Ref :Jiawei Han and Micheline Kamber

Data Preprocessing

Major Tasks in Data Preprocessing – A Structured Tutorial

1. What is Data Preprocessing?

Definition

Data preprocessing is the process of converting raw, real-world data into a clean, consistent, and usable format before applying data mining or machine learning techniques.

Why it is needed

- Real-world data are **incomplete**, **noisy**, and **inconsistent**
- Poor data quality leads to **unreliable mining results**
- High-quality decisions require **high-quality data**

Main Tasks in Data Preprocessing

1. Data Cleaning
2. Data Integration
3. Data Reduction
4. Data Transformation

---

2. Data Cleaning

Definition

Data cleaning (data cleansing) is the process of detecting and correcting incomplete, noisy, inconsistent, or erroneous data.

Objectives

- Handle missing values
- Smooth noisy data
- Identify and remove outliers
- Resolve inconsistencies

---

3. Handling Missing Values

Definition

A **missing value** occurs when no data value is stored for an attribute in a tuple.

Methods for Handling Missing Values

Method 1: Ignore the Tuple

- Drop the entire record
- Used mainly when the **class label is missing**

Limitation: Loss of potentially useful information

Method 2: Fill Missing Value Manually

- Human intervention
- Accurate but **not scalable**

Method 3: Use a Global Constant

Example constants: `"Unknown"`, `-1`

Problem: May create false patterns

Method 4: Use Mean or Median of the Attribute

Rule

- Symmetric distribution → Mean
- Skewed distribution → Median

Numerical Example
Customer incomes (₹):
40,000, 50,000, 60,000, **missing**, 70,000

$$\text{Mean} = \frac{40,000 + 50,000 + 60,000 + 70,000}{4} = 55,000$$

Filled value = ₹55,000

Method 5: Mean/Median of Same Class

Example
Credit Risk = *Low*

Incomes (Low Risk):
50,000, 55,000, 60,000

$$\text{Mean} = \frac{50,000 + 55,000 + 60,000}{3} = 55,000$$

Missing value replaced by ₹55,000

Method 6: Most Probable Value (Model-Based)

Techniques:

- Regression
- Decision Trees
- Bayesian Inference

Example (Regression)

$$\text{Income} = 20,000 + 1,500 \times \text{Age}$$

If Age = 30:

$$\text{Income} = 20,000 + (1,500 \times 30) = 65,000$$

Best method because it preserves attribute relationships.

---

4. Noisy Data

Definition

Noise is random error or variance in measured data.

---

5. Data Smoothing Techniques

5.1 Binning

Definition
Binning smooths data by grouping nearby values and replacing them with representative values.

Given Data (Sorted)

$$4,\ 8,\ 15,\ 21,\ 21,\ 24,\ 25,\ 28,\ 34$$

Equal-frequency bins (size = 3)

Bin	Values
Bin 1	4, 8, 15
Bin 2	21, 21, 24
Bin 3	25, 28, 34

(a) Smoothing by Bin Mean

$$\text{Bin 1 mean} = \frac{4 + 8 + 15}{3} = 9$$

Result

- Bin 1 → 9, 9, 9
- Bin 2 → 22, 22, 22
- Bin 3 → 29, 29, 29

(b) Smoothing by Bin Median

- Bin 1 median = 8
- Bin 2 median = 21
- Bin 3 median = 28

(c) Smoothing by Bin Boundaries

- Bin 1 boundaries: 4 and 15
  → 4, 4, 15

---

6. Regression-Based Smoothing

Definition

Regression fits data to a function to smooth values.

Linear Regression Formula

$$y = a + bx$$

Example

Predict salary from experience:

$$\text{Salary} = 25,000 + 3,000 \times \text{Experience}$$

For 5 years:

$$= 25,000 + (3,000 \times 5) = 40,000$$

---

7. Outlier Analysis

Definition

Outliers are values that significantly deviate from the majority.

Detection Techniques

- Clustering
- Statistical distance
- Visualization

Example

If most salaries are between ₹30k–₹70k, a value of ₹5,00,000 is an outlier.

---

8. Data Cleaning as a Process

Two-Step Process

1. **Discrepancy Detection**
2. **Data Transformation**

---

9. Discrepancy Detection

Causes

- Human data entry errors
- Inconsistent formats
- Data decay
- Integration conflicts

Role of Metadata

Metadata = data about data

Includes:

- Attribute type
- Domain
- Allowed values

Statistical Detection

Example (Outlier Rule)
If:

$$\mu = 50, \quad \sigma = 5$$

Values beyond:

$$\mu \pm 2\sigma = 50 \pm 10 = [40, 60]$$

Any value outside this range → **potential outlier**

---

10. Rules Used in Data Cleaning

(a) Unique Rule

- Every value must be unique
  Example: Customer ID

(b) Consecutive Rule

- No missing values in a sequence
  Example: Cheque numbers

(c) Null Rule

- Defines how missing values are represented
  Example: `0`, blank, `?`

---

11. Data Cleaning Tools

11.1 Data Scrubbing Tools

- Spell checking
- Address validation
- Fuzzy matching

11.2 Data Auditing Tools

- Detect rule violations
- Use statistics and clustering

---

12. Data Transformation

Definition

Applying functions to correct inconsistencies and standardize data.

Examples

- `"gender"` → `"sex"`
- `"25/12/2010"` → `"2010-12-25"`

ETL Tools

- Extract
- Transform
- Load

Often GUI-based but limited → scripts may be required.

---

13. Interactive Data Cleaning

Potter's Wheel

- Spreadsheet-style interface
- Immediate feedback
- Undo/redo support
- Automatic discrepancy detection

---

14. Declarative Data Cleaning

- SQL-like languages
- Specify *what* to clean, not *how*
- Efficient and expressive

---

15. Data Integration

Definition

Combining data from multiple sources into a unified view.

Problems

- Attribute name conflicts
  (e.g., `cust_id` vs `customer_id`)
- Redundant data
- Value inconsistencies

---

16. Data Reduction

Definition

Reducing data volume while preserving analytical results.

16.1 Dimensionality Reduction

Techniques:

- Attribute subset selection
- PCA
- Data compression

Example
Original attributes = 126
Selected attributes = 115
→ Faster processing

16.2 Numerosity Reduction

Techniques:

- Regression
- Histograms
- Clustering
- Sampling

---

17. Data Transformation

17.1 Normalization

Definition
Scaling data to a fixed range (e.g., [0,1])

Min–Max Normalization Formula

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Example
Salary = 50
Min = 20, Max = 100

$$x' = \frac{50 - 20}{100 - 20} = \frac{30}{80} = 0.375$$

17.2 Discretization

Example
Age:

- 0–18 → Youth
- 19–60 → Adult
- 60+ → Senior

17.3 Concept Hierarchy Generation

Price:

- ₹ → inexpensive
- ₹₹ → moderate

- ₹₹₹ → expensive

Python Code for the above concepts

1. Import Required Libraries

```python
```

```python
# Core libraries for data handling and numerical computation
import pandas as pd
import numpy as np

# Libraries for visualization and modeling
from sklearn.preprocessing import MinMaxScaler
from sklearn.linear_model import LinearRegression
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
```

---

2. Create a Sample Raw Dataset (Dirty Data)

```python
# Creating a dataset similar to a real-world customer database
data = {
    "Customer_ID": [101, 102, 103, 104, 105],
    "Age": [25, 30, np.nan, 45, 120],      # 120 is a noisy/outlier value
    "Income": [40000, np.nan, 60000, 80000, 50000],
    "Credit_Risk": ["Low", "Low", "High", "Low", "High"]
}

df = pd.DataFrame(data)

print("Original Raw Data:")
print(df)
```

