University of Mumbai

Data Warehousing and Mining

(Course Code : CSC504)

Semester V - Computer Engineering

Strictly as per the New Syllabus (REV-2019 'C' Scheme) of Mumbai University w.e.f. academic year 2021-2022

Dr. Nilesh M. Patil	Mrs. Anagha J. Patil
Ph.D. (Computer Engineering)	M.E. (Computer Engineering)
Department of Information Technology	Department of Information Technology
Fr. Conceicao Rodrigues College of Engineering	Vidyavardhini's College of Engineering & Technology
Bandra (West), Mumbai	Vasai Road (West), Palghar

TECH-NEO PUBLICATIONS

Where Authors Inspire Innovation

A Sachin Shah Venture

M5-56

Data Warehousing and Mining

(Course Code : CSC504)

(MU - Semester 5/ Computer Engineering

– For New Syll. 2021-2022)

Authors : Dr. Nilesh M. Patil, Mrs. Anagha J. Patil

First Edition for New Syllabus : July 2021

Tech-Neo ID : M5-56

Published by

Mrs. Nayana Shah, Mr. Sachin S. Shah

Managing Director, B. E (Industrial Electronics)

An Alumunus of IIM Ahmedabad

& Mr. Rahul S. Shah

Permanent Address

Tech-Neo Publications LLP

Sr. No. 38/1, Behind Pari Company, Khedekar Industrial Estate, Narhe, Maharashtra,

Pune-411041.

Email : info@techneobooks.com

Website : www.techneobooks.com

Printed at : Image Offset (Mr. Rahul Shah)

Dugane Ind. Area, Survey No. 28/25, Dhayari Near Pari Company, Pune - 411041. Maharashtra State, India.

E-mail : rahulshahimage@gmail.com

About Managing Director.... - Mr. Sachin Shah

Over 25 years of experience in Academic Publishing…

With over two and a half decades of experience in bringing out more than 1200 titles in Engineering, Polytechnic, Pharmacy, Computer Sciences and Information Technology,

Sachin Shah is a name synonymous with quality and innovative content.

A driven Educationalist…

1. A B.E. in Industrial Electronics (1992 Batch) from Bharati Vidyapeeth's College of Engineering, affiliated to University of Pune.
2. An Alumnus of IIM Ahmedabad.
3. A Co-Author of a bestselling book on "Engineering Mathematics" for Polytechnic Students of Maharashtra State.
4. Sachin has for over a decade, been working as a Consultant for Higher Education in USA and several other countries.

With path-breaking career…

A publishing career that started with handwritten cyclostyled notes back in 1992.

Sachin Shah has to his credit setting up and expansion of one of the leading companies in higher education publishing.

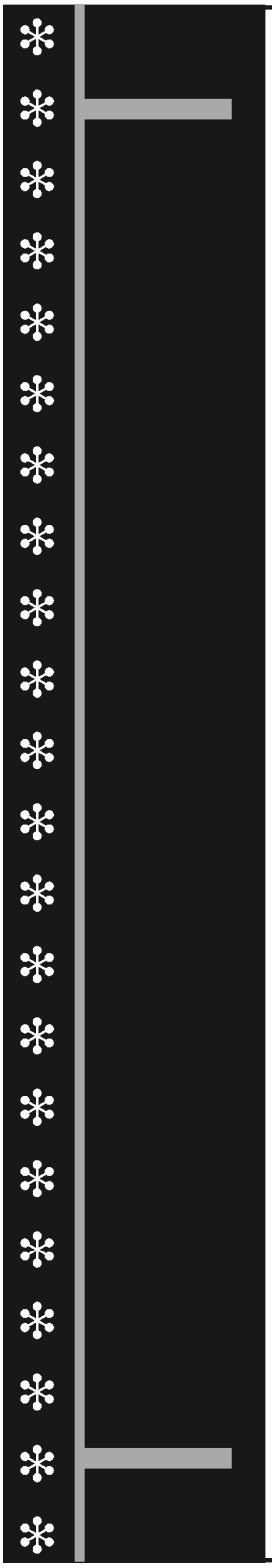
An experienced professional and an expert…

An energetic, creative & resourceful professional Sachin Shah's extensive experience of closely working with the best & the most eminent authors of Publishing Industry, ensures high standards of quality in contents.

This ability has helped students to attain better understanding and in-depth knowledge of the subject.

A visionary...

A gregarious person, **SACHIN SHAH** is a thought leader who has been simplifying the methods of learning and bridging the gap between the best authors in the publishing industry and the student community for decades.



Dedicated to .....

The Readers of this Book

- Authors

Preface

It delights us to write this book on **" Data Warehousing and Mining"** for the students of Mumbai University. This book has been strictly written as per the prescribed curriculum

Every chapter of the book corresponds to the respective module mentioned in the syllabus. The topics within the chapters have been arranged in a proper sequence to ensure smooth flow of the subject.

We are thankful to **Mr. Sachin Shah, Managing Director of Tech Neo Publications** for his encouragement and support.

We are also thankful to the staff of Tech-Neo Publications for their timely efforts in the making of this book. We, together have taken every possible care to eliminate errors in the book. If they still exist, kindly let us know.

We are also thankful to our family, friends, colleagues, and students.

- Dr. Nilesh M. Patil

Mrs. Anagha J. Patil

Syllabus…

Mumbai University

B. E. (Computer Engineering)

Course Code	Course Name	Credit
CSC504	Data Warehousing and Mining	3

Prerequisite: Database Concepts	
Course Objectives :	
1.	To identify the significance of Data Warehousing and Mining.
2.	To analyze data, choose relevant models and algorithms for respective applications.
3.	To study web data mining.
4.	To develop research interest towards advances in data mining.
Course Outcomes : At the end of the course, the student will be able to	
1.	Understand data warehouse fundamentals and design data warehouse with dimensional modelling and apply OLAP operations.
2.	Understand data mining principles and perform Data preprocessing and Visualization.
3.	Identify appropriate data mining algorithms to solve real world problems.
4.	Compare and evaluate different data mining techniques like classification, prediction, clustering and association rule mining
5.	Describe complex information and social networks with respect to web mining.

Module	Contents	Hrs.
1	**Data Warehousing Fundamentals**	**8**
	Introduction to Data Warehouse, Data warehouse architecture, Data warehouse versus Data Marts, E-R Modeling versus Dimensional Modeling, Information Package Diagram, Data Warehouse Schemas; Star Schema, Snowflake Schema, Factless Fact Table, Fact Constellation Schema. Update to the dimension tables. Major steps in ETL process, OLTP versus OLAP, OLAP operations: Slice, Dice, Rollup, Drilldown and Pivot. **(Refer Chapter 1)**	
2	**Introduction to Data Mining, Data Exploration and Data Pre-processing**	**8**
	Data Mining Task Primitives, Architecture, KDD process, Issues in Data Mining, Applications of Data Mining, Data Exploration: Types of Attributes, Statistical Description of Data, Data Visualization, Data Preprocessing: Descriptive data summarization, Cleaning, Integration & transformation, Data reduction, Data Discretization and Concept hierarchy generation. **(Refer Chapter 2)**	

Module	Contents	Hrs.
3	**Classification**	6
	Basic Concepts, Decision Tree Induction, Naïve Bayesian Classification, Accuracy and Error measures, Evaluating the Accuracy of a Classifier: Holdout & Random Subsampling, Cross Validation, Bootstrap. **(Refer Chapter 3)**	
4	**Clustering**	6
	Types of data in Cluster analysis, Partitioning Methods (k-Means, k-Medoids), Hierarchical Methods (Agglomerative, Divisive). **(Refer Chapter 4)**	
5	**Mining frequent patterns and associations**	6
	Market Basket Analysis, Frequent Item sets, Closed Item sets, and Association Rule, Frequent Pattern Mining, Apriori Algorithm, Association Rule Generation, Improving the Efficiency of Apriori, Mining Frequent Itemsets without candidate generation, Introduction to Mining Multilevel Association Rules and Mining Multidimensional Association Rules. **(Refer Chapter 5)**	
6	**Web Mining**	5
	Introduction, Web Content Mining: Crawlers, Harvest System, Virtual Web View, Personalization, Web Structure Mining: Page Rank, Clever, Web Usage Mining. **(Refer Chapter 6)**	

Lab Syllabus…

Lab Code	Lab Name	Credit
CSL503	Data Warehousing and Mining Lab	1

Prerequisite: Database Concepts	
Lab Objectives :	
1.	Learn how to build a data warehouse and query it.
2.	Learn about the data sets and data preprocessing.
3.	Demonstrate the working of algorithms for data mining tasks such Classification, clustering, Association rule mining & Web mining
4.	Apply the data mining techniques with varied input values for different parameters.
5.	Explore open source software (like WEKA) to perform data mining tasks.
Lab Outcomes : At the end of the course, the student will be able to	
1.	Design data warehouse and perform various OLAP operations.
2.	Implement data mining algorithms like classification.
3.	Implement clustering algorithms on a given set of data sample.
4.	Implement Association rule mining & web mining algorithm.

Suggested List of Experiments	
Sr. No.	**Title of Experiment**
1	One case study on building Data warehouse/Data Mart Write Detailed Problem statement and design dimensional modelling (creation of star and snowflake schema)
2	Implementation of all dimension table and fact table based on experiment 1 case study
3	Implementation of OLAP operations: Slice, Dice, Rollup, Drilldown and Pivot based on experiment 1 case study
4	Implementation of Bayesian algorithm
5	Implementation of Data Discretization (any one) & Visualization (any one)
6	Perform data Pre-processing task and demonstrate Classification, Clustering, Association algorithm on data sets using data mining tool (WEKA/R tool)
7	Implementation of Clustering algorithm (K-means/K-medoids)
8	Implementation of any one Hierarchical Clustering method
9	Implementation of Association Rule Mining algorithm (Apriori)
10	Implementation of Page rank/HITS algorithm

Term Work:	
1	Term work should consist of 10 experiments.
2	Journal must include at least 1 assignment on content of theory and practical of "Data Warehousing and Mining"
3	The final certification and acceptance of term work ensures that satisfactory performance of laboratory work and minimum passing marks in term work.
4	Total 25 Marks (Experiments: 15-marks, Attendance (Theory & Practical): 05-marks, Assignments: 05-marks)
Oral & Practical exam	
	Based on the entire syllabus of CSC504 : Data Warehousing and Mining

Index

❑❑❑

MODULE 1

CHAPTER 1

Data Warehousing Fundamentals

University Prescribed Syllabus w.e.f Academic Year 2021-2022

Introduction to Data Warehouse, Data warehouse architecture, Data warehouse versus Data Marts, E-R Modeling versus Dimensional Modeling, Information Package Diagram, Data Warehouse Schemas; Star Schema, Snowflake Schema, Factless Fact Table, Fact Constellation Schema. Update to the dimension tables. Major steps in ETL process, OLTP versus OLAP, OLAP operations: Slice, Dice, Rollup, Drilldown and Pivot.

1.1 INTRODUCTION TO DATA WAREHOUSE

- Data Warehouse is a relational database management system (RDBMS) constructed to meet the requirement of transaction processing systems.
- It can be loosely described as any centralized data repository which can be queried for business benefits.
- It is a database that stores information oriented to satisfy decision-making requests. It is a group of decision support technologies that targets to enabling the knowledge worker (executive, manager, and analyst) to make superior and higher decisions.
- So, data warehousing support architectures and tool for business executives to systematically organize, understand and use their information to make strategic decisions.
- It includes historical data derived from transaction data from single and multiple sources.
- According to William H. Inmon, **"A data warehouse is a subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management's decision making process."**
- The four keywords subject-oriented, integrated, time-variant and non-volatile distinguish data warehouses from other data repository systems, such as relational database systems, transaction processing systems, and file systems.

1.1.1 Features of Data Warehouse

UQ. Why is data integration required in a data warehouse more so than in an operational application? **MU - Dec. 2019**

The key features of a data warehouse are discussed below :

1. Subject-Oriented

- A data warehouse target on the modeling and analysis of data for decision-makers.
- Therefore, data warehouses typically provide a concise and straightforward view around a particular subject, such as customer, product, or sales, instead of the global organization's ongoing operations.
- This is done by excluding data that are not useful concerning the subject and including all data needed by the users to understand the subject.

2. Integrated

- A data warehouse integrates various heterogeneous data sources like RDBMS, flat files, and online transaction records.
- It requires performing data cleaning and integration during data warehousing to ensure consistency in naming conventions, encoding structures, attributes measures, etc., among different data sources.
- Operational back-ends do have integrations with other systems but generally speaking the integration breadth compared to data warehouse is much smaller because of the limited scope in particular with micro services.
- In contrast the scope of a data warehouse is very wide and encompasses practically all important operational systems. Thus data warehouse systems ingest data from practically all important operational systems to power analytics with a broad and complete picture of all enterprise data plus other data sources beyond the scope of the enterprise.
- Ultimately a lot of heterogeneous data needs to be integrated and combined. Data historization in the data warehouse requires surrogate keys or artificial hash keys.
- Data across separate sources needs to be aligned and harmonized, and standardized. The need for data cleansing and data quality control is significant. **This is why integration work and scope in data warehousing is higher than in operational systems**.

3. Time-variant

- Historical information is kept in a data warehouse. For example, one can retrieve files from 3 months, 6 months, 12 months, or even previous data from a data warehouse.
- These varies with a transactions system, where often only the most current file is kept.

- Every key structure in the data warehouse contains, either implicitly or explicitly, a time element.

4. Non-volatile

- The data warehouse is a physically separate data storage, which is transformed from the source operational RDBMS.
- The operational updates of data do not occur in the data warehouse, i.e., update, insert, and delete operations are not performed.
- It usually requires only two procedures in data accessing: initial loading of data and access to data.
- Therefore, the data warehouse does not require transaction processing, recovery, and concurrency capabilities, which allows for substantial speedup of data retrieval.
- Non-Volatile defines that once entered into the warehouse, and data should not change.

1.1.2 Need of Data Warehouse

Data Warehouse is needed for the following reasons:

1. **Business User :** Business users require a data warehouse to view summarized data from the past. Since these people are non-technical, the data may be presented to them in an elementary form.
2. **Store historical data :** Data Warehouse is required to store the time variable data from the past. This input is made to be used for various purposes.
3. **Make strategic decisions :** Some strategies may be depending upon the data in the data warehouse. So, data warehouse contributes to making strategic decisions.
4. **For data consistency and quality:** Bringing the data from different sources at a commonplace, the user can effectively undertake to bring the uniformity and consistency in data.
5. **High response time :** Data warehouse has to be ready for somewhat unexpected loads and types of queries, which demands a significant degree of flexibility and quick response time.

1.1.3 Applications of Data Warehouse

Here, are most common sectors where data warehouse is used :

1. **Airline :** In the Airline system, it is used for operation purpose like crew assignment, analyses of route profitability, frequent flyer program promotions, etc.
2. **Banking :** It is widely used in the banking sector to manage the resources available on desk effectively. Few banks also use for the market research, performance analysis of the product and operations.
3. **Healthcare :** Healthcare sector also use data warehouse to strategize and predict outcomes, generate patient's treatment reports, share data with tie-in insurance companies, medical aid services, etc.
4. **Public sector :** In the public sector, data warehouse is used for intelligence gathering. It helps government agencies to maintain and analyze tax records, health policy records, for every individual.
5. **Investment and Insurance sector :** In this sector, the warehouses are primarily used to analyze data patterns, customer trends, and to track market movements.
6. **Retain chain :** In retail chains, data warehouse is widely used for distribution and marketing. It also helps to track items, customer buying pattern, promotions and also used for determining pricing policy.
7. **Telecommunication :** A data warehouse is used in this sector for product promotions, sales decisions and to make distribution decisions.
8. **Hospitality Industry :** This Industry utilizes warehouse services to design as well as estimate their advertising and promotion campaigns where they want to target clients based on their feedback and travel patterns.

1.1.4 Benefits of Data Warehouse

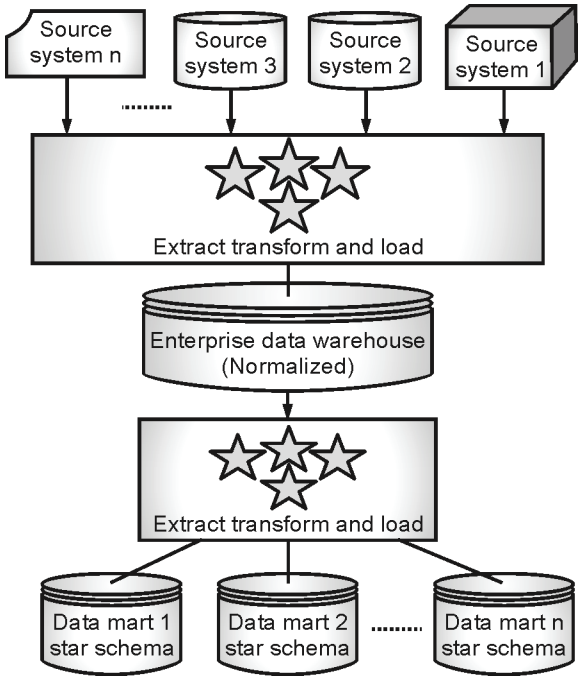
1. **Delivers enhanced business intelligence :** By having access to information from various sources from a single platform, decision makers will no longer need to rely on limited data or their instinct. Additionally, data warehouses can effortlessly be applied to a business's processes, for instance, market segmentation, sales, risk, inventory, and financial management.

2. **Saves times :** A data warehouse standardizes, preserves, and stores data from distinct sources, aiding the consolidation and integration of all the data. Since critical data is available to all users, it allows them to make informed decisions on key aspects. In addition, executives can query the data themselves with little to no IT support, saving more time and money.
3. **Enhances data quality and consistency :** A data warehouse converts data from multiple sources into a consistent format. Since the data from across the organization is standardized, each department will produce results that are consistent. This will lead to more accurate data, which will become the basis for solid decisions.
4. **Generates a high Return on Investment (ROI) :** Companies experience higher revenues and cost savings than those that haven't invested in a data warehouse.
5. **Provides competitive advantage :** Data warehouses help get a holistic view of their current standing and evaluate opportunities and risks, thus providing companies with a competitive advantage.
6. **Improves the decision-making process :** Data warehousing provides better insights to decision makers by maintaining a cohesive database of current and historical data. By transforming data into purposeful information, decision makers can perform more functional, precise, and reliable analysis and create more useful reports with ease.
7. **Enables organizations to forecast with confidence :** Data professionals can analyze business data to make market forecasts, identify potential KPIs, and gauge predicated results, allowing key personnel to plan accordingly.
8. **Streamlines the flow of information :** Data warehousing facilitates the flow of information through a network connecting all related or non-related parties.

1.1.5 Approaches to Build Data Warehouse

1. Top-Down Approach

- This is the big-picture approach to building the overall, massive, enterprise-wide data warehouse.
- There is no collection of information sources here.
- The data warehouse is large and well-integrated.
- This approach, on the other hand, would take longer to build and has a higher failure rate.
- This approach could be dangerous if you do not have experienced professionals on your team.
- It will also be difficult to sell this approach to senior management and sponsors.
- They are unlikely to see results soon enough.



(1A1)**Fig. 1.1.1 : Top-Down Approach of Data Warehouse Design**

☞ **Advantages**

(1) Represents a data view from the perspective of the enterprise.

(2) Inherently designed—not a mash-up of disparate data marts.

(3) Data about the content is stored in a single, central location.

(4) Centralized control and rules.

☞ **Disadvantages**

(1) Even with an iterative strategy, building takes longer.

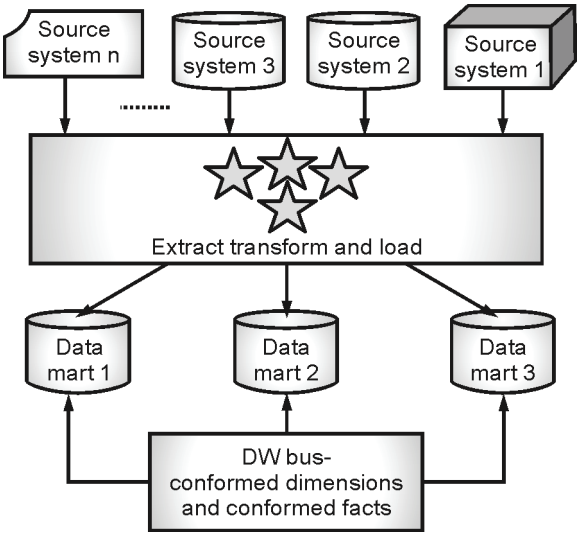
(2) High failure risk/exposure

(3) Requires a high level of cross-functional expertise

(4) Expenses are high without proof of concept.

2. Bottom-Up Approach

- You create departmental data marts one by one using this bottom-up method.
- To figure out which data marts to build first, you'd create a priority list.
- The most serious disadvantage of this method is data fragmentation.
- Each data mart will be blind to the organization's overarching requirements.



(1A2)**Fig. 1.1.2: Bottom-Up Approach of Data Warehouse Design**

☞ **Advantages**

(1) Implementation of small portions is faster and easier.

(2) Favourable return on investment and proof of concept.

(3) There is a lower chance of failure.

(4) Inherently incremental; significant data marts can be scheduled first.

(5) Allows the project team to grow and develop.

☞ **Disadvantages**

(1) Each data mart has its own skewed perspective on information.

(2) Every data mart is flooded with redundant information.

(3) Perpetuates data that is inconsistent and irreconcilable.

(4) Increases the number of unmanageable interfaces.

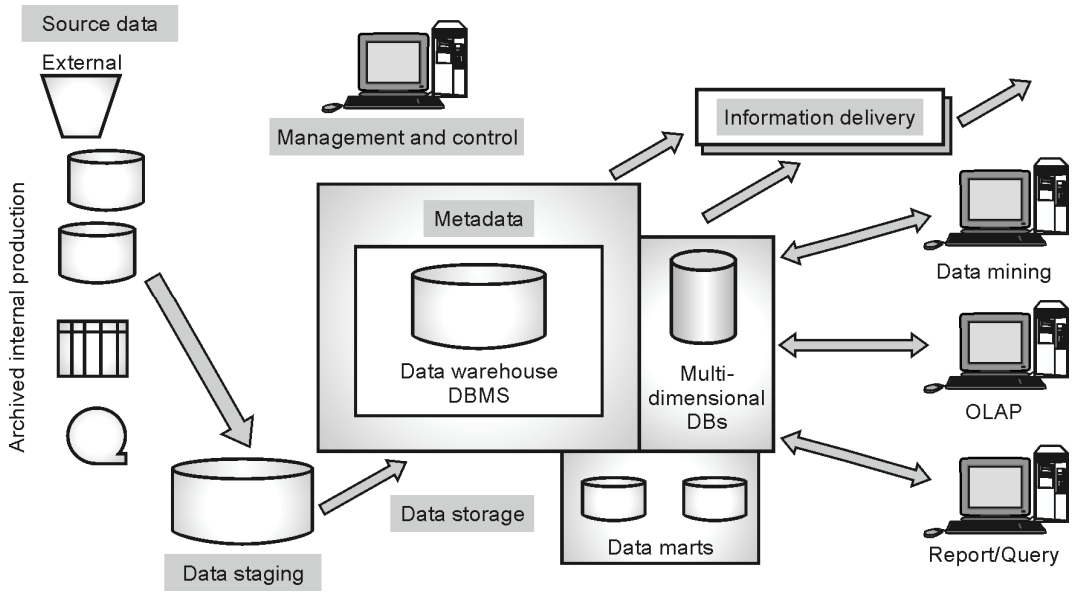
⏭ 1.2 DATA WAREHOUSE ARCHITECTURE

UQ. Consider Metadata as an equivalent of Amazon book store, where each data element is book. What this metadata will contain? Explain.

MU - June 2021

- We put together numerous components to make up an operational system like order entry, claims processing, or a savings account. The GUI (graphical user interface) is used to interact with users for data entry in the front-end component. The database management system, such as Oracle, Informix, or Microsoft SQL Server, is part of the data storage component. The display component consists of the user's screens and reports. The connectivity component consists of data interfaces and network software. We arrange these components in the most efficient manner possible, based on the information requirements and the structure of our firm.
- Architecture is the proper arrangement of the components. You put together a data warehouse using software and hardware. You organize these building blocks in a specific way to meet the needs of your company for optimal benefit.

NOTES



(1A3)**Fig. 1.2.1: Data Warehouse Architecture**

1.2.1 Source Data Component

Source data coming into the data warehouse may be grouped into four broad categories, as discussed here.

(1) **Production data**: This type of information originates from the company's numerous operational systems. You select segments of data from various operational systems based on the information requirements in the data warehouse.

(2) **Internal data**: Users preserve their "private" spreadsheets, documents, customer profiles, and sometimes even departmental databases in every company. This is internal data that could be used in a data warehouse in some form.

(3) **Archived data**: Operational systems are largely designed to run a company's existing operations. Every operational system takes outdated data and stores it in archived files on a regular basis. After a year, certain data is archived. Data is sometimes kept in operational system databases for up to five years.

(4) **External data**: Most executives rely on data from outside sources for a large portion of the information they use. They rely on statistics about their industry compiled by third-party organizations. They rely on competitor market share data. They use standard values of financial indicators to evaluate their company's performance.

1.2.2 Data Staging Component

- You must prepare the data for storage in the data warehouse after extracting it from various operational systems and external sources.
- The extracted data from various sources must be changed, converted, and prepared in a format suitable for storage for querying and analysis.
- To get the data ready, three major functions must be completed. You must first extract the data, then transform it and at last load it into the data warehouse storage. A staging area is where the three major functions of extraction, transformation, and loading take place.
- Data staging is a place with a set of functions that clean, change, combine, convert, deduplicate, and prepare source data for storage and use in the data warehouse.

1.2.3 Data Storage Component

- The data warehouse's data storage is kept in a separate repository.
- Large volumes of historical data must be kept in a data warehouse's data repository for analysis.
- Furthermore, the data in the data warehouse must be kept in structures suitable for analysis rather than for

quick retrieval of individual pieces of information.

- As a result, the data warehouse's data storage is kept separate from the data storage for operational systems.

1.2.4 Information Delivery Component

- The information delivery component includes various methods of information delivery in order to provide information to the large community of data warehouse users.
- Information can be delivered in the form of ad hoc reports, complex queries, multidimensional (MD) analysis, statistical analysis, enterprise information system (EIS) feeds, etc.
- This information can be delivered through online, intranet, internet or email mode.

1.2.5 Metadata Component

- Metadata is "data that describes other data".
- Metadata in a data warehouse is similar to the data dictionary or the data catalog in a database management system.
- In the data dictionary, you keep the information about the logical data structures, the information about the files and addresses, the information about the indexes, and so on. The data dictionary contains data about the data in the database.
- Similarly, the metadata component is the data about the data in the data warehouse.

1.2.5(A) Types of Metadata

Metadata in a data warehouse fall into three major parts:

(a) Operational Metadata

- As we know, data for the data warehouse comes from various operational systems of the enterprise. These source systems include different data structures. The data elements selected for the data warehouse have various fields lengths and data types.
- In selecting information from the source systems for the data warehouses, we divide records, combine factor of documents from different source files, and deal with multiple coding schemes and field lengths. When we deliver information to the end-users, we must be able to tie that back to the source data sets. Operational metadata contains all of this information about the operational data sources.

(b) Extraction and Transformation Metadata

- Extraction and transformation metadata include data about the removal of data from the source systems, namely, the extraction frequencies, extraction methods, and business rules for the data extraction.
- Also, this category of metadata contains information about all the data transformation that takes place in the data staging area.

(c) End-User Metadata

- The end-user metadata is the navigational map of the data warehouses. It enables the end-users to find data from the data warehouses.
- The end-user metadata allows the end-users to use their business terminology and look for the information in those ways in which they usually think of the business.

1.2.5(B) Examples of Metadata

1. A **library catalog** may be considered metadata. The directory metadata consists of several predefined components representing specific attributes of a resource, and each item can have one or more values. These components could be the name of the author, the name of the document, the publisher's name, the publication date, and the methods to which it belongs.
2. The table of content and the index in a book may be treated as metadata for **the book**.
3. Suppose we say that a data item about **a person** is 80. This must be defined by noting that it is the person's weight and the unit is kilograms. Therefore, (weight, kilograms) is the metadata about the data is 80.
4. Another examples of metadata are data about the tables and figures in **a report** like this book. A table (which is a record) has a name (e.g., table titles), and there are column names of the tables that may be treated metadata. The figures also have titles or names.
5. Metadata for **a web page** may contain the language it is coded in, the tools used to build it, supporting browsers, etc.

6. Metadata for **a digital image** may contain the size of the picture, resolution, color intensity, image creation date, etc.
7. Metadata for **a document** may contain the document created date, last modified date, it's size, author, description, etc.

UQ. What is Metadata? Why do we need metadata when search engines like Google seem so effective? **MU - May 2019**

- Metadata describes unseen HTML elements that directly communicate and clarify website information for search engines, playing a critical role in effective Search Engine Optimization for retailers. This series of micro-communications includes page titles, description tags and other protocols, and they may describe purposes, characteristics and general content.
- Metadata is a structured way to communicate information about a data set, which is used in a variety of settings with special relevance for ecommerce businesses.

Metadata for SEO and social media

- Metadata allows XML-based applications to categorize and contextualize pieces of data. For marketers, this data is usually a web pages. A search engine's job is to crawl a web page and interpret its relevancy to a given search query. While keywords matching within body content and backlinks to the page play a large role in determining ranking, metadata says more about the purpose of a page. Search engines can crawl a website and guess its general purpose based on these elements; metadata enables webmasters to tell search engines what a page's title is, which says a lot about what search queries it may be relevant for.
- Metadata is used in similar fashion by social media platforms such as Facebook. Open Graph (OG) protocol marks up web pages with information that is then displayed when a web page is shared on Facebook.

Common metadata elements

- **Meta tags** include basic keywords, description tags that summarize content, and robots that index pages or pass on link authority.
- **Title tags** are an important search engine ranking factor and should include most relevant keywords, product(s) and article name.
- **Image tags** identify URLs, while alternative attributes provide related text, measurements and some SEO signals.
- **Canonical tags** are used to consolidate similar pages and attribute the value to only one, reducing the likelihood of duplicate content penalty and providing a straightforward user experience.
- **Structured data** denote aspects of content that can help promote them within search results like Google answers and maps.

1.2.6 Management and Control Component

- This component of the data warehouse architecture sits on top of all the other components.
- The management and control component coordinates the services and activities within the data warehouse.
- This component controls the data transformation and the data transfer into the data warehouse storage. On the other hand, it moderates the information delivery to the users.
- It works with the database management systems and enables data to be properly stored in the repositories.
- It monitors the movement of data into the staging area and from there into the data warehouse storage itself.
- The management and control component interacts with the metadata component to perform the management and control functions.
- As the metadata component contains information about the data warehouse itself, the metadata is the source of information for the management module.

1.3 DATA WAREHOUSE Vs DATA MARTS

- A data mart is a small, single-subject data warehouse subset that provides decision support to a small group of people.
- Data Marts can serve as a test vehicle for companies exploring the potential benefits of Data Warehouses.

- Data Marts address local or departmental problems, while a Data Warehouse involves a company-wide effort to support decision making at all levels in the organization.

Table 1.3.1: Data Warehouse Vs Data Mart

Sr. No.	Data Warehouse	Data Mart
1.	Data warehouse is a centralized system.	Data mart is a decentralized system.
2.	In data warehouse, lightly denormalization takes place.	In Data mart, highly denormalization takes place.
3.	Data warehouse is top-down model.	Data mart is a bottom-up model.
4.	To build a warehouse is difficult.	To build a mart is easy.
5.	In data warehouse, Fact constellation schema is used.	In data mart, Star schema and snowflake schema are used.
6.	Data warehouse is flexible.	Data mart is not flexible.
7.	Data warehouse is the data-oriented in nature.	Data mart is the project-oriented in nature.
8.	Data warehouse has long life.	Data-mart has short life than warehouse.
9.	In data warehouse, data are contained in detail form.	In data mart , data are contained in summarized form.
10.	Data Warehouse is vast in size.	Data mart is smaller than warehouse.

1.4 E-R MODELLING Vs DIMENSIONAL MODELLING

UQ. Why is entity-relationship modeling technique not suitable for the data warehouse? How is dimensional modeling different? **MU – Dec. 2019**

- Dimensional Modeling (DM) is a data structure technique optimized for data storage in a Data warehouse. The purpose of dimensional modeling is to optimize the database for faster retrieval of data. The concept of Dimensional Modelling was developed by Ralph Kimball and consists of "fact" and "dimension" tables. Dimensional table records information on each dimension, and fact table records all the "fact", or measures.
- A dimensional model in data warehouse is designed to read, summarize, analyze numeric information like values, balances, counts, weights, etc. in a data warehouse. In contrast, relation models are optimized for addition, updating and deletion of data in a real-time Online Transaction System.
- These dimensional and relational models have their unique way of data storage that has specific advantages.
- For instance, in the relational model, normalization and ER models reduce redundancy in data. On the contrary, dimensional model in data warehouse arranges data in such a way that it is easier to retrieve information and generate reports.
- Hence, Dimensional models are used in data warehouse systems and not a good fit for relational systems.

1.4.1 Elements of Dimensional Data Model

1. **Fact :** Facts are the measurements/metrics or facts from your business process. For a Sales business process, a measurement would be quarterly sales number.
2. **Dimension :** Dimension provides the context surrounding a business process event. In simple terms, they give who, what, where of a fact. In the Sales business process, for the fact quarterly sales number, dimensions would be:
   - Who – Customer Names
   - Where – Location
   - What – Product Name

   In other words, a dimension is a window to view information in the facts.
3. **Attributes :** The Attributes are the various characteristics of the dimension in dimensional data modeling. In the Location dimension, the attributes can be
   - State
   - Country
   - Zipcode, etc.

Attributes are used to search, filter, or classify facts. Dimension Tables contain Attributes

4. **Fact Table :** A fact table is a primary table in dimension modelling. A Fact Table contains
   - Measurements/facts
   - Foreign key to dimension table
5. **Dimension Table**
   - A dimension table contains dimensions of a fact.
   - They are joined to fact table via a foreign key.
   - Dimension tables are de-normalized tables.
   - The dimension attributes are the various columns in a dimension table.
   - Dimensions offers descriptive characteristics of the facts with the help of their attributes.
   - No limit set for given for number of dimensions.
   - The dimension can also contain one or more hierarchical relationships

1.4.2 Steps of Dimensional Modelling

The accuracy in creating your dimensional modeling determines the success of your data warehouse implementation. The model should describe the Why, How much, When/Where/Who and What of your business process. Here are the steps to create dimension model.

1. Identify Business Process

- Identifying the actual business process, a data warehouse should cover. This could be Marketing, Sales, HR, etc. as per the data analysis needs of the organization. The selection of the business process also depends on the quality of data available for that process. It is the most important step of the data modelling process, and a failure here would have cascading and irreparable defects.
- To describe the business process, you can use plain text or use basic Business Process Modelling Notation (BPMN) or Unified Modelling Language (UML).

2. Identify the Grain

- The grain describes the level of detail for the business problem/solution. It is the process of identifying the lowest level of information for any table in your data warehouse. If a table contains sales data for every day, then it should be daily granularity. If a table contains total sales data for each month, then it has monthly granularity.
- During this stage, you answer questions like -
  1. Do we need to store all the available products or just a few types of products? This decision is based on the business processes selected for data warehouse.
  2. Do we store the product sale information on a monthly, weekly, daily or hourly basis? This decision depends on the nature of reports requested by executives.
  3. How do the above two choices affect the database size?
- Example of Grain: The CEO at an MNC wants to find the sales for specific products in different locations on a daily basis. So, the grain is "product sale information by location by the day."

3. Identify Dimensions and Attributes

- Dimensions are nouns like date, store, inventory, etc. These dimensions are where all the data should be stored. For example, the date dimension may contain data like a year, month and weekday.
- Example of Dimensions: The CEO at an MNC wants to find the sales for specific products in different locations on a daily basis.

  Dimensions : Product, Location and Time

  Attributes : For Product: Product key (Foreign Key), Name, Type, Specifications

  Hierarchies : For Location: Country, State, City, Street Address, Name

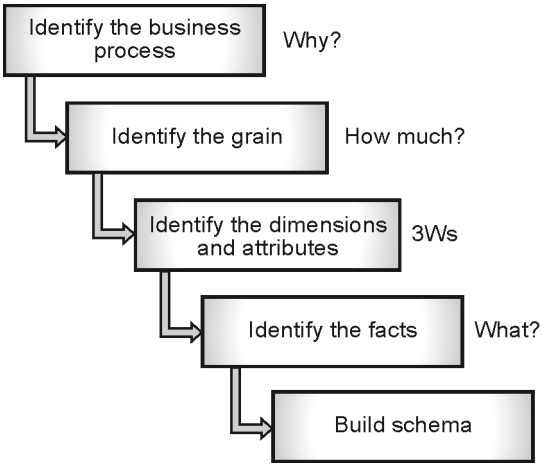
4. Identify Facts

- This step is co-associated with the business users of the system because this is where they get access to data stored in the data warehouse. Most of the fact table rows are numerical values like price or cost per unit, etc.
- Example of Facts**:** The CEO at an MNC wants to find the sales for specific products in different locations on a daily basis. The fact here is Sum of Sales by product by location by time.

5. Build Schema

- In this step, you implement the Dimension Model. A schema is nothing but the database structure (arrangement of tables). There are two popular schemas

1. Star Schema 2. Snowflake Schema



(1A4)**Fig. 1.4.1: Steps in Dimensional Modeling**

Table 1.4.1: E-R Modeling Vs Dimensional Modeling

Sr. No.	E-R Modeling	Dimensional Modeling
1.	E-R modeling have logical and physical model.	Dimensional modeling have physical model.
2.	E-R modeling is used for normalizing the OLTP design.	Dimensional modeling is used for de-normalizing the OLAP design.
3.	E-R modelling revolves around the entities and their relationships to capture the overall process of the system.	Dimensional modelling revolves around dimensions(point of analysis) for decision making and not to capture the process.
4.	In E-R modeling the data is in normalized form, so more number of joins are required, which may adversely affect the system performance.	In dimensional modeling the data is de-normalized, so less number of joins are required, by which system performance will improve.
5.	In E-R modeling, a view of data is from data processing.	In dimensional modeling, a view of data is from business processing.
6.	It is not mapped for creating schemas.	It is mapped for creating schemas.
7.	It uses the current data.	It uses the historical data.
8.	Size of data varies from MBs to GBs.	Size of data varies from GBs to TBs.
9.	Data storage is volatile.	Data storage is non-volatile.
10.	High Create/Read/Update/Delete (CRUD) activity.	High Select activity.
11.	**Advantages:** Removes data redundancy. Ensures data consistency. Expresses the relationship between the entities.	**Advantages:** Captures critical measures. Views along dimensions. Useful to business users.

1.5 DATA REPLICATION

UQ. What is the relationship between data warehousing and data replication? Which form of replication (synchronous or asynchronous) is better suited for data warehousing? Why? Explain with appropriate example. **MU – May 2019**

- Database replication is the process of copying or transferring data from a database on one server to a database on another server to improve data availability and accessibility. The process facilitates data sharing and data recovery.
- Database replication is performed in order to provide a duplication of the data environment in event of catastrophe. Additionally, recovery in the event of failure is fast, accurate and cost-effective.
- Data replication also enables accurate sharing of information so that all users have access to consistent data in real time.

- There are various methods of traditional data replication.

1. Synchronous replication
2. Asynchronous replication
3. Semi-synchronous replication

1. Synchronous Replication

- With synchronous replication, when a disk I/O is performed by the application or by the file system cache on the primary server, program waits for the I/O acknowledgement from the local disk and from the secondary server, before sending the I/O acknowledgement to the application or to the file system cache.
- This mechanism is essential for failover of transactional applications when they commit their transactions.

2. Asynchronous Replication

- With asynchronous replication, the I/O's are placed in a queue on the primary server but the primary server does not wait for the I/O acknowledgments of the secondary server.
- So, all data that did not have time to be copied across the network on the secondary server is lost if the primary server fails. In particular, a transactional application loses committed transactions in case of failure.

3. Semi-synchronous Replication

- With semi-synchronous replication, program always waits for the acknowledgement of the two servers before sending the acknowledgement to the application or the file system cache.
- But in the semi-synchronous case, the secondary sends the acknowledgement to the primary upon receipt of the I/O and writes to disk after. In the synchronous case, the secondary writes the I/O to disk and then sends the acknowledgement to the primary.
- With asynchronous replication, there is data loss on failure. Even with the semi-synchronous replication, there is data loss in the special case of a simultaneous double power outage of both servers, with inability to restart on the former primary server and the requirement to restart on the secondary server. So be very careful when choosing synchronous replication vs asynchronous replication. **Always prefer a synchronous or a semi-synchronous replication for a critical application**.

1.6 INFORMATION PACKAGE DIAGRAM

- The first and most generalized level of an information model is its information package diagram.
- This model focuses on the data gathering activities for the users' information packaging requirements.
- An information package diagram defines the relationships between subject matter and key performance measures.
- The information package diagram has a highly targeted purpose, providing a focused scope for user requirements.
- Because information package diagrams target what the users want, they are effective in facilitating communication between the technical staff and the users, indicating any inconsistencies between the requirements and what the data warehouse will deliver.
- **Example :** Information package for analyzing sales for a certain business. It allows users to evaluate sales metrics by time, product, location, and customer demographics.

	Dimensions					
Hierarchies	Time Periods	Locations	Products	Age Groups	…	…
	Year	Country	Class	Group 1		
	Facts: Forecast Sales, Budget Sales, Actual Sales					

- The subject here is sales. The measured facts or the measurements that are of interest for analysis are shown in the bottom section of the package diagram. In this case, the measurements are actual sales, forecast sales, and budget sales. The business dimensions along which these measurements are to be analyzed are shown at the top of diagram as column headings. In our example, these dimensions are time, location, product, and demographic age group. Each of these business dimensions contains a hierarchy or levels. For

example, the time dimension has the hierarchy going from year down to the level of individual day. The other intermediary levels in the time dimension could be quarter, month, and week. These levels or hierarchical components are shown in the information package diagram.

- Information Package Diagram enables you to
  a. Define the common subject areas
  b. Design key business metrics
  c. Decide how data must be presented
  d. Determine how users will aggregate or roll up
  e. Decide the data quantity for user analysis or query
  f. Decide how data will be accessed
  g. Establish data granularity
  h. Estimate data warehouse size
  i. Determine the frequency for data refreshing
  j. Ascertain how information must be packaged

1.7 DATA WAREHOUSE SCHEMAS

UQ. Suppose that a data warehouse for DB-University consists of the following four dimensions: student, course, semester, and instructor, and two measures count and avg_grade. When at the lowest conceptual level (e.g., for a given student, course, semester, and instructor combination), the avg_grade measure stores the actual course grade of the student. At higher conceptual levels, avg_grade stores the average grade for the given combination.

(a) Draw a snowflake schema diagram for the data warehouse.

(b) Starting with the base cuboid [student, course, semester, instructor], what specific OLAP operations

(e.g., roll-up from semester to year) should one perform in order to list the average grade of CS courses for each DB_University student.

MU - May 2019

UQ. Consider a data warehouse for a hospital where there are three dimensions namely (a) Doctor (b) Patient (c) Time and two measures (i) count (ii) charge where charge is the fee that the doctor charges a patient for a visit.

(i) Draw star and snowflake schema.

(ii) Starting with the base cuboid [day, doctor, patient], what specific OLAP operations should be performed in order to list the total fee collected by each doctor in 2010?

(iii) To obtain the same list, write an SQL query assuming the data are stored in a relational database with the schema fee (day, month, year, doctor, hospital, patient, count, charge).

MU -Dec. 2019

UQ. A dimension table is wide, the fact table is deep. Explain. **MU - Dec.-2019**

UQ. For a supermarket chain, consider the following dimensions namely product, store, time and promotion. The schema contains a central fact table for sales with three measures unit_sales, dollars_sales and dollar_cost. Draw star schema. Calculate the maximum number of base fact table records for warehouse with the following values given below:

Time period-5 years

Store - 300 stores reporting daily sales

Product - 40,000 products in each store (about 4000 sell in each store daily).

Promotion: a sold item may be in only one promotion in a store on a given day.

MU - June 2021

Schema is a logical description of the entire database. It includes the name and description of records of all record types including all associated data-items and aggregates. Much like a database, a data warehouse also requires to maintain a schema. A database uses relational model, while a data warehouse uses Star, Snowflake, and Fact Constellation schema.

1.7.1 Star Schema

- A star schema is the elementary form of a dimensional model, in which data are organized into facts and dimensions.
- A fact is an event that is counted or measured, such as a sale or dealer credits.
- A dimension includes reference data about the fact, such as date, item, or customer.

- A star schema is a relational schema where a relational schema whose design represents a multidimensional data model. The star schema is the explicit data warehouse schema.
- It is known as star schema because the entity-relationship diagram of this schemas simulates a star, with points, diverge from a central table.
- The center of the schema consists of a large fact table, and the points of the star are the dimension tables.

Fact Tables

- It is a table in a star schema which contains facts and connected to dimensions.
- A fact table has two types of columns: those that include fact and those that are foreign keys to the dimension table.
- The primary key of the fact tables is generally a composite key that is made up of all of its foreign keys.

Dimension Tables

- A dimension is an architecture usually composed of one or more hierarchies that categorize data.
- If a dimension has not got hierarchies and levels, it is called a **flat dimension** or **list**.
- The primary keys of each of the dimension table are part of the composite primary keys of the fact table.
- Dimensional attributes help to define the dimensional value. They are generally descriptive, textual values.
- Dimensional tables are usually small in size than fact table.

Note : Fact tables are deep whereas dimension tables are wide as fact tables will have a higher number of rows and a lesser number of columns. A primary key defined in the fact table is primarily to identify each row separately. The primary key is also called a Composite key in fact table. A dimension table contains a higher granular information so have less no of records and it needs to have all the necessary details (more columns) related to the grain of the table. On the other side, a fact table has the lowest level grain of a subject area. Lower grain causes more number of rows in the Fact table.

1.7.1(A) Characteristics of Star Schema

- It creates a de-normalized database that can quickly provide query responses.
- It provides a flexible design that can be changed easily or added to throughout the development cycle, and as the database grows.
- It provides a parallel in design to how end-users typically think of and use the data.
- It reduces the complexity of metadata for both developers and end-users.

1.7.1(B) Keys in Star Schema

1. **Primary Keys:** The primary key of the dimension table identifies each roe in a dimension table. Example: In a student dimension table, student_id is the primary key which identifies each student uniquely.
2. **Surrogate Keys:** System generated sequence numbers are called surrogate keys. They do not have any built in meanings.
3. **Foreign Keys:** Every dimension table has one-to-one relationship with the fact table. The primary key in the dimension table acts as a foreign key in the fact table.

1.7.1(C) Advantages of Star Schema

1. Query Performance

- A star schema database has a limited number of table and clear join paths, the query run faster than they do against OLTP systems. Small single-table queries, frequently of a dimension table, are almost instantaneous. Large join queries that contain multiple tables takes only seconds or minutes to run.
- In a star schema database design, the dimension is connected only through the central fact table. When the two-dimension table is used in a query, only one join path, intersecting the fact tables, exist between those two tables. This design feature enforces authentic and consistent query results.

2. Load performance and administration

- Structural simplicity also decreases the time required to load large batches of record into a star schema database. By describing facts and dimensions and separating them into the various table, the impact of a load structure is reduced.

- Dimension table can be populated once and occasionally refreshed. We can add new facts regularly and selectively by appending records to a fact table.

3. Built-in referential integrity

- A star schema has referential integrity built-in when information is loaded. Referential integrity is enforced because each data in dimensional tables has a unique primary key, and all keys in the fact table are legitimate foreign keys drawn from the dimension table.
- A record in the fact table which is not related correctly to a dimension cannot be given the correct key value to be retrieved.

4. Easily Understood

- A star schema is simple to understand and navigate, with dimensions joined only through the fact table.
- These joins are more significant to the end-user because they represent the fundamental relationship between parts of the underlying business. Customer can also browse dimension table attributes before constructing a query.

1.7.1(D) Disadvantages of Star Schema

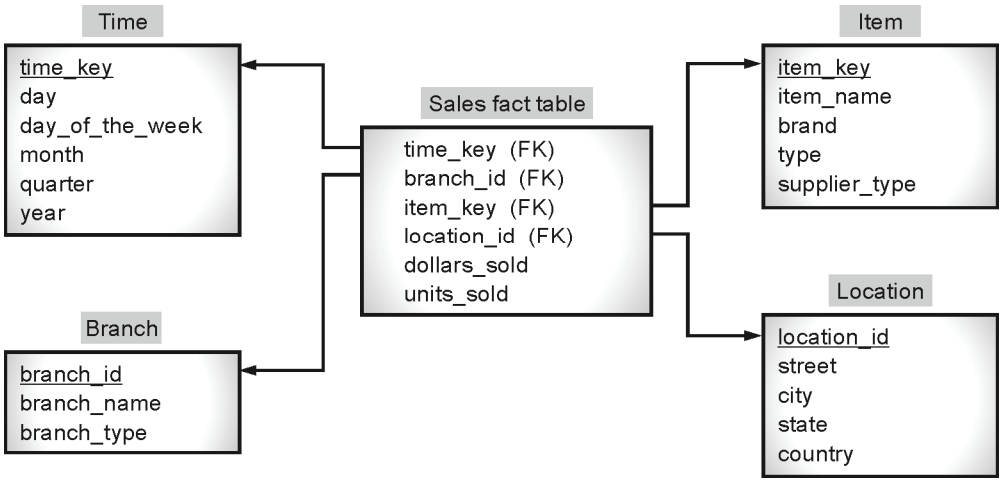
(1) Data integrity is not enforced well since in a highly de-normalized schema state.

(2) Not flexible in terms if analytical needs as a normalized data model.

(3) Star schemas don't reinforce many-to-many relationships within business entities, at least not frequently.

1.7.1(E) Example

A star schema for Digi1 Electronics sales is shown. Sales are considered along four dimensions: time, item, branch, and location. The schema contains a central fact table for sales that contains keys to each of the four dimensions, along with two measures: dollars sold and units sold.



(1A5)**Fig. 1.7.1: Star Schema for Digi1 Electronics Sale**

1.7.2 Snowflake Schema

- The snowflake schema is a variant of the star schema.
- Here, the centralized fact table is connected to multiple dimensions.
- In the snowflake schema, dimensions are present in a normalized form in multiple related tables.
- The snowflake structure is materialized when the dimensions of a star schema are detailed and highly structured, having several levels of relationship, and the child tables have multiple parent table.
- The snowflake schema affects only the dimension tables and does not affect the fact tables.
- In other words, a dimension table is said to be snowflaked if the low-cardinality attribute of the dimensions has been divided into separate normalized

tables.

- These tables are then joined to the original dimension table with referential constraints (foreign key constraint).

1.7.2(A) Characteristics of Snowflake Schema

(1) The snowflake schema uses small disk space.

(2) It is easy to implement dimension that is added to schema.

(3) There are multiple tables, so performance is reduced.

(4) The dimension table consist of two or more sets of attributes which define information at different grains.

(5) The sets of attributes of the same dimension table are being populated by different source systems.

1.7.2(B) Advantages of Snowflake Schema

(1) It provides structured data which reduces the problem of data integrity.

(2) It uses small disk space because data are highly structured.

1.7.2(C) Disadvantages of Snowflake Schema

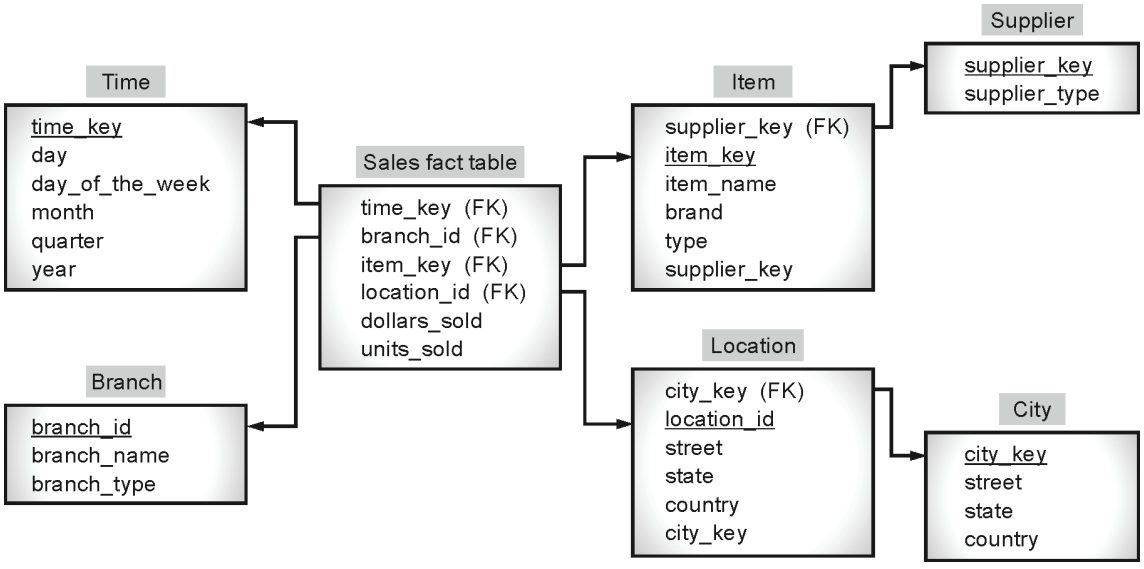
(1) Snowflaking reduces space consumed by dimension tables, but compared with the entire data warehouse the saving is usually insignificant.

(2) Avoid snowflaking or normalization of a dimension table, unless required and appropriate.

(3) Do not snowflake hierarchies of one-dimension table into separate tables. Hierarchies should belong to the dimension table only and should never be snowflaked.

1.7.2(D) Example

- A snowflake schema for Digi1 Electronics sales is given. Here, the sales fact table is identical to that of the star schema in Fig. 1.7.1. The main difference between the two schemas is in the definition of dimension tables. The single dimension table for item in the star schema is normalized in the snowflake schema, resulting in new item and supplier tables.
- For example, the item dimension table now contains the attributes item_key, item_name, brand, type, and supplier_key, where supplier_key is linked to the supplier dimension table, containing supplier_key and supplier_type information.
- Similarly, the single dimension table for location in the star schema can be normalized into two new tables: location and city. The city_key in the new location table links to the city dimension. Notice that, when desirable, further normalization can be performed on state and country in the snowflake schema.



(1A6)**Fig. 1.7.2: Snowflake Schema for Digi1 Electronics Sale**

1.7.3 Star Schema Vs Snowflake Schema

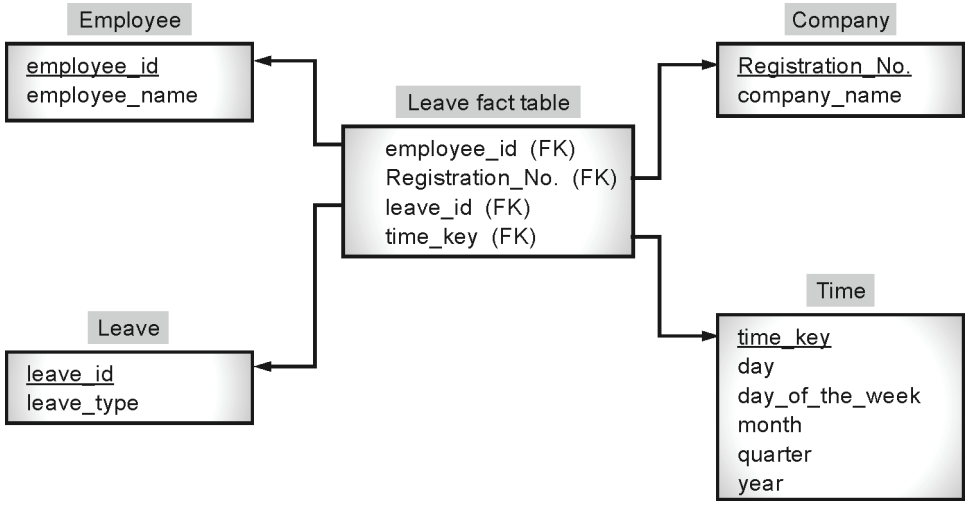
Table 1.7.1: Star Schema Vs Snowflake Schema

Sr. No.	Basis	Star Schema	Snowflake Schema
1.	Ease of Maintenance/change	It has redundant data and hence less easy to maintain/change.	No redundancy and therefore more easy to maintain and change.
2.	Ease of Use	Less complex queries and simple to understand.	More complex queries and therefore less easy to understand.
3.	Parent table	In a star schema, a dimension table will not have any parent table.	In a snowflake schema, a dimension table will have one or more parent tables.
4.	Query Performance	Less number of foreign keys and hence lesser query execution time.	More foreign keys and thus more query execution time.
5.	Normalization	It has De-normalized tables.	It has normalized tables.
6.	Type of Data Warehouse	Good for data marts with simple relationships (one to one or one to many).	Good to use for data warehouse core to simplify complex relationships (many to many).
7.	Joins	Fewer joins	Higher number of joins.
8.	Dimension Table	It contains only a single dimension table for each dimension.	It may have more than one dimension table for each dimension.
9.	Hierarchies	Hierarchies for the dimension are stored in the dimensional table itself in a star schema.	Hierarchies are broken into separate tables in a snowflake schema. These hierarchies help to drill down the information from topmost hierarchies to the lowermost hierarchies.
10.	When to use	When the dimensional table contains less number of rows, we can go for Star schema.	When dimensional table store a huge number of rows with redundancy information and space is such an issue, we can choose snowflake schema to store space.
11.	Data Warehouse system	Work best in any data warehouse/ data mart.	Better for small data warehouse/data mart.

1.7.4 Factless Fact Table

- A data warehouse factless fact table is a fact that does not have any measures stored in it.
- This table will only contain keys from different dimension tables.
- The fact-less fact is often used to resolve a many-to-many cardinality issue.
- There are two types of factless fact tables:

1. Event capturing factless fact
2. Coverage table – Describing condition



(1A7)**Fig. 1.7.3: Factless Fact Table for Leave**

1. Event Capturing Factless Fact

- This type of fact table establishes the relationship among the various dimension members from various dimension tables without any measured value.
- For example, Student attendance (student-teacher relation table) capturing table is the factless fact. Table will have entry into it whenever student attend class.
- Following questions can be answered by the student attendance table:
  1. Which student is taught by the maximum number of teachers?
  2. Which class has maximum number of attendance?
  3. Which teacher teaches maximum number of students?
  - All the above queries are based on the **COUNT ()**, **MAX ()** with **GROUP BY**.

2. Coverage Table-Describing Condition

- This is another kind of factless fact. A factless fact table can only answer 'optimistic' queries (positive query) but cannot answer a negative query.
- Coverage fact is used to support negative analysis reports. For example, an electronic store did not sell any product for given period of time.
- If you consider the student-teacher relation table, the event capturing fact table cannot answer 'which teacher did not teach any student?'
- Coverage fact attempts to answer this question by adding extra flag 0 for negative condition and 1 for positive condition.
- If the student table has 20 records and teacher table has 3 records, then coverage fact table will store 20 * 3 = 60 records for all possible combinations. If any teacher is not teaching particular student, then that record will have flag 0 in it.

1.7.5 Fact Constellation Schema

- Fact Constellation is a schema for representing multidimensional model.
- It is a collection of multiple fact tables having some common dimension tables.
- It can be viewed as a collection of several star schemas and hence, also known as *Galaxy schema*.
- It is one of the widely used schema for Data warehouse designing and it is much more complex than star and snowflake schema.
- For complex systems, we require fact constellations.

1.7.5(A) Advantages of Fact Constellation Schema

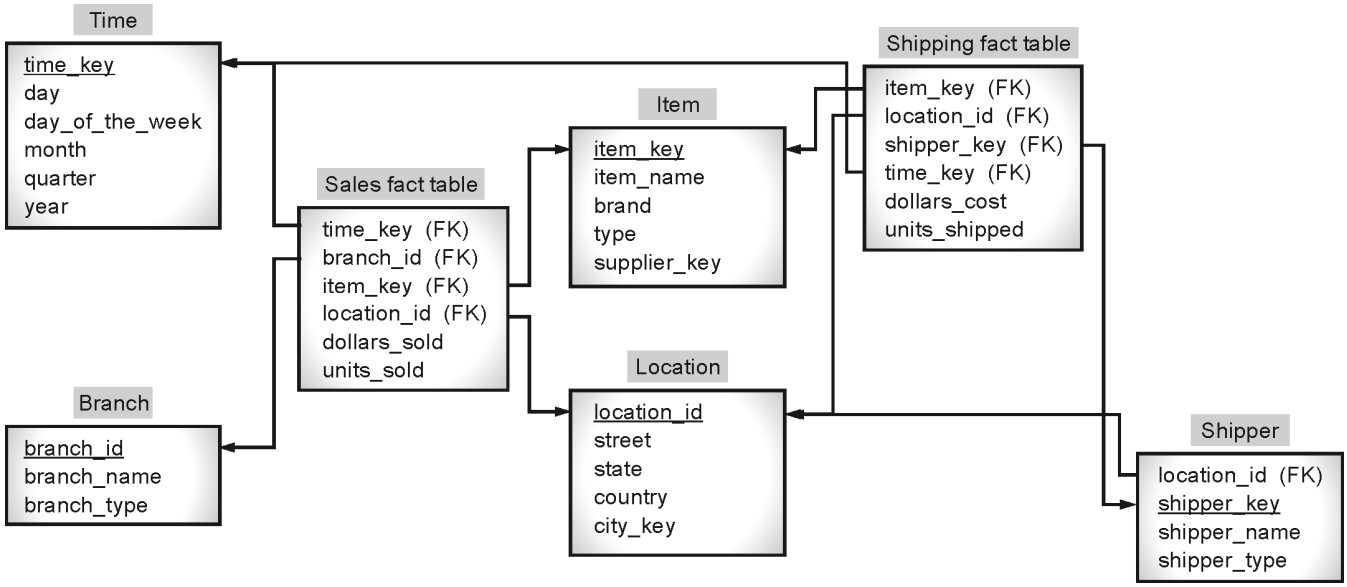
Provides a flexible schema.

1.7.5(B) Disadvantages of Fact Constellation Schema

It is much more complex and hence, hard to implement and maintain.

1.7.5(C) Example

A fact constellation schema is shown in Fig. 1.7.4 below. This schema specifies two fact tables, sales and shipping. The sales table definition is identical to that of the star schema above. The shipping table has four dimensions, or keys—item_key, time_key, shipper_key, location_id and two measures dollars_cost and units_shipped. A fact constellation schema allows dimension tables to be shared between fact tables. For example, the dimension tables for time, item, and location are shared between the sales and shipping fact tables.



(1A8)**Fig. 1.7.4: Fact Constellation Schema for Digi1 Electronics Sale**

1.7.6 Schema Definition

Multidimensional schema is defined using Data Mining Query Language (DMQL). The two primitives, cube definition and dimension definition, can be used for defining the data warehouses and data marts.

Syntax for Cube Definition

```
define cube < cube_name > [ < dimension-list >]: < measure_list >
```

Syntax for Dimension Definition

```
define dimension < dimension_name > as ( < attribute_or_dimension_list > )
```

Star Schema Definition

The star schema that we have discussed can be defined using Data Mining Query Language (DMQL) as follows:

```
define cube sales star [time, item, branch, location]:
dollars_sold = sum(sales in dollars), units_sold = count(*)
define dimension time as (time_key, day, day_of_week,
month, quarter, year)
define dimension item as (item_key, item_name, brand, type,
supplier_type)
define dimension branch as (branch_id, branch_name,
branch_type)
define dimension location as (location_id, street, city, state,
country)
```

Snowflake Schema Definition

Snowflake schema can be defined using DMQL as follows:

```
define cube sales snowflake [time, item, branch, location]:
dollars_sold = sum(sales in dollars), units_sold = count(*)
define dimension time as (time_key, day, day_of_week,
month, quarter, year)
define dimension item as (item_key, item_name, brand, type,
supplier (supplier_key, supplier_type))
define dimension branch as (branch_id, branch_name,
branch_type)
define dimension location as (location_id, street, city
(city_key, street, state, country))
```

Fact Constellation Schema Definition

Fact constellation schema can be defined using DMQL as follows :

```
define cube sales [time, item, branch, location]:dollars_sold = sum(sales in dollars), units sold = count(*)
define dimension time as (time_key, day, day_of_week, month, quarter, year)
define dimension item as (item_key, item_name, brand, type, supplier_type)
define dimension branch as (branch_id, branch_name, branch_type)
define dimension location as (location_id, street, city, state,country)

define cube shipping [time, item, shipper, location_id]:dollars_cost = sum(cost in dollars), units shipped = count(*)
define dimension time as time in cube sales
define dimension item as item in cube sales
define dimension shipper as (shipper_key, shipper_name, location as location in cube sales, shipper_type)
define dimension location as location in cube sales
```

Ex. 1.7.1 :

Consider a data warehouse for hotel occupancy, where there are four dimensions namely (a) Hotel (b) Room (c) Time (d) Customer and two measures (i) Occupied rooms (ii) Vacant rooms.

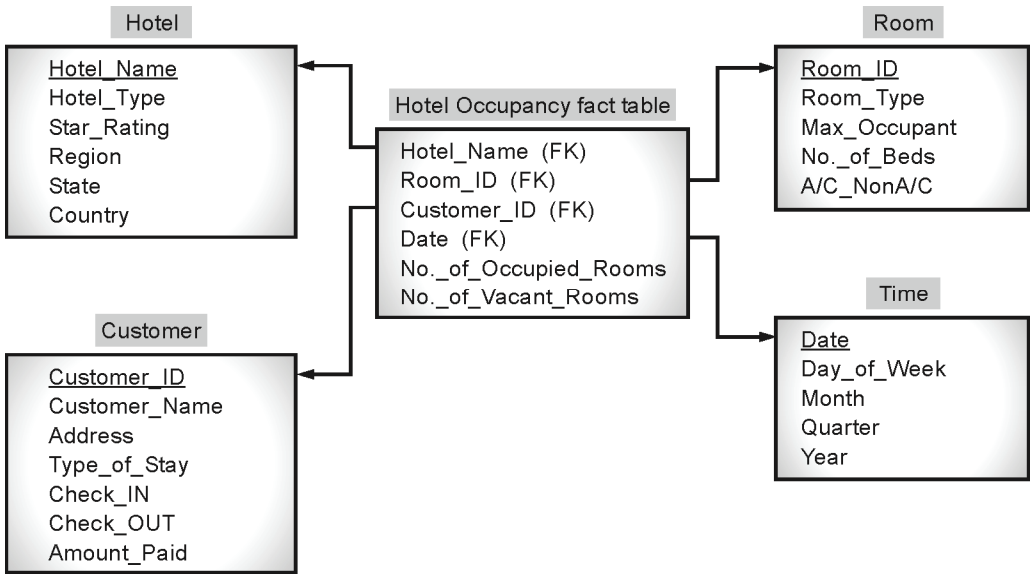
Draw information package diagram, star schema and snowflake schema.

☑ **Soln. :**

(a) Information Package Diagram

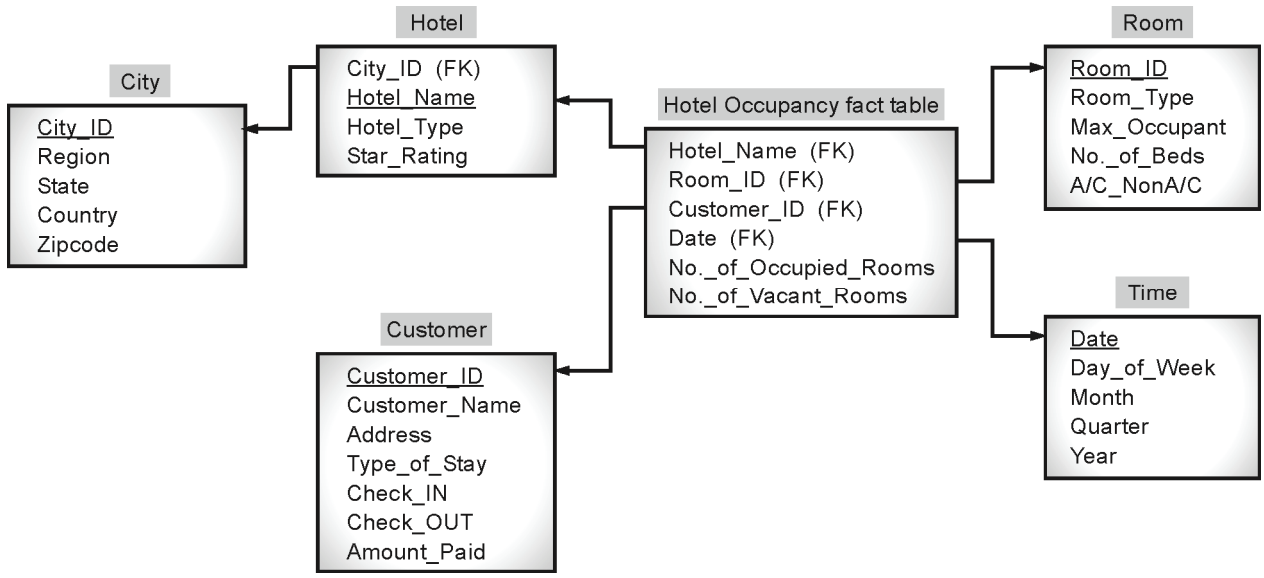
Hierarchies/Categories	Dimensions					
	Hotel	**Room**	**Time**	**Customer**	**...**	**...**
	Hotel Name	Room ID	Date	Customer ID		
	Hotel Type	Room Type	Day of Week	Customer Name		
	Star Rating	Max. Occupant	Month	Address		
	Region	No. of Beds	Quarter	Type of Stay		
	State	A/C Non A/C	Year	Check IN		
	Country			Check OUT		
				Amount Paid		
	Facts : Occupied Rooms, Vacant Rooms					

(b) Star Schema



(1A9)**Fig. P. 1.7.1(a): Star Schema for Hotel Occupancy**

(c) Snowflake Schema



(1A10)**Fig. P. 1.7.1(b): Snowflake Schema for Hotel Occupancy**

UEx. 1.7.2 MU - Dec. 2019

Consider a data warehouse for a hospital where there are three dimensions namely (a) Doctor (b) Patient (c) Time and two measures (i) count (ii) charge where charge is the fee that the doctor charges a patient for a visit.

(i) Draw star and snowflake schema.

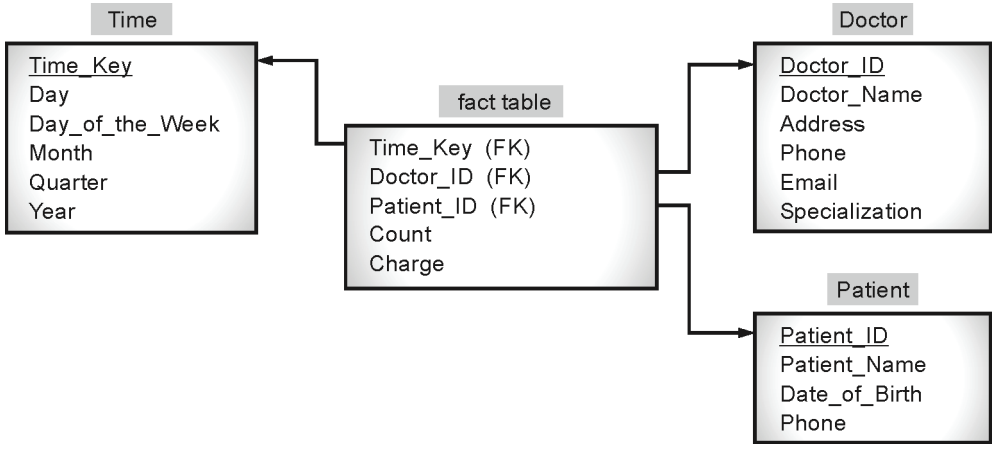
(ii) Starting with the base cuboid [day, doctor, patient], what specific OLAP operations should be performed in order to list the total fee collected by each doctor in 2010?

(iii) To obtain the same list, write an SQL query assuming the data are stored in a relational database with the schema fee (day, month, year, doctor, hospital, patient, count, charge).

☑ **Soln. :**

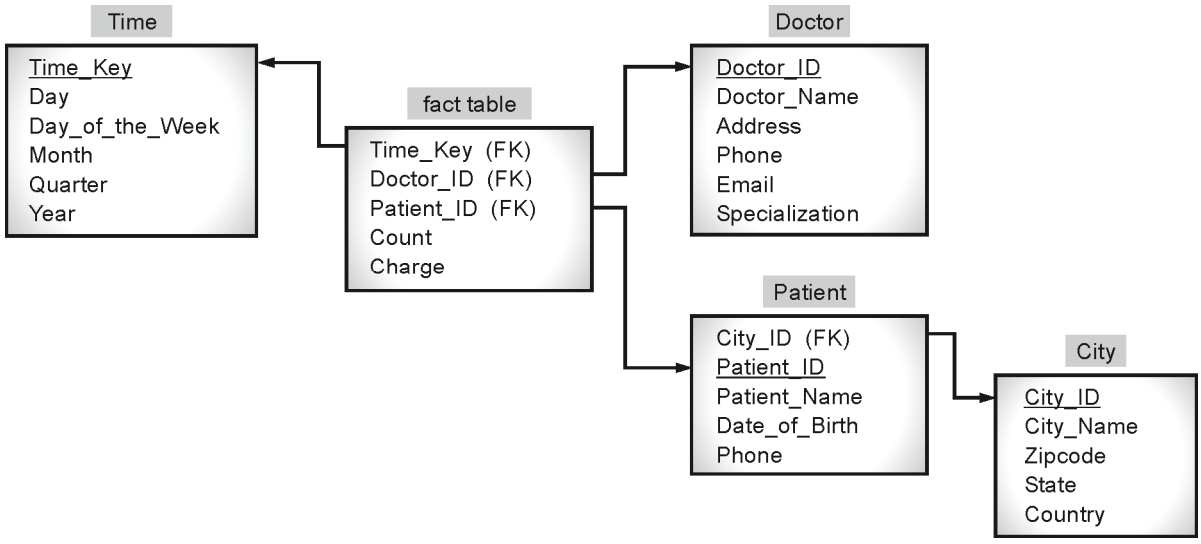
(i) Star Schema and Snowflake Schema

(a) Star Schema



(1A11)**Fig. 1.7.2(a): Star Schema for Hospital**

(b) Snowflake Schema



(1A12)**Fig. 17.2(b): Snowflake Schema for Hospital**

(ii) First, we should use roll-up operation to get the year 2010 (rolling-up from day then month to year). After getting that, we need to use slice operation to select (2010). Second, we should use roll-up operation again to get all patients. Then, we need to use slice operation to select (all). Finally, we get list the total fee collected by each doctor in 2010. So,

1. roll up from day to month to year
2. slice for year = "2010"
3. roll up on patient from individual patient to all
4. slice for patient = "all"
5. get the list of total fee collected by each doctor in 2010.

(iii) SELECT doctor, sum(charge) FROM fee WHERE year = 2004 GROUP BY doctor;

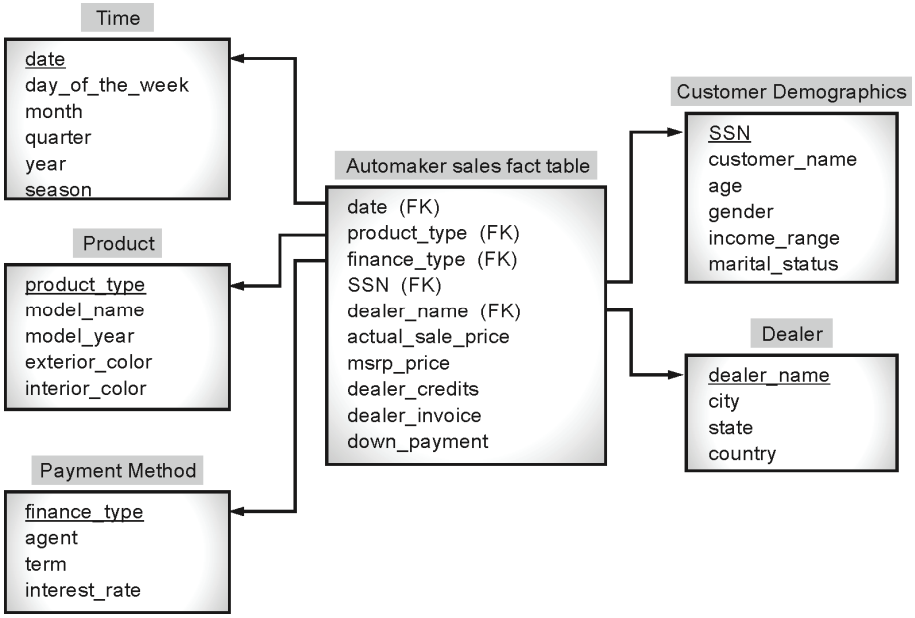
Ex. 1.7.3 : Consider a data warehouse for automaker sales where there are five dimensions, namely (a) Time (b) Product (c) Payment Method (d) Customer Demographics (e) Dealer and facts/measures like Actual Sale Price, MSRP Sale Price, Dealer Credits, Dealer Invoice, Down Payment. Draw the information package diagram and star schema.

☑ **Soln. :**

(a) Information Package Diagram

Hierarchies/Categories	Dimensions					
	Time	**Product**	**Payment Method**	**Customer Demographics**	**Dealer**	**…**
	Date	Product Type	Finance Type	SSN Number	Dealer Name	
	Day of week	Model Name	Agent	Customer Name	City	
	Month	Model Year	Term	Age	State	
	Quarter	Exterior Color	Interest Rate	Gender	Country	
	Year	Interior Color		Income Range		
	Season			Marital Status		
	Facts : Actual Sale Price, MSRP Sale Price, Dealer Credits, Dealer Invoice, Down Payment					

(b) Star Schema



(1A13)**Fig. P. 1.7.3: Star Schema for Automaker Sales**

UEx. 1.7.4 **MU - June 2021**

For a supermarket chain, consider the following dimensions namely product, store, time and promotion. The schema contains a central fact table for sales with three measures unit_sales, dollars_sales and dollar_cost.

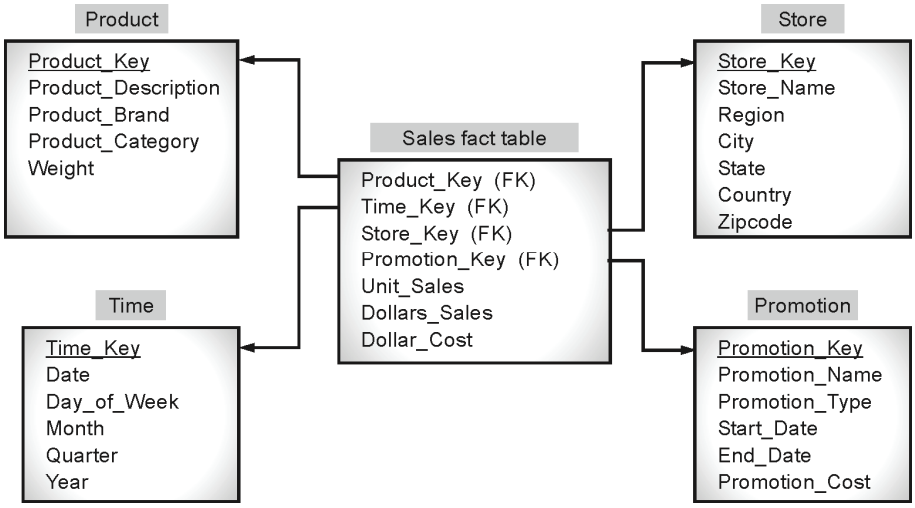
i. Draw star schema.

ii. Calculate the maximum number of base fact table records for warehouse with the followingvalues given below:

- Time period-5 years
- Store - 300 stores reporting daily sales
- Product - 40,000 products in each store (about 4000 sell in each store daily).
- Promotion: a sold item may be in only one promotion in a store on a given day.

