🧠 Unit 4 – Classification (Data Mining – 2101CS521)

🟩 1. Basic Concepts

🔹 Types of Data Mining Tasks

Data mining can find different kinds of patterns in data.
There are two main types of tasks:

1. **Descriptive Tasks**
    - Describe general properties and patterns of data.
    - Help to understand what is in the database.
    - Examples: finding frequent patterns, correlations, associations.
2. **Predictive Tasks**
    - Predict the value of one attribute using others.
    - Examples: predicting customer sales during festivals or predicting product demand.

🔹 What is Classification?

- **Classification** is a data analysis method that builds a **model** using past data.
- This model is called a **classifier** and it predicts the **class label** of new data.
- The class label is **categorical** (like “Yes/No”, “Safe/Risky”).

Example:
A bank uses a classifier to predict whether a **loan application** is *safe* or *risky* based on past data.

🔹 Classification vs Regression

Type	Used For	Example
Classification	Predicts class labels (categorical)	Safe/Risky, Yes/No
Regression	Predicts numeric values (continuous)	Predicting amount spent, temperature, etc.

🔹 Steps in Classification

Classification has **two main steps**:

1. **Learning/Training Step:**
    - The algorithm learns from past data (training data) that already has class labels.
    - It builds a **model**.
2. **Classification Step:**
    - The model is used to classify new data (testing data).

- This is known as the **testing phase**.

Since class labels are already known in training data, classification is a **supervised learning** method.

🔹 **Example Rules**

IF age = youth THEN loan_decision = risky

IF income = high THEN loan_decision = safe

IF age = middle_aged AND income = low THEN loan_decision = risky

🟩 2. Decision Tree Induction

🔹 **What is a Decision Tree?**

- A **Decision Tree** is a flowchart-like structure used to make decisions.
- It is learned from class-labeled training data.

Parts of a Decision Tree:

- **Inner node:** Attribute
- **Branch:** Test or condition
- **Leaf node:** Class label (final decision)

Example:
At *AllElectronics*, a decision tree can show whether a customer will buy a computer based on age, student status, and credit rating.

🔹 **History**

- **ID3 algorithm** by *J. Ross Quinlan* (1970s–80s)
- **C4.5 algorithm** – improved version of ID3
- **CART (Classification and Regression Trees)** by *Breiman et al.* (1984)
- These use a **greedy top-down recursive divide-and-conquer** approach.

🔹 **Decision Tree Algorithm (Simplified)**

1. Start with all training examples at the root.
2. Select the best attribute to split the data (using a measure like information gain).
3. Split the data based on that attribute.
4. Repeat for each branch until all nodes belong to one class or no attribute is left.

🔹 **Attribute Selection Measures**

These measures help to decide **which attribute** to split on:

1. **Information Gain** – Used in ID3; selects attribute with highest information gain.
2. **Gain Ratio** – Used in C4.5; corrects the bias of information gain.
3. **Gini Index** – Used in CART; measures impurity (lower = better).

🔹 **Tree Pruning**

To remove unnecessary or unreliable branches.

- **Pre-Pruning:** Stop growing tree early if split gives little improvement.
- **Post-Pruning:** Grow full tree first, then remove weak subtrees and replace them with leaf nodes.

Cost Complexity Pruning (CART):

- Compares cost of keeping vs pruning a subtree.
- If pruning reduces total cost, prune it.

🟩 3. Bayes Classification

🔹 **What is Bayesian Classification?**

- It is based on **Bayes Theorem**, which uses **probabilities** to classify data.
- It predicts how likely a record belongs to a certain class.

🔹 **Bayes' Theorem**

$$P(H \mid X) = \frac{P(X \mid H) * P(H)}{P(X)}$$

Where:

- $P(H \mid X)$: Posterior probability (after seeing evidence)
- $P(H)$: Prior probability (before seeing evidence)
- $P(X \mid H)$: Likelihood
- $P(X)$: Evidence probability

🔹 **Example**

To find probability of **Rain given Cloudy**,

$$P(Rain \mid Cloudy) = 1.0$$

→ Means if it's cloudy, rain chance is ~100%.