---

3. Handling Missing Values (Data Cleaning)

3.1 Fill Missing Values Using Mean / Median

```python
# Fill missing Age using median (robust to outliers)
age_median = df["Age"].median()
df["Age"].fillna(age_median, inplace=True)

# Fill missing Income using mean
```

```
income_mean = df["Income"].mean()
df["Income"].fillna(income_mean, inplace=True)

print("\nAfter Handling Missing Values:")
print(df)
```

Explanation

- Median is preferred for **Age** because of extreme value (120)
- Mean is acceptable for **Income**

---

4. Class-Based Missing Value Imputation

```
python

# Fill missing income using mean income of the same credit risk group
df["Income"] = df.groupby("Credit_Risk")["Income"].transform(
    lambda x: x.fillna(x.mean())
)
```

---

5. Noise Detection Using Statistical Rules (Outliers)

5.1 Detect Outliers Using Standard Deviation

```
python

# Calculate mean and standard deviation
age_mean = df["Age"].mean()
age_std = df["Age"].std()

# Flag outliers (more than 2 standard deviations away)
df["Age_Outlier"] = abs(df["Age"] - age_mean) > (2 * age_std)

print("\nOutlier Detection:")
print(df[["Age", "Age_Outlier"]])
```

---

6. Data Smoothing Using Binning

```python
# Example numeric data (price)
price = np.array([4, 8, 15, 21, 21, 24, 25, 28, 34])

# Convert to pandas Series for convenience
price_series = pd.Series(price)

# Equal-frequency binning into 3 bins
bins = pd.qcut(price_series, q=3)

# Smoothing by bin mean
bin_means = price_series.groupby(bins).transform("mean")

print("\nOriginal Prices:")
print(price_series.values)

print("\nSmoothed Prices (Bin Means):")
print(bin_means.values)
```

---

7. Regression for Data Smoothing & Prediction

```python
# Simple regression: Predict Income from Age
X = df[["Age"]]       # Independent variable
y = df["Income"]      # Dependent variable

model = LinearRegression()
model.fit(X, y)

# Predict income for a new age
predicted_income = model.predict([[35]])

print("\nPredicted Income for Age 35:")
print(predicted_income[0])
```

---

8. Outlier Detection Using Clustering

```python
```

```python
# Clustering customers based on Age and Income
kmeans = KMeans(n_clusters=2, random_state=42)
df["Cluster"] = kmeans.fit_predict(df[["Age", "Income"]])

print("\nCluster Assignment:")
print(df[["Age", "Income", "Cluster"]])
```

Explanation

- Points far from clusters can be considered **outliers**

---

9. Data Integration Example

```python
# Second dataset with naming inconsistency
df2 = pd.DataFrame({
    "cust_id": [101, 102, 103],
    "Annual_Salary": [42000, 52000, 61000]
})

# Rename attribute to resolve inconsistency
df2.rename(columns={"cust_id": "Customer_ID"}, inplace=True)

# Merge datasets
integrated_df = pd.merge(df, df2, on="Customer_ID", how="left")

print("\nIntegrated Data:")
print(integrated_df)
```

---

10. Data Reduction

10.1 Dimensionality Reduction (PCA)

```python
# Select numeric attributes
numeric_data = df[["Age", "Income"]]
```

```python
# Apply PCA
pca = PCA(n_components=1)
reduced_data = pca.fit_transform(numeric_data)

print("\nDimensionality Reduced Data (PCA):")
print(reduced_data)
```

10.2 Numerosity Reduction (Sampling)

```python
# Random sampling (60% of data)
sampled_df = df.sample(frac=0.6, random_state=42)

print("\nSampled Data:")
print(sampled_df)
```

---

11. Data Transformation

11.1 Normalization (Min-Max Scaling)

```python
scaler = MinMaxScaler()

df[["Age", "Income"]] = scaler.fit_transform(df[["Age", "Income"]])

print("\nNormalized Data:")
print(df[["Age", "Income"]])
```

11.2 Discretization

```python
# Discretize Age into categories
df["Age_Group"] = pd.cut(
    df["Age"],
```

```
    bins=[0, 0.3, 0.7, 1.0],
    labels=["Youth", "Adult", "Senior"]
)

print("\nDiscretized Age Groups:")
print(df[["Age", "Age_Group"]])
```

11.3 Concept Hierarchy Generation

python

```
# Concept hierarchy for income
def income_category(x):
    if x < 0.3:
        return "Low"
    elif x < 0.7:
        return "Medium"
    else:
        return "High"

df["Income_Level"] = df["Income"].apply(income_category)

print("\nConcept Hierarchy (Income Level):")
print(df[["Income", "Income_Level"]])
```

---

12. Discrepancy Detection (Rule-Based Checks)

python

```
# Unique rule check for Customer_ID
print("\nAre Customer_IDs unique?")
print(df["Customer_ID"].is_unique)

# Null rule check
print("\nMissing values per column:")
print(df.isnull().sum())
```

---

13. Final Cleaned Dataset

```python
print("\nFinal Preprocessed Dataset:")
print(df)
```

DATA INTEGRATION AND DATA REDUCTION

1. Data Integration

Definition

Data integration is the process of merging data from multiple heterogeneous sources (databases, files, data cubes) into a single, consistent data store.

Purpose

- Reduce redundancy
- Resolve inconsistencies
- Improve accuracy and efficiency of data mining

---

2. Challenges in Data Integration

1. **Semantic heterogeneity**
   - Same concept, different meanings
2. **Schema heterogeneity**
   - Different attribute names and structures
3. **Redundancy**
4. **Tuple duplication**

5. **Data value conflicts**

---

3. Entity Identification Problem (Section 3.3.1)

Definition

The **entity identification problem** refers to identifying whether different records or attributes from multiple data sources refer to the **same real-world entity**.

Example: Attribute Matching

Database A	Database B
customer_id	cust_number
pay_type = H/S	pay_type = 1/2

Solution:
Use **metadata**:

- Attribute name
- Meaning
- Data type
- Domain
- Null rules

Functional Dependency Issue (Important)

Example

System 1	System 2
Discount applied on order	Discount applied on each item

If integrated blindly → **incorrect discount calculation**

---

4. Redundancy and Correlation Analysis (Section 3.3.2)

Definition

An attribute is **redundant** if it can be derived from another attribute or set of attributes.

Example:

$$\text{Annual Revenue} = \text{Monthly Revenue} \times 12$$

---

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

Purpose

To test whether **two nominal attributes are independent or correlated**

Contingency Table Structure

B \ A	a_1	a_2	...	a_c
b_1	o_{11}	o_{12}	...	o_{1c}
b_2	o_{21}	o_{22}	...	o_{2c}
...				
b_r	o_{r1}	o_{r2}	...	o_{rc}

Formula 1: Expected Frequency

$$e_{ij} = \frac{\text{count}(A = a_i) \times \text{count}(B = b_j)}{n}$$

Formula 2: Chi-Square Statistic

$$\chi^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

Degrees of Freedom

$$(r-1)(c-1)$$

Example 3.1: Gender vs Preferred Reading

Observed and Expected Frequencies

Gender / Reading	Fiction	Non-Fiction	Total
Male	250 (90)	50 (210)	300
Female	200 (360)	1000 (840)	1200
Total	450	1050	1500

Step-by-Step χ^2 Calculation

$$\chi^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840}$$

$$= 284.44 + 121.90 + 71.11 + 30.48$$

$$\chi^2 = 507.93$$

Decision Rule

- Degrees of freedom = (2−1)(2−1) = 1
- Critical χ^2 at 0.001 level = **10.828**

Since:

$$507.93 > 10.828$$

Conclusion:

Gender and preferred reading are **strongly correlated**.

---

6. Correlation Coefficient for Numeric Data

Definition

Measures **strength and direction of linear relationship** between two numeric attributes.

