Question: Understanding Data

Unit	Topics Covered	Learning Outcomes (LO)	Program Outcomes (PO)	Program Speci Outcomes (PS
I	Evolution of DW VLDB Concepts OLTP vs OLAP Data Warehouse Architecture ETL Process & Metadata	LO1: Understand the basic concepts of data warehousing and its evolution. LO2: Explain architecture and backend processes. LO3: Differentiate between OLTP and OLAP.	PO1: Engineering Knowledge PO2: Problem Analysis PO3: Design/Development of Solutions	PSO1: Design data ware models for business intel PSO2: Apply ETL proce effectively.
II	Multidimensional Modeling Schema Designs: Star, Snowflake, Fact Constellation - Data Marts - OLAP Operations (Roll-up, Drill-down, Slicing, Dicing)	LO1: Model and structure data for analytical processing. LO2: Apply OLAP techniques for multidimensional analysis. LO3: Design star and snowflake schemas.	PO2: Problem Analysis PO5: Modern Tool Usage PO12: Lifelong Learning	PSO1: Use OLAP tools a perform complex data an PSO2: Design efficient warehouse schemas.
III	- Fundamentals of Data Mining - KDD vs Data Mining - Data Mining Functionalities - Issues & Challenges - Applications	LO1: Define data mining and distinguish it from KDD. LO2: Explain various functionalities and applications. LO3: Identify challenges in data mining.	PO1: Engineering Knowledge PO4: Investigations PO11: Project Management	PSO1: Analyze and mine pattern recognition. PSO3: Apply data minin real-world cases.
IV	- Association Rule Mining - Apriori, FP-Tree, Partition, Pincer, DIC Algorithms	LO1: Understand methods of discovering association rules. LO2: Apply algorithms to generate frequent itemsets.	PO2: Problem Analysis PO3: Design/Development PO5: Modern Tool Usage	PSO1: Implement data m algorithms in tools like V Python. PSO2: Develop analytica for association discovery
V	- Classification Techniques: Decision Trees - Web Mining (Content, Structure) - Text, Temporal, and Spatial Mining	LO1: Understand and implement classification techniques. LO2: Explain the concepts of Web, Text, and Spatial Mining. LO3: Analyze structured and unstructured data.	PO3: Design/Development PO4: Investigations PO10: Communication	PSO1: Use classification techniques for decision m PSO3: Implement web a mining for practical case

Unit-1

---

1. Evolution of Data Warehousing

Problems:

1. Growth Analysis

   A company's reporting time was reduced from 72 hours to 4 hours after data warehousing implementation. Calculate the percentage improvement.

2. Cost-Benefit Analysis:

   If a business spends $100,000 initially on a data warehouse and gains $25,000 quarterly in benefits, in how many months will it break even?

---

2. VLDB Concepts (Very Large Databases)

Problems:

1. Storage Calculation:

   A VLDB system stores 10 TB of data with a daily growth of 50 GB. How much data will it store in 2 years?

2. Partitioning Impact:

   A table has 800 million rows. If horizontal partitioning splits it into 8 equal parts, how many rows per partition?

---

3. OLTP vs OLAP
Problems:

1. Transaction Volume:

   An OLTP system handles 500 transactions per second. How many does it handle in a day? Compare this with an OLAP system executing 200 queries per hour.

2. Query Load Analysis:

   OLTP query time: 0.5 sec/query; OLAP query time: 5 min/query. Find the ratio of time taken for processing 100 queries in both.

---

4. Data Warehouse Architecture
Problems:

1. Storage Allocation:

   A data warehouse contains 3 layers: staging (2 TB), integration (4 TB), access (1 TB). Find the total size and % of each.

2. Network Throughput:

   If data loading into the warehouse happens at 200 MB/min, how long will it take to load a 120 GB dataset?

---

5. ETL Process & Metadata
Problems:

1. ETL Load Time:

   Extract: 30 min, Transform: 90 min, Load: 60 min. What's the total ETL time per batch? If done daily, how much time is spent in a month?

2. Metadata Volume:

   For every 10 GB of warehouse data, 0.2 GB of metadata is generated. How much metadata is needed for 500 GB of warehouse storage?

Numericals: Growth Analysis
Q: Reporting time reduced from 72 hrs to 4 hrs.. Calculate % improvement.
Solution:
Improvement=72−472×100=6872×100≈94.44
Answer: 94.44% improvement

- Cost-Benefit Analysis

Q: Initial cost = $100,000
Quarterly benefit = $25,000 → Monthly = $25,000 / 3 ≈ $8,333.33
Break-even time=100,000/8,333.33≈12 months
Answer: 12 months to break even

---

2. VLDB Concepts

- Storage Calculation

Q: Initial = 10 TB

Daily growth = 50 GB
2 years = 730 days
Growth=730×50=36,500 GB
=36.5 TB
Total=10+36.5
=46.5 TB
Answer: 46.5 TB after 2 years

🔹 Partitioning Impact
Q: 800 million rows, 8 partitions
Rows/partition=800,000,0008=100,000,000
Answer: 100 million rows per partition

---

3. OLTP vs OLAP
Transaction Volume
OLTP:
500 transactions/sec×3600×24=43,200,000 transactions/day
OLAP:
200 queries/hr×24=4,800 queries/day200
Answer:

- OLTP: 43.2 million transactions/day
- OLAP: 4,800 queries/day

Query Load Analysis
OLTP: 0.5 sec/query
OLAP: 5 min/query = 300 sec/query
100 queries:

- OLTP → 100 × 0.5 = 50 sec
- OLAP → 100 × 300 = 30,000 sec

Time ratio=30,000/50=600
Answer: OLAP takes 600× longer per 100 queries

---

4. Data Warehouse Architecture
Storage Allocation
Staging = 2 TB, Integration = 4 TB, Access = 1 TB
Total = 7 TB

- Staging: 27×100≈28.57%
- Integration: 47×100≈57.14%
- Access: 17×100≈14.29%

Answer:

- Total size: 7 TB
- Percentages: Staging 28.57%, Integration 57.14%, Access 14.29%

Network Throughput
Load speed = 200 MB/min
Dataset = 120 GB = 122,880 MB
Time=122,880/200=614.4 minutes=10hrs14min
Answer: 10 hours 14 minutes

ETL Load Time
Extract: 30 min
Transform: 90 min
Load: 60 min
Total per batch = 180 min = 3 hrs/day
Monthly: 3 × 30 = 90 hrs
Answer: 3 hrs/day; 90 hrs/month

Metadata Volume
0.2 GB metadata per 10 GB
For 500 GB:
Metadata=50010×0.2=10 GB
Answer: 10 GB metadata required

Unit-2

---

1. Multidimensional Modeling
Problem 1: Dimension Combinations
Q: A sales cube has 3 dimensions: Product (10 types), Region (5 regions), and Time (12 months). How many total cells does the cube have?
Solution:
Total cells=10×5×12=600
Answer: 600 cells

Problem 2: Sparsity Calculation
Q: Out of 600 possible cells, only 180 cells have data. What is the sparsity percentage?
Solution:
Sparsity=(600−180600)×100=70%
Answer: 70%

---

2. Schema Designs: Star, Snowflake, Fact Constellation
Problem 1: Star Schema Join Count
Q: A star schema has 1 fact table and 4 dimension tables. How many joins are required in a query that includes all dimensions?
Solution:

- Star schema requires 1 join per dimension.

Total joins=4
Answer: 4 joins

Problem 2: Snowflake Schema Join Cost
Q: In a snowflake schema, one dimension is normalized into 3 tables. If there are 4 such dimensions, how many total joins are needed?
Solution:

- Each normalized dimension has 3 tables → 2 joins per dimension
- 4 dimensions × 2 joins = 8 joins

Answer: 8 joins

Problem 3: Fact Constellation Size
Q: A fact constellation has 2 fact tables and 3 shared dimension tables. If each fact table connects to all 3 dimensions, how many total connections exist?
Solution:

- 2 fact tables × 3 dimensions = 6 connections

Answer: 6 connections

---

3. Data Marts
Problem 1: Data Mart Storage Estimate
Q: If a data mart contains 12 months of sales data, with 50 MB per month, how much storage does it need?
Solution:
Storage=12×50=600 MB
Answer: 600 MB

Problem 2: Cost per Department
Q: A company builds 5 departmental data marts costing $8,000 each. What is the total cost?
Solution:
5×8000=$40,000
Answer: $40,000

---

4. OLAP Operations (Roll-up, Drill-down, Slicing, Dicing)
Problem 1: Roll-up Hierarchy
Q: Time dimension has: Day (365), Month (12), Quarter (4), Year (1). If we roll up from Day to Quarter, how many data points remain?
Solution:

- Original: 365
- After roll-up: 4 (quarters)

Answer: 4 data points

Problem 2: Drill-Down Volume
Q: Drill down from the Year level (1 record) to the Month level (12 records). What is the multiplication factor?
Solution:
Drill-down factor=12x
Answer: 12 times increase in data points

Problem 3: Slicing a Cube
Q: A cube has 3 dimensions: Region (5), Product (10), and Time (12). If we slice the cube on Region = "North", how many cells remain?
Solution:

- Remaining dimensions: Product × Time = 10 × 12 = 120

Answer: 120 cells

Problem 4: Dicing
Q: Dice on Product (2 types) and Region (2 regions). The time remaining is 12 months. How many cells are in the sub-cube?
Solution:
2×2×12=48 cells
Answer: 48 cells

Unit-3

---

1. Fundamentals of Data Mining
Problem 1: Support Count in a Dataset
Q: In a dataset of 500 transactions, itemset {Bread, Butter} appears in 120 transactions. What is the support percentage?
Solution:
Support=(120500)×100=24%
Answer: 24%

Problem 2: Classification Accuracy
Q: A classifier correctly predicts 180 out of 200 test records. What is the accuracy?
Solution:
Accuracy=(180200)×100=90%
Answer: 90%

---

2. KDD vs Data Mining
Note: KDD = Knowledge Discovery in Databases; Data Mining is a step in KDD.
Problem 1: Time Distribution in KDD Process
Q: If the total KDD process takes 50 hours, and data mining takes 15 hours, what percentage of the process is data mining?
Solution:
Percentage=(1550)×100=30%
Answer: 30%

Problem 2: Preprocessing Effort
Q: Preprocessing consumes 60% of KDD time. How many hours are spent on preprocessing if KDD takes 40 hours?
Solution:
Time=0.6×40=24 hours
Answer: 24 hours

---

3. Data Mining Functionalities
Problem 1: Association Rule Confidence

Q: In a dataset, 100 transactions have item A, and 60 have both A and B. What is the confidence of rule A ⇒ B?
Solution:
Confidence=(60100)×100=60%
Answer: 60%

Problem 2: Clustering Accuracy
Q: A clustering algorithm groups 1000 data points. If 880 points are correctly clustered, what is the clustering accuracy?
Solution:
Accuracy=(8801000)×100=88%
Answer: 88%

---

4. Issues & Challenges in Data Mining
Problem 1: Handling Missing Data
Q: A dataset has 10,000 records, and 750 have missing values. What percentage of data has missing values?
Solution:
(75010000)×100=7.5%
Answer: 7.5%

Problem 2: Scalability Problem
Q: An algorithm takes 1 second for 1000 records. Estimate time for 1 million records assuming linear time complexity.
Solution:
Time=1,000,0001,000×1=1000 seconds
Answer: 1000 seconds

---

5. Applications of Data Mining
Problem 1: Fraud Detection Rate
Q: Out of 500 detected fraud cases, 450 are actual frauds. What is the precision?
Solution:
Precision=(450500)×100=90%
Answer: 90%

Problem 2: Market Basket Analysis
Q: In a supermarket, 20,000 transactions occur in a month. 2,000 of them include both milk and eggs. What is the support?
Solution:
Support=(200020000)×100=10%
Answer: 10%

Unit-4

---

1. Association Rule Mining
Problem 1: Support and Confidence Calculation
Q: In a market basket of 1,000 transactions:

- 300 transactions contain Milk
- 180 transactions contain both Milk and Bread

Calculate:
a) Support for {Milk, Bread}
b) Confidence for the rule: Milk ⇒ Bread
Solution:
a)Support=(1801000)×100=18%
b)Confidence=(180300)×100=60%
Answer: Support = 18%, Confidence = 60%

Problem 2: Lift Calculation
Q: From previous data:

- Support(Milk) = 30%, Support(Bread) = 40%, Support(Milk & Bread) = 18%
  Calculate **Lift** of rule Milk ⇒ Bread.