☑ **Soln. :**

i. Star Schema



(1A14)**Fig. P. 1.7.4: Star Schema for Supermarket Chain**

ii. Fact table records

- Time period = 5 years = 5 × 365 days = 1825 days
- Number of Stores = 300
- Product sell in each store daily = 4000
- Promotion = 1

Therefore, maximum number of fact table records = 1825 × 300 × 4000 × 1 = 2190000000 records

UEx. 1.7.5 MU - May 2019

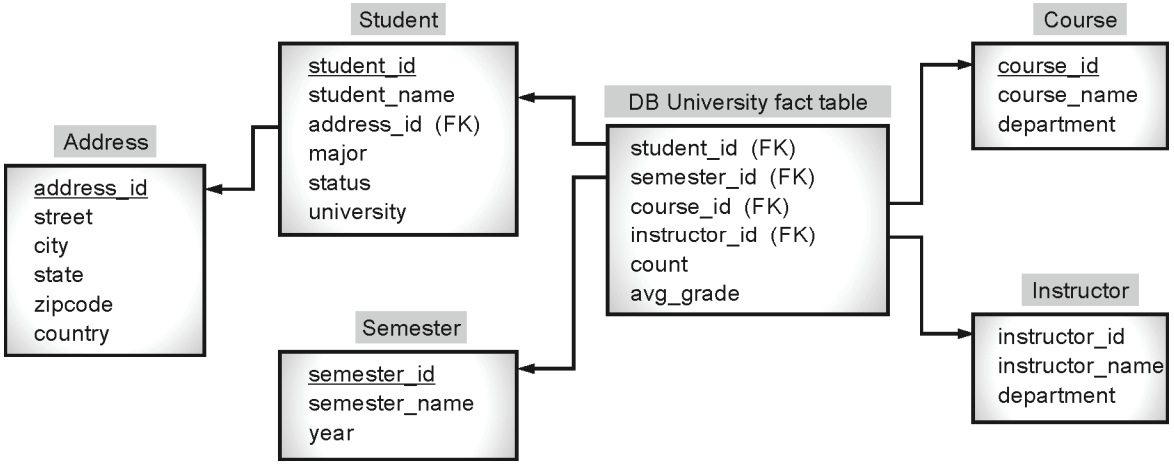
Suppose that a data warehouse for DB-University consists of the following four dimensions: student, course, semester, and instructor, and two measures count and avg_grade. When at the lowest conceptual level (e.g., for a given student, course, semester, and instructor combination), the avg_grade measure stores the actual course grade of the student. At higher conceptual levels, avg_grade stores the average grade for the given combination.

(a) Draw a snowflake schema diagram for the data warehouse.

(b) Starting with the base cuboid [student, course, semester, instructor], what specific OLAP operations (e.g., roll-up from semester to year) should one perform in order to list the average grade of CS courses for each DB_University student.

☑ **Soln. :**

(a) Snowflake Schema



(1A15)**Fig. P. 1.7.5: Snowflake Schema for DB_University**

(b) OLAP Operations to list the average grade of CS courses

i. Roll-up on course from course_id to department.

ii. Roll-up on student from student_id to university.

iii. Dice on course, student with department ="CS" and university = "DB_University".

iv. Drill-down on student from university to student_name.

⏭ 1.8 UPDATE TO THE DIMENSION TABLES

- The number of rows in the fact table continues to increase over time. Rows in a fact table are rarely updated with changes. Even if there are adjustments to the prior numbers, these are processed as additional adjustment rows and added to the fact table.
- Dimension tables are more stable and less volatile. However, unlike a fact table, which changes as the number of rows increases, a dimension table changes as the attributes themselves change.

1.8.1 Slowly Changing Dimensions

- Slowly changing dimensions are the dimensions that change slowly over time, rather than changing on regular schedule, time-base. In data warehouse there is a need to track changes in dimension attributes in order

to report historical data. In other words, implementing one of the slowly changing dimension types should enable users assigning proper dimension's attribute value for given date. Example of such dimensions could be: customer, geography, employee.

- There are many approaches how to deal with slowly changing dimension. The most popular are:
  - **Type 0 -** The passive method
  - **Type 1** - Overwriting the old value
  - **Type 2** - Creating a new additional record
  - **Type 3** - Adding a new column
  - **Type 4** - Using historical table
  - **Type 6** - Combine approaches of types 1,2,3 (1+2+3=6)
- **Type 0 : The passive method**. In this method no special action is performed upon dimensional changes. Some dimension data can remain the same as it was first time inserted, others may be overwritten.
- **Type 1 : Overwriting the old value**. In this method no history of dimension changes is kept in the database. The old dimension value is simply overwritten be the new one. This type is easy to maintain and is often use for data in which changes are caused by processing corrections(e.g. removal of special characters, correcting spelling errors).

Before the change :

Customer_ID	Customer_Name	Customer_Type
1	Cust_1	Corporate

After the change :

Customer_ID	Customer_Name	Customer_Type
1	Cust_1	Retail

- **Type 2 :** Creating a new additional record. In this methodology all history of dimension changes is kept in the database. You capture attribute change by adding a new row with a new surrogate key to the dimension table. Both the prior and new rows contain as attributes the natural key(or other durable identifier). Also 'effective date' and 'current indicator' columns are used in this method. There could be only one record with current indicator set to 'Y'. For 'effective date' columns, i.e. start_date and end_date, the end_date for current record usually is set to value 31-12-9999. Introducing changes to the dimensional model in type 2 could be very expensive database operation so it is not recommended to use it in dimensions where a new attribute could be added in the future.

Before the change:

Customer_ID	Customer_Name	Customer_Type	Start_Date	End_Date	Current_Flag
1	Cust_1	Corporate	22-07-2010	31-12-9999	Y

After the change:

Customer_ID	Customer_Name	Customer_Type	Start_Date	End_Date	Current_Flag
1	Cust_1	Corporate	22-07-2010	17-05-2012	N
2	Cust_1	Retail	18-05-2012	31-12-9999	Y

- **Type 3 : Adding a new column**. In this type usually only the current and previous value of dimension is kept in the database. The new value is loaded into 'current/new' column and the old one into 'old/previous' column. Generally speaking, the history is limited to the number of column created for storing historical data. This is the least commonly needed technique.

Before the change:

Customer_ID	Customer_Name	Current_Type	Previous_Type
1	Cust_1	Corporate	Corporate

After the change:

Customer_ID	Customer_Name	Current_Type	Previous_Type
1	Cust_1	Retail	Corporate

- **Type 4 : Using historical table**. In this method a separate historical table is used to track all dimension's attribute historical changes for each of the dimension. The 'main' dimension table keeps only the current data e.g. customer and customer_history tables.

Current table:

Customer_ID	Customer_Name	Customer_Type
1	Cust_1	Corporate

Historical table:

Customer_ID	Customer_Name	Customer_Type	Start_Date	End_Date
1	Cust_1	Retail	01-01-2010	21-07-2010
1	Cust_1	Other	22-07-2010	17-05-2012
1	Cust_1	Corporate	18-05-2012	31-12-9999

- **Type 6 - Combine approaches of types 1,2,3 (1+2+3=6)**. In this type we have in dimension table such additional columns as:
  - current_type - for keeping current value of the attribute. All history records for given item of attribute have the same current value.
  - historical_type - for keeping historical value of the attribute. All history records for given item of attribute could have different values.
  - start_date - for keeping start date of 'effective date' of attribute's history.
  - end_date - for keeping end date of 'effective date' of attribute's history.
  - current_flag - for keeping information about the most recent record.
- In this method to capture attribute change we add a new record as in type 2. The current_type information is overwritten with the new one as in type 1. We store the history in a historical_column as in type 3.

Customer_ID	Customer_Name	Current_Type	Historical_Type	Start_Date	End_Date	Current_Flag
1	Cust_1	Corporate	Retail	01-01-2010	21-07-2010	N
2	Cust_1	Corporate	Other	22-07-2010	17-05-2012	N
3	Cust_1	Corporate	Corporate	18-05-2012	31-12-9999	Y

1.8.2 Rapidly Changing Dimension (RCD)

- A dimension is a fast changing or rapidly changing dimension if one or more of its attributes in the table changes very fast and in many rows. Handling rapidly changing dimension in data warehouse is very difficult because of many performance implications.
- As you know slowly changing dimension type 2 is used to preserve the history for the changes. But the problem with type 2 is, with each and every change in the dimension attribute, it adds new row to the table. If in case there are dimensions that are changing a lot, table become larger and may cause serious performance issues. Hence, use of the type 2 may not be the wise decision to implement the rapidly changing dimensions.
- For example: Consider patient dimension where there are 1000 rows in it. On average basis, each patient changes the 10 of attributes in a year. If you use the type 2 to manage this scenario, there will be 1000 * 10 = 10000 rows. Imagine if the table has 1 million rows, it will become very hard to handle the situation with type 2. Hence we use rapidly changing dimension approach.

1.8.3 Conformed Dimension

- A conformed dimension is the dimension that is shared across multiple data mart or subject area. Company may use the same dimension table across different projects without making any changes to the dimension tables.
- Conformed dimension example would be Customer dimension, i.e. both marketing and sales department can use Customer dimension for their reporting purpose.

1.8.4 Junk Dimension

- A junk dimension is a grouping of typically low cardinality attributes, so you can remove them from main dimension.
- You can use Junk dimensions to implement the rapidly changing dimension where you can use it to stores the attribute that changes rapidly. For example, attributes such as flags, weights, BMI (body mass index), etc.

1.8.5 Degenerated Dimension

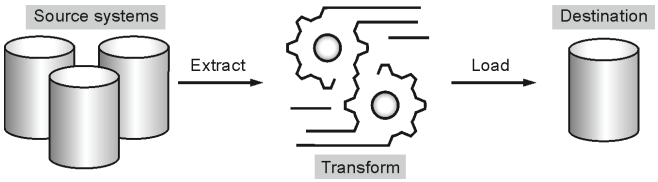
- A degenerated dimension is a dimension that is derived from fact table and does not have its own dimension table.For example, receipt number does not have dimension table associated with it. Such details are just for information purpose.

1.8.6 Role Playing Dimension

Dimensions which are often used for multiple purposes within the same database are called role-playing dimensions. For example, you can use a date dimension for "date of sale", as well as "date of delivery", or "date of hire".

1.9 MAJOR STEPS IN ETL PROCESS

- The process of extracting data from source systems and bringing it into the data warehouse is commonly called **ETL**, which stands for extraction, transformation, and loading. Note that ETL refers to a broad process, and not three well-defined steps.
- It is a process in which an ETL tool extracts the data from various data source systems, transforms it in the staging area and then finally, loads it into the Data Warehouse system.
- The ETL process requires active inputs from various stakeholders, including developers, analysts, testers, top executives and is technically challenging.
- To maintain its value as a tool for decision-makers, Data warehouse technique needs to change with business changes.
- ETL is a recurring method (daily, weekly, monthly) of a Data warehouse system and needs to be agile, automated, and well documented.
- **ETL Tools :** Most commonly used ETL tools are Sybase, Oracle Warehouse builder, Clover ETL and MarkLogic.



(1A16)**Fig. 1.9.1 : ETL Process**

Let us understand each step of the ETL process in depth:

1. Extraction

- The first step of the ETL process is extraction. In this step, data from various source systems is extracted which can be in various formats like relational databases, NoSQL, XML and flat files into the staging area.
- It is important to extract the data from various source systems and store it into the staging area first and not directly into the data warehouse because the extracted data is in various formats and can be corrupted also.
- Hence loading it directly into the data warehouse may damage it and rollback will be much more difficult. Therefore, this is one of the most important steps of ETL process.

Data Extraction Techniques

There are two types of data warehouse extraction methods: Logical and Physical extraction methods.

(A) Logical Extraction

Logical Extraction method in-turn has two methods:

(i) Full Extraction

- In this method, data is completely extracted from the source system. The source data will be provided as-is and no additional logical information is necessary on the source system. Since it is complete extraction, so no need to track source system for changes.
- For example, exporting complete table in the form of flat file.

(ii) Incremental Extraction

- In incremental extraction, the changes in source data need to be tracked since the last successful extraction.

Only these changes in data will be extracted and then loaded. Identifying the last changed data itself is the complex process and involve many logics.

- You can detect the changes in the source system from the specific column in the source system that has the last changed timestamp. You can also create a change table in the source system, which keeps track of the changes in the source data.

(B) Physical Extraction

Physical extraction has two methods: Online and Offline extraction.

(i) Online Extraction

In this process, extraction process directly connects to the source system and extract the source data.

(ii) Offline Extraction

- The data is not extracted directly from the source system but is staged explicitly outside the original source system.
- You can consider the following common structure in offline extraction:
  1. **Flat file:** Generic format
  2. **Dump file:** Database specific file

2. Transformation

- The second step of the ETL process is transformation. In this step, a set of rules or functions are applied on the extracted data to convert it into a single standard format.
- It may involve following processes/tasks:
  1. Filtering – loading only certain attributes into the data warehouse.
  2. Cleaning – filling up the NULL values with some default values, mapping U.S.A, United States and America into USA, etc.
  3. Joining – joining multiple attributes into one.
  4. Splitting – splitting a single attribute into multiple attributes.
  5. Sorting – sorting tuples on the basis of some attribute (generally key-attribute).

☞ **Data Transformation Techniques**

- **Data Smoothing:** This method is used for removing the noise from a dataset. Noise is referred to as the distorted and meaningless data within a dataset. Smoothing uses algorithms to highlight the special features in the data. After removing noise, the process can detect any small changes to the data to detect special patterns.
- **Data Aggregation:** Aggregation is the process of collecting data from a variety of sources and storing it in a single format. Here, data is collected, stored, analyzed and presented in a report or summary format. It helps in gathering more information about a particular data cluster. The method helps in collecting vast amounts of data.
- **Discretization:** This is a process of converting continuous data into a set of data intervals. Continuous attribute values are substituted by small interval labels. This makes the data easier to study and analyze.
- **Generalization:** In this process, low-level data attributes are transformed into high-level data attributes using concept hierarchies. For example, age data can be in the form of (20, 30) in a dataset. It is transformed into a higher conceptual level into a categorical value (young, old).
- **Attribute construction:** In the attribute construction method, new attributes are created from an existing set of attributes. For example, in a dataset of employee information, the attributes can be employee name, employee ID and address. These attributes can be used to construct another dataset that contains information about the employees who have joined in the year 2019 only. This method of reconstruction makes mining more efficient and helps in creating new datasets quickly.
- **Normalization:** Also called data pre-processing, this is one of the crucial techniques for **data transformation in data mining.** Here, the data is transformed so that it falls under a given range. When attributes are on different ranges or scales, data modelling and mining can be difficult. Normalization helps in applying data mining algorithms and extracting data faster.

3. Loading

- The third and final step of the ETL process is loading. In this step, the transformed data is finally loaded into the data warehouse.

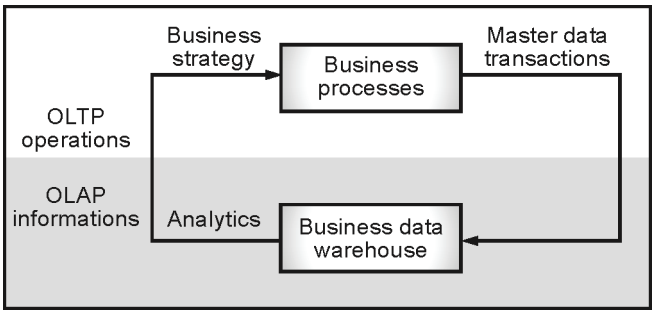
- Sometimes the data is updated by loading into the data warehouse very frequently and sometimes it is done after longer but regular intervals.
- The rate and period of loading solely depends on the requirements and varies from system to system.
- Loading can be carried in two ways:

  (A) Refresh : Data Warehouse data is completely rewritten. This means that older file is replaced. Refresh is usually used in combination with static extraction to populate a data warehouse initially.

  (B) Update : Only those changes applied to source information are added to the Data Warehouse. An update is typically carried out without deleting or modifying pre-existing data. This method is used in combination with incremental extraction to update data warehouses regularly.

⏭ 1.10 OLTP VS OLAP

- We can divide IT systems as transactional (OLTP) and analytical (OLAP). In general, we can assume that OLTP systems provide source data to data warehouses, whereas OLAP systems help to analyze it.
- **OLTP (On-line Transaction Processing)** is characterized by a large number of short on-line transactions (INSERT, UPDATE, DELETE). The main emphasis for OLTP systems is put on very fast query processing, maintaining data integrity in multi-access environments and an effectiveness measured by number of transactions per second. In OLTP database there is detailed and current data, and schema used to store transactional databases is the entity model (usually 3NF).
- **OLAP (On-line Analytical Processing)** is characterized by relatively low volume of transactions. Queries are often very complex and involve aggregations. For OLAP systems a response time is an effectiveness measure. OLAP applications are widely used by Data Mining techniques. In OLAP database there is aggregated, historical data, stored in multi-dimensional schemas (usually star schema). For example, a bank storing years of historical records of check deposits could use an OLAP database to provide reporting to business users. OLAP databases are divided into one or more cubes. The cubes are designed in such a way that creating and viewing reports become easy. At the core of the OLAP concept, is an OLAP Cube. The OLAP cube is a data structure optimized for very quick data analysis. The OLAP Cube consists of numeric facts called measures which are categorized by dimensions. OLAP Cube is also called the **hypercube**.



(1A17)**Fig. 1.10.1 : OLTP Vs OLAP Operations**

- The following table summarizes the major differences between OLTP and OLAP system design.

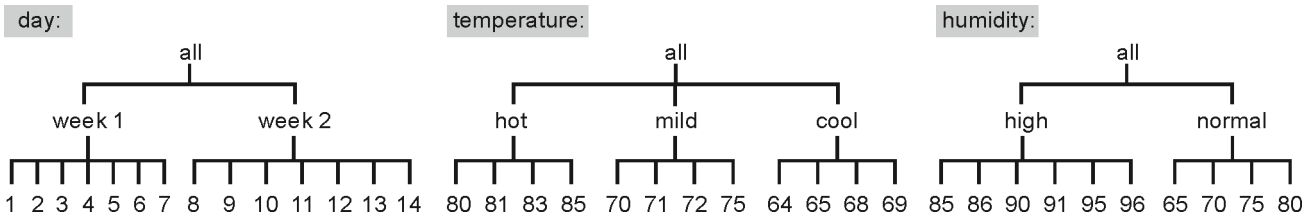
Table 1.10.1: OLTP Vs OLAP

Parameters	OLTP	OLAP
Process	It is an online transactional system. It manages database modification.	OLAP is an online analysis and data retrieving process.
Characteristic	It is characterized by large numbers of short online transactions.	It is characterized by a large volume of data.
Functionality	OLTP is an online database modifying system.	OLAP is an online database query management system.
Method	OLTP uses traditional DBMS.	OLAP uses the data warehouse.
Query	Insert, Update, and Delete information from the database.	Mostly Select operations
Table	Tables in OLTP database are normalized.	Tables in OLAP database are not normalized.

Parameters	OLTP	OLAP
Source	OLTP and its transactions are the sources of data.	Different OLTP databases become the source of data for OLAP.
Storage	The size of the data is relatively small as the historical data is archived. For e.g. MB, GB.	Large amount of data is stored typically in TB, PB.
Data Integrity	OLTP database must maintain data integrity constraint.	OLAP database does not get frequently modified. Hence, data integrity is not an issue.
Response time	It's response time is in millisecond.	Response time in seconds to minutes.
Data quality	The data in the OLTP database is always detailed and organized.	The data in OLAP process might not be organized.
Usefulness	It helps to control and run fundamental business tasks.	It helps with planning, problem-solving, and decision support.
Operation	Allow read/write operations.	Only read and rarely write.
Audience	It is a market orientated process.	It is a customer orientated process.
Query Type	Queries in this process are standardized and simple.	Complex queries involving aggregations.
Back-up	Complete backup of the data combined with incremental backups.	OLAP only need a backup from time to time. Backup is not important compared to OLTP
Design	DB design is application oriented. Example: Database design changes with industry like Retail, Airline, Banking, etc.	DB design is subject oriented. Example: Database design changes with subjects like sales, marketing, purchasing, etc.
User type	It is used by Data critical users like clerk, DBA & Data Base professionals.	Used by Data knowledge users like workers, managers, and CEO.
Purpose	Designed for real time business operations.	Designed for analysis of business measures by category and attributes.
Performance metric	Transaction throughput is the performance metric.	Query throughput is the performance metric.
Number of users	This kind of database allows thousands of users.	This kind of database allows only hundreds of users.
Productivity	It helps to Increase user's self-service and productivity.	Help to Increase productivity of the business analysts.
Challenge	Data Warehouses historically have been a development project which may prove costly to build.	An OLAP cube is not an open SQL server data warehouse. Therefore, technical knowledge and experience is essential to manage the OLAP server.
Process	It provides fast result for daily used data.	It ensures that response to the query is quicker consistently.
Characteristic	It is easy to create and maintain.	It lets the user create a view with the help of a spreadsheet.
Style	OLTP is designed to have fast response time, low data redundancy and is normalized.	A data warehouse is created uniquely so that it can integrate different data sources for building a consolidated database

⏭ 1.11 OLAP OPERATIONS

- In the multidimensional model, data are organized into multiple dimensions, and each dimension contains multiple levels of abstraction defined by concept hierarchies. This organization provides users with the flexibility to view data from different perspectives.
- For example, we have attributes as day, temperature and humidity, we can group values in subsets and name these subsets, thus obtaining a set of hierarchies as shown in Fig. 1.11.1.



(1A18)**Fig. 1.11.1**

- OLAP provides a user-friendly environment for interactive data analysis. A number of OLAP data cube operations exist to materialize different views of data, allowing interactive querying and analysis of the data.

- The most popular end user operations on dimensional data are:

1. Roll-Up

- The roll-up operation (also called drill-up or aggregation operation) performs aggregation on a data cube, either by climbing up a concept hierarchy for a dimension or by climbing down a concept hierarchy, i.e. dimension reduction. Let us explain roll up with an example:
- Consider the following cubes illustrating temperature of certain days recorded weekly:

Temperature	64	65	68	69	70	71	72	75	80	81	83	85
Week1	1	0	1	0	1	0	0	0	0	0	1	0
Week2	0	0	0	1	0	0	1	2	0	1	0	0

- Consider that we want to set up levels (hot (80-85), mild (70-75), cool (64-69)) in temperature from the above cubes.
- To do this, we have to group column and add up the value according to the concept hierarchies. This operation is known as a **roll-up**.
- By doing this, we get the following cube :

Temperature	Cool	mild	Hot
Week1	2	1	1
Week2	2	1	1

- The concept hierarchy can be defined as hot→day→week. The roll-up operation groups the data by levels of temperature.

2. Drill-Down

- The drill-down operation (also called roll-down) is the reverse operation of roll-up. Drill-down is like zooming-in on the data cube. It navigates from less detailed record to more detailed data. Drill-down can be performed by either stepping down a concept hierarchy for a dimension or adding additional dimensions.
- Figure shows a drill-down operation performed on the dimension time by stepping down a concept hierarchy which is defined as day, month, quarter, and year. Drill-down appears by descending the time hierarchy from the level of the quarter to a more detailed level of the month.
- Because a drill-down adds more details to the given data, it can also be performed by adding a new dimension to a cube. For example, a drill-down on the central cubes of the figure can occur by introducing an additional dimension, such as a customer group. Drill-down adds more details to the given data.

Temperature	Cool	mild	Hot
Day 1	0	0	0
Day 2	0	0	0
Day 3	0	0	1
Day 4	0	1	0
Day 5	1	0	0
Day 6	0	0	0
Day 7	1	0	0
Day 8	0	0	0
Day 9	1	0	0
Day 10	0	1	0
Day 11	0	1	0
Day 12	0	1	0
Day 13	0	0	1
Day 14	0	0	0

3. Slice

- A slice is a subset of the cubes corresponding to a single value for one or more members of the dimension. For example, a slice operation is executed when the customer wants a selection on one dimension of a three-dimensional cube resulting in a two-dimensional site. So, the slice operations perform a selection on one dimension of the given cube, thus resulting in a sub-cube. It will form a new sub-cubes by selecting one or more dimensions.
- For example, if we make the selection, temperature = cool we will obtain the following cube:

Temperature	Cool
Day 1	0
Day 2	0
Day 3	0
Day 4	0
Day 5	1
Day 6	1
Day 7	1
Day 8	1
Day 9	1
Day 11	0
Day 12	0
Day 13	0
Day 14	0

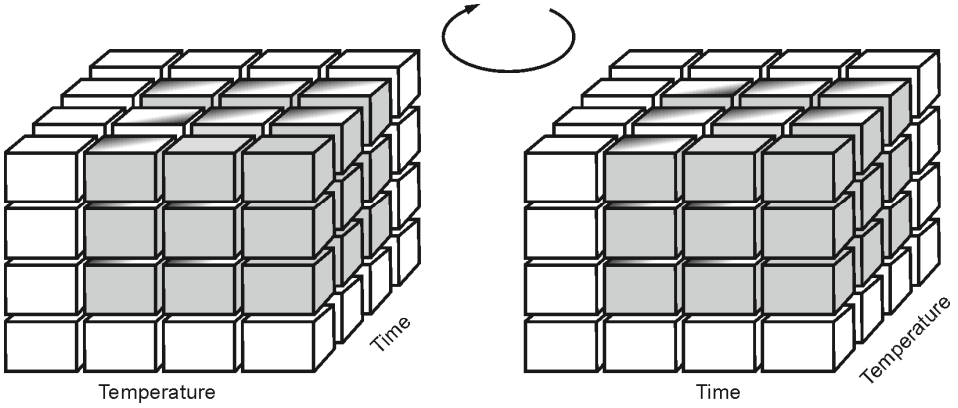
4. Dice

- The dice operation describes a sub-cube by operating a selection on two or more dimension.
- **For example**, Implement the selection (time = day 3 OR time = day 4) AND (temperature = cool OR temperature = hot) to the original cubes we get the following sub-cube (still two-dimensional)

Temperature	Cool	hot
Day 3	0	1
Day 4	0	0

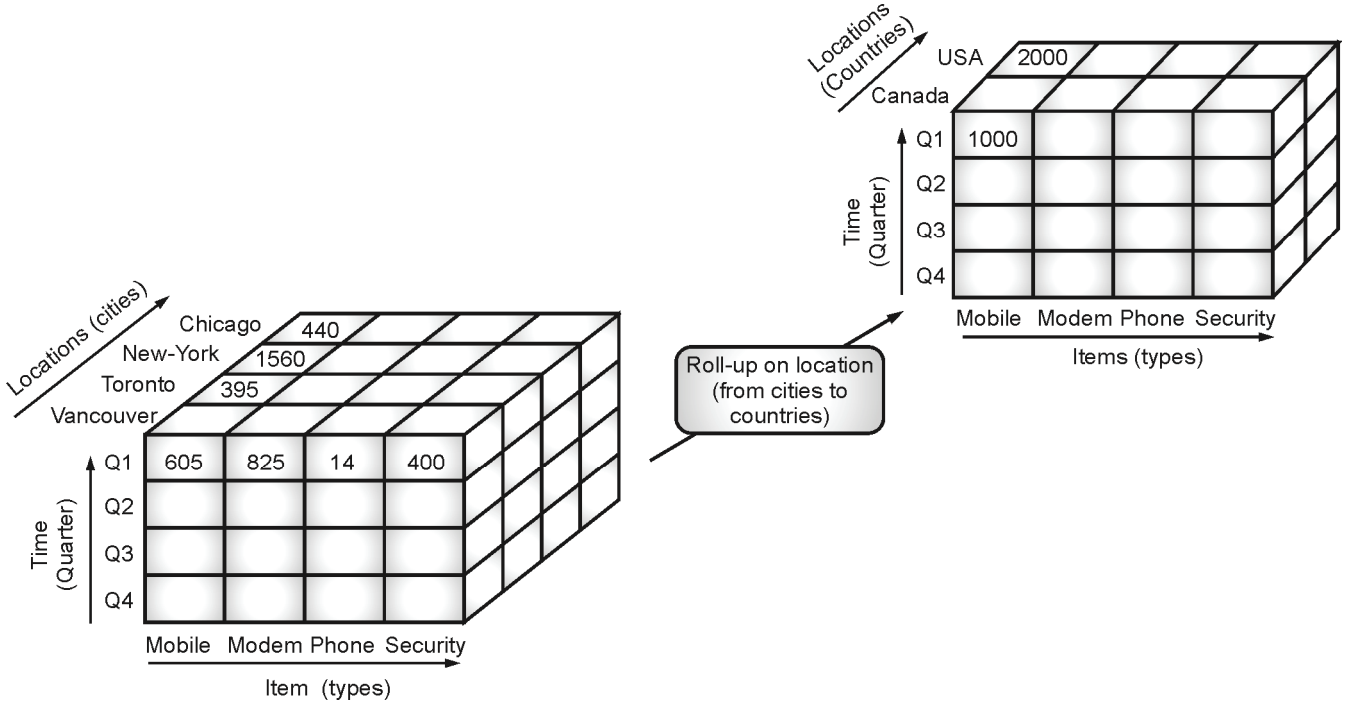
5. Pivot

- The pivot operation is also called a rotation. Pivot is a visualization operation which rotates the data axes in view to provide an alternative presentation of the data. It may contain swapping the rows and columns or moving one of the row-dimensions into the column dimensions.
- **Example :** Let's look at some typical OLAP operations for multidimensional data. Each of the following operations described is illustrated below. In the figure is a data cube for Digi1 Electronics sales. The cube contains the dimension location, time, and item, where location is aggregated with respect to city values, time is aggregated with respect to quarters, and item is aggregated with respect to item types. The measure displayed is dollars sold (in thousands). (For improved readability, only some of the cubes' cell values are shown.) The data examined are for the cities Chicago, NewYork, Toronto, and Vancouver.



(1A19)**Fig. 1.11.2: Pivot Operation on Multidimensional Cube**

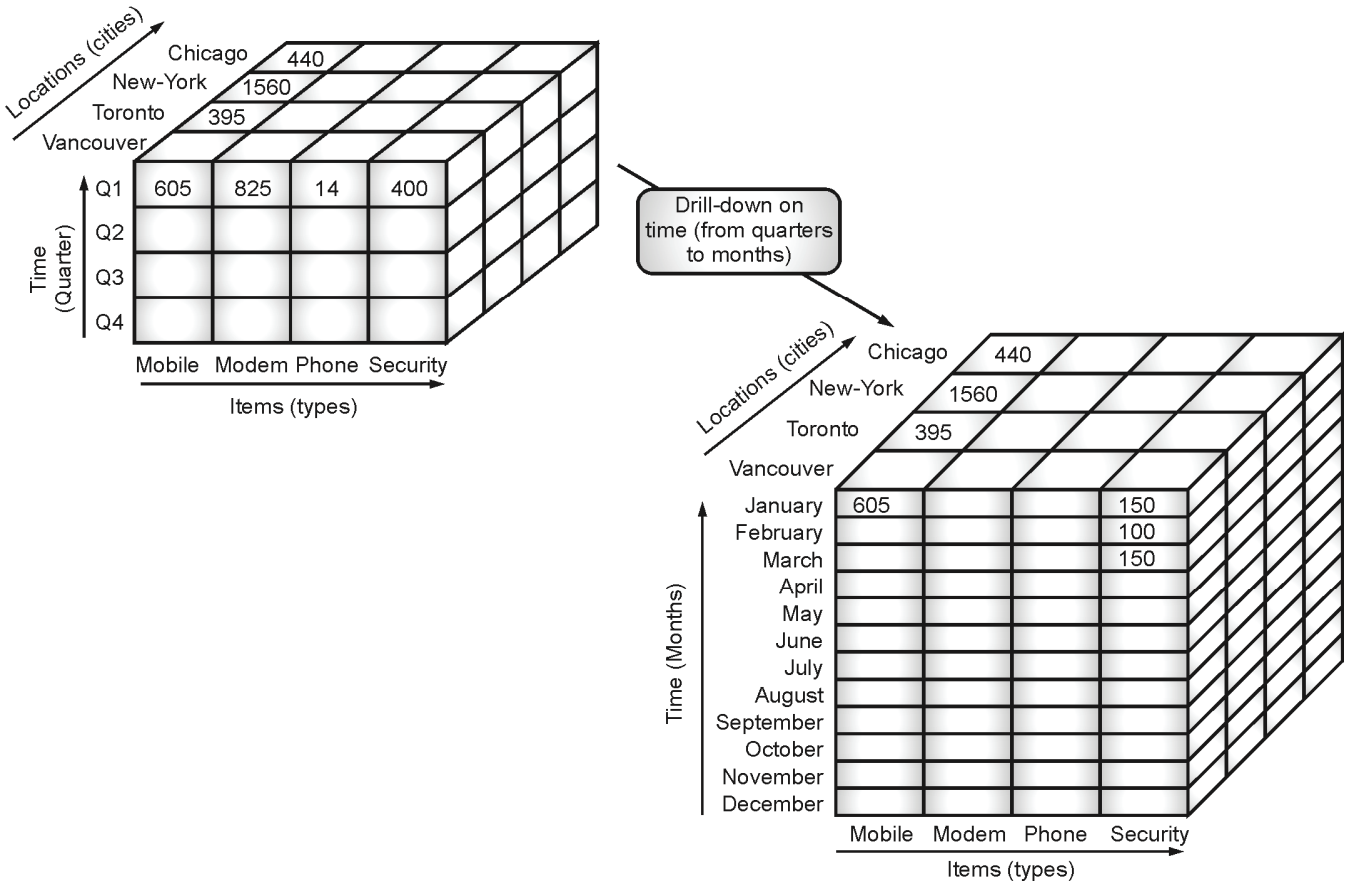
1. Roll-up Operation



(1A20) **Fig. 1.11.3: Roll-up Operation on Location Dimension**

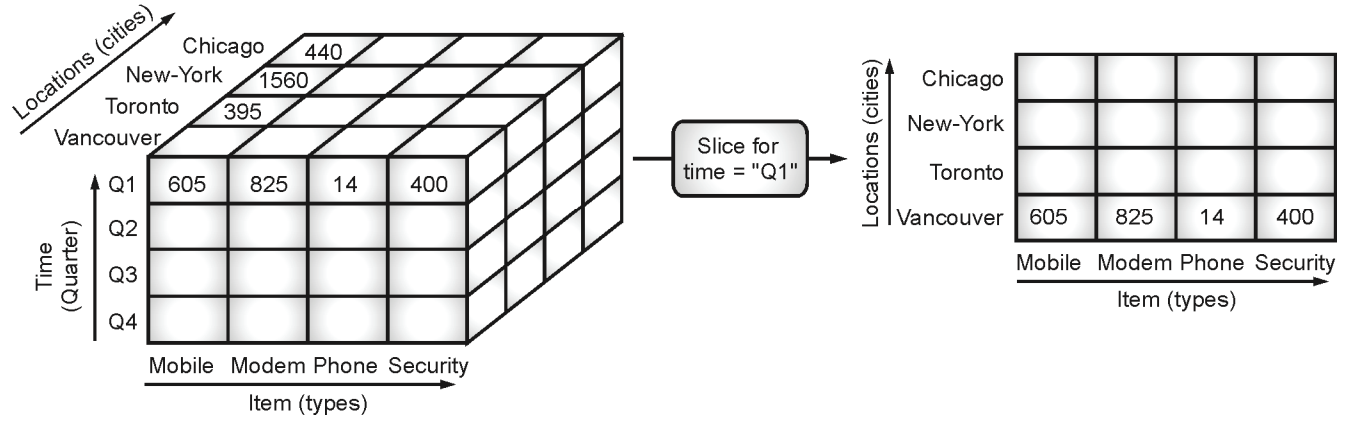
NOTES

2. Drill-down Operation



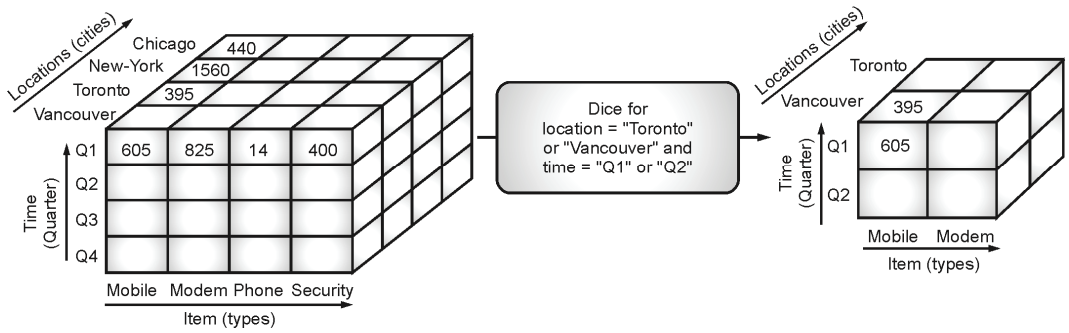
(1A21) **Fig. 1.11.4: Drill-down Operation on Time Dimension**

3. Slice Operation



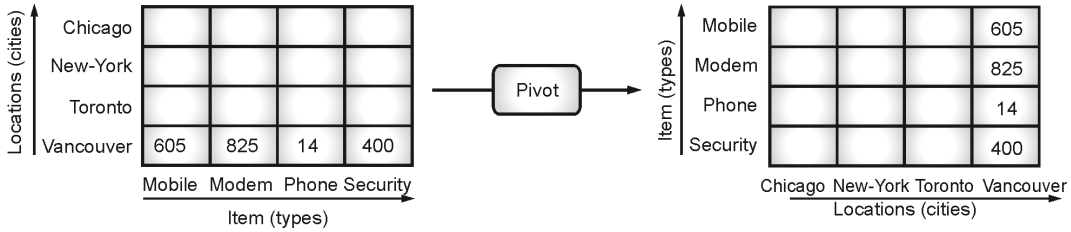
(1A22) **Fig. 1.11.5: Slice Operation for Time Dimension**

4. Dice Operation



(1A23) Fig. 1.11.6: Dice Operation for Location and Time Dimensions

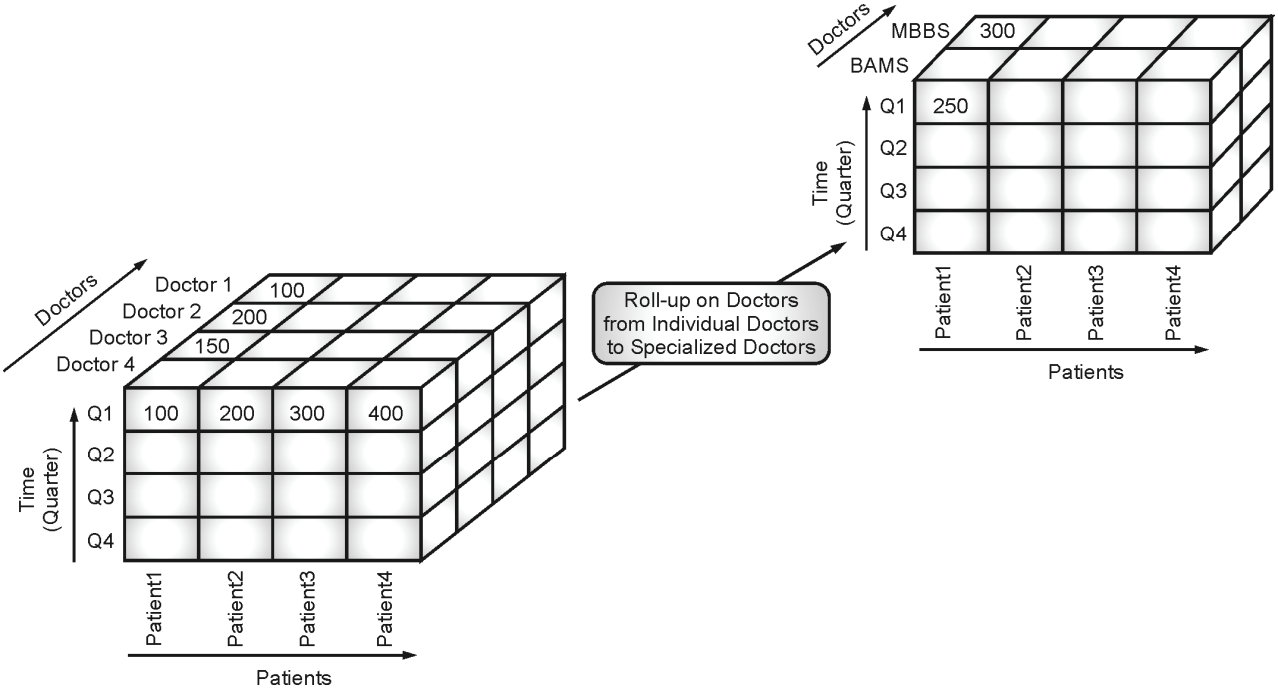
5. Pivot Operation



(1A24) Fig. 1.11.7: Pivot Operation for Location and Item Dimensions

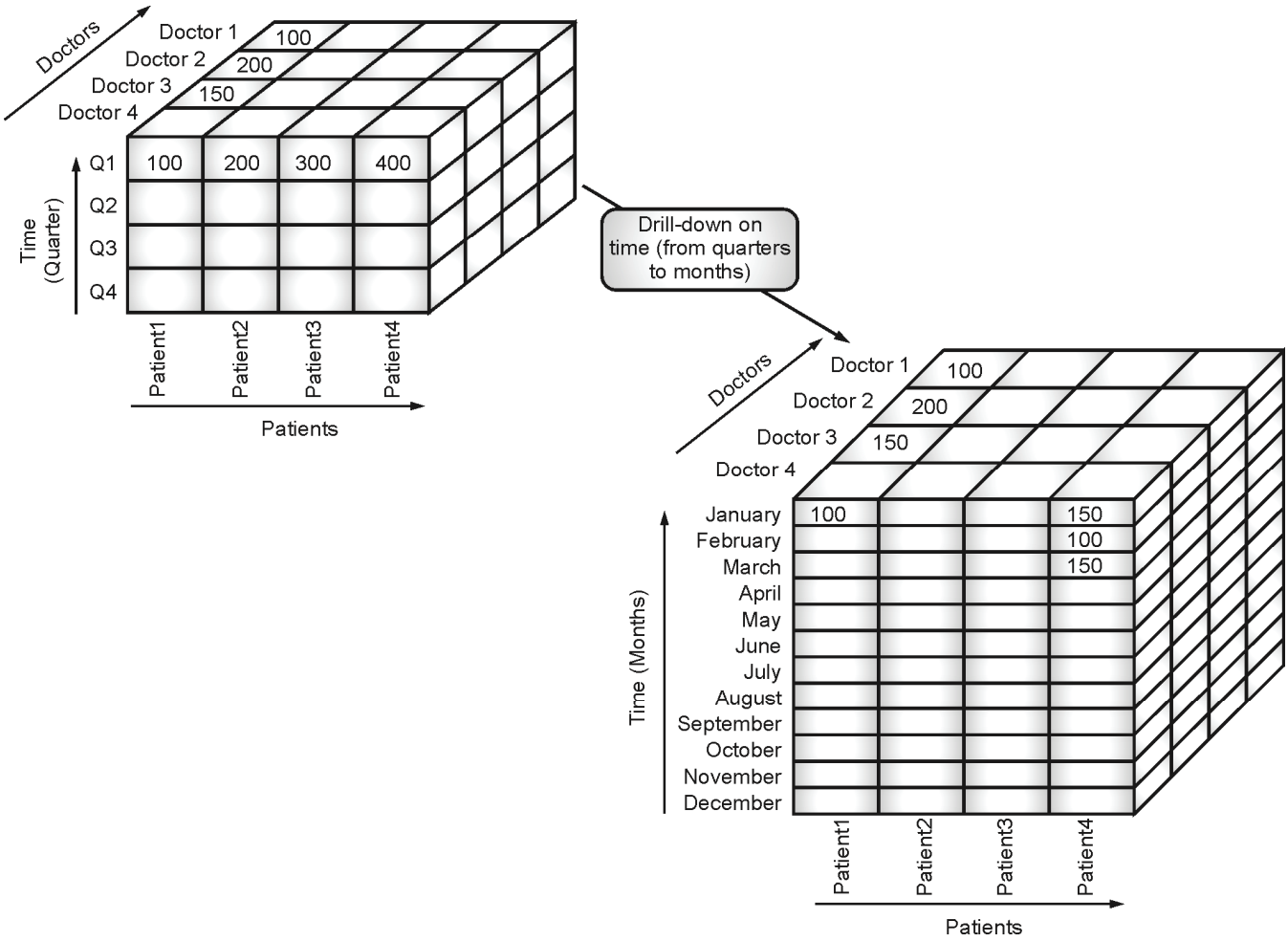
Ex. 1.11.1 : Consider a data warehouse for a hospital where there are three dimensions: a) Doctor b) Patient c) Time. Consider two measuresi) Countii) Charge where charge is the fee that the doctor charges a patient for a visit. For the above example create a cube and illustrate the following OLAP operations.1) Rollup 2) Drill down 3) Slice 4) Dice 5) Pivot.

☑ **Soln. : 1. Roll-up Operation**



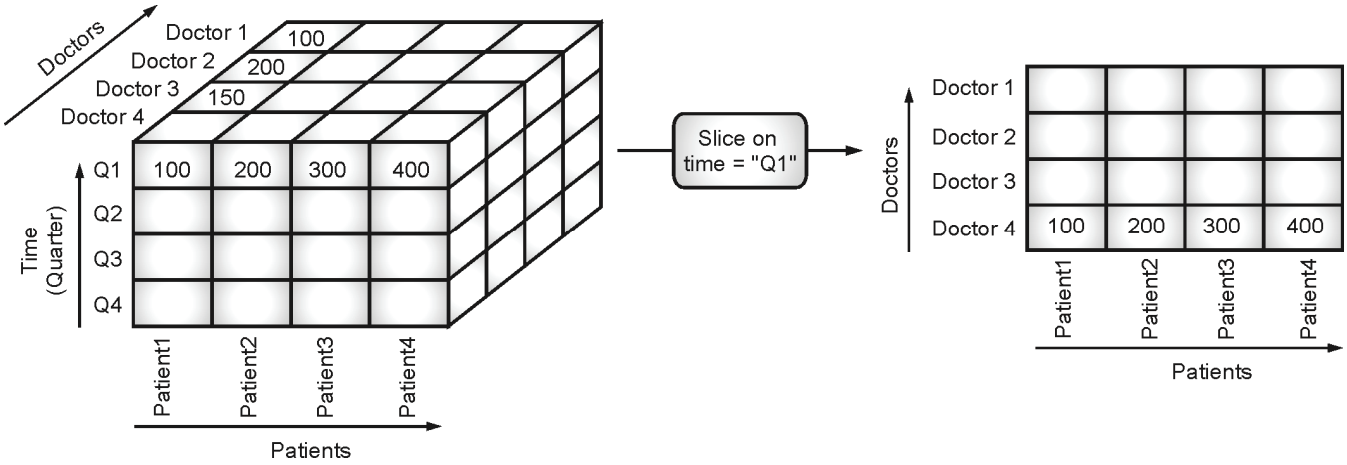
(1A25) Fig. P. 1.11.1(a): Roll-up Operation on Doctors Dimension

2. Drill-down Operation



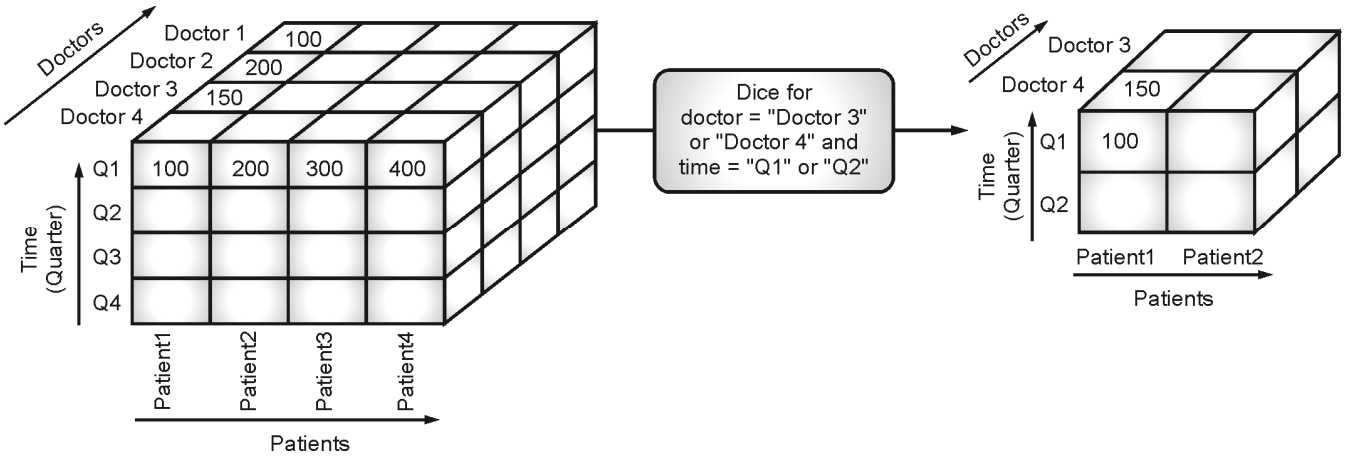
(1A26) **Fig. 1.11.1(b): Drill-down Operation on Time Dimension**

3. Slice Operation

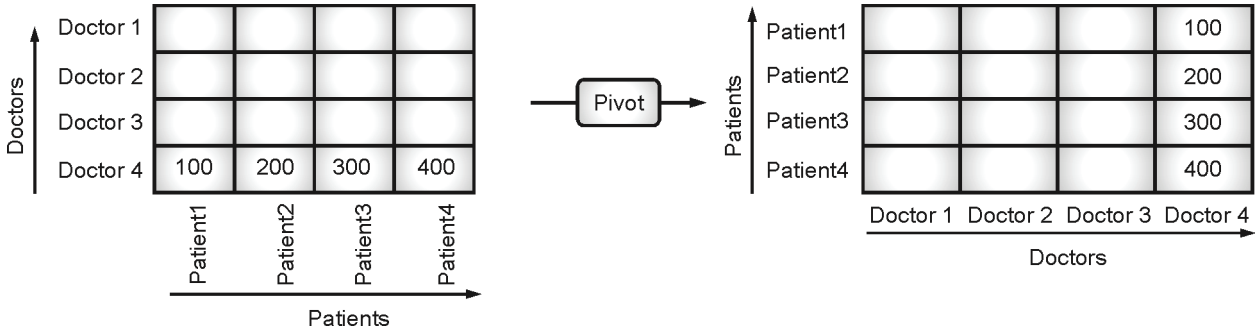


(1A27)**Fig. 1.11.1(c): Slice Operation for Time Dimension**

4. Dice Operation



(1A28) **Fig. 1.11.1(d): Dice Operation on Doctors and Time Dimensions**



(1A29) **Fig. 1.11.1(e): Pivot Operation on Doctors and Patients Dimensions**

⏭ 1.20 OLAP SERVERS

There are three types of OLAP servers, namely, Relational OLAP (ROLAP), Multidimensional OLAP (MOLAP) and Hybrid OLAP (HOLAP).

1. Relational OLAP (ROLAP)

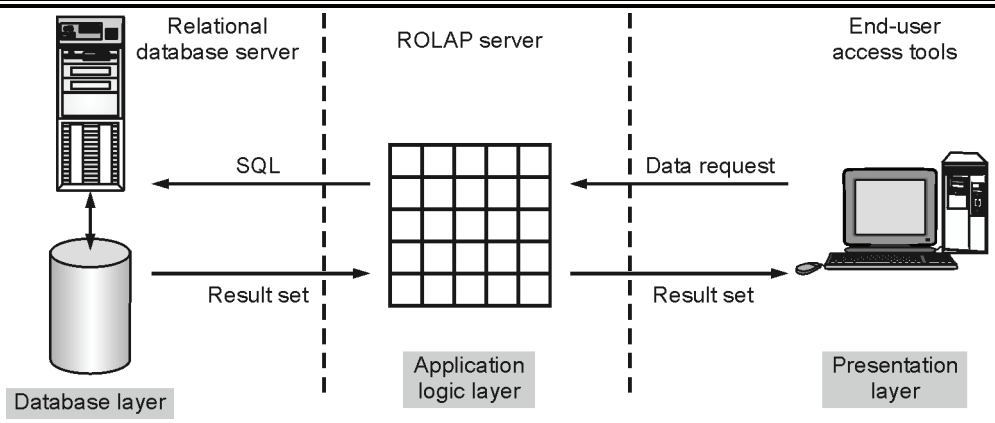
- Relational On-Line Analytical Processing (ROLAP) work mainly for the data that resides in a relational database, where the base data and dimension tables are stored as relational tables.
- ROLAP servers are placed between the relational back-end server and client front-end tools.
- ROLAP servers use RDBMS to store and manage warehouse data, and OLAP middleware to support missing pieces.
- **Example :** DSS Server of Microstrategy

☞ **Advantages of ROLAP**

1. ROLAP can handle large amounts of data.
2. Can be used with data warehouse and OLTP systems.

☞ **Disadvantages of ROLAP**

1. Limited by SQL functionalities.
2. Hard to maintain aggregate tables.



(1A30)**Fig.1.12.1 : ROLAP Server**

2. Multidimensional OLAP (MOLAP)

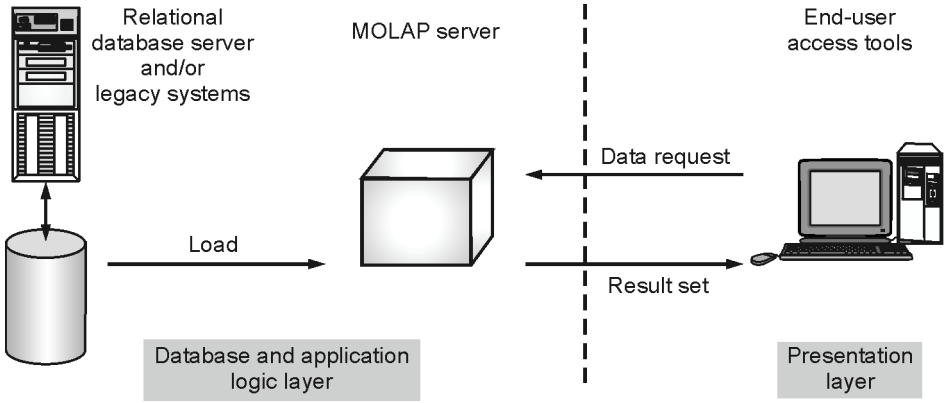
- Multidimensional On-Line Analytical Processing (MOLAP) support multidimensional views of data through array-based multidimensional storage engines.
- With multidimensional data stores, the storage utilization may be low if the data set is sparse.
- **Example :** Oracle Essbase

☞ **Advantages of MOLAP**

1. Optimal for slice and dice operations.
2. Performs better than ROLAP when data is dense.
3. Can perform complex calculations.

☞ **Disadvantages of MOLAP**

1. Difficult to change dimension without re-aggregation.
2. MOLAP can handle limited amount of data.



(1A31)**Fig.1.12.2 : MOLAP Server**

3. Hybrid OLAP (HOLAP)

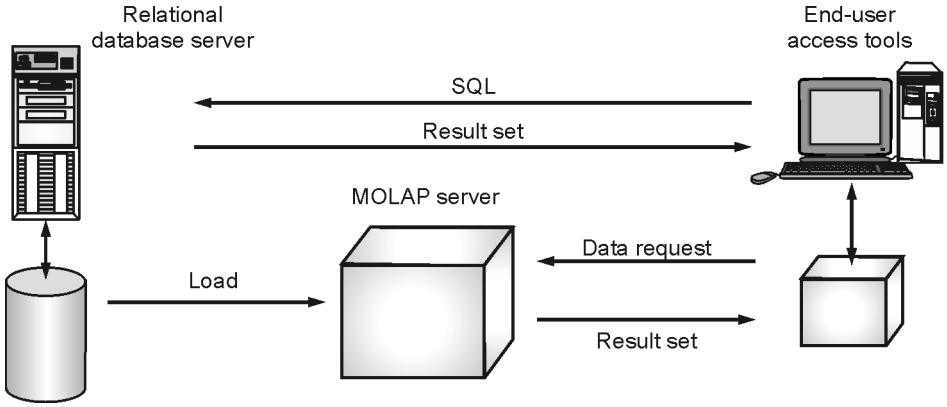
- Hybrid On-Line Analytical Processing (HOLAP) is a combination of ROLAP and MOLAP.
- HOLAP provide greater scalability of ROLAP and the faster computation of MOLAP.
- Example: Microsoft SQL Server 2000

☞ **Advantages of HOLAP**

1. HOLAP provide advantages of both MOLAP and ROLAP.
2. Provide fast access at all levels of aggregation.

☞ **Disadvantage of HOLAP**

1. HOLAP architecture is very complex because it supports both MOLAP and ROLAP servers.



(1A32)**Fig. 1.12.3: HOLAP Server**

⏭ 1.13 APPLICATIONS OF OLAP

OLAP system is to analyze the business which helps in decision-making, forecasting, planning, problem solving. Some of the applications of OLAP include:

1. Financial Applications

- Resource (man-power, raw material) allocation
- Budgeting

2. Sales Applications

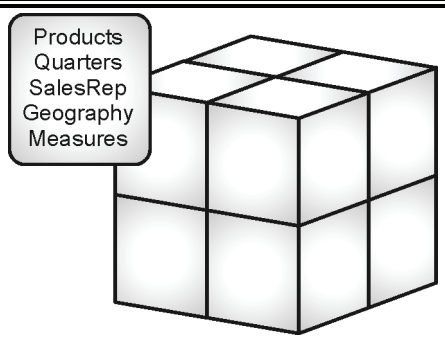
- Research on market analysis
- Forecasting sales
- Analyzing sales promotions
- Analyzing customer requirements
- Dividing market based on customer

3. Business Modelling

- Understanding and simulating the market trend and business behavior
- Decision support system for managers, executives, CEO, data scientists.

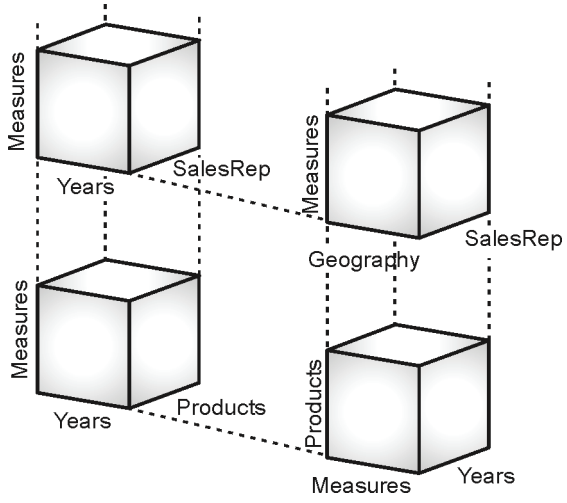
⏭ 1.14 HYPERCUBE

- Multidimensional databases can present their data for an application using two types of cubes: hypercube and multi-cubes. In a hypercube, as shown in Fig. 1.14.1, all data appears logically as a single cube. All parts of the manifold represented by this hypercube have identical dimensionality. Each dimension belongs to one cube only. A dimension is owned by the hypercube. This simplicity makes easy for users to understand.
- Designing a hypercube model is a top-down process with three major steps.
  1. You decide which process of the business you want to capture in the model, such as sales activity.
  2. You identify the values that you want to capture, such as sales amounts. This information is always numeric.
  3. You identify the granularity of the data, meaning the lowest level of detail at which you want to capture. These elements are the dimensions. And time, geography, product, and customer are some common dimensions. For example, a single cell in a cube could refer to the sales amount of Sony TVs in the first quarter of the year, in PA, USA.



(1A33)**Fig. 1.14.1 : Hypercube**

- In the multi-cube model, data is segmented into a set of smaller cubes, each of which is composed of a subset of the available dimensions, as shown in Fig. 1.14.2. They are used to handle multiple fact tables, each with different dimensionality.
- A dimension can be part of multiple cubes. Dimensions are *not owned* by any one cube, like under the hypercube model. Rather, they are available to all cubes, or there can be some dimensions that do not belong to any cube. This makes it much more efficient and versatile. It is also a more efficient way of storing very sparse data, and it can reduce the pre-calculation database explosion effect, which will be covered in a later section.
- The drawback is that this is less straightforward than hypercube and can carry steeper learning curves. Some systems use the combined approach of hypercube and multi-cubes, by separating the storage, processing, and presentation layers. It stores data as multi-cubes but presents as a hypercube.



(1A34)**Fig. 1.14.2 : Multi-cube**

1.15 AGGREGATE FACT TABLES

- Aggregate fact tables are special fact tables in a data warehouse that contain new metrics derived from one or more aggregate functions (AVERAGE, COUNT, MIN, MAX, etc.) or from other specialized functions that output totals derived from a grouping of the base data.
- These new metrics, called "aggregate facts" or "summary statistics" are stored and maintained in the data warehouse database in special fact tables at the grain of the aggregation.
- Likewise, the corresponding dimensions are rolled up and condensed to match the new grain of the fact.
- These specialized tables are used as substitutes whenever possible for returning user queries. The reason is Speed. Querying a tidy aggregate table is much faster and uses much less disk I/O than the base, atomic fact table, especially if the dimensions are large as well.
- If you want to wow your users, start adding aggregates. You can even use this "trick" in your operational systems to serve as a foundation for operational reports.
- For example, take the "Orders" business process from an online catalog company where you might have customer orders in a fact table called FactOrders with dimensions Customer, Product, and OrderDate.
- With possibly millions of orders in the transaction fact, it makes sense to start thinking about aggregates.
- To further the above example, assume that the business is interested in a report: "Monthly orders by state and product type".
- While you could generate this easily enough using the FactOrders fact table, you could likely speed up the data retrieval for the report by at least half (but likely much, much more) using an aggregate.

1.16 MULTIPLE CHOICE QUESTIONS

Q. 1.1 Among the following which is not a type of business data?

(a) Real time data (b) Application data
(c) Reconciled data (d) Derived data

✓**Ans. : (b)**

Q. 1.2 A data warehouse is which of the following?
(a) Can be updated by end users.
(b) Contains numerous naming conventions and formats.
(c) Organized around important subject areas.
(d) Contains only current data. ✓**Ans. : (c)**

Q. 1.3 An operational system is which of the following?
(a) A system that is used to run the business in real time and is based on historical data.
(b) A system that is used to run the business in real time and is based on current data.
(c) A system that is used to support decision making and is based on current data.
(d) A system that is used to support decision making and is based on historical data.
✓**Ans. : (b)**

Q. 1.4 What is the type of relationship in star schema?
(a) many-to-many (b) one-to-one
(c) many-to-one (d) one-to-many ✓**Ans. : (d)**

Q. 1.5 Fact tables are _____.
(a) completely demoralized.
(b) partially demoralized.
(c) completely normalized.
(d) partially normalized. ✓**Ans. : (c)**

Q. 1.6 Data warehouse is volatile, because obsolete data are discarded
(a) True (b) False ✓**Ans. : (b)**

Q. 1.7 Which is NOT a basic conceptual schema in Data Modeling of Data Warehouses?
(a) Star Schema (b) Tree Schema
(c) Snowflake Schema
(d) Fact Constellation Schema ✓**Ans. : (b)**

Q. 1.8 Among the followings which is not a characteristic of Data Warehouse?
(a) Integrated (b) Volatile
(c) Time-variant (d) Subject-oriented
✓**Ans. : (b)**

Q. 1.9 What is not considered as issues in data warehousing?
(a) Optimization (b) Data transformation
(c) Extraction (d) Intermediation
✓**Ans. : (d)**

Q. 1.10 Which is NOT considered as a standard querying technique?
(a) Roll-up (b) Drill-down
(c) DSS (d) Pivot ✓**Ans. : (c)**

Q. 1.11 Among the following which is not a type of business data?
(a) Real time data (b) Application data
(c) Reconciled data (d) Derived data
✓**Ans. : (b)**

Q. 1.12 A snowflake schema has which of the following types of tables?
(a) Fact (b) Dimension
(c) Helper (d) All of the above ✓**Ans. : (d)**

Q. 1.13 The extract process is which of the following?
(a) Capturing all of the data contained in various operational systems
(b) Capturing a subset of the data contained in various operational systems
(c) Capturing all of the data contained in various decision support systems
(d) Capturing a subset of the data contained in various decision support systems ✓**Ans. : (b)**

Q. 1.14 Which of the following is not true regarding characteristics of warehoused data?
(a) Changed data will be added as new data
(b) Data warehouse can contain historical data
(c) Obsolete data are discarded
(d) Users can change data once entered into the data warehouse ✓**Ans. : (d)**

Q. 1.15 Which of the following statements is incorrect?
(a) ROLAPs have large data volumes
(b) Data form of ROLAP is large multidimentional array made of cubes
(c) MOLAP uses sparse matrix technology to manage data sparcity
(d) Access for MOLAP is faster than ROLAP
✓**Ans. : (b)**

Q. 1.16 Which of the following standard query techniques increase the granularity
(a) roll-up (b) drill-down
(c) slicing (d) dicing ✓**Ans. : (b)**

Q. 1.17 The full form of OLAP is
(a) Online Analytical Processing
(b) Online Advanced Processing
(c) Online Analytical Performance
(d) Online Advanced Preparation ✓**Ans. : (a)**

Q. 1.18 _____ is a standard query technique that can be used within OLAP to zoom in to more detailed data by changing dimensions.

(a) Drill-up (b) Drill-down

(c) Pivoting (d) Drill-across ✓**Ans. : (b)**

Q. 1.19 In OLAP operations, Slicing is the technique of ____

(a) Selecting one particular dimension from a given cube and providing a new sub-cube

(b) Selecting two or more dimensions from a given cube and providing a new sub-cube

(c) Rotating the data axes in order to provide an alternative presentation of data

(d) Performing aggregation on a data cube

✓**Ans. : (a)**

Q. 1.20 Focusing on the modeling and analysis of data for decision makers, not on daily operations or transaction processing is known

(a) Integrated (b) Time-variant

(c) Subject oriented (d) Non-volatile

✓**Ans. : (c)**

Q. 1.21 The data is stored, updated and retrieved in _____.

(a) OLAP (b) OLTP

(c) SMTP (d) FTP ✓**Ans. : (b)**

Q. 1.22 _____ is a good alternative to star schema.

(a) Star schema

(b) Snowflake schema

(c) Fact constellation schema

(d) Star-snowflake schema ✓**Ans. : (c)**

Q. 1.23 The _____ exposes the information being captured, stored and managed by operational systems.

(a) Top-down view (b) Data warehouse view

(c) Data source view (d) Business query view

✓**Ans. : (b)**

Q. 1.24 The _____ allows the selection of the relevant information necessary for the data warehouse.

(a) Top-down view (b) Data warehouse view

(c) Data source view (d) Business query view

✓**Ans. : (a)**

Q. 1.25 Which of the following is not a component of a data warehouse?

(a) Metadata (b) Current detail data

(c) Lightly summarized data

(d) Component key ✓**Ans. : (d)**

Q. 1.26 Which of the following is not a kind of data warehouse application?

(a) Information Processing

(b) Analytical Processing

(c) Data Mining

(d) Transaction Processing ✓**Ans. : (d)**

Q. 1.27 The data warehouse is _____.

(a) Read only (b) Write only

(c) Read write only (d) None of the above

✓**Ans. : (a)**

Q. 1.28 The time horizon in data warehouse is usually _____.

(a) 1-2 years (b) 3-4 years

(c) 5-6 years (d) 5-10 years ✓**Ans. : (d)**

Q. 1.29 _____describes the data contained in the data warehouse.

(a) Relational data (b) Operational data

(c) Metadata (d) Informational data

✓**Ans. : (c)**

Q. 1.30 _____ is the heart of the warehouse.

(a) Data mining database servers

(b) Data warehouse database servers

(c) Data mart database servers

(d) Relational data base servers ✓**Ans. : (b)**

Q. 1.31 The star schema is composed of _____ fact table.

(a) one (b) two (c) three (d) four ✓**Ans. : (a)**

Q. 1.32 Data transformation includes _____.

(a) a process to change data from a detailed level to a summary level.

(b) a process to change data from a summary level to a detailed level.

(c) joining data from one source into various sources of data.

(d) separating data from one source into various sources of data. ✓**Ans. : (a)**

Q. 1.33 Data warehouse architecture is based on _____.

(a) DBMS (b) RDBMS

(c) Sybase (d) SQL Server ✓**Ans. : (b)**

Q. 1.34 _____ is a data transformation process.

(a) Comparison (b) Projection

(c) Selection (d) Filtering ✓**Ans. : (d)**

Q. 1.35 How many components are there in a data warehouse?

(a) two (b) three

(c) four (d) five ✓**Ans. : (d)**

Descriptive Questions

Q. 1 Define data warehouse. Explain the characteristics of data warehouse.

Q. 2 Why is data integration required in a data warehouse more so than in operational application?

(MU - Dec. 2019)

Q. 3 Explain top-down and bottom-up approach of data warehouse design.

Q. 4 Explain with neat diagram the architecture of data warehouse.

Q. 5 What is metadata? Explain different types of metadata.

Q. 6 What is Metadata? Why do we need metadata when search engines like Google seem so effective?

(MU - May-2019)

Q. 7 Compare data warehouse versus data mart.

Q. 8 Compare E-R modeling versus dimensional modeling.

Q. 9 Why is entity-relationship modeling technique not suitable for the data warehouse? How is dimensional modeling different? **(MU - DEC-2019)**

Q. 10 What is the relationship between data warehousing and data replication? Which form of replication (synchronous or asynchronous) is better suited for data warehousing? Why? Explain with appropriate example. **(MU- May 2019)**

Q. 11 Explain Information Package Diagram with suitable example.

Q. 12 Explain Star Schema with suitable example.

Q. 13 Explain different keys in star schema.

Q. 14 Explain Snowflake Schema with suitable example.

Q. 15 Compare Star Schema versus Snowflake Schema.

Q. 16 A dimension table is wide, the fact table is deep. Explain. **(MU - Dec. 2019)**

Q. 17 Consider a data warehouse for a hospital where there are three dimensions namely (a) Doctor (b) Patient (c) Time and two measures (i) count (ii) charge where charge is the fee that the doctor charges a patient for a visit.

(i) Draw star and snowflake schema.

(ii) Starting with the base cuboid [day, doctor, patient], what specific OLAP operations should be performed in order to list the total fee collected by each doctor in 2010?

(iii) To obtain the same list, write an SQL query assuming the data are stored in a relational database with the schema fee (day, month, year, doctor, hospital, patient, count, charge).

(MU - Dec. 2019)

Q. 18 Suppose that a data warehouse for DB-University consists of the following four dimensions: student, course, semester, and instructor, and two measures count and avg_grade. When at the lowest conceptual level (e.g., for a given student, course, semester, and instructor combination), the avg_grade measure stores the actual course grade of the student. At higher conceptual levels, avg_grade stores the average grade for the given combination.

(a) Draw a snowflake schema diagram for the data warehouse.

(b) Starting with the base cuboid [student, course, semester, instructor], what specific OLAP operations (e.g., roll-up from semester to year) should one perform in order to list the average grade of CS courses for each DB_University student. **(MU- May 2019)**

Q. 19 For a supermarket chain, consider the following dimensions namely product, store, time and promotion. The schema contains a central fact table for sales with three measures unit_sales, dollars_sales and dollar_cost.

(i) Draw star schema.

(ii) Calculate the maximum number of base fact table records for warehouse with the following values given below:

Time period-5 years

Store - 300 stores reporting daily sales

Product - 40,000 products in each store (about 4000 sell in each store daily).

Promotion: a sold item may be in only one promotion in a store on a given day.

(MU - June 2021)

Q. 20 Explain slowly changing dimensions in data warehouse.

Q. 21 Explain different types of dimensions in data warehouse.

Q. 22 Explain the major steps in ETL process.

Q. 23 Compare OLTP versus OLAP.

Q. 24 Explain different OLAP operations with suitable example.

Q. 25 Explain different OLAP Servers with suitable example.

Q. 26 Explain Hypercube in Data Warehouse with example.

Q. 27 Explain Aggregate Fact tables and Factless Fact tables.

Chapter Ends…

❑❑❑

MODULE 2

CHAPTER 2

Introduction to Data Mining, Data Exploration and Data Pre-Processing

University Prescribed Syllabus w.e.f Academic Year 2021-2022

Data Mining Task Primitives, Architecture, KDD process, Issues in Data Mining, Applications of Data Mining, Data Exploration: Types of Attributes, Statistical Description of Data, Data Visualization, Data Preprocessing : Descriptive data summarization, Cleaning, Integration & transformation, Data reduction, Data Discretization and Concept hierarchy generation.

2.1 INTRODUCTION TO DATA MINING

- There is a huge amount of data available in the Information Industry. This data is of no use until it is converted into useful information. It is necessary to analyze this huge amount of data and extract useful information from it.
- William J Frawley, Gregory Piatetsky-Shapiro and Christopher J Matheus define **data mining** as **"The non-trivial extraction of implicit, previously unknown, and potentially useful information from data."**
- The process of extracting information to identify patterns, trends, and useful data that would allow the business to take the data-driven decision from huge sets of data is called Data Mining.
- The computer is responsible for finding the patterns by identifying the underlying rules and features in the data.
- In other words, we can say that Data Mining is the process of investigating hidden patterns of information to various perspectives for categorization into useful data, which is collected and assembled in particular areas such as data warehouses, efficient analysis, data mining algorithm, helping decision making and other data requirement to eventually cost-cutting and generating revenue.

2.1.1 Sources of Data that can be Mined

The data from multiple sources are integrated into a common source known as Data Warehouse. Let's discuss what type of data can be mined:

1. Flat Files

- Flat files are defined as data files in text form or binary form with a structure that can be easily extracted by data mining algorithms.
- Data stored in flat files have no relationship or path among themselves, like if a relational database is stored on flat file, then there will be no relations between the tables.
- Flat files are represented by data dictionary. E.g.: CSV file.
- **Application :** Used in Data Warehousing to store data, used in carrying data to and from server, etc.

2. Relational Databases

- A Relational database is defined as the collection of data organized in tables with rows and columns.
- Physical schema in Relational databases is a schema which defines the structure of tables.
- Logical schema in Relational databases is a schema which defines the relationship among tables.
- Standard API of relational database is SQL.
- **Application**: Data Mining, ROLAP model, etc.

3. Data Warehouse

- A data warehouse is defined as the collection of data integrated from multiple sources that will queries and decision making.
- There are three types of data warehouse: Enterprise data warehouse, Data Mart and Virtual Warehouse.
- Two approaches can be used to update data in Data Warehouse: Query-driven Approach and Update-driven Approach.
- **Application**: Business decision making, Data mining, etc.

4. Transactional Databases

- Transactional database is a collection of data organized by time stamps, date, etc. to represent transaction in databases.
- This type of database has the capability to roll back or undo its operation when a transaction is not completed or committed.
- Highly flexible system where users can modify information without changing any sensitive information.
- Follows ACID (Atomicity, Consistency, Isolation and Durability) property of DBMS.
- **Application**: Banking, Distributed systems, Object databases, etc.

5. Multimedia Databases

- Multimedia databases consists audio, video, images and text media.
- They can be stored on Object-Oriented Databases.
- They are used to store complex information in a pre-specified formats.

- **Application**: Digital libraries, video-on demand, news-on demand, musical database, etc.

6. Spatial Database

- Store geographical information.
- Stores data in the form of coordinates, topology, lines, polygons, etc.
- **Application**: Maps, Global positioning, etc.

7. Time-series Databases

- Time series databases contains stock exchange data and user logged activities.
- Handles array of numbers indexed by time, date, etc.
- It requires real-time analysis.
- **Application :** eXtremeDB, Graphite, InfluxDB, etc.

8. WWW

- WWW refers to World wide web is a collection of documents and resources like audio, video, text, etc. which are identified by Uniform Resource Locators (URLs) through web browsers, linked by HTML pages, and accessible via the Internet network.
- It is the most heterogeneous repository as it collects data from multiple resources.
- It is dynamic in nature as Volume of data is continuously increasing and changing.
- **Application :** Online shopping, Job search, Research, studying, etc.

2.1.2 Data Mining Techniques

Data mining is highly effective, so long as it draws upon one or more of these techniques:

1. **Tracking patterns :** One of the most basic techniques in data mining is learning to recognize patterns in your data sets. This is usually a recognition of some aberration in your data happening at regular intervals, or an ebb and flow of a certain variable over time. For example, you might see that your sales of a certain product seem to spike just before the holidays, or notice that warmer weather drives more people to your website.
2. **Classification :** Classification is a more complex data mining technique that forces you to collect various attributes together into discernible categories, which you can then use to draw further conclusions, or serve some function. For example, if you're evaluating data on individual customers' financial backgrounds and purchase histories, you might be able to classify them as "low," "medium," or "high" credit risks. You could then use these classifications to learn even more about those customers.
3. **Association :** Association is related to tracking patterns, but is more specific to dependently linked variables. In this case, you'll look for specific events or attributes that are highly correlated with another event or attribute; for example, you might notice that when your customers buy a specific item, they also often buy a second, related item. This is usually what's used to populate "people also bought" sections of online stores.
4. **Outlier detection :** In many cases, simply recognizing the overarching pattern can't give you a clear understanding of your data set. You also need to be able to identify anomalies, or outliers in your data. For example, if your purchasers are almost exclusively male, but during one strange week in July, there's a huge spike in female purchasers, you'll want to investigate the spike and see what drove it, so you can either replicate it or better understand your audience in the process.
5. **Clustering :** Clustering is very similar to classification, but involves grouping chunks of data together based on their similarities. For example, you might choose to cluster different demographics of your audience into different packets based on how much disposable income they have, or how often they tend to shop at your store.
6. **Regression :** Regression, used primarily as a form of planning and modeling, is used to identify the likelihood of a certain variable, given the presence of other variables. For example, you could use it to project a certain price, based on other factors like availability, consumer demand, and competition. More specifically, regression's main focus is to help you uncover the exact relationship between two (or more) variables in a given data set.
7. **Prediction :** Prediction is one of the most valuable data mining techniques, since it's used to project the types of data you'll see in the future. In many cases, just recognizing and understanding historical trends is

enough to chart a somewhat accurate prediction of what will happen in the future. For example, you might review consumers' credit histories and past purchases to predict whether they will be a credit risk in the future.

2.1.3 Difference between Data Mining and Data Warehouse

Table 2.1.1 : Data Mining Vs Data Warehouse

Sr. No.	Data Mining	Data Warehouse
1.	Data mining is the process of analyzing unknown patterns of data.	A data warehouse is database system which is designed for analytical instead of transactional work.
2.	Data mining is a method of comparing large amounts of data to finding right patterns.	Data warehousing is a method of centralizing data from different sources into one common repository.
3.	Data mining is usually done by business users with the assistance of engineers.	Data warehousing is a process which needs to occur before any data mining can take place.
4.	Data mining is the considered as a process of extracting data from large data sets.	On the other hand, Data warehousing is the process of pooling all relevant data together.
5.	One of the most important benefits of data mining techniques is the detection and identification of errors in the system.	One of the pros of Data Warehouse is its ability to update consistently. That's why it is ideal for the business owner who wants the best and latest features.
6.	Data mining helps to create suggestive patterns of important factors. Like the buying habits of customers, products, sales. So that, companies can make the necessary adjustments in operation and production.	Data Warehouse adds an extra value to operational business systems like CRM systems when the warehouse is integrated.
7.	The Data mining techniques are never 100% accurate and may cause serious consequences in certain conditions.	In the data warehouse, there is great chance that the data which was required for analysis by the organization may not be integrated into the warehouse. It can easily lead to loss of information.
8.	The information gathered based on Data Mining by organizations can be misused against a group of people.	Data warehouses are created for a huge IT project. Therefore, it involves high maintenance system which can impact the revenue of medium to small-scale organizations.
9.	After successful initial queries, users may ask more complicated queries which would increase the workload.	Data Warehouse is complicated to implement and maintain.
10	Organisations can benefit from this analytical tool by equipping pertinent and usable knowledge-based information.	Data warehouse stores a large amount of historical data which helps users to analyze different time periods and trends for making future predictions.
11.	Organisations need to spend lots of their resources for training and Implementation purpose. Moreover, data mining tools work in different manners due to different algorithms employed in their design.	In Data warehouse, data is pooled from multiple sources. The data needs to be cleaned and transformed. This could be a challenge.
12.	The data mining methods are cost-effective and efficient compares to other statistical data applications.	Data warehouse's responsibility is to simplify every type of business data. Most of the work that will be done on user's part is inputting the raw data.

Sr. No.	Data Mining	Data Warehouse
13.	Another critical benefit of data mining techniques is the identification of errors which can lead to losses. Generated data could be used to detect a drop-in sale.	Data warehouse allows users to access critical data from the number of sources in a single place. Therefore, it saves user's time of retrieving data from multiple sources.
14.	Data mining helps to generate actionable strategies built on data insights.	Once you input any information into Data warehouse system, you will unlikely to lose track of this data again. You need to conduct a quick search, helps you to find the right statistic information.

2.2 DATA MINING TASK PRIMITIVES

- Each user will have a data mining task in mind, that is, some form of data analysis that he or she would like to have performed.
- A data mining task can be specified in the form of a data mining query, which is input to the data mining system.
- A data mining query is defined in terms of data mining task primitives.
- These primitives allow the user to interactively communicate with the data mining system during discovery in order to direct the mining process, or examine the findings from different angles or depths.
- Here is the list of Data Mining Task Primitives.