Formula (Pearson's Correlation Coefficient)

$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i - \bar{A})(b_i - \bar{B})}{n\sigma_A\sigma_B}$$

Alternate form:

$$r_{A,B} = \frac{\sum a_i b_i - n\bar{A}\bar{B}}{n\sigma_A\sigma_B}$$

Interpretation

r Value	Meaning
+1	Perfect positive correlation
0	No correlation
−1	Perfect negative correlation

Important Note

Correlation ≠ Causation

Example:
Hospitals ↔ Car thefts
Both depend on **population**

---

7. Covariance of Numeric Data

Definition

Measures how two variables **change together**

Mean (Expected Value)

$$E(A) = \bar{A} = \frac{\sum a_i}{n}, \quad E(B) = \bar{B} = \frac{\sum b_i}{n}$$

Covariance Formula

$$Cov(A, B) = \frac{\sum (a_i - \bar{A})(b_i - \bar{B})}{n}$$

Alternate:

$$Cov(A, B) = E(A \cdot B) - \bar{A}\bar{B}$$

Relationship Between Correlation and Covariance

$$r_{A,B} = \frac{Cov(A, B)}{\sigma_A \sigma_B}$$

Example 3.2: Stock Price Covariance

Given Data

Time	AllElectronics	HighTech
t_1	6	20
t_2	5	10
t_3	4	14
t_4	3	5
t_5	2	5

Step 1: Compute Means

$$\bar{A} = \frac{6+5+4+3+2}{5} = 4$$

$$\bar{B} = \frac{20+10+14+5+5}{5} = 10.8$$

Step 2: Compute Covariance

$$Cov(A,B) = \frac{(6\times20 + 5\times10 + 4\times14 + 3\times5 + 2\times5)}{5} - 4\times10.8$$

$$= \frac{251}{5} - 43.2$$

$$= 50.2 - 43.2 = 7$$

Conclusion

Positive covariance → **stocks rise and fall together**

---

8. Tuple Duplication (Section 3.3.3)

Definition

Occurs when **identical or near-identical tuples** exist for the same real-world entity.

Example

Order_ID	Customer	Address
101	Rahul	Delhi
101	Rahul	New Delhi

→ **Inconsistency due to duplication**

---

9. Data Value Conflict Detection (Section 3.3.4)

Causes

- Unit mismatch (kg vs lb)
- Currency mismatch (₹ vs $)
- Different abstraction levels

Example: Unit Conflict

$$\text{Weight (lb)} = \text{Weight (kg)} \times 2.20462$$

Example: Abstraction Conflict

Database A	Database B
Branch sales	Regional sales

---

10. Data Reduction (Section 3.4)

Definition

Data reduction produces a **smaller representation** of data while preserving analytical results.

---

11. Data Reduction Strategies

Strategy	Description
Dimensionality Reduction	Reduce number of attributes
Numerosity Reduction	Reduce number of records
Data Compression	Reduce storage size

---

12. Dimensionality Reduction

Techniques

- Wavelet Transform
- PCA
- Attribute Subset Selection

---

13. Numerosity Reduction

Parametric Methods

- Regression
- Log-linear models

Non-Parametric Methods

- Histograms
- Clustering
- Sampling
- Data cube aggregation

---

14. Data Compression

Type	Description
Lossless	Exact reconstruction
Lossy	Approximate reconstruction

---

15. Wavelet Transforms (Section 3.4.2)

Definition

A **Discrete Wavelet Transform (DWT)** converts a data vector into wavelet coefficients.

Data Representation

$$X' = (x_1, x_2, \ldots, x_n)$$

Steps in DWT Algorithm

1. Length must be power of 2
2. Apply smoothing and difference functions
3. Process pairs (x_{2i}, x_{2i+1})
4. Recursively repeat
5. Select wavelet coefficients

Advantages

- Better lossy compression than DFT
- Preserves local details
- Effective for noise removal

Applications

- Image compression
- Time-series analysis
- Data cleaning
- Computer vision

Summary

- Data integration ensures **semantic and structural consistency**
- Correlation and covariance help detect **redundancy**
- Data reduction improves **efficiency without sacrificing accuracy**
- Wavelet transforms provide **powerful lossy compression**

PYTHON IMPLEMENTATION: DATA INTEGRATION & DATA REDUCTION

1. Import Required Libraries

```python
import pandas as pd
import numpy as np

from scipy.stats import chi2_contingency
from sklearn.preprocessing import MinMaxScaler
from sklearn.decomposition import PCA
from sklearn.linear_model import LinearRegression
from sklearn.cluster import KMeans
```

2. Create Two Heterogeneous Data Sources

(Schema heterogeneity + semantic heterogeneity)

```python
# Database 1: Customer master table
db1 = pd.DataFrame({
    "customer_id": [101, 102, 103, 104],
    "Gender": ["Male", "Female", "Male", "Female"],
    "Reading": ["Fiction", "Non-Fiction", "Fiction", "Non-Fiction"],
    "Age": [25, 35, 45, 30]
})

# Database 2: Sales table with different attribute names
db2 = pd.DataFrame({
    "cust_number": [101, 102, 103, 105],
    "Annual_Revenue": [480000, 600000, 720000, 500000],
    "Monthly_Revenue": [40000, 50000, 60000, 42000]
})

print("Database 1:")
print(db1)

print("\nDatabase 2:")
print(db2)
```

---

3. Entity Identification & Schema Matching

```python
# Resolve naming inconsistency using metadata
db2.rename(columns={"cust_number": "customer_id"}, inplace=True)

# Integrate the databases
integrated_df = pd.merge(db1, db2, on="customer_id", how="left")

print("\nIntegrated Dataset:")
print(integrated_df)
```

What this does

- Solves the **entity identification problem**
- Aligns schema using **metadata**
- Performs **data integration**

---

4. Redundancy Detection using Correlation (Numeric Data)

Pearson Correlation Coefficient

```python

# Compute correlation between Monthly and Annual revenue
correlation = integrated_df["Monthly_Revenue"].corr(integrated_df["Annual_Revenue"])

print("\nCorrelation between Monthly and Annual Revenue:")
print(correlation)
```

Interpretation

- If correlation ≈ 1 → Annual_Revenue is **redundant**
- Can safely remove one attribute

---

5. Chi-Square Test for Nominal Data

(Gender vs Reading Preference)

Contingency Table

```python

# Create contingency table
contingency_table = pd.crosstab(
    integrated_df["Gender"],
    integrated_df["Reading"]
)

print("\nContingency Table:")
print(contingency_table)
```

Chi-Square Calculation

```python
chi2, p_value, dof, expected = chi2_contingency(contingency_table)

print("\nChi-square value:", chi2)
print("Degrees of freedom:", dof)
print("P-value:", p_value)

print("\nExpected Frequencies:")
print(pd.DataFrame(expected,
                   index=contingency_table.index,
                   columns=contingency_table.columns))
```

Decision Rule

- If `p_value < 0.05` → Attributes are **correlated**

---

6. Covariance Analysis (Numeric Data)

```python
# Covariance between Age and Monthly Revenue
covariance = np.cov(
    integrated_df["Age"].dropna(),
    integrated_df["Monthly_Revenue"].dropna()
)[0][1]

print("\nCovariance between Age and Monthly Revenue:")
print(covariance)
```

Interpretation

- Positive → increase together
- Negative → move opposite
- Zero → no linear dependency

---

7. Tuple Duplication Detection

```python
```

```
# Check duplicate tuples based on customer_id
duplicates = integrated_df.duplicated(subset=["customer_id"])

print("\nDuplicate Tuples Detected:")
print(integrated_df[duplicates])
```

---

8. Data Value Conflict Resolution

(Unit & abstraction conflicts)

Example: Weight stored in kg and lb

python

```
# Weight stored in two systems
integrated_df["Weight_kg"] = [70, 60, 80, 55]
integrated_df["Weight_lb"] = integrated_df["Weight_kg"] * 2.20462

print("\nResolved Weight Conflict:")
print(integrated_df[["Weight_kg", "Weight_lb"]])
```