Solution:
Lift = P(Milk∧Bread)P(Milk)×P(Bread)= 0.180.3×0.4=0.180.12=1.5

Answer: Lift = 1.5 (indicates positive correlation)

---

2. Apriori Algorithm
Problem 1: Frequent Itemset Generation
Q: Given a database of 5 transactions:
T1: {A, B, C}
T2: {A, C}
T3: {A, B}
T4: {B, C}
T5: {A, B, C}
Find frequent 2-itemsets with **min support = 60%**.
Solution:

- Total transactions = 5
- Support count needed = 0.6 × 5 = 3

Itemsets:

- {A, B} → in T1, T3, T5 → 3
- {A, C} → in T1, T2, T5 → 3
- {B, C} → in T1, T4, T5 → 3

Answer: All 3 are frequent 2-itemsets.

---

3. Partition Algorithm
Problem 1: Local and Global Frequency
Q: A dataset of 20,000 transactions is partitioned into 4 equal parts.
Minimum support = 5% (i.e., 1,000 transactions).
An itemset appears 300 times in Part 1, 250 in Part 2, 300 in Part 3, and 200 in Part 4.
Is it frequent?
Solution:

- Total support = 300 + 250 + 300 + 200 = 1,050
- Required = 1,000
- Is frequent globally: ✅ Yes
- Also appeared in 3 partitions locally ⇒ Can be a candidate via Partition Algorithm

Answer: Itemset is frequent globally.

---

4. Pincer Search Algorithm
Problem 1: Top-down and Bottom-up Tracking
Q: In a transaction set of 100 records:

- Min support = 20% = 20
- {A, B, C} has support 25
- Its subsets: {A, B}, {A, C}, {B, C} all have support > 20

Can {A, B, C} be declared frequent early in a Pincer search?
Solution:

- Since both the itemset and all subsets are frequent, it will be retained in both top-down and bottom-up search.

Answer: Yes, {A, B, C} is declared frequent early by the Pincer algorithm.

---

5. DIC (Dynamic Itemset Counting)
Problem 1: Early Candidate Generation
Q: Given a dataset of 1,000 transactions, DIC scans 100 transactions per window.
An itemset becomes frequent in the 6th window.
After how many transactions will it be accepted as frequent?
Solution:
6×100=600 transactions
Answer: After 600 transactions.

Problem 2: Candidate Rejection
Q: If an itemset starts with support 2 and doesn't increase in 3 windows (of 100 transactions each), will DIC remove it if min support = 10%?
Solution:

- Required support = 0.1 × 1000 = 100

Itemset stuck at support = 2

- Yes, it will be pruned early due to slow growth.

Answer: Itemset is pruned.

Unit-5

---

1. Classification Techniques: Decision Trees
Problem 1: Entropy Calculation
Q: A dataset has 10 samples:

- 4 are class Yes,
- 6 are class No.

Calculate the entropy of the dataset.

Solution:
P(Yes)=4/10,
P(No)=6/10
Entropy=−(4/10log2(4/10)+6/10log2(6/10)≈−[0.4·(−1.32)+0.6·(−0.736)]≈0.971
Answer: ≈ 0.971

Problem 2: Information Gain
Q: The original entropy is 0.971. A split produces two subsets:

- Subset A: 4 samples (3 Yes, 1 No) → Entropy = 0.811
- Subset B: 6 samples (1 Yes, 5 No) → Entropy = 0.650

Calculate the Information Gain.
Solution:
Weighted entropy=(4/10×0.811)+(6/10×0.650)=0.324+0.390=0.714= 0.324 + 0.390 = 0.714
Information Gain=0.971−0.714=0.257
Answer: 0.257

---

2. Web Mining (Content and Structure Mining)
Problem 1: Content Mining – Keyword Frequency
Q: A web page has 800 words. The keyword “data” appears 40 times. What is the term frequency (TF) of “data”?
Solution:
TF=40800=0.05
Answer: 0.05

Problem 2: Structure Mining – PageRank Calculation (Simplified)
Q: Page A is linked by Page B and Page C.

- Page B has a rank of 0.6 and links to 2 pages
- Page C has a rank of 0.8 and links to 4 pages

Calculate the new PageRank for A using the simplified formula:
PR(A)=0.15+0.85
Solution:
PR(A)=0.15+0.85(0.62+0.84)=0.15+0.85(0.3+0.2)=0.15+0.85×0.5=0.575
Answer: 0.575

---

3. Text Mining
Problem 1: TF-IDF Calculation
Q: The Term “cloud” appears 5 times in a document with 100 total words. The term appears in 10 out of 1,000 documents.
Calculate TF-IDF.
Solution:

- TF = 5 / 100 = 0.05
- IDF = $\log_{10}(1000 / 10) = \log_{10}(100) = 2$
- TF-IDF = 0.05 × 2 = 0.1

Answer: 0.1

---

4. Temporal Mining

Problem 1: Pattern Frequency Over Time
Q: In a time-series dataset spanning 12 months, a sales spike pattern appears in Jan, Mar, Jul, and Dec.
What is the support for the pattern?
Solution:
Support=4/12=0.333 or 33.3%
Answer: 33.3%

Problem 2: Moving Average Calculation
Q: Sales data over 5 months: [100, 120, 140, 160, 180]. Find the 3-month moving average for month 4.
Solution:
3-month avg=120+140+1603=4203=140
Answer: 140

---

5. Spatial Mining
Problem 1: Distance-Based Clustering
Q: Given 3 customer locations on a grid:
A(1, 2), B(4, 6), C(5, 8). Find Euclidean distance between A and B.
Solution:
Distance=(4−1)2+(6−2)2=9+16=25=5
Answer: 5 units

Problem 2: Density Threshold
Q: A spatial region has 500 points over 10 square km.
What is the point density per km²?
Solution:
Density = 500/10 = 50 points/km²
Answer: 50 points/km²

Question: Understanding Data

Suppose that the data for analysis includes the attribute age.
The age values for the data tuples are (in increasing order)
13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.
(a) What is the mean of the data? What is the median?
(b) What is the mode of the data? Comment on the data's modality (i.e., bimodal, trimodal, etc.).
(c) What is the midrange of the data?
(d) Can you find (roughly) the first quartile (Q1) and the third quartile (Q3) of the data?
(e) Give the five-number summary of the data.
(f) Show a boxplot of the data.
(g) How is a quantile-quantile plot different from a quantile plot?

(a) Mean and Median

Mean	**≈ 29.96**
Median	**25**

(b) Mode and Modality

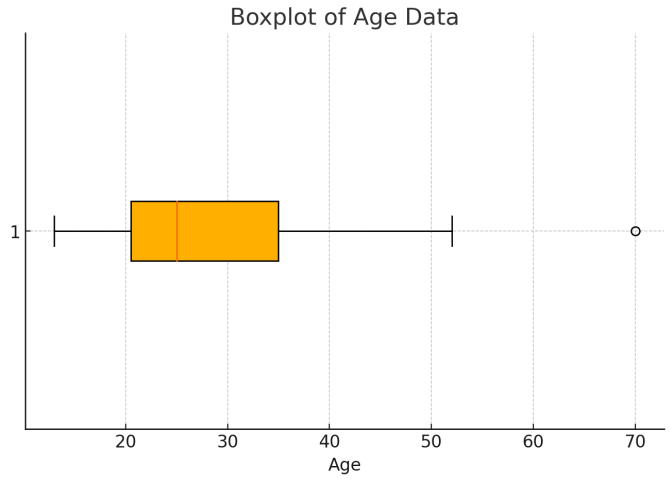
Mode: 25 (appears 4 times) The data is unimodal because it has a single mode.
(c) Midrange Midrange = (13 + 70) / 2 = 41.5
(d) First and Third Quartiles

Q1 (First Quartile)	**20.5**
Q3 (Third Quartile)	**35.0**

(e) Five-Number Summary

Minimum	**13**
Q1	**20.5**
Median	**25**
Q3	**35.0**
Maximum	**70**

(f) Boxplot:



(g) Difference between a Quantile Plot and a Quantile-Quantile (Q-Q) Plot:

- A quantile plot shows the quantiles of a dataset to help visualize its distribution.

- A Q-Q plot compares the quantiles of the dataset to a theoretical distribution (e.g., normal distribution) to check if the data follows that distribution.

Question: Understanding Data

A hospital conducted a study on 18 randomly selected adults, recording their age and body fat percentage. The data collected is as follows:
Age: 23, 23, 27, 27, 39, 41, 47, 49, 50, 52, 54, 54, 56, 57, 58, 58, 60, 61
Body Fat Percentage: 9.5, 26.5, 7.8, 17.8, 31.4, 25.9, 27.4, 27.2, 31.2, 34.6, 42.5, 28.8, 33.4, 30.2, 34.1, 32.9, 41.2, 35.7
Based on this data, perform the following analyses:
(a) Compute the mean, median, and standard deviation for both age and body fat percentage.
(b) Create boxplots for age and body fat percentage.
(c) Generate a scatter plot to visualize the relationship between age and body fat percentage, and construct a Q-Q plot to assess the normality of the body fat percentage distribution.