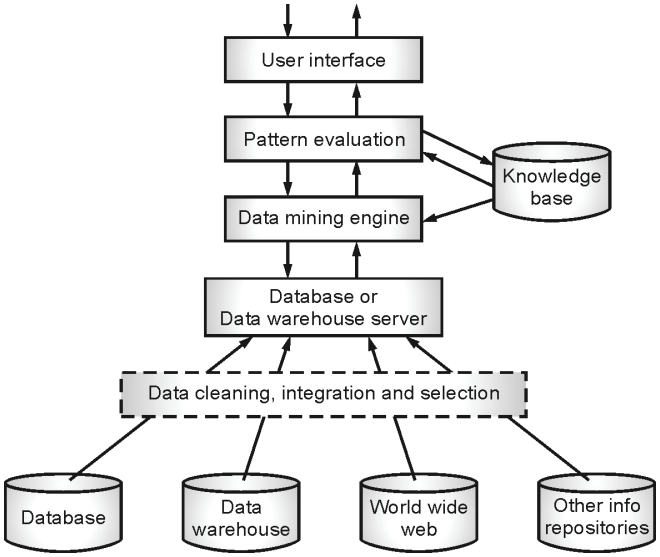
1. **The set of task-relevant data to be mined :** This specifies the portions of the database or the set of data in which the user is interested. This includes the database attributes or data warehouse dimensions of interest (referred to as the relevant attributes or dimensions).
2. **The kind of knowledge to be mined :** This specifies the data mining functions to be performed, such as characterization, discrimination, association or correlation analysis, classification, prediction, clustering, outlier analysis, or evolution analysis.
3. **The background knowledge to be used in the discovery process :** This knowledge about the domain to be mined is useful for guiding the knowledge discovery process and for evaluating the patterns found. Concept hierarchies are a popular form of background knowledge, which allow data to be mined at multiple levels of abstraction.
4. **The interestingness measures and thresholds for pattern evaluation :** They may be used to guide the mining process or, after discovery, to evaluate the discovered patterns. Different kinds of knowledge may have different interestingness measures. For example, interestingness measures for association rules include support and confidence. Rules whose support and confidence values are below user-specified thresholds are considered uninteresting.
5. **The expected representation for visualizing the discovered patterns :** This refers to the form in which discovered patterns are to be displayed, which may include rules, tables, charts, graphs, decision trees, and cubes.

2.3 DATA MINING ARCHITECTURE

The significant components of data mining systems are a data source, data mining engine, data warehouse server, the pattern evaluation module, graphical user interface, and knowledge base.

Data Source

- The actual source of data is the Database, data warehouse, World Wide Web (WWW), text files, and other documents. You need a huge amount of historical data for data mining to be successful.
- Organizations typically store data in databases or data warehouses. Data warehouses may comprise one or more databases, text files spreadsheets, or other repositories of data.
- Sometimes, even plain text files or spreadsheets may contain information. Another primary source of data is the World Wide Web or the internet.



(1B1)**Fig. 2.3.1: Data Mining Architecture**

☞ Data Cleaning, Integration and Selection

- Before passing the data to the database or data warehouse server, the data must be cleaned, integrated and selected.
- As the information comes from various sources and in different formats, it can't be used directly for the data mining procedure because the data may not be complete and accurate. So, the first data requires to be cleaned and unified.
- More information than needed will be collected from various data sources, and only the data of interest will have to be selected and passed to the server. These procedures are not as easy as we think. Several methods may be performed on the data as part of selection, integration, and cleaning.

☞ Database or Data Warehouse Server

The database or data warehouse server consists of the original data that is ready to be processed. Hence, the server is cause for retrieving the relevant data that is based on data mining as per user request.

☞ Data Mining Engine

- The data mining engine is a major component of any data mining system. It contains several modules for operating data mining tasks, including association, characterization, classification, clustering, prediction, time-series analysis, etc.
- In other words, we can say data mining is the root of our data mining architecture. It comprises instruments and software used to obtain insights and knowledge from data collected from various data sources and stored within the data warehouse.

☞ Pattern Evaluation Module

- The Pattern evaluation module is primarily responsible for the measure of investigation of the pattern by using a threshold value. It collaborates with the data mining engine to focus the search on exciting patterns.
- This segment commonly employs stake measures that cooperate with the data mining modules to focus the search towards fascinating patterns. It might utilize a stake threshold to filter out discovered patterns.
- On the other hand, the pattern evaluation module might be coordinated with the mining module, depending on the implementation of the data mining techniques used. For efficient data mining, it is abnormally suggested to push the evaluation of pattern stake as much as possible into the mining procedure to confine the search to only fascinating patterns.

☞ Graphical User Interface

- The graphical user interface (GUI) module communicates between the data mining system and the user. This module helps the user to easily and efficiently use the system without knowing the complexity of the process.
- This module cooperates with the data mining system when the user specifies a query or a task and displays the results.

☞ Knowledge Base

- The knowledge base is helpful in the entire process of data mining. It might be helpful to guide the search or evaluate the stake of the result patterns.
- The knowledge base may even contain user views and data from user experiences that might be helpful in the data mining process.
- The data mining engine may receive inputs from the knowledge base to make the result more accurate and reliable.
- The pattern assessment module regularly interacts with the knowledge base to get inputs, and also update it.

GQ. Suppose your task as a software engineer at DB-University is to design a data mining system to examine their university course database, which contains the following information: the name, address, and status (e.g., undergraduate or graduate) of each student, the courses taken, and their cumulative grade point average (GPA). Describe the architecture you would choose. What is the purpose of each component of this architecture?

A data mining architecture that can be used for this application would consist of the following major components :

- A database, data warehouse, or other information repository, which consists of the set of databases, data warehouses, spreadsheets, or other kinds of information repositories containing the student and course information.
- A database or data warehouse server which fetches the relevant data based on users' data mining requests.
- A knowledge base that contains the domain knowledge used to guide the search or to evaluate the interestingness of resulting patterns. For example, the knowledge base may contain metadata which describes data from multiple heterogeneous sources.
- A data mining engine, which consists of a set of functional modules for tasks such as classification, association, classification, cluster analysis, and evolution and deviation analysis.
- A pattern evaluation module that works in tandem with the data mining modules by employing interestingness measures to help focus the search towards interestingness patterns.
- A graphical user interface that allows the user an interactive approach to the data mining system.

2.4 KDD PROCESS

UQ. Describe the steps involved in Data Mining when viewed as a process of Knowledge Discovery.

MU - Dec. 2019

- Knowledge discovery in the database (KDD) is the process of searching for hidden knowledge in the massive amounts of data that we are technically capable of generating and storing.
- The basic task of KDD is to extract knowledge (or information) from a lower level data (databases).
- It is the non-trivial (significant) process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.
- The goal is to distinguish between unprocessed data something that may not be obvious but is valuable or enlightening in its discovery.
- The overall process of finding and interpreting patterns from data involves the repeated application of the following steps :

1. Data Cleaning

- Removal of noise, inconsistent data, and outliers
- Strategies to handle missing data fields.

2. Data Integration

- Data from various sources such as databases, data warehouse, and transactional data are integrated.
- Multiple data sources may be combined into a single data format.

3. Data Selection

- Data relevant to the analysis task is retrieved from the database.
- Collecting only necessary information to the model.
- Finding useful features to represent data depending on the goal of the task.

4. Data Transformation

- Data are transformed and consolidated into forms appropriate for mining by performing summary or aggregation operations.
- By using transformation methods invariant representations for the data is found.

5. Data Mining

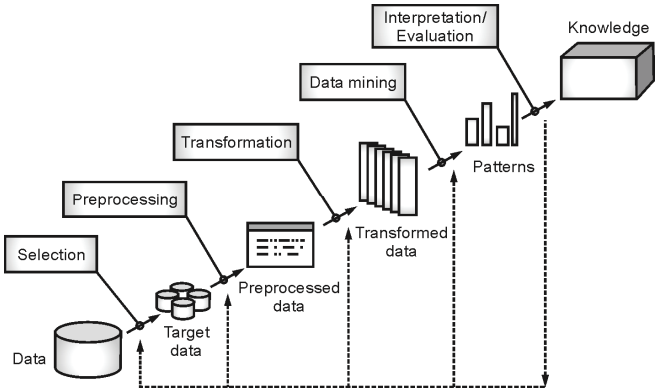
- An essential process where intelligent methods are applied to extract data patterns.
- Deciding which model and parameter may be appropriate.

6. Pattern Evaluation

To identify the truly interesting patterns representing knowledge based on interesting measures.

7. Knowledge Presentation

- Visualization and knowledge representation techniques are used to present mined knowledge to users.
- Visualizations can be in form of graphs, charts or table.

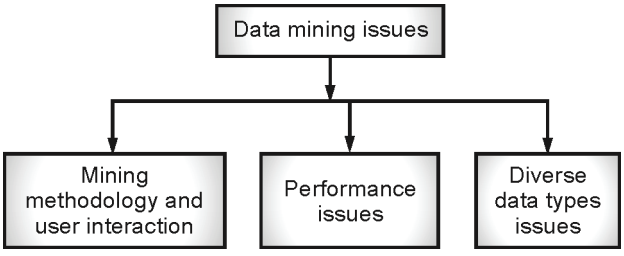


(1B2)**Fig. 2.4.1: KDD Process**

2.5 ISSUES IN DATA MINING

Data mining systems face a lot of challenges and issues in today's world. Some of them are:

1. Mining methodology and user interaction issues
2. Performance issues
3. Issues relating to the diversity of database types



(1B3)**Fig. 2.5.1 : Data Mining Issues**

1. Mining Methodology and User Interaction Issues

The issues in this category are as follows:

- **Mining different kinds of knowledge in databases :** This issue is responsible for addressing the problems of covering a big range of data in order to meet the needs of the client or the customer. Due to the different information or a different way, it becomes difficult for a user to cover a big range of knowledge discovery task.
- **Interactive mining of knowledge at multiple levels of abstraction :** Interactive mining is very crucial because it permits the user to focus the search for patterns, providing and refining data mining requests based on the results that were returned. In simpler words, it allows user to focus the search on patterns from various different angles.
- **Incorporation of background of knowledge :** The main work of background knowledge is to continue the process of discovery and indicate the patterns or trends that were seen in the process. Background knowledge can also be used to express the patterns or trends observed in brief and precise terms. It can also be represented at different levels of abstraction.
- **Data mining query languages and ad hoc data mining :** Data Mining Query language is responsible for giving access to the user such that it describes ad hoc mining tasks as well and it needs to be integrated with a data warehouse query language.
- **Presentation and visualization of data mining results:** In this issue, the patterns or trends that are discovered are to be rendered in high level languages and visual representations. The representation has to be written so that it is simply understood by everyone.
- **Handling noisy or incomplete data:** For this process, the data cleaning methods are used. It is a convenient way of handling the noise and the incomplete objects in data mining. Without data cleaning methods, there will be no accuracy in the discovered patterns. And then these patterns will be poor in quality.

2. Performance Issues

It has been noticed several times there are performance related issues in data mining as well. These issues are as follows:

- **Efficiency and Scalability of data mining algorithm :** Efficiency and Scalability is very important when it comes to data mining process. It is also very necessary because with the help of using this, the user can withdraw the information from the data in a more effective and productive manner. On top of

that, the user can withdraw that information effectively from the large amount of data in various databases.

- **Parallel, distributed and incremental mining algorithm :** There are a lot factors which can be responsible for the development of parallel and distributed algorithms in data mining. These factors are large in size of database, huge distribution of data, and data mining method that are complex. In this process, in the first and foremost step, the algorithm divides the data from database into various partition. In the next step, that data is processed such that it is situated in parallel manner. Then in the last step, the result from the partition is merged.

3. Diverse Data Types Issues

The issues in this category are given below:

- **Handling of relational and complex types of data :** The database may contain the various data objects. For example, complex, multimedia, temporal data, or spatial data objects. It is very difficult to mine all these data with the help of a single system.
- **Mining information from heterogeneous databases and global information systems :** The problem in this kind of issue is to mine the knowledge from various data sources. These data are not available as a single source instead these data are available at the different data sources on LAN or WAN. The structures of these data are different as well.

2.6 APPLICATIONS OF DATA MINING

The importance of data mining and analysis is growing day by day in our real life. Today most organizations use data mining for analysis of Big Data. Let us see how this technology benefit different users.

1. Mobile Service Providers

- Mobile service providers use data mining to design their marketing campaigns and to retain customers from moving to other vendors.
- From a large amount of data such as billing information, email, text messages, web data transmissions, and customer service, the data mining tools can predict "churn" that tells the customers who are looking to change the vendors.
- With these results, a probability score is given. The mobile service providers are then able to provide incentives, offers to customers who are at higher risk of churning. This kind of mining is often used by major service providers such as broadband, phone, gas providers, etc.

2. Retail Sector

- Data Mining helps the supermarket and retail sector owners to know the choices of the customers. Looking at the purchase history of the customers, the data mining tools show the buying preferences of the customers.
- With the help of these results, the supermarkets design the placements of products on shelves and bring out offers on items such as coupons on matching products, and special discounts on some products.
- These campaigns are based on RFM grouping. RFM stands for recency, frequency, and monetary grouping. The promotions and marketing campaigns are customized for these segments. The customer who spends a lot but very less frequently will be treated differently from the customer who buys every 2-3 days but of less amount.
- Data Mining can be used for product recommendation and cross-referencing of items.

3. Artificial Intelligence

- A system is made artificially intelligent by feeding it with relevant patterns. These patterns come from data mining outputs. The outputs of the artificially intelligent systems are also analysed for their relevance using the data mining techniques.
- The recommender systems use data mining techniques to make personalized recommendations when the customer is interacting with the machines. The artificial intelligence is used on mined data such as giving product recommendations based on the past purchasing history of the customer in Amazon.

4. E-commerce

- Many E-commerce sites use data mining to offer cross-selling and upselling of their products. The shopping sites such as Amazon, Flipkart show "People also viewed", "Frequently bought together" to the customers who are interacting with the site.

- These recommendations are provided using data mining over the purchasing history of the customers of the website.

5. Science and Engineering

- With the advent of data mining, scientific applications are now moving from statistical techniques to using "collect and store data" techniques, and then perform mining on new data, output new results and experiment with the process. A large amount of data is collected from scientific domains such as astronomy, geology, satellite sensors, global positioning system, etc.
- Data mining in computer science helps to monitor system status, improve its performance, find out software bugs, discover plagiarism and find out faults. Data mining also helps in analyzing the user feedback regarding products, articles to deduce opinions and sentiments of the views.

6. Crime Prevention

- Data Mining detects outliers across a vast amount of data. The criminal data includes all details of the crime that has happened. Data Mining will study the patterns and trends and predict future events with better accuracy.
- The agencies can find out which area is more prone to crime, how much police personnel should be deployed, which age group should be targeted, vehicle numbers to be scrutinized, etc.

7. Research

Researchers use Data Mining tools to explore the associations between the parameters under research such as environmental conditions like air pollution and the spread of diseases like asthma among people in targeted regions.

8. Farming

Farmers use Data Mining to find out the yield of vegetables with the amount of water required by the plants.

9. Automation

By using data mining, the computer systems learn to recognize patterns among the parameters which are under comparison. The system will store the patterns that will be useful in the future to achieve business goals. This learning is automation as it helps in meeting the targets through machine learning.

10. Dynamic Pricing

Data mining helps the service providers such as cab services to dynamically charge the customers based on the demand and supply. It is one of the key factors for the success of companies.

11. Transportation

Data Mining helps in scheduling the moving of vehicles from warehouses to outlets and analyze the product loading patterns.

12. Insurance

Data mining methods help in forecasting the customers who buy the policies, analyze the medical claims that are used together, find out fraudulent behavior and risky customers.

⏭ 2.7 DATA EXPLORATION

- Data exploration refers to the initial step in data analysis in which data analysts use data visualization and statistical techniques to describe dataset characterizations, such as size, quantity, and accuracy, in order to better understand the nature of the data.
- Data exploration techniques include both manual analysis and automated data exploration software solutions that visually explore and identify relationships between different data variables, the structure of the dataset, the presence of outliers, and the distribution of data values in order to reveal patterns and points of interest, enabling data analysts to gain greater insight into the raw data.
- Data is often gathered in large, unstructured volumes from various sources and data analysts must first understand and develop a comprehensive view of the data before extracting relevant data for further analysis, such as univariate, bivariate, multivariate, and principal components analysis.

- Manual data exploration methods entail either writing scripts to analyze raw data or manually filtering data into spreadsheets. Automated data exploration tools, such as data visualization software, help data scientists easily monitor data sources and perform big data exploration on otherwise overwhelmingly large datasets. Graphical displays of data, such as bar charts and scatter plots, are valuable tools in visual data exploration.
- A popular tool for manual data exploration is Microsoft Excel spreadsheets, which can be used to create basic charts for data exploration, to view raw data, and to identify the correlation between variables.

2.7.1 Types of Attributes

☞ Data Objects

- Data objects comprise to form data sets.
- A data object represents an entity, e.g. in a university database, the objects may be courses, professors and students.
- Data objects are typically described by attributes.
- If stored in a database, the data objects are referred to as data tuples. That is, the rows of the database correspond to the data objects and the columns correspond to the attributes.

☞ Attribute Types

- An attribute represents the characteristic or feature of a data object. E.g., attributes for customer object can include customer_ID, name, and address.
- The type of an attribute is determined by the set of possible values the attribute can have. They can be nominal, binary, ordinal, numeric, discrete or continuous.

(i) Nominal Attribute

- It is a qualitative attribute related to names.
- The values of a nominal attribute are names of things, some kind of symbols.
- Values of nominal attributes represents some category or state and thus, nominal attributes are also referred as **categorical attributes** and there is no order (rank, position) among values of the nominal attribute.
- **Example :**

Own House :	1. Yes 2. No
Marital status :	1. Unmarried 2. Married

(ii) Binary Attribute

- It is also a qualitative attribute.
- Binary data has only 2 values/states. For example, yes or no, affected or unaffected, true or false.
- Symmetric Binary Attribute: Both values are equally important (e.g. Gender).
- Asymmetric Binary Attribute: Both values are not equally important (e.g. Result).
- **Example :**

Gender :	Male, Female
Cancer Detected:	Yes, No
Result	Pass, Fail

(iii) Ordinal Attribute

- It is also a qualitative attribute.
- The Ordinal Attributes contains values that have a meaningful sequence or ranking(order) between them, but the magnitude between values is not actually known.
- The order of values shows what is important but don't indicate how important it is.
- **Example:**

Grade:	A, B, C, D, E, F, O
Income:	Low, Medium, High
Product Rating:	0, 1, 2, 3, 4, 5

(iv) Numeric Attribute

- A numeric attribute is quantitative because, it is a measurable quantity, represented in integer or real values.
- Numerical attributes are of 2 types, interval and ratio.
- An **interval-scaled attribute** has values, whose differences are interpretable, but the numerical attributes do not have the correct reference point, or we can call zero points. Data can be added and subtracted at an interval scale but cannot be multiplied or divided.

Consider an example of temperature in degrees Centigrade. If a day's temperature of one day is twice of the other day, we cannot say that one day is twice as hot as another day.

- A **ratio-scaled attribute** is a numeric attribute with a fix zero-point. If a measurement is ratio-scaled, we can say of a value as being a multiple (or ratio) of another value. The values are ordered, and we can also compute the difference between values, and the mean, median, mode, Quantile-range, and Five number summary can be given.

(v) Discrete Attribute

- It is also a quantitative attribute.
- It can be numerical and can also be in categorical form.
- These attributes have finite or countably infinite set of values.
- **Example:**

Profession:	Principal, Teacher, Clerk, Peon
Zipcode:	400050, 400051, 400052

(vi) Continuous Attribute

- It is also a quantitative attribute.
- It can take any value between two specified values.
- **Example:**

Height:	5.2, 5.4, 5.6, ….
Weight:	50.33, …

2.8 STATISTICAL DESCRIPTION AND DESCRIPTIVE DATA SUMMARIZATION

UQ. Suppose that the data for analysis includes the attribute salary. We have the following values for salary (in thousands of dollars), shown in increasing order: 30, 36, 47, 50, 52, 52, 56, 60, 63, 70, 70, 110.

(i) What are the mean, median, mode and midrange of the data?

(ii) Find the first quartile (Q_1) and the third quartile (Q_3) of the data.

(iii) Show the boxplot of the data.

MU - Dec. 2019

- It is essential to have an overall picture of the data, if data preprocessing is to be made successful.
- Statistical description of data is useful in identifying the properties of the data and highlight which data value should be treated at noise or outliers.
- Following are the basic statistical description of data:

2.8.1 Measures of Central Tendency

- A measure of central tendency is a number used to represent the center or middle of a set of data values.
- The mean, median, mode and midrange are commonly used measures of central tendency.

(i) Mean

- The mean, or average, of n numbers is the sum of the numbers divided by n.
- The mean is denoted by $\bar{x}$ and is read as "x-bar".
- For the data set $x_1, x_2, \ldots, x_n$, the mean is

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{x_1 + x_2 + \ldots + x_n}{n}$$

- Sometimes, weights are associated with the value x_i. The weights reflect the importance, significance or frequency of occurrence to their respective values. The weighted arithmetic mean or the weighted average is computed as

$$\bar{x} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i} = \frac{w_1 x_1 + w_2 x_2 + \ldots + w_n x_n}{w_1 + w_2 + \ldots + w_n}$$

- Mean has one limitation; it is highly sensitive to outliers. Under such condition, median would be a better measure of central tendency.

(ii) Median

- The median of n numbers is the middle number when numbers are written in order.
- If n is even, the median is the mean of the two middle numbers.
- When we have large number of observations, the median is expensive to compute.

- In such case, we can approximate the median of the entire data set by interpolation using the **formula** :

$$\text{median} = L_1 + \left(\frac{\frac{n}{2} - (\Sigma \text{freq})_1}{\text{freq}_{\text{median}}} \right) \text{width}$$

where,

L_1 is the lower boundary of the median interval,

n is the number of values in the entire data set,

$(\Sigma \text{ freq})_1$ is the sum of frequencies of all of the intervals that are lower than the median interval

$\text{Freq}_{\text{median}}$ is the frequency of the median interval and

width is the width of the median interval.

(iii) Mode

- The mode of n numbers is the number or numbers that occur most frequently.
- There may be one mode, no mode or more than one mode.
- For unimodal numeric data that are asymmetrical, we have the following empirical relation:

mean – median ≈ 3 × (mean – median)

(iv) Midrange

It is the average of the largest and smallest values in the set.

Ex. 2.8.1 : The data set below gives the waiting time (in minutes) of several people having the oil changed in their car at an auto mechanics shop. 22, 18, 25, 21, 28, 26, 20, 28, 20. Find the mean, median, mode and the midrange of the data set.

☑ **Soln. : Data set :** 22, 18, 25, 21, 28, 26, 20, 28, 20

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{x_1 + x_2 + \ldots + x_n}{n}$$

$$= \frac{22 + 18 + 25 + 21 + 28 + 26 + 20 + 28 + 20}{9} = 23.11$$

- To find median, arrange the values in order.

18, 20, 20, 21, 22, 25, 26, 28, 28

There are total 9 values i.e. n is odd. Thus, the median is the middle value.

Median = 22

- Mode is the number or numbers that occur most frequently. Here, 20 and 28 are repeated twice. Thus, data set is bimodal with values 20 and 28.
- Midrange = $\frac{\text{largest value + smallest value}}{2} = \frac{28 + 18}{2}$

= 23

2.8.2 Dispersion of Data

- A measure of dispersion is a statistic that tells you how dispersed, or spread out, data values are.
- The measures include range, quantiles, quartiles, percentiles, the interquartile range, and the five-number summary displayed as a boxplot, variance, and standard deviation.

(i) Quartiles

- Quartiles are values that divide your data into quarters.
- However, quartiles are not shaped like pizza slices; instead they divide your data into four segments according to which the numbers fall on the number line.
- The four quarters that divide a data set into quartiles are:
  a. The lowest 25% of numbers. Also called the 1st quartile (Q_1) or 25th percentile.
  b. The next lowest 25% of numbers (up to the median). Also called the 2nd quartile (Q_2) or 50th percentile.
  c. The second highest 25% of numbers (above the median). Also called the 3rd quartile (Q_3) or 75th percentile.
  d. The highest 25% of numbers. Also called the 4th quartile (Q_4) or 100th percentile.
- As quartiles divide numbers up according to where their position is on the number line, you have to put the numbers in order before you can figure out where the quartiles are.

(ii) Interquartile Range (IQR)

- Interquartile range is defined as the difference between the upper and lower quartile values in a set of data.
- It is commonly referred to as IQR and is used as a measure of spread and variability in a data set.
- $IQR = Q_3 - Q_1$

(iii) Five Number Summary

- The five number summary gives you a rough idea about what your data set looks like.
- It includes five items: the minimum value, the first quartile (Q_1), the median, the third quartile (Q_3), the maximum value.
- In order for the five numbers to exist, your data set must meet these two requirements:
  a. Your data must be **univariate**. In other words, the data must be a single variable. For example, this list of weights is one variable: 120, 100, 130, 145. If you have a list of ages and you want to compare the ages to weights, it becomes bivariate data (two variables). For example: age 1 (25 pounds), 5 (60 pounds), 15 (129 pounds). The matching pairs makes it impossible to find a five number summary.
  b. Your data must be **ordinal**, **interval**, or **ratio**.
- Steps to Find a Five-Number Summary

Step 1 : Put your numbers in ascending order (from smallest to largest).

For example, consider the data set in order as: 1, 2, 5, 6, 7, 9, 12, 15, 18, 19, 27.

Step 2: Find the minimum and maximum for your data set.

In the example in step 1, the minimum (the smallest number) is 1 and the maximum (the largest number) is 27.

Step 3 : Find the median. The median is the middle number.

Step 4 : Place parentheses around the numbers above and below the median. (This is not technically necessary, but it makes Q_1 and Q_3 easier to find).

(1, 2, 5, 6, 7), 9, (12, 15, 18, 19, 27).

Step 5 : Find Q_1 and Q_3. Q_1 can be thought of as a median in the lower half of the data, and Q_3 can be thought of as a median for the upper half of data.

(1, 2, 5, 6, 7), 9, (12,15,18,19,27).

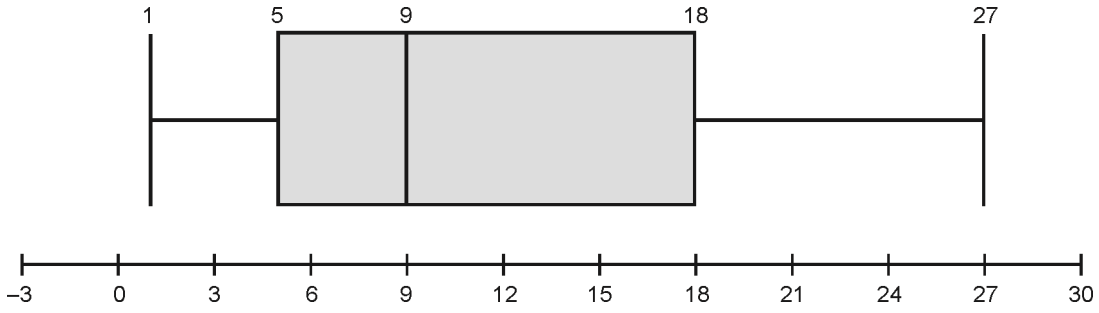
Step 6 : Write down your summary found in the above steps.

minimum = 1, Q_1 = 5, median = 9, Q_3 = 18 and maximum = 27.

(iv) Boxplot

- A boxplot (or whisker plot) is defined as a graphical method of displaying variation in a set of data.
- A boxplot incorporates the five-summary as follows:
  a. The ends of the box are at the quartiles and the box length is the interquartile range.
  b. The median is marked by a line within a box.
  c. Two lines (called whiskers) outside the box extend to the minimum and maximum values in the data set.

- The boxplot for five-number summary example above is as given below.



(1B4)**Fig. 2.8.1: Boxplot Example**

- Boxplots can be computed in *O(n logn)* time.

(v) Outlier

It is a value higher or lower than 1.5 × IQR (**Inter-Quartile Range**)

(vi) Variance and Standard Deviation

- Variance and standard deviation are measures of data dispersion. They indicate how spread out a data distribution is.
- For the data set $x_1, x_2, \ldots, x_n$, the variance is calculated as

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 = \left(\frac{1}{n}\sum_{i=1}^{n}x_i^2\right) - (\bar{x})^2$$

  where $\bar{x}$ is the mean value of the observation
- The standard deviation, σ, of the observations is the square root of the variance σ^2.
- A low standard deviation indicates that the data observations tend to be very close to the mean, while a high standard deviation indicates that the data observations are spread out over a large range of values.
- When all observations have the same value, σ = 0. Otherwise, σ > 0.

Ex. 2.8.2 : 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 (Q_1) and the third quartile (Q_3) of the data?

(e) Give the five-number summary of the data.

(f) Show a boxplot of the data.

☑ **Soln. :**

(a) Mean = $\bar{x} = \sum_{i=1}^{n}\frac{x_1}{n} = \frac{809}{27} = 29.96 = 30$

(b) The **median** (middle value of the ordered set, as the number of values in the set is odd) of the data is 25.

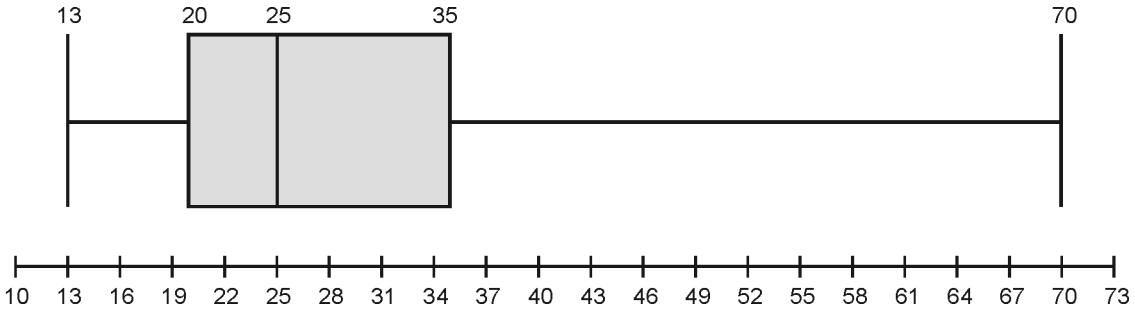
(c) This data set has two values that occur with the same highest frequency and is, therefore, **bimodal**. The modes (values occurring with the greatest frequency) of the data are 25 and 35.

(d) The **midrange** (average of the largest and smallest values in the data set) of the data is : $\frac{(70 + 13)}{2} = 41.5$

(e) The **first quartile** Q_1 (corresponding to the 25th percentile) of the data is : 20.

The **third quartile** Q_3 (corresponding to the 75th percentile) of the data is : 35.

(f) Box Plot



(1B5) **Fig. P. 2.8.2: Boxplot for Age Attribute**

UEx. 2.8.3 **MU - Dec. 2019**

Suppose that the data for analysis includes the attribute salary. We have the following values for salary (in thousands of dollars), shown in increasing order: 30, 36, 47, 50, 52, 52, 56, 60, 63, 70, 70, 110.

(i) What are the mean, median, mode and midrange of the data?

(ii) Find the first quartile (Q_1) and the third quartile (Q_3) of the data.

(iii) Show the boxplot of the data.

☑ **Soln. :**

(i) Mean = $\bar{x} = \sum_{i=1}^{n} = \frac{x_i}{n} = \frac{696}{12} = 58$

(ii) The **median** (mean of the two middle values of the ordered set, as the number of values in the set is even) of the data is 54.

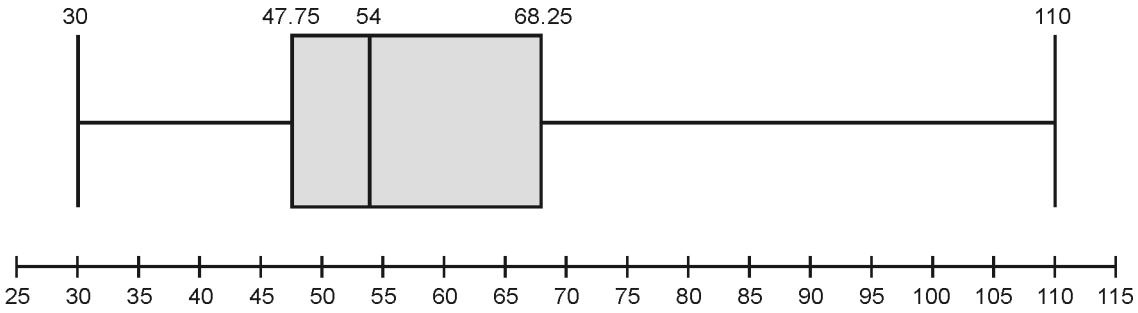
(iii) This data set has two values that occur with the same highest frequency and is, therefore, **bimodal**. The modes (values occurring with the greatest frequency) of the data are 52 and 70.

(iv) The **midrange** (average of the largest and smallest values in the data set) of the data is : $\frac{(110 + 30)}{2} = 70$

(v) The **first quartile** Q_1 (corresponding to the 25^{th} percentile) of the data is: 47.75.

The **third quartile** Q_3 (corresponding to the 75^{th} percentile) of the data is: 68.25.

(vi) Box Plot



(1B6)**Fig. P. 2.8.3: Boxplot for Salary Attribute**

2.8.3 Graphic Displays of Basic Statistical Descriptions of Data data visualization

- Graphic displays are helpful for visual inspection of data, which is useful for data preprocessing.
- These include quantile plots, quantile-quantile plots, histograms and scatter plots.
- Quantile plots, quantile-quantile plots and histograms show univariate distributions.
- Scatter plots show bivariate distributions.

(a) Quantile Plot

- A normal quantile plot (also known as a quantile-quantile plot or QQ plot) is a graphical way of checking whether your data are normally distributed.
- On one axis, you plot your data, sorted smallest to largest. On the other axis you plot the numbers you would expect to see if your data were normally distributed.
- If your data are normally distributed, you should see a nearly straight line.

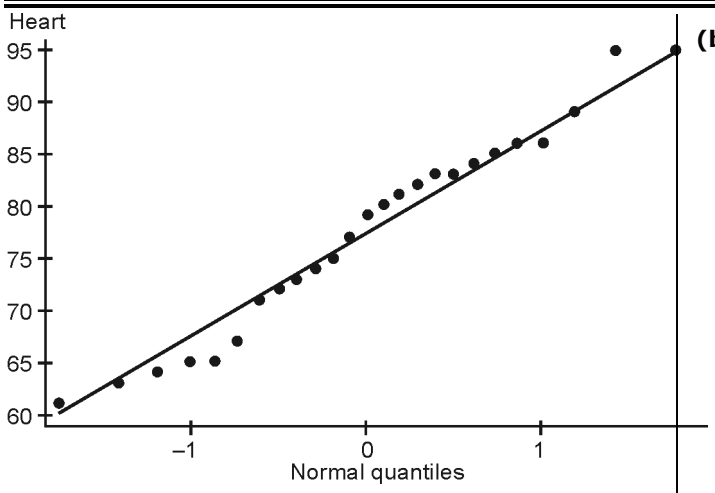
- **Example:** Suppose we wish to know whether the resting heart rates of a sample of students are normally distributed.

Heart rate				
61	63	64	65	65
67	71	72	73	74
75	77	79	80	81
82	83	83	84	85
86	86	89	95	95

- We compute for each data value, the normal quantile value as

$$z = \frac{x_i - \bar{x}}{\rho}$$

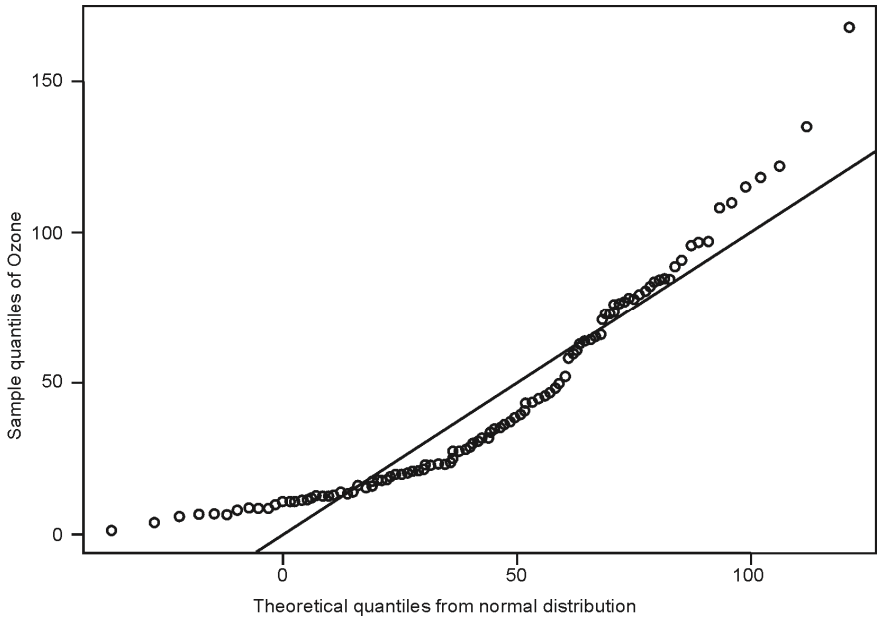
- Then we plot the normal quantile plot of heart rate.



(1B7)**Fig. 2.8.2: Normal Quantile Plot for Heart Rate**

(b) Quantile-quantile Plot

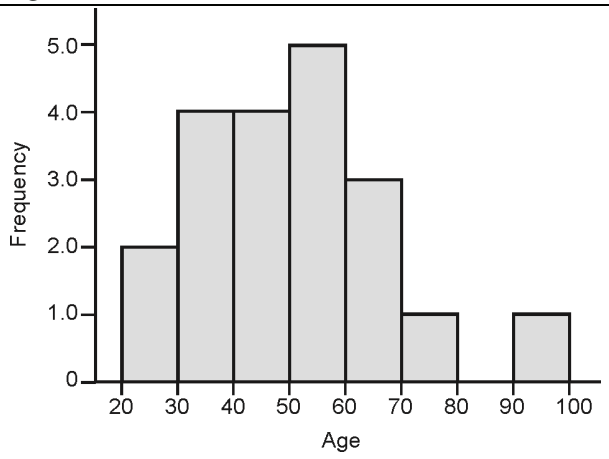
- Quantile-quantile Plots (Q-Q plots) are plots of two quantiles against each other.
- A quantile is a fraction where certain values fall below that quantile. For example, the median is a quantile where 50% of the data fall below that point and 50% lie above it.
- The purpose of Q-Q plots is to find out if two sets of data come from the same distribution.
- If the two data sets come from a common distribution, the points will fall on a single reference line.



(1B8)**Fig. 2.8.3: Quantile-Quantile Plot**

(c) Histogram

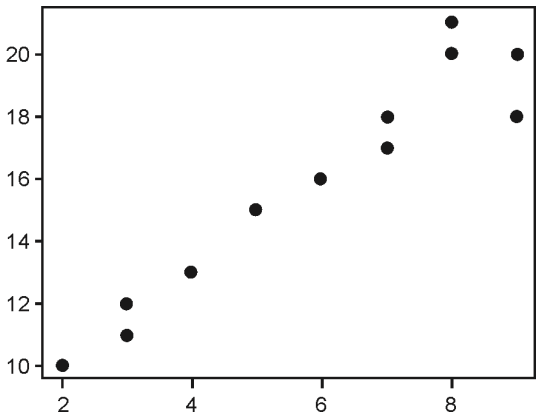
- Usually shows the distribution of values of a single variable.
- Divide the values into bins and show a bar plot of the number of objects in each bin.
- The height of each bar indicates the number of objects.
- Shape of histogram depends on the number of bins.



(1B9)**Fig. 2.8.4: Histogram**

(d) Scatter Plot

- Scatter plot determines if there is a relationship, pattern, or trend existing between two numeric attributes.
- It also explores the possibility of correlation relationships between two attributes.
- Correlations can be positive, negative or zero (uncorrelated).



(1B10)**Fig. 2.8.5: Scatter Plot**

2.9 DATA VISUALIZATION

- Visualization is the conversion of data into a visual or tabular format so that the characteristics of the data and the relationships among data items or attributes can be analyzed or reported.
- Visualization of data is one of the most powerful and appealing techniques for data exploration.
- Humans have a well-developed ability to analyze large amounts of information that is presented visually.
- Can detect general patterns and trends.
- Can detect outliers and unusual patterns.

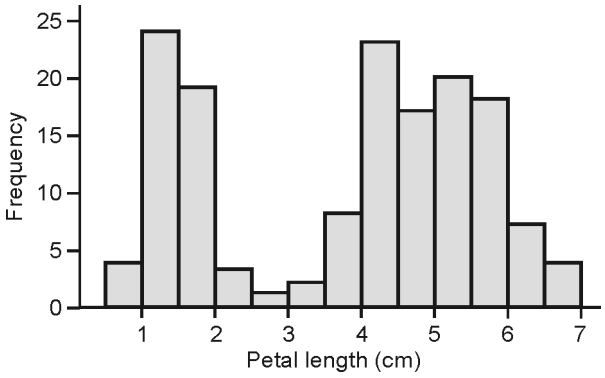
Visualization Techniques

(1) Histogram

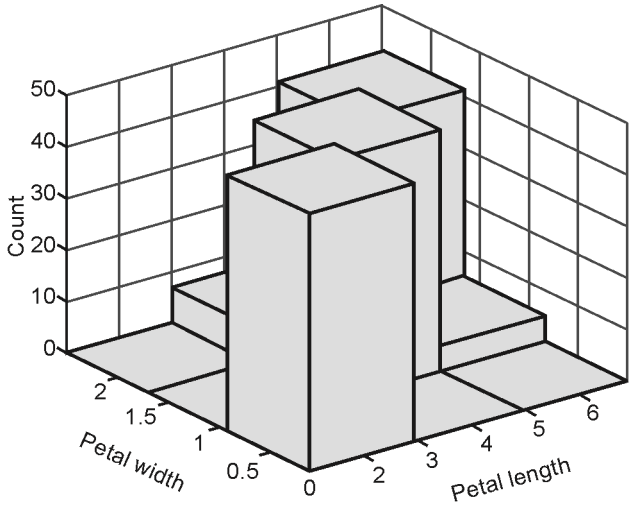
- Usually shows the distribution of values of a single variable.
- Divide the values into bins and show a bar plot of the number of objects in each bin.
- The height of each bar indicates the number of objects.
- Shape of histogram depends on the number of bins
- Example: Petal Width

Two-Dimensional Histograms

- Show the joint distribution of the values of two attributes
- Example: petal width and petal length



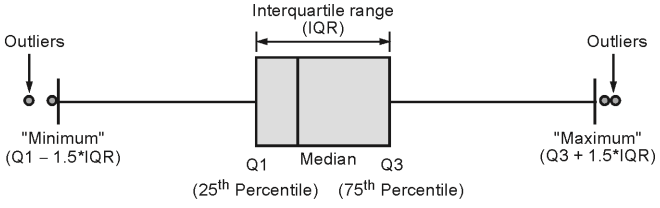
(1B11)**Fig. 2.9.1(a): 1D Histogram**



(1B12)**Fig. 2.9.1(b) 2D Histogram**

(2) Boxplots

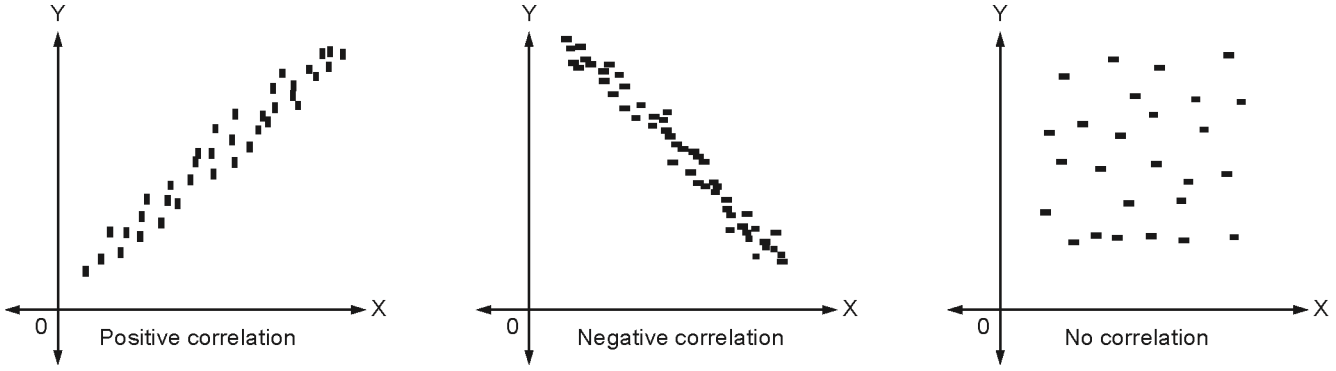
- Invented by J. Tukey.
- Another way of displaying the distribution of data.
- Following figure shows the basic part of a box plot.
- Box plots can be used to compare attributes.



(1B13)**Fig. 2.9.2: Boxplot**

(3) Scatter plots

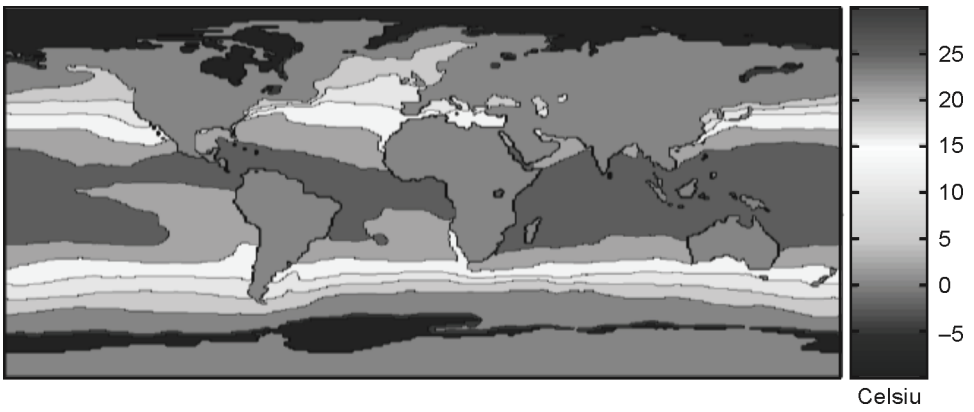
- Attributes values determine the position.
- Two-dimensional scatter plots most common, but can have three-dimensional scatter plots.
- Often additional attributes can be displayed by using the size, shape, and color of the markers that represent the objects.
- It is useful to have arrays of scatter plots can compactly summarize the relationships of several pairs of attributes.



(1B14)**Fig. 2.9.3: Scatter Plots**

(4) Contour plots

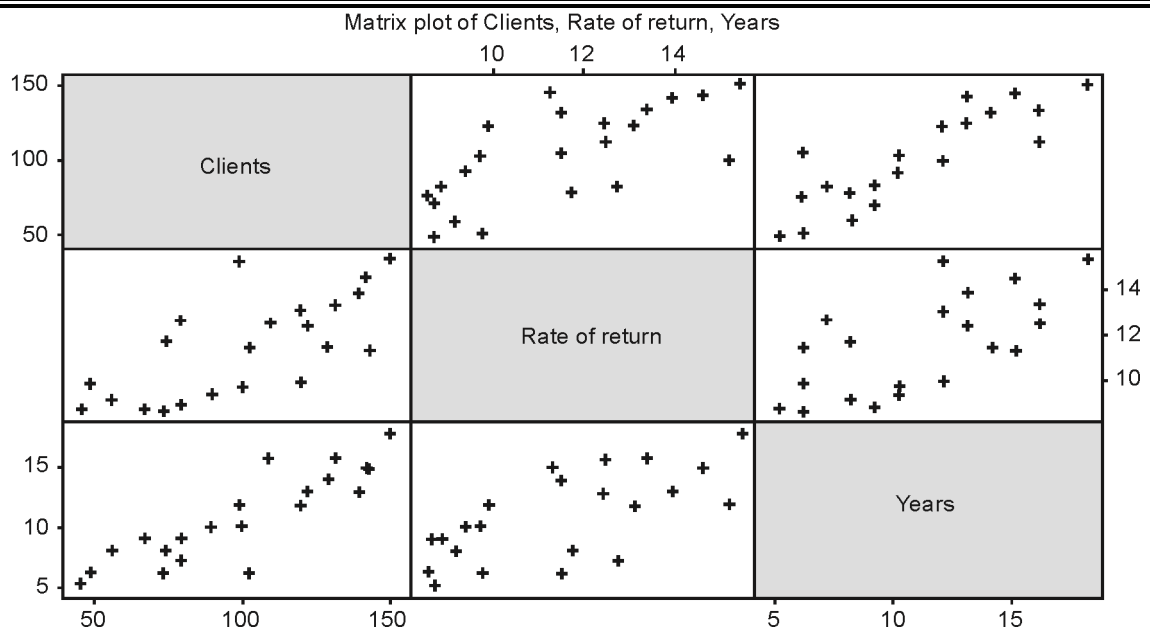
- Useful when a continuous attribute is measured on a spatial grid.
- They partition the plane into regions of similar values.
- The contour lines that form the boundaries of these regions connect points with equal values.
- The most common example is contour maps of elevation.
- Can also display temperature, rainfall, air pressure, etc.



(1B15)**Fig. 2.9.4: Contour Plot**

(5) Matrix plots

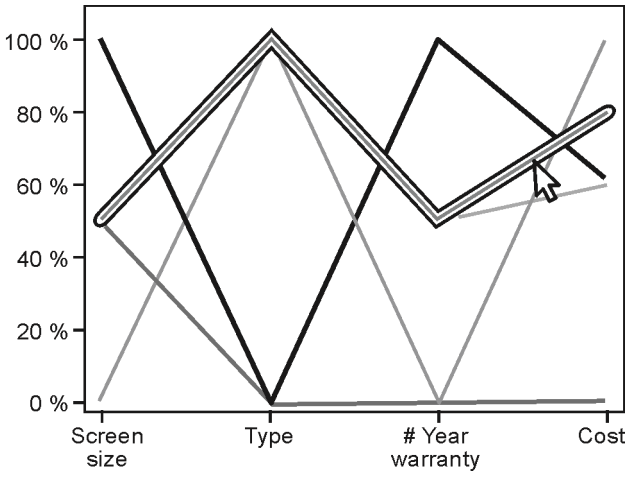
- Can plot the data matrix.
- This can be useful when objects are sorted according to class.
- Typically, the attributes are normalized to prevent one attribute from dominating the plot.
- Plots of similarity or distance matrices can also be useful for visualizing the relationships between objects.



(1B16)**Fig. 2.9.5: Matrix Plot**

(6) Parallel Coordinates

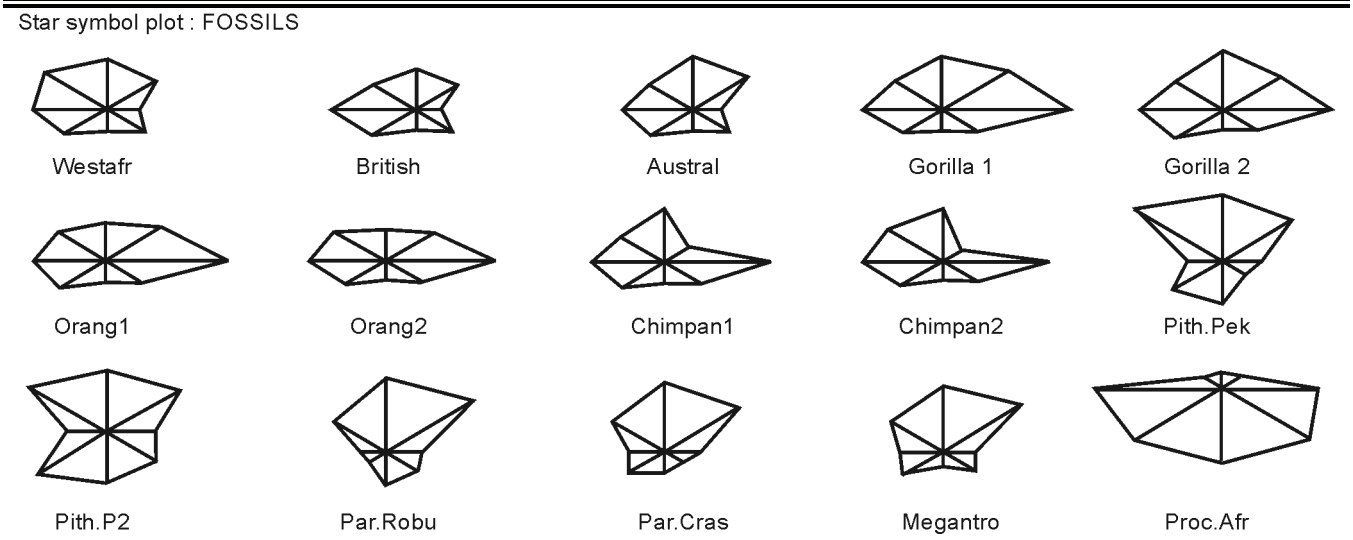
- Used to plot the attribute values of high-dimensional data.
- Instead of using perpendicular axes, use a set of parallel axes.
- The attribute values of each object are plotted as a point on each corresponding coordinate axis and the points are connected by a line.
- Thus, each object is represented as a line.
- Often, the lines representing a distinct class of objects group together, at least for some attributes.
- Ordering of attributes is important in seeing such groupings.



(1B17)**Fig. 2.9.6: Parallel Coordinates Plot**

(7) Star Plots

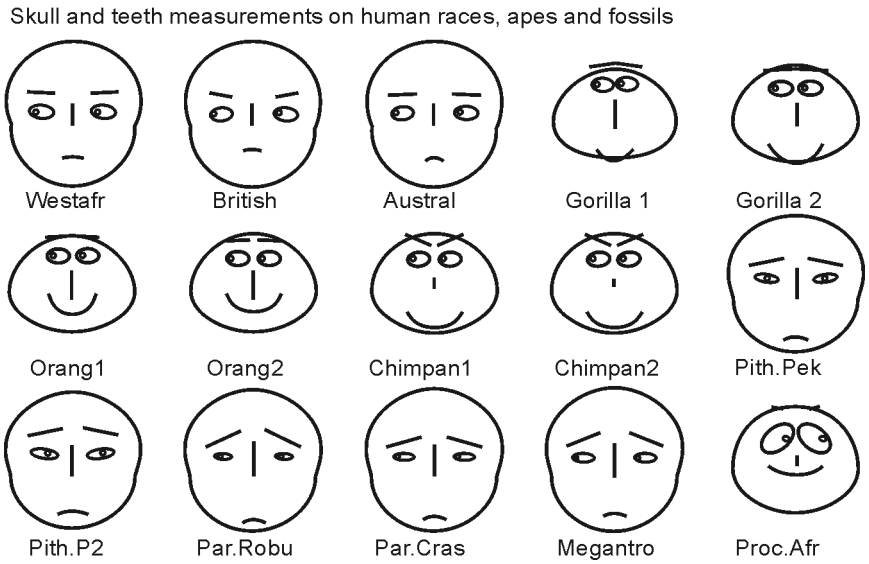
- Similar approach to parallel coordinates, but axes radiate from a central point.
- The line connecting the values of an object is a polygon.



(1B18)**Fig. 2.9.7: Star Plot**

(8) Chernoff Faces

- Approach created by Herman Chernoff.
- This approach associates each attribute with a characteristic of a face.
- The values of each attribute determine the appearance of the corresponding facial characteristic.
- Each object becomes a separate face.
- Relies on human's ability to distinguish faces.

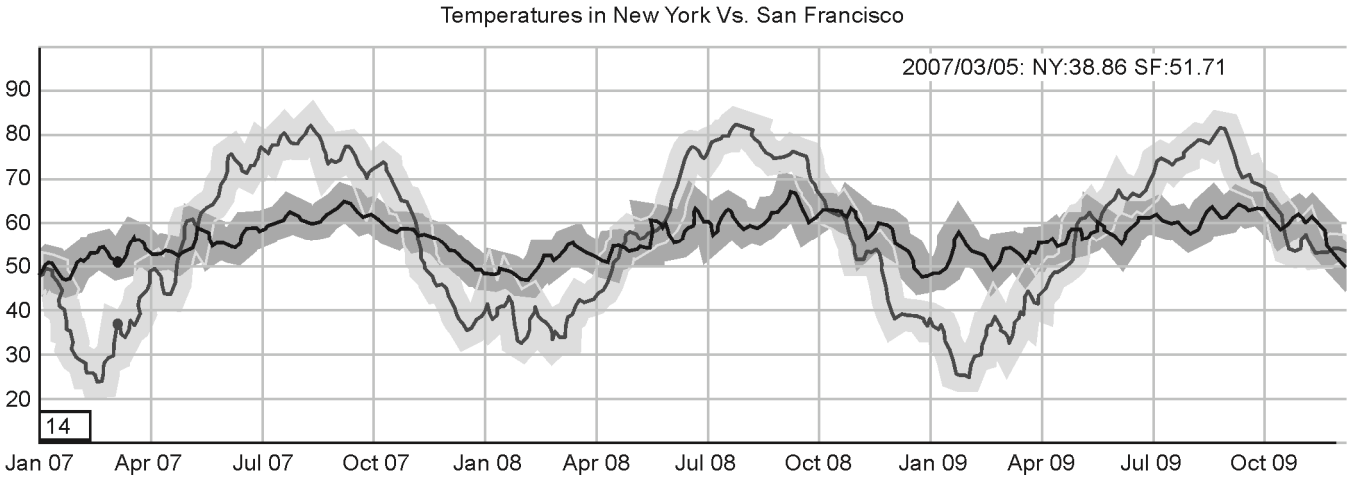


(1B19)**Fig. 2.9.8: Chernoff Faces**

(9) Dygraphs

- Dygraphs is an open source JavaScript library that produces interactive, zoomable charts of time series.
- It is designed to display dense data sets and enable users to explore and interpret them.
- Another significant feature of the dygraphs library is the ability to display error bars around data series.

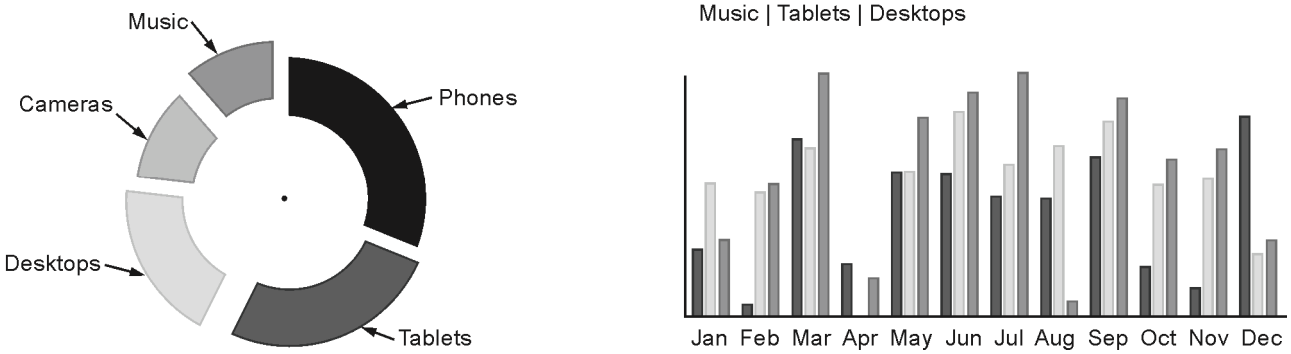
- dygraphs is purely client-side JavaScript. It does not send your data to any servers – the data is processed entirely in the client's browser.



(1B20)**Fig. 2.9.9: Dygraphs**

(10) Zing chart

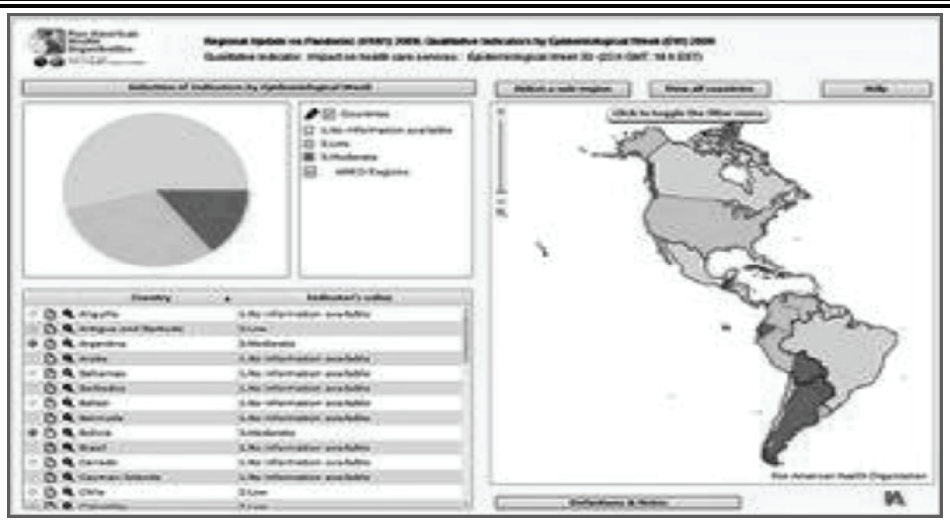
- Zing Chart is a JavaScript charting library that can help you manipulate data into visually appealing charts and graphs.
- First and foremost, Zing Chart is designed to handle big data and deliver fast results.
- Zing Chart is also mobile ready so data can adapt to screen size, or you can develop directly for mobile apps.



(1B21)**Fig. 2.9.10: Zing Chart**

(11) InstantAtlas

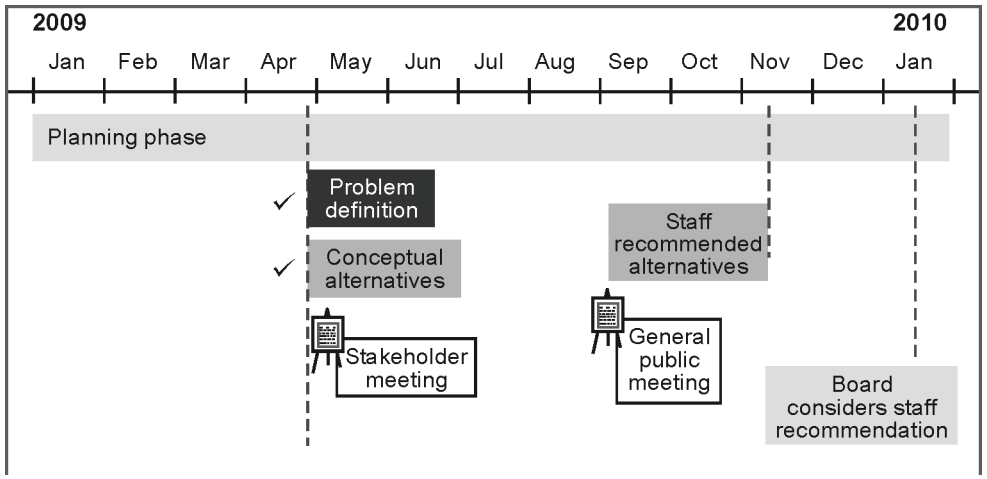
- InstantAtlas enables information analysts and researchers to create highly-interactive dynamic and profile reports that combine statistics and map data to improve data visualization, enhance communication, and engage people in more informed decision making.



(1B22)**Fig. 2.9.11 : InstantAtlas**

(12) Timeline

- A **timeline** is a way of displaying a list of events in chronological order, sometimes described as a project artifact.
- It is typically a graphic design showing a long bar labeled with dates alongside itself and usually events labeled on points where they would have happened.



(1B23)**Fig. 2.9.12: Timeline Chart**

2.10 MEASURING DATA SIMILARITY AND DISSIMILARITY

UQ. Briefly outline with example, how to compute the dissimilarity between objects described by the following :
(i) Nominal Attribute
(ii) Asymmetric binary attributes **MU - May 2019**

- Distance or similarity measures are essential in solving many pattern recognition problems such as classification and clustering.
- Various distance/similarity measures are available in the literature to compare two data distributions.
- **Similarity** measure
  (i) is a numerical measure of how alike two data objects are.
  (ii) is higher when objects are more alike.
  (iii) often falls in the range [0,1]
- Similarity might be used to identify

(i) duplicate data that may have differences due to typos.

(ii) equivalent instances from different data sets. E.g. names and/or addresses that are the same but have misspellings.

(iii) groups of data that are very close (clusters)

- **Dissimilarity** measure

(i) is a numerical measure of how different two data objects are

(ii) is lower when objects are more alike

(iii) minimum dissimilarity is often 0 while the upper limit varies depending on how much variation can be

- Dissimilarity might be used to identify

(i) Outliers

(ii) interesting exceptions, e.g. credit card fraud

(iii) boundaries to clusters

- **Proximity** refers to either a similarity or dissimilarity.

☞ Single attribute similarity/dissimilarity measures

Attribute	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } x \neq y \end{cases}$	$s = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$
Ordinal	$d = \frac{\lvert x - y \rvert}{n - 1}$ Values are mapped to integers 0 to n – 1 where n is the number of values.	$s = 1 - d$
Interval or Ratio	$d = \lvert x - y \rvert$	$s = -d$, $s = \frac{1}{1+d}$, $s = e^{-d}$, $s = 1 - \frac{d - \min\_d}{\max\_d - \min\_d}$

☞ Distance between instances with multiple attributes

(1) Euclidean Distance

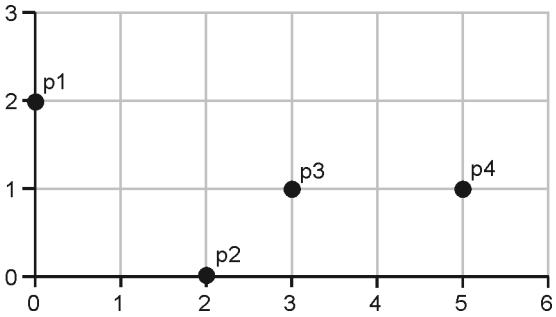
- The Euclidean distance is computed using formula

$$d(x, y) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}$$

where n is the number of dimensions (attributes) and x_k and y_k are, the *k-th* attributes (components) or data objects of x and y respectively**.**

- Standardization/normalization may be necessary to ensure an attribute does not skew the distances due to different scales.

Ex. 2.10.1 : Consider the data given below and compute the Euclidean distance between each point.



(1B24)**Fig. P. 2.10.1**

☑ **Soln. :** The x and y co-ordinate for each point is listed below.

p1(0,2), p2(2,0), p3(3,1) and p4(5,1)

The Euclidean distance formula is:

$$d(x, y) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}$$

The distance matrix between each point using the above formula is:

	p1	p2	p3	p4
p1	0			
p2	2.828	0		
p3	3.162	1.414	0	
p4	5.099	3.162	2	0

(2) Minkowski Distance

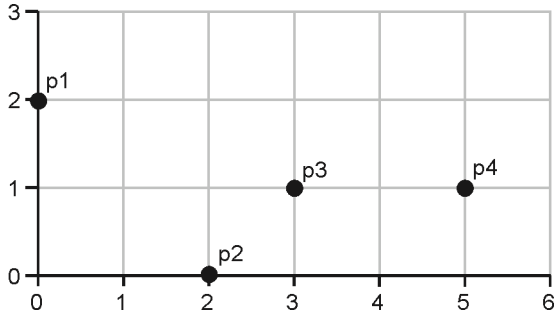
- It is a generalization of Euclidean distance.
- It is calculated using the formula :

$$d(x, y) = \sqrt[r]{\left(\sum_{k=1}^{n} | x_k - y_k | \right)}$$

where n is the number of dimensions (attributes) and x_k and y_k are, the *k-th* attributes (components) or data objects of x and y respectively**.**

- When r = 1, it is also called as City Block distance or Manhattan distance or L_1 norm distance.
- When r = 2, it is also called Euclidean distance or L_2 norm distance.
- When r = ∞, it is also called supremum or L_{max} norm or L_∞ norm distance. This is the maximum difference between any component of the vectors.

Ex. 2.10.2 : Consider the data given below and compute the Minkowski distance between each point.



(1B25)**Fig. P 2.10.2**

☑ **Soln. :** The x and y co-ordinate for each point is listed below.

p1(0,2), p2(2,0), p3(3,1) and p4(5,1)

Minkowski distance is computed using formula

$$d(x, y) = \sqrt[r]{\left(\sum_{k=1}^{n} | x_k - y_k | \right)}$$

L_1 norm distance where r = 1 is shown in the matrix below.

L_1	p1	p2	p3	p4
p1	0			
p2	4	0		
p3	4	2	0	
p4	6	4	2	0

L_2 norm distance where r = 2 is shown in the matrix below.

L_2	p1	p2	p3	p4
p1	0			
p2	2.828	0		
p3	3.162	1.414	0	
p4	5.099	3.162	2	0

L_∞ norm distance is the maximum difference between any component of the vectors.

L_∞	p1	p2	p3	p4
p1	0			
p2	2	0		
p3	3	1	0	
p4	5	3	2	0

(3) Cosine Similarity

- Cosine similarity is a measure of similarity between two vectors. The data objects here are treated as vectors.
- Similarity is measured as the angle θ between the two vectors. Similarity is 1 when θ = 0 and 0 when θ = 90°.
- If $\mathbf{d_1}$ and $\mathbf{d_2}$ are two document vectors, then

$$\cos(d_1, d_2) = \frac{d_1 \cdot d_2}{\| d_1 \| \times \| d_2 \|}$$

where $d_1 \cdot d_2$ indicates inner product or vector dot product of vectors $\mathbf{d_1}$ and $\mathbf{d_2}$, and $\| d \|$ is the length of vector d calculated as

$$\| d \| = \sqrt{\sum_{k=1}^{n} d_k^2}$$

Ex. 2.10.3 : Given two data objects

$\mathbf{d_1}$ = 3 2 0 5 0 0 0 2 0 0

$\mathbf{d_2}$ = 1 0 0 0 0 0 0 1 0 2

Calculate the cosine similarity.

☑ **Soln. :**

$\mathbf{d_1}$ = 3 2 0 5 0 0 0 2 0 0

$\mathbf{d_2}$ = 1 0 0 0 0 0 0 1 0 2

$\mathbf{d_1}.\mathbf{d_2}$ = 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5

$\| \mathbf{d_1} \|$ = (3*3 + 2*2 + 0*0 + 5*5 + 0*0 + 0*0 + 0*0 + 2*2 + 0*0 + 0*0)$^{1/2}$ = (42)$^{1/2}$ = 6.481

$\| \mathbf{d_2} \|$ = (1*1 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0 + 1*1 + 0*0 + 2*2)$^{1/2}$ = (6)$^{1/2}$ = 2.449

$$\cos(d_1, d_2) = \frac{d_1 \cdot d_2}{\| d_1 \| \times \| d_2 \|} = \frac{5}{6.481 \times 2.449} = 0.315$$

Thus, the cosine similarity between d_1 and d_2 is 0.315

The dissimilarity between d_1 and d_2 is $1 - \cos(d_1, d_2)$ = 1 – 0.315 = 0.685

(4) Jaccard Distance

- The Jaccard similarity index (sometimes called the Jaccard similarity *coefficient*) compares members for two sets to see which members are shared and which are distinct.
- It is a measure of similarity for the two sets of data, with a range from 0% to 100%.
- The formula to compute Jaccard index is:

$$J(x, y) = \frac{| x \cap y |}{| x \cup y |} = \frac{| x \cap y |}{| x | + | y | - | x \cap y |}$$

Ex. 2.10.4 : Given the record of users and movies viewed. Using Jaccard similarity measures, find similarity between {A-B, A-C, B-C}.

Users	Movie1	Movie2	Movie3	Movie4	Movie5
A	1	0	1	0	1
B	0	0	1	0	1
C	0	1	0	0	1

☑ **Soln. :**

$$J(A, B) = \frac{| A \cap B |}{| A \cup B |} = \frac{2}{3} = 0.67$$

$$J(A, C) = \frac{| A \cap C |}{| A \cup B |} = \frac{1}{4} = 0.25$$

$$J(B, C) = \frac{| B \cap C |}{| B \cup C |} = \frac{1}{3} = 0.33$$

2.11 DATA PREPROCESSING

- Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, lacking in certain behaviors or trends, and is likely to contain many errors.
- Data preprocessing is a proven method of resolving such issues. Data preprocessing prepares raw data for further processing.
- Data preprocessing is used in database-driven applications such as customer relationship management and rule-based applications (like neural networks).
- In Machine Learning (ML) processes, data preprocessing is critical to encode the dataset in a form that could be interpreted and parsed by the algorithm.
- Data goes through a series of steps during preprocessing :

(1) Data Cleaning : Data is cleansed through processes such as filling in missing values or deleting rows with missing data, smoothing the noisy data, or resolving the inconsistencies in the data. Smoothing noisy data is particularly important for ML datasets, since machines cannot make use of data they cannot interpret. Data can be cleaned by dividing it into equal size segments that are thus smoothed (binning), by fitting it to a linear or multiple regression function (regression), or by grouping it into clusters of similar data (clustering). Data inconsistencies can occur due to human errors (the information was stored in a wrong field). Duplicated values should be removed through deduplication to avoid giving that data object an advantage (bias).

(2) Data Integration : Data with different representations are put together and conflicts within the data are resolved.

(3) Data Transformation : Data is normalized and generalized. Normalization is a process that ensures that no data is redundant, it is all stored in a single place, and all the dependencies are logical.

(4) Data Reduction : When the volume of data is huge, databases can become slower, costly to access, and challenging to properly store. Data reduction step aims to present a reduced representation of the data in a data warehouse. There are various methods to reduce data. For example, once a subset of relevant attributes is chosen for its significance, anything below a given level is discarded. Encoding mechanisms can be used to reduce the size of data as well. If all original data can be recovered after compression, the operation is labelled as lossless. If some data is lost, then it's called a lossy reduction. Aggregation can also be used, for example, to condense countless transactions into a single weekly or monthly value, significantly reducing the number of data objects.

(5) Data Discretization : Data could also be discretized to replace raw values with interval levels. This step involves the reduction of a number of values of a continuous attribute by dividing the range of attribute intervals.

(6) Data Sampling : Sometimes, due to time, storage or memory constraints, a dataset is too big or too complex to be worked with. Sampling techniques can be used to select and work with just a subset of the dataset, provided that it has approximately the same properties of the original one.

2.12 CLEANING

UQ. In real-world data, tuples with missing values for some attributes are a common occurrence. Describe various methods for handling this problem. **MU - May 2019**

UQ. Develop a model to predict the salary of college graduates with 10 years of work experience using linear regression. **MU - Dec. 2019**

Years of Experience (x)	3	8	9	13	3	6	11	21	1	16
Salary in $100 (y)	30	57	64	72	36	43	59	90	20	83

UQ. Suppose a group of sales price records has been sorted as follows: 6, 9, 12, 13, 15, 25, 50, 70, 72, 92, 204, 232. Partition them into three bins by equal-frequency (Equi-depth) partitioning method. Perform data smoothing by bin mean. **MU - June 2021**

- Data cleansing or data cleaning is the process of identifying and removing (or correcting) inaccurate records from a dataset, table, or database and refers to recognizing unfinished, unreliable, inaccurate, or non-relevant parts of the data and then restoring, remodelling, or removing the dirty or crude data.
- Data cleaning techniques may be performed as batch processing through scripting or interactively with data cleansing tools.
- After cleaning, a dataset should be uniform with other related datasets in the operation. The discrepancies identified or eliminated may have been basically caused by user entry mistakes, by corruption in storage or transmission, or by various data dictionary descriptions of similar items in various stores.
- Some data cleaning methods are explained in this section.

2.12.1 Missing Values

Imagine that you are asked to analyze a dataset. You find that there are many tuples having no recorded value for several attributes such as customer income. So the question arising here is how to fill in the missing values for this attribute. There are several methods as discussed here.

(1) Ignore the tuple : When the class label is missing, this technique is used. However, unless the tuple contains numerous attributes with missing values, this approach is not particularly useful.

(2) Fill in the missing value manually : This approach is effective on small data set with some missing values.

(3) Use a global constant to fill in the missing value : You can replace all missing attribute values with global constant, such as a label like "Unknown" or $-\infty$.

(4) Use a measure of central tendency for attribute (e.g. the mean or median) to fill in the missing value : For example, suppose customer average income is $25000, then you can use this value to replace missing value for income.

(5) Use the attribute mean or median for all samples belonging to the same class as the given tuple : For example, if you are classifying customers according to their credit_score, then you can replace the missing

value with the mean income value for customers in the same credit_score category as that of the given tuple. If the data distribution for a given class is skewed, then use the median value.

(6) Use the most probable value to fill in the missing value : This can be determined using regression, Bayesian classification or decision-tree induction.

2.12.2 Noisy Data

- A random error or variance in a measured variable is referred to as noise.
- Noisy data can occur as a result of data gathering from malfunctioning instruments, data input issues, or technological limitations.
- Examples of noise in data:
  1. Encoding error: Gender has value, say, Z.
  2. Value beyond range: Age = 200
  3. Inconsistent entries: Date_of_joining = 32-Aug-2007

Techniques to Handle Noisy Data

(1) Binning

- Binning procedures smooth a sorted data item by examining its "neighborhood," or the values in its immediate vicinity.
- The sorted values are divided into several "buckets" or bins.

Different approaches of binning

(a) Smoothing by bin means : In smoothing by bin means, each value in a bin is replaced by the mean value of the bin.

(b) Smoothing by bin median : In this method each bin value is replaced by its bin median value.

(c) Smoothing by bin boundary : In smoothing by bin boundaries, the minimum and maximum values in a given bin are identified as the bin boundaries. Each bin value is then replaced by the closest boundary value.

Approach

(1) Sort the array of given data set.

(2) Divides the range into N intervals, each containing the approximately same number of samples (Equal-depth partitioning).

(3) Store mean/ median/ boundaries in each row.

Ex. 2.12.1 : Suppose a group of age records has been sorted as follows: 3, 7, 8,13, 22, 22, 22, 26, 26, 28, 30, 37. Partition them into three bins by equal-frequency (Equi-depth) partitioning method. Perform data smoothing by bin mean, and bin boundary.

☑ **Soln. :**

Smooth the data by equal frequency bins

Given three bins. There are total 12 observations. Hence, by equi-depth partitioning method, each bin will have 4 observations.

- Bin 1: 3, 7, 8, 13
- Bin 2: 22, 22, 22, 26
- Bin 3: 26, 28, 30, 37

Smooth the data by bin mean

We take average of each bin and replace each data value by mean value in corresponding bin.

- Bin 1: 8, 8, 8, 8
- Bin 2: 23, 23, 23, 23
- Bin 3: 30, 30, 30, 30

Smooth the data by bin boundary

We take difference of each data value and the bin boundaries. Each bin value is then replaced by the closest boundary value.

- Bin 1: 3, 3, 3, 13
- Bin 2: 22, 22, 22, 26
- Bin 3: 26, 26, 26, 37

UEx. 2.12.2 MU - June 2021

Suppose a group of sales price records has been sorted as follows: 6, 9, 12, 13, 15, 25, 50, 70, 72, 92, 204, 232. Partition them into three bins by equal-frequency (Equi-depth) partitioning method. Perform data smoothing by bin mean.

☑ **Soln. :**

Smooth the data by equal frequency bins

Given three bins. There are total 12 observations. Hence, by equi-depth partitioning method, each bin will have 4 observations.

- Bin 1: 6, 9, 12, 13
- Bin 2: 15, 25, 50, 70
- Bin 3: 72, 92, 204, 232

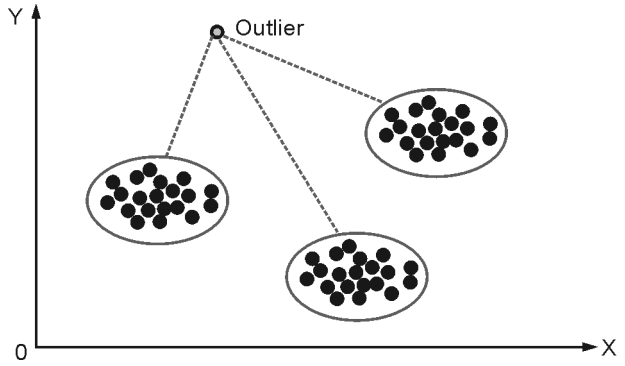
Smooth the data by bin mean

We take average of each bin and replace each data value by mean value in corresponding bin.

- Bin 1: 10, 10, 10, 10
- Bin 2: 40, 40, 40, 40
- Bin 3: 150, 150, 150, 150

(2) Outlier analysis by clustering

- Outliers are nothing but an extreme value that deviates from the other observations in the dataset.
- Outlier Analysis is a process that involves identifying the anomalous observation in the dataset.
- Outliers may be detected by clustering where similar values are organized into groups, or clusters.
- The values that fall outside of the set of clusters may be considered outliers.



(1B26)**Fig. 2.12.1 : Outlier Analysis by Clustering**

(3) Regression

- Regression is a data mining technique used to predict a range of numeric values (also called *continuous values*), given a particular dataset. For example, regression might be used to predict the cost of a product or service, given other variables.
- Data smoothing can also be done using regression.
- Regression is of two types:

(a) **Linear Regression :** It finds the best line to fit two variables or attributes so that one variable can be used to predict the other variable.

Linear Regression : $y = \alpha + \beta x$

(b) **Multiple Linear Regression :** It is an extension of linear regression where more than two variables are involved and the data are fit to a multidimensional surface.

Multiple Linear Regression : $y = \alpha + \beta_1 x_1 + \beta_1 x_2 + ... + \beta_n x_n$

Here, y is the variable that is to be predicted, x is the variable used to predict y, α is the intercept and β is the slope.

$$\beta = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2}$$

Where, $\bar{x}$ and $\bar{y}$ are the mean of variables x and y respectively.

UEx. 2.12.3 MU - Dec. 2019

Develop a model to predict the salary of college graduates with 10 years of work experience using linear regression.