---

9. DATA REDUCTION

9.1 Dimensionality Reduction – PCA

python

```
# Select numeric attributes
numeric_data = integrated_df[["Age", "Monthly_Revenue", "Annual_Revenue"]].dropna()

# Apply PCA (reduce to 2 dimensions)
pca = PCA(n_components=2)
pca_data = pca.fit_transform(numeric_data)

print("\nPCA Reduced Data:")
print(pca_data)
```

```
print("\nExplained Variance Ratio:")
print(pca.explained_variance_ratio_)
```

Meaning

- Retains maximum information
- Reduces dimensionality
- Faster mining

9.2 Numerosity Reduction – Regression Model

```
python

# Predict Annual Revenue from Monthly Revenue
X = integrated_df[["Monthly_Revenue"]].dropna()
y = integrated_df["Annual_Revenue"].dropna()

reg_model = LinearRegression()
reg_model.fit(X, y)

print("\nRegression Model:")
print("Intercept:", reg_model.intercept_)
print("Slope:", reg_model.coef_[0])
```

Model

$$Annual\_Revenue = b_0 + b_1 \times Monthly\_Revenue$$

9.3 Numerosity Reduction – Sampling

```
python

# Random sampling (50%)
sampled_data = integrated_df.sample(frac=0.5, random_state=42)

print("\nSampled Dataset:")
print(sampled_data)
```

9.4 Numerosity Reduction – Clustering

```python
# Clustering customers based on Age & Monthly Revenue
kmeans = KMeans(n_clusters=2, random_state=42)
integrated_df["Cluster"] = kmeans.fit_predict(
    integrated_df[["Age", "Monthly_Revenue"]].fillna(0)
)

print("\nClustered Data:")
print(integrated_df[["Age", "Monthly_Revenue", "Cluster"]])
```

---

10. Data Transformation

10.1 Normalization (Min–Max)

```python
scaler = MinMaxScaler()

integrated_df[["Age", "Monthly_Revenue"]] = scaler.fit_transform(
    integrated_df[["Age", "Monthly_Revenue"]]
)

print("\nNormalized Data:")
print(integrated_df[["Age", "Monthly_Revenue"]])
```

10.2 Discretization

```python
# Discretize Age into categories
integrated_df["Age_Group"] = pd.cut(
    integrated_df["Age"],
    bins=[0, 0.33, 0.66, 1.0],
    labels=["Youth", "Adult", "Senior"]
)
```

```python
print("\nDiscretized Age:")
print(integrated_df[["Age", "Age_Group"]])
```

10.3 Concept Hierarchy Generation

```python
# Income hierarchy
def income_level(x):
    if x < 0.33:
        return "Low"
    elif x < 0.66:
        return "Medium"
    else:
        return "High"

integrated_df["Income_Level"] = integrated_df["Monthly_Revenue"].apply(income_level)

print("\nIncome Concept Hierarchy:")
print(integrated_df[["Monthly_Revenue", "Income_Level"]])
```

---

11. Final Integrated & Reduced Dataset

```python
print("\nFinal Dataset After Integration, Reduction & Transformation:")
print(integrated_df)
```

KEY LEARNING

- Entity identification solved via **schema matching**
- Redundancy detected using **correlation & covariance**
- Nominal dependency tested using χ^2
- Conflicts resolved using **transformations**
- Data reduced using **PCA, regression, clustering, sampling**
- Data transformed using **normalization, discretization, hierarchies**

DIMENSIONALITY REDUCTION, DATA REDUCTION & DATA TRANSFORMATION

1. Principal Components Analysis (PCA)

1.1 Definition

Principal Components Analysis (PCA) is a statistical technique for **dimensionality reduction** that transforms an original set of **n correlated attributes** into a smaller set of **k uncorrelated (orthogonal) variables**, called **principal components**, where:

$$k \leqslant n$$

PCA is also known as:

- Karhunen–Loève (K–L) Transform

1.2 Key Idea

- PCA **does not select attributes**
- PCA **creates new attributes** as linear combinations of original ones
- These new attributes capture **maximum variance**

1.3 PCA vs Attribute Subset Selection

Aspect	PCA	Attribute Subset Selection
Method	Attribute transformation	Attribute selection
Output	New variables	Subset of original variables
Correlation	Removes correlation	May keep correlation

Aspect	PCA	Attribute Subset Selection
Interpretability	Lower	Higher

1.4 PCA Algorithm (Step-by-Step)

Step 1: Data Normalization

Each attribute must be scaled to the same range to avoid dominance.

Example dataset:

Tuple	X_1	X_2
T_1	2	200
T_2	4	400
T_3	6	600

Without normalization → X_2 dominates.

Step 2: Compute Principal Components

- Compute **orthonormal vectors**
- Each vector has:
  - Unit length
  - Perpendicular to others

These vectors form a **new coordinate system**

Step 3: Sort Components by Variance

Component	Variance Explained
PC_1	Highest
PC_2	Second highest

Component	Variance Explained
...	...

Graphically:

Original Axes	PCA Axes
X_1, X_2	Y_1, Y_2

Y_1 captures **maximum spread**

Step 4: Dimensionality Reduction

Drop components with **low variance**

1.5 PCA Numerical Example

Given Data

Object	X_1	X_2
A	2	0
B	0	2
C	3	3
D	4	4

Step 1: Mean Calculation

$$\bar{X_1} = \frac{2+0+3+4}{4} = 2.25$$

$$\bar{X_2} = \frac{0+2+3+4}{4} = 2.25$$

Step 2: Mean-Centered Data

Object	$X_1 - \mu$	$X_2 - \mu$
A	−0.25	−2.25
B	−2.25	−0.25
C	0.75	0.75
D	1.75	1.75

Step 3: Covariance Matrix

$$Cov = \begin{bmatrix} 1.583 & 1.417 \\ 1.417 & 1.583 \end{bmatrix}$$

Step 4: Eigenvectors (Principal Components)

Component	Direction	Variance
PC_1	(1,1)	High
PC_2	(1,−1)	Low

Step 5: Reduction

Keep **PC_1**, discard **PC_2**

✔️ Dimensionality reduced from **2D → 1D**

---

2. Attribute Subset Selection

2.1 Definition

Attribute subset selection removes **irrelevant or redundant attributes** while preserving the **class distribution**.

2.2 Problem Complexity

For **n attributes**:

$$\text{Total subsets} = 2^n$$

Exhaustive search → **infeasible**

2.3 Greedy (Heuristic) Methods

(a) Stepwise Forward Selection

Step	Selected Attributes
1	$\{A_1\}$
2	$\{A_1, A_4\}$
3	$\{A_1, A_4, A_6\}$

(b) Stepwise Backward Elimination

Step	Remaining Attributes
1	$\{A_1 \ldots A_6\}$
2	$\{A_1, A_3, A_4, A_5, A_6\}$
3	$\{A_1, A_4, A_6\}$

(c) Combined Approach

Add best + remove worst at each step

(d) Decision Tree Induction

- Attributes used in the tree = **selected**
- Others = **irrelevant**

2.4 Attribute Construction

Definition

New attributes created from existing ones

Example:

$$\text{Area} = \text{Height} \times \text{Width}$$

Improves:

- Accuracy
- Interpretability

---

3. Regression Models (Parametric Reduction)

3.1 Simple Linear Regression

$$y = wx + b$$

Where:

- y = response variable
- x = predictor variable
- w = slope
- b = intercept

Numerical Example

x	y
1	3
2	5
3	7

Result:

$$y = 2x + 1$$

Store only **w = 2**, **b = 1** → data reduced

3.2 Multiple Linear Regression

$$y = w_1x_1 + w_2x_2 + \cdots + b$$

3.3 Log-Linear Models

Used for:

- Discrete attributes
- Probability estimation

Key property:

$$Cov(A, B) = E(A \cdot B) - E(A)E(B)$$

---

4. Histograms

4.1 Definition

Histogram approximates data distribution using **bins**

Example Dataset (Prices)