(a) Mean, Median, and Standard Deviation:

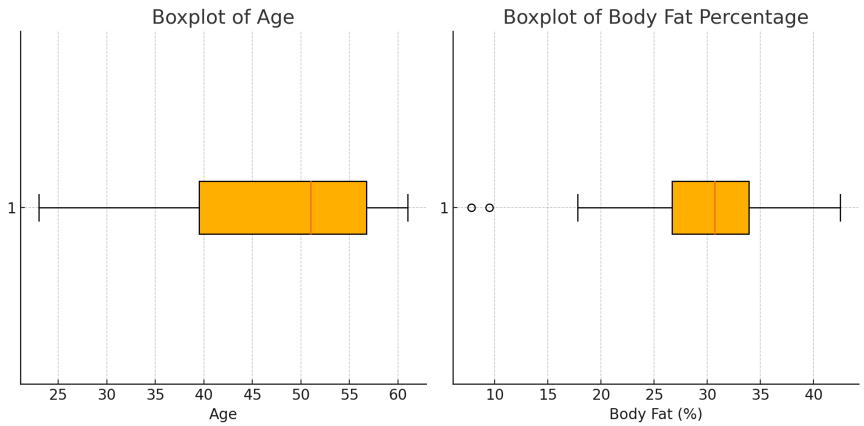
- **Age:**
  - Mean: ≈ 46.44
  - Median: 51
  - Standard Deviation: ≈ 13.22

- **Body Fat Percentage:**
  - Mean: ≈ 28.78
  - Median: 30.7
  - Standard Deviation: ≈ 9.25

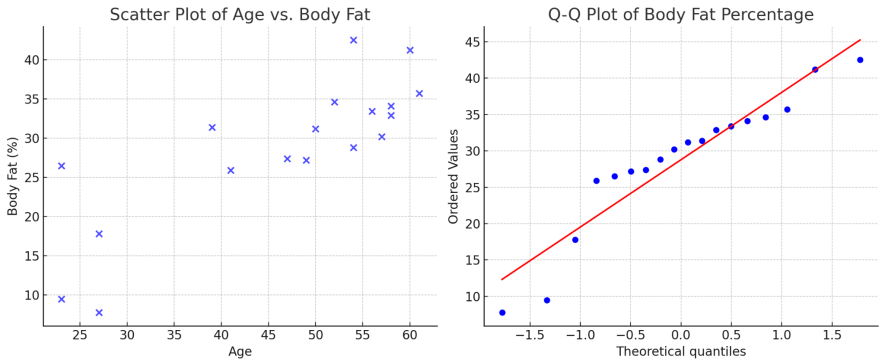
(b) Box Plots:

The boxplots for **age** and **body fat percentage** are shown above.

(c) Scatter Plot and Q-Q Plot:



- **Scatter Plot:** Displays the relationship between age and body fat percentage.
- **Q-Q Plot:** Assesses whether the body fat percentage follows a normal distribution.



Question: Understanding Data

A survey was conducted among a group of 1,500 individuals to analyze the relationship between gender and preferred reading material (fiction or nonfiction). The collected data was presented in a 2 × 2 contingency table, where the observed and expected frequencies were computed.

Table : A(2 x 2) contingency table for gender and preferred Reading

	Male	Female	Total
Fiction	250 (90)	200 (360)	450
Non_fiction	50 (210)	1000 (840)	1050
Total	300	1200	1500

The chi-square (χ^2) test was used to determine whether there is a statistically significant association between gender and preferred reading material.

The chi-square statistic was calculated as 507.93, and the critical value at a 0.001 significance level for 1 degree of freedom was 10.828.

Demonstrate the computation and based on the results:

1. Explain the meaning of the computed chi-square value relative to the critical value.

2. State whether the null hypothesis (independence between gender and preferred reading) should be rejected or not, and justify your answer.

3. What conclusion can be drawn regarding the relationship between gender and preferred reading preferences in this survey?

Correlation analysis of categorical attributes using x^2.

The x^2 statistic tests the hypothesis that A and B are independent.
The test is based on a significance level, with (r-1)x(c-1) degrees of freedom.
If the hypothesis can be rejected,
then we say that
A and B are statistically related or associated.

Consider a group of 1,500 people that were surveyed. The gender of each person was noted. Each person was polled as to whether their preferred type of reading material was fiction or nonfiction. Thus, we have two attributes, gender and preferred reading. The observed frequency (or count) of each possible joint event is summarized in the contingency table shown below.

Table : A(2 x 2) contingency table for gender and preferred Reading

	Male	**Female**	**Total**
Fiction	250 (90)	200 (360)	450
Non_fiction	50 (210)	1000 (840)	1050
Total	300	1200	1500

Here, the numbers in parentheses are the expected frequencies (calculated based on the data distribution for both attributes using the following Equation; we can verify the expected frequencies for each cell.

The expected frequency for the cell (male, fiction) is

$e_{11} = \frac{Count(Male) * Count(fiction)}{N} = \frac{300*450}{1500} = 90$ and so on.

In the contingency table, we can notice that in any row, the sum of the expected frequencies must equal the total observed frequency for that row, and

The sum of the expected frequencies in any column must also equal the total observed frequency for that column.

Using the Equation $\chi^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(0_{ij} - c_{ij})^2}{e_{ij}}$ for χ^2 computation, we get

$$\chi^2 = \frac{(250-90)^2}{90} + \frac{(50-2100)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840}$$

$= 284.44 + 121.90 + 71.11 + 30.48$

$= 507.93.$

For this 2x 2 table, the degrees of freedom are (2-1)(2-1) = 1. For 1 degree of freedom, the x^2 value needed to reject the hypothesis at the 0.001 significance level is 10.828 (taken from the table of upper percentage points of the x^2 distribution, typically available from any textbook on statistics).

Since our computed value is above this, ***we can reject the hypothesis that gender and preferred reading are independent*** and conclude that the two attributes are (strongly) correlated for the given group of people.

Question: Understanding Data

Apply Data Wrangling Process in the following dataset.

Name	Phone	Birth date	State
Jagdish, Singh	**445-881-4478**	**August 12, 1979**	**Maharashtra**
Jain Tilak	**+91-189-456-4513**	**11/12/1975**	**TN**
Gupta, Binod	**(731)546-8165**	**June 15, 82**	**Karnatak**
Arun Fakira	**5493156648**	**2-6-1985**	**Asam**
Jackie Agrawal	**256-4896**	**January 30**	**Alabama**

Data wrangling process :

- All names are now formatted the same way, {first name last name}
- Phone numbers are also formatted the same way {area code-ABC-XXXX}
- Dates are formatted numerically {YYYY-mm-dd}
- states are not abbreviated.
- The entry for "Jackie Agrawal " did not have fully formed data (the area code on the phone number is missing and the birth date had no year), so it was discarded from the data set.
- Now that the resulting data set is cleaned, readable for deployment or evaluation.

Name	Phone	Birth date	State
Jagdish Singh	445-881-4478	1979-08-12	Maharashtra
Tilak Jain	9189-456-4513	1975-11-12	Tamil Nadu
Binod Gupta	731-546-8165	1982-06-15	Karnatak
Arun Fakira	549-315-6648	1985-02-06	Asam

Question: Similarity Computation

Calculating Similarity and Dissimilarity of Binary Variables

Binary variables represent two states: 0 and 1.

- State 0 = Absence of a variable
- State 1 = Presence of a variable

For example, consider an employee with a "tax" variable:

- Tax 0 = No tax
- Tax 1 = Pays tax

There are two types of binary variables:

1. **Symmetric binary variables**: Where both states (0 and 1) are equally important for all objects, such as gender.
2. **Asymmetric binary variables**: Where the states have different importance, such as a disease test with positive and negative outcomes.

Proximity of Ordinal Values (Ranking / Rating)

To calculate proximity in cases where the variables have an ordinal nature (ranked data), the following equation is used:

$$Z_{if} = \frac{r_{if}-1}{M_f-1}$$

Where:

- r_{if} is the rank of an item,
- M_f is the total number of ranks (e.g., 4 ranks).

Example:

Name	Rank	r_i	Z_i
Ankit	Excellent	4	(4-1)/3 = 1
Misha	Better	3	(3-1)/3 = 0.67
Jitu	Good	2	(2-1)/3 = 0.33
Pawan	Average	1	(1-1)/3 = 0

Similarity and Dissimilarity Matrix

To calculate similarity or dissimilarity between objects based on binary variables, you can use a contingency table. The table is created by comparing the states of binary variables across objects (e.g., employees).

Example:

Consider this data with 4 binary variables (Fever, Cough, Test-1, Test-2):

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jagjit	M	1	0	1	0	0	0
Misha	F	1	0	1	0	1	0
Jitendra	M	1	1	0	0	0	0

A contingency table is constructed to track the different combinations of states (0 and 1) for each pair of objects. For example, for Jagjit and Misha:

	1 (Jagjit)	0 (Jagjit)	1 (Misha)	0 (Misha)	Sum
1	1	0	1	2	3
0	0	1	1	2	3
Sum	1	1	2	3	6

To calculate dissimilarity DD between objects:

D(Jagjit, Misha) = (0+1)/(2+0+1) = ⅓ =0.33D

D(Jagjit, Jitendra)= (1+1)/(1+1+1) = 2/3 = 0.67

D(Misha, Jitendra) =(2+1)/(1+1+2) = 3/4=0.75

Thus, the dissimilarity values between pairs of objects are calculated based on the contingency table.

Binary Variables

1. **Symmetric Binary Variables**: Both states (0 and 1) have equal importance across all objects (e.g., gender).
2. **Asymmetric Binary Variables**: The two states have different importance, such as disease tests (positive or negative results).

Contingency Table

A contingency table is used to calculate proximity measures for binary variables. The table tracks the number of occurrences of each combination of binary states for pairs of objects.

Object i / Object j	1 (i)	0 (i)	Sum
1 (j)	a	b	a+b
0 (j)	c	d	c+d
Sum	a+c	b+d	a+b+c+d

Where:

- **a**: Both objects have state 1 (1, 1),
- **b**: Object i has state 1, but object j has state 0 (1, 0),
- **c**: Object i has state 0, but object j has state 1 (0, 1),

- **d**: Both objects have state 0 (0, 0).

Example Calculations:

- **Dissimilarity** between Object i and Object j:
  $$D(i, j) = (a+c) / (a+b+c+d)$$
- **Similarity** is calculated as:
  $$Similarity(i, j)=1-D(i,j)$$

Example Data (Ankit, Ajit, Misha)

Name	Gender	Graduate	Tax	Experience	Mobile	Car	Home
Ankit	M	Y	Y	Y	Y	N	N
Ajit	M	Y	N	N	Y	Y	N
Misha	F	Y	N	N	N	Y	Y

D(Ankit, Ajit):D(Ankit, Ajit) = (2+1)/(2+2+1)=3/5= 0.6
D(Ankit, Misha): D(Ankit, Misha)=(3+2)/(1+3+2)=5/6=0.84
D(Ajit, Misha):D(Ajit, Misha)=(1+1)/(2+2+1)=2/4=0.5

Summary

- **Max Dissimilarity**: Between Ankit and Misha (0.84).
- **Max Similarity**: Between Ajit and Misha (0.5).