Years of Experience (x)	3	8	9	13	3	6	11	21	1	16
Salary in $100 (y)	30	57	64	72	36	43	59	90	20	83

☑ **Soln. :**

Linear Regression : $y = \alpha + \beta x$

where

$$\beta = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2}$$

$$\bar{x} = \frac{1}{10}\sum_{i=1}^{10} x_i = 9.1 \quad \bar{y} = \frac{1}{10}\sum_{i=1}^{10} y_i = 55.4$$

X	Y	$x_i - \bar{x}$	$y_i - \bar{y}$	$(x_i - \bar{x})(y_i - \bar{y})$	$(x_i - \bar{x})^2$
3	30	– 6.1	– 25.4	154.94	37.21
8	57	– 1.1	1.6	– 1.76	1.21
9	64	– 0.1	8.6	– 0.86	0.01
13	72	3.9	16.6	64.74	15.21
3	36	– 6.1	– 19.4	118.34	37.21
6	43	– 3.1	– 12.4	38.44	9.61
11	59	1.9	3.6	6.84	3.61
21	90	11.9	34.6	411.74	141.61
1	20	– 8.1	– 35.4	286.74	65.61
16	83	6.9	27.6	190.44	47.61
				$\sum$ = 1269.6	$\sum$ = 358.9

$$\beta = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2} = 3.54$$

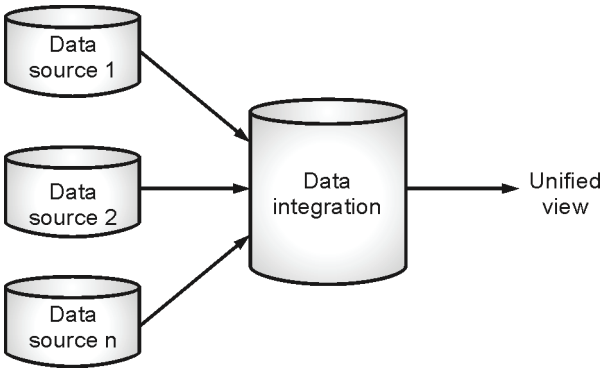
$$\alpha = \bar{y} - \beta \bar{x} = 55.4 - 3.54 \times 9.1 = 23.19$$

For x =10 years,

$$y = \alpha + \beta x = 23.19 + 3.54 \times 10 = 58.586 \text{ (in \$ 100)}$$

2.13 DATA INTEGRATION

- Data Integration is a data preprocessing technique that combines data from multiple sources and provides users a unified view of these data.



(1B27)**Fig. 2.13.1 : Data Integration**

- These sources may include multiple databases, data cubes, or flat files. One of the most well-known implementation of data integration is building an enterprise's data warehouse.
- The benefit of a data warehouse enables a business to perform analyses based on the data in the data warehouse.
- There are mainly 2 major approaches for data integration:

(a) Tight Coupling

In tight coupling data is combined from different sources into a single physical location through the process of ETL - Extraction, Transformation and Loading.

(b) Loose Coupling

In loose coupling data only remains in the actual source databases. In this approach, an interface is provided that takes query from user and transforms it in a way the source database can understand and then sends the query directly to the source databases to obtain the result.

2.13.1 Data Integration Techniques

Below explained the different data integration techniques.

(1) Manual Integration

- This technique avoids the use of automation during data integration. The data analyst himself collects the data, cleans it and integrate it to provide useful information.
- This technique can be implemented for a small organization with a small data set. But it would be tedious for the large, complex and recurring integration because it is a time consuming process as the entire process has to be done manually.

(2) Middleware Integration

- The middleware software is employed to collect the information from different sources, normalize the data and store into the resultant data set. This technique is adopted when the enterprise wants to integrate data from the legacy systems to modern systems.
- Middleware software act as an interpreter between the legacy systems and advanced systems. You can take an example of the adapter which helps in connecting two systems with different interfaces. It can be applied to some system only.

(3) Application-Based Integration

- This technique makes use of software application to extract, transform and load the data from the heterogeneous sources. This technique also makes the data from disparate source compatible in order to ease the transfer of the data from one system to another.
- This technique saves time and effort, but is little complicated as designing such an application requires technical knowledge.

(4) Uniform Access Integration

- This technique integrates data from a more discrepant source. But, here the location of the data is not changed, the data stays in its original location.
- This technique only creates a unified view which represents the integrated data. No separate storage is required to store the integrated data as only the integrated view is created for the end-user.

(5) Data Warehousing

- This technique loosely relates to the uniform access integration technique. But the difference is that the unified view is stored in certain storage. This allows the data analyst to handle more complex queries.
- Though this is a promising technique it has increased storage cost as the view or copy of the unified data needs separate storage and even it has an increase in maintenance cost.

2.13.2 Issues in Data Integration

While integrating the data we have to deal with several issues which are discussed below.

(1) Entity Identification Problem

- As we know, the data is unified from the heterogeneous sources; then how can we 'match the real-world entities from the data'. For example, we have customer data from two different data source. An entity from one data source has customer_id and the entity from the other data source has customer_number. Now how does the data analyst or the system would understand that these two entities refer to the same attribute?
- Well, here the **schema integration** can be achieved using metadata of each attribute. Metadata of an attribute incorporates its name, what does it mean in the particular scenario, what is its data type, up to what range it can accept the value. What rules does the attribute follow for the null value, blank, or zero? Analyzing this metadata information will prevent error in schema integration.
- **Structural integration** can be achieved by ensuring that the functional dependency of an attribute in the source system and its referential constraints matches the functional dependency and referential constraint of the same attribute in the target system.
- This can be understood with the help of an example. Suppose in the one system, the discount would be applied to an entire order but in another system, the discount would be applied to every single item in the order. This difference must be caught before the data from these two sources are integrated into the target system.

(2) Redundancy and Correlation Analysis

- Redundancy is one of the big issues during data integration. Redundant data is an unimportant data or the data that is no longer needed. It can also arise due to attributes that could be derived using another attribute in the data set.
- For example, one data set has the customer age and other data set has the customers date of birth, then age would be a redundant attribute as it could be derived using the date of birth.
- Inconsistencies in the attribute also raise the level of redundancy. The redundancy can be discovered using correlation analysis. The attributes are analyzed to detect their interdependency on each other thereby detecting the correlation between them.
- χ^2 (Chi-square) test is the test to analyze the correlation of nominal data.
- Correlation coefficient and covariance can be used to test the variation between the attributes of numeric data.

(3) Tuple Duplication

- Along with redundancies, data integration has also to deal with the duplicate tuples.
- Duplicate tuples may come in the resultant data if the denormalized table has been used as a source for data integration.

(4) Data Conflict Detection and Resolution

- Data conflict means the data merged from the different sources do not match. Like the attribute values may differ in different data sets. The difference maybe because they are represented differently in the different data sets.
- For example, the price of a hotel room may be represented in different currencies in different cities. This kind of issues are detected and resolved during data integration.

2.13.2(A) χ^2 (Chi-square) Test

- The Chi-square test is also known as the name of the "goodness of fit test".

- The chi-square test helps you to solve the problem in feature selection by testing the relationship between the features.
- χ^2 (Chi-square) test is defined by the formula

$$\chi^2_{df} = \sum \left[\frac{(0 - E)^2}{E} \right]$$

where df = degrees of freedom, and df = (number of rows – 1) × (number of columns – 1)

O = observed value(s)

E = expected value(s)

- Expected value in each cell is calculated as E = (row total × column total)/ table total

☞ Steps in χ^2 (Chi-square) test

(1) Define the hypotheses.

Null Hypothesis (H_0): Two variables are independent.

Alternate Hypothesis (H_1): Two variables are not independent.

(2) Create the contingency table.

It is a table showing the distribution of one variable in rows and another in columns. It is used to study the relation between two variables. It contains the observed value O.

Degrees of freedom for contingency table is given as (r – 1) × (c – 1) where r, c are rows and columns.

(3) Find the expected value E.

(4) Calculate the χ^2 (Chi-square) value.

(5) For the given level of significance (or confidence level, α) and the degree of freedom computed in contingency table, find the critical value of χ^2 (Chi-square) from the distribution table.

(6) If the calculated χ^2 (Chi-square) value < critical χ^2 (Chi-square) value, then accept H_0 and reject H_1. Else, reject H_0 and accept H_1.

Ex. 2.13.1 : Assume that an app provides ratings to all the restaurants under 3 categories, good, okay, and not recommended. Now the challenge is to segregate restaurants under correct categories. They can be created under the name of the seating capacity of the restaurant. Small is for a restaurant with a sitting capacity of 20 people, the medium is for sitting capacity of 100 people and large is for sitting capacity of more than 100 people. Results are shown in the contingency table below.

		Restaurant Size		
		Small	**Medium**	**Large**
Ratings	Good	30	10	20
	Okay	8	10	12
	Not Recommended	3	5	2

Level of significance is 0.05. Interpret the result.

☑ **Soln. :**

(1) Define the hypotheses.

Null Hypothesis (H_0): Ratings and size are independent.

Alternate Hypothesis (H_1): Ratings and size are not independent.

(2) Create the contingency table.

It contains the observed value O.

		Restaurant Size			
		Small	**Medium**	**Large**	**Column Total**
Ratings	Good	30	10	20	60
	Okay	8	10	12	30
	Not Recommended	3	5	2	10
	Row Total	41	25	34	**100**

Degrees of freedom for contingency table is given as (r – 1) × (c – 1) where r,c are rows and columns.

df = (3 – 1) × (3 – 1) = 4

(3) Find the expected value E.

$$E = \frac{(\text{row total} \times \text{column total})}{\text{table total}}$$

		Restaurant Size			
		Small	**Medium**	**Large**	**Column Total**
Ratings	Good	24.6	15	20.4	60
	Okay	12.3	7.5	10.2	30
	Not Recommended	4.1	2.5	3.4	10
	Row Total	41	25	34	**100**

(4) Calculate the χ^2 (Chi-square) value.

$$\chi^2_{df} = \sum \left[\frac{(0-E)^2}{E} \right]$$

$$\chi^2_4 = \frac{(30-24.6)^2}{24.6} + \frac{(10-15)^2}{15} + \frac{(20-20.4)^2}{20.4} + \frac{(8-12.3)^2}{12.3} + \frac{(10-7.5)^2}{7.5} + \frac{(12-10.2)^2}{10.2} + \frac{(3-4.1)^2}{4.1} + \frac{(5-2.5)^2}{2.5} + \frac{(2-3.4)^2}{3.4} = 8.88$$

(5) For $\alpha = 0.05$ and df = 4, find the critical value of χ^2 (Chi-square) from the distribution table.

$$\chi^2_{0.05,4} = 9.488$$

(6) Calculated χ^2 (Chi-square) value = 8.88 < critical χ^2 (Chi-square) value = 9.488.

Thus, accept H_0 and reject H_1. This states that, ratings and size are independent of each other.

2.13.2(B) The Correlation Coefficient and Covariance

- Covariance and correlation are two measures that can tell you, statistically, whether or not a real relationship exists between two variables.
- **Covariance** is a statistical measure that shows whether two variables are related by measuring how the variables change in relation to each other. This could be **positive covariance**, meaning as one increases the other also increases, or **negative covariance**, meaning that as one increases the other decreases.
- **Correlation**, like covariance, is a measure of how two variables change in relation to each other, but it goes one step further than covariance in that correlation tells how strong the relationship is.
- If correlation > 0, then two variables are positively correlated. The higher value, the stronger correlation.

If correlation = 0, then two variables are independent.

If correlation < 0, then two variables are negatively correlated.

- The covariance between X and Y is defined as

$$\text{Cov}(X, Y) = \frac{\sum (x_i - \bar{x})(y_1 - \bar{y})}{n}$$

where, $\bar{x}$ and $\bar{y}$ are the mean value of x and y variables respectively.

- The correlation coefficient is given by,

$$r_{x,y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y}$$

where, $\sigma_x = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n}}$, also called the standard deviation of X.

Ex. 2.13.2 : Assume you are the new owner of a small ice-cream shop in a little village near the beach. You noticed that there was more business in the warmer months than the cooler months as shown in table below. Before you alter your purchasing pattern to match this trend, you want to be sure that the relationship is real. Compute covariance and correlation coefficient to support your assumption.

Temperature	Customers
98	15
87	12
90	10
85	10
95	16
75	7

☑ **Soln. :**

$$\text{Cov}(X, Y) = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{n}$$

$$\bar{x} = 88.33, \ \bar{y} = 11.67$$

Temperature (x)	Customers (y)	$x_i - \bar{x}$	$y_i - \bar{y}$	$(x_i - \bar{x})^2$	$(y_i - \bar{y})^2$	$(x_i - \bar{x})(y_i - \bar{y})$
98	15	9.67	3.33	93.51	11.09	32.20
87	12	– 1.33	0.33	1.77	0.19	– 0.44
90	10	1.67	– 1.67	2.79	2.79	– 2.79
85	10	– 3.33	– 1.67	11.09	2.79	5.56
95	16	6.67	4.33	44.49	18.75	28.88
75	7	– 13.33	– 4.67	177.69	21.81	62.25

$$\text{Cov}(X, Y) = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{n} = \frac{125.66}{6} = 20.94$$

The covariance of this set of data is 20.94. The number is positive, so we can state that the two variables do have a positive relationship; as temperature rises, the number of customers in the store also rises.

$$\sigma_x = \sqrt{\frac{\Sigma (x_i - \bar{x})^2}{n}} = \sqrt{\frac{271.34}{6}} = 6.72$$

$$\sigma_x = \sqrt{\frac{\Sigma (y_i - \bar{y})^2}{n}} = \sqrt{\frac{57.42}{6}} = 3.09$$

$$r_{x,y} = \frac{Cov(X,Y)}{\sigma_x \sigma_y} = \frac{20.94}{6.72 \times 3.09} = 1$$

Thus, there is a strong linear relation between temperature and number of customers.

2.14 DATA REDUCTION

UQ. 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, 23, 29, 33, 41, 44, 53, 62, 69, 72

Use min-max normalization to transform the value 45 for age onto the range [0.0, 1.0].

MU – June 2021

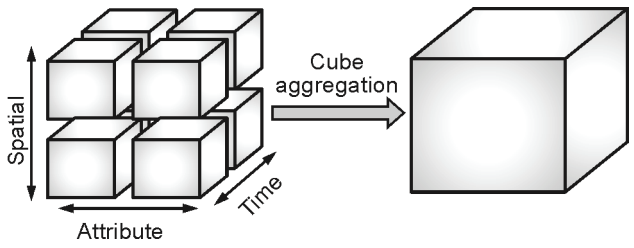
- A database or date warehouse may store terabytes of data. So it may take very long to perform data analysis and mining on such huge amounts of data.
- Data reduction techniques can be applied to obtain a reduced representation of the data set that is much smaller in volume but still contain critical information.

2.14.1 Methods of Data Reduction

Different methods of data reduction are explained below.

(1) Data Cube Aggregation

- This technique is used to aggregate data in a simpler form. For example, imagine that information you gathered for your analysis for the years 2012 to 2014, that data includes the revenue of your company every three months. They involve you in the annual sales, rather than the quarterly average.
- So we can summarize the data in such a way that the resulting data summarizes the total sales per year instead of per quarter. It summarizes the data.



(1B28)**Fig. 2.14.1 : Data Cube Aggregation**

(2) Dimensionality Reduction

- Whenever we come across any data which is weakly important, then we use the attribute required for our analysis.
- It reduces data size as it eliminates outdated or redundant features.
- Dimensionality reduction can be performed using the below approaches.

(a) Step-wise Forward Selection

- The selection begins with an empty set of attributes.
- Later on we decide best of the original attributes on the set based on their relevance to other attributes.
- This is called as a p-value in statistics.
- **Example :**

Suppose there are the following attributes in the data set in which few attributes are redundant.

Initial attribute Set: {X1, X2, X3, X4, X5, X6}

Initial reduced attribute set: { }

Step-1: {X1}

Step-2: {X1, X2}

Step-3: {X1, X2, X5}

Final reduced attribute set: {X1, X2, X5}

(b) Step-wise Backward Selection

This selection starts with a set of complete attributes in the original data and at each point, it eliminates the worst remaining attribute in the set.

Suppose there are the following attributes in the data set in which few attributes are redundant.

Initial attribute Set: {X1, X2, X3, X4, X5, X6}

Initial reduced attribute set: {X1, X2, X3, X4, X5, X6}

Step-1: {X1, X2, X3, X4, X5}

Step-2: {X1, X2, X3, X5}

Step-3: {X1, X2, X5}

Final reduced attribute set: {X1, X2, X5}

(c) Combination of forwarding and Backward Selection

It allows us to remove the worst and select best attributes, saving time and making the process faster.

(3) Data Compression

- The data compression technique reduces the size of the files using different encoding mechanisms (Huffman Encoding& Run Length Encoding). We can divide it into two types based on their compression techniques.
- **Lossless Compression:** Encoding techniques (Run Length Encoding) allows a simple and minimal data size reduction. Lossless data compression uses algorithms to restore the precise original data from the compressed data.
- **Lossy Compression:** Methods such as Discrete Wavelet Transform technique, PCA (principal component analysis) are examples of this compression. For e.g., JPEG image format is a lossy compression, but we can find the meaning equivalent to the original image. In lossy data compression, the decompressed data may differ from the original data, but they are useful enough to retrieve information from them.

(a) Wavelet Transform

- In the wavelet transform, a data vector X is transformed into a numerically different data vector X' such that both X and X' vectors are of the same length. Then how it is useful in reducing data?
- The data obtained from the wavelet transform can be truncated. The compressed data is obtained by retaining the smallest fragment of the strongest of wavelet coefficients.
- Wavelet transform can be applied to data cube, sparse data or skewed data.

(b) Principal Component Analysis

- Let us consider that we have a data set to be analyzed having tuples with n attributes, then the principal component analysis (PCA) identifies k independent tuples with n attributes that can represent the data set.
- In this way, the original data can be cast on a much smaller space and the dimensionality reduction can be achieved.
- Principal component analysis can be applied to sparse, and skewed data.

(4) Numerosity Reduction

- Numerosity Reduction is a data reduction technique which replaces the original data by smaller form of data representation.
- There are two techniques for numerosity reduction: **Parametric** and **Non-Parametric** methods.

(a) Parametric Methods

- For parametric methods, data is represented using some model. The model is used to estimate the data, so that only parameters of data are required to be stored, instead of actual data.
- Regression and Log-Linear methods are used for creating such models.
- Regression and log-linear model can both be used on sparse data, although their application may be limited.

(i) Regression

- Regression can be a simple linear regression or multiple linear regression.
- When there is only single independent attribute, such regression model is called simple linear regression and if there are multiple independent attributes, then such regression models are called multiple linear regression.
- In linear regression, the data are modeled to a fit straight line. For example, a random variable *y* can be modeled as a linear function of another random variable *x* with the equation y = ax + b where a and b (regression coefficients) specifies the slope and y-intercept of the line, respectively.
- In multiple linear regression, y will be modeled as a linear function of two or more predictor(independent) variables.

(ii) Log-Linear Model

- Log-linear model can be used to estimate the probability of each data point in a multidimensional space for a set of discretized attributes, based on a smaller subset of dimensional combinations.
- This allows a higher-dimensional data space to be constructed from lower-dimensional attributes.

(b) Non-Parametric Methods

- These methods are used for storing reduced representations of the data include histograms, clustering, sampling and data cube aggregation.

- Different techniques used for non-parametric methods are:

(i) Histograms

Histogram is the data representation in terms of frequency. It uses binning to approximate data distribution and is a popular form of data reduction.

(ii) Clustering

- Clustering divides the data into groups/clusters. This technique partitions the whole data into different clusters.
- In data reduction, the cluster representation of the data is used to replace the actual data.
- It also helps to detect outliers in data.

(iii) Sampling

- Sampling can be used for data reduction because it allows a large data set to be represented by a much smaller random data sample (or subset).
- There are four types of sampling data reduction methods:
  - **(a) Simple random sampling :** There is an equal probability of selecting any particular item.
  - **(b) Sampling without replacement :** Once an object is selected, it is removed from the population.
  - **(c) Sampling with replacement :** A selected object is not removed from the population.
  - **(d) Stratified sampling :** Partition the data set, and draw samples from each partition (proportionally, i.e., approximately the same percentage of the data). Used in conjunction with skewed data.

(iv) Data Cube Aggregation

- Data cube aggregation involves moving the data from detailed level to a fewer number of dimensions.
- The resulting data set is smaller in volume, without loss of information necessary for the analysis task.

(5) Data Transformation and Discretization

- Data transformation in data mining is done for combining unstructured data with structured data to analyze it later. It is also important when the data is transferred to a new cloud data warehouse.
- When the data is homogeneous and well-structured, it is easier to analyze and look for patterns.
- For example, a company has acquired another firm and now has to consolidate all the business data. The smaller company may be using a different database than the parent firm. Also, the data in these databases may have unique IDs, keys and values. All this needs to be formatted so that all the records are similar and can be evaluated.
- This is why data transformation methods are applied. And, they are described below:

(i) Data Smoothing

- This method is used for removing the noise from a dataset. Noise is referred to as the distorted and meaningless data within a dataset.
- Smoothing uses algorithms to highlight the special features in the data.
- After removing noise, the process can detect any small changes to the data to detect special patterns.
- Any data modification or trend can be identified by this method.

(ii) Data Aggregation

- Aggregation is the process of collecting data from a variety of sources and storing it in a single format. Here, data is collected, stored, analyzed and presented in a report or summary format.
- It helps in gathering more information about a particular data cluster. The method helps in collecting vast amounts of data.
- This is a crucial step as accuracy and quantity of data is important for proper analysis.
- Companies collect data about their website visitors. This gives them an idea about customer demographics and behaviour metrics. This aggregated data assists them in designing personalized messages, offers and discounts.

(iii) Discretization

- This is a process of converting continuous data into a set of data intervals. Continuous attribute values are substituted by small interval labels. This makes the data easier to study and analyze.

- If a continuous attribute is handled by a data mining task, then its discrete values can be replaced by constant quality attributes. This improves the efficiency of the task.
- This method is also called data reduction mechanism as it transforms a large dataset into a set of categorical data.
- Discretization can be done by Binning, Histogram Analysis, and Correlation Analyses.
- Discretization also uses decision tree-based algorithms to produce short, compact and accurate results when using discrete values.

(iv) Generalization

- In this process, low-level data attributes are transformed into high-level data attributes using concept hierarchies. This conversion from a lower level to a higher conceptual level is useful to get a clearer picture of the data.
- For example, age data can be in the form of (20, 30) in a dataset. It is transformed into a higher conceptual level into a categorical value (young, old).
- Data generalization can be divided into two approaches – data cube process (OLAP) and attribute oriented induction approach (AOI).

(v) Attribute construction

- In the attribute construction method, new attributes are created from an existing set of attributes.
- For example, in a dataset of employee information, the attributes can be employee name, employee ID and address.
- These attributes can be used to construct another dataset that contains information about the employees who have joined in the year 2019 only.
- This method of reconstruction makes mining more efficient and helps in creating new datasets quickly.

(vi) Normalization

- Also called data pre-processing, this is one of the crucial techniques for data transformation in data mining.
- Here, the data is transformed so that it falls under a given range. When attributes are on different ranges or scales, data modelling and mining can be difficult.
- Normalization helps in applying data mining algorithms and extracting data faster.
- The popular normalization methods are:

(a) Min-max normalization

In this technique of data normalization, linear transformation is performed on the original data. Minimum and maximum value from data is fetched and each value is replaced according to the following formula:

$$v' = \frac{v - \min_A}{\max_A - \min_A} (\text{new\_}\max_A - \text{new\_}\min_A) + \text{new\_}\min_A$$

Where A is the attribute data,

$\min_A$, $\max_A$ are the minimum and maximum absolute value of A respectively,

v' is the new value of each entry in data,

v is the old value of each entry in data,

new_max_A, new_min_A is the max and min value of the range (i.e. boundary value of range required) respectively.

Example: Suppose the income range from \$10,000 to \$95,000 is normalized to [0.0, 1.0]. By min-max normalization, a value of \$64,300 for income is transformed to

$$\frac{64300 - 10000}{95000 - 10000}(1.0 - 0.0) + 0.0 = 0.6388$$

(b) Z-score normalization

In this technique, values are normalized based on mean and standard deviation of the data A. The formula used is:

$$v' = \frac{v - \mu_A}{\sigma_A}$$

where v', v is the new and old value of each entry in data respectively. σ_A, μ_A is the standard deviation and mean of A respectively.

Example: If mean salary is \$54,000 and standard deviation is \$16,000, then the z-score value of salary \$73,600 will be $\frac{73600 - 54000}{16000} = 1.225$

(c) Decimal scaling

It normalizes by moving the decimal point of values of the data. To normalize the data by this technique, we divide each value of the data by the maximum absolute value of data. The data value, v_i, of data is normalized to v_i' by using the formula below:

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

where, j is the smallest integer such that max $(|v_i'|) < 1$.

Example : Let the input data be: -10, 201, 301, – 401, 501, 601, 701

To normalize the above data,

- **Step 1 :** Maximum absolute value in given data(m): 701
- **Step 2 :** Divide the given data by 1000 (i.e. j = 3)

Result : The normalized data is: – 0.01, 0.201, 0.301, – 0.401, 0.501, 0.601, 0.701

UEx. 2.24.1 MU - June 2021

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, 23, 29, 33, 41, 44, 53, 62, 69, 72

Use min-max normalization to transform the value 45 for age onto the range [0.0, 1.0].

☑ **Soln. :**

By min-max normalization,

$$v' = \frac{v - \min_A}{\max_A - \min_A} (new\_\max_A - new\_\min_A) + new\_\min_A$$

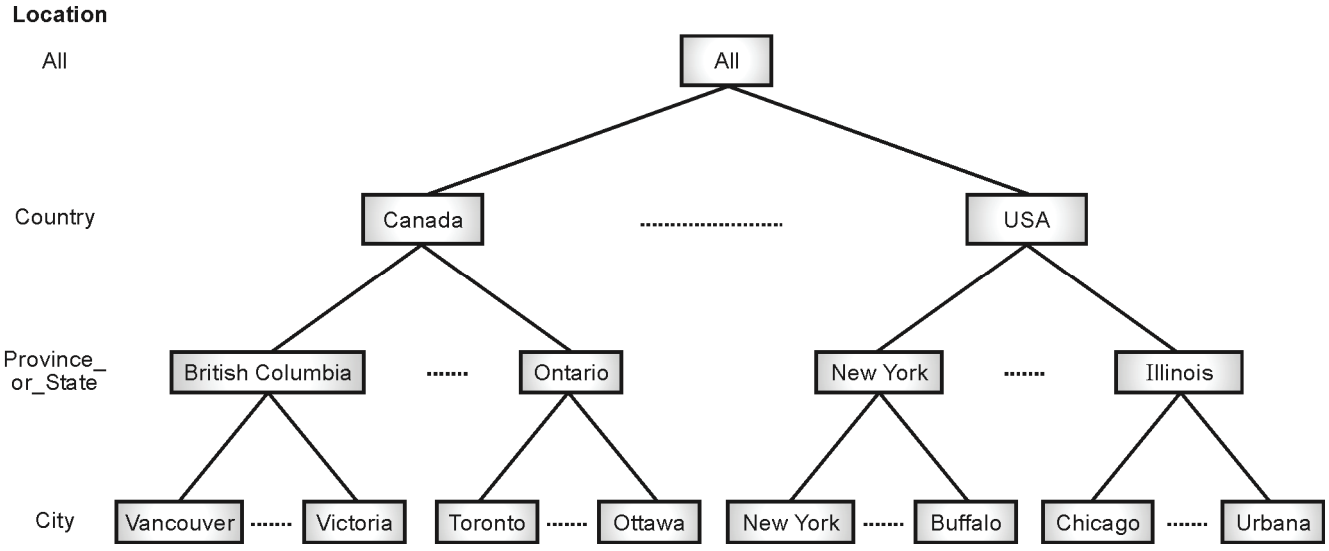
where $\min_A = 13$, $\max_A = 72$, $v = 45$, $new\_\max_A = 1.0$, $new\_\min_A = 0.0$

$$v' = \frac{45 - 13}{72 - 13} (1.0 - 0.0) + 0.0 = 0.5423$$

Therefore, the value 45 for age is transformed to 0.5423

(6) Concept Hierarchy Generation

- Concept hierarchies can be used to reduce the data by collecting and replacing low-level concepts with higher-level concepts. For example, the numeric value for age may be represented as Young, Middle-aged or Senior.
- In the multidimensional model, data are organized into multiple dimensions, and each dimension contains multiple levels of abstraction defined by concept hierarchies. This organization provides users with the flexibility to view data from different perspectives.
- Data mining on a reduced data set means fewer input/output operations and is more efficient than mining on a larger data set.
- Because of these benefits, discretization techniques and concept hierarchies are typically applied before data mining, rather than during mining.



(1B29)**Fig. 2.14.2 : Concept Hierarchy**

⏭ 2.15 MULTIPLE CHOICE QUESTIONS

Q. 2.1 Which of the following is an essential process in which the intelligent methods are applied to extract data patterns?
(a) Warehousing (b) Data Mining
(c) Text Mining (d) Data Selection ✓**Ans. : (b)**

Q. 2.2 What is KDD in data mining?
(a) Knowledge Discovery Database
(b) Knowledge Discovery Data
(c) Knowledge Data Definition
(d) Knowledge Data Discovery ✓**Ans. : (a)**

Q. 2.3 _____ predicts future trends and behaviors, allowing business managers to make proactive, knowledge-driven decisions.
(a) Data warehouse (b) Data mining
(c) Datamarts (d) Metadata ✓**Ans. : (b)**

Q. 2.4 Data transformation includes _____.
(a) a process to change data from a detailed level to a summary level
(b) a process to change data from a summary level to a detailed level
(c) joining data from one source into various sources of data
(d) separating data from one source into various sources of data. ✓**Ans. : (a)**

Q. 2.5 Which of the following achieves data reduction by detecting redundant attributes?
(a) Data cube aggregation
(b) Dimension reduction
(c) Compression
(d) Numerosity reduction ✓**Ans. : (c)**

Q. 2.6 Dimensionality reduction reduces the data set size by removing _____.
(a) relevant attributes (b) irrelevant attributes
(c) derived attributes (d) composite attributes.
✓**Ans. : (b)**

Q. 2.7 _____ is the input to KDD.
(a) Data (b) Information
(c) Query (d) Process ✓**Ans. : (a)**

Q. 2.8 The output of KDD is _____.
(a) Data (b) Information
(c) Query (d) Useful information ✓**Ans. : (d)**

Q. 2.9 The KDD process consists of _____ steps.
(a) three (b) four
(c) five (d) six ✓**Ans. : (c)**

Q. 2.10 Treating incorrect or missing data is called as _____.
(a) selection (b) preprocessing
(c) transformation (d) interpretation ✓**Ans. : (b)**

Q. 2.11 Converting data from different sources into a common format for processing is called as _____.
(a) selection (b) preprocessing
(c) transformation (d) interpretation ✓**Ans. : (c)**

Q. 2.12 Various visualization techniques are used in _____ step of KDD.
(a) selection (b) transformation
(c) data mining (d) interpretation ✓**Ans. : (d)**

Q. 2.13 Extreme values that occur infrequently are called as _____.
(a) outliers (b) rare values
(c) dimensionality reduction
(d) Missing values ✓**Ans. : (a)**

Q. 2.14 The _____ is the most common measure of the location of a set of points.
(a) Mean (b) Median
(c) Mode (d) standard deviation ✓**Ans. : (a)**

Q. 2.15 The _____ is the most common measure of the spread of a set of points.
(a) Mean (b) Median
(c) Mode (d) standard deviation ✓**Ans. : (d)**

Q. 2.16 _____ of data is one of the most powerful and appealing techniques for data exploration.
(a) Analysis (b) Visualization
(c) OLAP (d) Exploration ✓**Ans. : (b)**

Q. 2.17 _____ can detect outliers of data and unusual patterns.
(a) Box plot (b) Matrix plot
(c) Histogram (d) Scatter plot ✓**Ans. : (a)**

Q. 2.18 Usually _____ shows the distribution of values of a single variable and divide the values of objects into bins.
(a) Box plot (b) Matrix plot
(c) Histogram (d) Scatter plot ✓**Ans. : (c)**

Q. 2.19 It is useful to have arrays of _____ plots can compactly summarize the relationships of several pairs of attributes.
(a) box (b) contour
(c) histogram (d) scatter ✓**Ans. : (d)**

Q. 2.20 Which one of the following can be defined as the data object which does not comply with the general behavior (or the model of available data)?
(a) Evaluation Analysis (b) Outlier Analysis
(c) Classification (d) Prediction
✓**Ans. : (b)**

Q. 2.21 Which of the following refers to the steps of the knowledge discovery process, in which the several data sources are combined?
(a) Data selection (b) Data cleaning
(c) Data transformation (d) Data integration
✓**Ans. : (d)**

Q. 2.22 _____ routines attempt to fill in missing values, smooth out noise while identifying outlines, and correct inconsistencies in the data.
(a) Data selection (b) Data cleaning
(c) Data transformation (d) Data integration
✓**Ans. : (b)**

Q. 2.23 In _____ each value in a bin is replaced by the mean value of the bin.
(a) Smoothing by bin means
(b) Smoothing by bin medians
(c) Smoothing by bin boundary values
(d) Histogram analysis ✓**Ans. : (a)**

Q. 2.24 _____ involves finding the "best" line to fit two variables so that one variable can be used to predict the other.
(a) Linear Regression (b) Multiple Regression
(c) Log Linear Model (d) Boxplot ✓**Ans. : (a)**

Q. 2.25 The _____ technique uses encoding mechanisms to reduce the data set size.
(a) Dimensionality Reduction
(b) Data Compression
(c) Numerosity Reduction
(d) Data Transformation ✓**Ans. : (b)**

Q. 2.26 In which Strategy of data reduction redundant attributes are detected.
(a) Data cube aggregation
(b) Numerosity reduction
(c) Data compression
(d) Dimension reduction ✓**Ans. : (d)**

Q. 2.27 _____ can be used to reduce the data by collecting and replacing low-level concepts by higher-level concepts.
(a) Data Transformation (b) Data Discretization
(c) Concept Hierarchies (d) Data Compression
✓**Ans. : (c)**

Q. 2.28 The fraudulent usage of credit card can be detected using _____ data mining task.
(a) Prediction (b) Outlier Analysis
(c) Association Analysis (d) Correlation
✓**Ans. : (b)**

Q. 2.29 Five number summary of a distribution (Minimum, Q. 1, Median, Q. 3, Maximum) is displayed using _____
(a) Histogram (b) Quantile Plot
(c) Scatter Plot (d) Box Plot ✓**Ans. : (d)**

Q. 2.30 If the mean salary is Rs. 54,000 and the standard deviation is Rs. 16,000, then find z-score value of Rs.73,600 salary.
(a) 1.225 (b) 0.351
(c) 1.671 (d) 1.862 ✓**Ans. : (a)**

Q. 2.31 In KDD and data mining, noise is referred to as _____.
(a) Complex Data (b) Meta Data
(c) Error (d) Repeated Data ✓**Ans. : (d)**

Q. 2.32 Find IQR of the data set {3, 7, 8, 5, 12, 14, 21, 13, 18}
(a) 6 (b) 8 (c) 12 (d) 10 ✓**Ans. : (d)**

Q. 2.33 Which are not related to Ratio Attributes?
(a) Age Group 10-20, 30-50, 35-45 (in Years)
(b) Mass 20-30 kg, 10-15 kg
(c) Areas 10-50, 50-100 (in Kilometres)
(d) Temperature 10°-20°, 30°-50°, 35°-45°
✓**Ans. : (d)**

Q. 2.34 Find the range for given data 40, 30, 43, 48, 26, 50, 55, 40, 34, 42, 47, 50.
(a) 19 (b) 29 (c) 35 (d) 49 ✓**Ans. : (b)**

Q. 2.35 The number that occurs most often within a set of data called as _____.
(a) Mean (b) Median
(c) Mode (d) Range ✓**Ans. : (c)**

Descriptive Questions

Q. 1 Define Data Mining. Explain data mining task primitives.

Q. 2 Explain Data Mining Architecture with neat diagram.

Q. 3 Explain KDD process in detail.

Q. 4 Describe the steps involved in Data Mining when viewed as a process of Knowledge Discovery.
(MU - Dec. 2019)

Q. 5 Explain in detail the issues in data mining.

Q. 6 Write a note on: Applications of Data Mining

Q. 7 Differentiate between Data Mining and Data Warehouse

Q. 8 Explain different types of attributes with suitable example.

Q. 9 Briefly outline with example, how to compute the dissimilarity between objects described by the following:

(i) Nominal attributes

(ii) Asymmetric binary attributes **(MU - May 2019)**

Q. 10 The data set below gives the number of goals for the 10 players who scored the most goals during the 2003 –2004 National Hockey League regular season. 41, 41, 41, 38, 38, 36, 35, 35, 34, 33. Find the mean, median, mode and the midrange of the data set. Also draw the boxplot.

Q. 11 Suppose that the data for analysis includes the attribute salary. We have the following values for salary (in thousands of dollars), shown in increasing order: 30, 36, 47, 50, 52, 52, 56, 60, 63, 70, 70, 110.

(i) What are the mean, median, mode and midrange of the data?

(ii) Find the first quartile (Q_1) and the third quartile (Q_3) of the data.

(iii) Show the boxplot of the data.

(MU - Dec. 2019)

Q. 12 In real-world data, tuples with missing values for some attributes are a common occurrence. Describe various methods for handling this problem.

(MU - May 2019)

Q. 13 Develop a model to predict the salary of college graduates with 10 years of work experience using linear regression. **(MU - Dec. 2019)**

Years of Experience (x)	3	8	9	13	3	6	11	21	1	16
Salary in $100 (y)	30	57	64	72	36	43	59	90	20	83

Q. 14 Explain the techniques in data cleaning process.

Q. 15 Suppose a group of sales price records has been sorted as follows: 8, 9, 15, 16, 21, 21, 24, 26, 27, 30, 30, 34. Partition them into three bins by equal-frequency (Equi-depth) partitioning method. Perform data smoothing by bin mean.

Q.16 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, 23, 29, 33, 41, 44, 53, 62, 69, 72

Use min-max normalization to transform the value 45 for age onto the range [0.0, 1.0].

(MU - June 2021)

Q. 17 Explain different data reduction techniques.

Q. 18 Explain data transformation and discretization in detail.

Q. 19 Explain numerosity reduction in data preprocessing.

Q. 20 Explain data visualization in detail.

Chapter Ends…

❑❑❑

MODULE 3

CHAPTER 3

Classification

University Prescribed Syllabus w.e.f Academic Year 2021-2022

Basic Concepts, Decision Tree Induction, Naïve Bayesian Classification, Accuracy and Error measures, Evaluating the Accuracy of a Classifier: Holdout & Random Subsampling, Cross Validation, Bootstrap.

3.1 INTRODUCTION

- There are two forms of data analysis that can be used for extracting models describing important classes or to predict future data trends.
- These two forms are as follows:
  (i) Classification
  (ii) Prediction
- Classification is a type of data analysis in which models defining relevant data classes are extracted.
- Classification models, called classifier, predict categorical class labels; and prediction models predict continuous valued functions.
- For example, we can build a classification model to categorize bank loan applications as either safe or risky, or a prediction model to predict the expenditures in dollars of potential customers on computer equipment given their income and occupation.

3.2 BASIC CONCEPTS

In this section we will discuss what classification is, working of classification, issues in classification and criteria for comparing the methods of classification.

3.2.1 What is Classification?

- Following are the examples of cases where the data analysis task is Classification –
  (i) A bank loan officer wants to analyze the data in order to know which customer (loan applicant) are risky or which are safe.
  (ii) A marketing manager at a company needs to analyze a customer with a given profile, who will buy a new laptop.
- In both of the above examples, a model or classifier is constructed to predict the categorical labels. These labels are "risky" or "safe" for loan application data and "yes" or "no" for marketing data.

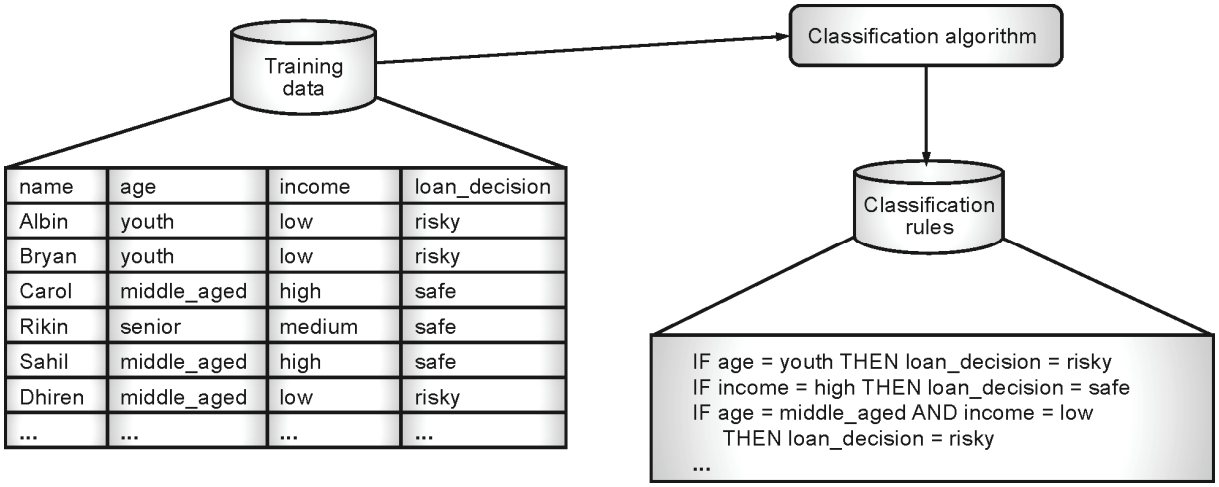
3.2.2 How Does Classification Work?

With the help of the bank loan application that we have discussed above, let us understand the working of classification. The Data Classification process includes two steps –

1. Building the Classifier or Model
2. Using Classifier for Classification

1. Building the Classifier or Model

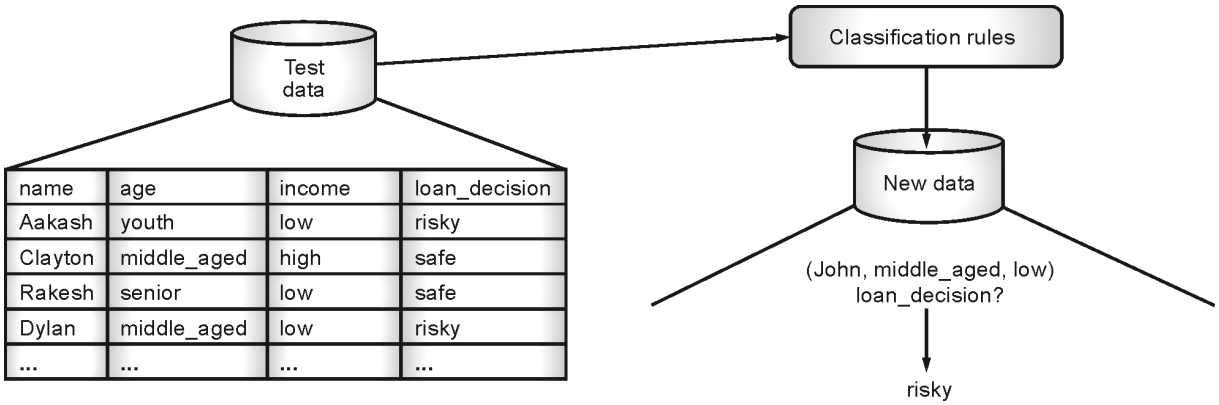
- This step is the learning step or the learning phase.
- In this step the classification algorithms build the classifier.
- The classifier is built from the training set made up of database tuples and their associated class labels.
- Each tuple that constitutes the training set is referred to as a category or class. These tuples can also be referred to as sample, object or data points.



(1C1)**Fig. 3.2.1: Building a Classifier**

2. Using Classifier for Classification

In this step, the classifier is used for classification. Here the test data is used to estimate the accuracy of classification rules. The classification rules can be applied to the new data tuples if the accuracy is considered acceptable.



name	age	income	loan_decision
Aakash	youth	low	risky
Clayton	middle_aged	high	safe
Rakesh	senior	low	safe
Dylan	middle_aged	low	risky
...	...	...	...

(1C2)**Fig. 3.2.2 : Testing a Classifier**

3.2.3 Classification Issues

The major issue is preparing the data for Classification. Preparing the data involves the following activities:

- **Data Cleaning :** Data cleaning involves removing the noise and treatment of missing values. The noise is removed by applying smoothing techniques and the problem of missing values is solved by replacing a missing value with most commonly occurring value for that attribute.
- **Relevance Analysis :** Database may also have the irrelevant attributes. Correlation analysis is used to know whether any two given attributes are related.
- **Data Transformation and Reduction :** The data can be transformed by any of the following methods.
  - **(i) Normalization :** The data is transformed using normalization. Normalization involves scaling all values for given attribute in order to make them fall within a small specified range. Normalization is used when in the learning step, the neural networks or the methods involving measurements are used.
  - **(ii) Generalization**: The data can also be transformed by generalizing it to the higher concept. For this purpose, we can use the concept hierarchies.
  - (iii) Data can also be reduced by some other methods such as wavelet transformation, binning, histogram analysis and clustering.

3.2.4 Comparison of Classification Methods

Here are the criteria for comparing the methods of Classification.

- **Accuracy :** Accuracy of classifier refers to the ability of classifier. It predicts the class label correctly and the accuracy of the predictor refers to how well a given predictor can guess the value of predicted attribute for a new data.
- **Speed :** This refers to the computational cost in generating and using the classifier or predictor.
- **Robustness :** It refers to the ability of classifier or predictor to make correct predictions from given noisy data.
- **Scalability :** Scalability refers to the ability to construct the classifier or predictor efficiently; given large amount of data.
- **Interpretability :** It refers to what extent the classifier or predictor understands.

3.3 DECISION TREE INDUCTION

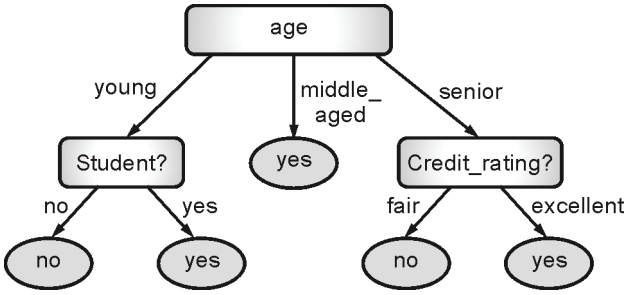
UQ. Why is tree pruning useful in decision tree induction? What is a drawback of using a separate set of tuples to evaluate pruning? Given a decision tree, you have the option of (a) converting the decision tree to rules and then pruning the resulting rules, or (b) pruning the decision tree and then converting the pruned tree to rules. What advantage does (a) have over (b)?

MU - May 2019

UQ. Why is tree pruning useful in decision tree induction? What is a drawback of using a separate set of tuples to evaluate pruning?

MU – Dec. 2019

- Decision tree induction is the learning of decision trees from class-labeled training tuples.
- A decision tree is a structure that includes a root node, branches, and leaf nodes.
- Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label.
- The topmost node in the tree is the root node.
- The following decision tree is for the concept buys_laptop that indicates whether a customer at a company is likely to buy a computer or not. Each internal node represents a test on an attribute. Each leaf node represents a class (either buys_laptop = yes or buys_laptop = no)



(1C3)**Fig. 3.3.1: Representation of a Decision Tree**

- The benefits of having a decision tree are as follows:
  1. It does not require any domain knowledge.
  2. It is easy to comprehend.
  3. The learning and classification steps of a decision tree are simple and fast.

3.3.1 Decision Tree Induction Algorithm

- A machine learning researcher named J. Ross Quinlan in 1980 developed a decision tree algorithm known as ID3 (Iterative Dichotomiser). Later, he presented C4.5, which was the successor of ID3. ID3 and C4.5 adopt a greedy approach. In this algorithm, there is no backtracking; the trees are constructed in a top-down recursive divide-and-conquer manner.
- **Algorithm : Generate_decision_tree.** Generating a decision tree form training tuples of data partition D

Input

- Data partition, D, which is a set of training tuples and their associated class labels.
- attribute_list, the set of candidate attributes.
- Attribute_selection_method, a procedure to determine the splitting criterion that "best" partitions the data tuples into individual classes. This criterion includes a splitting_attribute and either a splitting point or splitting subset.

Output : A Decision Tree

☞ **Method**

1. create a node N;
2. **if** tuples in D are all of the same class, C,**then**
3. return N as leaf node labeled with class C;
4. **if** attribute_list is empty **then**
5. return N as a leaf node with labeled with majority class in D; //majority voting
6. apply **attribute_selection_method**(D, attribute_list) to **find** the best splitting_criterion;
7. label node N with splitting_criterion;
8. **if** splitting_attribute is discrete-valued **and** multiway splits allowed **then**// no restricted to binary trees
9. attribute_list←attribute_list - splitting attribute; // remove splitting attribute
10. **for each** outcome j of splitting criterion

    // partition the tuples and grow subtrees for each partition
11. let D_j be the set of data tuples in D satisfying outcome j; // a partition
12. **if** D_j is empty **then**
13. attach a leaf labeled with the majority class in D to node N;
14. **else** attach the node returned by Generate_decision_tree (D_j, attribute list) to node N;

    end for

    return N;

3.3.2 Tree Pruning

- The decision tree built may overfit the training data. There could be too many branches, some of which may reflect anomalies in the training data due to noise or outliers.
- Tree pruning addresses this issue of overfitting the data by removing the least reliable branches (using statistical measures).
- This generally results in a more compact and reliable decision tree that is faster and more accurate in its classification of data.
- There are two approaches to prune a tree:
  - **(i) Pre-pruning –** The tree is pruned by halting its construction early.
  - **(ii) Post-pruning** - This approach removes a sub-tree from a fully grown tree.

Drawback of using a separate set of tuples to evaluate pruning

- If a separate set of tuples are used to evaluate pruning is that it may not be representative of the training tuples used to create the original decision tree.
- If the separate set of tuples are skewed, then using them to evaluate the pruned tree would not be a good indicator of the pruned tree's classification accuracy.
- Furthermore, using a separate set of tuples to evaluate pruning means there are less tuples to use for creation and testing of the tree. While this is considered a drawback in machine learning, it may not be so in data mining due to the availability of larger data sets.

GQ. Given a decision tree, you have the option of (a) converting the decision tree to rules and then pruning the resulting rules, or (b) pruning the decision tree and then converting the pruned tree to rules. What advantage does (a) have over (b)?

If pruning a subtree, we would remove the subtree completely with method (b). However, with method (a), if pruning a rule, we may remove any precondition of it. The latter is less restrictive.

3.3.3 Cost Complexity

The cost complexity is measured by the following two parameters –

- Number of leaves in the tree, and
- Error rate of the tree

3.3.4 Classification using Information Gain (ID3)

- ID3 stands for Iterative Dichotomiser 3 and is named such because the algorithm iteratively (repeatedly) dichotomizes(divides) features into two or more groups at each step.
- Invented by Ross Quinlan, ID3 uses a **top-down greedy** approach to build a decision tree. In simple words, the **top-down** approach means that we start building the tree from the top and the **greedy** approach means that at each iteration we select the best feature at the present moment to create a node.
- Most generally ID3 is used for classification problems with nominal features only.

Metrics in ID3

As mentioned previously, the ID3 algorithm selects the best feature at each step while building a Decision tree.

- ID3 uses Information Gain or just Gain to find the best feature.
- **Information Gain** calculates the reduction in the entropy and measures how well a given feature separates or classifies the target classes. The feature with the highest Information Gain is selected as the best one.
- In simple words, **Entropy** is the measure of disorder and the Entropy of a dataset is the measure of disorder in the target feature of the dataset. Expected amount of information (in bits) needed to assign a class to a randomly drawn object is called Entropy.
- In the case of binary classification (where the target column has only two types of classes) entropy is 0 if all values in the target column are homogenous(similar) and will be 1 if the target column has equal number values for both the classes.
  - We denote our dataset as D**,** entropy is calculated as:

$$\text{Entropy (D)} = \sum_{i=1}^{n} p_i \log_2 \frac{1}{p_i}$$

where,

n is the total number of classes in the target column

$\mathbf{p_i}$ is the **probability of class 'i'** or the ratio of "*number of rows with class i in the target column"* to the "*total number of rows"* in the dataset.

Information Gain for a feature column **A** is calculated as:

$$\text{Gain(A)} = \text{Entropy(D)} - \text{Entropy(A)}$$

ID3 Steps

1. Calculate the Information Gain of each feature.
2. Considering that all rows don't belong to the same class, split the dataset D into subsets using the feature for which the Information Gain is maximum.
3. Make a decision tree node using the feature with the maximum Information gain.
4. If all rows belong to the same class, make the current node as a leaf node with the class as its label.
5. Repeat the above steps for the remaining features until we run out of all features, or the decision tree has all leaf nodes.

Ex. 3.3.1 : Apply ID3 algorithm on the following training dataset and extract the classification rule from the tree.

Day	Outlook	Temp.	Humidity	Wind	Play_Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

☑ **Soln. :**

Let the class label attributes be as follows:

C1 = Play_Tennis = Yes = 9 Samples

C2 = Play_Tennis = No = 5 Samples

Therefore, P(C1) = 9/14 and P(C2) = 5/14

(i) Entropy before split for the given database D:

$$H(D) = \sum_{i=1}^{s} p_i \log_2 \left(\frac{1}{p_i}\right) = \frac{9}{14}\log_2 \frac{14}{9} + \frac{5}{14}\log_2 \frac{14}{5} =$$

$0.4097 + 0.5305 = 0.940$

(ii) Choosing Outlook as Splitting Attribute

Outlook	C1 Play_Tennis = Yes	C2 Play_Tennis = No	Entropy H
Sunny	2	3	0.971
Overcast	4	0	0
Rain	3	2	0.971

$$\therefore H(\text{outlook}) = \frac{5}{14} \times H(\text{Sunny}) + \frac{4}{14} \times H(\text{Overcast}) + \frac{5}{14} \times H(\text{Rain})$$

$$= \frac{5}{14} \times 0.971 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0.971 = 0.694$$

Gain (Outlook) = H(D) – H(Outlook) = 0.940 – 0.694

= 0.246

(iii) Choosing Temperature as Splitting Attribute

Temperature	C1 Play_Tennis = Yes	C2 Play_Tennis = No	Entropy H
Hot	2	2	1
Mild	4	2	0.92
Cool	3	1	0.81

$$\therefore H(\text{Temperature}) = \frac{4}{14} \times H(\text{Hot}) + \frac{6}{14} \times H(\text{Mild}) + \frac{4}{14} \times H(\text{Cool})$$

$$= \frac{4}{14} \times 1 + \frac{6}{14} \times 0.92 + \frac{4}{14} \times 0.81 = 0.911$$

Gain (Temperature) = H(D) – H(Temperature)

= 0.940 – 0.911 = 0.029

(iv) Choosing Humidity as Splitting Attribute

Humidity	C1 Play_Tennis = Yes	C2 Play_Tennis = No	Entropy H
High	3	4	0.985
Normal	6	1	0.592

$$\therefore H(Humidity) = \frac{7}{14} \times H(High) + \frac{7}{14} \times H\ (Normal)$$

$$= \frac{7}{14} \times 0.985 + \frac{7}{14} \times 0.592 = 0.789$$

$$Gain\ (Humidity) = H(D) - H(Humidity)$$

$$= 0.940 - 0.789 = 0.151$$

(v) Choosing Wind as Splitting Attribute

Wind	C1 Play_Tennis = Yes	C2 Play_Tennis = No	Entropy H
Strong	3	3	1
Weak	6	2	0.811

$$\therefore H(Wind) = \frac{6}{14} \times H(Strong) + \frac{8}{14} \times H\ (Weak)$$

$$= \frac{6}{14} \times 1 + \frac{8}{14} \times 0.811 = 0.892$$

$$Gain\ (Wind) = H(D) - H(Wind) = 0.940 - 0.892 = 0.048$$

Summary:

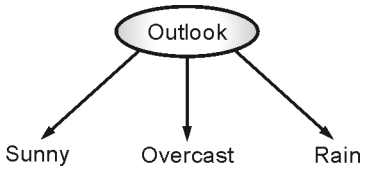
$$Gain(Outlook|D) = 0.246$$

$$Gain(Temperature|D) = 0.029$$

$$Gain(Humidity|D) = 0.151$$

$$Gain(Wind|D) = 0.048$$

Outlook attribute has the highest gain; therefore, it is used as the decision attribute in the root node. Since, Outlook has three possible values, the root node has three branches (Sunny, Overcast, Rain).



(1C4)**Fig. P. 3.3.1(a)**

Now, consider Outlook = Sunny and count the number of tuples from the original dataset D. Let us denote it as D1.

Day	Outlook	Temp.	Humidity	Wind	Play_Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Let the class label attributes be as follows:

C1 = Play_Tennis = Yes|Sunny = 2 Samples

C2 = Play_Tennis = No|Sunny = 3 Samples

Therefore, P(C1) = 2/5 and P(C2) = 3/5

(i) Entropy before split for the given database D1:

$$H(D_1) = \sum_{i=1}^{s} p_i \log_2\left(\frac{1}{p_i}\right) = \frac{2}{5}\log_2\frac{5}{2} + \frac{3}{5}\log_2\frac{5}{3} = 0.971$$

(ii) Choosing Temperature as Splitting Attribute

Temperature	C1 Play_Tennis = Yes|Sunny	C2 Play_Tennis = No|Sunny	Entropy H
Hot	0	2	0
Mild	1	1	1
Cool	1	0	0

$$\therefore H(Temperature) = \frac{2}{5} \times H(Hot) + \frac{2}{5} \times H(Mild) + \frac{1}{5} \times H(Cool)$$

$$= \frac{2}{5} \times 0 + \frac{2}{5} \times 1 + \frac{1}{5} \times 0 = 0.4$$

$$Gain\ (Temperature) = H(D1) - H(Temperature) = 0.971 - 0.4 = 0.571$$

(iii) Choosing Humidity as Splitting Attribute

Humidity	C1 Play_Tennis = Yes|Sunny	C2 Play_Tennis = No|Sunny	Entropy H
High	0	3	0
Normal	2	0	0

$$H(Humidity) = \frac{3}{5} \times H(High) + \frac{2}{5} \times H\ (Normal)$$

$$= \frac{3}{5} \times 0 + \frac{2}{5} \times 0 = 0$$

$$Gain\ (Humidity) = H(D1) - H(Humidity) = 0.971 - 0$$

$$= 0.971$$

(iv) Choosing Wind as Splitting Attribute

Wind	C1 Play_Tennis = Yes|Sunny	C2 Play_Tennis = No|Sunny	Entropy H
Strong	1	1	1
Weak	1	2	0.918

$$\therefore \text{Gain (Wind)} = \frac{2}{5} \times \text{H (Strong)} + \frac{3}{5} \times \text{H (Weak)}$$

$$= \frac{2}{5} \times 1 + \frac{3}{5} \times 0.918 = 0.951$$

$$\text{Gain (Wind)} = H(D_1) - H(\text{Wind})$$

$$= 0.971 - 0.951 = 0.02$$

Summary:

$$\text{Gain(Temperature|D1)} = 0.571$$

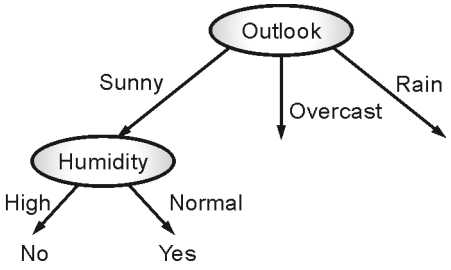
$$\text{Gain(Humidity|D1)} = 0.971$$

$$\text{Gain(Wind|D1)} = 0.02$$

Humidity attribute has the highest gain; therefore, it is placed below Outlook = "Sunny".

Since, Humidity has two possible values, the Humidity node has two branches (High, Normal).

From dataset D1, we find that when Humidity = High, Play_Tennis = No and when Humidity = Normal, Play_Tennis = Yes.

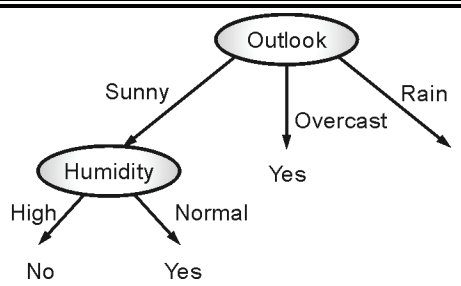


(1C5)**Fig. P.3.3.1(b)**

- Now, consider Outlook = Overcast and count the number of tuples from the original dataset D. Let us denote it as D2.

Day	Outlook	Temp.	Humidity	Wind	Play_Tennis
3	Overcast	Hot	High	Weak	Yes
7	Overcast	Cool	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes

- From dataset D2, we find that for all values of Outlook = "Overcast", Play_Tennis = Yes.



(1C6) **Fig. P.3.3.1(c)**

- Now, consider Outlook = Rain and count the number of tuples from the original dataset D. Let us denote it as D3.

Day	Outlook	Temp.	Humidity	Wind	Play_Tennis
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Let the class label attributes be as follows:

C1 = Play_Tennis = Yes|Rain = 3 Samples

C2 = Play_Tennis = No|Rain = 2 Samples

Therefore, P(C1) = 3/5 and P(C2) = 2/5

(i) Entropy before split for the given database D3:

$$H(D3) = \sum_{i=1}^{s} p_i \log_2 \left(\frac{1}{p_i}\right)$$

$$\therefore H(D3) = \frac{3}{5}\log_2 \frac{5}{3} + \frac{2}{5}\log_2 \frac{5}{2} = 0.971$$

(ii) Choosing Temperature as Splitting Attribute

Temperature	C1 Play_Tennis = Yes|Rain	C2 Play_Tennis = No|Rain	Entropy H
Hot	0	0	0
Mild	2	1	0.918
Cool	1	1	1

$$\therefore \text{H (Temperature)} = \frac{0}{5} \times \text{H(Hot)} + \frac{3}{5} \times \text{H(Mild)} + \frac{2}{5} \times \text{H(Cool)}$$

$$= \frac{0}{5} \times 0 + \frac{3}{5} \times 0.918 + \frac{2}{5} \times 1 = 0.951$$

$$\text{Gain (Temperature)} = H(D3) - H(\text{Temperature})$$

$$= 0.971 - 0.951 = 0.02$$

(iii) Choosing Wind as Splitting Attribute

Wind	C1 Play_Tennis = Yes|Rain	C2 Play_Tennis = No|Rain	Entropy H
Strong	0	2	0
Weak	3	0	0

$$\therefore \text{H (Wind)} = \frac{2}{5} \times \text{H (Strong)} + \frac{3}{5} \times \text{H (Weak)}$$

$$= \frac{2}{5} \times 0 + \frac{3}{5} \times 0 = 0$$

Gain (Wind) = H(D3) – H(Wind) = 0.971 – 0 = 0.971

Summary:

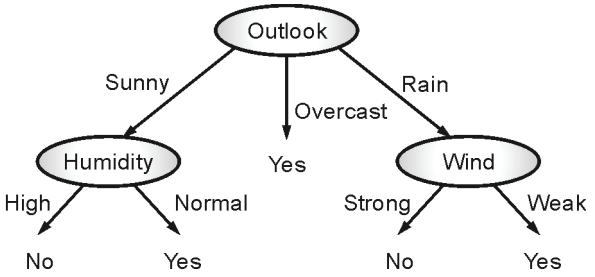
Gain(Temperature|D3) = 0.02

Gain(Wind|D3) = 0.971

Wind attribute has the highest gain; therefore, it is placed below Outlook = "Rain".

Since, Wind has two possible values, the Wind node has two branches (Strong, Weak).

From dataset D3, we find that when Wind = Strong, Play_Tennis = No and when Wind = Weak, Play_Tennis = Yes.



(1C7) **Fig. P.3.3.1(d)**

The decision tree can also be expressed in rule format as:

IF Outlook = Sunny AND Humidity

= High THEN Play_Tennis = No

IF Outlook = Sunny AND Humidity

= Normal THEN Play_Tennis = YES

IF Outlook = Overcast THEN Play_Tennis = YES

IF Outlook = Rain AND Wind

= Strong THEN Play_Tennis = No

IF Outlook = Rain AND Wind

= Weak THEN Play_Tennis = YES

Ex. 3.3.2 : A simple example from the stock market involving only discrete ranges has profit as categorical attribute with values {Up, Down} and the training data is:

Age	Competition	Type	Profit
Old	Yes	Software	Down
Old	No	Software	Down
Old	No	Hardware	Down
Mid	Yes	Software	Down
Mid	Yes	Hardware	Down
Mid	No	Hardware	Up
Mid	No	Software	Up
New	Yes	Software	Up
New	No	Hardware	Up
New	No	Software	Up

Apply decision tree algorithm and show the generated rules.

☑ **Soln. :**

Let the class label attributes be as follows:

C1 = Profit = Down = 5 Samples

C2 = Profit = Up = 5 Samples

Therefore, P(C1) = 5/10 and P(C2) = 5/10

(i) Entropy before split for the given database D :

$$\text{H(D)} = \sum_{i=1}^{s} p_i \log_2 \left(\frac{1}{p_i}\right)$$

$$\therefore \text{H(D)} = \frac{5}{10} \log_2 \frac{10}{5} + \frac{5}{10} \log_2 \frac{10}{5} = 0.5 + 0.5 = 1$$

(ii) Choosing Age as the Splitting Attribute.

Age	C1 Profit = Down	C2 Profit = Up	Entropy H
Old	3	0	0
Mid	2	2	1
New	0	3	0

$$\therefore \text{H (Age)} = \frac{3}{10} \times \text{H(Old)} + \frac{4}{10} \times \text{H (Mild)} + \frac{3}{10} \times \text{H(New)}$$

$$= \frac{3}{10} \times 0 + \frac{4}{10} \times 1 + \frac{3}{10} \times 0 = 0.4$$

Gain (Age) = H(D) – H(Age) = 1 – 0.4 = 0.6

(iii) Choosing Competition as the Splitting Attribute.

Competition	C1 Profit = Down	C2 Profit = Up	Entropy H
Yes	3	1	0.8113
No	2	4	0.9183

$$\therefore \text{H (Competition)} = \frac{4}{10} \times \text{H(Yes)} + \frac{6}{10} \times \text{H (No)}$$

$$= \frac{4}{10} \times 0.8113 + \frac{6}{10} \times 0.9183 = 0.8755$$

Gain(Competition) = H(D) – H(Competition)

= 1 – 0.8755 = 0.1245

(iv) Choosing Type as the Splitting Attribute.

Type	C1 Profit = Down	C2 Profit = Up	Entropy H
Software	3	3	1
Hardware	2	2	1

$$\text{H (Type)} = \frac{6}{10} \times \text{H(Software)} + \frac{4}{10} \times \text{H (Hardware)}$$

$$= \frac{6}{10} \times 1 + \frac{4}{10} \times 1 = 1$$

Gain(Type) = H(D) – H(Type) = 1 – 1 = 0

Summary

Gain(Age) = 0.6

Gain(Competition) = 0.1245

Gain (Type) = 0

Age attribute has the highest gain; therefore, it is used as the decision attribute in the root node.

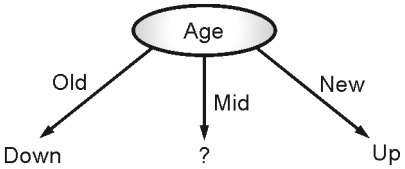
Since Age has three possible values, the root node has three branches (Old, Mid, New).

From dataset we find that,

IF Age = Old THEN Profit = Down

IF Age = Mid THEN Profit = Down OR Profit = Up

If Age = New THEN Profit = Up



(1C8)**Fig. P.3.3.2(a)**

Now, consider Age = Mid and count the number of tuples from the original dataset D. Let us denote it as D1.

Age	Competition	Type	Profit
Mid	Yes	Software	Down
Mid	Yes	Hardware	Down
Mid	No	Hardware	Up
Mid	No	Software	Up

Let the class label attributes be as follows:

C1 = Profit = Down = 2 Samples

C2 = Profit = Up = 2 Samples

Therefore, P(C1) = 2/4 and P(C2) = 2/4

(i) Entropy before split for the given database D1:

$$\text{H(D1)} = \sum_{i=1}^{s} p_i \log_2 \left(\frac{1}{p_i}\right)$$

$$\therefore \text{H(D1)} = \frac{2}{4}\log_z \frac{4}{2} + \frac{2}{4}\log_z \frac{4}{2} = 0.5 + 0.5 = 1$$

(ii) Choosing Competition as the Splitting Attribute.

Competition	C1 Profit = Down	C2 Profit = Up	Entropy H
Yes	2	0	0
No	0	2	0

$$\therefore \text{H (Competition)} = \frac{2}{4} \times \text{H(Yes)} + \frac{2}{4} \times \text{H (No)}$$

$$= \frac{2}{4} \times 0 + \frac{2}{4} \times 0 = 0$$

Gain(Competition) = H(D1) – H(Competition) = 1 – 0 = 1

(iii) Choosing Type as the Splitting Attribute.

Type	C1 Profit = Down	C2 Profit = Up	Entropy H
Software	1	1	1
Hardware	1	1	1

$$\therefore \text{H (Type)} = \frac{2}{4} \times \text{H(Software)} + \frac{2}{4} \times \text{H (Hardware)}$$

$$= \frac{2}{4} \times 1 + \frac{2}{4} \times 1 = 1$$

Gain(Competition) = H(D1) – H(Competition) = 1 – 0 = 1

Gain(Type) = H(D) – H(Type) = 1 – 1 = 0

Summary:

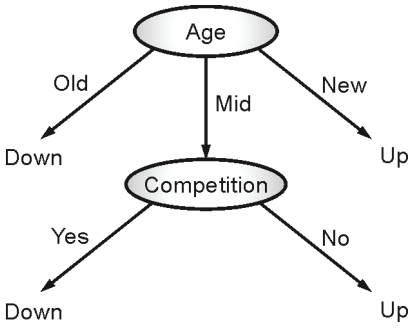
Gain(Competition|D1) = 1

Gain (Type|D1) = 0

Competition attribute has the highest gain; therefore, it is placed below Age = "Mid"

Since, Competition has two possible values, the Competition node has two branches (Yes, No).

From dataset D1, we find that when Competition = Yes, Profit = Down and when Competition = No, Profit = Up.



(1C9)**Fig. P.3.3.2(a)**

The decision tree can also be expressed in rule format as:

IF Age = Old THEN Profit = Down

IF Age = Mid AND Competition = Yes THEN Profit = Down

IF Age = Mid AND Competition = No THEN Profit = Up

If Age = New THEN Profit = Up

UEx. 3.3.3 **MU - June 2021**

Using the following training data set, create classification model using decision tree and draw the final tree.

Tid	Income	Age	Own House
1	Very High	Young	Yes
2	High	Medium	Yes
3	Low	Young	Rented
4	High	Medium	Yes
5	Very High	Medium	Yes
6	Medium	Young	Yes
7	High	Old	Yes
8	Medium	Medium	Rented
9	Low	Medium	Rented
10	Low	Old	Rented
11	High	Young	Yes
12	Medium	Old	Rented

☑ **Soln. :**

Let the class label attributes be as follows:

C1 = Own House = Yes = 7 Samples

C2 = Own House = Rented = 5 Samples

Therefore, P(C1) = 7/12 and P(C2) = 5/12

(i) Entropy before split for the given database D:

$$H(D) = \sum_{i=1}^{s} p_i \log_2\left(\frac{1}{p_i}\right)$$

$$\therefore H(D) = \frac{7}{12}\log_2\frac{12}{7} + \frac{5}{12}\log_2\frac{12}{5} = 0.454 + 0.526$$

$$= 0.980$$

(ii) Choosing Income as the Splitting Attribute

Income	C1 Own House = Yes	C2 Own House = Rented	Entropy H
Very High	2	0	0
High	4	0	0
Medium	1	2	0.918
Low	0	3	0

$$\therefore H(\text{Income}) = \frac{2}{12}\times H(\text{Very High}) + \frac{4}{12}\times H(\text{High}) + \frac{3}{12}\times H(\text{Medium}) + \frac{3}{12}\times H(\text{Low})$$

$$= \frac{2}{12}\times 0 + \frac{4}{12}\times 0 + \frac{3}{12}\times 0.918 + \frac{3}{12}\times 0$$

$$= 0.229$$

$$\text{Gain(Income)} = H(D) - H(\text{Income})$$

$$= 0.980 - 0.229 = 0.751$$

(iiii) Choosing Age as the Splitting Attribute

Age	C1 Own House = Yes	C2 Own House = Rented	Entropy H
Young	3	1	0.811
Medium	3	2	0.971
Old	1	2	0.918

$$\therefore H(\text{Age}) = \frac{4}{12}\times H(\text{Young}) + \frac{5}{12}\times H(\text{Medium}) + \frac{3}{12}\times H(\text{Old})$$

$$= \frac{4}{12}\times 0.811 + \frac{5}{12}\times 0.971 + \frac{3}{12}\times 0.918$$

$$= 0.904$$

Gain(Age) = H(D) – H(Age) = 0.980 – 0.904 = 0.076

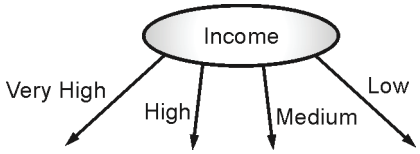
Summary:

Gain(Income) = 0.751

Gain(Age) = 0.076

Income attribute has the highest gain; therefore, it is used as the decision attribute in the root node.

Since Income has four possible values, the root node has four branches (Very High, High, Medium, Low).

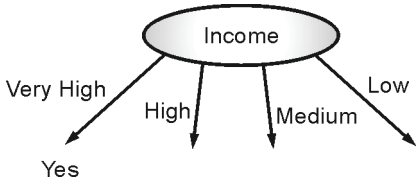


(1C10) **Fig. P.3.3.3(a)**

Now, consider Income = Very High and count the number of tuples from the original dataset D. Let us denote it as D1.

Tid	Income	Age	Own House
1	Very High	Young	Yes
5	Very High	Medium	Yes

Since both the tuples have class label Own House = Yes, we directly give "Yes" as a class below Income = Very High.

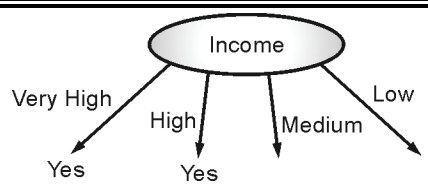


(1C11) **Fig. P.3.3.3(b)**

Now, consider Income = High and count the number of tuples from the original dataset D. Let us denote it as D2.

Tid	Income	Age	Own House
2	High	Medium	Yes
4	High	Medium	Yes
7	High	Old	Yes
11	High	Young	Yes

Since all the four tuples have class label Own House = Yes, we directly give "Yes" as a class below Income = High.

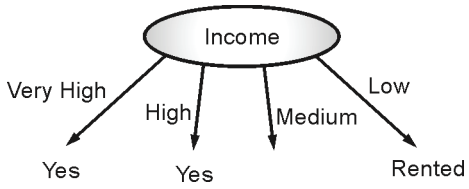


(1C12) **Fig. P.3.3.3(c)**

Now, consider Income = Low and count the number of tuples from the original dataset D. Let us denote it as D3.

Tid	Income	Age	Own House
3	Low	Young	Rented
9	Low	Medium	Rented
10	Low	Old	Rented

Since all the three tuples have class label Own House = Rented, we directly give "Rented" as a class below Income = Low.



(1C13) **Fig. P.3.3.3(d)**

Now, consider Income = Medium and count the number of tuples from the original dataset D. Let us denote it as D4.

Tid	Income	Age	Own House
6	Medium	Young	Yes
8	Medium	Medium	Rented
12	Medium	Old	Rented

We find that,

IF Income = Medium AND Age

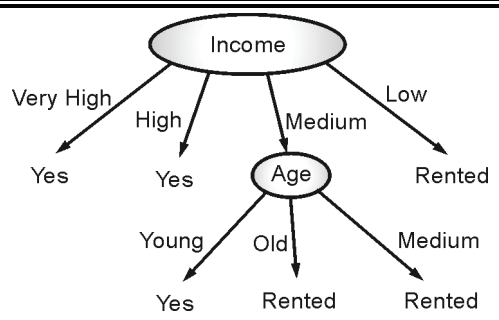
= Young THEN Own House = Yes

IF Income = Medium AND Age

= Medium THEN Own House = Rented

IF Income = Medium AND Age

= Old THEN Own House = Rented



(1C14) **Fig. P.3.3.3(e)**

The decision tree can also be expressed in rule format as:

IF Income = Very High THEN Own House = Yes

IF Income = High THEN Own House = Yes

IF Income = Low THEN Own House = Rented

IF Income = Medium AND Age

= Young THEN Own House = Yes

IF Income = Medium AND Age

= Medium THEN Own House = Rented

IF Income = Medium AND Age

= Old THEN Own House = Rented

UEx. 3.3.4 **MU - May 2019**

The following table consists of training data from an employee database. The data have been generalized. For example, "31 ... 35" for age represents the age range of 31 to 35. For a given row entry, count represents the number of data tuples having the values for department, status, age, and salary given in that row.

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

Let status be the class label attribute.

(a) How would you modify the basic decision tree algorithm to take into consideration the count of each generalized data tuple (i.e., of each row entry)?

(b) Use your algorithm to construct a decision tree from the given data.

☑ **Soln. :**

(a) The basic decision tree algorithm should be modified as follows to take into consideration the count of each generalized data tuple.

- The count of each tuple must be integrated into the calculation of the attribute selection measure (such as information gain).
- Take the count into consideration to determine the most common class among the tuples.

(b) Use **ID3 algorithm** to construct a decision tree from the given data.

Let the class label attributes be as follows:

C1 = status = junior = 113 Samples

C2 = status = senior = 52 Samples

Therefore, P(C1) = 113/165 and P(C2) = 52/165

(i) Entropy before split for the given database D:

$$H(D) = \sum_{i=1}^{s} p_i \log_2\left(\frac{1}{p_i}\right)$$

$$\therefore H(D) = \frac{113}{165}\log_2\frac{165}{113} + \frac{52}{165}\log_2\frac{165}{52}$$

$$= 0.3740 + 0.5250 = 0.899$$

(ii) Choosing department at the Splitting Attribute.

Department	C1 Status = junior	C2 Status = senior	Entropy H
Sales	80	30	0.8454
Systems	23	8	0.8238
Marketing	4	10	0.8631
Secretary	6	4	0.9709

$$\therefore H(\text{department}) = \frac{110}{165} \times H(\text{Sales}) + \frac{31}{165} \times H(\text{systems}) + \frac{14}{165} \times H(\text{marketing}) + \frac{10}{165} \times H(\text{secretary})$$

$$= \frac{110}{165} \times 0.8454 + \frac{31}{165} \times 0.8238 + \frac{14}{165} \times 0.8631 + \frac{10}{165} \times 0.9709 = 0.8504$$

Gain(department) = H(D) – H(department)

= 0.899 – 0.8504 = 0.0486

(ii) Choosing age as the Splitting Attribute.

age	C1 Status=junior	C2 Status=senior	Entropy H
21…25	20	0	0
26…30	49	0	0
31…35	44	35	0.9906
36…40	0	10	0
41…45	0	3	0
46…50	0	4	0

$$\therefore H(age) = \frac{20}{165} \times H(21....25) + \frac{49}{165} \times H(26.....30) + \frac{79}{165} \times H(31.....35) + \frac{10}{165} \times H(36.....40) + \frac{3}{165} \times H(41.....45) + \frac{4}{165} \times H(46....50)$$

$$= \frac{20}{165} \times 0 + \frac{49}{165} \times 0 + \frac{79}{165} \times 0.9906 + \frac{10}{165} \times 0 + \frac{3}{165} \times 0 + \frac{4}{165} \times 0 = 0.4743$$

Gain(age) = H(D) – H(age)

= 0.899 – 0.4743 = 0.4247

(ii) Choosing salary as the Splitting Attribute.

Salary	C1 Status=junior	C2 Status=senior	Entropy H
26K…30K	46	0	0
31K…35K	40	0	0
36K…40K	0	4	0
41K…45K	4	0	0
46K…50K	23	40	0.9468
66K…70K	0	8	0

$$\therefore H(salary) = \frac{46}{165} \times H(26K....30K) + \frac{40}{165} \times H(31K.....35K) + \frac{4}{165} \times H(36K.....40K) + \frac{4}{165} \times H(41K.....45K) + \frac{63}{165} \times H(46K.....50K) + \frac{8}{165} \times H(66K....70K)$$

$$= \frac{46}{165} \times 0 + \frac{40}{165} \times 0 + \frac{4}{165} \times 0 + \frac{4}{165} \times 0 + \frac{63}{165} \times 0.9468 + \frac{8}{165} \times 0 = 0.3615$$

Gain (salary) = H(D) – H(salary)

= 0.899 – 0.3615 = 0.5375

Summary:

Gain(department) = 0.0486

Gain(age) = 0.4247

Gain(salary) = 0.5375

salary attribute has the highest gain; therefore, it is used as the decision attribute in the root node.