Sorted prices:
1, 1, 5, 5, 5, 8, 10, 10, 12, 15, 18, 20, 25, 30

Singleton Histogram

Price	Frequency
5	3
10	2

Equal-Width Histogram (Width = 10)

Range	Count
1–10	8
11–20	4
21–30	2

Partitioning Rules

Method	Description
Equal-width	Fixed interval size
Equal-frequency	Equal data per bin

---

5. Clustering (Numerosity Reduction)

Definition

Groups similar objects into clusters

Cluster Quality Measures

Measure	Definition
Diameter	Max distance in cluster
Centroid distance	Avg distance to center

Reduction Principle

Replace raw data with:

- Cluster centroid
- Cluster summary

6. Sampling

Sampling Types

Type	Description
SRSWOR	Random, no replacement
SRSWR	Random, with replacement
Cluster sampling	Sample clusters
Stratified sampling	Sample per stratum

Numerical Illustration

Dataset size:

$$N = 1,000,000$$

Sample size:

$$s = 5,000$$

Cost reduction:

$$O(s) \ll O(N)$$

---

7. Data Cube Aggregation

Example: Quarterly to Annual Sales

Year	Q1	Q2	Q3	Q4	Annual
2008	224k	408k	350k	586k	1,568k

Year	Q1	Q2	Q3	Q4	Annual
2009	224k	408k	350k	586k	2,356k
2010	224k	408k	350k	586k	3,594k

Cuboids

Level	Description
Base cuboid	Lowest detail
Apex cuboid	Fully aggregated

---

8. Data Transformation Strategies

Strategy	Purpose
Smoothing	Remove noise
Construction	Add derived features
Aggregation	Summarize
Normalization	Scale values
Discretization	Convert numeric → labels
Hierarchy	Multi-level abstraction

---

9. Normalization Techniques

9.1 Min–Max Normalization

$$v_i' = \frac{v_i - \min A}{\max A - \min A}(\text{new max} - \text{new min}) + \text{new min}$$

Example

$$\frac{73,600 - 12,000}{98,000 - 12,000} = 0.716$$

9.2 Z-Score Normalization

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

Example:

$$\frac{73,600 - 54,000}{16,000} = 1.225$$

9.3 Decimal Scaling

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

Example:

$$986 \rightarrow 0.986$$

---

10. Discretization

10.1 Binning (Unsupervised)

Type	Description
Equal-width	Fixed intervals
Equal-frequency	Same count per bin

10.2 Histogram-Based Discretization

Recursive binning → **concept hierarchy**

10.3 Clustering-Based Discretization

Clusters form intervals

10.4 Decision Tree-Based (Supervised)

Split points minimize **entropy**

10.5 ChiMerge (Supervised, Bottom-Up)

- Uses χ^2 test
- Merges intervals with similar class distribution

---

11. Concept Hierarchy for Nominal Data

Method 1: Schema Ordering

$$street < city < state < country$$

Method 2: Explicit Grouping

$$\{Alberta, Manitoba\} \rightarrow Prairies$$

Method 3: Distinct-Value Heuristic

Attribute	Distinct Values
country	15
state	365
city	3567
street	674,339

Hierarchy:

country → state → city → street

Method 4: Semantic Expansion

Selecting **city** automatically pulls:

- street
- state
- country

FINAL SUMMARY

- PCA reduces dimensions via **variance maximization**
- Attribute selection removes **irrelevant features**
- Regression, histograms, clustering, sampling reduce **data volume**
- Normalization ensures **fair attribute contribution**
- Discretization simplifies numeric data
- Concept hierarchies enable **multi-level mining**

PYTHON IMPLEMENTATION

PCA → Attribute Selection → Regression → Histograms → Clustering → Sampling → Aggregation → Normalization → Discretization → Concept Hierarchies

---

1. Import Required Libraries

```python
import numpy as np
import pandas as pd

from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn.decomposition import PCA
from sklearn.feature_selection import SelectKBest, f_classif
from sklearn.linear_model import LinearRegression
from sklearn.cluster import KMeans
```

---

2. Sample Dataset (Numeric + Nominal)

```python
# Sample dataset
data = pd.DataFrame({
    "Age": [25, 30, 35, 40, 45, 50],
    "Income": [30000, 40000, 50000, 60000, 70000, 80000],
    "SpendingScore": [20, 30, 40, 50, 60, 70],
    "Purchased": [0, 0, 1, 1, 1, 1]   # class label (for supervised methods)
})

print("Original Data:")
print(data)
```

---

3. PRINCIPAL COMPONENTS ANALYSIS (PCA)

Step 1: Normalize Data (Important for PCA)

```python
# PCA requires normalization so that large-scale attributes do not dominate
scaler = StandardScaler()
scaled_data = scaler.fit_transform(data[["Age", "Income", "SpendingScore"]])
```

Step 2: Apply PCA

```python
# Reduce 3 dimensions to 2 principal components
pca = PCA(n_components=2)
principal_components = pca.fit_transform(scaled_data)

pca_df = pd.DataFrame(
    principal_components,
    columns=["PC1", "PC2"]
)

print("\nPrincipal Components:")
print(pca_df)
```

Step 3: Explained Variance

```python
print("\nExplained Variance Ratio:")
print(pca.explained_variance_ratio_)
```

Meaning

- PC1 explains the maximum variance
- PC2 explains the next highest variance

---

4. ATTRIBUTE SUBSET SELECTION (Feature Selection)

Using Statistical Significance (ANOVA / Information-Based)

```python
# Select best 2 attributes using supervised feature selection
X = data[["Age", "Income", "SpendingScore"]]
y = data["Purchased"]

selector = SelectKBest(score_func=f_classif, k=2)
X_selected = selector.fit_transform(X, y)

selected_features = X.columns[selector.get_support()]

print("\nSelected Attributes:")
print(selected_features.tolist())
```

---

5. REGRESSION (Parametric Data Reduction)

Simple Linear Regression

Model:

$$y = wx + b$$

```python
# Predict Income based on Age
X_reg = data[["Age"]]
y_reg = data["Income"]

reg = LinearRegression()
reg.fit(X_reg, y_reg)

print("\nRegression Coefficients:")
print("Slope (w):", reg.coef_[0])
print("Intercept (b):", reg.intercept_)
```

Prediction Example

```python
```

```
predicted_income = reg.predict([[37]])
print("\nPredicted Income for Age = 37:", predicted_income[0])
```

---

6. HISTOGRAMS (Numerosity Reduction)

```
python

# Histogram using equal-width bins
bins = [20000, 40000, 60000, 80000, 100000]
histogram = pd.cut(data["Income"], bins=bins)

print("\nHistogram Bins:")
print(histogram.value_counts().sort_index())
```

---

7. CLUSTERING (Data Reduction)

```
python

# Cluster customers based on Age and Income
kmeans = KMeans(n_clusters=2, random_state=42)
data["Cluster"] = kmeans.fit_predict(data[["Age", "Income"]])

print("\nClustered Data:")
print(data[["Age", "Income", "Cluster"]])
```

Meaning

- Each cluster replaces many points with a **centroid**
- Reduces data representation

---

8. SAMPLING (Numerosity Reduction)

Simple Random Sampling (Without Replacement)

```
python
```

```
sampled_data = data.sample(frac=0.5, random_state=1)

print("\nSampled Dataset:")
print(sampled_data)
```

---

9. DATA CUBE AGGREGATION

Quarterly → Annual Sales Example

python

```
sales = pd.DataFrame({
    "Year": [2008, 2008, 2008, 2008, 2009, 2009, 2009, 2009],
    "Quarter": ["Q1", "Q2", "Q3", "Q4"] * 2,
    "Sales": [224000, 408000, 350000, 586000,
              224000, 408000, 350000, 586000]
})

annual_sales = sales.groupby("Year")["Sales"].sum()

print("\nAnnual Sales (Aggregated):")
print(annual_sales)
```