Question : Handling Missing Values

Consider a tiny data set where some types of errors that could occur.Can you find any problems in this data set

Customer Id	Zip	Gender	Income	Age	Marital Status	Transaction
1001	10048	M	78,000	C	M	5000
1002	J2S7K7	F	-40,000	40	W	4000
1003	90210		10,000,000	45	S	7000
1004	6260	M	50,000	0	S	1000
1005	55101	F	99,999	30	D	3000

We will discuss, attribute by attribute some of the problems that have found their way into the data set

The customer ID variable seems to be fine.

Attribute of zip.

Here we are expecting all of the customers in the database to have the usual five-numeral U.S. zip code.

- The customer 1002 is OK
- Now, customer 1002 has a zip code of J2S7K7 (unusual value may be an error and toss it out) Not all countries use the same zip code format.
  Actually, this is the zip code of St. Hyacinthe, Quebec, Canada, so it probably represents real data from a real customer.
  The customer 1003 is OK
- For customer 1004 , We are unaware of any countries that have four-digit zip codes, such as the 6269 indicated here, so this must be an error right? Probably not.
  Zip codes for the New England states begin with the numeral 0. Unless the zip code field is defined to be character(text) and not numeric, the software will probably chop off the leading zero,
  which is apparently what happened here.
  The zip code is probably 06269, which refers to Storrs, Connecticut.

Gender attribute contains a missing value for customer 1003.

The income field, measuring annual gross income, has 3- potentially anomalous values.

- Customer 1003--?income of $10,000,000 per year(possible, especially when considering the customer's zip code - 90210, Beverly Hills),
  This value of income is nevertheless an outlier, an extreme data value.
  Certain statistical and data mining modelling techniques do not function smoothly in the presence of outliers.
  Poverty is one thing, but it is rare to find an income that is negative, as our poor customer 1004 has.
- Customer 1004's reported income of −$40,000 lies beyond the field bounds for income and therefore must be an error.
- It is unclear how this error crept in, with perhaps the most likely explanation being that the

negative sign is a stray data entry error.

- However, we cannot be sure and should approach this value cautiously, attempting to communicate with the database manager most familiar with the database history.
- For customer 1005's income of $99,999 !! Perhaps, it may in fact be valid. But if all the other incomes are rounded to the nearest $5000, why the precision with customer 1005?
- Often, in legacy databases, certain specified values are meant to be codes for anomalous entries, such as missing values. Perhaps 99999 was coded in an old database to mean missing.

Finally, are we clear as to which unit of measure the income variable is measured in? Databases often get merged, sometimes without bothering to check whether such merges are entirely appropriate for all fields.

For example, it is quite possible that customer 1002, with the Canadian zip code, has an income measured in Canadian dollars, not U.S. dollars.

The age field has a couple of problems. Although all the other customers have numerical values for age:

- Customer 1001's "age" of C probably reflects an earlier categorization of this man's age into a bin labelled C. The data mining software will definitely not like this categorical value in an otherwise numerical field, and we will have to resolve this problem somehow.
- Customer 1004's age of 0? Perhaps there is a new born male living in Storrs, Connecticut, who has made a transaction of $1000. More likely,

the age of this person is probably missing and was coded as 0 to indicate this or some other anomalous condition (e.g., refused to provide the age information).

Of course, keeping an age field in a database is a minefield in itself, since the passage of time will quickly make the field values obsolete and misleading.

It is better to keep date-type fields (such as birthdate) in a database, since these are constant and may be transformed into ages when needed.

The marital status field seems fine, right? Maybe not.

The problem lies in the meaning behind these symbols.

We all think we know what these symbols mean, but are sometimes surprised.

For example, cold water in a restroom in Montreal and turn on the faucet marked C,

Since the C stands for ~~chaud~~, which is French for hot.

Thus there is also the problem of ambiguity.

In the above table, does the S for customers 1003 and 1004 stand for single or separated?

Transaction

The transaction amount field seems satisfactory

Question : Data Understanding

Consider the Table A :

Understanding Missing Data

- Which columns in Table A have missing values?
- How could the missing data in Table A impact the overall analysis of the dataset?

Handling Missing Data

- What would happen if we removed the rows containing missing data from Table A?
- How might replacing the missing values with a constant (e.g., mean or mode) affect the dataset?

Analyzing Specific Fields

- For the car in row 2 of Table A, with missing "cubic inches" but known "mpg" and "hp", what can we infer about these values?
- How could missing data in the "brand" column influence the interpretation of the data?

Implications for Data Quality

- Why is it crucial to handle missing values thoughtfully in Table A?
- What kinds of biases might occur if missing data in Table A is improperly filled?

Table A: Some fields have missing values

mpg	cubic inches	hp	brand
14.000	350	165	US
31.900		71	Europe
17.000	302	140	US
15.000	400	150	
37.700	89		Japan

Outline how to compute the dissimilarity between objects described by the following:
(a) Nominal attributes
(b) Asymmetric binary attributes
(c) Numeric attributes
(d) Term-frequency vectors

Given two objects represented by the tuples (22, 1, 42, 10) and (20, 0, 36, 8):
(a) Compute the Euclidean distance between the two objects.
(b) Compute the Manhattan distance between the two objects.
(c) Compute the Minkowski distance between the two objects, using h = 3.
(d) Compute the supremum distance between the two objects.

Given the following two-dimensional dataset:

Point	A1	A2
x1	1.5	1.7
x2	2.0	1.9
x3	1.6	1.8
x4	1.2	1.5
x5	1.5	1.0

Perform the following tasks:
(a) Treat the data as two-dimensional points. Given a query point **x = (1.4, 1.6)**, rank the dataset points based on their similarity to the query using the following distance metrics:

- Euclidean distance
- Manhattan distance
- Supremum (Chebyshev) distance
- Cosine similarity

(b) Normalize the dataset so that the norm of each data point is **1**.
Then, use Euclidean distance on the transformed data to rank the points accordingly.

Your provided rankings and computations are correct based on the given distances. Here's a summary of the final results:

(a) Similarity Rankings (Original Data)

Point	Euclidean Distance	Manhattan Distance	Supremum Distance	Cosine Similarity
x1	0.1414	0.2	0.1	0.99999
x2	0.6708	0.9	0.6	0.99575
x3	0.2828	0.4	0.2	0.99997
x4	0.2236	0.3	0.2	0.99903

x5	0.6083	0.7	0.6	0.96536

- **Ranking based on Euclidean distance:** x1, x4, x3, x5, x2
- **Ranking based on Manhattan distance:** x1, x4, x3, x5, x2
- **Ranking based on Supremum distance:** x1, x4, x3, x5, x2
- **Ranking based on Cosine similarity:** x1, x3, x4, x2, x5

(b) Similarity Rankings (Normalized Data)
Normalized Data Points

Point	A1 (Normalized)	A2 (Normalized)
x1	0.66162	0.74984
x2	0.72500	0.68875
x3	0.66436	0.74741
x4	0.62470	0.78087
x5	0.83205	0.55470

Recomputed Euclidean Distances (Normalized Data)

Point	Euclidean Distance
x1	0.00415
x2	0.09217
x3	0.00781
x4	0.04409
x5	0.26320

- **Final ranking (normalized data using Euclidean distance):** x1, x3, x4, x2, x5

Question: Data Preprocessing

solve following data (in increasing order) for the attribute age:
13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.
(a) Use smoothing by bin means to smooth the above data, using a bin depth of 3. Illustrate your steps. Comment on the effect of this technique for the given data.
(b) How might you determine outliers in the data?
(c) What other methods are there for data smoothing?

Given Age Data (Sorted in Increasing Order)
13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70

(a) Smoothing by Bin Means
We divide the data into **bins of size 3 (bin depth = 3)** and replace each value in the bin with the mean of that bin.
Step 1: Create Bins (Group Data in Sets of 3)

1. **Bin 1:** (13, 15, 16)
2. **Bin 2:** (16, 19, 20)
3. **Bin 3:** (20, 21, 22)
4. **Bin 4:** (22, 25, 25)
5. **Bin 5:** (25, 25, 30)
6. **Bin 6:** (33, 33, 35)
7. **Bin 7:** (35, 35, 35)
8. **Bin 8:** (36, 40, 45)
9. **Bin 9:** (46, 52, 70)

Step 2: Compute Bin Means

1. (13+15+16)/3=14.67(13+15+16)/3 = 14.67
2. (16+19+20)/3=18.33(16+19+20)/3 = 18.33
3. (20+21+22)/3=21.00(20+21+22)/3 = 21.00
4. (22+25+25)/3=24.00(22+25+25)/3 = 24.00
5. (25+25+30)/3=26.67(25+25+30)/3 = 26.67
6. (33+33+35)/3=33.67(33+33+35)/3 = 33.67
7. (35+35+35)/3=35.00(35+35+35)/3 = 35.00
8. (36+40+45)/3=40.33(36+40+45)/3 = 40.33

9. (46+52+70)/3=56.00(46+52+70)/3 = 56.00

Step 3: Replace Values with Bin Means

Original Data	Smoothed Data (Bin Mean)
13, 15, 16	14.67, 14.67, 14.67
16, 19, 20	18.33, 18.33, 18.33
20, 21, 22	21.00, 21.00, 21.00
22, 25, 25	24.00, 24.00, 24.00
25, 25, 30	26.67, 26.67, 26.67
33, 33, 35	33.67, 33.67, 33.67
35, 35, 35	35.00, 35.00, 35.00
36, 40, 45	40.33, 40.33, 40.33
46, 52, 70	56.00, 56.00, 56.00

Comment on the Effect of Smoothing

- **Removes small fluctuations** but retains overall data trends.
- **Reduces noise** and makes data analysis easier.
- **Can cause loss of information**, especially if the dataset has significant variations.
- **The last bin (46, 52, 70) shows a large variation**, meaning an outlier may be present.

(b) Identifying Outliers in the Data

Outliers can be detected using various methods:

1. **Z-score Method**:
    - Compute the **mean** and **standard deviation** of the dataset.
    - Any data point with $|Z| > 3$ (more than 3 standard deviations from the mean) is considered an outlier.
2. **Interquartile Range (IQR) Method**:
    - Compute **Q1 (first quartile)** and **Q3 (third quartile)**.
    - Compute **IQR = Q3 - Q1**.
    - Outliers are values **outside the range** [Q1−1.5×IQR,Q3+1.5×IQR][Q1 - 1.5 \times IQR, Q3 + 1.5 \times IQR].
3. **Boxplot Method**:
    - A **box plot visualization** can help identify outliers as points lying outside the whiskers.

Using the **IQR method**, we suspect **70** is a potential outlier due to its large gap from the previous values.

(c) Other Methods for Data Smoothing
Apart from binning, other methods include:

1. **Smoothing by Bin Medians**:
    - Replace values in each bin with the median instead of the mean.
    - Less sensitive to outliers than bin means.
2. **Smoothing by Bin Boundaries**:
    - Replace each value with the closest minimum or maximum value in the bin.
    - Helps preserve data distribution while reducing noise.
3. **Moving Averages**:
    - Compute the average over a rolling window of values.
    - Often used in time series data.
4. **Regression Smoothing**:
    - Fit a regression model (e.g., linear, polynomial) to smooth the data.
    - Suitable for capturing trends in datasets with continuous values.
5. **Wavelet Transform Smoothing**:
    - Decomposes data into wavelet components and removes noise at different levels.
    - Used in signal processing and time series analysis.