🔹 **Naive Bayes Classifier**

- A simple version of Bayes theorem.
- Assumes **attributes are independent** of each other.
- Works well and is easy to compute.

🔹 **Steps**

1. Take training data with known classes.
2. For a new tuple X, compute $P(C_i \mid X)$for each class.
3. Choose the class with the highest probability.

🔹 **Example**

For weather data:
x' = (Sunny, Cool, High, Strong)
P(Yes|x')=0.0053,P(No|x')=0.0206

→ Therefore, class label = **No**.

🟩 4. Rule-Based Classification

🔹 Concept

- Classification is done using **IF–THEN rules**.
- Easy to read and interpret.

Example:
IF age = youth AND student = yes THEN buys_computer = yes

🔹 Parts of a Rule

- **IF part** – Antecedent (condition)
- **THEN part** – Consequent (class label)

If a tuple satisfies all conditions, the rule "covers" that tuple.

🔹 Coverage and Accuracy

For rule R: (Outlook = Sunny) → PlayTennis = Yes

- **Coverage** = 4/10 = 40%
- **Accuracy** = 2/4 = 50%

🔹 Resolving Conflicts (Resolution Strategies)

If more than one rule applies or none applies:

1. **Size Ordering:**
    - Rule with **more conditions** has higher priority.
2. **Rule Ordering:**

- **Class-Based Ordering:** Rules grouped by class importance.
- **Rule-Based Ordering:** All rules in one priority list (based on accuracy or coverage).

🔹 **Default Rule**

If no rule applies, a **default rule** assigns the most common class.
Example: IF _ THEN buys_computer = No

🔹 **Rule Extraction from a Decision Tree**

- Each leaf → one rule
- Path from root to leaf → IF part
- Leaf's class label → THEN part

Rules are:

- **Mutually exclusive** (no overlap)
- **Exhaustive** (every record is covered)

🔹 **Sequential Covering Algorithm**

Rules can be learned directly from data (without decision tree).

Steps:

1. Learn one rule at a time.
2. Remove data covered by that rule.
3. Repeat until all data is covered.

Popular algorithms: AQ, CN2, RIPPER.

Learning:

- Starts with a general rule → adds conditions gradually.
- Uses **greedy search** (no backtracking).

🟩 5. Model Evaluation and Selection

🔹 **Purpose**

- To check how well a classifier performs on unseen data.
- To compare accuracy of different models.

🔹 **Confusion Matrix**

Actual / Predicted	Yes	No
Yes	TP	FN
No	FP	TN

Definitions:

- **TP (True Positive):** Correctly predicted as positive.
- **TN (True Negative):** Correctly predicted as negative.
- **FP (False Positive):** Wrongly predicted as positive.
- **FN (False Negative):** Wrongly predicted as negative.

- **Accuracy and Error Rate**

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

Error Rate=1−Accuracy

- **Class Imbalance**
  - When one class is very rare (e.g., only 3% cancer cases).
  - High accuracy can be misleading.
  - Use **precision, recall, and F-measure** instead.

- **Precision**

  Precision= TP/TP+FP

  → How many predicted positives are actually positive.

- **Recall**

  Recall=TP / TP+FN

  → How many real positives were correctly found.

- **F1-Score**

  F1= 2×Precision×Recall / Precision+Recall

  → Balances precision and recall.

- **Fβ-Measure**
  - Gives more weight to either precision or recall.
    - $\beta > 1$ → focus on recall
    - $\beta < 1$ → focus on precision

🔹 **Methods to Evaluate Accuracy**

1. **Holdout Method:**
    - Split data into training (2/3) and testing (1/3).
    - Test on the testing set.
    - Repeat several times (random sampling).
2. **Cross-Validation (k-Fold):**
    - Divide data into k parts (usually k = 10).
    - Each part becomes test set once; others are training sets.
    - Final accuracy = average of all rounds.
3. **Bootstrap Method:**
    - Sample data with replacement.
    - 63.2% used for training, 36.8% for testing (.632 bootstrap).

🔹 **Comparing Classifiers – ROC Curve**

- **ROC curve** shows True Positive Rate vs False Positive Rate.
- **Area under the ROC curve (AUC)** measures accuracy.
- Perfect model = AUC = 1.0.
- Larger area = better performance.

