Points in the **same cluster** are more **similar** to each other
  - Points in **different clusters** are more **dissimilar** to each other
- Also called **unsupervised classification** — no predefined class labels
- Goal: Find natural groupings in data
- **Intra-cluster distances are minimized; Inter-cluster distances are maximized**

2. Key Differences — Clustering vs. Classification

Feature	Clustering	Classification
Type	Unsupervised	Supervised
Labels	No predefined class labels	Predefined class labels
Goal	Find natural groups	Assign to known groups
Examples	K-Means, DBSCAN	Decision Tree, Naïve Bayes

3. Types of Clustering

Type	Description	Example
Partitional	Divides data into k non-overlapping groups; each object in exactly one cluster	K-Means, K-Medoids
Hierarchical	Creates a hierarchical tree (dendrogram) of nested clusters	Agglomerative, Divisive
Density-Based	Based on density functions; finds clusters of arbitrary shape	DBSCAN
Grid-Based	Divides space into finite grid cells; operations on grids	STING
Model-Based	Assumes a model for each cluster; finds best fit	SOM (Self-Organizing Maps)

4. Desirable Properties of Clustering

- **Scalability:** Efficient in both time and space for large datasets
- **Ability to handle different data types:** Works with numeric, categorical, binary data
- **Minimal domain knowledge:** Minimal input parameters needed
- **Handles noise and outliers:** Robust to noisy data
- **Insensitive to input order:** Results should not change with different data ordering
- **Supports user constraints:** Allows domain-specific constraints to be incorporated

5. Applications of Clustering

- Customer segmentation in marketing
- Document clustering in text mining
- Image segmentation in computer vision
- Anomaly/fraud detection in banking
- Gene expression analysis in bioinformatics

Q11: Explain K-Means Clustering Algorithm in detail with steps, algorithm, example, and its advantages and disadvantages.

Answer:

1. K-Means Clustering — Definition

- K-Means is an **unsupervised partitional clustering algorithm** that groups n objects into **k clusters** based on their attributes
- **K** = pre-defined number of clusters (positive integer, $k < n$)

- It is a **centroid-based algorithm** — each cluster is associated with a centroid (mean/center point)
- Goal: **Minimize the sum of squared distances** between data points and their cluster centroid
- Assumes object attributes form a **vector space**

2. Objective Function (Sum-of-Squares Criterion)

$$J = \sum_{j=1}^{K} \sum_{n \in S_j} |x_n - \mu_j|^2$$

where:

- x_n = vector representing the nth data point
- μ_j = geometric centroid (mean) of data points in cluster S_j
- Goal = minimize J

3. Algorithm of K-Means

Input: k (number of clusters), D (dataset with n objects) **Output:** A set of k clusters

Method:

1. Arbitrarily choose k objects from D as initial cluster centers
2. **Repeat:**
   - (Re)assign each object to the cluster to which it is **most similar** (nearest centroid, based on distance/mean)
   - Update the cluster means → recalculate mean value of objects in each cluster
3. **Until** no change (no objects move to a different cluster)
4. Return resulting clusters

4. Working — Step by Step

- Step 1: Decide value of k (number of clusters)
- Step 2: Randomly select k objects as initial cluster centroids
- Step 3: Calculate distance of each data point from all k centroids
- Step 4: Assign each data point to the nearest centroid → forms k clusters
- Step 5: Recalculate mean of each cluster (new centroid)
- Step 6: Reassign data points to nearest new centroid
- Step 7: Repeat steps 5-6 until no reassignment occurs (convergence)
- Step 8: Return final clusters

5. Solved Example

Given: {2, 3, 6, 8, 9, 12, 15, 18, 22}, k = 3

Iteration 1 — Initial random partition:

- K1 = {2, 8, 15} → mean = 8.3
- K2 = {3, 9, 18} → mean = 10
- K3 = {6, 12, 22} → mean = 13.3

Reassign based on new means:

- K1 = {2, 3, 6, 8, 9} → mean = 5.6
- K2 = {} → mean = 0
- K3 = {12, 15, 18, 22} → mean = 16.75

Next iteration:

- K1 = {3, 6, 8, 9} → mean = 6.5
- K2 = {2} → mean = 2
- K3 = {12, 15, 18, 22} → mean = 16.75

Next iteration:

- K1 = {6, 8, 9} → mean = 7.6
- K2 = {2, 3} → mean = 2.5
- K3 = {12, 15, 18, 22} → mean = 16.75

No change → STOP

Final Clusters:

- K1 = {6, 8, 9}
- K2 = {2, 3}
- K3 = {12, 15, 18, 22}

Q12: What are the Advantages, Disadvantages, Weaknesses, and Applications of K-Means Clustering?