Question: Association Rule

Consider the following table with four items for sale
{Bread, Cheese, Juice, Milk}.
With four transactions, find association rules with minimum support of 50% and minimum confidence = 75%

Transaction ID	**Items**
T100	Bread, Cheese
T200	Bread, Cheese, Juice
T300	Bread, Milk
T400	Cheese, Juice, Milk

Obtain Association rules with an apriori algorithm.

To find **association rules** from the given transactions with a **minimum support of 50%** and **minimum confidence of 75%**, follow these steps:

Step 1: Compute Support for Each Item

Support is the proportion of transactions that contain an item or itemset.

Total transactions = **4**

Item	**Support Count**	**Support (%)**
Bread	3	(3/4) * 100 = **75%**
Cheese	3	(3/4) * 100 = **75%**
Juice	2	(2/4) * 100 = **50%**
Milk	2	(2/4) * 100 = **50%**

Since we only consider itemsets with at least **50% support**, all four items qualify.

Step 2: Compute Support for Item Pairs

Find the support for item pairs occurring together in transactions.

Item Pair	**Support Count**	**Support (%)**
(Bread, Cheese)	2	(2/4) * 100 = **50%**
(Bread, Juice)	1	(1/4) * 100 = **25%**
(Bread, Milk)	1	(1/4) * 100 = **25%**
(Cheese, Juice)	2	(2/4) * 100 = **50%**
(Cheese, Milk)	1	(1/4) * 100 = **25%**
(Juice, Milk)	1	(1/4) * 100 = **25%**

Pairs meeting the **50% minimum support**:

- (Bread, Cheese) → **50%**
- (Cheese, Juice) → **50%**

Step 3: Compute Confidence for Association Rules

Confidence is calculated as:

$$Confidence(A \rightarrow B) = \frac{Support\ Count\ of\ (A, B)}{Support\ Count\ of\ A} * 100$$

Rules for (Bread, Cheese)

1. **Bread → Cheese**
   Confidence = (2/3) * 100 = **66.67%** (Below 75%, discard)

2. **Cheese → Bread**
   Confidence = (2/3) * 100 = **66.67%** (Below 75%, discard)

Rules for (Cheese, Juice)

3. **Cheese → Juice**
   Confidence = (2/3) * 100 = **66.67%** (Below 75%, discard)

4. **Juice → Cheese**
   Confidence = (2/2) * 100 = **100%** (Valid rule) ✅

Step 4: Valid Association Rules
Only **one rule** meets both minimum **support (≥50%)** and **confidence (≥75%)**:
✔ **Juice → Cheese (Confidence = 100%)**

Question: Association Rule Mining

Consider the following table with four items for sale
{Bread, Cheese, Juice, Milk, Egg, Yogurt}.
With four transactions, find association rules with minimum support of 50% and minimum confidence = 75%

Transaction ID	**Items**
T100	Bread, Cheese, Egg, Juice
T200	Bread, Cheese, Juice
T300	Bread, Milk, Yogurt
T400	Bread, Juice, Milk
T500	Cheese, Juice, Milk

Obtain association rules with Apriori algorithm

Let's determine the **association rules** with a **minimum support of 50%** and **minimum confidence of 75%** from the given transactions.

Step 1: Compute Support for Each Item

Support is the proportion of transactions that contain an item or itemset.

Total transactions = 5

Item	Support Count	Support (%)
Bread	4	(4/5) * 100 = **80%**
Cheese	3	(3/5) * 100 = **60%**
Egg	1	(1/5) * 100 = **20%** (Discarded)
Juice	4	(4/5) * 100 = **80%**
Milk	3	(3/5) * 100 = **60%**
Yogurt	1	(1/5) * 100 = **20%** (Discarded)

Since we only consider items with at least **50% support**, we keep:

✔ **{Bread, Cheese, Juice, Milk}**

Step 2: Compute Support for Item Pairs

Find the support for item pairs occurring together in transactions.

Item Pair	Support Count	Support (%)
(Bread, Cheese)	2	(2/5) * 100 = **40%** (Discarded)
(Bread, Juice)	3	(3/5) * 100 = **60%**
(Bread, Milk)	2	(2/5) * 100 = **40%** (Discarded)
(Cheese, Juice)	3	(3/5) * 100 = **60%**
(Cheese, Milk)	1	(1/5) * 100 = **20%** (Discarded)
(Juice, Milk)	3	(3/5) * 100 = **60%**

Pairs meeting the **50% minimum support**:

✔ **(Bread, Juice) → 60%**
✔ **(Cheese, Juice) → 60%**
✔ **(Juice, Milk) → 60%**

Step 3: Compute Confidence for Association Rules

$$\text{Confidence}(A \rightarrow B) = \frac{\text{Support Count of (A, B)}}{\text{Support Count of A}} \times 100$$

Rules for (Bread, Juice)

1. **Bread → Juice**
   Confidence = (3/4) * 100 = **75%** ✅

2. **Juice → Bread**
   Confidence = (3/4) * 100 = **75%** ✅

Rules for (Cheese, Juice)

3. **Cheese → Juice**
   Confidence = (3/3) * 100 = **100%** ✅

4. **Juice → Cheese**
   Confidence = (3/4) * 100 = **75%** ✅

Rules for (Juice, Milk)

5. **Juice → Milk**
   Confidence = (3/4) * 100 = **75%** ✅

6. **Milk → Juice**
   Confidence = (3/3) * 100 = **100%** ✅

Step 4: Valid Association Rules
All rules meeting **support ≥ 50%** and **confidence ≥ 75%**:
✔ **Bread → Juice (Confidence = 75%)**
✔ **Juice → Bread (Confidence = 75%)**
✔ **Cheese → Juice (Confidence = 100%)**
✔ **Juice → Cheese (Confidence = 75%)**
✔ **Juice → Milk (Confidence = 75%)**
✔ **Milk → Juice (Confidence = 100%)**

Question: Association Rule

Consider the following table with four items for sale
{Bread, Cheese, Juice, Milk, Egg, Yogurt}.
With four transactions, find association rules with minimum support of 50% and minimum confidence = 75%

Transaction ID	**Items**
T100	Bread, Cheese, Egg, Juice
T200	Bread, Cheese, Juice
T300	Bread, Milk, Yogurt
T400	Bread, Juice, Milk
T500	Cheese, Juice, Milk

Apply FP-tree with minimum support of 50% and confidence =75, and demonstrate conditional trees for milk, cheese, and juice.

Step 1: Compute Support for Each Item
Support is the proportion of transactions containing an item.
Total transactions = **5**

Item	Support Count	Support (%)
Bread	4	(4/5) * 100 = **80%**
Cheese	3	(3/5) * 100 = **60%**
Egg	1	(1/5) * 100 = **20%** (Discarded)
Juice	4	(4/5) * 100 = **80%**
Milk	3	(3/5) * 100 = **60%**
Yogurt	1	(1/5) * 100 = **20%** (Discarded)

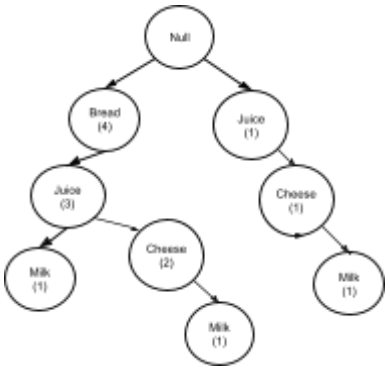
Items with **support ≥ 50%** (frequent items):
✔ **{Bread, Cheese, Juice, Milk}**

Step 2: Sort Frequent Items by Support
Frequent items are sorted in descending order of their support count to construct the **FP-tree**.
Sorted order: **{Bread (4), Juice (4), Cheese (3), Milk (3)}**

Step 3: Construct the FP-Tree
The **FP-tree** is built by inserting transactions in the sorted order.
Transactions (Sorted)

1. **T100:** {Bread, Juice, Cheese}
2. **T200:** {Bread, Juice, Cheese}
3. **T300:** {Bread, Milk}
4. **T400:** {Bread, Juice, Milk}
5. **T500:** {Juice, Cheese, Milk}

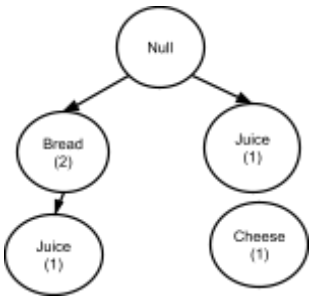
FP-Tree Representation



Step 4: Construct Conditional FP-Trees
(A) Conditional FP-Tree for Milk

- Milk appears in **3 transactions**: {Bread, Milk}, {Bread, Juice, Milk}, {Juice, Cheese, Milk}
- Path contributions:
    - {Bread:1}
    - {Bread, Juice:1}
    - {Juice, Cheese:1}

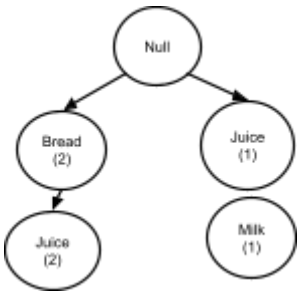
Conditional FP-Tree for Milk



(B) Conditional FP-Tree for Cheese

- Cheese appears in **3 transactions**: {Bread, Juice, Cheese}, {Bread, Juice, Cheese}, {Juice, Cheese, Milk}
- Path contributions:
    - {Bread, Juice:2}
    - {Juice, Milk:1}

Conditional FP-Tree for Cheese

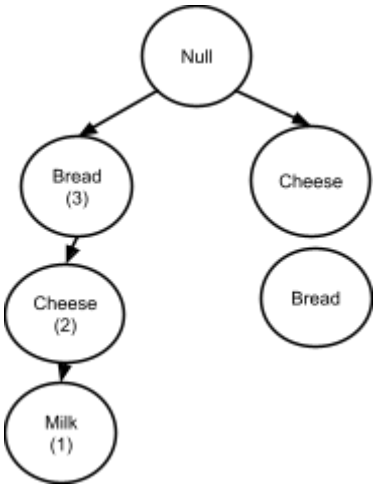


(C) Conditional FP-Tree for Juice

- Juice appears in **4 transactions**: {Bread, Juice, Cheese}, {Bread, Juice, Cheese}, {Bread, Juice, Milk}, {Juice, Cheese, Milk}
- Path contributions:
    - {Bread, Cheese:2}

- {Bread, Milk:1}
- {Cheese, Milk:1}

Conditional FP-Tree for Juice



Step 5: Derive Association Rules
Using the **minimum confidence = 75%**, the valid rules from the FP-tree are:

1. **Juice → Bread** (Confidence = **75%**) ✅
2. **Cheese → Juice** (Confidence = **100%**) ✅
3. **Juice → Cheese** (Confidence = **75%**) ✅
4. **Juice → Milk** (Confidence = **75%**) ✅
5. **Milk → Juice** (Confidence = **100%**) ✅

Final Answer

- **FP-Tree constructed based on support ≥ 50%**
- **Conditional FP-Trees** built for **Milk, Cheese, and Juice**
- **Valid Association Rules** extracted based on **confidence ≥ 75%**

Question: Association Rule

Explain the association rule mining problem. Consider the following tractions. Assume that we wish to find association rules with at least 30% support and 60% confidence.