🔹 **Improving Classification Accuracy (Ensemble Methods)**

Idea: Combine several models to get better overall results.

🧩 **Bagging**

- Creates multiple training sets by **bootstrap sampling**.
- Trains several models → takes **majority vote**.
- Reduces error and works for both classification and regression.

⚙️ **Boosting**

- Gives **weights** to training samples.
- Next models focus more on **misclassified tuples**.

- Final prediction = weighted vote of all models.
- High accuracy but may cause **overfitting**.

🌲 **Random Forest**

- An ensemble of many **decision trees**.
- Each tree is made with **randomly selected attributes**.
- The final class = majority vote of all trees.

Two methods:

1. **Forest-RI:** Random input selection at each node (uses CART).
2. **Forest-RC:** Creates new attributes as linear combinations of existing ones.

🧠 **UNIT – 2 : DATA PREPROCESSING (Easy English Notes)**

📍 **1. Why to Preprocess Data**

📘 **Meaning:**

- **Data Preprocessing** means **cleaning and converting raw data** into a proper format before analysis.
- Real-world data is often:
    - **Incomplete** (missing values),
    - **Noisy** (contains errors/outliers),
    - **Inconsistent** (spelling or format mistakes).

📗 **Examples:**

- Missing value → Occupation = " "
- Error in data → Salary = "abcxy"
- Inconsistent → “Gujarat” & “Gujrat”

⚙️ **Purpose:**

- Follows the rule: **Garbage In → Garbage Out (GIGO)**.
  → Poor quality data = wrong results.
- Data preprocessing improves data quality for better decision-making.

✅ **Note:** Around **90% of work** in data mining is done on data cleaning and transformation.

🖌️ 2. Data Cleaning

📘 Meaning:

Cleans data by fixing missing, wrong, or duplicate information.

🔹 Tasks in Data Cleaning:

1. Fill missing values
2. Identify outliers and smooth noisy data
3. Correct inconsistent data
4. Remove redundancy

🧩 1) Fill Missing Values

Methods:

1. **Ignore the tuple** – delete record with missing data (if few).
2. **Fill manually** – not practical for large data.
3. **Use global constant** – replace with "Unknown" or -∞.
4. **Use Mean/Median** –
    - For normal data → use **Mean**
    - For skewed data → use **Median**
5. **Use Mean/Median of same class** – replace within that class.
6. **Use most probable value** – predict using **regression** or **decision tree**.

🧩 2) Identify Outliers and Smooth Noisy Data

Techniques:

1. **Binning** – group data into bins, replace with bin mean/median/boundary.
2. **Regression** – fit data to a line/function to find patterns.
3. **Clustering** – group similar data and detect outliers.

🧱 Example (Binning)

Data: 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34
→ Divide into bins (equal depth):

- Bin 1: 4, 8, 9, 15 → Mean = 9
- Bin 2: 21, 21, 24, 25 → Mean = 23
- Bin 3: 26, 28, 29, 34 → Mean = 29

Replace values with bin mean.

🧩 **3) Correct Inconsistent Data**

- Solve spelling/grammar issues (e.g., “Gujrat” → “Gujarat”).
- Use:
    - Domain knowledge
    - Standard formatting
    - Reference tables (master data)
    - Duplicate detection tools

🧩 **4) Resolve Redundancy**

- Occurs when same data is stored multiple times.
- Example: Customer info stored with every purchase record.
- Solution:
    - Use **Normalization**
    - Use **Foreign Keys**

✅ **Result:** Data becomes accurate, consistent, and non-repetitive.

🔗 **3. Data Integration**

📘 **Meaning:**

Combining data from multiple sources into one consistent dataset.