---

10. DATA TRANSFORMATION – NORMALIZATION

10.1 Min–Max Normalization

$$v' = \frac{v - \min}{\max - \min}$$

python

```
minmax = MinMaxScaler()
data[["Age", "Income"]] = minmax.fit_transform(data[["Age", "Income"]])
```

```
print("\nMin–Max Normalized Data:")
print(data[["Age", "Income"]])
```

10.2 Z-Score Normalization

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

```python
zscore = StandardScaler()
zscore_values = zscore.fit_transform(data[["SpendingScore"]])

print("\nZ-Score Normalized Spending Score:")
print(zscore_values)
```

10.3 Decimal Scaling

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

```python
# Decimal scaling for Income
j = len(str(int(data["Income"].max())))
data["Income_decimal_scaled"] = data["Income"] / (10 ** j)

print("\nDecimal Scaled Income:")
print(data["Income_decimal_scaled"])
```

---

11. DISCRETIZATION

11.1 Binning (Equal-Width)

```python
```

```python
data["Age_Bin"] = pd.cut(
    data["Age"],
    bins=3,
    labels=["Young", "Middle", "Senior"]
)

print("\nDiscretized Age (Binning):")
print(data[["Age", "Age_Bin"]])
```

11.2 Histogram-Based Discretization

```python
data["Income_Hist"] = pd.qcut(
    data["Income"],
    q=3,
    labels=["Low", "Medium", "High"]
)

print("\nHistogram-Based Discretization:")
print(data[["Income", "Income_Hist"]])
```

11.3 Clustering-Based Discretization

```python
# Discretize Income using clustering
kmeans_income = KMeans(n_clusters=3, random_state=42)
data["Income_Cluster"] = kmeans_income.fit_predict(data[["Income"]])

print("\nCluster-Based Discretization:")
print(data[["Income", "Income_Cluster"]])
```

---

12. CONCEPT HIERARCHY GENERATION (NOMINAL DATA)

12.1 Schema-Based Hierarchy

```python
# Manual hierarchy definition
concept_hierarchy = {
    "street": "city",
    "city": "state",
    "state": "country"
}

print("\nConcept Hierarchy (Schema-Level):")
print(concept_hierarchy)
```

12.2 Automatic Hierarchy (Distinct Value Heuristic)

```python
# Distinct-value based hierarchy generation
distinct_counts = {
    "country": 15,
    "state": 365,
    "city": 3567,
    "street": 674339
}

hierarchy = sorted(distinct_counts.items(), key=lambda x: x[1])

print("\nAutomatically Generated Hierarchy:")
for level, (attr, count) in enumerate(hierarchy):
    print(f"Level {level}: {attr} ({count} distinct values)")
```

12.3 Semantic Expansion Example

```python
# Semantic pinning
semantic_group = ["number", "street", "city", "state", "country"]
```

```
selected_attribute = "city"

print("\nSemantic Expansion Triggered By:", selected_attribute)
print("Expanded Hierarchy:", semantic_group)
```

Solved Examples of all key concepts of data preprocessing

A. DATA CLEANING – NUMERICAL & TABULAR EXAMPLES

A1. Missing Value Imputation using Mean

Given Data (Income in ₹)

Customer	Income (₹)
C1	40,000
C2	50,000
C3	?
C4	60,000
C5	70,000

Formula (Mean)

$$\bar{x} = \frac{\sum x_i}{n}$$

Where:

- x_i = observed values
- n = number of observed values

Calculation

$$\bar{x} = \frac{40,000 + 50,000 + 60,000 + 70,000}{4} = \frac{220,000}{4} = 55,000$$

Final Table (After Imputation)

Customer	Income (₹)
C1	40,000
C2	50,000
C3	**55,000**

Customer	Income (₹)
C4	60,000
C5	70,000

A2. Missing Value Imputation using Median

Given Data (Sorted)

Customer	Age
C1	22
C2	25
C3	**?**
C4	45
C5	50

Formula (Median)

For **odd n**:

$$\text{Median} = x_{\frac{n+1}{2}}$$

Calculation

Observed values: 22, 25, 45, 50

Median = average of middle two:

$$\text{Median} = \frac{25 + 45}{2} = 35$$

Final Table

Customer	Age
C1	22
C2	25
C3	**35**
C4	45
C5	50

A3. Class-Wise Mean Imputation

Given Data

Customer	Credit Risk	Income (₹)
C1	Low	40,000
C2	Low	**?**
C3	High	60,000
C4	High	80,000

Formula

$$\bar{x}_{class} = \frac{\sum x_{class}}{n_{class}}$$

Step 1: Mean Income per Class

Low Risk

$$\bar{x}_{low} = \frac{40,000}{1} = 40,000$$

High Risk

$$\bar{x}_{high} = \frac{60,000 + 80,000}{2} = 70,000$$

Final Table

Customer	Credit Risk	Income (₹)
C1	Low	40,000
C2	Low	**40,000**
C3	High	60,000
C4	High	80,000

A4. Outlier Detection using Mean ± 2σ

Given Data (Daily Sales)

Day	Sales
D1	100
D2	105
D3	98
D4	102
D5	500

Formulae

Mean

$$\mu = \frac{\sum x_i}{n}$$

Standard Deviation

$$\sigma = \sqrt{\frac{\sum (x_i - \mu)^2}{n}}$$

Step 1: Mean

$$\mu = \frac{100 + 105 + 98 + 102 + 500}{5} = 181$$

Step 2: Standard Deviation

$$\sigma = 159.1$$

Step 3: Outlier Range

$$\mu \pm 2\sigma = 181 \pm 318.2 = [-137.2, 499.2]$$

Conclusion

Sales = **500** lies **outside normal pattern** → **Outlier**

B. DATA INTEGRATION – NUMERICAL EXAMPLES

B1. Chi-Square (χ^2) Test for Nominal Data

Given: Gender vs Reading Preference

Gender	Fiction	Non-Fiction	Total
Male	250	50	300
Female	200	1000	1200
Total	450	1050	1500

Expected Frequency Formula

$$e_{ij} = \frac{(\text{Row Total})(\text{Column Total})}{n}$$

Example Cell (Male, Fiction)

$$e_{11} = \frac{300 \times 450}{1500} = 90$$

χ^2 Formula

$$\chi^2 = \sum \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

Calculation

$$\chi^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840}$$

$$\chi^2 = 507.93$$

Conclusion

Strong correlation exists.

B2. Correlation Coefficient (Numeric Data)

Formula (Pearson r)

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{n\sigma_x\sigma_y}$$

Interpretation Table

r Value	Meaning
+1	Perfect positive
0	No correlation
−1	Perfect negative

B3. Covariance Calculation

$$Cov(A, B) = \frac{\Sigma (A - \bar{A})(B - \bar{B})}{n}$$

Positive covariance → variables rise together.

C. DATA REDUCTION – PCA (FULL NUMERICAL)

Given Data

Object	X_1	X_2
A	2	0
B	0	2
C	3	3
D	4	4

Step 1: Mean

$$\bar{X_1} = \bar{X_2} = 2.25$$

Step 2: Mean-Centered Data

Object	$X_1 - \mu$	$X_2 - \mu$
A	−0.25	−2.25
B	−2.25	−0.25
C	0.75	0.75
D	1.75	1.75

Step 3: Covariance Matrix

$$\begin{bmatrix} 1.583 & 1.417 \\ 1.417 & 1.583 \end{bmatrix}$$

Step 4: Eigenvalues

Component	Eigenvalue
PC1	High
PC2	Low

Conclusion

Drop PC2 → dimensionality reduced **2 → 1**

D. PARAMETRIC DATA REDUCTION – REGRESSION

D1. Simple Linear Regression

Formula

$$y = wx + b$$

Where:

- y = dependent variable
- x = independent variable
- w = slope
- b = intercept

Given Data

x	y
1	3
2	5
3	7

Solution

$$w = 2, \quad b = 1$$

$$y = 2x + 1$$

Prediction Example

For $x = 4$:

$$y = 2(4) + 1 = 9$$

D2. Why Regression Reduces Data

Instead of storing all points, store only:

Parameter	Value
Slope (w)	2
Intercept (b)	1

E. HISTOGRAMS (DATA REDUCTION)

E1. Histogram using Singleton Buckets

Definition

A **histogram** approximates the data distribution of a numeric attribute by grouping values into **bins (buckets)**.