Since salary has six possible values, the root node has six branches (26K…30K, 31K…35K, 36K…40K, 41K…45K, 46K…50K, 66K…70K).

From the dataset D, we find that

IF salary = 26K…30K THEN status = junior

IF salary = 30K…35K THEN status = junior

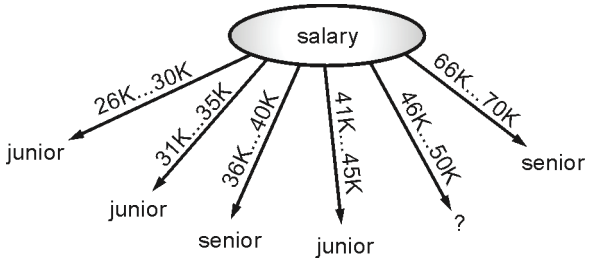
IF salary = 36K…40K THEN status = senior

IF salary = 41K…45K THEN status = junior

IF salary = 46K…50K THEN status = junior OR status = senior

IF salary = 66K…70K THEN status = senior

Only one branch i.e. 46K…50K is not giving a unique class label attribute.



(1C15)**Fig. P. 3.3.4(a)**

Now, consider salary = 46K…50K and count the number of tuples from the original dataset D. Let us denote it as D1.

department	status	Age	Salary	count
Sales	senior	31…35	46K…50K	30
Systems	junior	21…25	46K…50K	20
Systems	junior	26…30	46K…50K	3
marketing	senior	36…40	46K…50K	10

Let the class label attributes be as follows:

C1 = status = junior|salary = 46K…50K = 23 Samples

C2 = status = senior|salary

= 46K…50K = 40 Samples

Therefore, P(C1) = 23/63 and P(C2) = 40/63

(i) Entropy before split for the given database D1:

$$H(D1) = \sum_{i=1}^{s} p_i \log_2\left(\frac{1}{p_i}\right)$$

$$\therefore H(D1) = \frac{23}{63}\log_2\frac{63}{23} + \frac{40}{63}\log_2\frac{63}{40} = 0.9468$$

(ii) Choosing department as the Splitting Attribute.

department	C1 Status=junior	C2 Status=senior	Entropy H
Sales	0	30	0
Systems	23	0	0
Marketing	0	10	0
Secretary	0	0	0

$$\therefore H(\text{department}) = \frac{30}{63}\times H(\text{sales}) + \frac{23}{63}\times H(\text{systems}) + \frac{10}{63} \times H(\text{marketing}) + \frac{0}{63}\times H(\text{secretary})$$

$$= \frac{30}{63}\times 0 + \frac{23}{63}\times 0 + \frac{10}{63}\times 0 + \frac{0}{63}\times 0 = 0$$

Gain(department) = H(D1) – H(department)

= 0.9468 – 0 = 0.9468

(iii) Choosing age as the Splitting Attribute.

age	C1 Status=junior	C2 Status=senior	Entropy H
21…25	20	0	0
26…30	3	0	0
31…35	0	30	0
36…40	0	10	0
41…45	0	0	0
46…50	0	0	0

$$\therefore H(\text{age}) = \frac{20}{63}\times H(21...25) + \frac{3}{63}\times H(26...30) + \frac{30}{63} \times H(31....35) + \frac{10}{63}\times H(36.....40) + \frac{0}{63} \times H(41.....45) + \frac{0}{63}\times H(46.....50)$$

$$= \frac{20}{63}\times 0 + \frac{3}{63}\times 0 + \frac{30}{63}\times 0 + \frac{10}{63}\times 0 + \frac{0}{63}\times 0 + \frac{0}{63}\times 0 = 0$$

Gain(age) = H(D1) – H(age) = 0.9468 – 0 = 0.9468

Summary

Gain(department|D1) = 0.9468

Gain(age|D1) = 0.9468

Both the attributes have the same gain; therefore, we choose one of the attribute arbitrarily and place it below "salary = 46K…50K".

We choose **department** as the attribute below "salary = 46K…50K".

From dataset D1, we find that

IF salary = 46K…50K AND department

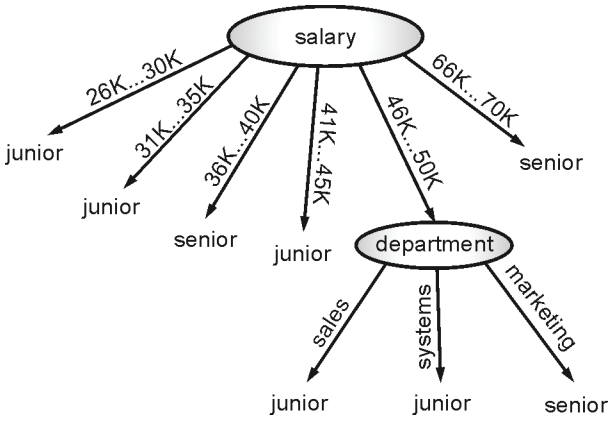
= sales THEN status = junior.

IF salary = 46K…50K AND department

= systems THEN status = junior.

IF salary = 46K…50K AND department

= marketing THEN status = senior.



(1C16)**Fig. 3.3.4(b)**

The decision tree can also be expressed in rule format as:

IF salary = 26K…30K THEN status = junior

IF salary = 30K…35K THEN status = junior

IF salary = 36K…40K THEN status = senior

IF salary = 41K…45K THEN status = junior

IF salary = 46K…50K AND department = sales THEN status = junior

IF salary = 46K…50K AND department = systems THEN status = junior

IF salary = 46K…50K AND department = marketing THEN status = senior

IF salary = 66K…70K THEN status = senior

☞ Advantages of ID3 Algorithm

1. Evident prediction rules are constructed form the training data.
2. It builds the short tree.
3. It searches entire dataset to create the tree.
4. It searches complete hypothesis space to predict unlabeled instances.

5. It is less sensitive toward errors of individual training examples because of statistical properties of instances are utilized

☞ Disadvantages of ID3 Algorithm

1. Over-fitting of the data may happen while classification.
2. It does not perform backtracking while searching.
3. It may converge in locally optimal solution.
4. Computational complexity may be very high for the continuous data.

⏭ 3.4 NAÏVE BAYESIAN CLASSIFICATION

- Bayesian classification is based on Bayes' Theorem. Bayesian classifiers are the statistical classifiers.
- Bayesian classifiers can predict class membership probabilities such as the probability that a given tuple belongs to a particular class.

3.4.1 Baye's Theorem

- Bayes's theorem is expressed mathematically by the following equation that is given below.

$$P(X|Y) = \frac{P(Y|X)\,P(X)}{P(Y)}$$

Where X and Y are the events and $P(Y) \neq 0$

- P(X|Y) is a **conditional probability** that describes the occurrence of event X given that Y is true.
- P(Y|X) is a **conditional probability** that describes the occurrence of event Y given that X is true.
- P(X) and P(Y) are the probabilities of observing X and Y independently of each other. This is known as the **marginal probability**.

3.4.2 Bayesian Interpretation

- In the Bayesian interpretation, probability determines a "degree of belief."
- Bayes theorem connects the degree of belief in a hypothesis before and after accounting for evidence.
- For example, lets us consider an example of the coin. If we toss a coin, then we get either head or tail, and the percent of occurrence of either head and tail is 50%. If the coin is flipped numbers of times, and the outcomes are observed, the degree of belief may rise, fall, or remain the same depending on the outcomes.
- For proposition X and evidence Y,
  - (a) P(X), the priori probability, is the primary degree of belief in X
  - (b) P(X|Y), the posterior probability, is the degree of belief having accounted for Y.
  - (c) The quotient $\frac{P(Y|X)}{P(Y)}$ represents the support Y provides for X.
- Bayes theorem can be derived from the conditional probability :

$$P(X|Y) = \frac{P(X \cap Y)}{P(Y)}, \text{ if } P(Y) \neq 0$$

$$P(Y|X) = \frac{P(Y \cap X)}{P(X)}, \text{ if } P(X) \neq 0$$

where $P(X \cap Y)$ is the joint probability of both X and Y being true, because

$$P(X \cap Y) = P(Y \cap X)$$

$$\text{Or, } P(X \cap Y) = P(X|Y)\,P(Y) = P(Y|X)\,P(X)$$

$$\text{Or, } P(X|Y) = \frac{P(Y|X)\,P(X)}{P(Y)}, \text{ if } P(Y) \neq 0$$

Ex. 3.4.1 : Apply statistical based algorithm to obtain the actual probabilities of each event to classify the new tuple as "Buys_Computer = Yes". Use the following data:

Classify X = (Age = "<=30", Income = "Medium", Student = "Yes", Credit_rating = "Fair").

Id	Age	Income	Student	Credit_rating	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

☑ **Soln. :**

Class label attribute is Buys_Computer.

C_1: Buys_Computer = Yes = 9 samples

C_2: Buys_Computer = No = 5 Samples

$\therefore P(C_1) = \frac{9}{14}$ and $P(C_2) = \frac{5}{14}$

Let event X_1 be Age = "<=30",

event X_2 be Income = "Medium",

event X_3 be Student = "Yes", and

event X_4 be Credit_rating = "Fair"

To compute: $P(X|C_1)$

From Naïve Bayesian Classification

$$P(X|C_i) = \prod_{k=1}^{m} P(X_k|C_i)$$

$$P(X|C_1) = P(X_1|C_1)\, P(X_2|C_1)\, P(X_3|C_1)\, P(X_4|C_1)$$

$$P(X_1|C_1) = P\left(\frac{\text{Age} <= 30}{\text{Buys\_Computer} = \text{Yes}}\right) = \frac{2}{9}$$

$$P(X_2|C_1) = P\left(\frac{\text{Income} = \text{Medium}}{\text{Buys\_Computer} = \text{Yes}}\right) = \frac{4}{9}$$

$$P(X_3|C_1) = P\left(\frac{\text{Student} = \text{Yes}}{\text{Buys\_Computer} = \text{Yes}}\right) = \frac{6}{9}$$

$$P(X_4|C_1) = P\left(\frac{\text{Credit\_rating} = \text{Yes}}{\text{Buys\_Computer} = \text{Yes}}\right) = \frac{6}{9}$$

From Naïve Bayesian Classification

$$P(X|C_1) = \frac{2}{9} \times \frac{4}{9} \times \frac{6}{9} \times \frac{6}{9} = 0.044$$

$$P(X|C_1)\, P(C_1) = 0.044 \times \frac{9}{14} = 0.028 \quad \text{...(A)}$$

To compute: $P(X|C_2)$

From Naive Bayesian Classification

$$P(X|C_i) = \prod_{k=1}^{m} P(X_k|C_i)$$

$$P(X|C_2) = P(X_1|C_2)\, P(X_2|C_2)\, P(X_3|C_2)\, P(X_4|C_2)$$

$$P(X_1|C_2) = P\left(\frac{\text{Age} <= 30}{\text{Buys\_Computer} = \text{No}}\right) = \frac{3}{5}$$

$$P(X_2|C_2) = P\left(\frac{\text{Income} = \text{Medium}}{\text{Buys\_Computer} = \text{No}}\right) = \frac{2}{5}$$

$$P(X_3|C_2) = P\left(\frac{\text{Student} = \text{Yes}}{\text{Buys\_Computer} = \text{No}}\right) = \frac{1}{5}$$

$$P(X_4|C_2) = P\left(\frac{\text{Credit\_rating} = \text{Yes}}{\text{Buys\_Computer} = \text{No}}\right) = \frac{2}{5}$$

From Naïve Bayesian Classification

$$P(X|C_2) = \frac{3}{5} \times \frac{2}{5} \times \frac{1}{5} \times \frac{2}{5} = 0.019$$

$$P(X|C_2)\, P(C_2) = 0.019 \times \frac{5}{14} = 0.007 \quad \text{...(B)}$$

Naive Bayesian classification will assign a sample X to class C_i if and only if

$P(C_i|X) > P(C_j|X)$

i.e. $\frac{P(X|C_i)P(C_i)}{P(X)} > \frac{P(X|C_j)P(C_j)}{P(X)}$

$P(X)$ is constant for both classes C_i and C_j.

$\therefore P(X|C_i)\, P(C_i) > P(X|C_j)\, P(C_j)$ ....(C)

From (A), (B) and (C),

$P(X|C_1)\, P(C_1) > P(X|C_2)\, P(C_2)$

$\therefore$ We conclude that the unknown sample

X = (Age = "<=30", Income = "Medium", Student = "Yes", Credit_rating = "Fair") belongs to class **C_1 = Buys_Computer = Yes.**

Ex. 3.4.2 : Apply statistical based algorithm to obtain the actual probabilities of each event to classify

X = (Dept = "Systems", Status = "Junior", Age = "26 -30"). Use the following table:

Dept	Status	Age	Salary	Count
Sales	Senior	31-35	46k – 50k	30
Sales	Junior	26-30	26k – 30k	40
Sales	Junior	26-30	31k – 35k	40
Systems	Junior	21-25	46k – 50k	20
Systems	Junior	31-35	66k – 70k	5
Systems	Junior	26-30	46k – 50k	5
Systems	Senior	41-45	66k – 70k	5

☑ **Soln. :**

Class label attribute is Salary.

C_1 = Salary between 46k – 50k = 55 Samples

C_2 = Salary between 26k – 30k = 40 Samples

C_3 = Salary between 31k – 40k = 40 Samples

C_4 = Salary between 66k – 70k = 10 Samples

$\therefore P(C_1) = \frac{55}{145}$, $P(C_2) = \frac{40}{145}$, $P(C_3) = \frac{40}{145}$ and $P(C_4) = \frac{40}{145}$

Let event X_1 be Dept = "Systems",

event X_2 be Status = "Junior", and

event X_3 be Age = "26 -30"

To compute: $P(X|C_1)$

From Naïve Bayesian Classification

$$P(X|C_i) = \prod_{k=1}^{m} P(X_k|C_i)$$

$$P(X|C_1) = P(X_1|C_1)\, P(X_2|C_1)\, P(X_3|C_1)\, P(X_4|C_1)$$

$$P(X_1|C_1) = P\left(\frac{\text{Dept = Systems}}{\text{Salary 46 k – 50 k}}\right) = \frac{25}{55}$$

$$P(X_2|C_1) = P\left(\frac{\text{Status = Junior}}{\text{Salary 46 k – 50k}}\right) = \frac{25}{55}$$

$$P(X_3|C_1) = P\left(\frac{\text{Age = 26 – 30}}{\text{Salary 46 k – 50k}}\right) = \frac{5}{55}$$

From Naïve Bayesian Classification

$$P(X|C_1) = \frac{25}{55} \times \frac{25}{55} \times \frac{5}{55} = 0.019$$

$$P(X|C_1)P(C_1) = 0.019 \times \frac{55}{145} = 0.007 \quad \text{...(A)}$$

To compute: $P(X|C_2)$

From Naïve Bayesian Classification

$$P(X|C_i) = \prod_{k=1}^{m} P(X_k|C_i)$$

$$P(X|C_2) = P(X_1|C_2)\, P(X_2|C_2)\, P(X_3|C_2)\, P(X_4|C_2)$$

$$P(X_1|C_2) = P\left(\frac{\text{Dept = Systems}}{\text{Salary 26k – 30 k}}\right) = \frac{0}{40}$$

$$P(X_2|C_2) = P\left(\frac{\text{Status = Junior}}{\text{Salary 26 k – 30k}}\right) = \frac{40}{40}$$

$$P(X_3|C_2) = P\left(\frac{\text{Age = 26 – 30}}{\text{Salary 26 k – 30k}}\right) = \frac{40}{40}$$

From Naïve Bayesian Classification

$$P(X|C_2) = \frac{0}{40} \times \frac{40}{40} \times \frac{40}{40} = 0$$

$$P(X|C_2)\, P(C_2) = 0 \times \frac{40}{145} = 0 \quad \text{...(B)}$$

To compute: $P(X|C_3)$

From Naïve Bayesian Classification

$$P(X|C_i) = \prod_{k=1}^{m} P(X_k|C_i)$$

$$P(X|C_3) = P(X_1|C_3)\, P(X_2|C_3)\, P(X_3|C_3)\, P(X_4|C_3)$$

$$P(X_1|C_3) = P\left(\frac{\text{Dept = Systems}}{\text{Salary 31k – 35 k}}\right) = \frac{0}{40}$$

$$P(X_2|C_3) = P\left(\frac{\text{Status = Junior}}{\text{Salary 31 k – 35k}}\right) = \frac{40}{40}$$

$$P(X_3|C_3) = P\left(\frac{\text{Age = 26 – 30}}{\text{Salary 31 k – 35k}}\right) = \frac{40}{40}$$

From Naïve Bayesian Classification

$$P(X|C_3) = \frac{0}{40} \times \frac{40}{40} \times \frac{40}{40} = 0$$

$$P(X|C_3)\, P(C_3) = 0 \times \frac{40}{145} = 0 \quad \text{...(C)}$$

To compute: $P(X|C_4)$

From Naïve Bayesian Classification

$$P(X|C_i) = \prod_{k=1}^{m} P(X_k|C_i)$$

$$P(X|C_4) = P(X_1|C_4)\, P(X_2|C_4)\, P(X_3|C_4)\, P(X_4|C_4)$$

$$P(X_1|C_4) = P\left(\frac{\text{Dept = Systems}}{\text{Salary 66k – 70 k}}\right) = \frac{10}{10}$$

$$P(X_2|C_4) = P\left(\frac{\text{Status = Junior}}{\text{Salary 66 k – 70k}}\right) = \frac{5}{10}$$

$$P(X_3|C_4) = P\left(\frac{\text{Age = 26 – 30}}{\text{Salary 66 k – 70k}}\right) = \frac{0}{10}$$

From Naïve Bayesian Classification

$$P(X|C_4) = \frac{10}{10} \times \frac{5}{10} \times \frac{0}{10} = 0$$

$$P(X|C_4)\, P(C_4) = 0 \times \frac{10}{145} = 0 \quad \text{...(D)}$$

Naïve Bayesian classification will assign a sample X to class C_i if and only if,

$$P(C_i|X) > P(C_j|X)$$

$$\text{i.e. } \frac{P(X|C_i)\, P(C_i)}{P(X)} > \frac{P(X|C_j)\, P(C_j)}{P(X)}$$

P(X) is constant for both classes C_i and C_j.

$$\therefore P(X|C_i)\, P(C_i) > P(X|C_j)\, P(C_j) \quad \text{...(E)}$$

From (A), (B), (C), (D) and (E)

$$P(X|C_1)\, P(C_1) > P(X|C_2)\, P(C_2) \geq P(X|C_3)\, P(C_3) \geq P(X|C_4)\, P(C_4)$$

$\therefore$ We conclude that the unknown sample

X = (Dept = "Systems", Status = "Junior", Age = "26 -30") belongs to class **C_1 = Salary between 46k – 50k**.

Ex. 3.4.3 : Given the training data for height classification, classify the tuple, t = <Rohit, M, 1.95> using Bayesian Classification.

Name	Gender	Height	Output
Kiran	F	1.6m	Short
Jatin	M	2m	Tall
Madhuri	F	1.09m	Medium
Manisha	F	1.88m	Medium
Shilpa	F	1.7m	Short
Bobby	M	1.85m	Medium
Kavita	F	1.6m	Short
Dinesh	M	1.7m	Short
Rahul	M	2.2m	Tall
Shree	M	2.1m	Tall
Divya	F	1.8m	Medium
Tushar	M	1.95m	Medium
Kim	F	1.9m	Medium
Aarti	F	1.8m	Medium
Rajashree	F	1.75m	Medium

☑ **Soln. :**

From the above table, it is clear that there are 4 tuples classified as Short, 8 tuples classified as Medium and 3 tuples classified as Tall.

We divide the height attribute into six ranges as given below:

(0,1.6], (1.6,1.7], (1.7,1.8], (1.8,1.9], (1.9,2.0] and (2.0, ∞]

Note : (a,b] indicates value greater than a and less than or equal to b.

Probabilities associated with attributes is given in table below.

Attribute	Value	Count			Probabilities		
		Short	Medium	Tall	Short	Medium	Tall
Gender	M	1	2	3	1/4	2/8	3/3
	F	3	6	0	3/4	6/8	0/3
	Total	**4**	**8**	**3**			
Height	(0.1.6]	2	0	0	2/4	0	0
	(1.6,1.7]	2	0	0	2/4	0	0
	(1.7,1.8]	0	3	0	0	3/8	0
	(1.8,1.9]	0	4	0	0	4/8	0
	(1.9,2.0]	0	1	1	0	1/8	1/3
	(2.0, ∞]	0	0	2	0	0	2/3
	Total	**4**	**8**	**3**			

From the given training data, we estimate

$$P(\text{Short}) = 4/15$$

$$P(\text{Medium}) = 8/15$$

$$P(\text{Tall}) = 3/15$$

The unseen tuple is t = <Name = Rohit, Gender = M, Height = 1.95 >

$$P(t|\text{Short}) \times P(\text{Short}) = P(M|\text{Short}) \times P(1.9.2.0|\text{Short}) \times P(\text{Short})$$

$$= \frac{1}{4} \times 0 \times \frac{4}{15} = 0$$

$$P(t|\text{Medium}) \times P(\text{Medium}) = P(M|\text{Medium}) \times P((1.9.2.0)|\text{Medium}) \times P(\text{Medium})$$

$$= \frac{2}{8} \times \frac{1}{8} \times \frac{8}{15} = 0.0167$$

$$P(t|\text{tall}) \times P(\text{tall}) = P(M|\text{tall}) \times P((1.9.2.0)|\text{tall}) \times P(\text{tall})$$

$$= \frac{3}{3} \times \frac{1}{3} \times \frac{3}{15} = 0.067$$

Based on the above probabilities, we classify the new tuple as **Tall** because it has the highest probability.

Ex. 3.4.4 : Given the training data for credit transaction. Classify a new transaction with (Income = Medium and credit = Good) using Naïve Bayes classification.

Transaction	Income	Credit	Decision
1	Very high	Excellent	AUTHORIZE
2	High	Good	AUTHORIZE
3	Medium	Excellent	AUTHORIZE
4	High	Good	AUTHORIZE
5	Very high	Good	AUTHORIZE
6	Medium	Excellent	AUTHORIZE
7	High	Bad	REQUEST ID
8	Medium	Bad	REQUEST ID
9	High	Bad	REJECT
10	Low	Bad	CALL POLICE

☑ **Soln. :**

From the above table, it is clear that there are 6 tuples classified as AUTHORIZE, 2 tuples classified as REQUEST ID, 1 tuple classified as REJECT and 1 tuple classified as CALL POLICE.

From the given training data, we estimate

P(AUTHORIZE) = 6/10 P(REQUEST ID) = 2/10

P(REJECT) = 1/10 P(CALL POLICE) = 1/10

Probabilities associated with attributes is given in table below.

Attribute	Value	Count				Probabilities			
		AUTHORIZE	REQUEST ID	REJECT	CALL POLICE	AUTHORIZE	REQUEST ID	REJECT	CALL POLICE
Income	Very High	2	0	0	0	2/6	0	0	0
	High	2	1	1	0	2/6	1/2	1	0
	Medium	2	1	0	0	2/6	1/2	0	0
	Low	0	0	0	1	0	0	0	1
	Total	6	2	1	1				
Credit	Excellent	3	0	0	0	3/6	0	0	0
	Good	3	0	0	0	3/6	0	0	0
	Bad	0	2	1	1	0	2/2	1	1
	Total	6	2	1	1				

The unknown tuple is t = <Income = Medium, Credit = Good>

$$P(t|AUTHORIZE) \times P(AUTHORIZE) = P(Income = Medium|AUTHORIZE) \times P(Credit = Good|AUTHORIZE) \times P(AUTHORIZE)$$

$$= \frac{2}{6} \times \frac{3}{6} \times \frac{6}{10} = 0.1$$

$$P(t|REQ.ID) \times P(REQ.ID) = P(Income = Medium|REQ.ID) \times P(Credit = Good|REQ.ID) \times P(REQ.ID)$$

$$= \frac{1}{2} \times 0 \times \frac{2}{10} = 0$$

$$P(t|REJECT) \times P(REJECT) = P(Income = Medium| REJECT) \times P(Credit = Good|REJECT) \times P(REJECT)$$

$$= 0 \times 0 \times \frac{1}{10} = 0$$

$$P(t|CALL\ POLICE) \times P(CALL\ POLICE) = P(Income = Medium|CALL\ POLICE) \times P(Credit = Good|CALL\ POLICE) \times P(CALL\ POLICE)$$

$$= 0 \times 0 \times \frac{1}{10} = 0$$

Based on the above probabilities, we classify the new tuple as **AUTHORIZE** because it has the highest probability.

⏭ 3.5 RULE BASED CLASSIFICATION : IF-THEN RULES

- Rule-based classifier makes use of a set of IF-THEN rules for classification.
- We can express a rule in the following form

  IF condition THEN conclusion

- Let us consider a rule R1,

  R1: IF age = youth AND student = yes THEN buy_computer = yes
- The IF part of the rule is called **rule antecedent** or **precondition**.
- The THEN part of the rule is called **rule consequent**.

- The antecedent part the condition consists of one or more attribute tests and these tests are logically ANDed.
- The consequent part consists of class prediction.
- To extract a rule from a decision tree –
  1. One rule is created for each path from the root to the leaf node.
  2. To form a rule antecedent, each splitting criterion is logically ANDed.
  3. The leaf node holds the class prediction, forming the rule consequent.

3.6 ACCURACY AND ERROR MEASURES

- In data mining, classification involves the problem of predicting which category or class a new observation belongs in.
- The derived model (classifier) is based on the analysis of a set of training data where each data is given a class label.
- The trained model (classifier) is then used to predict the class label for new, unseen data.
- To understand classification metrics, one of the most important concepts is the confusion matrix.
- The different classifier evaluation measures are discussed below:

1. Confusion Matrix : It is a useful tool for analyzing how well your classifier can recognize tuples of different classes. It is also called as contingency matrix. Each row in a confusion matrix represents an actual class, while each column represents a predicted class.

The 2×2 confusion matrix is denoted as:

		Predicted Class	
		1	0
Actual Class	1	TP	FN
	0	FP	TN

- TP: It represents the values which are predicted to be true and are actually true.
- TN: It represents the values which are predicted to be false and are actually false.
- FP: It represents the values which are predicted to be true, but are false. Also called Type I error.
- FN: It represents the values which are predicted to be false, but are true. Also called Type II error.

2. Sensitivity : Also called the true positive recognition rate. It is proportion of positive tuples that are correctly identified.

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

3. Specificity : Also called the true negative rate. It is proportion of negative tuples that are correctly identified.

$$\text{Specificity} = \frac{TN}{TN + FP}$$

4. Accuracy : The accuracy of a classifier on a given test set is the percentage of test set tuples that are correctly classified by the classifier. It is also referred to as the overall recognition rate of the classifier.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

5. Precision : It is the measure of exactness. It determines what percentage of tuples labelled as positive are actually positive.

$$\text{Precision} = \frac{TP}{TP + FP}$$

6. Recall : It is the measure of completeness. It determines what percentage of positive tuples are labelled as positive.

$$\text{Recall} = \frac{TP}{TP + FN}$$

7. F-Score : It is the harmonic mean of precision and recall. It gives equal weight to precision and recall. It is also called F-measure or F_1 score.

$$F = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

8. F_β Score: It is the weighted measure of precision and recall. It assigns β times as much weight to recall as to precision. Commonly used F_β measures are F_2 (which weights recall twice as much as precision) and $F_{0.5}$ (which weights precision twice as much as recall).

$$F_\beta = \frac{(1 + \beta^2) \times \text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}}$$

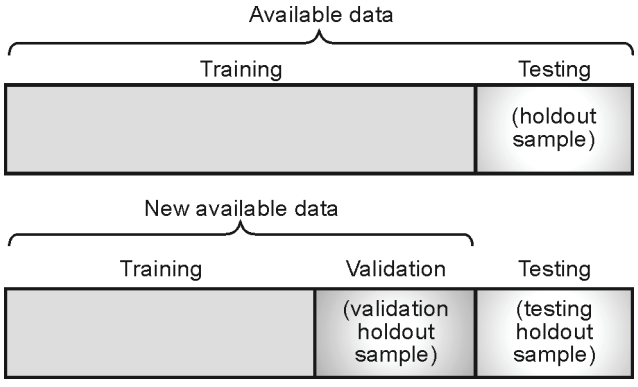
9. Error Rate : It is also called misclassification rate of a classifier and is simply (1 – Accuracy).

3.7 EVALUATING THE ACCURACY OF A CLASSIFIER

Besides the evaluation measure discussed above, other techniques to evaluate the accuracy of a classifier are discussed below.

3.7.1 Holdout

- In this method, the mostly large dataset is *randomly* divided to three subsets.
  1. **Training set** is a subset of the dataset used to build predictive models.
  2. **Validation set** is a subset of the dataset used to assess the performance of model built in the training phase. It provides a test platform for fine tuning model's parameters and selecting the best-performing model. Not all modeling algorithms need a validation set.
  3. **Test set** or unseen examples is a subset of the dataset to assess the likely future performance of a model. If a model fit to the training set much better than it fits the test set, overfitting is probably the cause.
- Typically, two-thirds of the data are allocated to the training set and the remaining one-third is allocated to the test set.

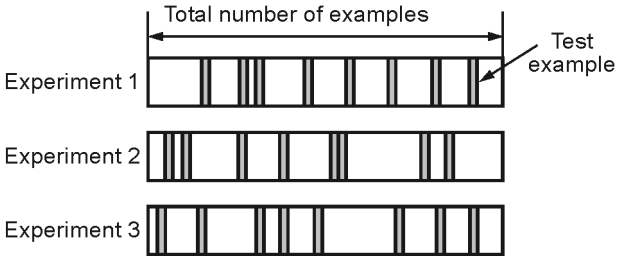


(1C18)**Fig. 3.7.1 : Holdout**

3.7.2 Random Subsampling

- It is a variation of the holdout method. The holdout method is repeated k times.
- It involves randomly splitting the data into a training and a test set.
- The data is trained on the training set and the mean square error (MSE) is obtained from the predictions on the test set.
- This method is not recommended because the MSE would depend on the split. So a new split can give you a new MSE and then you don't know which to trust.
- The overall accuracy is calculated by taking the average of the accuracies obtained from each iteration.

$$E = \frac{1}{k} \sum_{i=1}^{k} E_i$$



(1C19)**Fig. 3.7.2 : Random Subsampling**

3.7.3 Cross Validation

- When only a limited amount of data is available, to achieve an unbiased estimate of the model performance we use k-fold cross-validation.
- In k-fold cross-validation, we divide the data into k subsets of equal size.
- We build models k times, each time leaving out one of the subsets from training and use it as the test set.
- If k equals the sample size, this is called "leave-one-out".

3.7.4 Bootstrapping

- Bootstrapping is a technique used to make estimations from data by taking an average of the estimates from smaller data samples.
- The bootstrap method involves iteratively resampling a dataset with replacement.
- Instead of only estimating our statistic once on the complete data, we can do it many times on a re-sampling (with replacement) of the original sample.
- Repeating this re-sampling multiple times allows us to obtain a vector of estimates.
- We can then compute variance, expected value, empirical distribution, and other relevant statistics of these estimates.

⏭ 3.8 MULTIPLE CHOICE QUESTIONS

Q. 3.1 Which of the following statement is true about the classification?

(a) It is a measure of accuracy

(b) It is a subdivision of a set

(c) It is the task of assigning a classification

(d) It is an aggregation of a set ✓ **Ans. : (b)**

Q. 3.2 Expected amount of information (in bits) needed to assign a class to a randomly drawn object is _____.

(a) Entropy (b) Gain Ratio

(c) Information Gain (d) Gini Index ✓ **Ans. : (a)**

Q. 3.3 In tree prepruning, a tree is pruned by _____.

(a) halting its construction early

(b) halting construction of the entire tree

(c) halting after the tree construction

(d) nothing is required. ✓ **Ans. : (a)**

Q. 3.4 The formula for accuracy is _____.

(a) Accuracy = (TP - TN)/All

(b) Accuracy = (TP + TN)/P

(c) Accuracy = (TP + TN)/All

(d) Accuracy = (TP + TN)/N ✓ **Ans. : (c)**

Q. 3.5 Which of the following is not a method to estimate a classifier's accuracy?

(a) Holdout method (b) Random subsampling

(c) Information Gain (d) Bootstrap ✓ **Ans. : (c)**

Q. 3.6 What is Decision Tree?

(a) Flow-Chart

(b) Structure in which internal node represents test on an attribute, each branch represents outcome of test and each leaf node represents class label

(c) Flow-Chart & Structure in which internal node represents test on an attribute, each branch represents outcome of test and each leaf node represents class label

(d) None of the mentioned ✓ **Ans. : (c)**

Q. 3.7 What is ID3 algorithm used to build?

(a) Association rules (b) Statistical induction

(c) Regression analysis (d) Decision tree

✓ **Ans. : (d)**

Q. 3.8 Where can the Bayes rule be used?

(a) Solving queries

(b) Increasing complexity

(c) Decreasing complexity

(d) Answering probabilistic query ✓ **Ans. : (d)**

Q. 3.9 Given the Confusion matrix, the accuracy is:

Classes	Yes	No
Yes	90	210
No	140	9560

(a) 60% (b) 100%

(c) 96.5% (d) 35% ✓ **Ans. : (c)**

Q. 3.10 Methods used to estimate classifier accuracy are:

(a) Holdout (b) Cross-Validation

(c) Bootstrap (d) All of the above ✓ **Ans. : (d)**

Q. 3.11 What are True positive observations?

(a) the values which are predicted to be false and are actually false.

(b) the values which are predicted to be true and are actually true.

(c) the values which are predicted to be true but are false. Also called Type I error.

(d) the values which are predicted to be false but are true. Also called Type II error. ✓ **Ans. : (b)**

Q. 3.12 Select the measure which is not used to calculate the accuracy.

(a) Lift (b) Precision

(c) Recall (d) F-measure ✓ **Ans. : (a)**

Q. 3.13 _____ is a measure of completeness.

(a) Sensitivity (b) Precision

(c) Recall (d) F-measure ✓ **Ans. : (c)**

Q. 3.14 _____ is a measure of exactness.

(a) Sensitivity (b) Precision

(c) Recall (d) F-measure ✓ **Ans. : (b)**

Q. 3.15 _____ is a true negative rate.

(a) Sensitivity (b) Precision

(c) Recall (d) Specificity ✓ **Ans. : (d)**

Q. 3.16 _____ is a harmonic mean of precision and recall.

(a) Sensitivity (b) Precision

(c) Recall (d) F-measure ✓ **Ans. : (d)**

Q. 3.17 Classification is a form of _____ type of learning.

(a) Supervised (b) Unsupervised

(c) Semi-supervised (d) None of the above

✓ **Ans. : (a)**

Q. 3.18 Select the term which is not required for the classification.

(a) Training data (b) Test data

(c) Clusters (d) Class labels ✓ **Ans. : (c)**

Q. 3.19 In Holdout method, the training set is generally _____ of the total dataset.

(a) 25% (b) 50%

(c) 75% (d) 100% ✓ **Ans. : (c)**

Q. 3.20 In Holdout method, the test set is generally _____ of the total dataset.

(a) 25% (b) 50%

(c) 75% (d) 100% ✓ **Ans. : (a)**

Q. 3.21 _____ refers to the computational cost in generating and using the classifier or predictor.

(a) Speed (b) Robustness

(c) Scalability (d) Interpretability ✓ **Ans. : (a)**

Q. 3.22 _____ refers to the ability of classifier or predictor to make correct predictions from given noisy data.

(a) Speed (b) Robustness

(c) Scalability (d) Interpretability ✓ **Ans. : (b)**

Q. 3.23 _____ refers to the ability to construct the classifier or predictor efficiently; given large amount of data.

(a) Speed (b) Robustness

(c) Scalability (d) Interpretability ✓ **Ans. : (c)**

Q. 3.24 _____ refers to what extent the classifier or predictor understands.

(a) Speed (b) Robustness

(c) Scalability (d) Interpretability ✓ **Ans. : (d)**

Q. 3.25 Attribute with _____ information gain is used as the decision node in decision tree using ID3 algorithm.

(a) Zero (b) Maximum

(c) Minimum (d) Any ✓ **Ans. : (b)**

Descriptive Questions

Q. 1 Briefly outline the major steps of decision tree classification.

Q. 2 Explain the issues in classification. Also discuss the parameters to compare different classification methods.

Q. 3 Explain decision tree induction algorithm.

Q. 4 Why is tree pruning useful in decision tree induction? What is a drawback of using a separate set of tuples to evaluate pruning?Given a decision tree, you have the option of (a) converting the decision tree to rules and then pruning the resulting rules, or (b) pruning the decision tree and then converting the pruned tree to rules. What advantage does (a) have over (b)?

(MU - May 2019, Dec. 2019)

Q. 5 Apply ID3 algorithm on the following training dataset and extract the classification rule from the tree.

Id	Age	Income	Student	Credit_rating	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

Q. 6 The following table consists of training data from an employee database. The data have been generalized. For example, "31 ... 35" for age represents the age range of 31 to 35. For a given row entry, count represents the number of data tuples having the values for department, status, age, and salary given in that row. Let status be the class label attribute.

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

(a) How would you modify the basic decision tree algorithm to take into consideration the count of each generalized data tuple (i.e., of each row entry)?

(b) Use your algorithm to construct a decision tree from the given data. **(MU - May 2019)**

Q. 7 Using the following training data set, create classification model using decision tree and draw the final tree. **(MU - June 2021)**

Tid	Income	Age	Own House
1	Very High	Young	Yes
2	High	Medium	Yes
3	Low	Young	Rented
4	High	Medium	Yes
5	Very High	Medium	Yes
6	Medium	Young	Yes
7	High	Old	Yes
8	Medium	Medium	Rented
9	Low	Medium	Rented
10	Low	Old	Rented
11	High	Young	Yes
12	Medium	Old	Rented

Q. 8 Given the training data for credit transaction. Classify a new transaction with (Outlook=Sunny, Temperature = Cool, Humidity = High and Windy = True) using Naïve Bayes classification.

Outlook	Temperature	Humidity	Windy	Class
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rain	Mild	High	False	Yes
Rain	Cool	Normal	False	Yes
Rain	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rain	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rain	Mild	High	True	No

Q. 9 Discuss the different accuracy and error measures of the classifier. Show that accuracy is a function of sensitivity and specificity.

Q. 10 Explain in detail the methods to evaluate the accuracy of a classifier.

Chapter Ends…

❑❑❑

Note

MODULE 4

CHAPTER 4

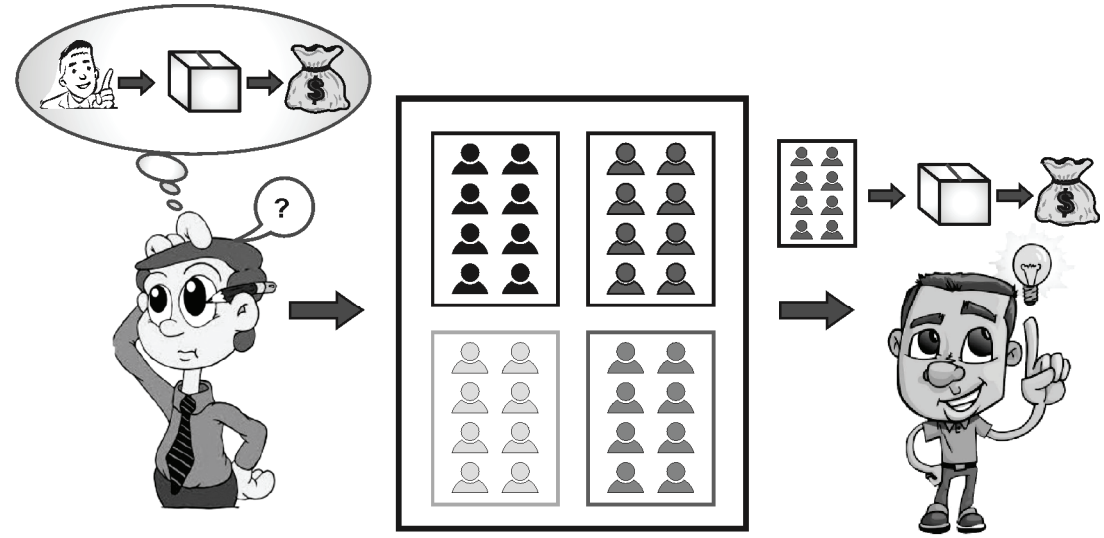
Clustering

University Prescribed Syllabus w.e.f Academic Year 2021-2022

Types of data in Cluster analysis, Partitioning Methods (k-Means, k-Medoids), Hierarchical Methods (Agglomerative, Divisive).

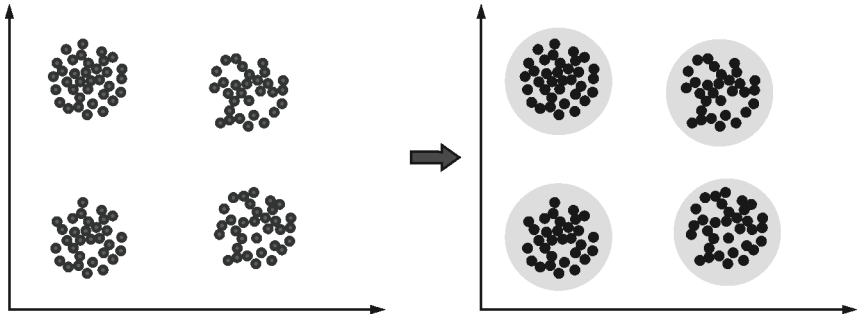
4.1 INTRODUCTION

- Clustering is an unsupervised Machine Learning-based Algorithm that comprises a group of data points into clusters so that the objects belong to the same group.
- Clustering helps to splits data into several subsets. Each of these subsets contains data similar to each other, and these subsets are called **clusters**.
- Now that the data from our customer base is divided into clusters, we can make an informed decision about who we think is best suited for this product.



(1D1)**Fig. 4.1.1: Clustering Example**

- Let's understand this with an example, suppose we are a market manager, and we have a new tempting product to sell. We are sure that the product would bring enormous profit, as long as it is sold to the right people. So, how can we tell who is best suited for the product from our company's huge customer base? So here is where clustering plays an important role.



(1D2)**Fig. 4.1.2 : Four Clusters from the set of unlabeled data**

4.1.1 Cluster

- Cluster is a group of objects that belongs to the same class. In other words, similar objects are grouped in one cluster and dissimilar objects are grouped in another cluster.
- Dissimilarities and similarities are assessed based on the attribute values describing the objects and often involve distance measures.

4.1.2 Cluster Analysis

- Cluster Analysis in data mining means that to find out the group of objects which are similar to each other in the group but are different from the object in other groups.
- A good clustering algorithm aims to obtain clusters whose:
  (a) The intra-cluster similarities are high. It implies that the data present inside the cluster is similar to one another.
  (b) The inter-cluster similarity is low. It means each cluster holds data that is not similar to other data.

4.1.3 Applications of Cluster Analysis

- In many applications, clustering analysis is widely used, such as data analysis, market research, pattern recognition, and image processing.
- It assists marketers to find different groups in their client base and based on the purchasing patterns. They can characterize their customer groups.
- It helps in allocating documents on the internet for data discovery.
- Clustering is also used in tracking applications such as detection of credit card fraud.
- As a data mining function, cluster analysis serves as a tool to gain insight into the distribution of data to analyze the characteristics of each cluster.
- In terms of biology, it can be used to determine plant and animal taxonomies, categorization of genes with the same functionalities and gain insight into structure inherent to populations.
- It helps in the identification of areas of similar land that are used in an earth observation database and the identification of house groups in a city according to house type, value, and geographical location.

4.1.4 Requirements of Clustering in Data Mining

The following points throw light on why clustering is required in data mining :

- **Scalability :** We need highly scalable clustering algorithms to deal with large databases.
- **Ability to deal with different kinds of attributes :** Algorithms should be capable to be applied on any kind of data such as interval-based (numerical) data, categorical, and binary data.
- **Discovery of clusters with attribute shape :** The clustering algorithm should be capable of detecting clusters of arbitrary shape. They should not be bounded to only distance measures that tend to find spherical cluster of small sizes.
- **High dimensionality :** The clustering algorithm should not only be able to handle low-dimensional data but also the high dimensional space.
- **Ability to deal with noisy data :** Databases contain noisy, missing or erroneous data. Some algorithms are sensitive to such data and may lead to poor quality clusters.
- **Interpretability :** The clustering results should be interpretable, comprehensible and usable.

4.1.5 Difference between Classification and Clustering

Sr. No.	Classification	Clustering
1	Classification is the process of classifying the data with the help of class labels.	In clustering, there are no predefined class labels.
2	Classification is supervised learning.	Clustering is unsupervised learning.
3	In classification, algorithms like Decision trees, Bayesian classifiers are used	In Clustering, algorithms like k-means, k-medoids, Expectation-Maximization is used.
4	Classification has prior knowledge of classes.	The cluster doesn't have any prior knowledge of classes.
5	Example: classification between gender.	Example: discovery of patterns.

4.2 TYPES OF DATA IN CLUSTER ANALYSIS

4.2.1 Types of Data Structures

- First of all, let us know what types of data structures are widely used in cluster analysis.
- Suppose that a data set to be clustered contains n objects, which may represent persons, houses, documents, countries, and so on.
- Main memory-based clustering algorithms typically operate on either of the following two data structures.

1. **Data Matrix** (or object by variable structure)
2. **Dissimilarity Matrix** (or object by object structure)

1. Data Matrix

- This represents n objects, such as persons, with p variables (also called measurements or attributes), such as age, height, weight, gender, race and so on.
- The structure is in the form of a relational table, or n-by-p matrix (n objects x p variables).
- The Data Matrix is often called a two-mode matrix since the rows and columns of this represent the different entities.

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ x_{f1} & \cdots & x_{ff} & \cdots & x_{fp} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

2. Dissimilarity Matrix

- This stores a collection of proximities that are available for all pairs of n objects.
- It is often represented by a n-by-n table, where d(i, j) is the measured difference or dissimilarity between objects i and j.
- In general, d(i, j) is a non-negative number that is close to 0 when objects i and j are highly similar or "near" each other and becomes larger when they differ more.
- Here, d(i, j) = d(j, i) and d(i, i) = 0.
- This is also called as one mode matrix since the rows and columns of this represent the same entity.

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,2) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & d(n,3) & \cdots & 0 \end{bmatrix}$$

4.2.2 Types of Data

1. Interval-Scaled Variables

- Interval-scaled variables are continuous measurements of a roughly linear scale.
- Typical examples include weight and height, latitude and longitude coordinates (e.g., when clustering houses), and weather temperature.
- The measurement unit used can affect the clustering analysis. For example, changing measurement units from meters to inches for height, or from kilograms to pounds for weight, may lead to a very different clustering structure.
- In general, expressing a variable in smaller units will lead to a larger range for that variable, and thus a larger effect on the resulting clustering structure.
- To help avoid dependence on the choice of measurement units, the data should be standardized. Standardizing measurements attempts to give all variables an equal weight.
- This is especially useful when given no prior knowledge of the data. However, in some applications, users may intentionally want to give more weight to a certain set of variables than to others.
- For example, when clustering basketball player candidates, we may prefer to give more weight to the variable height.
- Distances are normally used to measure the similarity or dissimilarity between two data objects.
- One of the popular distance measure is ***Minkowski distance.***

$$d(i, j) = \sqrt[q]{|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \ldots + |x_{ip} - x_{jp}|^q}$$

where $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ are two p-dimensional data objects, and q is a positive integer

- If $q = 1$, d is Manhattan distance given by

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \ldots + |x_{ip} - x_{jp}|$$

- If q = 2, d is Euclidean distance measure by

$$d(i, j) = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \ldots + |x_{ip} - x_{jp}|^2}$$

- Both the Euclidean distance and Manhattan distance satisfy the following mathematical requirements of a distance function:

$$d(i,j) \geq 0$$

$$d(i,i) = 0$$

$$d(i,j) = d(j,i)$$

$$d(i,j) \leq d(i,k) + d(k,j)$$

2. Binary Variables

- A binary variable is a variable that can take only 2 values.
- For example, generally, gender variables can take 2 variables male and female.
- **Contingency Table for Binary Data**

Let us consider binary values 0 and 1

	1	0	sum
1	a	b	a + b
0	c	d	c + d
sum	a + c	b + d	p

Let p = a + b + c + d

- Distance measure of symmetric binary variables :

$$d(i, j) = \frac{b + c}{a + b + c + d}$$

- Distance measure of asymmetric binary variables:

$$d(i, j) = \frac{b + c}{a + b + c}$$

- Jaccard coefficient (*similarity* measure for *asymmetric* binary variables):

$$sim_{jaccard}\, d(i, j) = \frac{a}{a + b + c}$$

3. Nominal or Categorical Variables

- A generalization of the binary variable in that it can take more than 2 states, e.g., red, yellow, blue, green.
- **Method 1 :** Simple matching
- The dissimilarity between two objects i and j can be computed based on the simple matching.
- **m:** Let 'm' be number of matches (i.e., the number of variables for which i and j are in the same state).
- **p:** Let 'p' be total no of variables.

$$d(i, j) = \frac{p - m}{p}$$

- **Method 2 :** use a large number of binary variables
- Creating a new binary variable for each of the M nominal states.

4. Ordinal Variables

- An ordinal variable can be discrete or continuous.
- Order is important, e.g., rank.
- Can be treated like interval-scaled.
- Replace x_{if} by their rank, $r_{if} \in \{1, \ldots, M_f\}$
- Map the range of each variable onto [0, 1] by replacing *i*-th object in the *f*-th variable.

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

- Compute the dissimilarity using methods for interval-scaled variables.

5. Ratio-scaled Variables

- A positive measurement on a nonlinear scale, approximately at exponential scale, such as Ae^{Bt} or Ae^{-Bt}.

Method

- Treat them like interval-scaled variables.
- Apply logarithmic transformation $y_{if} = \log(x_{if})$.
- Treat them as continuous ordinal data, treat their rank as interval-scaled.

6. Variables of Mixed Type

- A database may contain all the six types of variables symmetric binary, asymmetric binary, nominal, ordinal, interval, and ratio.
- And those combined are called as mixed-type variables.

4.3 CLUSTERING METHODS

Clustering methods can be classified into the following categories

1. Partitioning Method
2. Hierarchical Method
3. Density-based Method
4. Grid-Based Method
5. Model-Based Method
6. Constraint-based Method

4.3.1 Partitioning Method

Suppose we are given a database of 'n' objects and the partitioning method constructs 'k' partition of data. Each partition will represent a cluster and $k \leq n$. It means that it will classify the data into k groups, which satisfy the following requirements:

- Each group contains at least one object.
- Each object must belong to exactly one group.

Points to remember

- For a given number of partitions (say k), the partitioning method will create an initial partitioning.
- Then it uses the iterative relocation technique to improve the partitioning by moving objects from one group to other.

4.3.2 Hierarchical Methods

This method creates a hierarchical decomposition of the given set of data objects. We can classify hierarchical methods on the basis of how the hierarchical decomposition is formed. There are two approaches here –

- Agglomerative Approach
- Divisive Approach

Agglomerative Approach

This approach is also known as the bottom-up approach. In this, we start with each object forming a separate group. It keeps on merging the objects or groups that are close to one another. It keeps on doing so until all of the groups are merged into one or until the termination condition holds.

Divisive Approach

This approach is also known as the top-down approach. In this, we start with all of the objects in the same cluster. In the continuous iteration, a cluster is split up into smaller clusters. It is down until each object in one cluster or the termination condition holds. This method is rigid, i.e., once a merging or splitting is done, it can never be undone.

Approaches to Improve Quality of Hierarchical Clustering

Here are the two approaches that are used to improve the quality of hierarchical clustering

- Perform careful analysis of object linkages at each hierarchical partitioning.
- Integrate hierarchical agglomeration by first using a hierarchical agglomerative algorithm to group objects into micro-clusters, and then performing macro-clustering on the micro-clusters.

4.3.3 Density-based Method

This method is based on the notion of density. The basic idea is to continue growing the given cluster as long as the density in the neighborhood exceeds some threshold, i.e., for each data point within a given cluster, the radius of a given cluster has to contain at least a minimum number of points.

4.3.4 Grid-based Method

In this, the objects together form a grid. The object space is quantized into finite number of cells that form a grid structure. The major advantage of this method is fast processing time. It is dependent only on the number of cells in each dimension in the quantized space.

4.3.5 Model-based Method

In this method, a model is hypothesized for each cluster to find the best fit of data for a given model. This method locates the clusters by clustering the density function. It reflects spatial distribution of the data points. This method also provides a way to automatically determine the number of clusters based on standard statistics, taking outlier or noise into account. It therefore yields robust clustering methods.

4.3.6 Constraint-based Method

In this method, the clustering is performed by the incorporation of user or application-oriented constraints. A constraint refers to the user expectation or the properties of desired clustering results. Constraints provide us with an interactive way of communication with the clustering process. Constraints can be specified by the user or the application requirement.

4.4 PARTITIONING METHODS

- It is the simplest and most fundamental version of cluster analysis.
- It organizes the objects of a set into several exclusive clusters or groups.
- Here, we assume that the number of clusters is given prior. This is the starting point for partitioning methods.
- We have two types of partitioning methods:
  1. **k-Means** : Each cluster is represented by the centre of the cluster.
  2. **k-Medoids or PAM (Partitioning Around Medoids)** : Each cluster is represented by one of the objects in the cluster.

4.4.1 k-Means Clustering

- k-means clustering is simple unsupervised learning algorithm developed by J. MacQueen in 1967 and then J.A Hartigan and M.A Wong in 1975.
- k-means tries to partition x data points into the set of k clusters where each data point is assigned to its closest cluster.
- This method is defined by the objective function which tries to minimize the sum of all squared distances within a cluster, for all clusters.

The objective function is defined as :

$$\arg_s \min \sum_{x_j \in S_i} \|x_j - \mu_i\|^2$$

where x_j is a data point in the data set, S_i is a cluster (set of data points and μ_i is the cluster mean (the center of cluster of S_i)

K-Means Clustering Algorithm

Algorithm : k-means.

The k-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

Input :

k : the number of clusters,

D : a data set containing n objects.

Output : A set of k clusters.

Method :

(1) arbitrarily choose k objects from D as the initial cluster centers;

(2) **repeat**

(3) (re)assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster;

(4) update the cluster means, that is, calculate the mean value of the objects for each cluster;

(5) **until** no change;

Ex. 4.4.1 : Using k-means clustering, cluster the following data into two clusters and show each step.
{2, 4, 10, 12, 3, 20, 30, 11, 25}

Soln. :

Step 1 : Randomly partition the given data set into two clusters and find the cluster mean.

$k_1 = \{2, 3\} \rightarrow m_1 = 2.5$

$k_2 = \{4, 10, 12, 20, 30, 11, 25\} \rightarrow m_2 = 16$

Step 2 : Reassign the data points to each cluster and find again the cluster mean.

$k_1 = \{2, 3, 4\} \rightarrow m_1 = 3$

$k_2 = \{10, 12, 20, 30, 11, 25\} \rightarrow m_2 = 18$

Step 3 : Reassign and find cluster mean.

$k_1 = \{2, 3, 4, 10\} \rightarrow m_1 = 4.75$

$k_2 = \{12, 20, 30, 11, 25\} \rightarrow m_2 = 19.6$

Step 4 : Reassign and find cluster mean.

$k_1 = \{2, 3, 4, 10, 11, 12\} \rightarrow m_1 = 7$

$k_2 = \{20, 30, 25\} \rightarrow m_2 = 25$

Step 5 : Reassign and find cluster mean.

$k_1 = \{2, 3, 4, 10, 11, 12\} \rightarrow m_1 = 7$

$k_2 = \{20, 30, 25\} \rightarrow m_2 = 25$

Step 6 : Stop. The clusters in step 4 and 5 are same.

Final answer: $k_1 = \{2, 3, 4, 10, 11, 12\}$ and $k_2 = \{20, 30, 25\}$

UEx. 4.4.2 MU - June 2021

Use K-means algorithm to create 3 - clusters for given set of values :{2, 3, 6, 8, 9, 12, 15, 18, 22}

☑ **Soln. :**

- **Step 1 : Randomly partition data into three clusters and calculate the mean value for each cluster.**

  $k_1 = \{2, 8, 15\} \rightarrow m_1 = 8.3$

  $k_2 = \{3, 9, 18\} \rightarrow m_2 = 10$

  $k_3 = \{6, 12, 22\} \rightarrow m_3 = 13.3$

- **Step 2 : Reassign the data points to each cluster and find again the cluster mean.**

  $k_1 = \{2, 3, 6, 8, 9\} \rightarrow m_1 = 5.6$

  $k_2 = \{\ \} \rightarrow m_2 = 0$

  $k_3 = \{12, 15, 18, 22\} \rightarrow m_3 = 16.75$

- **Step 3 : Reassign and find cluster mean**

  $k_1 = \{3, 6, 8, 9\} \rightarrow m_1 = 6.5$

  $k_2 = \{2\} \rightarrow m_2 = 2$

  $k_3 = \{12, 15, 18, 22\} \rightarrow m_3 = 16.75$

- **Step 4 : Reassign and find cluster mean**

  $k_1 = \{6, 8, 9\} \rightarrow m_1 = 7.6$

  $k_2 = \{2, 3\} \rightarrow m_2 = 2.5$

  $k_3 = \{12, 15, 18, 22\} \rightarrow m_3 = 16.75$

- **Step 5 : Reassign and find cluster mean**

  $k_1 = \{6, 8, 9\} \rightarrow m_1 = 7.6$

  $k_2 = \{2, 3\} \rightarrow m_2 = 2.5$

  $k_3 = \{12, 15, 18, 22\} \rightarrow m_3 = 16.75$

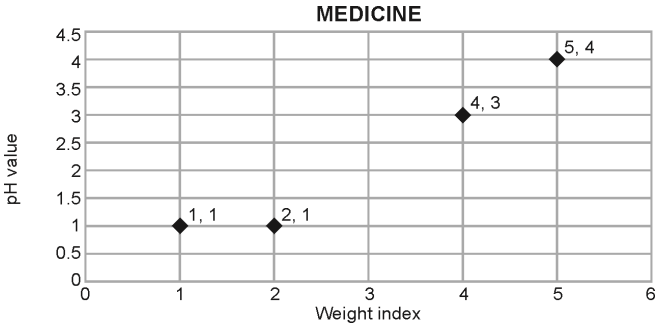
- **Step 6 : Stop. The clusters in step 4 and 5 are same.**

  Final answer: $k_1 = \{6, 8, 9\}$, $k_2 = \{2, 3\}$, $k_3 = \{12, 15, 18, 22\}$

Ex. 4.4.3 : Find clusters using k-means clustering algorithm if we have several objects (4 types of medicines) and each object have two attributes or features as shown in the table below. The goal is to group these objects into k = 2 group of medicine based on the two features (pH and weight index).

Object	Attribute 1 (X) Weight Index	Attribute 2 (Y) pH
Medicine A	1	1
Medicine B	2	1
Medicine C	4	3
Medicine D	5	4

☑ **Soln. :**



(1D3)**Fig. P. 4.4.3(a) : Graphical Representation of Data Points**

Number of clusters k = 2

Initial cluster centres be C1= (1,1) and C2= (2,1)

We will check distance between data points and all cluster centres. We will use Euclidean distance formula for finding distance.

$$\text{Distance } (x, a) = \sqrt{(x - a)^2}$$

OR

$$\text{Distance } [(x, y), (a, b)] = \sqrt{(x - a)^2 + (y - b)^2}$$

As given data is in pair, we will use second formula of Euclidean distance.

Iteration 1 :

We will use following notations for calculating distance.

D1 = Distance from cluster C1 (1, 1)

D2 = Distance from cluster C2 (2, 1)

Data Point (1,1) :

$$D1\ [(1, 1), (1, 1)] = \sqrt{(1 - 1)^2 + (1 - 1)^2} = 0$$

$$D2\ [(1, 1), (2, 1)] = \sqrt{(1 - 2)^2 + (1 - 1)^2} = 1$$

Here 0 is the smallest distance. So data point (1, 1) belongs to cluster C1.

Data Point (2,1) :

$$D1\ [(2, 1), (1, 1)] = \sqrt{(2 - 1)^2 + (1 - 1)^2} = 1$$

$$D2\ [(2, 1), (2, 1)] = \sqrt{(2 - 2)^2 + (1 - 1)^2} = 0$$

Here 0 is the smallest distance. So data point (2,1) belongs to cluster C2.

Data Point (4,3) :

$$D1\ [(4, 3), (1, 1)] = \sqrt{(4 - 1)^2 + (3 - 1)^2} = 3.6$$

$$D2\ [(4, 3), (2, 1)] = \sqrt{(4 - 2)^2 + (3 - 1)^2} = 2.83$$

Here 2.83 is the smallest distance. So data point (4,3) belongs to cluster C2.

Data Point (5,4) :

$$D1\ [(5, 4), (1, 1)] = \sqrt{(5-1)^2 + (4-1)^2} = 5$$

$$D2\ [(5, 4), (2, 1)] = \sqrt{(5-2)^2 + (4-1)^2} = 4.24$$

Here 4.24 is the smallest distance. So data point (5,4) belongs to cluster C2.

Hence, the clusters are

C1 : {A (1, 1)}

C2 : {B (2, 1), C (4, 3), D (5, 4)}

Now recalculate the centre of cluster C2 as:

$$\text{Centre}\ [(x, y), (a, b)] = \left(\frac{x+a}{2}, \frac{y+b}{2}\right)$$

Here, (x, y) = current data point

(a, b) = old data point

$$\text{Updated centre of Cluster C2} = \left(\frac{2+4+5}{3}, \frac{1+3+4}{3}\right)$$

$$= (3.67, 2.67)$$

Iteration 2 :

We will use following notations for calculating distance.

D1 = Distance from cluster C1(1,1)

D2 = Distance from cluster C2(3.67,2.67)

Data Point (1, 1) :

$$D1\ [(1, 1), (1, 1)] = \sqrt{(1-1)^2 + (1-1)^2} = 0$$

$$D2\ [(1, 1), (3.67, 2.67)] = \sqrt{(1-3.67)^2 + (1-2.67)^2}$$

$$= 3.14$$

Here 0 is the smallest distance. So data point (1,1) belongs to cluster C1.

Data Point (2, 1) :

$$D1\ [(2, 1), (1, 1)] = \sqrt{(2-1)^2 + (1-1)^2} = 1$$

$$D2\ [(2, 1), (3.67, 2.67)] = \sqrt{(2-3.67)^2 + (1-2.67)^2}$$

$$= 2.75$$

Here 1 is the smallest distance. So data point (2,1) belongs to cluster C1.

Data Point (4, 3) :

$$D1\ [(4, 3), (1, 1)] = \sqrt{(4-1)^2 + (3-1)^2} = 3.6$$

$$D2\ [(4, 3), (3.67, 2.67)] = \sqrt{(4-3.67)^2 + (3-2.67)^2}$$

$$= 0.47$$

Here 0.47 is the smallest distance. So data point (4,3) belongs to cluster C2.

Data Point (5, 4) :

$$D1\ [(5, 4), (1, 1)] = \sqrt{(5-1)^2 + (4-1)^2} = 6$$

$$D2\ [(5, 4), (3.67, 2.67)] = \sqrt{(5-3.67)^2 + (4-2.67)^2}$$

$$= 1.88$$

Here 1.88 is the smallest distance. So data point (5,4) belongs to cluster C2.

Hence, the clusters are

C1: {A(1, 1), B(2, 1)}

C2: {C(4, 3), D(5, 4)}

Now recalculate the centre of cluster C1 and C2 as :

$$\text{Centre}\ [(x, y), (a, b)] = \left(\frac{x+a}{2}, \frac{y+b}{2}\right)$$

Here, (x, y) = current data point

(a, b) = old data point

$$\text{Updated centre of Cluster C1} = \left(\frac{1+2}{2}, \frac{1+1}{2}\right) = (1.5, 1)$$

$$\text{Updated centre of Cluster C2} = \left(\frac{4+5}{2}, \frac{3+4}{2}\right) = (4.5, 3.5)$$

Iteration 3 :

We will use following notations for calculating distance.

D1 = Distance from cluster C1 (1.5, 1)

D2 = Distance from cluster C2 (4.5, 3.5)

Data Point (1, 1) :

$$D1\ [(1, 1), (1.5, 1)] = \sqrt{(1-1.5)^2 + (1-1)^2} = 0.5$$

$$D2\ [(1, 1), (4.5, 3.5)] = \sqrt{(1-3.67)^2 + (1-2.67)^2}$$

$$= 4.3$$

Here 0.5 is the smallest distance. So data point (1, 1) belongs to cluster C1.

Data Point (2, 1):

$$D1\ [(2, 1), (1.5, 1)] = \sqrt{(2-1.5)^2 + (1-1)^2} = 0.5$$

$$D2\ [(2, 1), (4.5, 3.5)] = \sqrt{(2-4.5)^2 + (1-3.5)^2} = 3.54$$

Here 0.5 is the smallest distance. So data point (2, 1) belongs to cluster C1.

Data Point (4, 3) :

$$D1\ [(4, 3), (1.5, 1)] = \sqrt{(4-1.5)^2 + (3-1)^2} = 3.2$$

$$D2\ [(4, 3)\ ,\ (4.5, 3.5)] = \sqrt{(4 - 4.5)^2 + (3 - 3.5)^2}$$
$$= 0.71$$

Here 0.71 is the smallest distance. So data point (4,3) belongs to cluster C2.

Data Point (5, 4):

$$D1\ [(5, 4), (1.5, 1)] = \sqrt{(5 - 1.5)^2 + (4 - 1)^2} = 4.6$$
$$D2\ [(5, 4)\ ,\ (4.5, 3.5)] = \sqrt{(5 - 4.5)^2 + (4 - 3.5)^2}$$
$$= 0.71$$

Here 0.71 is the smallest distance. So data point (5, 4) belongs to cluster C2.

Hence, the clusters are

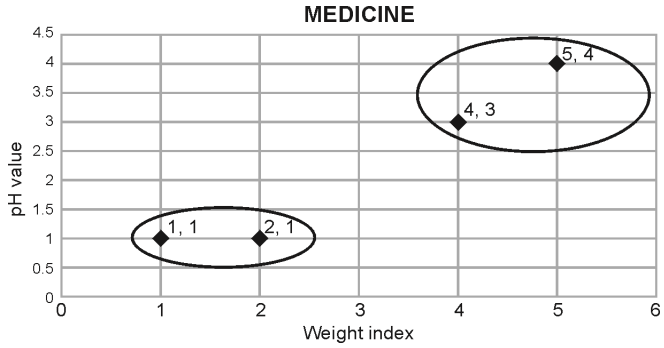
C1: {A(1, 1), B(2, 1)}

C2: {C(4, 3), D(5, 4)}

Comparing the clustering of iteration 2 and iteration 3, we find that objects does not move cluster anymore. Thus, the computation of the k-mean clustering has reached its stability and no more iteration is needed. So the final clusters are:

C1: {A(1, 1), B(2, 1)}

C2: {C(4, 3), D(5, 4)}



(1D4)**Fig. 4.4.3(b) : Clustered Data Points**

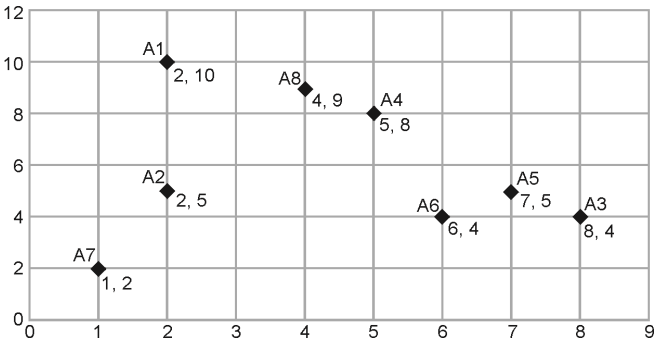
UEx. 4.4.4 (MU - May 2019)

Suppose that the data mining task is to cluster the following eight points (with (x; y) representing location) into three clusters. A1(2,10); A2(2,5); A3(8,4); A4(5,8); A5(7,5); A6(6,4); A7(1,2); A8(4,9). The distance function is Euclidean distance. Suppose initially we assign A1, A4, and A7 as the center of each cluster, respectively. Use the k-means algorithm to show only

(a) The three cluster centers after the first round of execution and

(b) The final three clusters

☑ **Soln. :**



(1D5)**Fig. P. 4.4.3(a) : Data Points**

Number of clusters k = 3

Initial cluster centres be C1= A1(2, 10), C2 = A4(5, 8) and C3 = A7(1, 2)

We will check distance between data points and all cluster centres. We will use Euclidean distance formula for finding distance.

$$Distance\ [(x, y), (a, b)] = \sqrt{(x - a)^2 + (y - b)^2}$$

Iteration 1 :

A1(2, 10)

$$Distance\ [(2, 10), (2.10)] = \sqrt{(2 - 2)^2 + (10 - 10)^2}$$
$$= 0 \leftarrow \text{smaller}$$
$$Distance\ [(2, 10), (5, 8)] = \sqrt{(2 - 5)^2 + (10 - 8)^2} = 3.6$$
$$Distance\ [(2, 10), (1, 2)] = \sqrt{(2 - 1)^2 + (10 - 2)^2} = 8.06$$

A1(2, 10) belongs to cluster C1.

A2(2, 5)

$$Distance\ [(2, 5), (2, 10)] = \sqrt{(2 - 2)^2 + (5 - 10)^2} = 5$$
$$Distance\ [(2, 5), (5, 8)] = \sqrt{(2 - 5)^2 + (5 - 8)^2} = 4.2$$
$$Distance\ [(2, 5), (1, 2)] = \sqrt{(2 - 1)^2 + (5 - 2)^2}$$
$$= 3.16 \leftarrow \text{smaller}$$

A2(2, 5) belongs to cluster C3.

A3(8, 4)

$$Distance\ [(8, 4), (2, 10)] = \sqrt{(8 - 2)^2 + (4 - 10)^2} = 8.5$$
$$Distance\ [(8, 4), (5, 8)] = \sqrt{(8 - 5)^2 + (4 - 8)^2}$$
$$= 5 \leftarrow \text{smaller}$$
$$Distance\ [(8, 4), (1, 2)] = \sqrt{(8 - 1)^2 + (4 - 2)^2} = 7.28$$

A3(8, 4) belongs to cluster C2.

A4(5, 8)

Distance [(5, 8), (2, 10)] = $\sqrt{(5-2)^2 + (8-10)^2} = 3.6$

Distance [(5, 8), (5, 8)] = $\sqrt{(5-5)^2 + (8-8)^2}$

= 0 ← smaller

Distance [(5, 8), (1, 2)] = $\sqrt{(5-1)^2 + (8-2)^2} = 7.21$

A4(5, 8) belongs to cluster C2.

A5(7, 5)

Distance [(7, 5), (2, 10)] = $\sqrt{(7-2)^2 + (5-10)^2} = 7.07$

Distance [(7, 5), (5, 8)] = $\sqrt{(7-5)^2 + (5-8)^2}$

= 3.61 ← smaller

Distance [(7, 5), (1, 2)] = $\sqrt{(7-1)^2 + (5-2)^2} = 6.71$

A5(7,5) belongs to cluster C2.

A6(6, 4)

Distance [(6, 4), (2, 10)] = $\sqrt{(6-2)^2 + (4-10)^2} = 7.21$

Distance [(6, 4), (5, 8)] = $\sqrt{(6-5)^2 + (4-8)^2} = 4.12$ ← smaller

Distance [(6, 4), (1, 2)] = $\sqrt{(6-1)^2 + (4-2)^2} = 5.38$

A6(6,4) belongs to cluster C2.

A7(1, 2)

Distance [(1, 2), (2, 10)] = $\sqrt{(1-2)^2 + (2-10)^2} = 8.06$

Distance [(1, 2), (5, 8)] = $\sqrt{(1-5)^2 + (2-8)^2} = 7.21$

Distance [(1, 2), (1, 2)] = $\sqrt{(1-1)^2 + (2-2)^2}$

= 0 ← smaller

A7(1,2) belongs to cluster C3.

A8(4, 9)

Distance [(4, 9), (2, 10)] = $\sqrt{(4-2)^2 + (9-10)^2} = 2.24$

Distance [(4, 9), (5, 8)] = $\sqrt{(4-5)^2 + (9-8)^2}$

= 1.41 ← smaller

Distance [(4, 9), (1, 2)] = $\sqrt{(4-1)^2 + (9-2)^2} = 7.62$

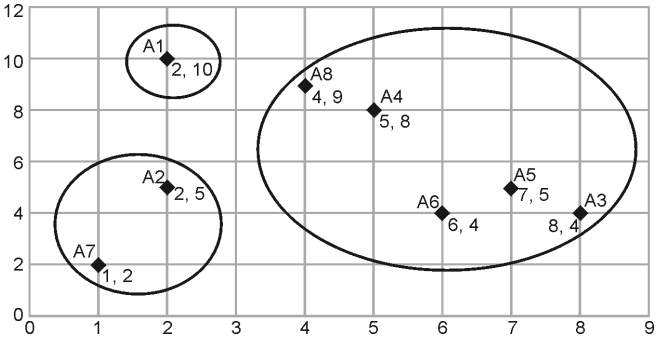
A8(4, 9) belongs to cluster C2.

After iteration 1,

Cluster C1 = {A1(2,10)}

Cluster C2 = {A3(8,4); A4(5,8); A5(7,5); A6(6,4); A8(4,9)}

Cluster C2 = {A2(2,5); A7(1,2)}



(1D6)**Fig. P. 4.4.4(b) : Clustering after 1st iteration**

Iteration 2 :

Centres of new clusters

Cluster C1 = (2, 10)

Cluster C2 = $\left(\frac{8+5+7+6+4}{5}, \frac{4+8+5+4+9}{5}\right) = (6, 6)$

Cluster C3 = $\left(\frac{2+1}{2}, \frac{5+2}{2}\right) = (1.5, 3.5)$

A1(2, 10)

Distance [(2, 10), (2, 10)] = $\sqrt{(2-2)^2 + (10-10)^2}$

= 0 ← smaller

Distance [(2, 10), (6, 6)] = $\sqrt{(2-6)^2 + (10-6)^2} = 5.66$

Distance [(2, 10), (1.5, 3.5)] = $\sqrt{(2-1.5)^2 + (10-3.5)^2}$

= 6.52

A1(2, 10) belongs to cluster C1.

A2(2, 5)

Distance [(2, 5), (2, 10)] = $\sqrt{(2-2)^2 + (5-10)^2} = 5$

Distance [(2, 5), (6, 6)] = $\sqrt{(2-6)^2 + (5-6)^2} = 4.12$

Distance [(2, 5), (1.5, 3.5)] = $\sqrt{(2-1.5)^2 + (5-3.5)^2}$

= 1.58 ← smaller

A2(2, 5) belongs to cluster C3.

A3(8, 4)

Distance [(8, 4), (2, 10)] = $\sqrt{(8-2)^2 + (4-10)^2} = 8.5$

Distance [(8, 4), (6, 6)] = $\sqrt{(8-6)^2 + (4-6)^2}$

= 2.83← smaller

Distance [(8, 4), (1.5, 3.5)] = $\sqrt{(8-1.5)^2 + (4-3.5)^2} = 6.52$

A3(8,4) belongs to cluster C2.

A4(5, 8)

Distance [(5, 8), (2, 10)] = $\sqrt{(5-2)^2 + (8-10)^2} = 3.6$

Distance [(5, 8), (6, 6)] = $\sqrt{(5-6)^2 + (8-6)^2}$

= 2.24← smaller

Distance [(5, 8), (1.5, 3.5)] = $\sqrt{(8-1.5)^2 + (8-3.5)^2}$

= 5.70

A4(5,8) belongs to cluster C2.

A5(7, 5)

Distance [(7, 5), (2, 10)] = $\sqrt{(7-2)^2 + (5-10)^2}$ = 7.07

Distance [(7, 5), (6, 6)] = $\sqrt{(7-6)^2 + (5-6)^2}$ = 1.41 ← smaller

Distance [(7, 5), (1.5, 3.5)] = $\sqrt{(7-1.5)^2 + (5-3.5)^2}$ = 5.70

A5(7, 5) belongs to cluster C2.

A6(6, 4)

Distance [(6, 4), (2, 10)] = $\sqrt{(6-2)^2 + (4-10)^2}$ = 7.21

Distance [(6, 4), (6, 6)] = $\sqrt{(6-6)^2 + (4-6)^2}$

= 2← smaller

Distance [(6, 4), (1.5, 3.5)] = $\sqrt{(6-1.5)^2 + (4-3.5)^2}$ = 4.52

A6(6, 4) belongs to cluster C2.

A7(1, 2)

Distance [(1, 2), (2, 10)] = $\sqrt{(1-2)^2 + (2-10)^2}$ = 8.06

Distance [(1, 2), (6, 6)] = $\sqrt{(1-6)^2 + (2-6)^2}$ = 6.40

Distance [(1, 2), (1.5, 3.5)] = $\sqrt{(1-1.5)^2 + (2-3.5)^2}$

= 1.58 ← smaller

A7(1, 2) belongs to cluster C3.

A8(4, 9)

Distance [(4, 9), (2, 10)] = $\sqrt{(4-2)^2 + (9-10)^2}$ = 2.24 ← smaller

Distance [(4, 9), (6, 6)] = $\sqrt{(4-6)^2 + (9-6)^2}$ = 3.61

Distance [(4, 9), (1.5, 3.5)] = $\sqrt{(4-1.5)^2 + (9-3.5)^2}$ = 6.04

A8(4, 9) belongs to cluster C1.

After iteration 2,

Cluster C1 = {A1 (2, 10); A8(4, 9)}

Cluster C2 = {A3 (8, 4); A4(5, 8); A5(7, 5); A6(6, 4)}

Cluster C2 = {A2 (2, 5); A7(1, 2)}

Iteration 3 :

Centres of new clusters

Cluster C1 = $\left(\frac{2+4}{2}, \frac{10+9}{2}\right)$ = (3, 9.5)

Cluster C2 = $\left(\frac{8+5+7+6}{4}, \frac{4+8+5+4}{4}\right)$ = (6.5, 5.25)

Cluster C3 = $\left(\frac{2+1}{2}, \frac{5+2}{2}\right)$ = (1.5, 3.5)

A1(2, 10)

Distance [(2, 10), (3, 9.5)] = $\sqrt{(2-3)^2 + (10-9.5)^2}$

= 1.12 ← smaller

Distance [(2, 10), (6.5, 5.25)] = $\sqrt{(2-6.5)^2 + (10-5.25)^2}$ = 6.54

Distance [(2, 10), (1.5, 3.5)] = $\sqrt{(2-1.5)^2 + (10-3.5)^2}$

= 6.52

A1(2, 10) belongs to cluster C1.

A2(2, 5)

Distance [(2, 5), (3, 9.5)] = $\sqrt{(2-3)^2 + (5-9.5)^2}$ = 2.35

Distance [(2, 5), (6.5, 5.25)] = $\sqrt{(2-6.5)^2 + (5-5.25)^2}$ = 4.51

Distance [(2, 5), (1.5, 3.5)] = $\sqrt{(2-1.5)^2 + (5-3.5)^2}$

= 1.58 ← smaller

A2(2, 5) belongs to cluster C3.

A3(8, 4)

Distance [(8, 4), (3, 9.5)] = $\sqrt{(8-3)^2 + (4-9.5)^2}$ = 7.43

Distance [(8, 4), (6.5, 5.25)] = $\sqrt{(8-6.5)^2 + (4-5.25)^2}$

= 1.95← smaller

Distance [(8, 4), (1.5, 3.5)] = $\sqrt{(8-1.5)^2 + (4-3.5)^2}$ = 6.52

A3(8, 4) belongs to cluster C2.

A4(5, 8)

Distance [(5, 8), (3, 9.5)] = $\sqrt{(5-3)^2 + (8-9.5)^2}$

= 2.5 ← smaller

Distance [(5, 8), (6.5, 5.25)] = $\sqrt{(5-6.5)^2 + (8-5.25)^2}$ = 3.13

Distance [(5, 8), (1.5, 3.5)] = $\sqrt{(5-1.5)^2 + (8-3.5)^2}$

= 5.70

A4(5, 8) belongs to cluster C1.

A5(7, 5)

Distance [(7, 5), (3, 9.5)] = $\sqrt{(7-3)^2 + (5-9.5)^2}$ = 6.02

Distance [(7, 5), (6.5, 5.25)] = $\sqrt{(7-6.5)^2 + (5-5.25)^2}$

= 0.56 ← smaller

Distance [(7, 5), (1.5, 3.5)] = $\sqrt{(7-1.5)^2 + (5-3.5)^2}$ = 5.70

A5(7, 5) belongs to cluster C2.

A6(6, 4)

Distance [(6, 4), (3, 9.5)] = $\sqrt{(6-3)^2 + (4-9.5)^2} = 6.26$

Distance [(6, 4), (6.5, 5.25)] = $\sqrt{(6-6.5)^2 + (4-5.25)^2}$

= 1.35 ← smaller

Distance [(6, 4), (1.5, 3.5)] = $\sqrt{(6-1.5)^2 + (4-3.5)^2} = 4.52$

A6(6, 4) belongs to cluster C2.