🎯 **Purpose:**

- Merge related data
- Remove duplicates
- Ensure same format and meaning

🧩 **Schema Integration**

- Match attributes from different databases.
  Example: A.cust_id ≡ B.cust#

🧩 **Entity Identification Problem**

- Identify real-world entities across sources.
  Example: cust_number = customer_id

🧩 Redundancy & Correlation Analysis

- **Redundancy** → storing same info twice
  Example: annual revenue = sum of monthly revenue
- **Correlation** → check relation between attributes
  - **Positive** → both increase (Study hours ↑, Marks ↑)
  - **Negative** → one increases, other decreases (Temp ↑, Hot coffee sales ↓)
  - **None** → unrelated (Shoe size vs IQ)

🧮 Chi-Square (χ^2) Test for Nominal Data

Used to check if two **categorical attributes** are related.

Steps:

1. Define hypothesis (H_o: no relation, H_1: relation exists)
2. Make contingency table (Observed data)
3. Calculate Expected values:

   $E = (\text{Row Total}) \times (\text{Column Total}) / \text{Grand Total}$
4. Find $\chi^2 = \Sigma (O-E)^2 / E$

Large χ^2 → strong relation between variables.

🔄 4. Data Transformation

📘 Meaning:

Converting data into another useful format for analysis.

Method	Description
Smoothing	Removes noise
Feature Construction	Create new attributes
Aggregation	Summarize data
Normalization	Scale data to smaller range
Discretization	Convert numeric to categorical

🔹 1) **Min–Max Normalization:**

$$v' = \frac{v - \text{Min}(A)}{\text{Max}(A) - \text{Min}(A)} \times (\text{NewMax} - \text{NewMin}) + \text{NewMin}$$

Example:
Min = 16, Max = 40
Value = 30
→ $\frac{30-16}{40-16} = 0.58$

🔹 **2) Decimal Scaling**

$$v' = \frac{v}{10^j}$$

where j = smallest integer so Max(|v'|) < 1

Example:
Max = 3 → divide by 10 → values between 0 and 1.

🔹 **3) Z–Score Normalization**

$$v' = \frac{v - \mu}{\sigma}$$

where μ = mean, σ = standard deviation.

Example:
μ = 54,000, σ = 16,000, v = 73,600
→ z = 1.225

🔹 **4) Discretization**

Convert continuous → discrete intervals.
Example:
Age 10–22 = Young, 23–70 = Mature, 71–100 = Senior.

🔹 **5) Concept Hierarchy**

Arrange data into levels:

Country → State → City → Street

Higher level = general, lower level = detailed.

📉 **5. Data Reduction**

📘 **Meaning:**

Reducing data size but keeping important info.

🎯 **Purpose:**

- Faster analysis
- Less storage
- Maintain data accuracy

⚙️ **Techniques**:

Type	Purpose
Dimensionality Reduction	Reduce attributes
Numerosity Reduction	Reduce records
Compression	Compact data storage

🔹 **1) Dimensionality Reduction**

Remove irrelevant attributes.

Techniques:

- **PCA (Principal Component Analysis)**
  Converts many variables → few main ones capturing most information.
  Steps: Standardize → Covariance → Eigenvalues → Sort → Transform.
- **Attribute Subset Selection**
  Keep only useful attributes (using forward/backward/decision tree).

🔹 **2) Numerosity Reduction**

Replace large data with smaller representation.

Techniques:

- **Histograms** – group values into bins.
- **Clustering** – group similar data, use cluster centers.
- **Sampling** – select part of data representing full dataset.

Types of Sampling:

Type	Description
SRSWOR	Random selection, no repeats
SRSWR	Random selection, repeats allowed
Cluster Sampling	Select whole clusters randomly

Type	Description
Stratified Sampling	Sample from each group (strata)

🔹 **3) Data Compression**

Store same data in smaller space.

Type	Description	Example
Lossless	100% recoverable	ZIP, PNG
Lossy	Some info lost	MP3, JPEG

⚙️ **6. Data Discretization**

📘 **Meaning:**

Convert continuous numeric data → discrete (category) data.
E.g., Age = 23 → "Mature".

⚙️ **Techniques:**

Method	Description
Binning	Divide into intervals, replace by mean/median
Histogram Analysis	Use data distribution for bins
Clustering	Group similar values
Decision Tree	Auto-split intervals
Correlation	Use related attributes to make bins

✅ Used to make data simpler and more understandable.

🌳 **7. Concept Hierarchy (Detailed)**

📘 **Meaning:**

Organize data in **multiple levels** — from detailed to summarized.