Answer:

1. Advantages of K-Means

- Relatively **scalable and efficient** — good performance on large datasets
- **Computational complexity** = O(nkt) where n = objects, k = clusters, t = iterations (normally k << n, t << n)
- Simple to understand and implement

- Converges to a local minimum in finite iterations
- Works well when clusters are **compact, well-separated, and spherical**

2. Disadvantages of K-Means

- User must **specify k in advance** — choosing the wrong k gives poor results
- Results depend on **initial random assignment** — different runs may give different clusters
- **Sensitive to outliers and noise** — a single outlier can distort the cluster mean significantly
- Can only be applied when the **mean of a cluster is defined** (doesn't work for categorical data)
- Cannot handle clusters of **very different sizes or non-spherical shapes**

3. Weaknesses (Important for Exam)

- When data is small, **initial grouping determines the cluster significantly**
- Does **not yield same result** with each run
- The real cluster is unknown — different input order may produce different clusters
- May be **trapped in local optimum** instead of global optimum
- Must know k beforehand — difficult for unknown data

4. Applications of K-Means

- **Customer Segmentation:** Group customers by purchasing behavior, demographics, preferences for targeted marketing
- **Document Clustering:** Cluster documents by word frequency/topic — useful for information retrieval
- **Anomaly Detection:** Points far from any cluster are flagged as outliers — useful in fraud detection, network intrusion
- **Image Segmentation/Vector Quantization:** Convert image waveforms into k categories
- **Machine Learning preprocessing:** Used for dimensionality reduction, data compression

5. Advantages vs. Disadvantages — Summary Table

Aspect	Advantage	Disadvantage
Speed	Fast, O(nkt)	Slow for very large k
Ease	Simple algorithm	Need to specify k
Scalability	Works on large data	Sensitive to outliers
Consistency	Converges always	Not always same result
Shape handling	Works for spherical clusters	Fails for arbitrary shapes

Q13: Categorize the following applications as Supervised or Unsupervised Learning. Justify your answer.

Answer:

The following applications are categorized based on whether labeled training data is required:

I. Spam and Legitimate Mail Classification → Supervised Learning

- Training data consists of emails labeled as "spam" or "not spam"
- The classifier learns from these labels to classify new emails
- Predefined classes exist → Supervised

II. Handwritten Character Recognition → Supervised Learning

- Training data contains images of handwritten characters with known labels (A, B, C... or 0-9)
- Model learns the pattern for each character class
- Predefined categories → Supervised

III. Flower Type Recognition → Supervised/Unsupervised (depends on context)

- If trained with labeled flower images → **Supervised**
- If grouping unknown flower images by visual similarity → **Unsupervised (Clustering)**
- Most commonly treated as **Supervised** in classification context

IV. Customer Segmentation → Unsupervised Learning

- No predefined classes for customer groups
- Algorithm finds natural groupings (clusters) based on behavior, demographics, purchase history
- No labels available → Unsupervised (Clustering — K-Means)

V. Disease Type Identification → Supervised Learning

- If training data with known disease labels is available → Supervised Classification
- Model predicts disease class for new patient data
- Examples: Predicting cancer type, diabetes diagnosis

Summary Table:

Application	Type	Algorithm Example
Spam/Legitimate Mail	Supervised	Naive Bayes, SVM
Handwritten Character Recognition	Supervised	CNN, Decision Tree
Flower Type Recognition	Supervised	KNN, Decision Tree
Customer Segmentation	Unsupervised	K-Means Clustering
Disease Type Identification	Supervised	Naive Bayes, Decision Tree

---

Q14: Explain the Training and Testing Phase of Classification in detail.

Answer:

1. Overview

- Classification is a **two-phase process**:
  - Phase 1: Model Construction (Training Phase)
  - Phase 2: Model Usage (Testing + Classification Phase)

2. Phase 1: Model Construction (Training Phase)

- A **training set** of labeled tuples is used
- Each tuple is assumed to belong to a predefined class determined by the **class label attribute**
- A classification algorithm analyzes the training data and builds a **Classifier (Model)**
- The model can be represented as:
  - Classification rules (IF-THEN rules)
  - Decision trees
  - Mathematical formulae
- Example: From a training set of professor data → learns rule: `IF rank = professor OR years > 6 THEN tenured = yes`