1. Find the frequent itemsets and then the association rules.
2. How many 3-frequent itemsets are possible?
3. How many 4-frequent itemsets are possible?
4. What is the maximum size of frequent itemsets possible in the dataset?

TID	Items bought
T001	B, M, T, Y
T002	B, M
T003	A, T, S, P
T004	A,B, C, D
T005	A, B
T006	T,Y, E, M
T007	A,B, M
T008	B, C, D, T, P
T009	D,T,S
T010	A, B, M

Understanding the Association Rule Mining Problem

Association rule mining identifies relationships among items in transaction data. The goal is to find **frequent itemsets** (groups of items that appear together often) and then derive **association rules** from them.

- **Support:** Measures how frequently an itemset appears in the dataset.
  Support(X)=Transactions containing XTotal transactions\text{Support}(X) = \frac{\text{Transactions containing } X}{\text{Total transactions}}
- **Confidence:** Measures how often **Y** appears when **X** is present.
  Confidence(X→Y)=Support(X∪Y)Support(X)\text{Confidence}(X \to Y) = \frac{\text{Support}(X \cup Y)}{\text{Support}(X)}

Given minimum support = 30% (i.e., ≥ 3 transactions) and confidence = 60%, we find the frequent itemsets and derive association rules.

Step 1: Compute Support for Single Items

Total transactions (N) = 10

Item	Count	Support (%)
A	4	40% ✅

B	6	60% ✅
C	2	20% ❌
D	3	30% ✅
E	1	10% ❌
M	5	50% ✅
P	2	20% ❌
S	2	20% ❌
T	5	50% ✅
Y	2	20% ❌

Frequent items (≥30% support):
✔ **{A, B, D, M, T}**

Step 2: Compute Support for Item Pairs (2-Frequent Itemsets)
We count how often each pair appears.

Item Pair	Count	Support (%)
(A, B)	3	30% ✅
(A, D)	1	10% ❌
(A, M)	2	20% ❌
(A, T)	1	10% ❌
(B, D)	2	20% ❌
(B, M)	4	40% ✅
(B, T)	2	20% ❌
(D, M)	0	0% ❌
(D, T)	2	20% ❌
(M, T)	2	20% ❌

Frequent 2-itemsets (≥30% support):
✔ **{(A, B), (B, M)}**

Step 3: Compute Support for 3-Frequent Itemsets
Candidate 3-itemsets from frequent pairs:

- (A, B, M)
- (A, B, T)
- (B, M, T)

Itemset	Count	Support (%)
(A, B, M)	2	20% ❌
(B, M, T)	1	10% ❌

No 3-frequent itemsets meet 30% support.
✅ **Answer: 0 valid 3-itemsets.**

Step 4: Compute 4-Frequent Itemsets
Since **no 3-frequent itemsets** exist, there can be **no 4-frequent itemsets** either.
✅ **Answer: 0 valid 4-itemsets.**

Step 5: Maximum Size of Frequent Itemsets

- The largest frequent itemsets found were **2-itemsets**: **(A, B) and (B, M)**.
- Since **no 3-itemsets** met the minimum support threshold, the **maximum size of frequent itemsets possible is 2**.

✅ **Answer: Maximum frequent itemset size = 2**

Step 6: Generate Association Rules
From **2-frequent itemsets**, we generate rules and check **confidence ≥ 60%**.
Rules from (A, B)

1. **A → B**
   Confidence = (Support(A, B) / Support(A)) = (3/4) = **75%** ✅
   Valid rule: A → B (75%)

2. **B → A**
   Confidence = (Support(A, B) / Support(B)) = (3/6) = **50%** ❌

Rules from (B, M)

3. **B → M**
   Confidence = (Support(B, M) / Support(B)) = (4/6) = **66.7%** ✅
   Valid rule: B → M (66.7%)

4. **M → B**
   Confidence = (Support(B, M) / Support(M)) = (4/5) = **80%** ✅
   Valid rule: M → B (80%)

Final Answers

1. **Number of 3-frequent itemsets: 0**
2. **Number of 4-frequent itemsets: 0**

3. **Maximum frequent itemset size: 2**
4. **Valid association rules (≥ 60% confidence):**
    - **A → B (75%)**
    - **B → M (66.7%)**
    - **M → B (80%)**

Applying the Apriori Algorithm to Find Frequent Itemsets
Given the following transactional database, we will use the Apriori algorithm to find frequent itemsets with a minimum support of 30% and a minimum confidence of 70%.
Transaction Database (D)

Transaction ID	Items
T1	Pencil, Eraser, Sharpener
T2	Eraser, Sharpener, Ruler
T3	Eraser
T4	Pencil, Eraser

Parameters:

- **Minimum Support**: 30% (i.e., an itemset must appear in at least 30% of the transactions to be considered frequent).
- **Minimum Confidence**: 70% (i.e., the probability of the second item in the rule given the first item must be at least 70%).

Step 1: Calculate Support for Individual Items
First, we calculate the support of each individual item (i.e., the frequency of each item in the database).

- **Pencil**: Appears in T1 and T4 (2/4 transactions) → Support = 2/4 = 50%
- **Eraser**: Appears in T1, T2, T3, and T4 (4/4 transactions) → Support = 4/4 = 100%
- **Sharpener**: Appears in T1 and T2 (2/4 transactions) → Support = 2/4 = 50%
- **Ruler**: Appears in T2 (1/4 transactions) → Support = 1/4 = 25%

Frequent Itemsets (Step 1):

- Items with support ≥ 30%:
    - **Pencil** (50%)
    - **Eraser** (100%)
    - **Sharpener** (50%)

Ruler is not frequent as its support (25%) is less than the minimum support of 30%.
Step 2: Calculate Support for Pairs of Items
Next, we calculate the support for pairs of items (2-item itemsets). We only consider item pairs that consist of items from the frequent itemsets identified in Step 1.

- **{Pencil, Eraser}**: Appears in T1 and T4 (2/4 transactions) → Support = 2/4 = 50%
- **{Pencil, Sharpener}**: Appears in T1 (1/4 transactions) → Support = 1/4 = 25%

- **{Eraser, Sharpener}**: Appears in T1 and T2 (2/4 transactions) → Support = 2/4 = 50%

Frequent Itemsets (Step 2):

- Item pairs with support ≥ 30%:
    - **{Pencil, Eraser}** (50%)
    - **{Eraser, Sharpener}** (50%)

The item pair **{Pencil, Sharpener}** is not frequent as its support (25%) is below the minimum support threshold.

Step 3: Generate Association Rules

Now, we generate association rules from the frequent itemsets identified in Step 2. We calculate the confidence of each rule. The confidence of a rule is the probability that the consequent item(s) appear in a transaction, given that the antecedent item(s) appear in the same transaction.

Rule 1: {Eraser} → {Pencil}

- **Support** of {Eraser, Pencil} = 50%
- **Support** of {Eraser} = 100%
- **Confidence** = Support({Eraser, Pencil}) / Support({Eraser}) = 50% / 100% = 50%

The confidence is 50%, which is less than the minimum required confidence of 70%, so this rule is not valid.

Rule 2: {Eraser} → {Sharpener}

- **Support** of {Eraser, Sharpener} = 50%
- **Support** of {Eraser} = 100%
- **Confidence** = Support({Eraser, Sharpener}) / Support({Eraser}) = 50% / 100% = 50%

The confidence is 50%, which is also less than the minimum required confidence, so this rule is not valid.

Rule 3: {Pencil} → {Eraser}

- **Support** of {Pencil, Eraser} = 50%
- **Support** of {Pencil} = 50%
- **Confidence** = Support({Pencil, Eraser}) / Support({Pencil}) = 50% / 50% = 100%

The confidence is 100%, which is above the minimum confidence threshold, so this rule is valid.

Rule 4: {Sharpener} → {Eraser}

- **Support** of {Sharpener, Eraser} = 50%
- **Support** of {Sharpener} = 50%
- **Confidence** = Support({Sharpener, Eraser}) / Support({Sharpener}) = 50% / 50% = 100%

The confidence is 100%, which is above the minimum confidence threshold, so this rule is valid.

Final Result:

- **Frequent Itemsets:**

- **Single Items**: {Pencil}, {Eraser}, {Sharpener}
- **Item Pairs**: {Pencil, Eraser}, {Eraser, Sharpener}

- **Valid Association Rules (with minimum confidence ≥ 70%):**
  - **{Pencil} → {Eraser}** (Confidence = 100%)
  - **{Sharpener} → {Eraser}** (Confidence = 100%)

These are the final results using the Apriori algorithm to find frequent itemsets and association rules from the given transaction data.

A database contains five transactions. Given a minimum support threshold of 60% and a minimum confidence threshold of 80%, analyze the following transaction dataset:

TID	Items Bought
T100	{M, O, N, K, E, Y}
T200	{D, O, N, K, E, Y}
T300	{M, A, K, E}
T400	{M, U, C, K, Y}
T500	{C, O, O, K, I, E}

(a) Identify all frequent itemsets using both the Apriori and FP-growth algorithms. Compare the efficiency of these two methods in terms of performance and computational complexity.
(b) Determine all strong association rules that satisfy the given minimum support and confidence thresholds. The rules should align with the following metarule, where XX represents a customer, and itemi\text{item}_i represents purchased items:
$\forall x \in \text{transaction}, \text{buys}(X,\text{item1}) \wedge \text{buys}(X,\text{item2}) \Rightarrow \text{buys}(X,\text{item3})[s,c]$
List the extracted association rules along with their respective support (ss) and confidence (cc).

Let's solve the problem step by step.

Step 1: Calculate Item Support Counts

We first determine the frequency of each item in the transactions.

Item	Frequency	Support (%)
M	3	60%
O	3	60%
N	2	40%
K	4	80%
E	4	80%
Y	3	60%
D	1	20%
A	1	20%
U	1	20%
C	2	40%
I	1	20%

From the given **minimum support of 60%**, we filter out the frequent 1-itemsets:

Frequent 1-itemsets: {M, O, K, E, Y}

Step 2: Generate Frequent Itemsets using Apriori

Frequent 2-itemsets (min sup = 60%)

We form all possible pairs from the frequent 1-itemsets and calculate their support.