Example:

Country → State → City → Street

Level	Example	Distinct Values
1	Country	15
2	State	365

Level	Example	Distinct Values
3	City	3,567
4	Street	6,74,339

⚙️ **Operations:**

- **Roll-up:** Move up (City → State → Country)
- **Drill-down:** Move down (Country → State → City)

✅ Helps in summarizing, comparing, and analyzing data efficiently.

🧩 UNIT 5 – CLUSTERING (Easy English Notes)

🔹 1. What is Clustering?

- **Clustering** means **grouping similar data items** together.
- Each group is called a **cluster**.
- **Objects in the same cluster** are similar;
  Objects in different clusters are different.
- Example: grouping customers with similar buying habits.

🔸 Difference between Classification and Clustering

Classification	Clustering
Supervised Learning (uses labeled data)	**Unsupervised Learning** (no labels used)
Example: Train data like “apple” or “banana”	Example: Group unknown fruits based on shape/color
Learns from examples	Learns by finding patterns

🔹 2. Applications of Clustering

- **Marketing:** Group customers with similar behavior.
- **Biology:** Classify plants/animals based on features.
- **Insurance:** Identify groups with high claim amounts.
- **City Planning:** Group houses by area or type.
- **Libraries:** Group books by topics.
- **Earthquake Studies:** Find risky zones.

🔹 3. Features of a Good Clustering Algorithm

A good clustering method:

- Has **high similarity within** a cluster (intra-class similarity).
- Has **low similarity between** clusters (inter-class similarity).

🔹 **4. Requirements for Cluster Analysis**

1. **Scalability:** Should handle large data (not just samples).
2. **Different Data Types:** Should work with numeric, binary, or categorical data.
3. **Arbitrary Shape:** Should find clusters of any shape (not only round).
4. **Domain Knowledge:** Should not depend too much on user input (like number of clusters).
5. **Deal with Noise:** Should handle missing or wrong data.
6. **Incremental Updates:** Should work when new data arrives.
7. **High Dimensionality:** Should work with many attributes.
8. **Constraint-Based Clustering:** Should allow real-world limits (like roads or rivers for ATM placement).
9. **Interpretability:** Results should be easy to understand.

🔹 **5. Basic Types of Clustering Methods**

1. **Partitioning Methods** (like K-Means, K-Medoids)
2. **Hierarchical Methods** (like AGNES, DIANA)
3. **Density-Based Methods** (like DBSCAN, OPTICS)
4. **Grid-Based Methods** (fast, based on dividing data into cells)

🔸 **(A) Partitioning Methods**

- Divide data into **k groups (clusters)**.
- Each object belongs to **exactly one** cluster.
- Works best for **small to medium datasets**.

➤ **K-Means Algorithm (Centroid-based)**

Steps:

1. Choose **k** (number of clusters).
2. Pick **k random points** as initial centers (centroids).
3. Assign each data point to the **nearest centroid**.

4. Recalculate the centroids of new clusters.
5. Repeat steps 3–4 until centroids don't change.

Example:
If k = 2, first centers are (1,1) and (5,7).
After several updates, when no change occurs → stop.

Note:

- Fast and simple.
- **Weakness:** Affected by **outliers** (very different data points).

➤ **K-Medoids Algorithm (Representative Object-based)**

- Similar to K-Means, but uses **medoid** instead of mean.
- **Medoid = the most central object** in the cluster.
- More **robust** (not easily affected by outliers).

Steps:

1. Select **k random medoids**.
2. Assign each point to the nearest medoid.
3. Try swapping medoid with a non-medoid to reduce cost.
4. Keep the change if total cost decreases.
5. Repeat until no improvement.

🔸 **(B) Hierarchical Methods**

- Create a **tree-like structure (dendrogram)** of clusters.
- Can be **Agglomerative (Bottom-Up)** or **Divisive (Top-Down)**.

➤ **Agglomerative (AGNES):**

- Start with each object as a separate cluster.
- Merge closest clusters step by step.
- Continue until only one cluster remains.

➤ **Divisive (DIANA):**

- Start with all objects in one cluster.
- Keep splitting until each object stands alone.