A7(1, 2)

Distance [(1, 2), (3, 9.5)] = $\sqrt{(1-3)^2 + (2-9.5)^2} = 7.76$

Distance [(1, 2), (6.5, 5.25)] = $\sqrt{(1-6.5)^2 + (2-5.25)^2} = 6.38$

Distance [(1, 2), (1.5, 3.5)] = $\sqrt{(1-1.5)^2 + (2-3.5)^2}$

= 1.58 ← smaller

A7(1, 2) belongs to cluster C3.

A8(4, 9)

Distance [(4, 9), (3, 9.5)] = $\sqrt{(4-3)^2 + (9-9.5)^2}$

= 1.12← smaller

Distance [(4, 9), (6.5, 5.25)] = $\sqrt{(4-6.5)^2 + (9-5.25)^2} = 7.68$

Distance [(4, 9), (1.5, 3.5)] = $\sqrt{(4-1.5)^2 + (9-3.5)^2} = 6.04$

A8(4, 9) belongs to cluster C1.

After iteration 3,

Cluster C1 = {A1(2, 10); A4(5, 8); A8(4, 9)}

Cluster C2 = {A3(8, 4); A5(7, 5); A6(6, 4)}

Cluster C2 = {A2(2, 5); A7(1, 2)}

Iteration 4 :

Centres of new clusters

Cluster C1 = $\left(\frac{2+5+4}{3}, \frac{10+8+9}{3}\right) = (3.67, 9)$

Cluster C2 = $\left(\frac{8+7+6}{3}, \frac{4+5+4}{3}\right) = (7, 4.33)$

Cluster C3 = $\left(\frac{2+1}{2}, \frac{5+2}{2}\right) = (1.5, 3.5)$

A1(2, 10)

Distance [(2, 10), (3.67, 9)] = $\sqrt{(2-3.67)^2 + (10-9)^2}$

= 1.95 ← smaller

Distance [(2, 10), (7, 4.33)] = $\sqrt{(2-7)^2 + (10-4.33)^2} = 6.01$

Distance [(2, 10), (1.5, 3.5)] = $\sqrt{(2-1.5)^2 + (10-3.5)^2} = 6.52$

A1(2, 10) belongs to cluster C1.

A2(2, 5)

Distance [(2, 5), (3.67, 9)] = $\sqrt{(2-3.67)^2 + (5-9)^2} = 4.33$

Distance [(2, 5), (7, 4.33)] = $\sqrt{(2-7)^2 + (5-4.33)^2} = 5.04$

Distance [(2, 5), (1.5, 3.5)] = $\sqrt{(2-1.5)^2 + (5-3.5)^2}$

= 1.58 ← smaller

A2(2, 5) belongs to cluster C3.

A3(8, 4)

Distance [(8, 4), (3.67, 9)] = $\sqrt{(8-3.67)^2 + (4-9)^2} = 6.61$

Distance [(8, 4), (7, 4.33)] = $\sqrt{(8-7)^2 + (4-4.33)^2}$

= 1.05 ← smaller

Distance [(8, 4), (1.5, 3.5)] = $\sqrt{(8-1.5)^2 + (4-3.5)^2} = 6.52$

A3(8, 4) belongs to cluster C2.

A4(5, 8)

Distance [(5, 8), (3.67, 9)] = $\sqrt{(5-3.67)^2 + (8-9)^2}$

= 1.66← smaller

Distance [(5, 8), (7, 4.33)] = $\sqrt{(5-7)^2 + (8-4.33)^2} = 4.17$

Distance [(5, 8), (1.5, 3.5)] = $\sqrt{(5-1.5)^2 + (8-3.5)^2} = 5.70$

A4(5, 8) belongs to cluster C1.

A5(7, 5)

Distance [(7, 5), (3.67, 9)] = $\sqrt{(7-3.67)^2 + (5-9)^2} = 5.20$

Distance [(7, 5), (7, 4.33)] = $\sqrt{(7-7)^2 + (5-4.33)^2}$

= 0.67← smaller

Distance [(7, 5), (1.5, 3.5)] = $\sqrt{(7-1.5)^2 + (5-3.5)^2} = 5.70$

A5(7, 5) belongs to cluster C2.

A6(6, 4)

Distance [(6, 4), (3.67, 9)] = $\sqrt{(6-3.67)^2 + (4-9)^2} = 5.52$

Distance [(6, 4), (7, 4.33)] = $\sqrt{(6-7)^2 + (4-4.33)^2}$

= 1.05 ← smaller

Distance [(6, 4), (1.5, 3.5)] = $\sqrt{(6-1.5)^2 + (4-3.5)^2} = 4.52$

A6(6, 4) belongs to cluster C2.

A7(1, 2)

Distance [(1, 2), (3.67, 9)] = $\sqrt{(1-3.67)^2 + (2-9)^2} = 7.49$

Distance [(1, 2), (7, 4.33)] = $\sqrt{(1-7)^2 + (2-4.33)^2} = 6.44$

Distance [(1, 2), (1.5, 3.5)] = $\sqrt{(1-1.5)^2 + (2-3.5)^2}$

= 1.58 ← smaller

A7(1, 2) belongs to cluster C3.

A8(4, 9)

Distance [(4, 9), (3.67, 9)] = $\sqrt{(4-3.67)^2 + (9-9)^2}$

= 0.33← smaller

Distance [(4, 9), (7, 4.33)] = $\sqrt{(4-7)^2 + (9-4.33)^2} = 5.55$

Distance [(4, 9),(1.5, 3.5)] = $\sqrt{(4-1.5)^2 + (9-3.5)^2} = 6.04$

A8 (4, 9) belongs to cluster C1.

After iteration 4,

Cluster C1 = {A1(2, 10); A4(5, 8); A8(4, 9)}

Cluster C2 = {A3(8, 4); A5(7, 5); A6(6, 4)}

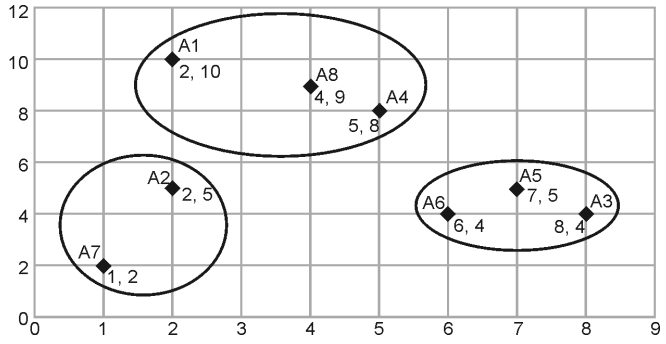
Cluster C2 = {A2(2, 5); A7(1, 2)}

Comparing the clustering of iteration 3 and iteration 4, we find that objects does not move cluster anymore. Thus, the computation of the k-mean clustering has reached its stability and no more iteration is needed. So the final clusters are:

Cluster C1 = {A1(2, 10); A4(5, 8); A8(4, 9)}

Cluster C2 = {A3(8, 4); A5(7, 5); A6(6, 4)}

Cluster C2 = {A2(2, 5); A7(1, 2)}



(1D7)**Fig. P. 4.4.4(c) : Clustering after last iteration**

4.4.2 k-Medoids Clustering

- k-Medoids (also called as Partitioning Around Medoid) algorithm was proposed in 1987 by Kaufman and Rousseeuw.
- A medoid can be defined as the point in the cluster, whose dissimilarities with all the other points in the cluster is minimum.
- The dissimilarity of the medoid(C_i) and object(P_i) is calculated by using $\mathbf{E = |P_i - C_i|}$
- The cost in K-Medoids algorithm is given as,

$$C = \sum_{C_i} \sum_{P_i \in C_i} |P_i - C_i|$$

(A) Algorithm of k-Medoids Clustering

☞ **Algorithm :** k-medoids. PAM, a k-medoids algorithm for partitioning based on medoid or central objects.

Input :

- k : the number of clusters,
- D : a data set containing n objects.

Output : A set of k clusters.

Method :

(1) arbitrarily choose k objects in D as the initial representative objects or seeds;

(2) **repeat**

(3) assign each remaining object to the cluster with the nearest representative object;

(4) randomly select a nonrepresentative object, $\mathbf{o_{random}}$;

(5) compute the total cost, S, of swapping representative object, $\mathbf{o_j}$, with $\mathbf{o_{random}}$;

(6) **if** S < 0 **then** swap $\mathbf{o_j}$with $\mathbf{o_{random}}$ to form the new set of k representative objects;

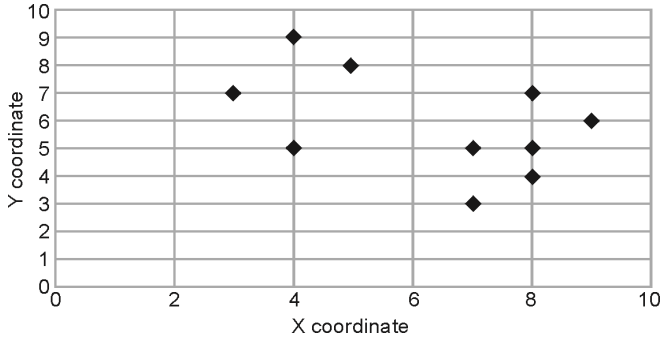
(7) **until** no change;

Ex. 4.4.5 : Coordinates of objects are given below. Apply k-medoids (PAM) to cluster the coordinates into two clusters.

Objects	X	Y
0	8	7
1	3	7
2	4	9
3	9	6
4	8	5
5	5	8
6	7	3
7	8	4
8	7	5
9	4	5

☑ **soln. :**

If a graph is drawn using the above data points, we obtain the following :



(1D8)**Fig. 4.4.5(a) : Object Data Points**

- **Step 1 :**

Randomly select 2 medoids as number of clusters k = 2.

Let **C1 = (4, 5)** and **C2 = (8, 5)** be the two medoids.

- **Step 2 : Calculating cost.**

The dissimilarity of each non-medoid point with the medoids is calculated and tabulated :

Consider object 0 with coordinate (8,7).

Dissimilarity from C1 = |4 – 8| + |5–7| = 6

Dissimilarity from C2 = |8 – 8| + |5–7| = 2

Therefore, the closest medoid is C2.

Likewise, we find dissimilarity and its closest representative medoid for each object which is not the medoid.

Objects	X	Y	Dissimilarity from C1	Dissimilarity from C2	Closest Representative Centroid
0	8	7	6	2	C2
1	3	7	3	7	C1
2	4	9	4	8	C1
3	9	6	6	2	C2
4	8	5	-	-	-
5	5	8	4	6	C1
6	7	3	5	3	C2
7	8	4	5	1	C2
8	7	5	3	1	C2
9	4	5	-	-	-

Each point is assigned to the cluster of that medoid whose dissimilarity is less.

The points 1, 2, 5 go to cluster C1 and 0, 3, 6, 7, 8 go to cluster C2.

The Cost = (3 + 4 + 4) + (3 + 1 + 1 + 2 + 2) = 20

(Cost is the sum of minimum value of dissimilarity value for each data point).

- **Step 3 : Randomly select one non-medoid point and recalculate the cost.**

Let the randomly selected point be (8, 4).

The dissimilarity of each non-medoid point with the medoids C1(4, 5) and C2(8, 4) is calculated and tabulated.

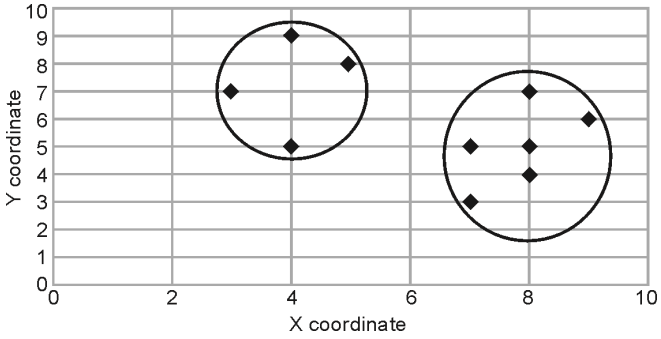
Objects	X	Y	Dissimilarity from C1	Dissimilarity from C2	Closest Representative Centroid
0	8	7	6	3	C2
1	3	7	3	8	C1
2	4	9	4	9	C1
3	9	6	6	3	C2
4	8	5	4	1	C2
5	5	8	4	7	C1
6	7	3	5	2	C2
7	8	4	-	-	-
8	7	5	3	2	C2
9	4	5	-	-	-

Each point is assigned to that cluster whose dissimilarity is less. So, the points 1, 2, 5 go to cluster C1 and 0, 3, 4, 6, 8 go to cluster C2.

The New cost = (3 + 4 + 4) + (3 + 3 + 1 + 2 + 2) = 22

Swap Cost = New Cost – Previous Cost = 22 – 20 = **2 > 0**

As the swap cost is not less than zero, we undo the swap. Hence (4, 5) and (8, 5) are the final medoids. The clustering would be in the following way



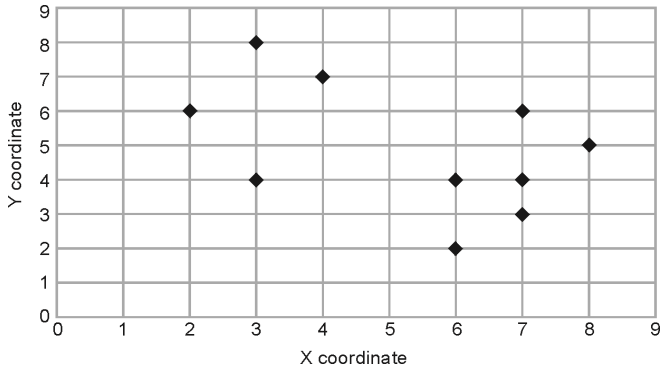
(1D9)**Fig. 4.4.5(b) : Clustered Objects Based on Medoids**

Ex. 4.4.6 : Coordinates of objects are given below. Apply k-medoids (PAM) to cluster the coordinates into two clusters.

Objects	X	Y
1	2	6
2	3	4
3	3	8
4	4	7
5	6	2
6	6	4
7	7	3
8	7	4
9	8	5
10	7	6

☑ **Soln. :**

If a graph is drawn using the above data points, we obtain the following:



(1D10)**Fig. 4.4.6(a) : Object Data Points**

▶ **Step 1 :**

Randomly select 2 medoids as number of clusters k = 2.

Let **C1 = (3, 4)** and **C2 = (7, 4)** be the two medoids.

▶ **Step 2 : Calculating cost.**

The dissimilarity of each non-medoid point with the medoids is calculated and tabulated:

Consider object 1 with coordinate (2, 6).

Dissimilarity from C1 = |3 – 2| + |4 – 6| = 3

Dissimilarity from C2 = |7 – 2| + |4 – 6| = 7

Therefore, the closest medoid is C1.

Likewise, we find dissimilarity and its closest representative medoid for each object which is not the medoid.

Objects	X	Y	Dissimilarity from C1	Dissimilarity from C2	Closest Representative Centroid
1	2	6	3	7	C1
2	3	4	-	-	-
3	3	8	4	8	C1
4	4	7	4	6	C1
5	6	2	5	3	C2
6	6	4	3	1	C2
7	7	3	5	1	C2
8	7	4	-	-	-
9	8	5	6	2	C2
10	7	6	5	2	C2

Each point is assigned to the cluster of that medoid whose dissimilarity is less.

The points 1, 3, 4 go to cluster C1 and 5, 6, 7, 9, 10 go to cluster C2.

The Cost = (3 + 4 + 4) + (3 + 1 + 1 + 2 + 2) = 20

(Cost is the sum of minimum value of dissimilarity value for each data point).

▶ **Step 3 : Randomly select one non-medoid point and recalculate the cost.**

Let the randomly selected point be (7, 3).

The dissimilarity of each non-medoid point with the medoids C1(4, 5) and C2(7, 3) is calculated and tabulated.

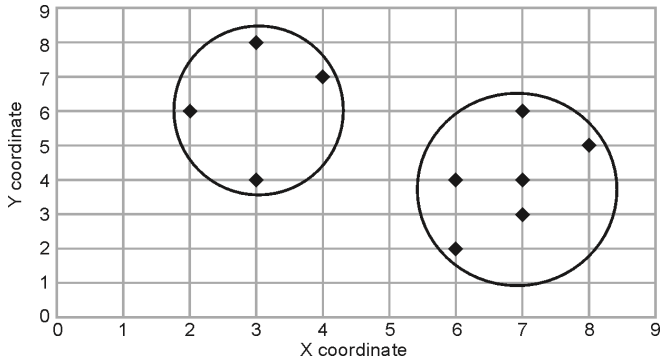
Objects	X	Y	Dissimilarity from C1	Dissimilarity from C2	Closest Representative Centroid
1	2	6	3	8	C1
2	3	4	-	-	-
3	3	8	4	9	C1
4	4	7	4	7	C1
5	6	2	5	2	C2
6	6	4	3	2	C2
7	7	3	-	-	-
8	7	4	4	1	C2
9	8	5	6	3	C2
10	7	6	5	3	C2

Each point is assigned to that cluster whose dissimilarity is less. So, the points 1, 3, 4 go to cluster C1 and 5, 6, 8, 9, 10 go to cluster C2.

The New cost = (3 + 4 + 4) + (3 + 3 + 1 + 2 + 2) = 22

Swap Cost = New Cost – Previous Cost = 22 – 20 = **2 > 0**

As the swap cost is not less than zero, we undo the swap. Hence (3,4) and (7,4) are the final medoids. The clustering would be in the following way



(1D11)**Fig. P. 4.4.6(b) : Clustered Objects Based on Medoids**

(B) The time complexity is O $(k \times (n - k)^2)$

(C) Advantages

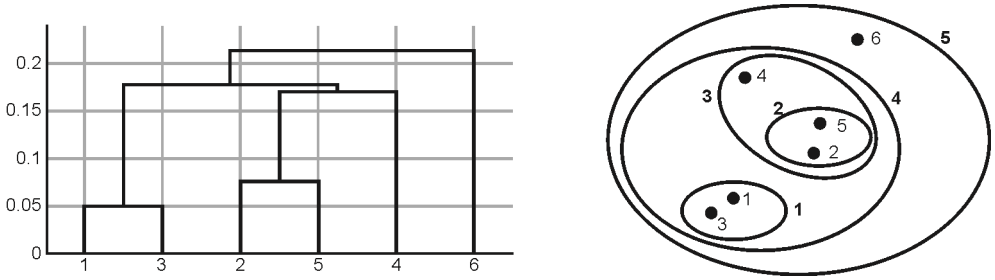
1. It is simple to understand and easy to implement.
2. K-Medoid Algorithm is fast and converges in a fixed number of steps.
3. PAM is less sensitive to outliers than other partitioning algorithms.

(D) Disadvantages

1. The main disadvantage of K-Medoid algorithms is that it is not suitable for clustering non-spherical (arbitrary shaped) groups of objects. This is because it relies on minimizing the distances between the non-medoid objects and the medoid (the cluster centre) – briefly, it uses compactness as clustering criteria instead of connectivity.
2. It may obtain different results for different runs on the same dataset because the first k medoids are chosen randomly.

4.5 HIERARCHICAL CLUSTERING

- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram – A tree-like diagram that records the sequences of merges or splits.



(1D12)**Fig. 4.5.1 : Dendrogram for Hierarchical Clustering**

- Any desired number of clusters can be obtained by 'cutting' the dendrogram at the proper level. So, no assumptions can be made on the number of clusters.
- Hierarchical clustering may correspond to meaningful taxonomies e.g. web (product catalogs).

▶ **Definition** : Given a set of points X = $\{x_1, x_2,\ldots,x_n\}$ find a sequence of nested partitions $P_1,P_2,\ldots,P_n$ of X, consisting of 1, 2,…,n clusters respectively such that $\sum_{i=1}^{n}$ cost (P_i) is minimized.

- Two main types of hierarchical clustering:

1. Agglomerative :

- Start with the points as individual clusters
- At each step, merge the closest pair of clusters until only one cluster (or k clusters) left

2. Divisive :

- Start with one, all-inclusive cluster
- At each step, split a cluster until each cluster contains a point (or there are k clusters)

4.5.1 Agglomerative Clustering Algorithm

(A) Algorithm :

1. Compute the distance matrix between the input data points
2. Let each data point be a cluster
3. **Repeat**
4. Merge the two closest clusters
5. Update the distance matrix
6. Until only a single cluster remains

(B) Distance between two clusters

1. **Single-link distance** between clusters C_i and C_j is the **minimum distance** between any object in C_i and any object in C_j. The distance is defined by the two most similar objects as

   $$\mathbf{D_{sl}\,(C_i, C_j) = \min_{x, y}\,\{d\,(x, y) \mid x \in C_i, y \in C_j\}}$$

2. **Complete-link distance** between clusters C_i and C_j is the **maximum distance** between any object in C_i and any object in C_j. The distance is defined by the two most dissimilar objects as

   $$\mathbf{D_{cl}\,(C_i, C_j) = \max_{x, y}\,\{d\,(x, y) \mid x \in C_i, y \in C_j\}}$$

3. **Average-link distance** between clusters C_i and C_j is the **average distance** between any object in C_i and any object in C_j. The distance is defined as

   $$\mathbf{D_{cl}\,(C_i, C_j)} = \frac{1}{|C_i| \times |C_j|} \sum_{x \in C_i\, y \in C_j} \mathbf{d\,(x, y)}$$

(C) Advantages

1. No prior information about the number of clusters required.
2. Easy to implement and gives best result in some cases.

(D) Disadvantages

1. Algorithm can never undo what was done previously.
2. Time complexity of at least $O(n^2 \log n)$ is required, where *'n'* is the number of data points.
3. Based on the type of distance matrix chosen for merging different algorithms can suffer with one or more of the following:
   (i) Sensitivity to noise and outliers.
   (ii) Breaking large clusters.
   (iii) Difficulty in handling different sized clusters and convex shapes.
   (iv) No objective function is directly minimized.
   (v) Sometimes it is difficult to identify the correct number of clusters by the dendrogram.

Ex. 4.5.1 : Assume that the database D is given by the table below. Follow single link, complete link and average link technique to find clusters in D. Use Euclidean distance measure.

		X	Y
D	P1	0.40	0.53
	P2	0.22	0.38
	P3	0.35	0.32
	P4	0.26	0.19
	P5	0.08	0.41
	P6	0.45	0.30

☑ **Soln. :**

1. Single link distance

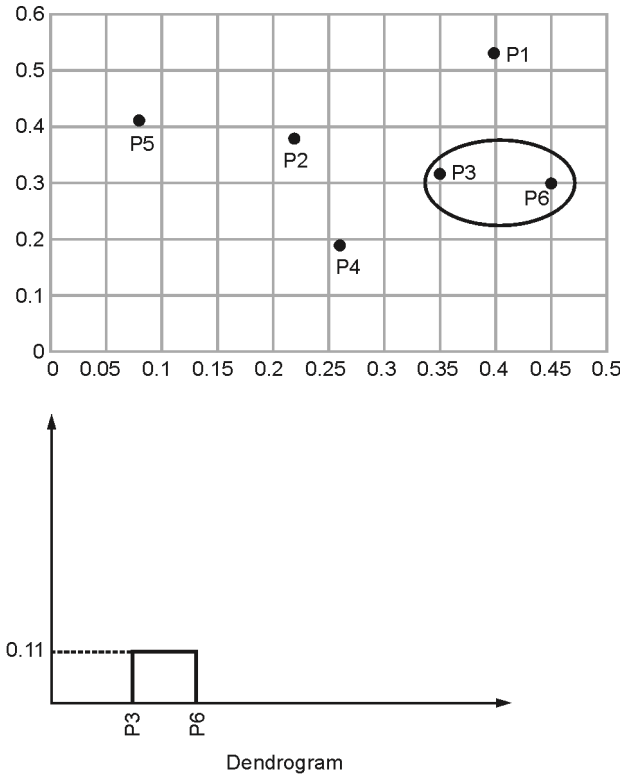
▶ **Step 1 :** Calculate the distance from each object (point) to all other points using Euclidean distance measure and place the numbers in the distance matrix.

Distance $[(x, y), (a, b)] = \sqrt{(x - a)^2 + (y - b)^2}$

Distance Matrix :

P1	0					
P2	0.24	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	**0.11**	0.22	0.39	0
	P1	P2	P3	P4	P5	P6

▶ **Step 2 :** In the above matrix, P6 and P3 are two clusters with shortest distance 0.11, so merge P6 and P3 and make a single cluster (P3, P6). Now, re-compute the distance matrix.



(1D13)**Fig. P. 4.5.1(a)**

To calculate the distance of P1 from (P3, P6) :

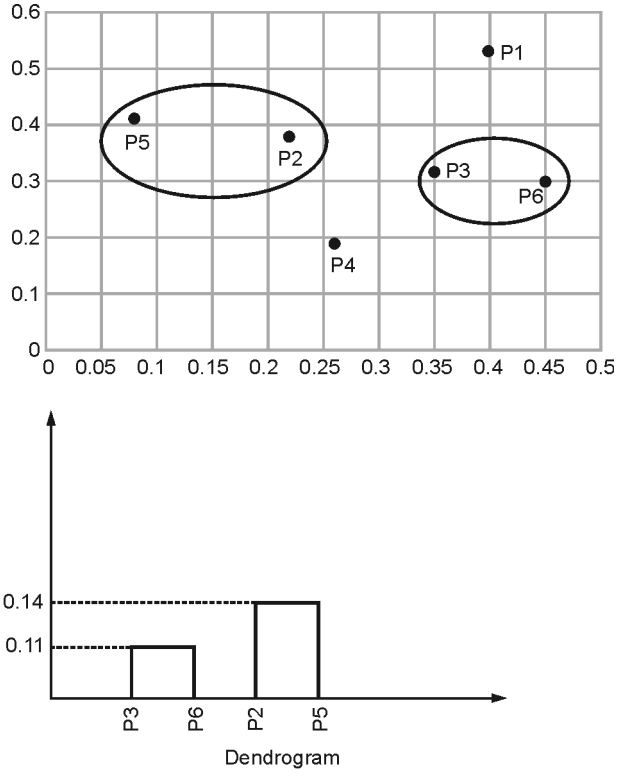
dist((P3, P6), P1) = Min(dist(P3, P1), dist(P6, P1))

= Min(0.22, 0.23) // from original distance matrix

= 0.22

P1	0				
P2	0.24	0			
(P3,P6)	0.22	0.15	0		
P4	0.37	0.20	0.15	0	
P5	0.34	**0.14**	0.28	0.29	0
	P1	P2	(P3,P6)	P4	P5

- **Step 3 :** In the above matrix, P2 and P5 are two clusters with shortest distance 0.14, so merge P2 and P5 and make a single cluster (P2, P5). Now, re-compute the distance matrix as above.

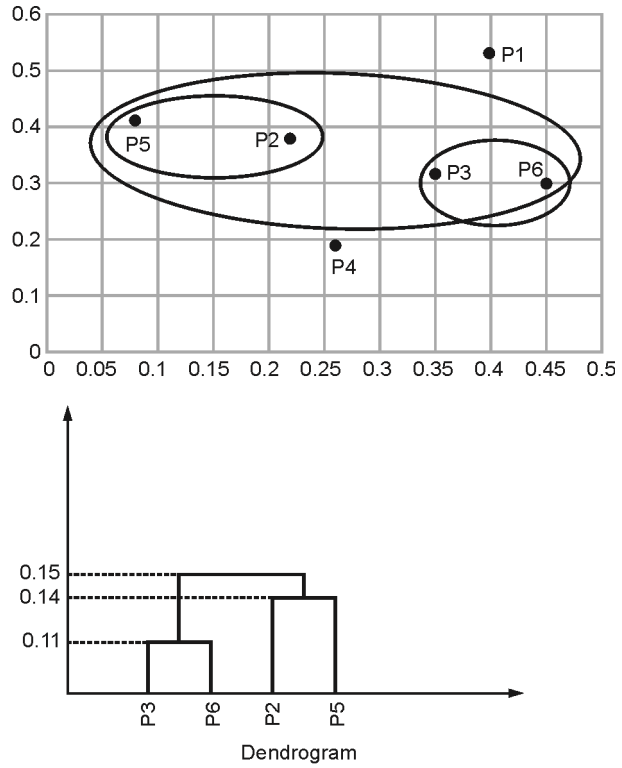


(1D14)**Fig. P. 4.5.1(b)**

P1	0			
(P2,P5)	0.24	0		
(P3,P6)	0.22	**0.15**	0	
P4	0.37	0.20	0.15	0
	P1	(P2,P5)	(P3,P6)	P4

- **Step 4 :** In the above matrix, (P2, P5) and (P3, P6) are two clusters with shortest distance 0.15, so merge P2 and P5 and make a single cluster (P2, P3, P5, P6). Now, re-compute the distance matrix as above.

NOTES

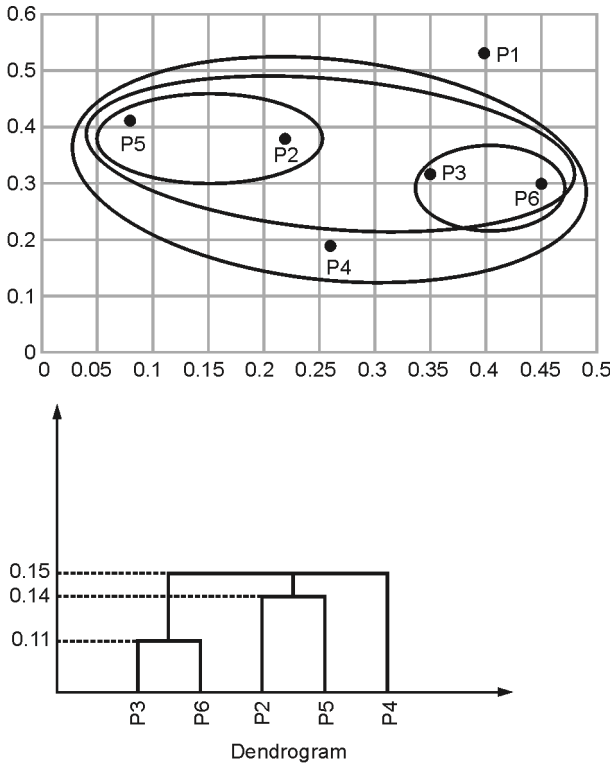


(1D15)**Fig. P. 4.5.1(c)**

P1	0		
(P2,P3,P5,P6)	0.22	0	
P4	0.37	**0.15**	0
	P1	(P2,P3,P5,P6)	P4

- **Step 5 :** In the above matrix, (P2, P3, P5, P6) and P4 are two clusters with shortest distance 0.14, so merge (P2, P3, P5, P6) and P4 and make a single cluster (P2, P3, P4, P5, P6). Now, re-compute the distance matrix as above.

NOTES

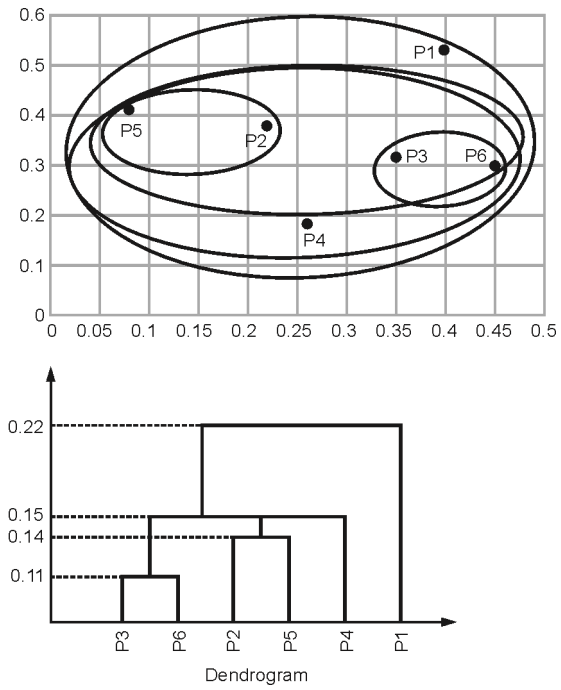


(1D16)**Fig. P. 4.5.1(d)**

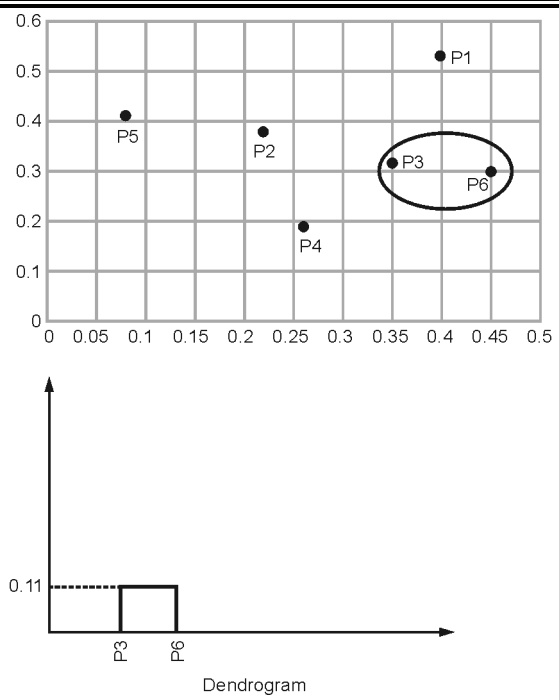
P1	0	
(P2,P3,P4,P5,P6)	**0.22**	0
	P1	(P2,P3,P4,P5,P6)

- **Step 6 :** Looking at the above distance matrix in step 5, we see that (P2,P3,P4,P5,P6) and P1 have the smallest distance 0.22 (the only one left). So, we merge those two in a single cluster. There is no need to re-compute the distance matrix, as there are no more clusters to merge.

NOTES



(1D17)**Fig. P. 4.5.1(e)**



(1D18)**Fig. P. 4.5.1(f)**

2. Complete link distance

- **Step 1 :** Calculate the distance from each object (point) to all other points using Euclidean distance measure and place the numbers in the distance matrix.

Distance $[(x, y), (a, b)] = \sqrt{(x-a)^2 + (y-b)^2}$

Distance Matrix :

P1	0					
P2	0.24	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	**0.11**	0.22	0.39	0
	P1	P2	P3	P4	P5	P6

- **Step 2 :** In the above matrix, P6 and P3 are two clusters with shortest distance 0.11, so merge P6 and P3 and make a single cluster (P3, P6). Now, re-compute the distance matrix.

To calculate the distance of P1 from (P3, P6) :

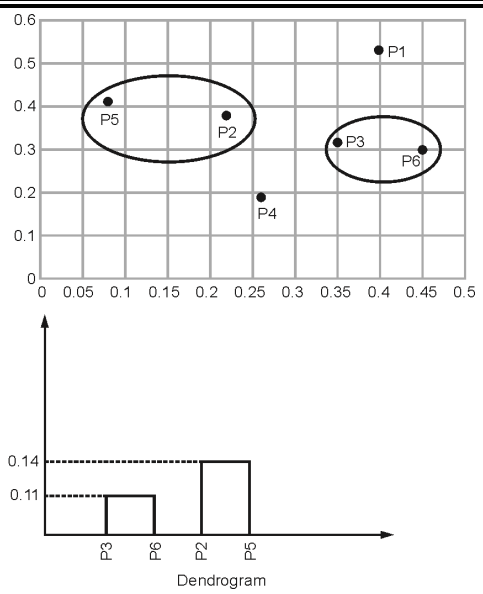
dist((P3, P6), P1) = Max (dist(P3, P1), dist(P6, P1))

= Max (0.22, 0.23) // from original distance matrix

= 0.23

P1	0				
P2	0.24	0			
(P3,P6)	0.23	0.25	0		
P4	0.37	0.20	0.22	0	
P5	0.34	**0.14**	0.39	0.29	0
	P1	P2	(P3,P6)	P4	P5

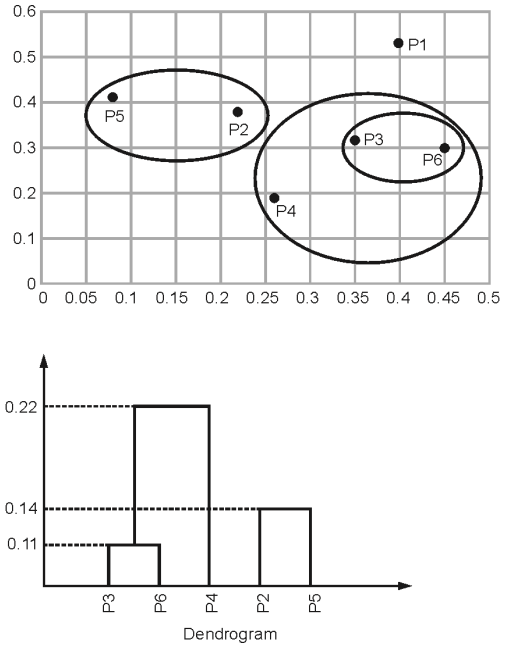
- **Step 3 :** In the above matrix, P2 and P5 are two clusters with shortest distance 0.14, so merge P2 and P5 and make a single cluster (P2, P5). Now, re-compute the distance matrix as above.



(1D19)Fig. P. 4.5.1(g)

P1	0			
(P2,P5)	0.34	0		
(P3,P6)	0.23	0.39	0	
P4	0.37	0.29	**0.22**	0
	P1	(P2,P5)	(P3,P6)	P4

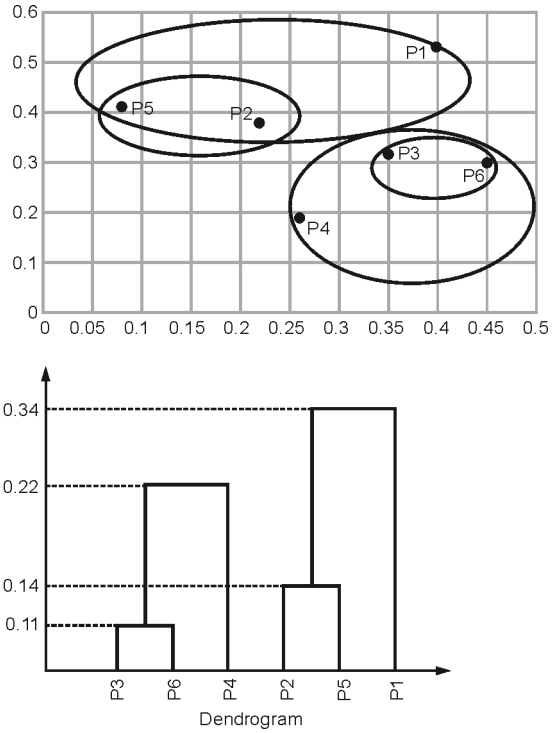
- **Step 4 :** In the above matrix, (P3, P6) and P4 are two clusters with shortest distance 0.22, so merge (P3, P6) and P4 and make a single cluster (P3, P4, P6). Now, re-compute the distance matrix as above.



(1D20)Fig. P. 4.5.1(h)

P1	0		
(P2,P5)	**0.34**	0	
(P3,P4,P6)	0.37	0.39	0
	P1	(P2,P5)	(P3,P4,P6)

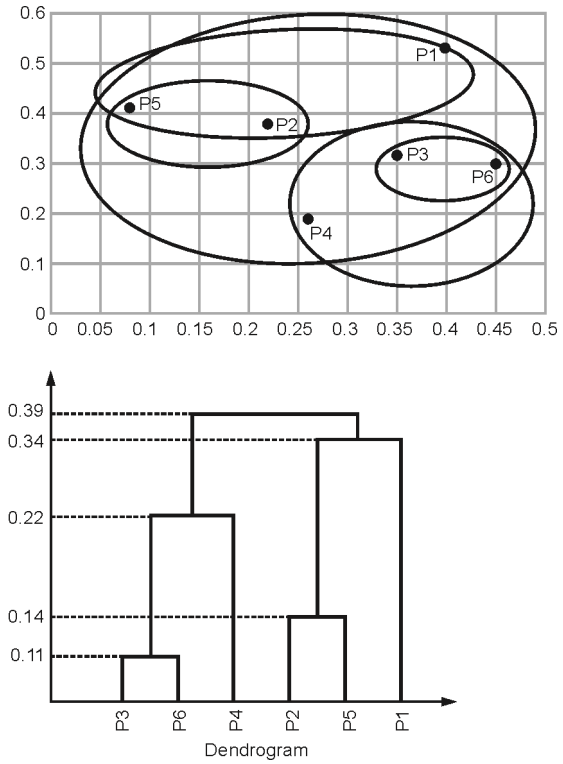
- **Step 5 :** In the above matrix, (P2, P5) and P1 are two clusters with shortest distance 0.34, so merge (P2, P5) and P1 and make a single **c**luster (P1, P2, P5). Now, re-compute the distance matrix as above.



(1D21)Fig. P. 4.5.1(i)

(P1,P2,P5)	0	
(P3,P4,P6)	**0.39**	0
	(P1,P2,P5)	(P3,P4,P6)

- **Step 6 :** Looking at the above distance matrix in step 5, we see that (P1, P2, P5) and (P3, P4, P6) have the smallest distance 0.39 (the only one left). So, we merge those two in a single cluster. There is no need to re-compute the distance matrix, as there are no more clusters to merge.



(1D22)**Fig. P. 4.5.1(j)**

3. Average link distance

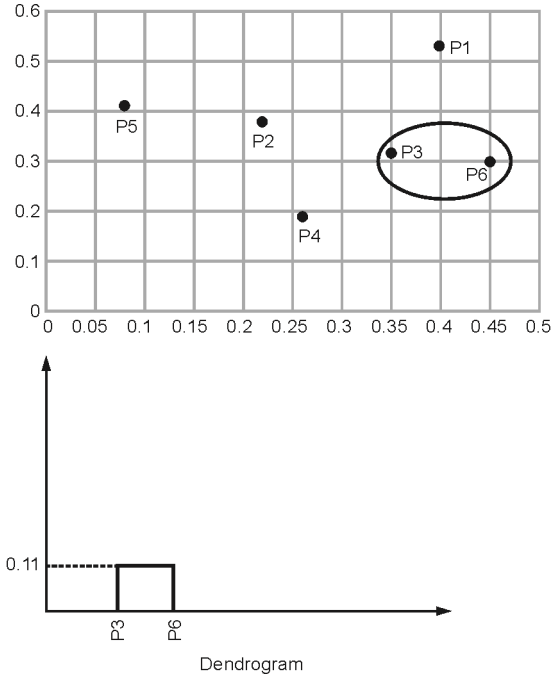
- **Step 1 :** Calculate the distance from each object (point) to all other points using Euclidean distance measure and place the numbers in the distance matrix.

Distance $[(x, y), (a, b)] = \sqrt{(x-a)^2 + (y-b)^2}$

Distance Matrix :

P1	0					
P2	0.24	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	**0.11**	0.22	0.39	0
	P1	P2	P3	P4	P5	P6

- **Step 2 :** In the above matrix, P6 and P3 are two clusters with shortest distance 0.11, so merge P6 and P3 and make a single cluster (P3, P6). Now, re-compute the distance matrix.



(1D23)**Fig. P. 4.5.1(k)**

To calculate the distance of P1 from (P3, P6) :

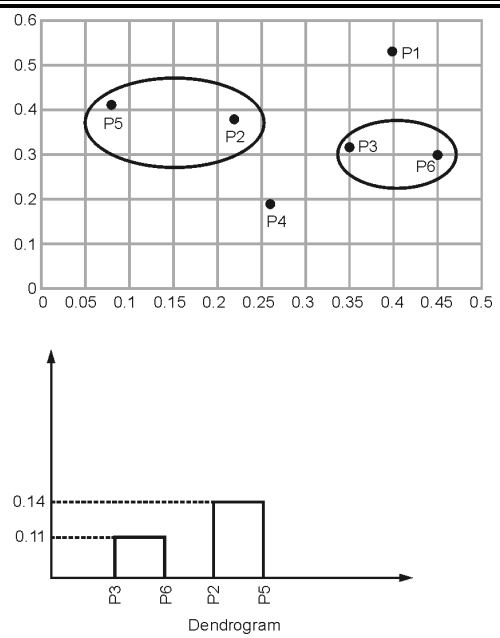
dist((P3, P6), P1) = Avg (dist(P3, P1), dist(P6, P1))

= Avg (0.22, 0.23) // from original distance matrix

= 0.23

P1	0				
P2	0.24	0			
(P3,P6)	0.23	0.20	0		
P4	0.37	0.20	0.19	0	
P5	0.34	**0.14**	0.34	0.29	0
	P1	P2	(P3,P6)	P4	P5

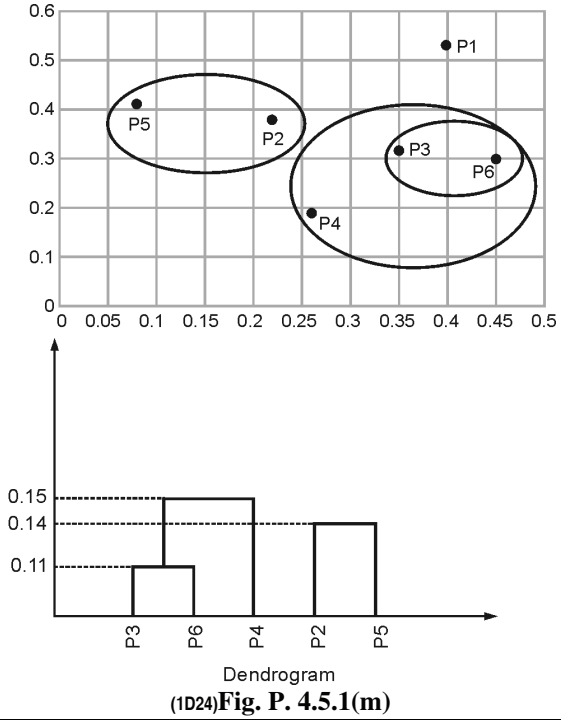
- **Step 3 :** In the above matrix, P2 and P5 are two clusters with shortest distance 0.14, so merge P2 and P5 and make a single cluster (P2, P5). Now, re-compute the distance matrix as above.



(1D23A)**Fig. P. 4.5.1(*l*)**

P1	0			
(P2,P5)	0.29	0		
(P3,P6)	0.22	0.27	0	
P4	0.37	0.22	**0.15**	0
	P1	(P2,P5)	(P3,P6)	P4

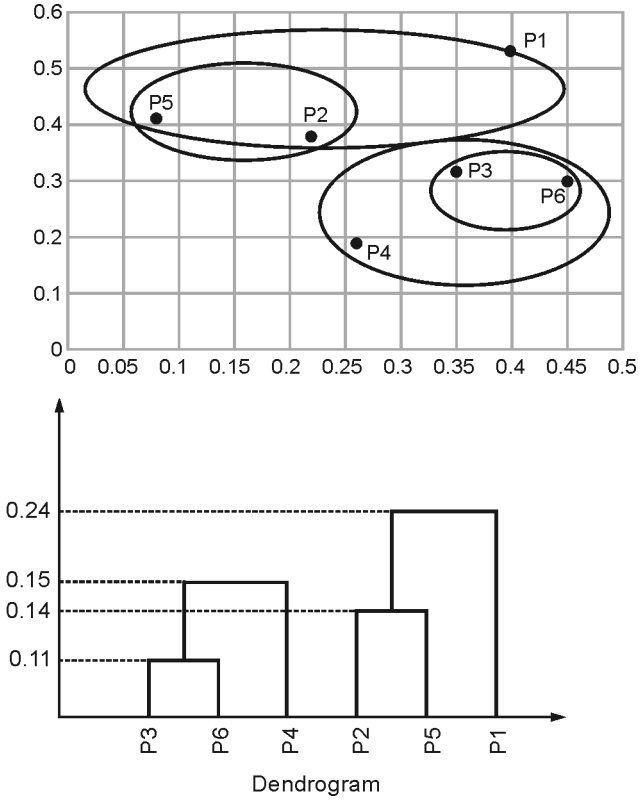
- **Step 4 :** In the above matrix, (P3, P6) and P4 are two clusters with shortest distance 0.15, so merge (P3, P6) and P4 and make a single cluster (P3, P4, P6). Now, re-compute the distance matrix as above.



(1D24)**Fig. P. 4.5.1(m)**

P1	0		
(P2,P5)	**0.24**	0	
(P3,P4,P6)	0.27	0.26	0
	P1	(P2,P5)	(P3,P4,P6)

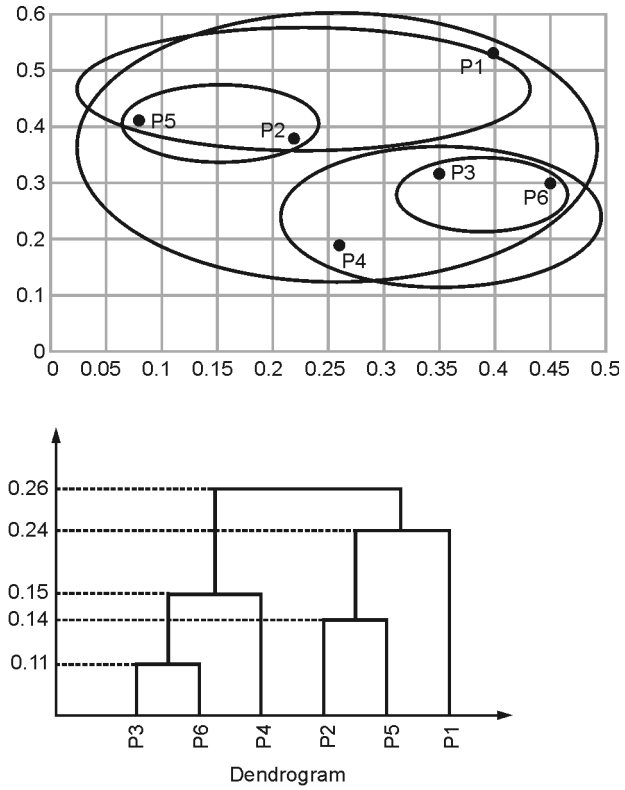
- **Step 5 :** In the above matrix, (P2, P5) and P1 are two clusters with shortest distance 0.24, so merge (P2, P5) and P1 and make a single cluster (P1, P2, P5). Now, re-compute the distance matrix as above.



(1D25)**Fig. P. 4.5.1(n)**

(P1, P2, P5)	0	
(P3, P4, P6)	**0.26**	0
	(P1, P2, P5)	(P3, P4, P6)

- **Step 6 :** Looking at the above distance matrix in step 5, we see that (P1, P2, P5) and (P3, P4, P6) have the smallest distance 0.26 (the only one left). So, we merge those two in a single cluster. There is no need to re-compute the distance matrix, as there are no more clusters to merge.



(1D26)**Fig. P. 4.5.1(o)**

UEx. 4.5.2 **MU - May 2019**

Use complete linkage algorithm to find the clusters from the following dataset.

X	4	8	15	24	24
Y	4	4	8	4	12

☑ **Soln. :**

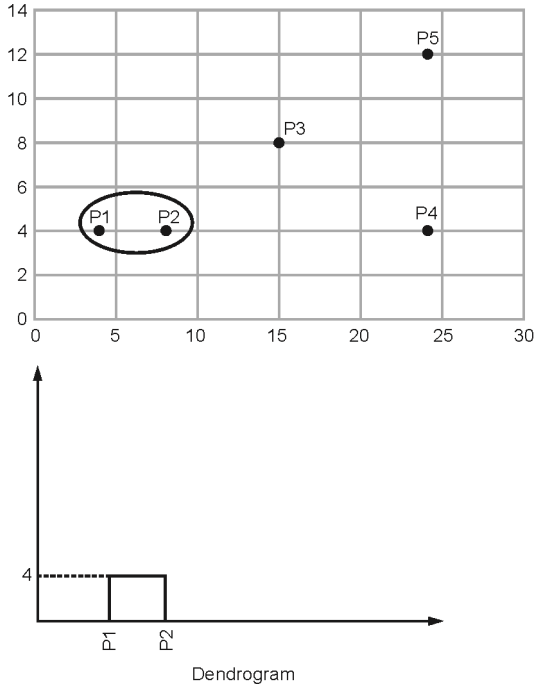
- **Step 1 :** Calculate the distance from each object (point) to all other points using Euclidean distance measure and place the numbers in the distance matrix.

Distance [(x, y), (a, b)] $= \sqrt{(x-a)^2 + (y-b)^2}$

Distance Matrix :

P1	0				
P2	**4**	0			
P3	11.7	8.06	0		
P4	20	16	9.85	0	
P5	21.54	17.89	9.85	8	0
	P1	P2	P3	P4	P5

- **Step 2 :** In the above matrix, P1 and P2 are two clusters with shortest distance 4, so merge P1 and P2 and make a single cluster (P1, P2). Now, re-compute the distance matrix.



(1D27)**Fig. P. 4.5.2(a)**

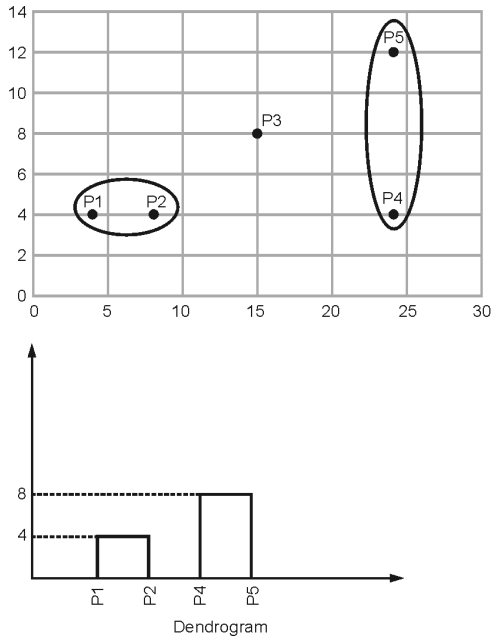
To calculate the distance of P3 from (P1, P2) :

dist((P1, P2), P3) = Max (dist(P1, P3), dist(P2, P3))

= Max (11.7, 8.06) // from original distance matrix = 11.7

(P1,P2)	0			
P3	11.7	0		
P4	20	9.85	0	
P5	21.54	9.85	**8**	0
	(P1,P2)	P3	P4	P5

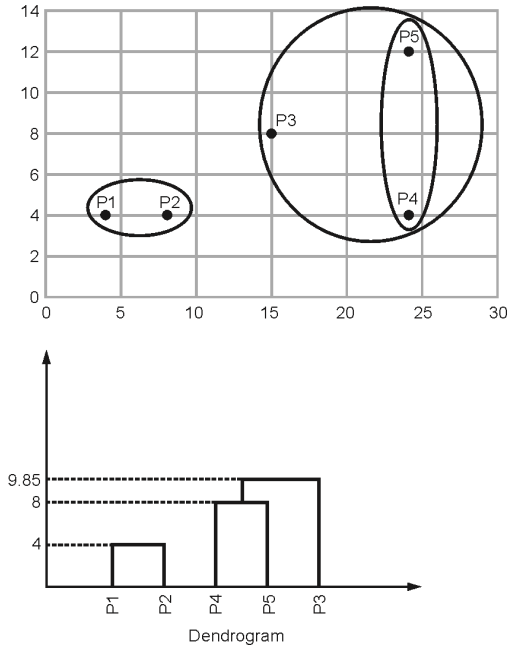
- **Step 3 :** In the above matrix, P4 and P5 are two clusters with shortest distance 8, so merge P4 and P5 and make a single cluster (P4, P5). Now, re-compute the distance matrix.



(1D28)**Fig. P. 4.5.2(b)**

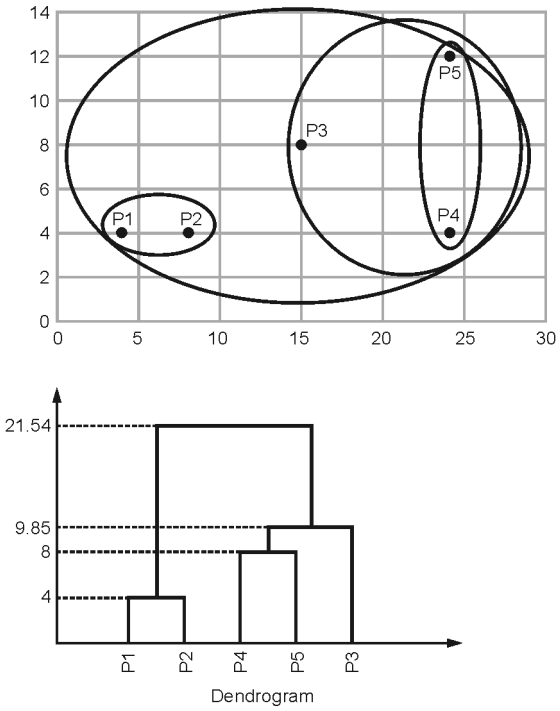
(P1,P2)	0		
P3	11.7	0	
(P4,P5)	21.54	**9.85**	0
	(P1,P2)	P3	(P4,P5)

- **Step 4 :** In the above matrix, P3 and (P4, P5) are two clusters with shortest distance 9.85, so merge P3 and (P4, P5) and make a single cluster (P3, P4, P5). Now, re-compute the distance matrix.



(1D29)**Fig. P. 4.5.2(c)**

(P1,P2)	0	
(P3,P4,P5)	**21.54**	0
	(P1,P2)	(P3,P4,P5)



(1D30)**Fig. P. 4.5.2(d)**

- **Step 5 :** Looking at the above distance matrix in step 4, we see that (P1, P2) and (P3, P4, P5) have the smallest distance 21.54 (the only one left). So, we merge those two in a single cluster. There is no need to re-compute the distance matrix, as there are no more clusters to merge.

UEx. 4.5.3 (MU - Dec. 2019)

Show the dendrogram created by the complete link clustering algorithm for the given set of points.

Points	A	B
P1	2	4
P2	8	2
P3	9	3
P4	1	5
P5	8.5	1

☑ **Soln. :**

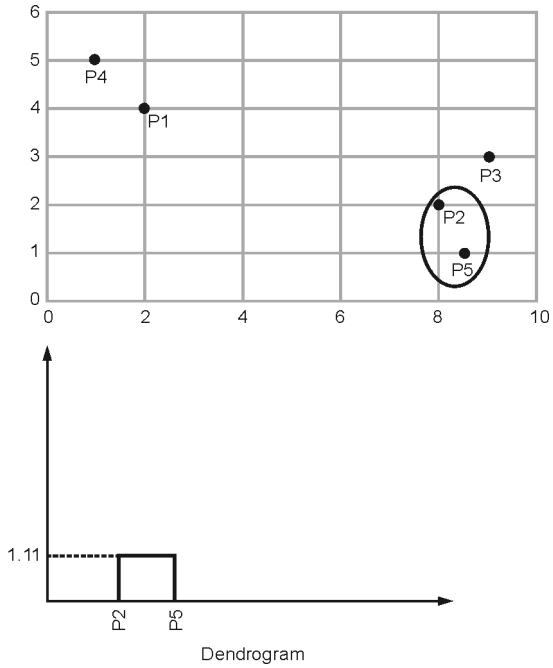
- **Step 1 :** Calculate the distance from each object (point) to all other points using Euclidean distance measure and place the numbers in the distance matrix.

Distance [(x, y), (a, b)] = $\sqrt{(x-a)^2 + (y-b)^2}$

Distance Matrix :

P1	0				
P2	6.32	0			
P3	7.07	1.41	0		
P4	1.41	7.62	8.25	0	
P5	7.16	**1.11**	2.06	8.5	0
	P1	P2	P3	P4	P5

- **Step 2 :** In the above matrix, P2 and P5 are two clusters with shortest distance 1.11, so merge P2 and P5 and make a single cluster (P2, P5). Now, re-compute the distance matrix.



(1D31)**Fig. P. 4.5.3(a)**

To calculate the distance of P1 from (P2, P5) :

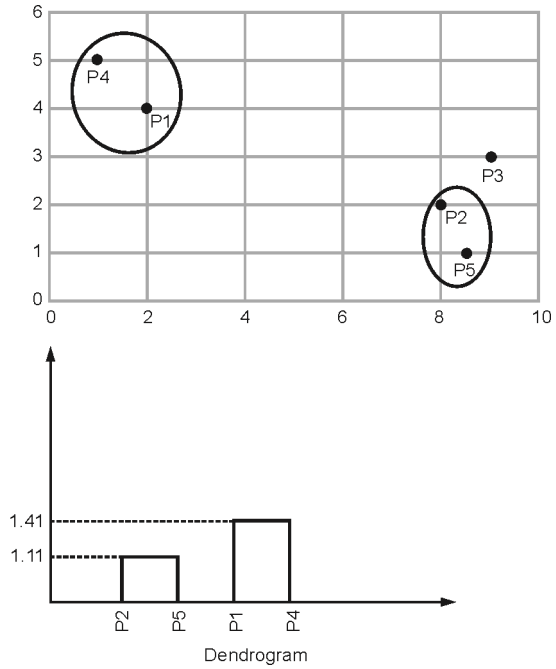
dist((P2, P5), P1) = Max(dist(P2, P1), dist(P5, P1))

= Max(6.32, 7.16) // from original distance matrix

= 7.16

P1	0			
(P2,P5)	7.16	0		
P3	7.07	2.06	0	
P4	**1.41**	8.5	8.25	0
	P1	(P2,P5)	P3	P4

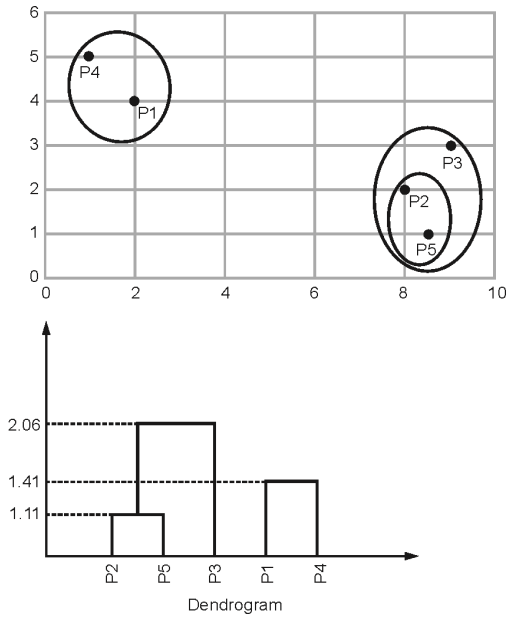
- **Step 3 :** In the above matrix, P1 and P4 are two clusters with shortest distance 1.41, so merge P1 and P4 and make a single cluster (P1, P4). Now, re-compute the distance matrix.



(1D32)**Fig. P. 4.5.3(b)**

(P1,P4)	0		
(P2,P5)	8.5	0	
P3	8.25	**2.06**	0
	(P1,P4)	(P2,P5)	P3

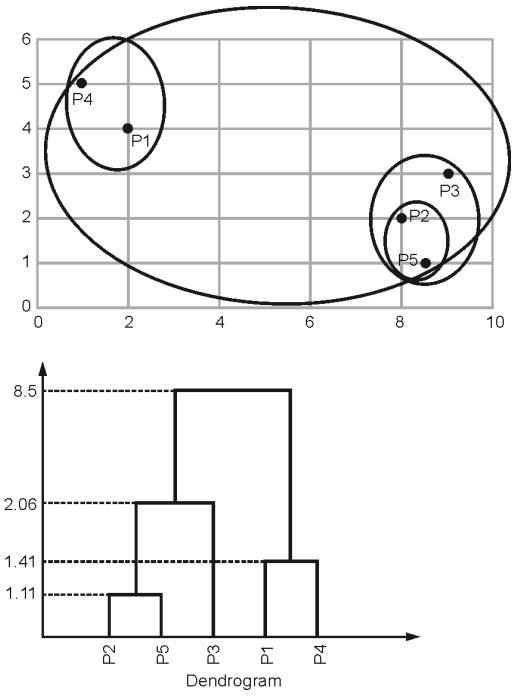
- **Step 4 :** In the above matrix, (P2, P5) and P3 are two clusters with shortest distance 2.06, so merge (P2, P5) and P3 and make a single cluster (P2, P3, P5). Now, re-compute the distance matrix.



(1D33)**Fig. P. 4.5.3(c)**

(P1,P4)	0	
(P2,P3,P5)	**8.5**	0
	(P1,P4)	(P2,P3,P5)

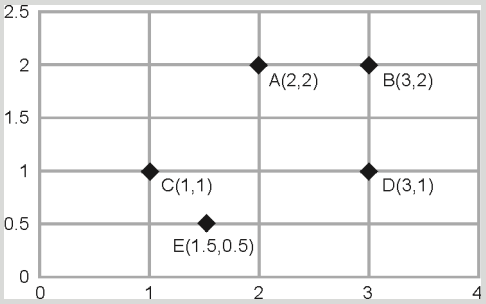
- **Step 5 :** Looking at the above distance matrix in step 4, we see that (P1, P4) and (P2, P3, P5) have the smallest distance 8.5 (the only one left). So, we merge those two in a single cluster. There is no need to re-compute the distance matrix, as there are no more clusters to merge.



(1D34)**Fig. P. 4.5.3(d)**

UEx. 4.5.4 (MU - June 2021)

Use the data given below. Create adjacency matrix. Use complete link algorithm to cluster given data set. Draw dendrogram.



(1D35)**Fig. P. 4.5.4(a)**

☑ **Soln. :**

- **Step 1 :** Calculate the distance from each object (point) to all other points using Euclidean distance measure and place the numbers in the distance matrix.

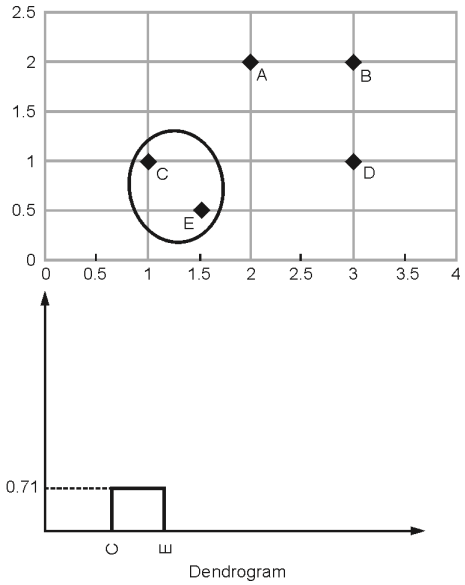
Distance $[(x, y), (a, b)] = \sqrt{(x-a)^2 + (y-b)^2}$

Object	X	Y
A	2	2
B	3	2
C	1	1
D	3	1
E	1.5	0.5

Distance Matrix :

A	0				
B	1	0			
C	1.41	2.24	0		
D	1.41	1	2	0	
E	1.58	2.12	**0.71**	1.58	0
	A	B	C	D	E

- **Step 2 :** In the above matrix, C and E are two clusters with shortest distance 0.71, so merge C and E and make a single cluster (C,E). Now, re-compute the distance matrix.



(1D36)**Fig. P. 4.5.4(b)**

To calculate the distance of A from (C, E) :

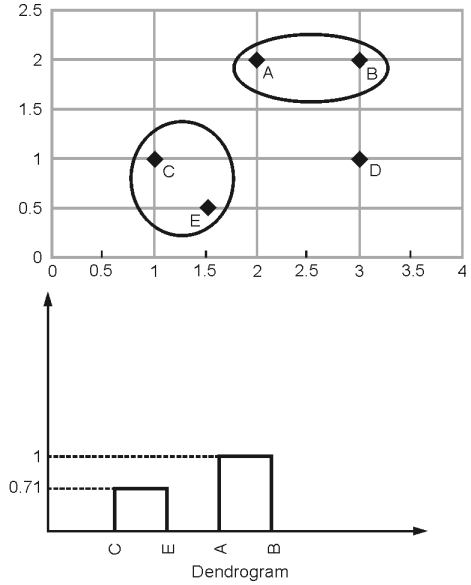
dist((C, E), A) = Max(dist(C, A), dist(E, A))

= Max(1.41, 1.58) // from original distance matrix

= 1.58

A	0			
B	**1**	0		
(C,E)	1.58	2.24	0	
D	1.41	1	2	0
	A	B	(C,E)	D

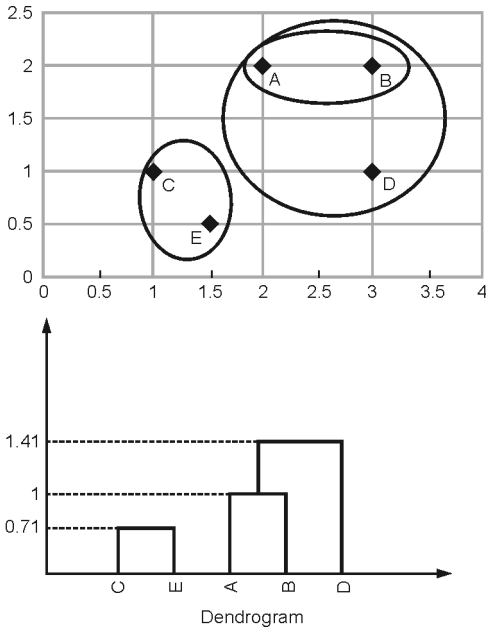
- **Step 3 :** In the above matrix, A and B are two clusters with shortest distance 1, so merge A and B and make a single cluster (A, B). Now, re-compute the distance matrix.



(1D37)**Fig. P. 4.5.4(c)**

(A,B)	0		
(C,E)	2.24	0	
D	**1.41**	2	0
	(A,B)	(C,E)	D

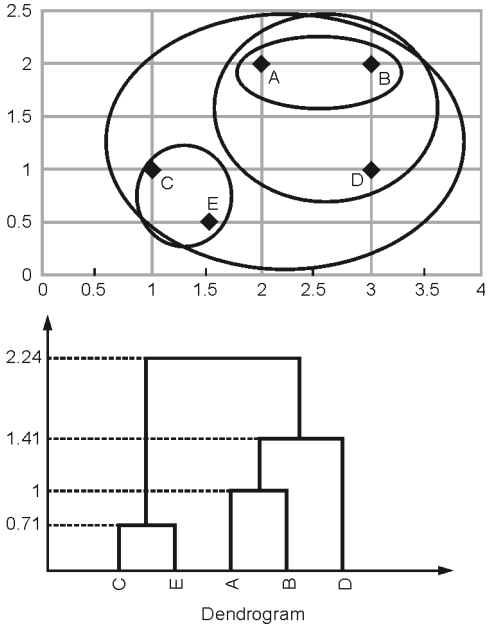
- **Step 3 :** In the above matrix, (A, B) and D are two clusters with shortest distance 1.41, so merge (A, B) and D and make a single cluster (A, B, D). Now, re-compute the distance matrix.



(1D38)**Fig. P. 4.5.4(d)**

(A,B,D)	0	
(C,E)	**2.24**	0
	(A,B,D)	(C,E)

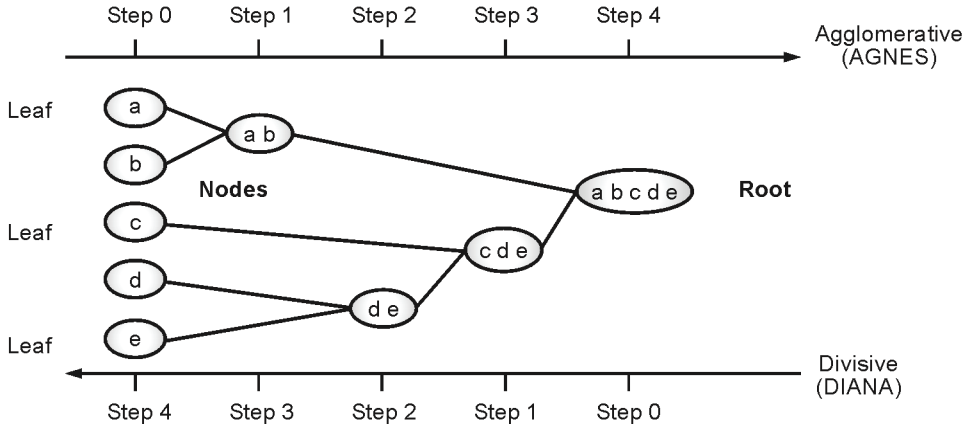
- **Step 5 :** Looking at the above distance matrix in step 4, we see that (A, B, D) and (C, E) have the smallest distance 2.24 (the only one left). So, we merge those two in a single cluster. There is no need to re-compute the distance matrix, as there are no more clusters to merge.



(1D39)**Fig. P. 4.5.4(e)**

4.5.2 Divisive Hierarchical Clustering

- Also called Divisive Analysis (DIANA).
- Divisive Hierarchical approach is commonly known as the top-down approach because in this, it generally starts with all of the objects in the same cluster.
- Then with the continuous iteration, a cluster is split up into smaller clusters by the application of K-means Clustering.
- It is done until each object is in one cluster or the termination condition takes holds.
- This method is rigid i.e., once a merging or splitting is done, it can never be undone.



(1D40)**Fig. P. 4.5.2**

(A) Advantages

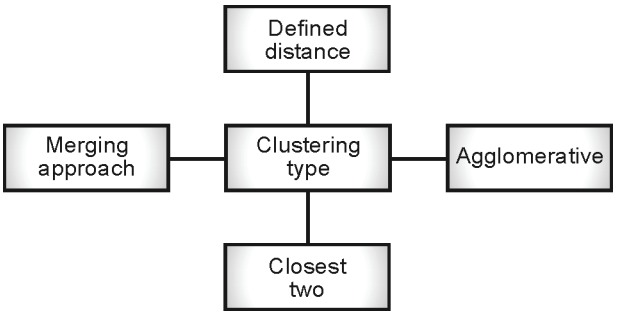
1. It is simple and it derives a hierarchical structure which is more informative.
2. It does not require prior knowledge about the number of clusters.

(B) Disadvantages

1. Divisive Hierarchical clustering works on the principle of merge and split. Selection of merge or split points is critical because once the group of objects is merged or split, it will take into consideration the newly generated clusters for further operations. Thus, undo operation is not supported in this algorithm.
2. Improper selection of merge or split points may lead to low quality clusters.

4.6 MULTIPLE CHOICE QUESTIONS

Q. 4.1 Which of the following clustering type has characteristic shown in the below figure?



(1D41)**Fig. Q. 4.1**

(a) Partitioning (b) Hierarchical
(c) Naïve Bayes (d) ID3 ✓ **Ans. : (b)**

Q. 4.2 The goal of clustering a set of data is to
(a) divide them into groups of data that are near each other
(b) choose the best data from the set
(c) determine the nearest neighbors of each of the data
(d) predict the class of data ✓ **Ans. : (a)**

Q. 4.3 The k-means algorithm _____
(a) always converges to a clustering that minimizes the mean-square vector-representative distance
(b) can converge to different final clustering, depending on initial choice of representatives and is widely used
(c) is typically done by hand, using paper and pencil
(d) should only be attempted by trained professionals ✓ **Ans. : (b)**

Q. 4.4 The choice of k, the number of clusters to partition a set of data into, _____.
(a) is a personal choice that shouldn't be discussed in public
(b) depends on why you are clustering the data
(c) should always be as large as your computer system can handle
(d) has maximum value 10 ✓ **Ans. : (b)**

Q. 4.5 Which of the following statements about the K-means algorithm are correct?
(a) The K-means algorithm is sensitive to outliers.
(b) For different initializations, the K-means algorithm will definitely give the same clustering results.
(c) The centroids in the K-means algorithm may be any observed data points.
(d) The K-means algorithm can detect non-convex clusters. ✓ **Ans. : (a)**

Q. 4.6 The Iris dataset contains information about Iris setosa and versicolor. What is the Euclidean distance between these two objects?

Species	Sepal length	Sepal width	Petal length	Petal width
Iris setosa	4.9	3.0	1.4	0.2
Iris versicolor	5.6	2.5	3.9	1.1

(a) 2.8 (b) 4.6
(C) 22.6 (d) –3.6 ✓ **Ans. : (a)**

Q. 4.7 Which of the following statements is FALSE?
(a) Graphs, time-series data, text, and multimedia data are all examples of data types on which cluster analysis can be performed.
(b) Agglomerative clustering is an example of a hierarchical and distance-based clustering method.
(c) When dealing with high-dimensional data, we sometimes consider only a subset of the dimensions when performing cluster analysis.
(d) We can only visualize the clustering results when the data is 2-dimensional. ✓ **Ans. : (d)**

Q. 4.8 Which of the following statements are true?

(a) Clustering analysis is supervised learning since it does require labeled training data.

(b) It is impossible to cluster objects in a data stream. We must have all the data objects that we need to cluster ready before clustering can be performed.

(c) Clustering analysis has a wide range of applications in tasks such as data summarization, dynamic trend detection, multimedia analysis, and biological network analysis.

(d) When clustering, we want to put two dissimilar data objects into the same cluster. ✓ **Ans. : (c)**

Q. 4.9 Which of the following is not a common consideration and requirement for cluster analysis?

(a) We need to consider how to incorporate user preference for cluster size and shape into the clustering algorithm.

(b) In order to perform cluster analysis, we need to have a similarity measure between data objects.

(c) We need to be able to handle a mixture of different types of attributes (e.g., numerical, categorical).

(d) We must know the number of output clusters a priori for all clustering algorithms. ✓ **Ans. : (d)**

Q. 4.10 Which of the following is not the type of Hierarchical Clustering?

(a) Top-Down Clustering (Divisive)

(b) Bottom-Top Clustering (Agglomerative)

(c) BIRCH

(d) K-means ✓ **Ans. : (d)**

Q. 4.11 What is a Dendrogram?

(a) A tree diagram used to illustrate the arrangement of clusters in hierarchical clustering.

(b) A tree diagram used to illustrate the arrangement of clusters in partitional clustering.

(c) A type of hierarchical clustering.

(d) A type of bar chart diagram to visualize k-means clusters. ✓ **Ans. : (a)**

Q. 4.12 The most important part of _____ is selecting the variables on which clustering is based.

(a) interpreting and profiling clusters

(b) selecting a clustering procedure

(c) assessing the validity of clustering

(d) formulating the clustering problem ✓ **Ans. : (d)**

Q. 4.13 The most commonly used measure of similarity is the _____ or its square.

(a) Euclidean distance (b) City-block distance

(c) Chebychev's distance

(d) Manhattan distance ✓ **Ans. : (a)**

Q. 4.14 _____ is a clustering procedure where all objects start out in one giant cluster. Clusters are formed by dividing this cluster into smaller and smaller clusters.

(a) Non-hierarchical clustering

(b) Divisive clustering

(c) Agglomerative clustering

(d) K-means clustering ✓ **Ans. : (b)**

Q. 4.15 Which of the following is required by K-means clustering?

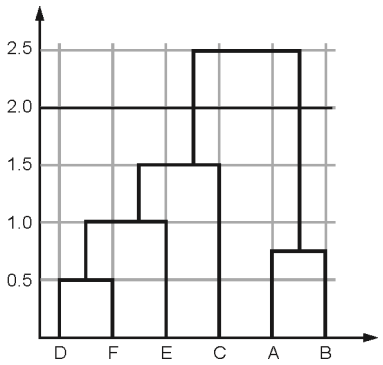
(a) defined distance metric

(b) number of clusters

(c) initial guess as to cluster centroids

(d) all answers are correct ✓ **Ans. : (b)**

Q. 4.16 In the figure above, if you draw a horizontal line on y-axis for y=2. What will be the number of clusters formed?



(1D42)**Fig. Q. 4.16**

(a) 2 (b) 3 (c) 4 (d) 5 ✓ **Ans. : (a)**

Q. 4.17 For which of the following tasks might clustering be a suitable approach?

(a) Given sales data from a large number of products in a supermarket, estimate future sales for each of these products.

(b) Given a database of information about your users, automatically group them into different market segments.

(c) Given the database of mails, identifying them as spam or ham mails.

(d) Given historical weather records, predict if tomorrow's weather will be sunny or rainy.

✓ **Ans. : (b)**

Q. 4.18 K-means is an iterative algorithm, and one of the following steps is repeatedly carried out in its inner-loop. Which one?

(a) Select the number of clusters.

(b) Test on the cross-validation set

(c) Update the cluster centroids based the current assignment

(d) Using the elbow method to choose K.

✓ **Ans. : (c)**

Q. 4.19 _____ clustering algorithm terminates when mean values computed for the current iteration of the algorithm are identical to the computed mean values for the previous iteration Select one:

(a) K-Means clustering

(b) conceptual clustering

(c) expectation maximization

(d) agglomerative clustering ✓ **Ans. : (a)**

Q. 4.20 Find odd man out Select one:

(a) DBSCAN (b) K mean

(c) PAM (d) K medoid ✓ **Ans. : (a)**

Q. 4.21 Which statement is true about the K-Means algorithm?

(a) The output attribute must be categorical.

(b) All attribute values must be categorical.

(c) All attributes must be numeric

(d) Attribute values may be either categorical or numeric ✓ **Ans. : (c)**

Q. 4.22 Clustering is _____ and is example of _____learning.

(a) Predictive and supervised

(b) Predictive and unsupervised

(c) Descriptive and supervised

(d) Descriptive and unsupervised ✓ **Ans. : (d)**

Q. 4.23 A good clustering method will produce high quality clusters with _____.

(a) high inter class similarity

(b) low intra class similarity

(c) high intra class similarity

(d) no inter class similarity ✓ **Ans. : (c)**

Q. 4.24 What is the final resultant cluster size in Divisive algorithm, which is one of the hierarchical clustering approaches?