3. Phase 2: Model Usage (Testing / Classification Phase)

- A separate **test set** (independent of training set) is used
- Known labels of test samples are compared with the classifier's output
- **Accuracy Rate** = (Number of correctly classified test tuples / Total test tuples) × 100%
- If accuracy is acceptable → model is used to classify **unseen/unknown data**
- If accuracy is not acceptable → model is retrained or algorithm is changed

4. Why Must Test Set Be Independent of Training Set?

- If the same data is used for training and testing → model memorizes answers → **over-fitting**
- Over-fitting: Model performs very well on training data but poorly on new/unseen data
- Independent test set gives a realistic measure of model accuracy

5. Key Points to Remember

Aspect	Training Phase	Testing Phase
Data Used	Training set (labeled)	Test set (labeled for comparison)
Goal	Build the classifier/model	Evaluate accuracy of model
Output	Classification model (rules/tree)	Accuracy rate + Classification of unseen data
Labels	Used to train	Used to verify correctness
Data overlap	Must NOT overlap with test set	Must NOT overlap with training set

---

Additional Practice Questions (Frequently Asked in Exams)

Q15: Compare Supervised vs. Unsupervised Learning Algorithms.

Answer:

1. Comparison Table

Feature	Supervised Learning	Unsupervised Learning
Definition	Learns from labeled data	Learns from unlabeled data
Class labels	Required	Not required
Goal	Predict class/value for new data	Discover hidden patterns
Type of task	Classification, Regression	Clustering, Association
Human effort	High (needs labeling)	Low
Output	Predefined classes	Natural groups/patterns
Algorithms	Decision Tree, Naive Bayes, SVM, Random Forest	K-Means, DBSCAN, Hierarchical Clustering
Examples	Email spam detection, Credit approval	Customer segmentation, Document clustering

2. Supervised Learning — Key Algorithms

- **Naïve Bayes:** Probabilistic classifier based on Bayes' Theorem
- **Decision Tree:** Tree-based model using Information Gain
- **Support Vector Machines (SVM):** Finds optimal hyperplane to separate classes
- **Random Forests:** Ensemble of multiple decision trees

3. Unsupervised Learning — Key Algorithms

- **K-Means:** Partition-based clustering (specify k in advance)
- **Fuzzy Clustering:** Objects can belong to multiple clusters with a degree
- **Hierarchical Clustering:** Creates a dendrogram (tree) of nested clusters

4. When to Use What?

- Use **Supervised** when: labeled data is available, goal is prediction, output classes are known
- Use **Unsupervised** when: no labels available, goal is exploration, finding natural patterns in data
- Use **Semi-Supervised** when: small labeled data + large unlabeled data, labeling is expensive

5. Exam Important Points

- Supervised = labeled data = predict output
- Unsupervised = no labels = find structure

- K-Means is unsupervised; Decision Tree and Naïve Bayes are supervised
- Semi-supervised falls between both — uses small labeled + large unlabeled data
- Classification is supervised; Clustering is unsupervised

Summary: Unit IV — Key Definitions at a Glance

Term	One-line Definition
Data Mining	Extraction of implicit, useful patterns from large datasets
Supervised Learning	Learning from labeled training data to predict output
Unsupervised Learning	Finding patterns/structure in unlabeled data
Semi-Supervised Learning	Uses small labeled + large unlabeled data
Classification	Maps data into predefined categorical classes
Decision Tree	Tree-based classifier using root, internal, and leaf nodes
Information Gain	Measure of effectiveness of an attribute in classifying data
Entropy	Measure of impurity/disorder in a dataset
Naïve Bayes	Probabilistic classifier based on Bayes' theorem with independence assumption
Clustering	Grouping similar data points together without predefined classes
K-Means	Partition-based clustering algorithm that groups data into k clusters
Centroid	Mean/center point of a cluster in K-Means
Over-fitting	Model performs well on training data but poorly on new/unseen data
Accuracy Rate	Percentage of correctly classified test samples

---

Frequently Asked Exam Questions — Quick Reference

Question Pattern	Important Topic
Define Data Mining / Need / Applications	Introduction to Data Mining
Supervised vs Unsupervised	Learning Types Comparison

Question Pattern	Important Topic
Explain Classification Process	Two phases: Model Construction + Model Usage
Build Decision Tree / Calculate Information Gain	Decision Tree Algorithm + Formulae
Naïve Bayes with solved example	Bayesian Classification
Categorize applications as supervised/unsupervised	Classification of real-world problems
Training and Testing phase	Classification Process
Explain Clustering / Types	Clustering Concepts
K-Means Algorithm with example	K-Means Step-by-Step
Advantages and disadvantages of K-Means	K-Means evaluation