Dendrogram:
A tree diagram showing how clusters are merged or divided.
Cut the tree at any level to get desired clusters.

➤ **Distance Measures Used**

1. **Single Link:** Minimum distance between clusters.
2. **Complete Link:** Maximum distance.
3. **Average Link:** Average distance.
4. **Centroid Link:** Distance between cluster centers.

➤ **Weakness of Hierarchical Methods**

- Cannot undo previous steps (once merged/split).
- Time-consuming ($O(n^2)$).
- Not suitable for very large data.

➤ **Improved Methods**

BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies):

- Handles **large numeric data** efficiently.
- Builds a **CF tree (Clustering Feature tree)** in two phases:
    1. Build tree structure from data.
    2. Cluster leaf nodes using another method.
- **Fast and scalable**, but works only for **numeric data**.

CHAMELEON:

- Uses **graph-based hierarchical clustering**.
- Merges clusters based on:
    - **Interconnectivity** (how well points are linked).
    - **Proximity** (how close they are).
- Uses **k-nearest neighbor (KNN) graph**.
- Gives **dynamic and flexible** clusters.

🔸 **(C) Density-Based Methods**

- Clusters are formed in **dense regions** separated by **low-density areas**.
- Can find **arbitrary shapes** and **handle noise**.

➤ **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)**

Parameters:

- **Eps (ε):** Radius of neighborhood.
- **MinPts:** Minimum points in a neighborhood.

Definitions:

- **Core Object:** Has at least MinPts within Eps distance.
- **Density-Reachable:** If you can reach one point from another through dense areas.
- **Noise:** Points that don't belong to any cluster.

Algorithm Steps:

1. Pick an unvisited point.
2. If it has ≥ MinPts neighbors → form new cluster.
3. Expand cluster by including nearby points.
4. If not enough neighbors → mark as noise.
5. Repeat until all points visited.

➤ **OPTICS (Ordering Points To Identify the Clustering Structure)**

- Solves DBSCAN's problem of choosing perfect ε.
- Produces an **ordering of points** instead of fixed clusters.
- Shows cluster structure visually.
- Works with **core distance** and **reachability distance**.

🔸 **(D) Grid-Based Methods**

- Divide data space into small cells (like a grid).
- Clustering happens on cells, not individual points.
- **Very fast** — depends on number of cells, not data size.
- Example algorithms: STING, CLIQUE.

🔹 **6. Outlier Detection**

🔸 **What is an Outlier?**

- A **data point very different** from the rest.
- Example: A student who always scores 90+ but suddenly gets 30.
- **Outliers are interesting** — they may show fraud, error, or special cases.
- **Noise ≠ Outlier** → Noise is random error, outlier is a meaningful exception.

Applications:

- Credit card fraud detection
- Network/telecom fraud
- Medical diagnosis
- Customer segmentation

🔸 **Types of Outliers**

1. **Global Outlier:**
    - Deviates from all data.
    - Example: One salary is 10× higher than others.
2. **Contextual Outlier:**
    - Unusual only in certain context (time/place).
    - Example: 28°C in winter (Toronto) = outlier; in summer = normal.
3. **Collective Outlier:**
    - A group of points behave abnormally together.
    - Example: Many credit cards used at same shop suddenly.

🔸 **Challenges in Outlier Detection**

- Hard to define "normal."
- Application-specific.
- Noise can hide true outliers.
- Need to explain why something is an outlier.

🔸 **Outlier Detection Methods**

Method Type	Description
Supervised	Learn using labeled examples (normal/outlier).
Unsupervised	No labels; assumes normal points form clusters; far ones are outliers.

Method Type **Description**

Semi-Supervised Some labels available; combines both approaches.

➤ **1. Statistical Methods**

- Assume normal data follows a statistical model (like Gaussian).
- Data not fitting the model → outliers.

➤ **2. Proximity-Based Methods**

- A point is an outlier if it is **far from its neighbors**.
- Distance or density used for detection.

➤ **3. Clustering-Based Methods**

- **Normal data:** in large, dense clusters.
- **Outliers:** in small or no clusters.
- Easy concept, but can be slow for large data.