(a) Zero (b) Three (c) singleton (d) Two

✓ **Ans. : (c)**

Q. 4.25 What does K refers in the K-Means algorithm which is a non-hierarchical clustering approach?

(a) Complexity (b) Fixed value

(c) Number of iterations (d) Number of clusters

✓ **Ans. : (d)**

Descriptive Questions

Q. 1 Define cluster analysis. Give the applications of cluster analysis.

Q. 2 Differentiate between classification and clustering.

Q. 3 Explain different types of data used in clustering.

Q. 4 Explain k-Means algorithm in detail with suitable example.

Q. 5 Use K-means algorithm to create 3 - clusters for given set of values :

{2, 3, 6, 8, 9, 12, 15, 18, 22} **(MU - June 2021)**

Q. 6 Confer the K-means algorithm with the following data for two clusters. Data Set {10, 4, 2, 12, 3, 20, 30, 11, 25, 31}

Q. 7 Suppose that the data mining task is to cluster the following eight points (with (x; y) representing location) into three clusters.

A1(2,10); A2(2,5); A3(8,4); A4(5,8); A5(7,5); A6(6,4); A7(1,2); A8(4,9). The distance function is Euclidean distance. Suppose initially we assign A1, A4, and A7 as the center of each cluster, respectively. Use the k-means algorithm to show only

(a) The three cluster centers after the first round of execution and

(b) The final three clusters **(MU - May-2019)**

Q. 8 Differentiate between simple-linkage, average-linkage and complete linkage algorithms. Use complete linkage algorithm to find the clusters from the following dataset. **(MU - May 2019)**

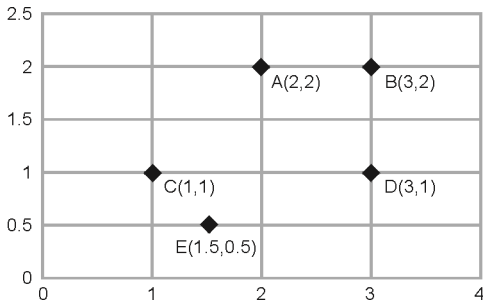
X	4	8	15	24	24
Y	4	4	8	4	12

Q. 9 Show the dendrogram created by the complete link clustering algorithm for the given set of points.

(MU - Dec. 2019)

Points	A	B
P1	2	4
P2	8	2
P3	9	3
P4	1	5
P5	8.5	1

Q. 10 Use the data given below. Create adjacency matrix. Use complete link algorithm to cluster given data set. Draw dendrogram. **(MU - June 2021)**



(1D43)**Fig. Q. 10**

Chapter Ends…

❑❑❑

MODULE 5

CHAPTER 5

Mining Frequent Patterns and Associations

University Prescribed Syllabus w.e.f Academic Year 2021-2022

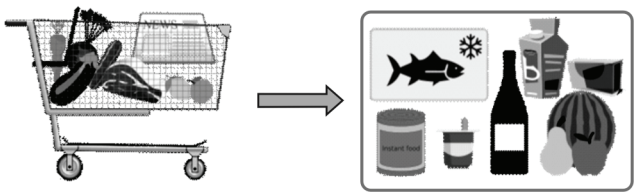
Market Basket Analysis, Frequent Itemsets, Closed Itemsets, and Association Rule, Frequent Pattern Mining, Apriori Algorithm, Association Rule Generation, Improving the Efficiency of Apriori, Mining Frequent Itemsets without candidate generation, Introduction to Mining Multilevel Association Rules and Mining Multidimensional Association Rules.

5.1 MARKET BASKET ANALYSIS

UQ. Elucidate Market Basket Analysis with an example.

MU - Dec. 2019

- Market Basket Analysis is a data mining technique that is used to uncover purchase patterns in any retail setting.
- The goal of Market Basket Analysis is to understand consumer behavior by identifying relationships between the items that people buy.
- It which identifies the strength of association between pairs of products purchased together and identify patterns of co-occurrence. A co-occurrence is when two or more things take place together.
- Market Basket Analysis creates *If-Then* scenario rules, for example, if item A is purchased then item B is likely to be purchased.
- For example, people who buy green tea are also likely to buy honey. So Market Basket Analysis would quantitatively establish that there is a relationship between Green Tea and Honey. The same goes for bread, butter, and jam.



(1E1)**Fig. 5.1.1 : Market Basket Analysis**

- The rules are probabilistic in nature or, in other words, they are derived from the frequencies of co-occurrence in the observations. Frequency is the proportion of baskets that contain the items of interest.
- The technique determines relationships of what products were purchased with which other product(s). These relationships are then used to build profiles containing If-Then rules of the items purchased.
- The rules could be written as: If {A} Then {B}
- The *If* part of the rule (the {A} above) is known as the antecedent and the *THEN* part of the rule is known as the consequent (the {B} above).
- The antecedent is the condition and the consequent is the result.
- The association rule has three measures that express the degree of confidence in the rule. They are Support, Confidence, and Lift.
- The **support** is the number of transactions that include items in the {A} and {B} parts of the rule as a percentage of the total number of transactions. It is a measure of how frequently the collection of items occur together as a percentage of all transactions. Also called the **occurrence frequency**, **frequency**, **support count**, or **count**.

$$\text{Support } (A \rightarrow B) = \frac{\text{Number of transactions containing both A and B}}{\text{Total number of transactions}}$$

- The **confidence** of the rule is the ratio of the number of transactions that include all items in {B} as well as the number of transactions that include all items in {A} to the number of transactions that include all items in {A}.

$$\text{Confidence } (A \rightarrow B) = \frac{\text{Number of transactions containing both A and B}}{\text{Number of transactions containing A}}$$

- The **lift or lift ratio** is the ratio of confidence to expected confidence. Expected confidence is the confidence divided by the frequency of B. The Lift tells us how much better a rule is at predicting the result than just assuming the result in the first place. Greater lift values indicate stronger associations.

$$\text{Lift } (A \rightarrow B) = \frac{\dfrac{\text{Number of transactions containing both A and B}}{\text{Number of transactions containing A}}}{\dfrac{\text{Number of transactions containing B}}{\text{Total number of transactions}}}$$

- **Example :** Consider there are nine baskets containing varying combinations of milk, cheese, apples, and bananas.

Basket	Product 1	Product 2	Product 3
1	Milk	Cheese	
2	Milk	Apples	Cheese
3	Apples	Banana	
4	Milk	Cheese	
5	Apples	Banana	
6	Milk	Cheese	Banana
7	Milk	Cheese	
8	Cheese	Banana	
9	Cheese	Milk	

$$\text{Support (Milk} \rightarrow \text{Cheese)} = \frac{\text{Number of baskets containing both Milk and Cheese}}{\text{Total number of baskets}}$$

$$= \frac{6}{9} = 0.67$$

$$\text{Confidence (Milk} \rightarrow \text{Cheese)} = \frac{\text{Number of baskets containing both Milk and Cheese}}{\text{Number of baskets containing Milk}}$$

$$= \frac{6}{6} = 1.00$$

$$\text{Lift (A} \rightarrow \text{B)} = \frac{\frac{\text{Number of baskets containing both Milk and Cheese}}{\text{Number of baskets containing Milk}}}{\frac{\text{Number of baskets containing Cheese}}{\text{Total number of baskets}}}$$

$$= \frac{\frac{6}{6}}{\frac{7}{9}} = 1.29$$

5.1.1 Applications of Market Basket Analysis

- **Retail :** In Retail, Market Basket Analysis can help determine what items are purchased together, purchased sequentially, and purchased by season. This can assist retailers to determine product placement and promotion optimization (for instance, combining product incentives). Does it make sense to sell soda and chips or soda and crackers?
- **Telecommunications :** In Telecommunications, where high churn rates continue to be a growing concern, Market Basket Analysis can be used to determine what services are being utilized and what packages customers are purchasing. They can use that knowledge to direct marketing efforts at customers who are more likely to follow the same path. For instance, Telecommunications these days is also offering TV and Internet. Creating bundles for purchases can be determined from an analysis of what customers' purchase, thereby giving the company an idea of how to price the bundles. This analysis might also lead to determining the capacity requirements.
- **Banks :** In Financial (banking for instance), Market Basket Analysis can be used to analyze credit card purchases of customers to build profiles for fraud detection purposes and cross-selling opportunities.
- **Insurance :** In Insurance, Market Basket Analysis can be used to build profiles to detect medical insurance claim fraud. By building profiles of claims, you are able to then use the profiles to determine if more than 1 claim belongs to a particular claimer within a specified period of time.
- **Medical :** In Healthcare or Medical, Market Basket Analysis can be used for comorbid conditions and symptom analysis, with which a profile of illness can be better identified. It can also be used to reveal biologically relevant associations between different genes or between environmental effects and gene expression.

5.2 FREQUENT ITEMSETS, CLOSED ITEMSETS AND ASSOCIATION RULE

5.2.1 Frequent ItemSets

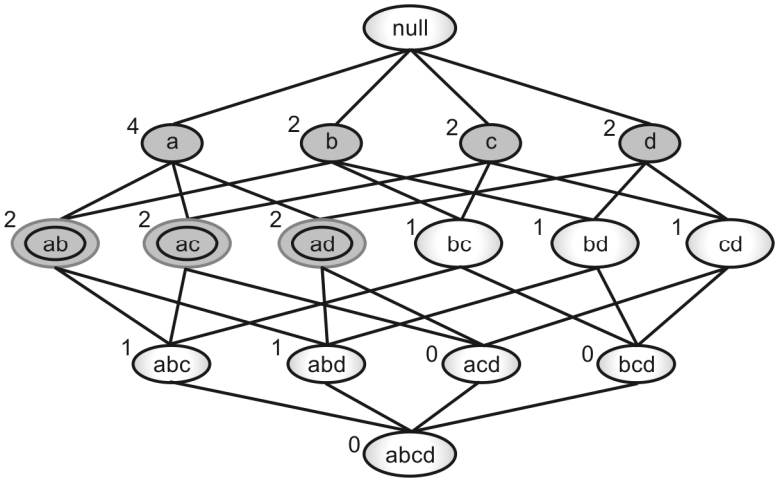
- A set of items together is called an **itemset**.
- If any itemset has k-items it is called a **k-itemset**.
- The set {Milk, Cheese} is a 2-itemset.
- An itemset consists of two or more items.
- The occurrence frequency of an itemset is the number of transactions that contain the itemset.
- An itemset that occurs frequently is called a frequent itemset.
- A set of items is called **frequent** if it satisfies a minimum threshold value for support and confidence.

5.2.2 Closed Itemsets

- An itemset is **closed** if none of its immediate supersets have support count same as Itemset.
- An itemset is **closed frequent** itemset if it is both closed and frequent.
- **Identification**
  1. First identify all frequent itemsets.
  2. Then from this group find those that are closed by checking to see if there exists a superset that has the same support as the frequent itemset, if there is, the itemset is disqualified, but if none can be found, the itemset is closed.
- **Maximal frequent** itemsets are the sets S such that no proper superset of S is frequent.

- To illustrate this concept, consider the example given below :

TID	Items
1	a,b,d
2	a,c
3	a,b,c
4	a,d



(1E2)Fig. 5.2.1: Lattice Structure for Closed, Maximal and Frequent Itemsets

- The support counts are shown on the top left of each node.
- Assume **support count threshold = 50%**, that is, each item must occur in 2 or more transactions.
- Based on that threshold, the frequent itemsets are: a, b, c, d, ab, ac and ad (shaded nodes).
- Out of these 7 frequent itemsets, 3 are identified as maximal frequent (having double circles):
  i. ab: Immediate supersets abc and abd are infrequent.
  ii. ac: Immediate supersets abc and acd are infrequent.
  iii. ad: Immediate supersets abd and acd are infrequent.
- The remaining 4 frequent nodes (a, b, c and d) cannot be maximal frequent because they all have at least 1 immediate superset that is frequent.

5.2.3 Association Rules

- An implication expression of the form X → Y, where X and Y are any 2 itemsets.
- Example: {Milk, Diaper} →{Beer}
- These rules must satisfy the confidence value.

5.3 FREQUENT PATTERN MATCHING

Frequent pattern mining is classified in the various ways based on following criteria:

1. Completeness of the pattern to be mined

- We can mine the complete set of frequent itemsets, the closed frequent itemsets, and the maximal frequent itemsets, given a minimum support threshold.
- We can also mine constrained frequent itemsets, approximate frequent itemsets, near-match frequent itemsets, top-k frequent itemsets and so on.

2. Levels of abstraction involved in the rule set

- Some methods for association rule mining can find rules at differing levels of abstraction.
- For example, suppose that a set of association rules mined includes the following rules where X is a variable representing a customer:

  buys(X, "computer")) → buys(X, "Canon Printer") ...(1)

  buys(X, "laptop computer")) → buys(X, "Canon Printer") ...(2)
- In rule (1) and (2), the items bought are referenced at different levels of abstraction (e.g., "computer" is a higher-level abstraction of "laptop computer").

3. Number of data dimensions involved in the rule

- If the items or attributes in an association rule reference only one dimension, then it is a **single-dimensional association rule.**

  buys(X, "computer")) → buys(X, "Canon printer")
- If a rule references two or more dimensions, such as age, income, and buys, then it is a **multidimensional**

association rule. The following rule is an example of a multidimensional rule:age(X, "30,31…39") ^ income(X, "42K,…48K")) → buys(X, "Apple Smartphone")

4. Types of valued handled in the rule

- If a rule involves associations between the presence or absence of items, it is a Boolean association rule.
- If a rule describes associations between quantitative items or attributes, then it is a quantitative association rule.

5. Kinds of rules to be mined

- Frequent pattern analysis can generate various kinds of rules and other interesting relationships.
- Association rule mining can generate a large number of rules, many of which are redundant or do not indicate a correlation relationship among itemsets.
- The discovered associations can be further analyzed to uncover statistical correlations, leading to correlation rules.

6. Kinds of patterns to be mined

- Many kinds of frequent patterns can be mined from different kinds of data sets.
- **Sequential pattern mining** searches for frequent subsequences in a sequence data set, where a sequence records an ordering of events.
- For example, with sequential pattern mining, we can study the order in which items are frequently purchased. For instance, customers may tend to first buy a laptop, followed by a smartphone, and then a smartwatch.
- **Structured pattern mining** searches for frequent substructures in a structured data set.
- Single items are the simplest form of structure. Each element of an itemset may contain a subsequence, a subtree, and so on.
- Therefore, structured pattern mining can be considered as the most general form of frequent pattern mining.

7. Application domain-specific semantics

- Because of the huge diversity in data and applications, the patterns to be mined can differ largely based on their domain-specific semantics.
- Application data can be of different types like spatial data, temporal data, spatiotemporal data, multimedia data, text data, time-series data, DNA and biological sequences, software programs, web structures, sensor network data, social and information networks data, and so on.

8. Data analysis usages

- For improved data understanding, pattern-based classification and pattern-based clustering can be used for semantic annotation or contextual analysis.
- Pattern analysis can be also used in recommender systems, which recommend items that are likely to be of interest to the user based on user's patterns.

5.4 APRIORI ALGORITHM

- Apriori algorithm was the first algorithm that was proposed for frequent itemset mining.
- It was later improved by R Agarwal and R Srikant and came to be known as Apriori.
- It is an iterative approach to discover the most frequent itemsets.
- This algorithm uses two steps "join" and "prune" to reduce the search space.
- **Join Step :** This step generates (K+1) itemset from K-itemsets by joining each item with itself.
- **Prune Step :** This step scans the count of each item in the database. If the candidate item does not meet minimum support, then it is regarded as infrequent and thus it is removed. This step is performed to reduce the size of the candidate itemsets.

5.4.1 Steps in Apriori Algorithm

Apriori algorithm is a sequence of steps to be followed to find the most frequent itemset in the given database. This data mining technique follows the join and the prune steps iteratively until the most frequent itemset is achieved. A minimum support threshold is given in the problem or it is assumed by the user.

1. In the first iteration of the algorithm, each item is taken as a 1-itemsets candidate. The algorithm will count the occurrences of each item.
2. Let there be some minimum support, min_sup (e.g. 2). The set of 1–itemsets whose occurrence is satisfying the min_supare determined. Only those candidates

which count more than or equal to min_sup, are taken ahead for the next iteration and the others are pruned.

3. Next, 2-itemset frequent items with min_supare discovered. For this in the join step, the 2-itemset is generated by forming a group of 2 by combining items with itself.
4. The 2-itemset candidates are pruned using min-sup threshold value. Now the table will have 2–itemsets with min-sup only.
5. The next iteration will form 3 –itemsets using join and prune step. This iteration will follow antimonotone property where the subsets of 3-itemsets, that is the 2 – itemset subsets of each group fall in min_sup. If all 2-itemset subsets are frequent then the superset will be frequent otherwise it is pruned.
6. Next step will follow making 4-itemset by joining 3-itemset with itself and pruning if its subset does not meet the min_sup criteria. The algorithm is stopped when the most frequent itemset is achieved.

5.4.2 Apriori Algorithm given by Jiawei Han, Micheline Kamber and Jian Pei

Algorithm: Apriori. Find frequent itemsets using an iterative level-wise approach based on candidate generation.

Input :

D: a database of transactions;

min_sup : the minimum support count threshold.

Output : L : frequent itemsets in D.

Method :

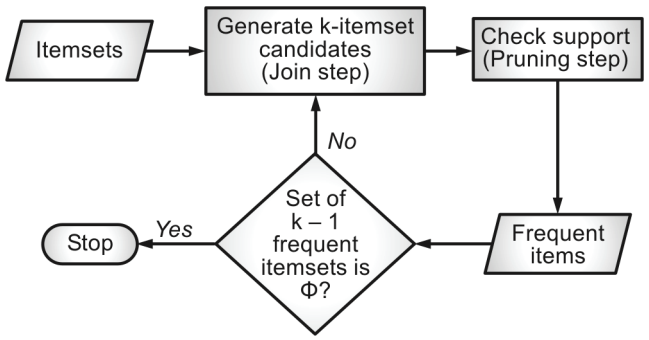
(1) L_1 = find_frequent_1-itemsets(D);
(2) for (k = 2; $L_{k-1} \neq \phi$; k++){
(3) C_k = apriori_gen (L_{k-1});
(4) for each transaction t ∈ D { // scan D for counts
(5) C_t = subset (C_k, t); // get the subsets of t that are candidates
(6) for each candidate c∈ C_t
(7) c.count ++ ;
(8) }
(9) $L_k = \{c \in C_k \mid c.count \geq min\_sup\}$
(10) }
(11) return $L = U_k L_k$;

procedure apriori_gen (L_{k-1}:frequent (k – 1) - itemsets)

(1) for each itemset $l_1 \in L_{k-1}$
(2) for each itemset $l_2 \in L_{k-1}$
(3) if ((l_1 [1] = l_2 [1]) ∧ (l_1 [2] = l_2 [2]) ∧…∧ (l_1 [k – 2] = l_2 [k – 2]) ∧ (l_1 [k – 1] < l_2 [k – 1]) then {
(4) c = $l_1 \bowtie l_2$; // join step: generate candidates
(5) if has_infrequent_subset(c, L_{k-1}) then
(6) delete c; // prune step: remove unfruitful candidate
(7) else add c to C_k
(8) }
(9) return Ck;

procedure has_infrequent_subset(c: candidate k-itemset; L_{k-1}: frequent (k – 1) –itemsets); // use prior knowledge
(1) for each (k – 1)-subset s of c
(2) if s ∉ L_{k-1} then
(3) return TRUE;
(4) return FALSE;

5.4.3 Flowchart for Apriori Algorithm



(1E3)**Fig. 5.4.1: Apriori Algorithm Flowchart**

5.4.4 Advantages

1. Easy to understand algorithm.
2. Join and Prune steps are easy to implement on large itemsets in large databases.

5.4.5 Disadvantages

1. It requires high computation if the itemsets are very large and the minimum support is kept very low.
2. The entire database needs to be scanned.

Ex. 5.4.1 : Given the following data, apply the apriori algorithm. Given **Support threshold=50%, Confidence= 60%.**

Transaction	List of items
T1	I1, I2, I3
T2	I2, I3, I4
T3	I4, I5
T4	I1, I2, I4
T5	I1, I2, I3, I5
T6	I1, I2, I3, I4

☑ **Soln. :**

Support threshold = 50%

Therefore, min_sup = 0.5 × number of transactions

= 0.5 × 6 = 3

Thus, min_sup = 3

1. **Count of Each Itemset (C_1) by scanning the database.**

Itemset	Count
{I1}	4
{I2}	5
{I3}	4
{I4}	4
{I5}	2

2. **Prune Step (L_1) : C_1** shows that I5 itemset does not meet min_sup=3, thus it is deleted, only I1, I2, I3, I4 meet min_sup count.

Itemset	Count
{I1}	4
{I2}	5
{I3}	4
{I4}	4

3. **Join Step :** Form C_2 from L_1 using $L_1 \bowtie L_1$ and find out their occurrences.

Itemset	Count
{I1,I2}	4
{I1,I3}	3
{I1,I4}	2
{I2,I3}	4
{I2,I4}	3
{I3,I4}	2

4. **Prune Step (L_2) : C_2** shows that itemset {I1, I4} and {I3, I4} does not meet min_sup, thus it is deleted.

Itemset	Count
{I1,I2}	4
{I1,I3}	3
{I2,I3}	4
{I2,I4}	3

5. **Join Step :** Form C_3 from L_2 using $L_2 \bowtie L_2$ and find out their occurrences.

Itemset	Count
{I1,I2, I3}	3
{I1,I2, I4}	2
{I1,I3, I4}	1
{I2,I3, I4}	2

6. **Prune Step (L_3) : C_3** shows that itemset {I1, I2, I4}, {I1, I3, I4} and {I2, I3, I4} does not meet min_sup, thus it is deleted.

Itemset	Count
{I1,I2, I3}	3

Thus {I1, I2, I3} is frequent.

7. **Generate Association Rules:** From the frequent itemset discovered above, the association could be:

- {I1, I2} → {I3}

  Confidence = support {I1, I2, I3} / support {I1, I2} = (3/ 4) × 100 = 75%
- {I1, I3} → {I2}

  Confidence = support {I1, I2, I3} / support {I1, I3} = (3/ 3) × 100 = 100%
- {I2, I3} → {I1}

  Confidence = support {I1, I2, I3} / support {I2, I3} = (3/ 4) × 100 = 75%
- {I1} → {I2, I3}

  Confidence = support {I1, I2, I3} / support {I1} = (3/ 4) × 100 = 75%
- {I2} → {I1, I3}

  Confidence = support {I1, I2, I3} / support {I2} = (3/ 5) × 100 = 60%
- {I3} → {I1, I2}

  Confidence = support {I1, I2, I3} / support {I3} = (3/ 4) × 100 = 75%

- This shows that all the above association rules are strong if minimum confidence threshold is 60%.

Ex. 5.4.2 : A database has four transactions. Let min_sup = 60%, min conf = 80%. Apply Apriori algorithm to find the frequent itemsets and the strong association rules.

Transaction	Date	List of items
T100	10/15/99	{K,A,D,B}
T200	10/15/99	{D,A,C,E,B}
T300	10/19/99	{C,A,B,E}
T400	10/22/99	{B,A,D}

☑ **Soln. :**

Support threshold = 60%

Therefore, min_sup = 0.6 × number of transactions = $0.6 \times 4 = 2.4$

Thus, min_sup = 3

1. **Count of Each Itemset (C_1) by scanning the database.**

Itemset	Count
{A}	4
{B}	4
{C}	2
{D}	3
{E}	2
{K}	1

2. **Prune Step (L_1) : C_1** shows that C, E, K item does not meet min_sup=3, thus it is deleted, only A, B, D meet min_sup count.

Itemset	Count
{A}	4
{B}	4
{D}	3

3. **Join Step :** Form C_2 from L_1 using $L_1 \bowtie L_1$ and find out their occurrences.

Itemset	Count
{A,B}	4
{A,D}	3
{B,D}	3

4. **Prune Step (L_2): C_2** shows that all item meet support count, so nothing is deleted.

Itemset	Count
{A,B}	4
{A,D}	3
{B,D}	3

5. **Join Step:** Form C_3 from L_2 using $L_2 \bowtie L_2$ and find out their occurrences.

Itemset	Count
{A, B, D}	3

6. **Prune Step (L_3) : C_3** shows that {A, B, D} meet min_sup.

Itemset	Count
{A, B, D}	3

Thus {A, B, D} is frequent.

7. **Generate Association Rules :** From the frequent itemset discovered above, the association could be:

- {A,B} → {D}

  Confidence = support {A, B, D} / support {A, B}
  = (3/ 4) × 100 = 75%

- {A,D} → {B}

  Confidence = support {A, B, D} / support {A, D}
  = (3/ 3) × 100 = 100%

- {B,D} → {A}

  Confidence = support {A, B, D} / support {B, D}
  = (3/ 3) × 100 = 100%

- {A} → {B,D}

  Confidence = support {A, B, D} / support {A}
  = (3/ 4) × 100 = 75%

- {B} → {A,D}

  Confidence = support {A, B, D} / support {B}
  = (3/ 4) × 100 = 75%

- {D} → {A,B}

  Confidence = support {A,B,D} / support {D}
  = (3/ 3) × 100 = 100%

- This shows that association rules {A,D} → {B}, {B,D} → {A} and {D} → {A,B} are strong as they satisfy minimum confidence threshold of 80%.

Ex. 5.4.3 : Consider the transaction data given below. Use Apriori Algorithm with min_sup count = 2 and min_confidence = 70% to find all frequent itemsets and strong association rules.

TID	List of Items
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1.I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

☑ **Soln. :**

Given : min_sup = 2

1. **Count of Each Itemset (C_1) by scanning the database.**

Itemset	Count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

2. **Prune Step (L_1) : C_1** shows that I4 itemset does not meet min_sup=3, thus it is deleted, only I1, I2, I3, I5 meet min_sup count.

Itemset	Count
{I1}	6
{I2}	7
{I3}	6
{I4]	2
{I5}	2

3. **Join Step :** Form C_2 from L_1 using $L_1 \bowtie L_1$ and find out their occurrences.

Itemset	Count
{I1,I2}	4
{I1,I3}	4
{I1,I4}	1
{I1,I5}	2
{I2,I3}	4
{I2,I4}	2
{I2,I5}	2
{I3,I4]	0
{I3,I5}	1
{I4,I5}	0

4. **Prune Step (L_2) : C_2** shows that itemset {I1, I4}, {I3, I4}, {I3, I5} and {I4, I5} does not meet min_sup, thus it is deleted.

Itemset	Count
{I1,I2}	4
{I1,I3}	4
{I1,I5}	2
{I2,I3}	4
{I2,I4}	2
{I2,I5}	2

5. **Join Step:** Form C_3 from L_2 using $L_2 \bowtie L_2$ and find out their occurrences.

Itemset	Count
{I1,I2,I3}	2
{I1,I2,I5}	2
{I1,I3,I5}	1
{I2,I3,I4}	0
{I2,I3,I5}	1
{I2,I4,I5}	0

6. **Prune Step (L_3) : C_3** shows that itemset{I1,I3,I5}, {I2,I3,I4}, {I2, I3, I5} and {I2,I4,I5}does not meet min_sup, thus it is deleted.

Itemset	Count
{I1,I2,I3}	2
{I1,I2,I5}	2

7. **Join Step:** Form C_4 from L_3 using $L_3 \bowtie L_3$ and find out their occurrences.

Itemset	Count
{I1,I2,I3,I5}	1

{I1,I2,I3,I5} does not meet min_sup, thus it is deleted. So we move back to step 6 and find that **{I1, I2, I3} and**
{I1, I2, I5} is frequent.

8. **Generate Association Rules :** From the frequent itemset discovered above, the association could be:

- {I1, I2} → {I3}

  Confidence = support {I1, I2, I3} / support {I1, I2}
  = 2/4 × 100 = 50%
- {I1, I3} → {I2}

  Confidence = support {I1, I2, I3} / support {I1, I3}
  = 2/4 × 100 = 50%
- {I2, I3} → {I1}

  Confidence = support {I1, I2, I3} / support {I2, I3}
  = 2/4 × 100 = 50%
- {I1, I2} → {I5}

  Confidence = support {I1, I2, I5} / support {I1, I2}
  = 2/4 × 100 = 50%
- {I1, I5} → {I2}

  Confidence = support {I1, I2, I5} / support {I1, I5}
  = 2/2 × 100 = 100%
- {I2, I5} → {I1}

  Confidence = support {I1, I2, I5} / support {I2, I5}
  = 2/2 × 100 = 100%
- {I3} → {I1, I2}

  Confidence = support {I1, I2, I3} / support {I1}
  = 2/6 × 100 = 33.33%
- {I2} → {I1, I3}

  Confidence = support {I1, I2, I3} / support {I2}
  = 2/7 × 100 = 28.57%
- {I1} → {I2, I3}

  Confidence = support {I1, I2, I3} / support {I1}
  = 2/6 × 100 = 33.33%
- {I5} → {I1, I2}

  Confidence = support {I1, I2, I5} / support {I5}
  = 2/2 × 100 = 100%
- {I2} → {I1, I5}

  Confidence = support {I1, I2, I5} / support {I2}
  = 2/7 × 100 = 28.57%
- {I1} → {I2, I5}

  Confidence = support {I1, I2, I5} / support {I1}
  = 2/6 × 100 = 33.33%
- This shows that association rules {I1, I5}→{I2}, {I2, I5}→{I1}, and {I5}→{I1, I2}are strong as they satisfy minimum confidence threshold of 70%.

UEx. 5.4.4 (MU - Dec. 2019)

Consider the transaction database given below :

TID	Items
10	1, 3, 4
20	2, 3, 5
30	1, 2, 3, 5
40	2, 5
50	1, 3, 5

Use Apriori Algorithm with min-support count = 2 and min-confidence = 60% to find all frequent itemsets and strong association rules.

☑ **Soln. :**

Given : min_sup = 2

1. **Count of Each Itemset (C_1) by scanning the database.**

Itemset	Count
{I1}	3
{I2}	3
{I3}	4
{I4}	1
{I5}	4

2. **Prune Step (L_1) :** C_1 shows that I4 itemset does not meet min_sup = 3, thus it is deleted, only I1, I2, I3, I5 meet min_sup count.

Itemset	Count
{I1}	3
{I2}	3
{I3}	4
{I5}	4

3. **Join Step:** Form C_2 from L_1 using $L_1 \bowtie L_1$ and find out their occurrences.

Itemset	Count
{I1,I2}	1
{I1,I3}	3
{I1,I5}	2
{I2,I3}	2
{I2,I5}	3
{I3,I5}	3

4. **Prune Step (L_2) : C_2** shows that itemset {I1, I2} does not meet min_sup, thus it is deleted.

Itemset	Count
{I1,I3}	3
{I1,I5}	2
{I2,I3}	2
{I2,I5}	3
{I3,I5}	3

5. **Join Step:** Form C_3 from L_2 using $L_2 \bowtie L_2$ and find out their occurrences.

Itemset	Count
{I1,I2, I3}	1
{I1,I2, I5}	1
{I1,I3, I5}	2
{I2,I3, I5}	1

6. **Prune Step (L_3) : C_3** shows that itemset {I1, I2, I3}, {I1, I2, I5} and {I2, I3, I5} does not meet min_sup, thus it is deleted.

Itemset	Count
{I1,I3, I5}	2

Thus {I1, I3, I5} is frequent.

7. **Generate Association Rules :** From the frequent itemset discovered above, the association could be:

- {I1, I3} → {I5}

  Confidence = support {I1, I3, I5} / support {I1, I3}
  = (2/ 3) × 100 = 67%

- {I1, I5} → {I3}

  Confidence = support {I1, I3, I5} / support {I1, I5}
  = (2/ 2) × 100 = 100%

- {I3, I5} → {I1}

  Confidence = support {I1, I3, I5} / support {I3, I5}
  = (2/ 3) × 100 = 67%

- {I5} → {I1, I3}

  Confidence = support {I1, I3, I5} / support {I5}
  = (2/ 4) × 100 = 50%

- {I3} → {I1, I5}

  Confidence = support {I1, I3, I5} / support {I3}
  = (2/ 4) × 100 = 50%

- {I1} → {I3, I5}

  Confidence = support {I1, I3, I5} / support {I1}
  = (2/ 3) × 100 = 67%

- This shows that association rules {I1, I3}→{I5}, {I1, I5}→{I3}, {I3, I5}→{I1} and {I1}→{I3, I5}are strong as they satisfy minimum confidence threshold of 60%.

5.5 ASSOCIATION RULE GENERATION

- Once the frequent itemsets from transactions in the database D are found, we can generate strong association rules from them.
- Strong association rules satisfy both minimum support and minimum confidence.

$$\text{Confidence } (A \rightarrow B) = \frac{\text{Support\_count } (A \cup B)}{\text{Support\_count } (A)}$$

5.6 IMPROVING EFFICIENCY OF APRIORI

Many variations of Apriori algorithm have been proposed that focus on improving the efficiency of the original algorithm. Several of these variations are summarized as follows:

1. **Hash-Based Technique :** This method uses a hash-based structure called a hash table for generating the k-itemsets and its corresponding count. It uses a hash function for generating the table.
2. **Transaction Reduction :** This method reduces the number of transactions scanning in iterations. The transactions which do not contain frequent items are marked or removed.
3. **Partitioning :** This method requires only two database scans to mine the frequent itemsets. It says that for any itemset to be potentially frequent in the database, it should be frequent in at least one of the partitions of the database.
4. **Sampling :** This method picks a random sample S from Database D and then searches for frequent itemset in S. It may be possible to lose a global frequent itemset. This can be reduced by lowering the min_sup.
5. **Dynamic Itemset Counting :** This technique can add new candidate itemsets at any marked start point of the database during the scanning of the database.

5.7 MINING FREQUENT ITEMSETS WITHOUT CANDIDATE KEY GENERATION (FREQUENT PATTERN GROWTH OR FP-GROWTH)

- This algorithm is an improvement to the Apriori method.
- A frequent pattern is generated without the need for candidate generation.
- FP growth algorithm represents the database in the form of a tree called a frequent pattern tree or FP tree.
- This tree structure will maintain the association between the itemsets.
- The database is fragmented using one frequent item. This fragmented part is called "pattern fragment". The itemsets of these fragmented patterns are analyzed.
- Thus with this method, the search for frequent itemsets is reduced comparatively.

5.7.1 FP Tree

- Frequent Pattern Tree is a tree-like structure that is made with the initial itemsets of the database.
- The purpose of the FP tree is to mine the most frequent pattern.
- Each node of the FP tree represents an item of the itemset.
- The root node represents null while the lower nodes represent the itemsets.
- The association of the nodes with the lower nodes i.e. the itemsets with the other itemsets are maintained while forming the tree.

5.7.2 Steps in FP Growth Algorithm

1. The first step is to scan the database to find the occurrences of the itemsets in the database. This step is the same as the first step of Apriori. The count of 1-itemsets in the database is called support count or frequency of 1-itemset.
2. The second step is to construct the FP tree. For this, create the root of the tree. The root is represented by null.
3. The next step is to scan the database again and examine the transactions. Examine the first transaction and find out the itemset in it. The itemset with the max count is taken at the top, the next itemset with lower count and so on. It means that the branch of the tree is constructed with transaction itemsets in descending order of count.
4. The next transaction in the database is examined. The itemsets are ordered in descending order of count. If any itemset of this transaction is already present in another branch (for example in the 1st transaction), then this transaction branch would share a common prefix to the root. This means that the common itemset is linked to the new node of another itemset in this transaction.
5. Also, the count of the itemset is incremented as it occurs in the transactions. Both the common node and new node count is increased by 1 as they are created and linked according to transactions.
6. The next step is to mine the created FP Tree. For this, the lowest node is examined first along with the links of the lowest nodes. The lowest node represents the frequency pattern length 1. From this, traverse the path in the FP Tree. This path or paths are called a conditional pattern base. Conditional pattern base is a sub-database consisting of prefix paths in the FP tree occurring with the lowest node (suffix).
7. Construct a Conditional FP Tree, which is formed by a count of itemsets in the path. The itemsets meeting the threshold support are considered in the Conditional FP Tree.
8. Frequent Patterns are generated from the Conditional FP Tree.

5.7.3 FP-Growth Algorithm by Jiawei Han, Micheline Kamber and Jian Pei

Algorithm

FP growth. Mine frequent itemsets using an FP-tree by pattern fragment growth.

Input

- D, a transaction database;
- min_sup, the minimum support count threshold.

Output

The complete set of frequent patterns.

☞ **Method**

1. The FP-tree is constructed in the following steps:

(a) Scan the transaction database D once. Collect F, the set of frequent items, and their support counts. Sort F in support count descending order as L, the list of frequent items.

(b) Create the root of an FP-tree, and label it as "null." For each transaction Trans in D do the following.

- Select and sort the frequent items in Trans according to the order of L. Let the sorted frequent item list in Trans be [p|P], where p is the first element and P is the remaining list.
- Call **insert_tree([p|P], T)**, which is performed as follows.
- If T has a child N such that N.item-name = p.item-name, then increment N's count by 1; else create a new node N, and let its count be 1, its parent link be linked to T, and its node-link to the nodes with the same item-name via the node-link structure.
- If P is nonempty, call **insert_tree(P, N)** recursively.

2. The FP-tree is mined by calling **FP_growth(FP tree, null)**, which is implemented as follows.

procedure FP_growth (Tree, α)

(1) **if** Tree contains a single path P then
(2) **for each** combination (denoted as β) of the nodes in the path P
(3) generate pattern $\beta \cup \alpha$ with support_count = minimum support count of nodes in β;
(4) **else for each** a_i in the header of Tree{
(5) generate pattern $\beta = a_i \cup \alpha$ with support_count = a_i.support_count;
(6) construct β's conditional pattern base and then β's conditional FP_tree Treeβ;
(7) **if** $Tree_\beta = \phi$ **then**
(8) call FP_growth($Tree_\beta$, β);}

5.7.4 Advantages of FP Growth Algorithm

1. This algorithm needs to scan the database only twice when compared to Apriori which scans the transactions for each iteration.
2. The pairing of items is not done in this algorithm and this makes it faster.
3. The database is stored in a compact version in memory.
4. It is efficient and scalable for mining both long and short frequent patterns.

5.7.5 Disadvantages of FP Growth Algorithm

1. FP Tree is more cumbersome and difficult to build than Apriori.
2. It may be expensive.
3. When the database is large, the algorithm may not fit in the shared memory.

Ex. 5.7.1 : A database has five transactions. Let min_sup = 60% and min_conf = 80%. Find all the frequent itemsets using FP-growth.

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]

☑ **Soln. :**

Given : min_sup = 60%

∴ sup_count to be satisfied $= 5 \times 0.6 = 3$

▶ **Step 1:** Scan the database for count of each itemset.

Itemset	sup_count
{A}	1
{C}	2
{D}	1
{E}	4
{I}	1
{K}	5
{M}	3
{N}	2
{O}	4
{U}	1
{Y}	3

▶ **Step 2 :** Sort the set of frequent itemsets in the order of descending support count and denote that lists as L.

L :

Itemset	sup_count
{K}	5
{E}	4
{O}	4
{M}	3
{Y}	3

- **Step 3 :** Scan the database for second time and sort items in each transaction according to descending support count.

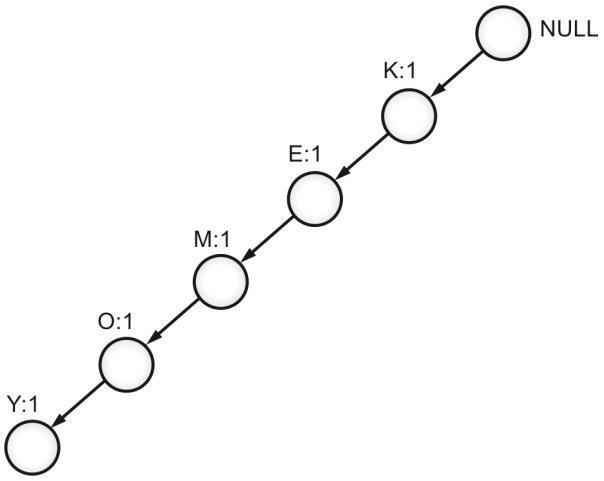
TID	List of Items
T100	{K,E,M,O,Y}
T200	{K,E,O,Y}
T300	{K,E,M}
T400	{K,M,Y}
T500	{K,E,O}

- **Step 4 :** Construct the FP-tree.

4.1 Create a root node with label "NULL".

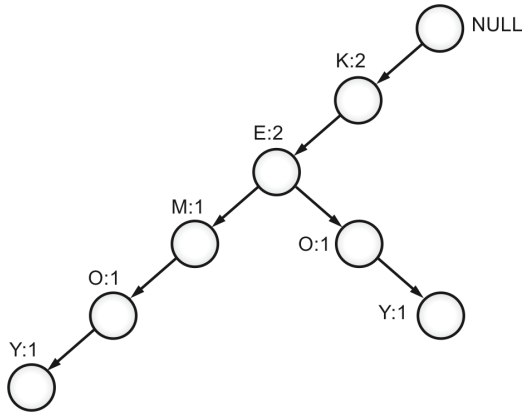
(1E4)**Fig. P. 5.7.1(a)**

4.2 Scan T100 and construct branch with nodes K:1, E:1, M:1, O:1, Y:1 linked to each other from root node.



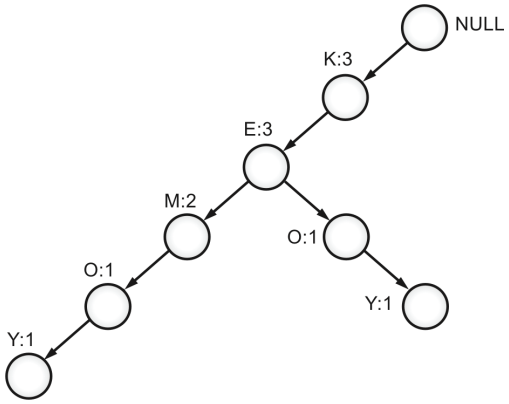
(1E5) **Fig. P. 5.7.1(b)**

4.3 Scan T200. It contains itemsets K, E, O, Y in L-order. Nodes K and E already exists. Increment their count as K:2, E: 2 and make a branch for O:1 and Y:1 from E:2.



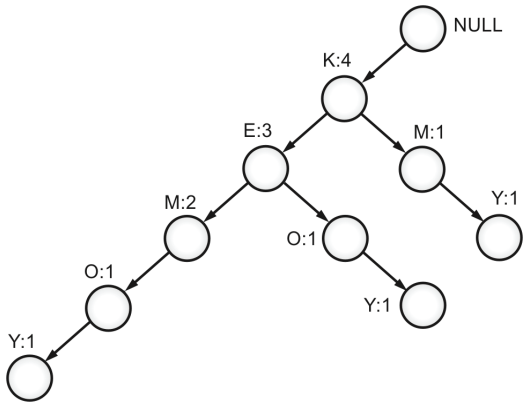
(1E6) **Fig. P. 5.7.1(c)**

4.4 Scan T300. It contains itemsets K, E, M in L-order. Branch with nodes K, E and M already exists. Increment their count as K:3, E:3 and M:2.



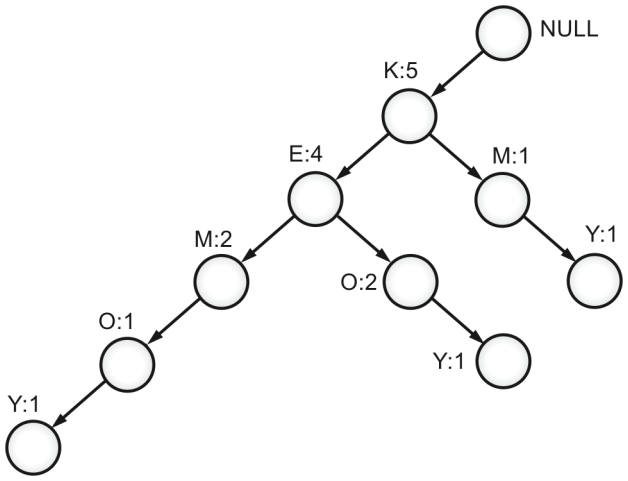
(1E7)**Fig. P. 5.7.1(d)**

4.5 Scan T400. It contains itemsets K, M, Y in L-order. Node K already exists. Increment its count by 1 as K:4 and make branch for M:1, Y:1 from K:4.



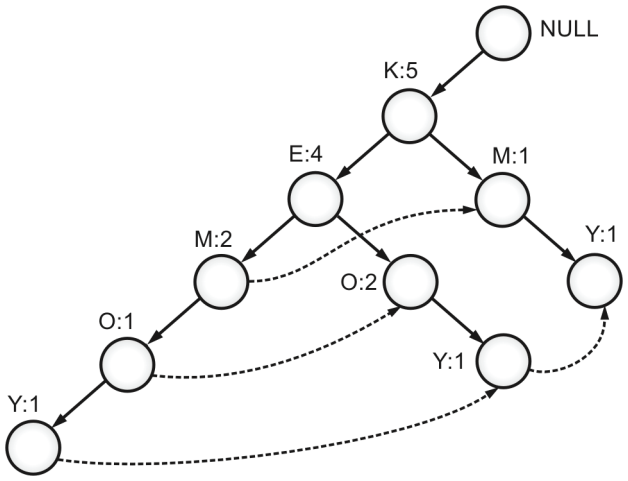
(1E8)**Fig. P. 5.7.1(e)**

4.6 Scan T500. It contains itemsets K, E, O in L-order. Branch with nodes K, E and O exists. Just increment their count.



(1E9)**Fig. P. 5.7.1(f)**

4.7 Now also connect all the similar nodes.



(1E10)**Fig. 5.7.1(g)**

▶ **Step 5 :** Mining FP-tree.

Start from each frequent length-1 pattern, construct its conditional pattern base, then construct its conditional FP-tree, and perform mining recursively on the tree. Start with the last itemset in L.

Note : For generating frequent patterns, consider the items which satisfy min_sup = 3 (given) criteria from conditional FP-tree.

Itemset	Conditional Pattern base	Conditional FP-tree	Frequent Patterns Generated
{Y}	{{K,E,M,O:1},{K,E,O:1}, {K,M:1}}	(K:3)	{K,Y:3}
{O}	{{K,E,M:1}, {K,E:2}}	(K:3,E:3)	{K,O:3}, {E,O:3}, {K,E,O:3}
{M}	{{K,E:2},{K:1}}	(K:3)	{K,M:3}
{E}	{{K:4}}	(K:4)	{K,E:4}
{K}	-	-	-

UEx. 5.7.2 (MU - May 2019)

Generate Frequent Pattern Tree for the following transaction with 30% minimum support.

Transaction ID	Items
T1	E,A,D,B
T2	D,A,C,E,B
T3	C,A,B,E
T4	B,A,D
T5	D
T6	D,B
T7	A,D,E
T8	B,C

☑ **Soln. :**

Given : min_sup = 30%

$\therefore$ sup_count to be satisfied $= 8 \times 0.3 = 2.4 \approx 3$.

▶ **Step 1 :** Scan the database for count of each itemset.

Itemset	sup_count
{A}	5
{B}	6
{C}	3
{D}	6
{E}	4

- **Step 2 :** Sort the set of frequent itemsets in the order of descending support count and denote that lists as L.

  L :

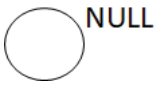
Itemset	Count
{B}	6
{D}	6
{A}	5
{E}	4
{C}	3

- **Step 3 :** Scan the database for second time and sort items in each transaction according to descending support count.

Transaction ID	Items
T1	B, D, A, E
T2	B, D, A, E, C
T3	B, A, E, C
T4	B, D, A
T5	D
T6	B, D
T7	D, A, E
T8	B, C

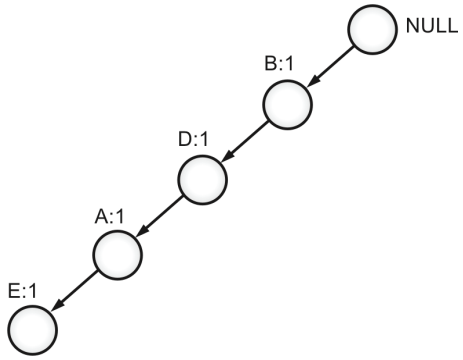
- **Step 4 :** Construct the FP-tree.

  4.1 Create the root of the tree, labelled with "NULL".



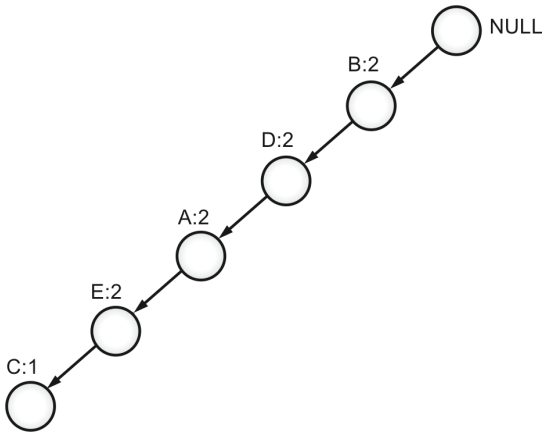
(1E11)**Fig. P. 5.7.2(a)**

  4.2 Scan T1 and construct branch with nodes B:1, D:1, A:1, E:1 linked to each other from the root node.



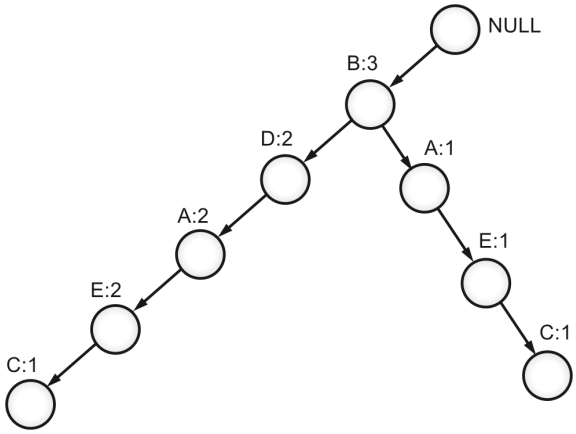
(1E12)**Fig. P. 5.7.2(b)**

  4.3 Scan T2. It contains itemsets B, D, A, E, C in L-order. The branch with nodes B, D, A, E already exists. Simply increment their count as B:2, D:2, A:2, E:2 and then C:1 to E:2.



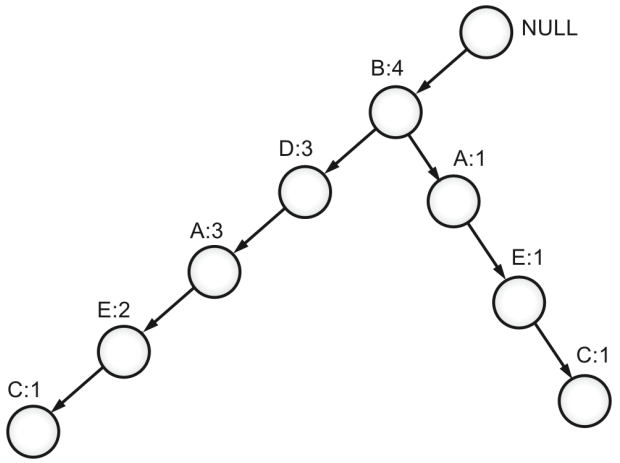
(1E13)**Fig. P. 5.7.2(c)**

  4.4 Scan T3. It contains itemsets B, A, E, C in L-order. Increment count of B by 1 as B:3 and connect A:1, E:1, C:1 to it.



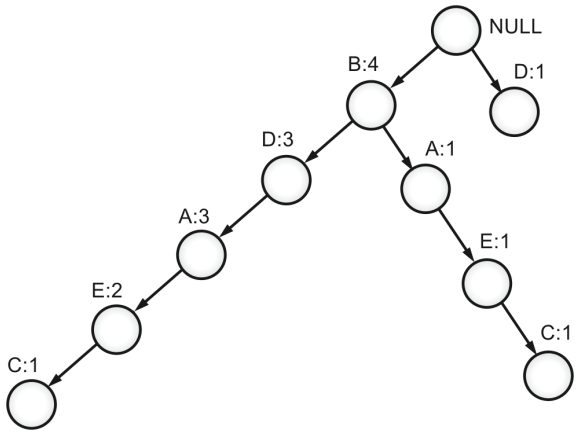
(1E14)**Fig. P. 5.7.2(d)**

4.5 Scan T4. It contains itemsets B, D, A in L-order. This branch already exists, so simply increment their count as B:4, D:3, A:3.



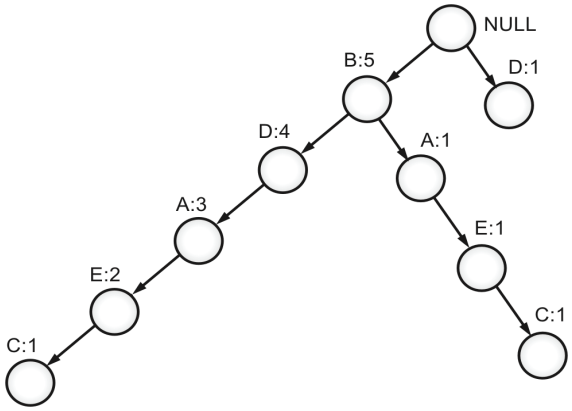
(1E15)**Fig. P. 5.7.2(e)**

4.6 Scan T5. It contains only itemset D. So we make a new branch from root node with count D:1.



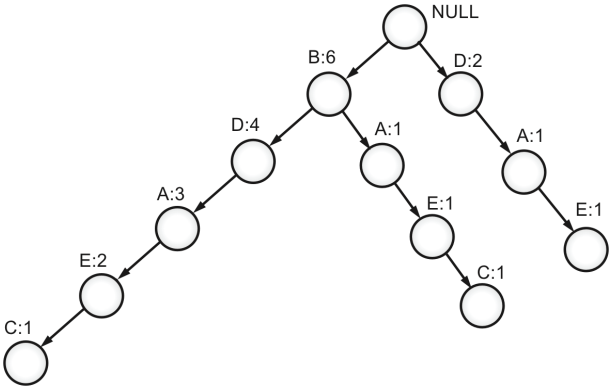
(1E16)**Fig. P. 5.7.2(f)**

4.7 Scan T6. It contains itemsets B and D in L-order. Branch B-D already exists. So simply increment the count by 1 as B:5 and D:4.



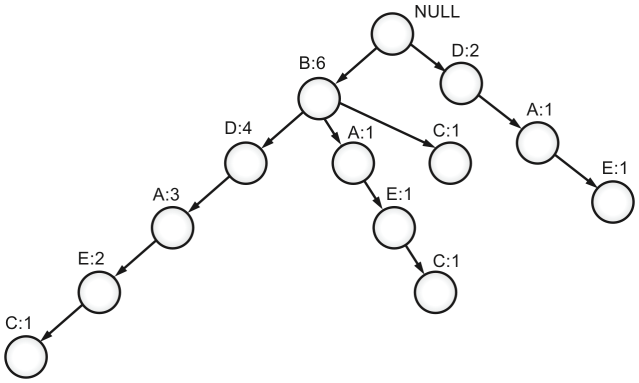
(1E17)**Fig. P. 5.7.2(g)**

4.8 Scan T7. It contains itemsets D, A, E in L-order. There is a branch with node D. Increment its count by 1 as D:2 and connect A:1, E:1 to this node.



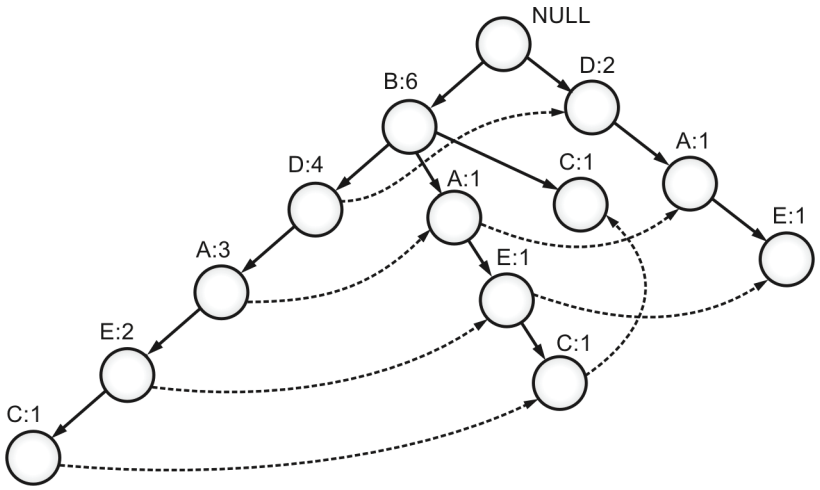
(1E18)**Fig. P. 5.7.2(h)**

4.9 Scan T8. It contains itemsets B, C in L-order. Node B already exists from root node. So increment its count by 1 as B:6 and connect node C:1 to it.



(1E19)**Fig. P. 5.7.2(i)**

4.10 Now, also connect the similar nodes.



(1E20)**Fig. P. 5.7.2(j)**

▶ **Step 5 :** Mining FP-tree.

Start from each frequent length-1 pattern, construct its conditional pattern base, then construct its conditional FP-tree, and perform mining recursively on the tree. Start with the last itemset in L.

Note : For generating frequent patterns, consider the items which satisfy min_sup = 3 (given) criteria from conditional FP-tree.

Itemset	Conditional Pattern base	Conditional FP-tree	Frequent Patterns Generated
{C}	{{B,D,A,E:1}, {B,A,E:1}, {B:1}}	(B:3)	{B,C:3}
{E}	{{B,D,A:2}, {B,A:1}}, {D,A:1}	(B:3, A:3)	{B,E:3}, {A,E:3}, {B,A,E:3}
{A}	{{B,D:3}, {B:1}},{D:1}	(B:4, D:3)	{B,A:4}, {D,A:3}, {B,D,A:3}
{D}	{B:4}	(B:4)	{B,D:4}
{B}	-	-	-

UEx. 5.7.3 (MU - June 2021)

A database has 9 transactions. Let min_sup = 2.

TID	List of Items
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1.I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Find all the frequent itemsets using FP-Growth algorithm.

☑ **Soln. :**

Given : min_sup = 2.

▶ **Step 1:** Scan the database for count of each itemset.

Itemset	Count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Here, each itemset satisfy the criteria of min_sup = 2.

▶ **Step 2 :** Sort the set of frequent itemsets in the order of descending support count and denote that lists as L.

L :

Itemset	Count
{I2}	7
{I1}	6
{I3}	6
{I4}	2
{I5}	2

- **Step 3 :** Scan the database for second time and sort items in each transaction according to descending support count.

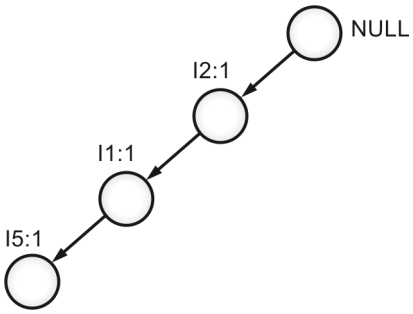
TID	List of Items
T100	I2,I1,I5
T200	I2,I4
T300	I2,I3
T400	I2,I1,I4
T500	I1.I3
T600	I2,I3
T700	I1,I3
T800	I2,I1,I3,I5
T900	I2,I1,I3

- **Step 4 :** Construct the FP-tree.

4.1 Create the root of the tree, labelled with "NULL".

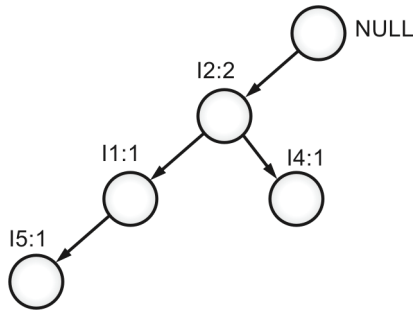
(1E21)**Fig. P. 5.7.3(a)**

4.2 Scan the first transaction T100 which contains three itemsets in L-order as I2, I1, I5. Construct the first branch of the tree with nodes I2:1, I1:1, and I5:1; where I2 is linked as a child to the root, I1 is linked to I2 and I5 is linked to I1.



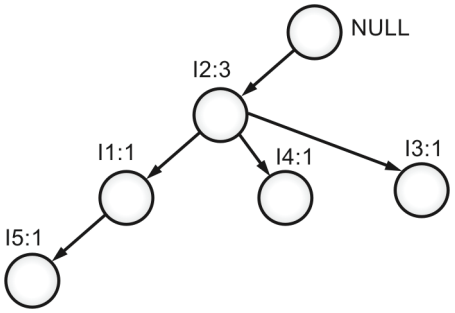
(1E22)**Fig. P. 5.7.3(b)**

4.3 Scan T200 which contains the itemsets I2 and I4 in L-order. This results in a branch where I2 is linked to the root and I4 is linked to I2. However, this branch would share a common prefix I2 with the existing path for T100. Therefore, we increment the count of I2 node by 1, and create a new node I4:1 which is linked as a child to I2:2.



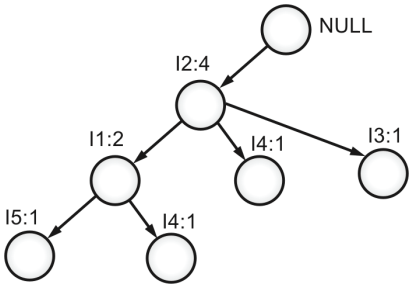
(1E23)**Fig. P. 5.7.3(c)**

4.4 Scan T300 which contains the itemsets I2 and I3 in L-order. This results in a branch where I2 is linked to the root and I3 is linked to I2. This branch would also share I2 as common prefix with the existing path for T200. Increment count of I2 node by 1, and create a new node I3:1 which is linked as a child to I2:3.



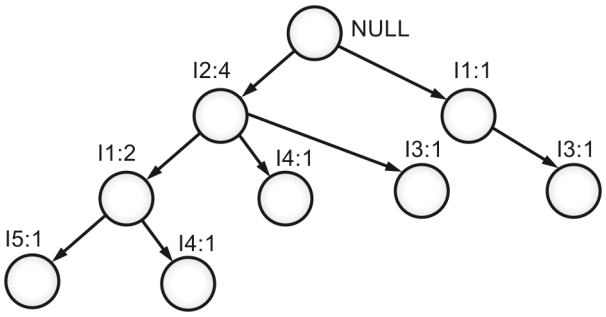
(1E24)**Fig. P. 5.7.3(d)**

4.5 Scan T400. It contains I2, I1, I4 in L-order. Branch I2-I1 exists from root node. Increment the count of nodes I2 and I1 by 1 and link I4:1 and I1:2.



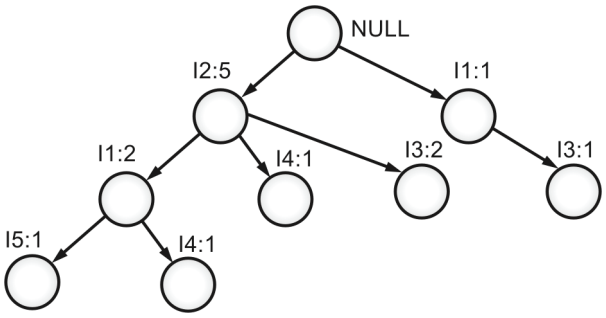
(1E25)**Fig. P. 5.7.3(e)**

4.6 Scan T500. It contains I1 and I3 in L-order. Here, I1 will be connected as child to the roots as I1:1 and I3:1 will be linked to I1:1. This results in a new branch from root node.



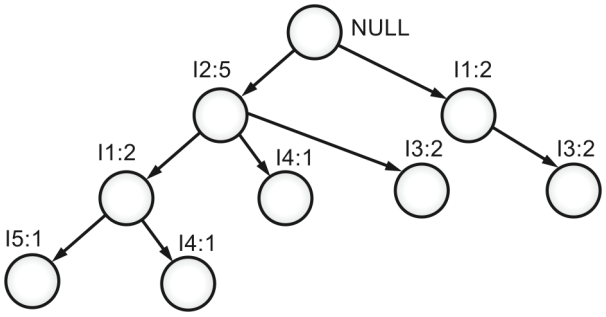
(1E26)**Fig. P. 5.7.3(f)**

4.7 Scan T600. It contains I2 and I3 in L-order. This branch with nodes I2 and I3 from root already exists. Simply increment the count of nodes.



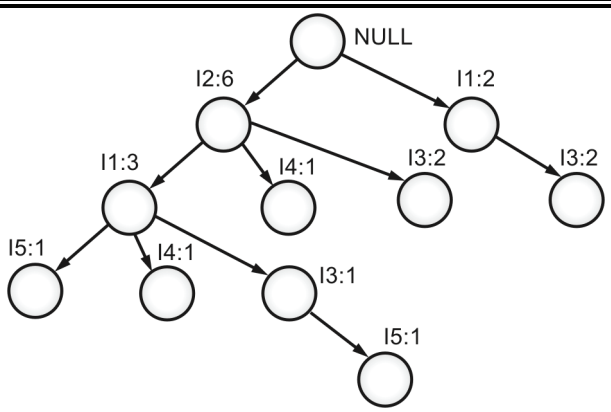
(1E27)**Fig. P. 5.7.3(g)**

4.8 Scan T700. It contains I1 and I3 in L-order. I1-I3 branch already exists from the root node. Hence, simply increment the count of I1 and I3.



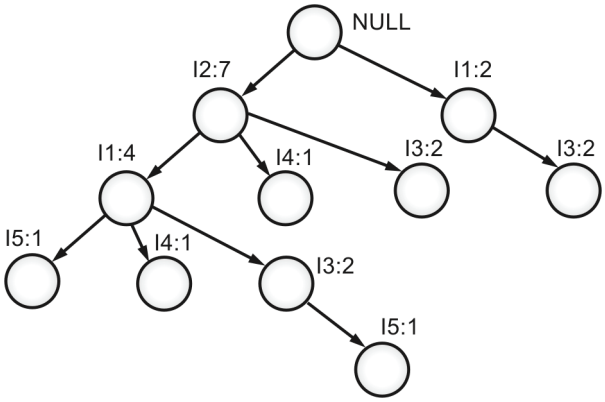
(1E28)**Fig. P. 5.7.3(h)**

4.9 Scan T800. It contains I2, I1, I3 and I5 in L-order. The branch I2-I1 exists from root node. Increment count of I2 and I1 respectively. This results in I2:6 and I1:3. Now link I3:1 to I1:3 and node I5:1 to I3:1.



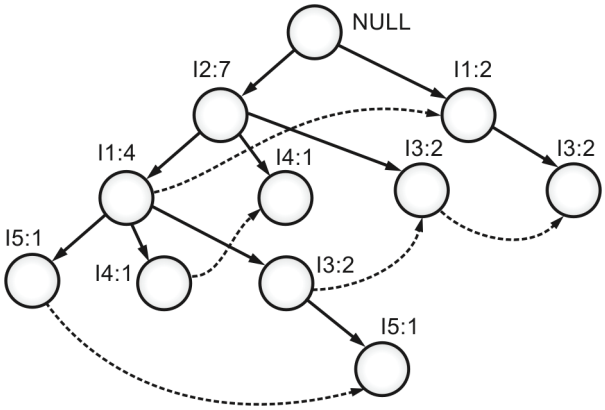
(1E29)**Fig. P. 5.7.3(i)**

4.10 Scan T900. It contains I2, I1 and I3 in L-order. The branch I2-I1-I3 from root node exists. Simply, increment the count of node by 1.



(1E30)**Fig. P. 5.7.3(j)**

4.11 Now also connect the similar nodes.



(1E31)**Fig. P. 5.7.3(k)**

- **Step 5 :** Mining FP-tree.

  Start from each frequent length-1 pattern, construct its conditional pattern base, then construct its conditional

FP-tree, and perform mining recursively on the tree. Start with the last itemset in L.

Note : For generating frequent patterns, consider the items which satisfy min_sup = 2 (given) criteria from conditional FP-tree.

Itemset	Conditional Pattern base	Conditional FP-tree	Frequent Patterns Generated
{I5}	{{I2, I1:1}, {I2, I1, I3:1}}	(I2:2, I1:2)	{I2,I5:2}, {I1,I5:2}, {I2,I1,I5:2}
{I4}	{{I2,I1:1}, {I2:1}}	(I2:2)	{I2,I4:2}
{I3}	{{I2,I1:2},{I2:2}, {I1:2}}	(I2,I1:4), (I1:2)	{I2,I3:4}, {I1,I3:4}, {I2,I1,I3:2}
{I1}	{{I2:4}}	(I2:4)	{I2,I1:4}
{I2}	-	-	-

5.7.6 Difference between Apriori Algorithm and FP-Growth Algorithm

Sr. No.	Apriori Algorithm	FP-Growth Algorithm
1	It is an array based algorithm.	It is a tree based algorithm.
2	It uses Join and Prune technique.	It constructs conditional frequent pattern tree and conditional pattern base from database which satisfy minimum support.
3	Apriori uses a breadth-first search.	FP Growth uses a depth-first search.
4	Apriori utilizes a level-wise approach where it generates patterns containing 1 item, then 2 items, then 3 items, and so on.	FP Growth utilizes a pattern-growth approach means that, it only considers patterns actually existing in the database.
5	Candidate generation is extremely slow. Runtime increases exponentially depending on the number of different items.	Runtime increases linearly, depending on the number of transactions and items.
6	Candidate generation is very parallelizable.	Data are very interdependent, each node needs the root.
7	It requires large memory space due to large number of candidate generation.	It requires less memory space due to compact structure and no candidate generation.
8	It scans the database multiple times for generating candidate sets.	It scans the database only twice for constructing frequent pattern tree.

5.8 MINING FREQUENT ITEMSETS USING VERTICAL DATA FORMATS

- There are usually two ways of representing transactional data, Horizontal Data Format and Vertical Data Format.
- In Horizontal Data Format, the transactional data is represented as TID-itemset where TID is the transaction id and itemset is the set of items bought in that particular transaction.
- Example of Horizontal Data Format:

Transaction	List of items
T1	A,B,C
T2	A,C
T3	A,D
T4	B,C

- In Vertical Data Format, transactional data is represented as item-TIDset. Item is the item name and TID set is the set of transactions containing that item.
- Example of Vertical Data Format

Items Bought	Transaction ID Set
A	T1,T2,T3
B	T1, T4
C	T1,T2,T4
D	T3

- 2-itemset in vertical data format

Items Bought	Transaction ID Set
A,B	T1
A,C	T1, T2
A,D	T3
B,C	T1, T4

- 3-itemset in vertical data format

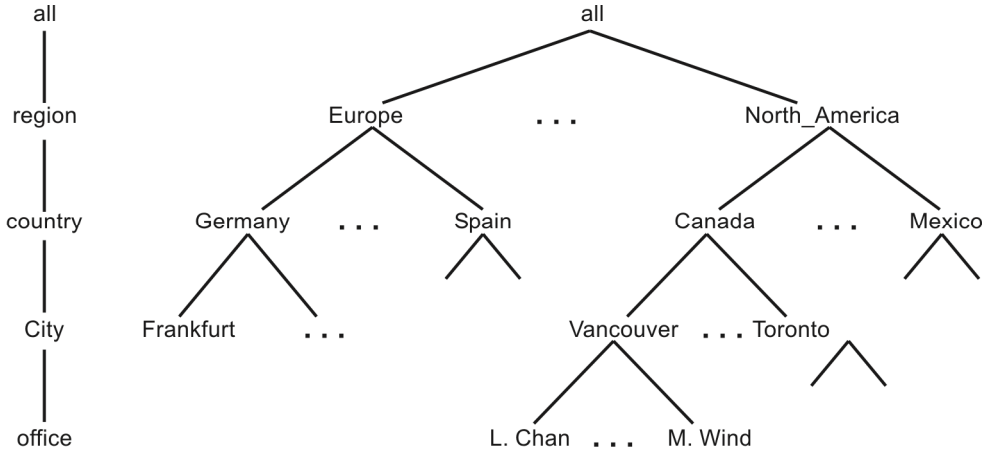
Items Bought	Transaction ID Set
A,B,C	T1

- Therefore, there is only one frequent 3-itemset {A,B,C}

5.9 INTRODUCTION TO MINING MULTILEVEL ASSOCIATION RULES

UQ. Demonstrate Multidimensional and Multilevel Association Rue Mining with suitable examples. **MU - Dec. 2019**

- Association rules generated from mining data at multiple levels of abstraction are called multiple-level or multilevel association rules.
- Multilevel association rules can be mined efficiently using concept hierarchies under a support-confidence framework.
- In general, a top-down approach is used, with counts accumulated for the computation of frequent itemsets at each concept level, starting at concept level 1 and continuing down the hierarchy toward more detailed concept levels until no more itemsets can be discovered.
- For each level, any algorithm for discovering frequent itemsets may be used, such as Apriori or its variations.



(1E32)**Fig. 5.9.1: Concept Hierarchy**

- There are different of variations to this approach, where each variation involves “playing" with support threshold in a slightly different way.

5.9.1 Support and Confidence of Multilevel Association Rules

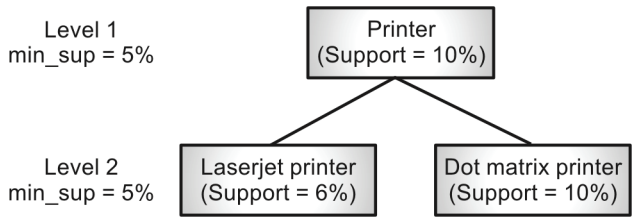
- Generalizing / specializing values of attributes affects support and confidence of an item.
- Support of rules increases from specialized to generalized itemsets.
- Support of rules decreases from generalized to specialized itemsets.
- Confidence is not affected for general or specialized.
- If the support is below the threshold value, then that rule becomes invalid.

5.9.2 Approaches of Multilevel Association Rules

1. Using uniform support level for all levels

- Consider the same minimum support for all levels of hierarchy.
- There is only one minimum support threshold, so no need to examine itemsets.

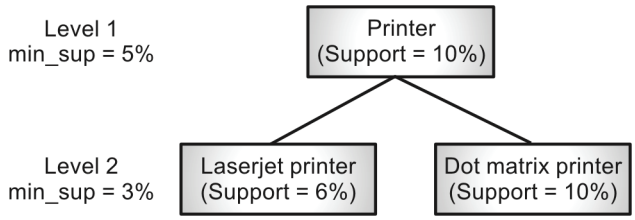
- If support threshold is too high, then low level associations may get missed.
- If the support threshold is too low, it may generate too many high level associations.



(1E33)**Fig. 5.9.2: Multilevel Mining with Uniform Support**

2. Using reduced minimum support at lower level

- Consider separate minimum support for all levels of hierarchy.
- At every level of abstraction, there is its own minimum support threshold; So minimum support at lower levels reduces.



(1E34)**Fig. 5.9.3: Multilevel Mining with Reduced Support**

- For mining multiple-level associations with reduced support, there are a number of alternative search strategies:
  - **(a) Level-by-Level independent**: This is a full-breadth search, where no background knowledge of frequent itemsets is used for pruning. Each node is examined, regardless of whether or not its parent node is found to be frequent.
  - **(b) Level -cross-filtering by single item**: An item at the i^{th} level is examined if and only if its parent node at the $(i - 1)^{th}$ level is frequent .In other words ,we investigate a more specific association from a more general one. If a node is frequent, its children will be examined; otherwise, its descendants are pruned from the search.
  - **(c) Level-cross filtering by k-itemset**: A k-itemset at the i^{th} level is examined if and only if its corresponding parent k-itemset at the $(i - 1)^{th}$ level is frequent.

3. Using item or group-based minimum support

- When mining multilevel rules, it's sometimes preferable to build up user-specific, item-based, or group-based minimal support criteria because users or experts sometimes have insight into which groups are more significant than others.
- For example, a user could establish minimal support criteria based on product pricing or items of interest, such as setting extremely low support thresholds for laptop computers and flash drives to focus on association patterns comprising products from these categories.

5.10 MINING MULTIDIMENSIONAL ASSOCIATION RULES

Following are the terminologies used in multidimensional database.

- **Single – dimension rules :** It contains the single distinct predicate like "buys" in the example given. buys(X, "milk") → buys (X, "bread")
- **Multi-dimensional rule :** It contains more than one predicate
  1. Inter-dimension association rule: It has no repeated predicate

     age (X,"19-25") ^ occupation (X, "student") → buys (X, "coke").
  2. Hybrid dimension association rules: It contains multiple occurrence of the same predicate like "buys" in the below example.

     age(X, "19-25") ^ buys (X, "popcorn") → buys (X, "coke")
- **Categorical Attributes :** This have finite number of possible values, no ordering among values. Example: brand, color.
- **Quantitative Attributes :** These are numeric and implicit ordering among values Example; age, income.

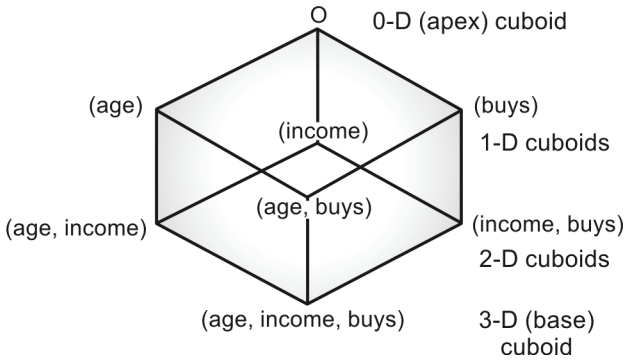
5.10.1 Techniques for Mining Multidimensional Associations

- Database attributes can be categorical or quantitative.

- Categorical attributes have a finite number of possible values, with no ordering among the values.
- Quantitative attributes are numeric and have an implicit ordering among values.
- Techniques for mining multidimensional association rules can be categorized into two basic approaches regarding the treatment of quantitative attributes:

(i) Static Discretization of Quantitative Attributes

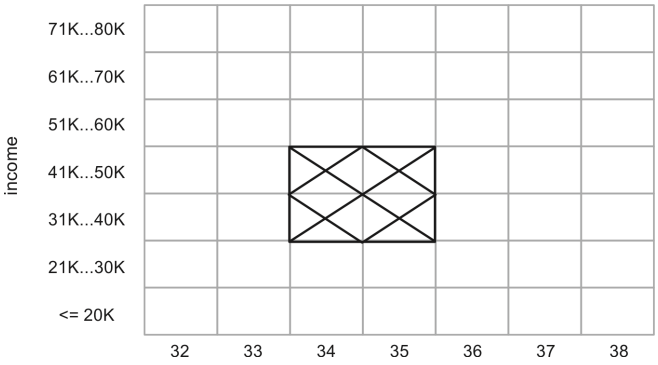
- Quantitative attributes are discretized using predefined concept hierarchies in this method. This discretization takes place prior to mining.
- For instance, a concept hierarchy for income may be used to replace the original numeric values of this attribute by interval labels, such as "0.....20K", "21K.....30K", "31K.....40K", and so on. Here, discretization is static and predetermined.
- The discretized numeric attributes, with their interval labels, can then be treated as categorical attributes (where each interval is considered a category).
- We refer to this as mining multidimensional association rules using static discretization of quantitative attributes.



(1E35)**Fig. 5.10.1 : Lattice of cuboids to form a 3-D data cube**

(ii) Dynamic Quantitative Association Rules

- In this approach, quantitative attributes are discretized or clustered into "bins" based on the distribution of the data.
- These bins may be further combined during the mining process.
- The discretization process is dynamic and set up to meet certain mining criteria, such as increasing the confidence in the rules mined.
- Association rules derived from this procedure are referred to as (dynamic) quantitative association rules since the numeric attribute values are treated as quantities rather than predetermined ranges or categories.
- The strong association rules obtained are mapped a 2-D grid as shown.



(1E36)**Fig. 5.10.2 : A 2-D grid for tuples representing customers purchasing SUV**

- Following four customers correspond to the rules:

  age(X, 34) ^ income(X,"30k-40K") → buys(X,"SUV")

  age(X, 35) ^ income(X,"30k-40K") → buys(X,"SUV")

  age(X, 34) ^ income(X,"40k-50K") → buys(X,"SUV")

  age(X, 35) ^ income(X,"40k-50K") → buys(X,"SUV")
- Above rules are close to each other, they can be clustered to form the following rule:

  age(X, "34-35") ^ income(X,"30k-50K") → buys(X,"SUV")

⏭ 5.11 MULTIPLE CHOICE QUESTIONS

Q. 5.1 A collection of one or more items is called as _____.

(a) Itemset (b) Support

(c) Confidence (d) Support Count ✓**Ans. : (a)**

Q. 5.2 Frequency of occurrence of an itemset is called as _____.

(a) Support (b) Confidence

(c) Support Count (d) Rules ✓**Ans. : (c)**

Q. 5.3 An itemset whose support is greater than or equal to a minimum support threshold is _____.

(a) Itemset (b) Frequent Itemset

(c) Infrequent Itemset (d) Threshold Values

✓**Ans. : (b)**

Q. 5.4 What does FP growth algorithm do?

(a) It mines all frequent patterns through pruning rules with lesser support.

(b) It mines all frequent patterns through pruning rules with higher support.

(c) It mines all frequent patterns by constructing a FP tree.