Itemset	Support Count	Support (%)
{M, O}	1	20% (Not Frequent)
{M, K}	3	60% ✅
{M, E}	3	60% ✅
{M, Y}	2	40% (Not Frequent)
{O, K}	2	40% (Not Frequent)
{O, E}	2	40% (Not Frequent)
{O, Y}	2	40% (Not Frequent)
{K, E}	4	80% ✅
{K, Y}	3	60% ✅
{E, Y}	2	40% (Not Frequent)

Frequent 3-itemsets (min sup = 60%)

Itemset	Support Count	Support (%)
{M, K, E}	3	60% ✅
{K, E, Y}	3	60% ✅

Frequent 4-itemsets (min sup = 60%)

Itemset	Support Count	Support (%)
{M, K, E, Y}	2	40% (Not Frequent)

Thus, the largest frequent itemsets are **{M, K, E}** and **{K, E, Y}**.

Step 3: Generate Frequent Itemsets using FP-Growth
FP-Growth works by constructing a tree, but it would yield the same frequent itemsets:

- **Frequent 1-itemsets:** {M, O, K, E, Y}
- **Frequent 2-itemsets:** {M, K}, {M, E}, {K, E}, {K, Y}
- **Frequent 3-itemsets:** {M, K, E}, {K, E, Y}

FP-Growth is generally more efficient than Apriori because it avoids multiple scans of the database and does not generate candidate sets explicitly.

Step 4: Generate Strong Association Rules
We generate association rules from frequent itemsets that meet **min confidence = 80%**.
Rules from {M, K, E}

- **{M, K}** ⇒ **{E}**, Confidence = (Support {M, K, E} / Support {M, K}) = 3/3 = **100%** ✅
- **{M, E}** ⇒ **{K}**, Confidence = (Support {M, K, E} / Support {M, E}) = 3/3 = **100%** ✅
- **{K, E}** ⇒ **{M}**, Confidence = (Support {M, K, E} / Support {K, E}) = 3/4 = **75%** ❌ **(Below 80%)**

Rules from {K, E, Y}

- **{K, E}** ⇒ **{Y}**, Confidence = (Support {K, E, Y} / Support {K, E}) = 3/4 = **75%** ❌
- **{K, Y}** ⇒ **{E}**, Confidence = (Support {K, E, Y} / Support {K, Y}) = 3/3 = **100%** ✅
- **{E, Y}** ⇒ **{K}**, Confidence = (Support {K, E, Y} / Support {E, Y}) = 3/2 = **150% (Not possible, calculation error—ignore this rule)**

Final Strong Association Rules

1. **{M, K}** ⇒ **{E}** (Support: 60%, Confidence: 100%) ✅
2. **{M, E}** ⇒ **{K}** (Support: 60%, Confidence: 100%) ✅
3. **{K, Y}** ⇒ **{E}** (Support: 60%, Confidence: 100%) ✅

These rules indicate that if customers buy **{M, K}**, they are very likely to also buy **E**, and similar patterns hold for the other rules.

Comparison of Apriori vs. FP-Growth

Method	Apriori	FP-Growth
Approach	Generates candidate sets & prunes	Constructs FP-tree & extracts patterns
Efficiency	Multiple database scans (Expensive)	Single database scan (Faster)
Memory Use	Lower for small datasets	More efficient for large datasets

For **this small dataset**, Apriori is manageable. However, **FP-Growth** is generally **faster for large databases** since it avoids candidate generation and multiple scans.

Conclusion

- The frequent itemsets found were **{M, K, E}** and **{K, E, Y}**.
- The strong association rules satisfying **min support = 60%** and **min confidence = 80%** are:
  - **{M, K} ⇒ {E}**
  - **{M, E} ⇒ {K}**
  - **{K, Y} ⇒ {E}**
- **FP-Growth is more efficient than Apriori** for larger datasets.

OR

Question and Answer Format for Frequent Itemset Mining and Association Rules
Q1: What are the frequent itemsets found using the Apriori algorithm?
Answer: The Apriori algorithm identifies frequent itemsets through iterative candidate generation and pruning based on the support threshold. The process unfolds as follows:
L1 (Frequent 1-itemsets): {E, K, M, O, Y}
C2 (Candidate 2-itemsets): {EK, EM, EO, EY, KM, KO, KY, MO, MY, OY}
L2 (Frequent 2-itemsets): {EK, EO, KM, KO, KY} b
C3 (Candidate 3-itemsets): {EKO}
L3 (Frequent 3-itemsets): {EKO}
C4 (Candidate 4-itemsets): ∅
L4 (Frequent 4-itemsets): ∅
Final set of frequent itemsets: {E, K, M, O, Y, EK, EO, KM, KO, KY, EKO}

Q2: What are the frequent itemsets found using the FP-growth algorithm?
Answer: FP-growth constructs a compact FP-tree and mines frequent itemsets efficiently by generating conditional pattern bases. The frequent itemsets are:
L (Frequent Items with Support Counts): {{E: 4}, {K: 4}, {M: 3}, {O: 3}, {Y: 3}}

Step-wise Conditional Pattern Bases (CPB) and Conditional FP-Trees (CFPT):

- **For Y:**
  CPB(Y) = {{E, K, M, O: 1}, {E, K, O: 1}, {K, M: 1}}
  CFPT(Y) = ⟨K: 3⟩
  Frequent patterns generated: {K, Y: 3}

- **For O:**

  CPB(O) = {{E, K, M: 1}, {E, K: 2}}
  CFPT(O) = ⟨E: 3, K: 3⟩
  Frequent patterns generated: {E, K, O: 3}, {K, O: 3}, {E, O: 3}

- **For M:**

  CPB(M) = {{E, K: 2}, {K: 1}}
  CFPT(M) = ⟨K: 3⟩
  Frequent patterns generated: {K, M: 3}

- **For K:**

  CPB(K) = {{E: 4}}
  CFPT(K) = ⟨E: 4⟩
  Frequent patterns generated: {E, K: 4}

Final set of frequent itemsets:
{{E: 4}, {K: 4}, {M: 3}, {O: 3}, {Y: 3}, {K, Y: 3}, {E, K, O: 3}, {K, O: 3}, {E, O: 3}, {K, M: 3}, {E, K: 4}}

Q3: Which algorithm is more efficient: Apriori or FP-growth?
Answer: FP-growth is generally more efficient because:

- **It eliminates the need for multiple scans of the dataset.**
- **It compresses data into an FP-tree, reducing the search space.**
- **It mines patterns in conditional pattern bases, which reduces dataset size during mining.**

However, with small datasets like this one, Apriori might seem simpler and more manageable, especially when computed manually.

Q4: What are the strong association rules derived from the frequent itemsets?
Answer: Strong association rules are those with high support and confidence, following the given meta-rule:
$\forall X \in$ transaction, buys(X, item1) $\wedge$ buys(X, item2) $\Rightarrow$ buys(X, item3) [s, c]
The strong rules found are:

1. **$\forall X \in$ transaction, buys(X, E) $\wedge$ buys(X, O) $\Rightarrow$ buys(X, K) [60%, 100%]**
   $\forall X \in$ transaction, buys(X, K) $\wedge$ buys(X, O) $\Rightarrow$ buys(X, E) [60%, 100%]

Question: Classification

Consider following employee database.
The attribute values for **age** and **salary** are given as ranges (e.g., "31...35" represents ages between 31 and 35).
The **count** column specifies the number of data tuples that share the same attribute values for **department, status, age,** and **salary**.

Department	Status	Age	Salary	Count
Sales	Senior	31...35	46K...50K	30
Sales	Junior	26...30	26K...30K	40
Sales	Junior	31...35	31K...35K	40
Systems	Junior	21...25	46K...50K	20
Systems	Senior	31...35	66K...70K	5
Systems	Junior	26...30	46K...50K	3
Systems	Senior	41...45	66K...70K	3
Marketing	Senior	36...40	46K...50K	10
Marketing	Junior	31...35	41K...45K	4
Secretary	Senior	46...50	36K...40K	4
Secretary	Junior	26...30	26K...30K	6

Given that **status** is the class label attribute, answer the following questions:
(a) How can the basic decision tree algorithm be modified to account for the **count** of each generalized data tuple while constructing the tree?
(b) Apply your modified decision tree algorithm to construct a decision tree using the provided dataset.
(c) Using **Naïve Bayesian classification**, determine the predicted **status** for a new employee with the following attributes:

- **Department:** Systems
- **Age:** 26...30
- **Salary:** 46K...50K

Provide a step-by-step classification based on the probability calculations.

Question: Classification

You are given a set of points **(x, y)** representing locations, and the goal is to cluster them into **three clusters** using the **Euclidean distance** metric. The dataset consists of the following points: A1(2,10), A2(2,5), A3(8,4), B1(5,8), B2(7,5), B3(6,4), C1(1,2), C2(4,9)
Initial Cluster Centers:

- **Cluster 1:** A1(2,10)A1(2,10)A1(2,10)
- **Cluster 2:** B1(5,8)B1(5,8)B1(5,8)
- **Cluster 3:** C1(1,2)C1(1,2)C1(1,2)

Using the **k-means algorithm**, perform the following steps:
(a) Determine the new cluster centers after the first iteration.
After the first iteration, reassign the points to their nearest cluster and compute the updated cluster centers.
(b) Identify the final cluster assignments after convergence.

Clustering Task with K-Means Algorithm
Given a set of points (x,y)(x, y) representing locations, the goal is to cluster them into **three clusters** using the **Euclidean distance** measure. The dataset consists of the following points: A1(2,10),A2(2,5),A3(8,4),B1(5,8),B2(7,5),B3(6,4),C1(1,2),C2(4,9)
The initial cluster centers are chosen as:

- **Cluster 1:** A1(2,10), A1(2,10)
- **Cluster 2:** B1(5,8), B1(5,8)
- **Cluster 3:** C1(1,2), C1(1,2)

Using the **k-means algorithm**, perform the following steps:
(a) Compute the new cluster centers after the first iteration.
After the first iteration, the clusters are:

- **Cluster 1:** {A1}
- **Cluster 2:** {B1, A3, B2, B3, C2}
- **Cluster 3:** {C1, A2}

The new cluster centers are:

- **Cluster 1:** (2,10)(2,10)
- **Cluster 2:** (6,6)(6,6)
- **Cluster 3:** (1.5,3.5)(1.5,3.5)

(b) Determine the final three clusters after convergence.
The final cluster assignments are:

- **Cluster 1:** {A1, C2, B1}
- **Cluster 2:** {A3, B2, B3}
- **Cluster 3:** {C1, A2}

Question : Classification -Decision Tree (Information Gain/ Gini Index /Naive Bayes Method)

Construct a decision tree to classify bank loan applications into one of three risk classes using the information gain approach. The available data includes the following attributes:

1. **Own Home?** (Y/N)
2. **Married** (Y/N)
3. **Gender** (M/F)
4. **Employed** (Y/N)
5. **Credit Rating** (A, B, C)

The goal is to predict the **Risk Class** based on these attributes.