Given Data (Sorted Prices in ₹)

Observation No.	Price
1	5
2	5
3	5
4	10
5	10
6	15
7	15
8	20
9	20
10	25

Singleton Histogram Table

Price (₹)	Frequency
5	3
10	2
15	2
20	2
25	1

Interpretation

- Each price value has its **own bucket**

- Useful for detecting **high-frequency outliers**

E2. Equal-Width Histogram

Formula (Bin Width)

$$\text{Bin Width} = \frac{\max - \min}{k}$$

Where:

- k = number of bins

Step 1: Compute Bin Width

$$\text{Bin Width} = \frac{25 - 5}{4} = 5$$

Histogram Table

Price Range (₹)	Frequency
5–10	5
11–15	2
16–20	2
21–25	1

Conclusion

- Equal-width histograms simplify data
- Good for **uniform distributions**

E3. Equal-Frequency Histogram

Definition

Each bin contains approximately the **same number of values**.

Histogram Table (10 values, 2 per bin)

Bin	Price Values	Frequency
B1	5, 5	2
B2	5, 10	2
B3	10, 15	2
B4	15, 20	2
B5	20, 25	2

Conclusion

- Effective for **skewed data**
- Bins have uneven width but equal depth

F. CLUSTERING (NUMEROSITY REDUCTION)

F1. Clustering for Data Reduction

Definition

Clustering groups similar objects such that:

- Intra-cluster similarity is high
- Inter-cluster similarity is low

Given Data (2-D Points)

Object	X	Y
A	2	3

Object	X	Y
B	3	4
C	8	7
D	9	8

Step 1: Assume 2 Clusters

- Cluster 1: A, B
- Cluster 2: C, D

Step 2: Compute Cluster Centroids

Centroid Formula

$$C = (\frac{\Sigma\, x}{n}, \frac{\Sigma\, y}{n})$$

Cluster 1 Centroid

$$C_1 = (\frac{2+3}{2}, \frac{3+4}{2}) = (2.5, 3.5)$$

Cluster 2 Centroid

$$C_2 = (\frac{8+9}{2}, \frac{7+8}{2}) = (8.5, 7.5)$$

Reduced Representation

Cluster	Centroid
C1	(2.5, 3.5)
C2	(8.5, 7.5)

Conclusion

Instead of storing 4 points, store **2 centroids** → data reduced.

F2. Cluster Quality – Diameter

Formula

$$\text{Diameter} = \max\left(d(p_i, p_j)\right)$$

Cluster 1 Diameter

Distance between A and B:

$$d = \sqrt{(3-2)^2 + (4-3)^2} = \sqrt{2} = 1.41$$

G. SAMPLING (DATA REDUCTION)

G1. Simple Random Sampling Without Replacement (SRSWOR)

Definition

Each tuple has equal probability $1/N$ of being selected.

Given Dataset

ID	Value
1	A
2	B
3	C
4	D
5	E

Sample Size $s = 3$

Possible Sample

Sampled ID	Value
1	A
3	C
5	E

Conclusion

- No tuple appears more than once
- Most common sampling method

G2. Simple Random Sampling With Replacement (SRSWR)

Example Sample

Draw	ID	Value
1	2	B
2	4	D
3	2	B

Conclusion

- Same tuple may appear multiple times

G3. Stratified Sampling

Given Customer Data

Customer	Age Group
C1	Youth
C2	Youth
C3	Adult
C4	Adult
C5	Senior

Sampling Strategy

Stratum	Sample Taken
Youth	C1
Adult	C3
Senior	C5

Conclusion

Ensures **fair representation of all groups**

H. DATA TRANSFORMATION & DISCRETIZATION

H1. Min–Max Normalization

Formula

$$v' = \frac{v - \min A}{\max A - \min A}$$

Given Data

Value	Income (₹)
Min	12,000
Max	98,000
v	73,600

Calculation

$$v' = \frac{73,600 - 12,000}{98,000 - 12,000} = \frac{61,600}{86,000} = 0.716$$

H2. Z-Score Normalization

Formula

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

Given

Parameter	Value
Mean (μ)	54,000
Std. Dev. (σ)	16,000
v	73,600

Calculation

$$v' = \frac{73,600 - 54,000}{16,000} = 1.225$$

H3. Decimal Scaling Normalization

Formula

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

Given

Value	v
Max	986
j	3

Result

$$986 \rightarrow 0.986$$

H4. Discretization by Binning

Given Age Data

Age
18
22
35
47
60

Equal-Width Bins (3 bins)

Age Range	Label
0–25	Youth

Age Range	Label
26–45	Adult
46–65	Senior

Discretized Table

Age	Category
18	Youth
22	Youth
35	Adult
47	Senior
60	Senior

H5. Concept Hierarchy (Numeric)

Level	Concept
High	Expensive
Mid	Moderate
Low	Cheap

I. CONCEPT HIERARCHY GENERATION (NUMERIC & NOMINAL DATA)

This section is written **exactly in the same rigor and depth** as A–H, with **definitions, full tables, hierarchy construction logic, variables, and conclusions**.
Nothing is skipped.

I. CONCEPT HIERARCHY GENERATION

I1. Concept Hierarchy – Numeric Attribute (Price)

Definition

A **concept hierarchy** organizes numeric values into **multiple abstraction levels**, allowing data mining at different granularities (low → high).

Given Numeric Data (Price in ₹)

Transaction	Price
T1	50
T2	120
T3	180
T4	260
T5	340
T6	480
T7	620
T8	790
T9	920

Step 1: Define Base-Level Intervals (Fine Granularity)

Price Range (₹)	Symbol
(0 – 100]	P1
(100 – 200]	P2
(200 – 300]	P3
(300 – 400]	P4
(400 – 500]	P5
(500 – 600]	P6
(600 – 700]	P7
(700 – 800]	P8
(800 – 900]	P9

Price Range (₹)	Symbol
(900 – 1000]	P10

Step 2: Map Prices to Base-Level Concepts

Transaction	Price	Base Concept
T1	50	P1
T2	120	P2
T3	180	P2
T4	260	P3
T5	340	P4
T6	480	P5
T7	620	P7
T8	790	P8
T9	920	P10

Step 3: Intermediate-Level Abstraction

Intermediate Level	Covered Ranges
Low	(0 – 200]
Medium	(200 – 500]
High	(500 – 800]
Very High	(800 – 1000]

Step 4: High-Level Concept Hierarchy

High-Level Concept	Range
Cheap	(0 – 200]
Moderate	(200 – 500]
Expensive	(500 – 1000]

Final Hierarchical Mapping Table

Transaction	Price	High-Level Concept
T1	50	Cheap
T2	120	Cheap
T3	180	Cheap
T4	260	Moderate
T5	340	Moderate
T6	480	Moderate
T7	620	Expensive
T8	790	Expensive
T9	920	Expensive

I2. Concept Hierarchy – Nominal Data (Schema-Level Ordering)

Given Nominal Attributes (Location)

Attribute
Street
City
State
Country

Hierarchy Definition (Schema-Level)

Street < City < State < Country

Hierarchy Table

Level	Attribute
Highest	Country
	State
	City
Lowest	Street

Example Data Transformation

Street	City	State	Country
MG Road	Bengaluru	Karnataka	India

Higher Abstraction View

City	State	Country
Bengaluru	Karnataka	India

I3. Concept Hierarchy – Explicit Data Grouping (Manual)

Given Provinces (Canada Example)

Province
Alberta
Saskatchewan
Manitoba
British Columbia

User-Defined Grouping

Provinces	Region
Alberta, Saskatchewan, Manitoba	Prairies
British Columbia	Western Canada

Hierarchy Table

Level	Concept
Country	Canada
Region	Prairies / Western Canada
Province	Individual provinces

Conclusion

- Partial hierarchy manually specified
- Useful when **domain knowledge is strong**

I4. Concept Hierarchy – Distinct-Value Heuristic (Automatic)

Definition

Attributes with **fewer distinct values** represent **higher abstraction levels**.

Given Attributes with Distinct Counts

Attribute	Distinct Values
Country	15
State	365
City	3,567
Street	674,339

Step 1: Sort Attributes (Ascending)

Order	Attribute	Distinct Count
1	Country	15

Order	Attribute	Distinct Count
2	State	365
3	City	3,567
4	Street	674,339

Step 2: Generate Hierarchy

Level	Attribute
Highest	Country
	State
	City
Lowest	Street

Conclusion

- Automatic
- Based on **data statistics**
- Works well in practice (not foolproof)

I5. Concept Hierarchy – Semantic Attribute Expansion

Given

Administrator pins semantically linked attributes:

Semantic Group	Attributes
Location	Number, Street, City, State, Country

User Specifies Only

User Input
City

System-Generated Hierarchy

Level	Attribute
Country	
State	
City	
Street	
Number	

User Adjustment (Optional)

Retained	Removed
City, State, Country	Street, Number

Conclusion

- Ensures **completeness**
- Maintains **semantic consistency**
- User retains **full control**

I6. Summary Table – Concept Hierarchy Methods

Method	Data Type	Manual / Auto	Uses Semantics
Numeric intervals	Numeric	Manual	No

Method	Data Type	Manual / Auto	Uses Semantics
Schema ordering	Nominal	Manual	Yes
Explicit grouping	Nominal	Manual	Yes
Distinct-value heuristic	Nominal	Automatic	No
Semantic expansion	Nominal	Semi-auto	Yes

DATA REDUCTION – SOLVED TABULAR NUMERICALS

---

1. Covariance (Numeric Attributes)

Given Data

Observation	X (Sales A)	Y (Sales B)	$X - \bar{X}$	$Y - \bar{Y}$	$(X - \bar{X})(Y - \bar{Y})$
1	6	20	2	9.2	18.4
2	5	10	1	−0.8	−0.8
3	4	14	0	3.2	0
4	3	5	−1	−5.8	5.8
5	2	5	−2	−5.8	11.6
Σ					**35.0**

$$\bar{X} = 4, \quad \bar{Y} = 10.8$$

Covariance

$$Cov(X, Y) = \frac{\Sigma (X - \bar{X})(Y - \bar{Y})}{n} = \frac{35}{5} = \boxed{7}$$

---

2. Hypothesis Test – Chi-Square (χ^2)

Observed vs Expected Frequencies

Gender	Reading	Observed (O)	Expected (E)	$(O-E)^2/E$
Male	Fiction	250	90	284.44

Gender	Reading	Observed (O)	Expected (E)	$(O-E)^2/E$
Male	Non-Fiction	50	210	121.90
Female	Fiction	200	360	71.11
Female	Non-Fiction	1000	840	30.48
			χ^2	**507.93**

Decision

$$\chi^2_{calculated} = 507.93 > \chi^2_{critical}(10.828) \Rightarrow \boxed{\text{Reject } H_0}$$

---

3. Euclidean Distance

Given Points

Object	X_1	X_2
A	2	3
B	6	7

Distance Table

Step	Calculation	Value
$(x_2-x_1)^2$	$(6-2)^2$	16
$(y_2-y_1)^2$	$(7-3)^2$	16
Sum	16+16	32
$\sqrt{}$ Sum	$\sqrt{}32$	**5.66**

$$d_E(A, B) = \boxed{5.66}$$

---

4. Manhattan Distance

Object	X_1	X_2
A	2	3
B	6	7

Step		Value
	6−2	
	7−3	
Distance		**8**

$$d_M = \boxed{8}$$

---

5. Minkowski Distance (p = 3)

Step	Expression	Value
	6−2	^{3}
	7−3	^{3}
Sum	64+64	128
Cube root	$\sqrt[3]{128}$	**5.04**

$$d_{Minkowski} = \boxed{5.04}$$

---

6. Jaccard Similarity

Given Sets

Object	Items
A	{a, b, c, d}
B	{b, c, e}

Calculation Table

Measure	Set	Count
Intersection	{b, c}	2
Union	{a,b,c,d,e}	5

$$J(A, B) = \frac{2}{5} = \boxed{0.4}$$

---

7. Cosine Similarity

Given Vectors

Term	Vector A	Vector B	A×B
t1	1	2	2
t2	2	1	2
t3	3	2	6
Σ			**10**

Norm	Value
‖A‖	$\sqrt{(1^2+2^2+3^2)}=3.74$
‖B‖	$\sqrt{(2^2+1^2+2^2)}=3$

$$\cos(\theta) = \frac{10}{3.74 \times 3} = \boxed{0.892}$$

DATA TRANSFORMATION – SOLVED TABULAR NUMERICALS

---

8. Binning (Equal-Width)

Given Data

Value
4
8
15
21
24
28

Bins (Width = 10)

Bin Range	Values	Bin Mean
0–10	4,8	6
11–20	15	15
21–30	21,24,28	24.33

---

9. Encoding – One-Hot Encoding

Original Data

ID	Color
1	Red
2	Blue
3	Green

Encoded Table

ID	Red	Blue	Green
1	1	0	0

ID	Red	Blue	Green
2	0	1	0
3	0	0	1

---

10. Normalization (Min–Max)

Given

Value	Income
Min	12,000
Max	98,000
v	73,600

Normalized Table

Step	Formula	Result
v−min	73,600−12,000	61,600
max−min	98,000−12,000	86,000
v′	61,600 / 86,000	**0.716**

$$\boxed{v' = 0.716}$$

Above below topics covered

- Covariance
- Hypothesis Testing (χ^2)
- Euclidean Distance
- Manhattan Distance
- Minkowski Distance
- Jaccard Similarity

- Cosine Similarity
- Binning
- Encoding
- Normalization

All formulas

DATA REDUCTION – FORMULAS USED

1. Covariance

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i$$

$$\bar{Y} = \frac{1}{n} \sum_{i=1}^{n} Y_i$$

$$\text{Cov}(X, Y) = \frac{1}{n} \sum_{i=1}^{n} (X_i - \bar{X})(Y_i - \bar{Y})$$

---

2. Chi-Square (χ^2) Test

$$e_{ij} = \frac{(\text{Row Total}_i)(\text{Column Total}_j)}{n}$$

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

---

3. Euclidean Distance

$$d_E(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

---

4. Manhattan Distance

$$d_M(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} | x_i - y_i |$$

---

5. Minkowski Distance

$$d_p(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{n} | x_i - y_i |^p\right)^{1/p}$$

---

6. Jaccard Similarity

$$J(A, B) = \frac{| A \cap B |}{| A \cup B |}$$

---

7. Cosine Similarity

$$\cos(\theta) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

DATA TRANSFORMATION – FORMULAS USED

8. Equal-Width Binning

$$\text{Bin Width} = \frac{\max(A) - \min(A)}{k}$$

---

9. One-Hot Encoding

$$x_{ij} = \begin{cases} 1, & \text{if category } j \text{ applies to record } i \\ 0, & \text{otherwise} \end{cases}$$

---

10. Min–Max Normalization

$$v_i' = \frac{v_i - \min(A)}{\max(A) - \min(A)}$$