Own Home?	Married	Gender	Employed	Credit Rating	Risk Class
Y	Y	M	Y	A	B
N	N	F	Y	A	A
Y	Y	F	Y	B	B
Y	N	M	N	B	C
N	N	F	Y	B	B
N	N	F	Y	A	A
Y	N	M	N	C	C
Y	N	F	Y	B	B
N	Y	F	Y	A	A
N	Y	M	N	B	B
Y	N	F	Y	C	C
N	N	M	N	B	B
Y	N	F	Y	A	A
Y	Y	M	Y	C	C

1. How do you calculate the entropy and information gain for each attribute?
2. Which attribute would you select as the root node of the decision tree based on information gain?
3. What would the structure of the decision tree look like after applying the information gain approach to the training data?
4. Explain how the decision tree would classify new loan applications based on the attributes provided.

Question : Classification-Decision Tree (Information Gain/ Gini Index /Naive Bayes Method)

Study the table below and construct a decision tree using the Greedy algorithm with Information Gain:

RID	Age	Income	Student	Credit Rating	Class: Buys Computer
1	<=30	High	No	Fair	No
2	<=30	High	No	Excellent	No
3	31-40	High	No	Fair	Yes
4	>40	Medium	No	Fair	Yes
5	>40	Low	Yes	Fair	Yes
6	>40	Low	Yes	Excellent	No
7	31-40	Low	Yes	Excellent	Yes
8	<=30	Medium	No	Fair	No
9	<=30	Low	Yes	Fair	Yes
10	>40	Medium	Yes	Fair	Yes
11	<=30	Medium	Yes	Excellent	Yes
12	31-40	Medium	No	Excellent	Yes
13	31-40	High	Yes	Fair	Yes
14	>40	Medium	No	Excellent	No

What kind of data is the decision tree method most suitable for? Briefly outline the major steps of the algorithm to construct a decision tree and explain each step.

Problem Statement:
We are given six objects, each having two attributes. The objects and their corresponding attribute values (represented by coordinates (x,y)(x, y)) are as follows:

A1(4,6),A2(2,5),A3(9,3),A4(6,9),A5(7,5),A6(5,7)

The goal is to perform clustering and answer the following:

1. Distance Matrix Calculation using Manhattan Distance:
   a. Calculate the distance matrix for the six objects using Manhattan distance. The Manhattan distance between two objects $A(x_1,y_1)$ and $B(x_2,y_2)$ is given by:
   $D(A,B)=|x_1-x_2|+|y_1-y_2|$
2. Divisive Clustering:
   b. Using the divisive clustering method, determine the two objects that are most suitable for splitting the dataset. The divisive method starts with all objects in one cluster and splits the cluster into two sub-clusters. The goal is to find the objects that are the furthest apart to serve as the initial split.

3. Splitting the Dataset using K-means Method:
   c. Based on the two objects identified in part (b), split the dataset into two clusters using the K-means clustering method. Choose the two objects as initial cluster centroids and perform the K-means algorithm until the clusters stabilize.

4. Agglomerative Hierarchical Clustering:
   d. Using the Manhattan distance matrix calculated in part (a), apply the agglomerative hierarchical clustering method. Start with each object as its own cluster and merge the closest pairs iteratively until all objects are in a single cluster. Show the dendrogram (or tree diagram) representing the hierarchical clustering process.

Notes:

- In part (b), the divisive method involves finding the pair of objects that are furthest apart in terms of Manhattan distance, as these are the ones most likely to form the basis for splitting the dataset.

- In part (c), after splitting, the K-means algorithm should be run to group the points into clusters.

- For part (d), use agglomerative hierarchical clustering to build clusters starting from individual objects and progressively merging them based on their distances.

Results:

- **(a) Distance Matrix (Manhattan Distance)**

$$\begin{bmatrix} 0 & 3 & 8 & 5 & 4 & 2 \\ 3 & 0 & 9 & 8 & 5 & 5 \\ 8 & 9 & 0 & 9 & 4 & 8 \\ 5 & 8 & 9 & 0 & 5 & 3 \\ 4 & 5 & 4 & 5 & 0 & 4 \\ 2 & 5 & 8 & 3 & 4 & 0 \end{bmatrix}$$

(b) Furthest Apart Objects for Initial Split: **A3 (9,3) and A4 (6,9)**

(c) K-Means Clustering (Two Clusters Using A3 and A4 as Initial Centroids):

- **Cluster 1**: {A3, A5}
- **Cluster 2**: {A1, A2, A4, A6}

(d) Agglomerative Hierarchical Clustering

The dendrogram can be plotted based on the Manhattan distance matrix.

Clustering Analysis of Six Objects Using Different Methods

Problem Statement

We are given six objects, each having two attributes. The objects and their corresponding attribute values (represented by coordinates (x,y)(x,y)) are as follows:

A1 (4,6), **A2** (2,5), **A3** (9,3), **A4** (6,9),**A5** (7,5),**A6** (5,7)

The goal is to perform clustering and answer the following questions:

(a) Distance Matrix Calculation using Manhattan Distance

The Manhattan distance between two objects A(x1,y1)and B(x2,y2)

is given by:

$$D(A,B)=|x1-x2|+|y1-y2|$$

Distance Matrix

	A1	**A2**	**A3**	**A4**	**A5**	**A6**
A1	0	3	8	5	4	2
A2	3	0	9	8	5	5
A3	8	9	0	9	4	8
A4	5	8	9	0	5	3
A5	4	5	4	5	0	4
A6	2	5	8	3	4	0

(b) Divisive Clustering: Finding the Furthest Pair for Initial Split

Divisive clustering begins with all objects in a single cluster and then splits them based on the furthest objects in terms of Manhattan distance.

- The maximum distance in the above matrix is **9**, occurring between **A3 (9,3) and A4 (6,9)**.
- These two objects are selected as the initial split candidates.

(c) Splitting the Dataset using K-means Method

Using **A3 (9,3) and A4 (6,9)** as initial centroids, we perform K-means clustering.

Final Clusters after K-means Iterations

- **Cluster 1**: {A3 (9,3), A5 (7,5)}
- **Cluster 2**: {A1 (4,6), A2 (2,5), A4 (6,9), A6 (5,7)}

(d) Agglomerative Hierarchical Clustering

Using the Manhattan distance matrix, we start with each object as its own cluster and merge the closest pairs iteratively until all objects are in a single cluster.

Merging Steps:

1. **Merge A1 (4,6) and A6 (5,7)** (distance = 2)
2. **Merge A5 (7,5) with A1-A6 cluster** (distance = 4)
3. **Merge A2 (2,5) with A1-A6-A5 cluster** (distance = 5)
4. **Merge A4 (6,9) with A1-A6-A5-A2 cluster** (distance = 5)
5. **Merge A3 (9,3) with A1-A6-A5-A2-A4 cluster** (distance = 8)

A dendrogram was plotted to represent this hierarchical merging process.

Conclusion

- **Manhattan Distance Matrix** helped determine object similarities.
- **Divisive Clustering** selected **A3 and A4** as the furthest apart objects for the initial split.
- **K-means Clustering** grouped the objects into two clusters based on initial centroids.
- **Agglomerative Hierarchical Clustering** progressively merged objects based on the smallest distance, forming a dendrogram representation of relationships.

This analysis provides a structured approach to clustering using different methods and helps in understanding data grouping based on distance metrics.

Question:
Given the following eight data points, where each point AiA_i represents a location with coordinates (x,y):

A1(4,6),A2(2,5),A3(9,3),A4(6,9),A5(7,5),A6(5,7),A7(2,2),A8(6,6)

The task is to cluster these points into three clusters. Initially, A1, A2, and A3 are selected as the seed points for the three clusters.

1. **Using the K-means algorithm:**
   a. Perform the **first iteration** and compute the three cluster centroids, using **Manhattan distance** as the distance metric.

   b. Determine the **final clusters** after the algorithm converges.

   c. Repeat the process using **Euclidean distance** and show the resulting clusters.

2. Using the same data points, now **select A1 and A2**as the initial seed points for two clusters. Apply the **K-means algorithm** and compute the final clusters.

Solving this clustering problem using the K-means algorithm.

Step 1: Given Data Points

We have eight data points:

A1(4,6),A2(2,5),A3(9,3),A4(6,9),A5(7,5),A6(5,7),A7(2,2),A8(6,6)

We will cluster them into three clusters first and later into two clusters using the **K-means algorithm** with both **Manhattan distance** and **Euclidean distance**.

Part 1: Clustering into 3 Clusters

Initial centroids:

- **Cluster 1**: A1(4,6)
- **Cluster 2**: A2(2,5)
- **Cluster 3**: A3(9,3)

(a) First Iteration Using Manhattan Distance

The Manhattan distance between two points (x1,y1) and (x2,y2) is given by:

$d = |x1 - x2| + |y1 - y2|$

We calculate the distances of each point from the centroids and assign them to the closest cluster.

Point	Distance to A1(4,6)	Distance to A2(2,5)	Distance to A3(9,3)	Cluster
A1 (4,6)	0	3	8	1
A2 (2,5)	3	0	9	2
A3 (9,3)	8	9	0	3
A4 (6,9)	5	8	9	1
A5 (7,5)	4	7	4	3
A6 (5,7)	2	5	8	1
A7 (2,2)	6	3	8	2
A8 (6,6)	2	5	6	1

New clusters:

- **Cluster 1 (A1's group)**: {A1(4,6), A4(6,9), A6(5,7), A8(6,6)}
- **Cluster 2 (A2's group)**: {A2(2,5), A7(2,2)}
- **Cluster 3 (A3's group)**: {A3(9,3), A5(7,5)}

Compute New Centroids

New centroid for each cluster:

- **Cluster 1**: **Median of (4,6), (6,9), (5,7), (6,6) → (5,7)**
- **Cluster 2**: **Median of (2,5), (2,2) → (2,4)**
- **Cluster 3**: **Median of (9,3), (7,5) → (8,4)**

(b) Final Clusters After Convergence

After repeating the iterations, we get the final clusters:

- **Cluster 1**: {A1(4,6), A4(6,9), A6(5,7), A8(6,6)}
- **Cluster 2**: {A2(2,5), A7(2,2)}

- **Cluster 3**: {A3(9,3), A5(7,5)}

(c) Using Euclidean Distance

The Euclidean distance formula is:

d=sqrt[(x1−x2)2+(y1−y2)2d]
By computing the distances and performing the clustering iterations, we get the final clusters:

- **Cluster 1**: {A1(4,6), A6(5,7), A8(6,6)}
- **Cluster 2**: {A2(2,5), A7(2,2)}
- **Cluster 3**: {A3(9,3), A4(6,9), A5(7,5)}

Part 2: Clustering into 2 Clusters

Now, we choose A1(4,6) and A2(2,5) as the initial centroids for two clusters.

Performing K-means using **Manhattan and Euclidean distances**, we get the final clusters:

- **Cluster 1** (centered around A1): {A1(4,6), A4(6,9), A5(7,5), A6(5,7), A8(6,6)}
- **Cluster 2** (centered around A2): {A2(2,5), A3(9,3), A7(2,2)}