(d) It mines all frequent patterns by constructing an itemsets. ✓**Ans. : (c)**

Q. 5.5 What techniques can be used to improve the efficiency of Apriori algorithm?

(a) Hashing-based techniques

(b) Transaction Reduction

(c) Sampling

(d) Cleaning ✓**Ans. : (a)**

Q. 5.6 What do you mean by support(A)?

(a) Total number of transactions containing A

(b) Total Number of transactions not containing A

(c) Number of transactions containing A / Total number of transactions

(d) Number of transactions not containing A / Total number of transactions ✓**Ans. : (c)**

Q. 5.7 How do you calculate Confidence (A → B)?

(a) Support (A ∩ B) / Support (A)

(b) Support (A ∩ B) / Support (B)

(c) Support (A ∪ B) / Support (A)

(d) Support (A ∪ B) / Support (B) ✓**Ans. : (c)**

Q. 5.8 Which of the following is the direct application of frequent itemset mining?

(a) Social Network Analysis

(b) Market Basket Analysis

(c) Outlier Detection

(d) Intrusion Detection ✓**Ans. : (b)**

Q. 5.9 What is not true about FP growth algorithms?

(a) It mines frequent itemsets without candidate generation.

(b) There are chances that FP trees may not fit in the memory.

(c) FP trees are very expensive to build.

(d) It expands the original database to build FP trees. ✓**Ans. : (d)**

Q. 5.10 When do you consider an association rule interesting?

(a) If it only satisfies min_support.

(b) If it only satisfies min_confidence.

(c) If it satisfies both min_support and min_confidence.

(d) There are other measures to check so.

✓**Ans. : (c)**

Q. 5.11 What is the relation between a candidate and frequent itemsets ?

(a) A candidate itemset is always a frequent itemset.

(b) A frequent itemset must be a candidate itemset.

(c) No relation between these two.

(d) Strong relation with transactions ✓**Ans. : (b)**

Q. 5.12 Which of the following is not a frequent pattern mining algorithm?

(a) Apriori (b) FP Growth

(c) Decision Trees (d) ECLAT ✓**Ans. : (c)**

Q. 5.13 Which algorithm requires fewer scans of data?

(a) Apriori (b) FP Growth

(c) Naïve Bayesian (d) Decision Tree ✓**Ans. : (b)**

Q. 5.14 For the question given below consider the data Transactions:

1. I1, I2, I3, I4, I5, I6
2. I7, I2, I3, I4, I5, I6
3. I1, I8, I4, I5
4. I1, I9, I10, I4, I6
5. I10, I2, I4, I11, I5

With support as 0.6 find all frequent itemsets?

(a) <I1>, <I2>, <I4>, <I5>, <I6>, <I1, I4>, <I2, I4>, <I2, I5>, <I4, I5>, <I4, I6>, <I2, I4, I5>

(b) <I2>, <I4>, <I5>, <I2, I4>, <I2, I5>, <I4, I5>, <I2, I4, I5>

(c) <I1>, <I4>, <I5>, <I6>, <I1, I4>, <I5, I4>, <I1, I5>, <I4, I6>, <I2, I4, I5>

(d) <I1>, <I4>, <I5>, <I6> ✓**Ans. : (a)**

Q. 5.15 What will happen if support is reduced?

(a) Number of frequent itemsets remains the same.

(b) Some itemsets will add to the current set of frequent itemsets.

(c) Some itemsets will become infrequent while others will become frequent.

(d) Cannot say. ✓**Ans. : (b)**

Q. 5.16 What is association rule mining?

(a) Same as frequent itemset mining

(b) Finding of strong association rules using frequent itemsets

(c) Using association to analyze correlation rules

(d) Finding Itemsets for future trends ✓**Ans. : (b)**

Q. 5.17 The number of iterations in Apriori _____.

(a) increases with the size of the data

(b) decreases with the increase in size of the data

(c) increases with the size of the maximum frequent set

(d) decreases with increase in size of the maximum frequent set ✓**Ans. : (c)**

Q. 5.18 Which of the following are interestingness measures for association rules?

(a) Recall (b) Lift

(c) Accuracy (d) Compactness ✓**Ans. : (b)**

Q. 5.19 Which Association Rule would you prefer?

(a) High support and medium confidence

(b) High support and low confidence

(c) Low support and high confidence

(d) Low support and low confidence ✓**Ans. : (c)**

Q. 5.20 The Apriori property means

(a) If a set cannot pass a test, its supersets will also fail the same test.

(b) To decrease the efficiency, do level-wise generation of frequent itemsets.

(c) To improve the efficiency, do level-wise generation of frequent itemsets.

(d) If a set can pass a test, its supersets will fail the same test. ✓**Ans. : (a)**

Q. 5.21 If an itemset 'XYZ' is a frequent itemset, then all subsets of that frequent itemset are

(a) Undefined (b) Not frequent

(c) Frequent (d) Cannot say ✓**Ans. : (c)**

Q. 5.22 The _____ step eliminates the extensions of (k-1) itemsets which are not found to be frequent, from being considered for counting support

(a) Partitioning (b) Candidate generation

(c) Itemset eliminations

(d) Pruning ✓**Ans. : (d)**

Q. 5.23 To determine association rules from frequent itemsets

(a) Only minimum confidence needed

(b) Neither support not confidence needed

(c) Both minimum support and confidence are needed

(d) Minimum support is needed ✓**Ans. : (c)**

Q. 5.24 If {A,B,C,D} is a frequent itemset, candidate rules which is not possible is

(a) C→ A (b) D → ABCD

(c) A → BC (d) B → ADC ✓**Ans. : (b)**

Q. 5.25 Given the transaction in Table 1 and min_sup s = 50%, how many frequent 3-itemsets are there?

TID	Items Bought
10	Ball, Nuts, Pen
20	Ball, Coffee, Pen, Nuts
30	Ball, Pen, Eggs
40	Ball, Nuts, Eggs, Milk
50	Nuts, Coffee, Pen, Eggs, Milk

(a) 1 (b) 0 (c) 2 (d) 3 ✓**Ans. : (b)**

Q. 5.26 If we know the support of itemset {a, b} is 10, which of the following numbers are the possible supports of itemset {a, b, c}?

(a) 9 (b) 10 (c) 11 (d) 12 ✓**Ans. : (a, b)**

Q. 5.27 Choose which data mining task is suitable for the following scenario: first buy digital camera, then buy large SD memory cards.

(a) Classification (b) Sequential Pattern Analysis

(c) Association Rule (d) Prediction ✓**Ans. : (b)**

Q. 5.28 Choose which data mining task is the most suitable for the following scenario: To identify items that are bought concomitantly by a reasonable fraction of customers so that they can be shelved.

(a) Classification (b) Association Rules

(c) Clustering (d) Prediction ✓**Ans. : (b)**

Q. 5.29 Choose which data mining task is the most suitable for the following scenario : Given the records of books that a group of people read, find relationship of the genre pattern.

(a) Classification (b) Association Rules

(c) Clustering (d) Prediction ✓**Ans. : (b)**

Q. 5.30 What is the relation between candidate and frequent itemsets?

(a) A candidate itemset is always a frequent itemset

(b) A frequent itemset must be a candidate itemset

(c) No relation between the two

(d) Both are same ✓**Ans. : (b)**

Q. 5.31 Which technique finds the frequent itemsets in just two database scans?

(a) Partitioning (b) Sampling

(c) Hashing (d) Dynamic itemset counting

✓**Ans. : (a)**

Q. 5.32 A sub-database which consists of set of prefix paths in the FP-tree co-occurring with the suffix pattern is called as _____.

(a) Suffix path (b) FP-tree

(c) Prefix path (d) Conditional pattern base

✓**Ans. : (d)**

Q. 5.33 Consider the data transactions given below:

T1: {F,A,D,B} T2: {D,A,C,E,B}

T3: {C,A,B,E} T4: {B,A,D}

With minimum support = 60% and the minimum confidence = 80%, which of the following is not valid association rule?

(a) A→ B (b) B → A

(c) D → A (d) A → D ✓**Ans. : (d)**

Q. 5.34 The proportion of transaction supporting X in T is called _____

(a) confidence (b) support

(c) support count (d) lift ✓**Ans. : (b)**

Q. 5.35 The absolute number of transactions supporting X in T is called _____.

(a) confidence (b) support

(c) support count (d) lift ✓**Ans. : (c)**

Q. 5.36 The value that says that transactions in D that support X also support Y is called _____.

(a) confidence (b) support

(c) support count (d) lift ✓**Ans. : (a)**

Q. 5.37 If T consist of 500000 transactions, 20000 transaction contain bread, 30000 transaction contain jam, 10000 transaction contain both bread and jam. Then the support of bread and jam is _____.

(a) 2% (b) 20% (c) 3% (d) 30%

✓**Ans. : (a)**

Q. 5.38 If T consist of 500000 transactions, 20000 transaction contain bread, 30000 transaction contain jam, 10000 transaction contain both bread and jam. Then the confidence of buying bread with jam is _____.

(a) 33.33% (b) 66.66%

(c) 45% (d) 50% ✓**Ans. : (d)**

Q. 5.39 The left hand side of an association rule is called _____.

(a) consequent (b) onset

(c) antecedent (d) precedent ✓**Ans. : (c)**

Q. 5.40 The right hand side of an association rule is called _____.

(a) consequent (b) onset

(c) antecedent (d) precedent ✓**Ans. : (a)**

Descriptive Questions

Q. 1 Elucidate Market Basket Analysis with an example.

(MU - Dec. 2019)

Q. 2 Explain the terms: Frequent Itemsets, Closed Itemsets and Association Rule

Q. 3 Explain Apriori algorithm with its advantages and disadvantages.

Q. 4 Explain the techniques to improve efficiency of Apriori Mining.

Q. 5 Explain FP-Growth algorithm with its advantages and disadvantages.

Q. 6 Demonstrate Multidimensional and Multilevel Association Rule Mining with suitable examples.

(MU - Dec. 2019)

Q. 7 Consider the transaction database given below:

TID	Items
10	1, 3, 4
20	2, 3, 5
30	1, 2, 3, 5
40	2, 5
50	1, 3, 5

Use Apriori Algorithm with min-support count = 2 and min-confidence = 60% to find all frequent itemsets and strong association rules.

(MU - Dec. 2019)

Q. 8 Consider the following transactions. Apply the Apriori with minimum support of 30% and minimum confidence of 75% and find large itemset L.

TID	Items
01	1, 3, 4, 6
02	2, 3, 5, 7
03	1, 2, 3, 5, 8
04	2, 5, 9, 10
05	1, 4

Q. 9 Generate Frequent Pattern Tree for the following transaction with 30% minimum support.

(MU - May 2019)

Transaction ID	Items
T1	E, A, D, B
T2	D, A, C, E, B
T3	C, A, B, E
T4	B, A, D
T5	D
T6	D, B
T7	A, D, E
T8	B, C

Q. 10 A database has 9 transactions. Let min_sup = 2.

TID	List of Items
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1.I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Find all the frequent itemsets using FP-Growth algorithm. **(MU - June 2021)**

Chapter Ends…

❑❑❑

MODULE 6

CHAPTER 6

Web Mining

University Prescribed Syllabus w.e.f Academic Year 2021-2022

Introduction, Web Content Mining: Crawlers, Harvest System, Virtual Web View, Personalization, Web Structure Mining: Page Rank, Clever, Web Usage Mining.

6.1 INTRODUCTION

UQ. With respect to web mining, is it possible to detect visual objects using meta-objects ?
MU - May 2019

- Web mining is an application of data mining techniques to find information patterns from the web data. It is the process of using data mining techniques and algorithms to extract information directly from the Web by extracting it from Web documents and services, Web content, hyperlinks and server logs.
- Web data can be:
  a. Content of actual Web pages
  b. Intra page structure which includes the HTML or XML node for the page.
  c. Inter page structure which is the actual linkage structure between Web pages.
  d. Usage data that describe how Web pages are accessed by visitors.
  e. User profiles include demographic and registration information obtained about users.
- The contents of data mined from the Web may consist of text, structured data such as lists and tables, and even images, video and audio.
- The goal of Web mining is to look for patterns in Web data by collecting and analyzing information in order to gain insight into trends, the industry and users in general.

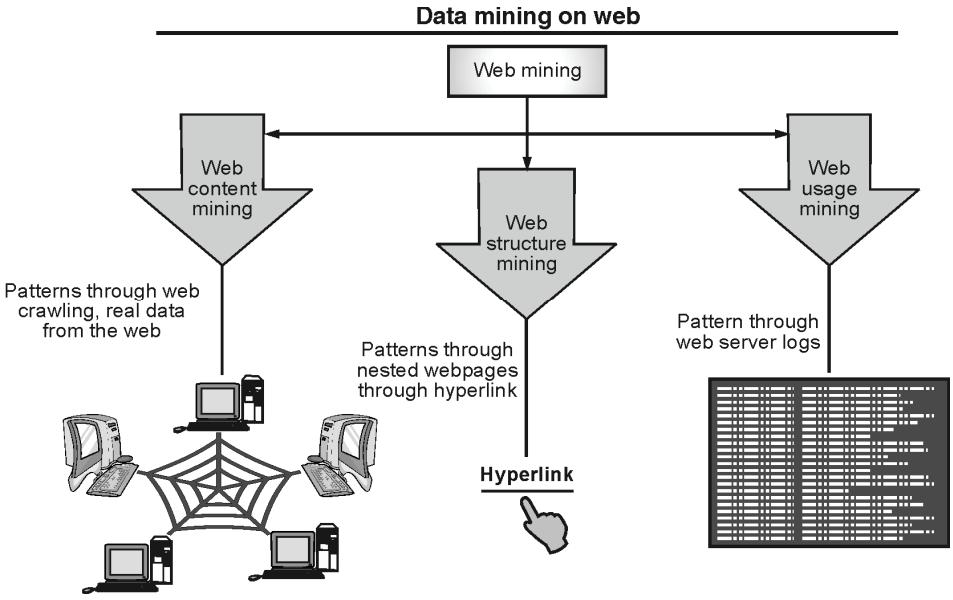
6.1.1 Applications of Web Mining

- Web mining helps to improve the power of web search engines such as Google, Yahoo, etc. by classifying the web documents and identifying the web pages.
- Web mining is used to predict user behavior.
- Web mining is very useful of a particular Website and e-service e.g., landing page optimization.
- Web mining is very useful to e-commerce websites and e-services.

6.2 TECHNIQUES OF WEB MINING

Web mining can be broadly divided into three different types as shown in Fig. 6.2.1.

1. **Web Content Mining** - used for mining of useful data, information and knowledge from web page content.
2. **Web Structure Mining** - helps to find useful knowledge or information pattern from the structure of hyperlinks.
3. **Web Usage Mining** - is used for mining the web log records (access information of web pages) and helps to discover the user access patterns of web pages.



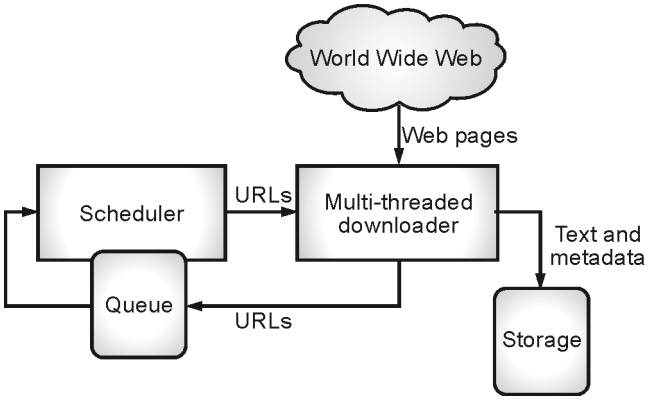
(1F1)**Fig. 6.2.1 : Categories of Web Mining**

6.3 WEB CONTENT MINING

- Web Content Mining is the process of mining useful information from the contents of Web pages'/Web documents.
- Web pages mostly contain text, images and audio/video files. Web content mining performs scanning and mining of the text, images and groups of web pages according to the content of the input (query), by displaying the list in search engines. For example, if a user wants to search for a particular book, then search engine provides the list of suggestions.
- It is related to text mining because much of the web contents are texts. However, it is also quite different from data mining because Web data are mainly semi-structured and/or unstructured, while data mining deals primarily with structured data.
- Web content mining is also different from text mining because of the semi-structure nature of the Web, while text mining focuses on unstructured texts.
- Data from the web pages are extracted in order to discover different patterns that give a significant insight.
- Techniques used in this discipline have been heavily drawn from natural language processing (NLP) and information retrieval.
- There are many techniques to extract the data like web scraping. **Scrapy** and **Octoparse** are the well-known tools that performs the web content mining process.

6.3.1 Crawlers

- Traditional search engines use **crawlers** to search the Web, to gather information, **indexing techniques** to store the information and **query processing** to provide fast and accurate information to users.
- **Web crawler** is a program that acts as an automated script which browses through the internet in a systematic way.
- Crawlers are primarily programmed for repetitive actions so that browsing is automated. Search engines use crawlers most frequently to browse the internet and build an index.
- Other crawlers search different types of information such as RSS feeds and email addresses.
- The term crawler comes from the first search engine on the Internet: "**The Web Crawler**" or "**Bot**" or "**Spider**."
- The web crawler is **keyword based**, it looks at the keywords in the pages, the kind of content each page has and the links, before returning the information to the search engine. This process is known as **Web crawling.**
- The page you need is indexed by a software known as web crawler. A web crawler gathers pages from the web and then, indexes them in a methodical and automated manner to support search engine queries.
- Crawlers would also help in validating HTML codes and checking links.
- These web crawlers go by different names, like bots, automatic indexers and robots. Once you type a search query, these crawlers scan all the relevant pages that contain these words and turn it into a huge index.
- For example, if you are using Google's search engine, then the crawlers would go through each of the pages indexed in their database and fetch those pages to Google's servers. The web crawler follows all the hyperlinks in the websites and visits other websites as well.
- So, when you ask the search engine for a 'web mining ', it will come up with all the web pages that feature the term. Web crawlers are configured to monitor the web regularly so the results they generate are updated and timely. Some of the popular web crawlers are Googlebot, Scrapy (the Python Scraper), Storm-crawler, Elasticsearch River Web, etc.



(1F2)**Fig. 6.3.1 : Web Crawling**

6.3.1(A) Working of Web Crawlers

- The spider begins its crawl by going through the websites or list of websites that it visited the previously. When the crawlers visit a website, they search for other pages that are worth visiting.
- Web crawlers can link to new sites, note changes to existing sites and mark dead links.

Google Search - How it works ?

- In the World Wide Web, there are trillions and trillions of pages. Web Crawlers crawl through these pages to bring back the results demanded by customers. Site owners can decide which of their pages they want the web crawlers to index, and they can block the pages that need not be indexed.
- The indexing is done by sorting the pages and looking at the quality of the content and other factors. Google then generates algorithms to get a better view of what you are searching for, and provides a **number of features** that make your search more effective, such as:
  - **(a) Spelling** : In case there is an error in the word you typed, Google comes up with several alternatives to help you get on track.
  - **(b) Google Instant** : Instant results as you type.
  - **(c) Search methods** : Different options for searching, other than just typing out the words. This includes images and voice search.
  - **(d) Synonyms** : Tackles similar worded meanings and produces results.
  - **(e) Auto complete** : Anticipates what you need from what you type.
  - **(f) Query understanding** : An in-depth understanding of what you type.

6.3.1(B) Types of Crawlers

1. **Periodic crawlers :** A traditional crawler, in order to refresh its collection, periodically replaces the old documents with the newly downloaded documents. As it is activated periodically; every time it is activated it replaces the existing index.
2. **Incremental crawler :** This crawler incrementally refreshes the existing collection of pages by visiting them frequently and updates the index incrementally instead of replacing it.
3. **Focused crawler :** This web crawler tries to download the web pages that are related to each other i.e. it visits pages related to topics of interest. This is also known as Topic crawler.

6.3.1(C) Applications

- The classic goal of a crawler is to create an index. Thus, crawlers are the basis for the work of search engines.
- Price comparison portals search for information on specific products on the Web, so that prices or data can be compared accurately using crawlers.
- In the area of data mining, a crawler may collect publicly available e-mail or postal addresses of companies.
- Web analysis tools use crawlers or spiders to collect data for page views, or incoming or outbound links.
- Crawlers serve to provide information hubs with data, for example, news sites.

6.3.2 Harvest System

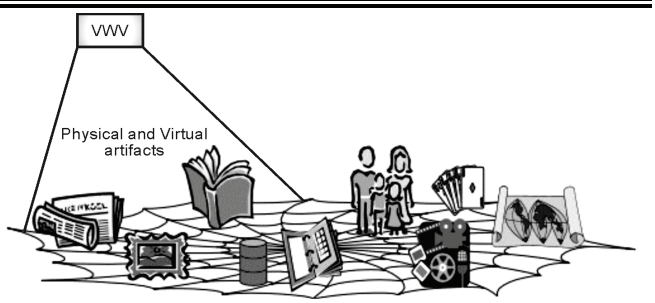
- **Data harvesting** is similar to data mining, but one of the key differences is that data harvesting uses a process that extracts and analyzes data collected from online sources.
- The Harvest system is based on the use of caching, indexing, and crawling. Harvest is actually a set of tools that facilitate gathering of information from diverse sources.
- For data harvesting, a website is targeted, and the data from that site is extracted. That data can be anything the harvester wants. It might be simple text found on the page or within the page's code. It could be directory information from a retail site. It might even be a series of images and videos. Or it could be all of those items at once.
- Data harvesting can be very beneficial, especially when using a third-party service. The data gathered from websites can provide organizations with helpful

information and insights that can inform their business practices and help them reach out to prospective consumers.

- The Harvest design is centered around the use of gatherers and brokers. A gatherer obtains information for indexing from an Internet service provider, while a broker provides the index and query interface.
- The relationship between brokers and gatherers can vary. Brokers may interface directly with gatherers or may go through other brokers to get to the gatherers. Indices in Harvest are topic-specific, as are brokers.
- Harvest gatherers use Essence system to assist in collecting data. Essence classifies documents by creating a semantic index.
- Semantic indexing generates different types of information for different types of files. It then creates indices on this information.

6.3.3 Virtual Web View

- Web-server administrators send their own indexes or pointers to resources to be indexed. When documents are changed, added or removed, the indexing process is triggered again.
- A Multiple layered database **(MLDB)** database is massive and distributed. Each layer is more generalized than the layer beneath it. To handle large amounts of unstructured data on the Web, Multiple layered database (MLDB) is used.
- It is a database composed of several layers of information, with the lowest layer (i.e. layer-0) corresponding to the primitive information stored in the global information base and the higher ones (i.e., layer-1 and above) storing generalized information extracted from the lower layers.
- The MLDB provides an abstracted and condensed view of a portion of the Web. A view of the MLDB, which is called a **Virtual Web View (VWV)** can be constructed. A VWV abstracts a selected set of resources and makes the WWW appear as structured.



(1F3)**Fig. 6.3.2: Virtual Web View**

- Various Generalization tools are proposed, and concept hierarchies are used to assist in the generalization process for constructing the higher levels of the MLDB.
- In general, the global MLDB structure is constructed based on the study of frequent accessing patterns. It is also possible to construct higher layered databases for a special-interest community of users on top of a common layer of the global database. This generates partial views on the global information network, hence, the name Virtual Web View (VWV). A VWV provides a window to observe a subset of Web resources and gives the illusion of a structured world.
- **WebML**, a web data mining query language is proposed to provide data mining operations on the MLDB.
- It is an extension of Data mining query language(DMQL). Web ML can be defined for resource and knowledge discovery using a syntax similar to the relational language SQL.

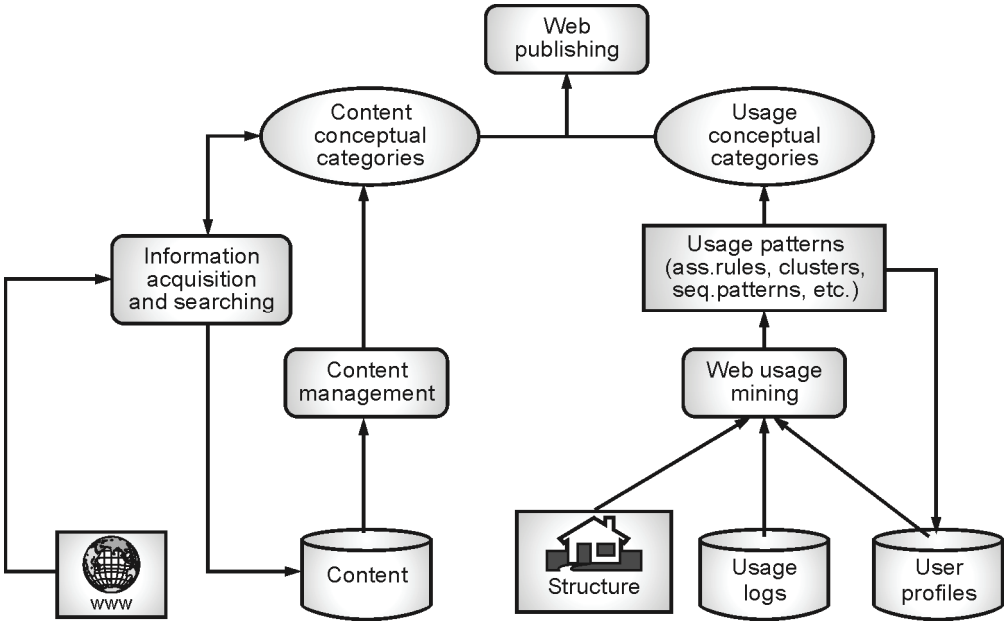
6.3.4 Personalization

- **Web personalization** is the process of customizing a web site to the needs of each specific user or set of users.
- Personalization of a web site may be performed by the provision of recommendations to the users, highlighting/adding links, creation of index pages, etc.
- The web personalization systems are mainly based on the exploitation of the navigational patterns of the web site's visitors.
- The process of providing information that is related to user's current page is known as web personalization. This information is usually displayed on the current page in the form of web page links. The idea behind

web personalization is that the web page currently being browsed by a user indicates his/her interest in that topic and it is likely that the user would be interested in more similar information.

- For example, in case of e-commerce the related information could be about other similar products to those that the user is viewing or about products that other users who bought or viewed this product also bought. This example would also work for a research or target oriented web browsing.
- The key information that is required for suggesting these similar web pages comes from the knowledge of other users who have also visited the current page as well as other pages before and after this current page.
- In addition to other users' browsing information, web personalization can also take advantage of the web page content, the structure of the web page or the user's profile information. All these help in creating a focused and personalized web browsing experience for the user.
- Web mining is the process that helps to collect and analyse information such as users' browsing pattern, content of the web pages, structure of the web site, etc. and aid in creating a personalized web browsing experience.

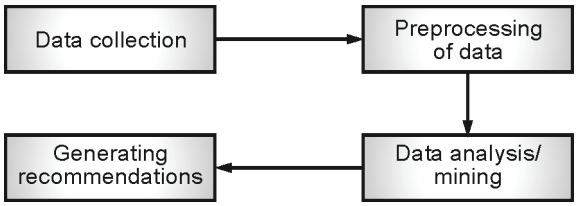
- Fig. 6.3.3 represents the components of web personalization.



(1F4)**Fig. 6.3.3 : Components of Web Personalization**

6.3.4(A) Phases of Web Personalization

- The web personalization process can be divided into four phases namely Data collection, pre-processing of web data, analysis of web data, and finally decision making or recommendation.



(1F5)**Fig. 6.3.4 : Components of Web Personalization**

1. Data Collection

- Data collection is the process of gathering information either explicitly or implicitly specific to each visitor for recording their interests and behavior while they browse a web site.
- The collection of activities completed in the past and recorded in Web server logs is known as the implicit data. This activity is performed by the web server and the user is not directly involved in collection of such data.
- The information submitted by the user at the time of registration or in response to the rating questionnaires

is considered as the explicit data and it usually comes from the active involvement of the user. Explicit data collection requires users to exert most of the efforts.

- Web data in the form of content, structure, semantic, usage and user profile may be collected and used in the context of Web personalization.
- A user profile includes information about users' interests and preferences, and it contains demographic information for each user of a Web site.

2. Pre-processing of data

- The data collected in previous step consists of various irrelevant information. For example, the log data collected from the web server are in the form of text files with a row for each http transaction.
- These data need to be cleaned before putting them for analysis. The Pre-processing task is performed to clean the data form inconsistencies. It filters out irrelevant information according to the goal of analysis.

3. Data analysis / Mining

- In this phase, the specific data mining techniques which are used for mining of web data are applied to the pre-processed data to discover interesting usage patterns.
- These usage patterns may form the groups according to the users' behavior. It classifies the content of a web site into semantic categories in order to make information retrieval and presentation easier for the user.
- This step is applied offline for automatic user profiling without adding the burden to the web server.

4. Recommendation Phase

- This is the last phase, and it deals with the actions that should be performed after taking the results of the previous analysis step.
- This phase usually performs the recommendations to the users by determining existing hyperlinks, the dynamic insertion of new hyperlinks that seems to be of interest for the current user to the last web page requested by the user, or even the creation of new index pages.

6.3.4(B) Types of Personalization

There are three approaches used for generating a personalized Web experience for a user:

1. Content-Based Filtering

- Content based filtering systems have their roots in information retrieval. The approach to recommendation generation is based around the analysis of items previously rated by a user and generating a profile for a user based on the content descriptions of these items.
- The profile is then used to predict a rating for previously unseen items and those deemed as being potentially interesting are presented to the user.
- Several early recommender systems were based on content-based filtering including Personal Web-Watcher, Info Finder, Newsreaders, Letizia and Syskill and Webert.

2. Collaborative Filtering

- Collaborative filtering was introduced as an alternative to content based filtering of a stream of electronic documents.
- The basic idea as presented by Goldberg et al. was that people collaborate to help each other perform filtering by recording their reactions to e-mails in the form of annotations. The application of this technology for recommending products has gained popularity and commercial success.
- Users provide feedback on the items that they consume, in the form of ratings. To recommend items to the active user, previous feedback is used to find other likeminded users. These are users that have provided similar feedback to many items that have been consumed by users.
- Items that have been consumed by compatible users but not by the current user are candidates for recommendation. The assumption made by these systems is that users that have had common interests in the past, defined by feedback on items consumed, will have similar tastes in the future.

3. Model Based Techniques

- **Model based collaborative filtering** techniques use a two-stage process for recommendation generation.
- The first stage is carried out offline, where user behavioral data collected during previous interactions is mined and an explicit model generated for use in future online interactions.
- The second stage is carried out in real-time as a new visitor begins an interaction with the Web site.
- Data from the current user session is scored using the models generated offline, and recommendations generated based on this scoring.
- The application of these models are generally computationally inexpensive compared to memory-based approaches such as traditional collaborative filtering, aiding scalability of the real time component of the recommender system.

6.4 WEB STRUCTURE MINING

UQ. What is Web Structure Mining ? List the approaches used to structure the web pages to improve on effectiveness of search engines and crawlers. Explain Page Rank technique in detail.

MU - Dec. 2019

- Hypertext documents in the World-Wide Web have several interconnections among them. These hyperlinks can reveal more information than just the information contained in documents.
- For example, links pointing to a document indicate the popularity of the document, while links coming out of a document indicate the richness or perhaps the variety of topics covered in the document.
- **Web structure mining** is used for creating a model of the web organization. It is the process of analyzing the nodes and connection structure of a website using graph theory.
- There are two things that can be obtained from this: the structure of a website in terms of how it is connected to other sites and the document structure of the website itself, as to how each page is connected.
- The web structure mining can be used to discover the link structure of hyperlink. It is used to identify that the web pages are either linked by information or direct link connection.
- The purpose of web structure mining is to produce the structural summary of website and similar web pages. This can be used to classify web pages or to create similarity measures between documents.
- Data from hyperlinks that lead to different pages are gathered and prepared to discover a pattern. In order to view a person's public profile from a blog or any other webpage, there are chances that they would embed their social media links.
- So, the data is not only extracted from a single source but also from the nested pages through the hyperlinks associated with each page. There are various algorithms to perform this. (Example: Page Rank algorithm, CLEVER)
- Example: Web structure mining can be very useful to companies to determine the connection between two commercial websites.

6.4.1 Page Rank

- The Page Rank (PR) algorithm is applicable in web pages. Page Rank is an algorithm used by Google Search to rank websites in their search engine results.
- Page Rank was named after Larry Page, one of the founders of Google.
- Page Rank algorithm is designed to increase the effectiveness of search engines and improve their efficiency.
- It is a way of measuring the importance of website pages.
- Page Rank is used to prioritize the pages returned from a traditional search engine using keyword searching.
- Page Rank is calculated based on the number of pages that point to it. The value of the Page Rank is the probability will be between 0 and 1.
- A Web page is a directed graph having two important components: nodes and connections. The pages are nodes and hyperlinks are the connections, the connection between two nodes.
- Page Rank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is

that more important websites are likely to receive more links from other websites.

- The Page Rank value of individual node in a graph depends on the Page Rank value of all the nodes which connect to it and those nodes are cyclically connected to the nodes whose ranking we want; we use converging iterative method for assigning values to Page Rank.
- In short Page Rank is a "vote", by all the other pages on the Web, about how important a page is. A link to a page counts as a vote of support. If there's no link, there's no support (but it's an abstention from voting rather than a vote against the page).

6.4.2 Algorithm

- We assume page A has pages B...N which point to it (i.e., are citations).
- The parameter d is a damping factor/teleportation factor which can be set between 0 and 1. We usually set it to 0.85.
- Also, $C_{out}(A)$ is defined as the number of links going out of page A.
- A page with no link out is called a **dead end**.
- A **spider trap** is a set of nodes with no dead ends but no arcs out. These structures can appear intentionally or unintentionally on the Web, and they cause the Page Rank calculation to place all the Page Rank within the spider traps. That is spider traps make an infinite number of requests or cause a poorly constructed crawler to crash.
- Both dead end and spider traps are to be avoided.
- The Page Rank of a page A is given as follows:

$$PR(A) = (1 - \beta) + \beta\ (PR(B) / C_{out}(B) + PR(C) / C_{out}(C) + ... + PR(N) / C_{out}(N))$$

OR

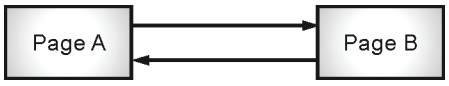
$$PR(A) = (1 - d) + d\ (PR(B) / C_{out}(B) + PR(C) / C_{out}(C) + ... + PR(N) / C_{out}(N))$$

> **Note :** The Page Ranks form a probability distribution over web pages, so the sum of all web pages' Page Ranks will be one.

Page Rank or PR(A) can be calculated using a simple iterative algorithm and corresponds to the principal eigenvector of the normalized link matrix of the web. Here are some important terms:

1. **PR(A) :** Each page has a notion of its own self-importance. That's "PR(A)" for the first page in the web all the way up to "PR(N)" for the last page.
2. **$C_{out}(N)$:** Each page spreads its vote out evenly amongst all of it's outgoing links. The count, or number, of outgoing links for page 1 is "C(A)", "C(N)" for page N, and so on for all pages.
3. **PR(N) / $C_{out}(N)$:** If our page (page A) has a backlink from page "N" the share of the vote page A will get is "PR(N)/ $C_{out}(N)$"
4. **d(... :** All these fractions of votes are added together but, to stop the other pageshaving too much influence, this total vote is "damped down" by multiplying it by 0.85 (the factor "d" or " β")
5. **Teleportation factor (β)/ dampling factor (d) :** There could be problem of Dead ends in a graph, in case, if there are no outlinks. Teleportation consists of connecting each node of the graph to all other nodes. The graph will be then complete.
6. **(1 – d) or (1 – β) :** The (1 – d) or (1 – β) bit at the beginning is a bit of probability math magic so the "sum of all web pages' Page Ranks will be one": it adds in the bit lost by the d(.... It also means that if a page has no links to it (no back links) even then it will still get a small PR of 0.15 (i.e. 1 – 0.85). (Aside: The Google paper says "the sum of all pages" but they mean the "the normalized sum" – otherwise known as "the average" to you and me.

Ex. 6.4.1 : Let's take the simplest example network: two pages, each pointing to the other:



(1F6)**Fig. P. 6.4.1**

☑ **Soln. :**

Each page has one outgoing link (the outgoing count is 1, i.e. $C_{out}(A) = 1$ and $C_{out}(B) = 1$).

We don't know what their PR should be to begin with, so let's take a guess at 1.0 and do the calculations:

$$\beta = 0.85$$

$$PR(A) = (1 - \beta) + \beta\ (PR(B)/1)$$

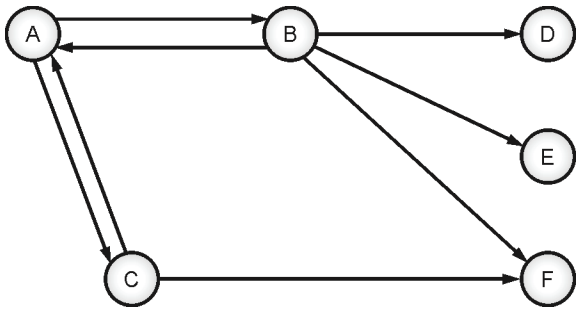
$$PR(B) = (1-\beta) + \beta\,(PR(A)/1)$$

i.e.

$$PR(A) = 0.15 + 0.85 * 1 = 1$$

$$PR(B) = 0.15 + 0.85 * 1 = 1$$

Ex. 6.4.2 : Using the web graph shown below, compute the Page Rank at every node at the end of the second iteration. Use teleportation factor = 0.8.



(1F6)**Fig. P. 6.4.2**

☑ **Soln. :**

Using the formula, $PR(A) = (1-\beta) + \beta\,(PR(B)/C_{out}(B) + PR(C)/C_{out}(C) + \ldots + PR(N)/C_{out}(N))$

$$PR(A) = (1 - 0.8) + 0.8 * (PR(B)/4 + PR(C)/2)$$

$$PR(B) = (1 - 0.8) + 0.8 * (PR(A)/2)$$

$$PR(C) = (1 - 0.8) + 0.8 * (PR(A)/2)$$

$$PR(D) = (1 - 0.8) + 0.8 * (PR(B)/4)$$

$$PR(E) = (1 - 0.8) + 0.8 * (PR(B)/4)$$

$$PR(F) = (1 - 0.8) + 0.8 * (PR(B)/4 + PR(C)/2)$$

☞ **Iteration 0 :**

Assume each page is having Page Rank = 1/Total no. of nodes

Therefore, PR(A) = PR(B) = PR(C) = PR(D) = PR(E) = PR(F) = 1/6 = 0.167

☞ **Iteration 1 :**

$$PR(A) = (1 - 0.8) + 0.8 * (PR(B)/4 + PR(C)/2)$$
$$= (1 - 0.8) + 0.8 * (0.167/4 + 0.167/2) = 0.3$$

Now, use this updated Page Rank for further calculations.

$$PR(B) = (1 - 0.8) + 0.8 * (PR(A)/2)$$
$$= 0.2 + 0.8 * (0.3/2) = 0.32$$

$$PR(C) = (1 - 0.8) + 0.8 * (PR(A)/2)$$
$$= 0.2 + 0.8 * (0.3/2) = 0.32$$

$$PR(D) = (1 - 0.8) + 0.8 * (PR(B)/4)$$
$$= 0.2 + 0.8 * (0.32/4) = 0.264$$

$$PR(E) = (1 - 0.8) + 0.8 * (PR(B)/4)$$
$$= 0.2 + 0.8 * (0.32/4) = 0.264$$

$$PR(F) = (1 - 0.8) + 0.8 * (PR(B)/4 + PR(C)/2)$$
$$= 0.2 + 0.8 * ((0.32/4) + (0.32/2)) = 0.392$$

☞ **Iteration 2 :**

$$PR(A) = (1 - 0.8) + 0.8 * (PR(B)/4 + PR(C)/2)$$
$$= (1 - 0.8) + 0.8 * (0.32/4 + 0.32/2) = 0.392$$

Now, use this updated Page Rank for further calculations.

$$PR(B) = (1 - 0.8) + 0.8 * (PR(A)/2)$$
$$= 0.2 + 0.8 * (0.392/2) = 0.3568$$

$$PR(C) = (1 - 0.8) + 0.8 * (PR(A)/2)$$
$$= 0.2 + 0.8 * (0.392/2) = 0.3568$$

$$PR(D) = (1 - 0.8) + 0.8 * (PR(B)/4)$$
$$= 0.2 + 0.8 * (0.3568/4) = 0.2714$$

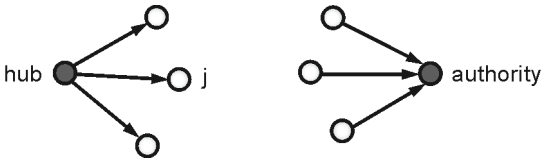
$$PR(E) = (1 - 0.8) + 0.8 * (PR(B)/4)$$
$$= 0.2 + 0.8 * (0.3568/4) = 0.2714$$

$$PR(F) = (1 - 0.8) + 0.8 * (PR(B)/4 + PR(C)/2)$$
$$= 0.2 + 0.8 * ((0.3568/4) + (0.3568/2))$$
$$= 0.4141$$

6.4.2 Hyperlink-induced Topic Search(HITS)

- **HITS**, also known as **Hubs and authorities**, developed by Jon Kleinberg is a link analysis algorithm that rates Web pages. It was a precursor to Page Rank.
- The idea behind **Hubs** and **Authorities** stemmed from a particular insight into the creation of web pages when the Internet was originally forming; that is, certain web pages, known as hubs, served as large directories that were not actually authoritative in the information that it held, but were used as compilations of a broad catalog of information that led users directly to other authoritative pages.
- In other words, a good hub represented a page that pointed to many other pages, and a good authority represented a page that was linked by many different hubs.

- The scheme therefore assigns two scores for each page: its authority, which estimates the value of the content of the page, and its hub value, which estimates the value of its links to other pages. A page may be a good hub and a good authority at the same time.
- The HITS algorithm treats WWW as directed graph G(V,E),where V is a set of vertices representing pages and E is set of edges corresponds to link.
- Attempts to computationally determine hubs and authorities on a particular topic through analysis of a relevant sub graph of the web.
- Based on mutually recursive facts: Hubs point to lots of authorities. Authorities are pointed to by lots of hubs.
  - **(a) Authority :** A valuable and informative webpage usually pointed to by many hyperlinks. A page that provide an important, trustworthy information on a given topic.
  - **(b) Hub :** A webpage that points to many authority pages is itself a resource and is called a hub.
- Authorities and hubs reinforce one another. A good authority is pointed to by many good hubs. A good hubpoints to many good authorities.



(1F7)**Fig. 6.4.1 : Hub and Authority**

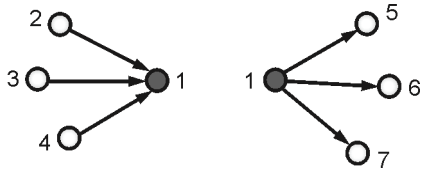
6.4.2(A) Calculating the Hub and Authority Weights

If A is the adjacency matrix of graph G = (V,E), then

$$\text{Authority weight : } \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

$$\text{Hub weight : } \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix}$$

For example,



(1F8)**Fig. 6.4.2**

Then, a(1) = h(2) + h(3) + h(4) and h(1) = a(5) + a(6) + a(7)

☞ **Algorithm**

- Each page p is assigned two non negative weights, an authority weight(a) and a hub weight(h). Update the weights of **a** and **h.**

$$\text{Authority Weight a (j)} = \sum_{i : (i, j) \in E} h(i)$$

$$\text{Hub Weight a (j)} = \sum_{i : (i, j) \in E} a(i)$$

- These operations add the weights of hubs into the authority weight and add the authority weights into the hub weight respectively.
- Alternating these two operations will eventually result in an equilibrium value, or weight, for each page.

G : a collection of n linked pages

Set $a_0 = [1/n \ldots 1/n]^T$

Set $h_0 = [1/n \ldots 1/n]^T$

For t = 1, 2, … k

For j = 1, 2, …, n

- Obtain new authority weights $a_t'(j) = \sum_{i : (i, j) \in E} h_{t-1}(i-1)$
- Normalize weights $a_t(j) = \dfrac{a_t'(j)}{\sum_j a_t'(j)}$
- Obtain new hub weights $h_t'(j) = \sum_{i : (j, i) \in E} a_t(j)$
- Normalize weights $h_t(j) = \dfrac{h_t'(j)}{\sum_j h_t'(j)}$

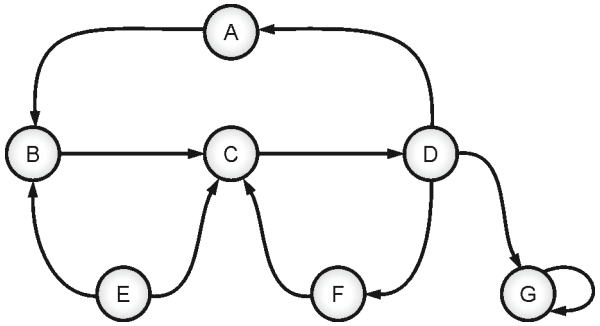
End

6.4.2(B) Constraints of HITS Algorithm

- **Hubs and authorities :** It is not easy to distinguish between hubs and authorities because many sites are hubs as well as authorities.

- **Topic drift :** Sometime HITS may not produce the most relevant documents to the user queries because of equivalent weights.
- **Automatically generated links :** HITS gives equal importance for automatically generated links which may not produce relevant topics for the user query.
- **Efficiency :** HITS algorithm is not efficient in real time.

Ex. 6.4.3 : Consider the portion of the web graph shown below **:**



(1F9)**Fig. P. 6.4.3**

(a) Compute the hub and authority scores for all nodes.

(b) Does this graph contain Spider traps ? Dead nodes ? If so, which nodes ?

(c) Compute the Page Rank of the nodes with the teleportation factor = 0.8.

☑ **Soln. :**

(a)

▶ **Step 1 :** Calculate the adjacency matrix of the graph.

$$Adj = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

▶ **Step 2 :** Calculate transpose of the adjacency matrix of the graph.

$$Adj^T = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

▶ **Step 3 :** Assume the initial hub weight vector (for 7 nodes)

$$h = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

▶ **Step 4 :** Calculate authority weight vector.

$$A = Adj^T \times h$$

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \\ 0 \\ 1 \\ 2 \end{bmatrix}$$

▶ **Step 5 :** Calculate the updated hub vector.

$$h = Adj \times A$$

$$h = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \\ 0 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 1 \\ 4 \\ 5 \\ 3 \\ 2 \end{bmatrix}$$

(b) This graph has dead end at node G where it is a loop to the self.

For other nodes, there are outgoing links. So, no dead end at other nodes.

(C) To compute the Page Rank

Using the formula, $PR(A) = (1 - \beta) + \beta\ (PR(B)/C_{out}(B) + PR(C)/C_{out}(C) + \ ... + PR(N)/C_{out}(N))$

$PR(A) = (1 - 0.8) + 0.8 * (PR(D)/3)$

$PR(B) = (1 - 0.8) + 0.8 * ((PR(A)/1) + (PR(E)/2))$

$PR(C) = (1 - 0.8) + 0.8 * ((PR(B)/1) + (PR(E)/2) + (PR(F)/2))$

$PR(D) = (1 - 0.8) + 0.8 * (PR(C)/1)$

$PR(E) = (1 - 0.8) + 0.8 * (0)$
$\rightarrow$ {Since there are no links}

$PR(F) = (1 - 0.8) + 0.8 * (PR(D)/3)$

$PR(G) = (1 - 0.8) + 0.8 * (PR(D)/3 + PR(G)/1)$

☞ **Iteration 0 :**

Assume each page is having Page Rank = 1/Total no. of nodes

Therefore, PR(A) = PR(B) = PR(C) = PR(D)
= PR(E) = PR(F) = PR(G)= 1/7 = 0.1428

☞ **Iteration 1 :**

$PR(A) = (1 - 0.8) + 0.8 * (PR(D)/3)$
$= (1 - 0.8) + 0.8 * (0.0476) = 0.2381$

$PR(B) = (1 - 0.8) + 0.8 * ((PR(A)/1) + (PR(E)/2))$
$= (1 - 0.8) + 0.8 * (0.2381/1) + (0.1428/2))$
$= 0.4476$

$PR(C) = (1 - 0.8) + 0.8 * ((PR(B)/1) + (PR(E)/2) + (PR(F)/2))$
$= (1 - 0.8) + 0.8 * ((0.4476/1) + (0.1428/2) + (0.1428/2)) = 0.7295$

$PR(D) = (1 - 0.8) + 0.8 * (PR(C)/1) = (1 - 0.8) + 0.8 * (0.7295/1) = 0.7836$

$PR(E) = (1 - 0.8) + 0.8 * (0) = 0.2$

$PR(F) = (1 - 0.8) + 0.8 * (PR(D)/3) = (1 - 0.8) + 0.8 * (0.7836/3) = 0.4089$

$PR(G) = (1 - 0.8) + 0.8 * (PR(D)/3 + PR(G)/1)$
$= (1 - 0.8) + 0.8 * (0.7836/3 + 0.1428/1)$
$= 0.5232$

☞ **Iteration 2 :**

$PR(A) = (1 - 0.8) + 0.8 * (PR(D)/3) = (1 - 0.8) + 0.8 * (0.7836/3) = 0.4089$

$PR(B) = (1 - 0.8) + 0.8 * ((PR(A)/1) + (PR(E)/2))$
$= (1 - 0.8) + 0.8 * (0.4089/1) + (0.2/2))$
$= 0.6071$

$PR(C) = (1 - 0.8) + 0.8 * ((PR(B)/1) + (PR(E)/2) + (PR(F)/2))$
$= (1 - 0.8) + 0.8 * ((0.6071/1) + (0.2/2) + (0.4089/2)) = 1.0928$

$PR(D) = (1 - 0.8) + 0.8 * (PR(C)/1) = (1 - 0.8) + 0.8 * (1.0928/1) = 1.0742$

$PR(E) = (1 - 0.8) + 0.8 * (0) = 0.2$

$PR(F) = (1 - 0.8) + 0.8 * (PR(D)/3) = (1 - 0.8) + 0.8 * (1.0742/3) = 0.4875$

$PR(G) = (1 - 0.8) + 0.8 * (PR(D)/3 + PR(G)/1)$
$= (1 - 0.8) + 0.8 * (1.0742/3 + 0.5232/1)$
$= 0.9050$

Ex. 6.4.4 : Let the adjacency matrix for a graph of four vertices $\{n_1$ to $n_4\}$ is as given below **:**

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Calculate the authority and hub scores for this graph using HITS algorithm with K= 6 and identify the best authority and hub nodes.

☑ **Soln. :**

$$Adj = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Adj^T = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

For K = 1;

$$Let\ h = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Authority Vector (A) $= Adj^T \times h$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 4 \end{bmatrix}$$

The updated hub vector, $h = Adj \times A$

$$h = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 2 \\ 4 \end{bmatrix} = \begin{bmatrix} 7 \\ 6 \\ 5 \\ 4 \end{bmatrix}$$

For K = 2;

Authority Vector (A) $= Adj^T \times h$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 7 \\ 6 \\ 5 \\ 4 \end{bmatrix} = \begin{bmatrix} 5 \\ 7 \\ 13 \\ 22 \end{bmatrix}$$

The updated hub vector, $h = Adj \times A$

$$h = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 5 \\ 7 \\ 13 \\ 22 \end{bmatrix} = \begin{bmatrix} 42 \\ 35 \\ 27 \\ 22 \end{bmatrix}$$

For K = 3;

Authority Vector (A) $= Adj^T \times h$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 42 \\ 35 \\ 27 \\ 22 \end{bmatrix} = \begin{bmatrix} 27 \\ 42 \\ 77 \\ 126 \end{bmatrix}$$

The updated hub vector, $h = Adj \times A$

$$h = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 27 \\ 42 \\ 77 \\ 126 \end{bmatrix} = \begin{bmatrix} 245 \\ 203 \\ 153 \\ 126 \end{bmatrix}$$

For K = 4;

Authority Vector (A) $= Adj^T \times h$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 245 \\ 203 \\ 153 \\ 126 \end{bmatrix} = \begin{bmatrix} 153 \\ 245 \\ 448 \\ 727 \end{bmatrix}$$

The updated hub vector, $h = Adj \times A$

$$h = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 153 \\ 245 \\ 448 \\ 727 \end{bmatrix} = \begin{bmatrix} 1420 \\ 1175 \\ 880 \\ 727 \end{bmatrix}$$

For K = 5;

Authority Vector (A) $= Adj^T \times h$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1420 \\ 1175 \\ 880 \\ 727 \end{bmatrix} = \begin{bmatrix} 880 \\ 1420 \\ 2595 \\ 4202 \end{bmatrix}$$

The updated hub vector, $h = Adj \times A$

$$h = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 880 \\ 1420 \\ 2595 \\ 4202 \end{bmatrix} = \begin{bmatrix} 8217 \\ 6797 \\ 5082 \\ 4202 \end{bmatrix}$$

For K = 6;

Authority Vector (A) $= Adj^T \times h$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 8217 \\ 6797 \\ 5082 \\ 4202 \end{bmatrix} = \begin{bmatrix} 5082 \\ 8217 \\ 15014 \\ \mathbf{24298} \end{bmatrix}$$

The updated hub vector, $h = Adj \times A$

$$h = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 5082 \\ 8217 \\ 15014 \\ 24298 \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{47529} \\ 39312 \\ 29380 \\ 24298 \end{bmatrix}$$

Hence after 6 iterations, we can say that the best authority node is n_4 and the best hub node is n_1 having highest values.

6.4.3 Clever

- **IBM** has developed a new search engine process which blends the speed and comprehension of an automated search engine with the discerning results of a human-generated index.

- A team at **Big Blue's Almaden Research Lab** has named their new technique **Clever**, for client-side, eigenvector-based retrieval.
- Clever distinguishes itself from conventional search engines by analyzing how documents on the Internet are linked to each other.
- At the heart of the Clever system is an algorithm which helps synthesize the information contained in a large number of hyperlinks on the Web, together with the 'context' of the contents on each page.
- To minimize the problems in the original HITS algorithm, a clever algorithm is proposed. Clever algorithm is the modification of standard original HITS algorithm. This algorithm provides a weight value to every link depending on the terms of queries and endpoints of the link.
- An anchor tag is combined to decide the weights to the link and a large hub is broken down into smaller parts so that every hub page is concentrated only on one topic.
- For example, Clever would respond to a search on "AIDS" by generating a rapid, preliminary list of about 300 pages. The engine would expand its search to include documents linked to and from those 300 pages, until it had gathered about 20,000 documents on the disease.
- The engine then analyzes and ranks those by assigning greater importance to the most frequently cited pages. The engine assumes such documents are more useful, in the way that important academic articles are cited frequently by other academic papers. The hubs are finally ranked on the number of links they have to those authorities.

6.4.4 HITS Vs Page Rank

The Table 6.4.1 depicts the comparison between Page Rank and HITS algorithm.

Table 6.4.1 : HITS Vs Page Rank

Algorithm	Page Rank	HITS
Mining technique used	Web structure mining	Web structure mining and Web content mining
Working	Computes scores at indexing time. Results are sorted according to importance of pages.	Computes hub and authority scores of n highly relevant pages on the fly.
Applied on	Page Rank is applied to the entire web.	HITS is applied to the local neighborhood of pages surrounding the results of a query.
Input parameters	Back links	Back links, Forward links and content
Complexity	O(log N)	O(log N)
Limitations	Query independent	Topic drift and efficiency problem
Search Engine	Google	Clever

6.5 WEB USAGE MINING

- **Web usage mining** is the process of extracting patterns and information from server logs to gain insight on user activity including where the users are from, how many clicked what item on the site and the types of activities being done on the site.
- When a web application is hosted, there are plenty of web server logs that gets generated about the application's user web activity. These logs are considered as a raw data in return meaningful data are extracted and patterns are identified.
- For instance, for any e-commerce business, when they want to increase the scope of business or add an enhancement for better customer experience, user's web activity through the application logs are monitored and data mining is applied to it.
- Talking about the data from the web, there are varieties of data that can be observed. It could be structured data (database data are pulled through API if it is released for public). Semi-structured data – any web activity

related or even server logs pull. Or even unstructured data like images etc. (if any analysis are performed on images)

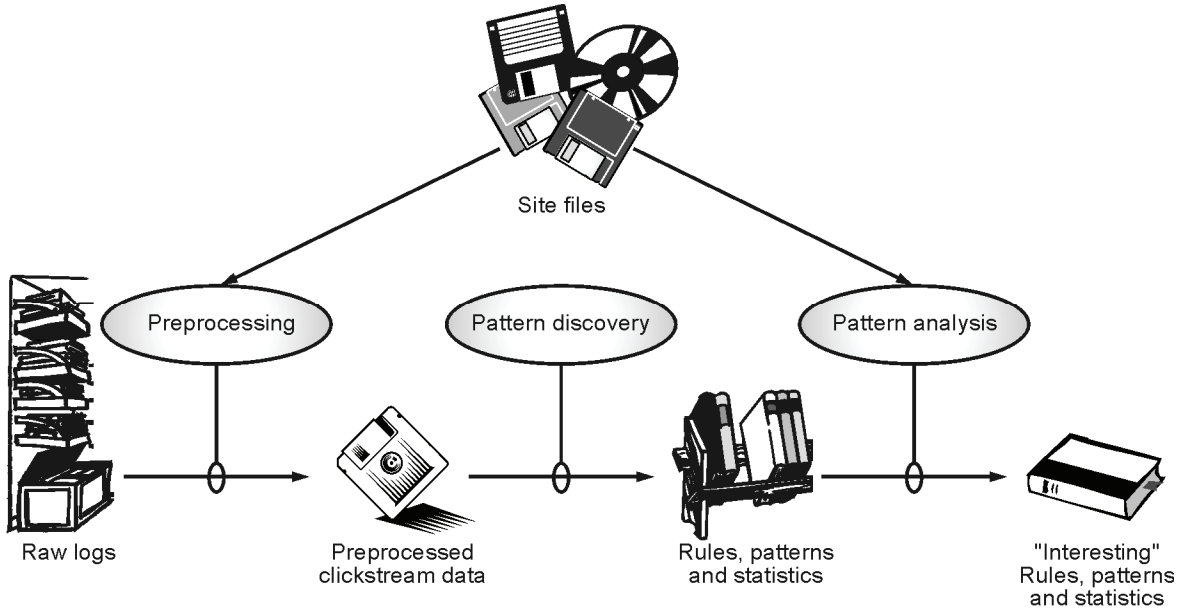
- Web server registers a web log entry for every web page. Analysis of similarities in web log records can be useful to identify the potential customers for e-commerce companies.
- Some of the techniques to discover and analyze the web usage pattern are **:**

(i) Session and visitor analysis

The analysis of pre-processed data can be performed in session analysis, which includes the record of visitors, days, sessions etc. This information can be used to analyze the behavior of visitors. Report is generated after this analysis, which contains the details of frequently visited web pages, common entry and exit.

(ii) OLAP (Online Analytical Processing)

OLAP performs Multidimensional analysis of complex data. OLAP can be performed on different parts of log related data in a certain interval of time. The OLAP tool can be used to derive the important business intelligence metrics.



(1F10)**Fig. 6.5.1 : Web Usage Mining process**

As shown in Fig. 6.5.1, there are three main tasks for performing Web Usage Mining or Web Usage Analysis.

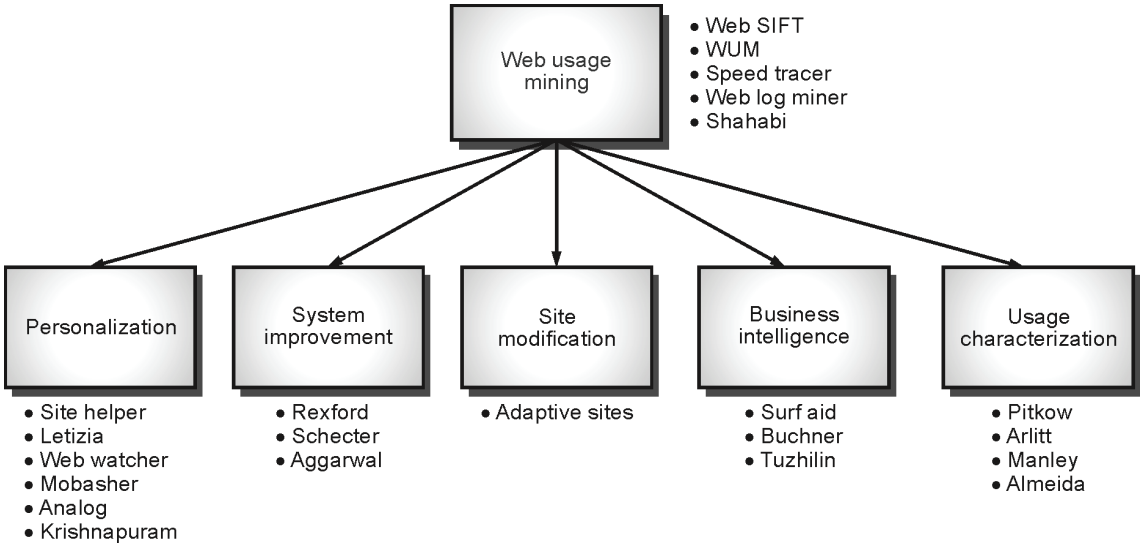
1. **Preprocessing :** Preprocessing consists of converting the usage, content, and structure information contained in the various available data sources into the data abstractions necessary for pattern discovery.
   - **Usage Preprocessing :** Usage preprocessing is arguably the most difficult task in the Web Usage Mining process due to the incompleteness of the available data. Unless a client-side tracking mechanism is used, only the IP address, agent, and server-side click stream are available to identify users and server sessions.
   - **Content preprocessing :** consists of converting the text, image, scripts, and other files such as multimedia into forms that are useful for the Web Usage Mining process. Often, this consists of performing content mining such as classification or clustering.
   - **Structure Preprocessing :** The structure of a site is created by the hypertext links between page views. The structure can be obtained and preprocessed in the same manner as the content of a site. Again, dynamic content (and therefore links) pose more problems than static page views. A different site structure may have to be constructed for each server session.

2. **Pattern Discovery :** Pattern discovery draws upon methods and algorithms developed from several fields such as statistics, data mining, machine learning and pattern recognition. Following the kinds of mining activities that have been applied to the Web domain **:**
   - **Statistical Analysis :** Statistical techniques are the most common method to extract knowledge about visitors to a Web site. By analyzing the session file, one can perform different kinds of descriptive statistical analyses (frequency, mean, median, etc.) on variables such as page views, viewing time and length of a navigational path.
   - **Association Rules :** Association rule generation can be used to relate pages that are most often referenced together in a single server session. In the context of Web Usage Mining, association rules refer to sets of pages that are accessed together with a support value exceeding some specified threshold.
   - **Clustering :** Clustering is a technique to group together a set of items having similar characteristics. In the Web Usage domain, there are two kinds of interesting clusters to be discovered **:** usage clusters and page clusters. Clustering of users tends to establish groups of users exhibiting similar browsing patterns. On the other hand, clustering of pages will discover groups of pages having related content.
   - **Classification :** Classification is the task of mapping a data item into one of several predefined classes. In the Web domain, one is interested in developing a profile of users belonging to a particular class or category. This requires extraction and selection of features that best describe the properties of a given class or category. Classification can be done by using supervised inductive learning algorithms such as decision tree classifiers, naive Bayesian classifiers, k-nearest neighbor classifiers, Support Vector Machines etc.
   - **Sequential Patterns :** The technique of sequential pattern discovery attempts to find inter-session patterns such that the presence of a set of items is followed by another item in a time-ordered set of sessions or episodes. By using this approach, Web marketers can predict future visit patterns which will be helpful in placing advertisements aimed at certain user groups.
   - **Dependency modeling :** Dependency modeling is another useful pattern discovery task in Web Mining. The goal here is to develop a model capable of representing significant dependencies among the various variables in the Web domain.
3. **Pattern Analysis :** Pattern analysis is the last step in the overall Web Usage mining process as described in Fig. 6.5.1. The motivation behind pattern analysis is to filter out uninteresting rules or patterns from the set found in the pattern discovery phase. The exact analysis methodology is usually governed by the application for which Web mining is done. The most common form of pattern analysis consists of a knowledge query mechanism such as SQL. Another method is to load usage data into a data cube to perform OLAP operations. Visualization techniques, such as graphing patterns or assigning colors to different values, can often highlight overall patterns or trends in the data. Content and structure information can be used to filter out patterns containing pages of a certain usage type, content type, or pages that match a certain hyperlink structure.

NOTES

6.5.1 Applications

1. Personalization for a user.
2. From frequent access behavior of user, overall performance can be improved.
3. Caching of frequently accessed pages.
4. Modifications of linkage structure, common access behavior are accessed.
5. Gather business intelligence to improve sales and advertisements.



(1F11)**Fig. 6.5.2: Major application areas for Web Usage Mining**

6.6 DATA MINING Vs. WEB MINING

The table below depicts the difference between Data mining and Web mining.

Table 6.6.1 : Data Mining Vs. Web Mining

Technique	Data Mining	Web Mining
Definition	Data Mining is the process that attempts to discover pattern and hidden knowledge in large data sets in any system.	Web Mining is the process of data mining techniques to automatically discover and extract information from web documents.
Concept	Pattern Identification from data available in any system.	Pattern Identification from web data.
Categories	Clustering, classification, regression, prediction, optimization and control.	Web content mining, Web structure mining, Web Usage Mining.
Target Users (Who does this ?)	Data scientist and data engineers.	Data scientists/data analysts and data engineers.
Skills	Data Cleansing, Statistics, Probability and Machine Learning algorithms.	Statistics, Probability, Application Level knowledge and Data engineering.
Access	Data Mining is access data privately.	Web Mining is access data publicly.
Applied on	Data Mining is very useful for web page analysis.	Web Mining is very useful for a particular website and e-service.
Tools	It includes tools like machine learning algorithms such as Decision tree, Naïve Bays, Support Vector Machine, etc.	Special tools for web mining are Rapid Miner, Scrapy, Page Rank and Apache logs.

⏭ 6.7 MULTIPLE CHOICE QUESTIONS

Q. 6.1 Page Rank is a metric for _____documents based on their quality.
(a) ranking hypertext
(b) ranking document structure
(c) ranking web content
(d) None of these ✓**Ans. : (a)**

Q. 6.2 _____ refers to the discovery of user access patterns from Web usage logs.
(a) Web content mining (b) Web structure mining
(c) Web usage mining (d) Data mining
✓**Ans. : (c)**

Q. 6.3 Web mining - is the application of _____
(a) Data Mining (b) Text Mining
(c) Both a and b (d) None of these ✓**Ans. : (a)**

Q. 6.4 The main purpose for structure mining is to extract previously unknown relationships between _____
(a) Web contents (b) Web pages
(c) Web hyperlinks (d) Web data ✓**Ans. : (b)**

Q. 6.5 Web structure mining is the process of discovering ____ information from the web.
(a) Semi structured (b) Unstructured
(c) Structured (d) None of the above
✓**Ans. : (c)**

Q. 6.6 Web Server Data includes _____
(a) IP address, (b) page reference
(c) access time (d) All of the Above ✓**Ans. : (d)**

Q. 6.7 Which of the following algorithm is used by Google to determine the importance of a particular page ?
(a) SVD (b) Page Rank
(c) FastMap (d) All of the mentioned ✓**Ans. : (b)**

Q. 6.8 HITS works on _____.
(a) Entire web graph
(b) Small subgraph of the web graph
(c) Page Rank
(d) Idea of Random surfer ✓**Ans. : (b)**

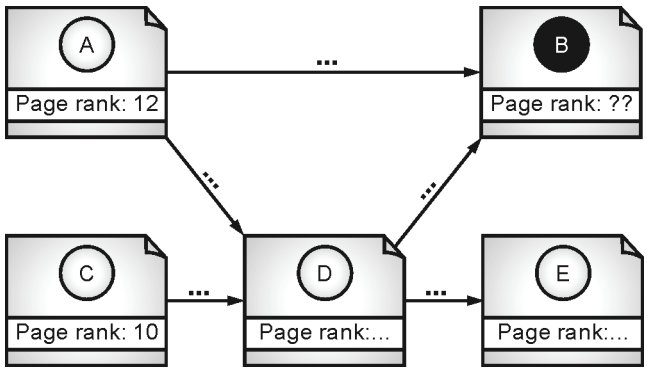
Q. 6.9 Page Rank algorithm works on _____.
(a) Entire web graph
(b) Small subgraph of the web graph
(c) Page Rank
(d) Idea of Random surfer ✓**Ans. : (a)**

Q. 6.10 When a search engine recognizes pages from other sites their database, the ranking is determined by **:**
(a) Index Analysis (b) Link Analysis
(c) Connectivity (d) Crawler scanner
✓**Ans. : (a)**

Q. 6.11 _____ is a mining task that examines the web and hyperlink structure that connect web pages.
(a) Web content mining (b) Web structure mining
(c) Web usage mining (d) Web link mining
✓**Ans. : (b)**

Q. 6.12 What does wen content mining involve ?
(a) Web content mining (b) Web structure mining
(c) Web usage mining (d) Web link mining
✓**Ans. : (b)**

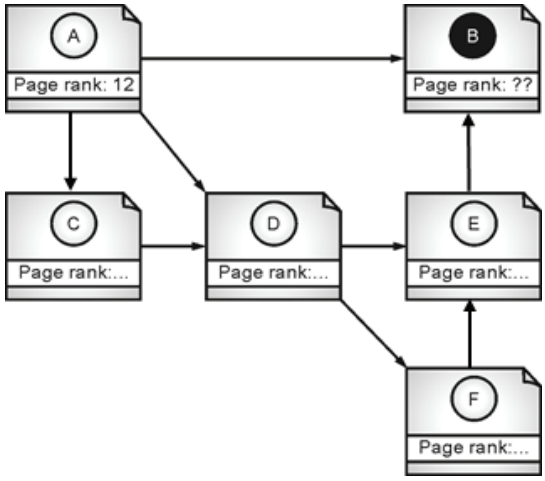
Q. 6.13 Find the page rank score of web page B.



(1F12)**Fig. Q. 6.13**

(a) 12 (b) 14 (c) 10 (d) 16 ✓**Ans. : (b)**

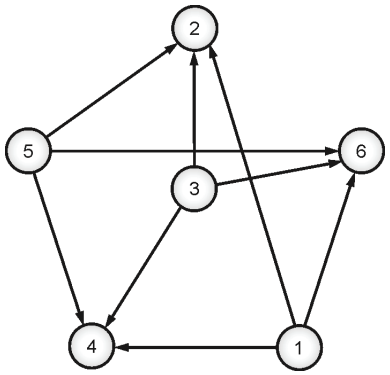
Q. 6.14 Find the page rank score of web page B.



(1F13)**Fig. Q. 6.14**

(a) 12 (b) 14 (c) 10 (d) 16 ✓**Ans. : (a)**

Q. 6.15 What is the score of authority (a) and hub (h) respectively for node 4 in the following figure after applying 1-step hub-authority computation (i.e. when k = 1) ?



(1F14)**Fig. Q. 6.15**

(a) a(1) = 9, h(1) = 0 (b) a(1) = 0, h(1) = 9

(c) a(1) = 3, h(1) = 0 (d) a(1) = 0, h(1) = 3

✓**Ans. : (c)**

Descriptive Questions

Q. 1 With respect to web mining, is it possible to detect visual objects using meta-objects ?**(MU - May 2019)**

☑ **Ans. :** Yes, it is possible to detect visual objects and attempt to describe video using metadata. However, that may not always be enough and allow for a thorough and accurate description of a video file. Additional information extraction towards understanding high-level meanings in visual data by possibly translating computable low-level multimedia features (like color histogram, shape, texture etc.) into high-level semantic concepts which humans can relate to; will be quite useful.

Q. 2 What is Web Structure Mining ? List the approaches used to structure the web pages to improve on effectiveness of search engines and crawlers. Explain Page Rank technique in detail.

(MU - Dec. 2019)

Q. 3 What is Web mining ? What are the three categories for web mining ?

Q. 4 Differentiate between HITS and Page Rank algorithm.

Q. 5 What is Dead end and Spider Trap ? What is significance of these two on Page Rank algorithm ?

Q. 6 Compare and contrast Data mining and Web mining.

Q. 7 Write a short note on **:** Harvest system.

Q. 8 Explain the three main tasks for performing Web Usage Mining.

Q. 9 What is Personalization ? Explain the different types of Personalization.

Q. 10 Explain HITS algorithm in detail.

Q. 11 What is Web content mining ? Explain the role of crawler in web content mining.

Q. 12 Write a short note on **:** Virtual Web View

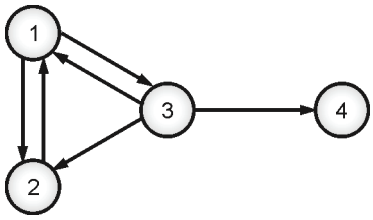
Q. 13 Explain components of Personalization.

Q. 14 What is crawler ? What are the factors based on which crawler works ? What are different types of crawlers ?

Q. 15 What is Hub and Authority node in HITS algorithm ? Write constraints on HITS algorithm.

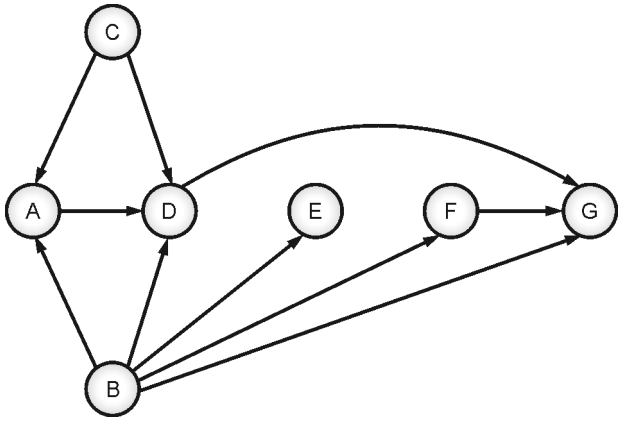
Q. 16 What is the role of Teleportation factor in Page Rank algorithm ?

Q. 17 For the graph given below show the page ranks of all the nodes after running the Page Rank algorithm for two iterations considering the teleportation factor $\beta = 0.8$.



(1F15)**Fig. Q. 17**

Q. 18 Compute hub and authority scores for the following web graph. After two iterations find the best authority and the hub node.



(1F16) **Fig. Q. 18**

Chapter Ends…

❑❑❑

Lab Manual

▶ Experiment No. 1 : Case Study on building Data Warehouse/Data Mart

In this experiment, we

1. Define the problem statement for building the data warehouse.
2. Draw the Star-schema for the above case study.
3. Draw the Snowflake schema for the above case study.

Here, we define the case study for "Hotel Occupancy".

It consists of four dimension tables and one fact table.

Dimension Tables :

Hotel	Room	Customer	Time
HotelID	RoomID	CustomerID	Date
HotelName	RoomType	CustomerName	Day_of_week
Rooms	Max_Occupant	Address	Day_of_month
StarRating	No_of_beds	Type_of_stay	Week
Region	Room_side	Check_in	Month
City	AC	Check_out	Year
State	Renovation_year	Amount_paid	Holiday
Country			

In the above tables, Primary Keys are underlined.

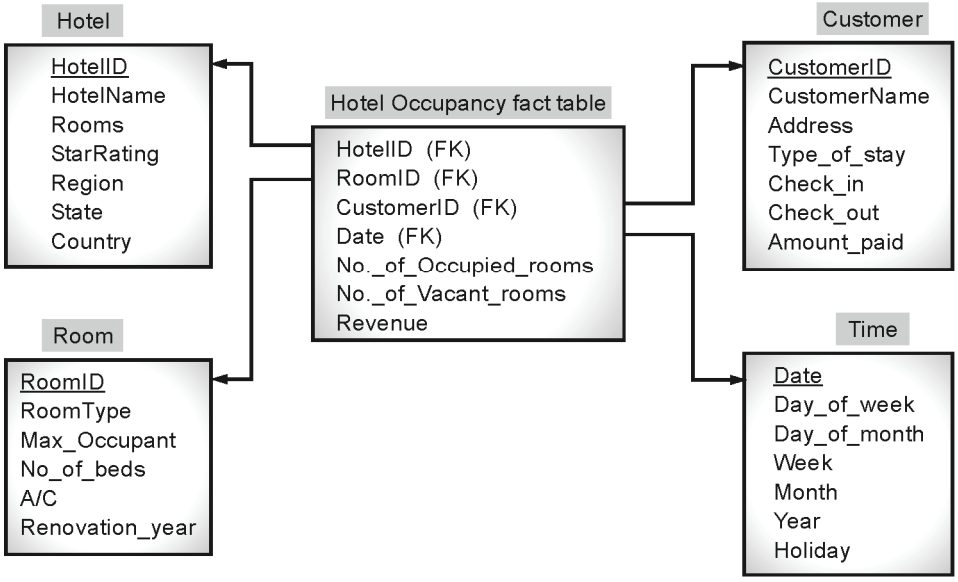
Attributes AC and Holiday are Flag Variables with values 'Y' or 'N'.

Fact Table :

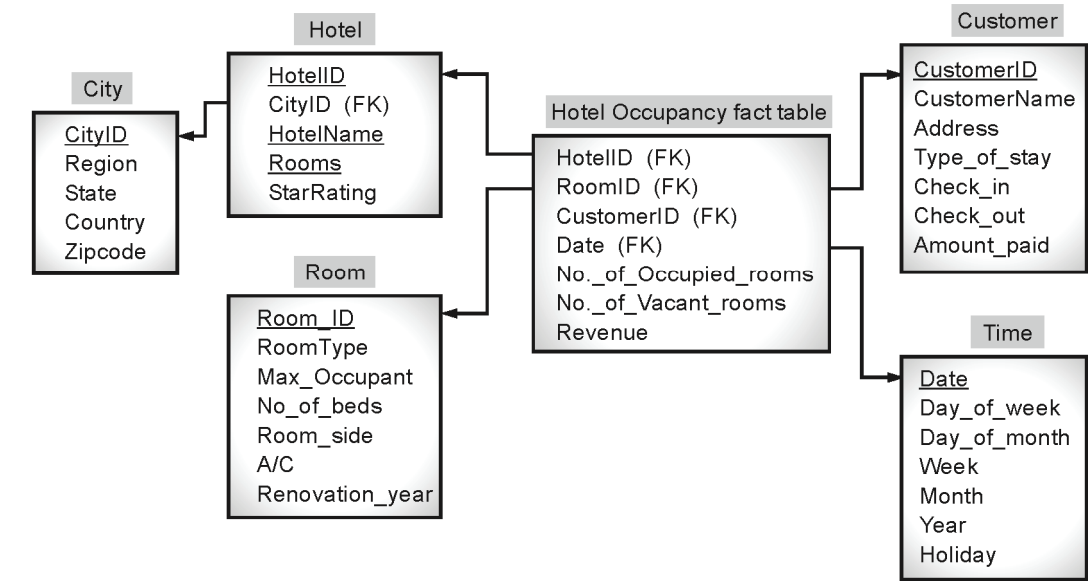
HotelOccupancy
HotelID (FK)
RoomID (FK)
CustomerID (FK)
Date (FK)
No_of_occupied_rooms
No_of_vacant_rooms
Revenue

FK indicates Foreign Key Referencing the respective dimension tables.

Star Schema:



Snowflake Schema:



(1G1)**Fig. L.1**

▶ Experiment No. 2 : Implementation of all dimension table and Fact table

In this experiment, we write the SQL queries for

1. Creating the Dimension Tables
2. Creating the Fact Table
3. Inserting values in both dimension and fact tables
4. Displaying the tables

For this experiment, we take the same case study as in Experiment 1.

#Creating database in MYSQL WorkBench

create database hoteldw;

#Use database

use hoteldw;

#Creating Dimension Hotel

CREATE TABLE DimHotel (HotelID int auto_increment primary key, HotelName varchar(50) not null, Rooms int, HotelType varchar(50), StarRating int, Region varchar(50), City varchar(50), State varchar(50), Country varchar(50));

#Inserting Values in Dimension Hotel

INSERT INTO DimHotel (HotelName, Rooms, HotelType, StarRating, Region, City, State, Country) VALUES ('Treehotel', 300, 'Inn', 4, 'Edeforsvag 2 A','Vidväg 97', 'Harads', 'Sweden'),

('Kakslauttanen Arctic Resort', 200, 'Hotel', 4, 'Kiilopaantie 9','Kiilopaantie 9', 'Saariselka', 'Finland'),

('Giraffe Manor', 500, 'Specialty Lodging', 3, 'Koitobos Rd','Koitobos Rd', 'Nairobi', 'Kenya'),

('Fantasyland Hotel & Resort', 450, 'Hotel', 4, '17700 87 Ave NW','Edmonton', 'Alberta', 'Canada'),

('Ottoman Cave Suites', 700, 'Hotel', 5, 'AvcilarMahallesi','IlkokulSokak No 16', 'Goreme', 'Turkey'),

("Helga's Folly", 700, 'Hotel', 4, '70 Rajaphilla Mawatha','70 RajaphillaMawatha', 'Kandy', 'Sri Lanka');

#Displaying Dimension Hotel

select * from DimHotel;

Hotel ID	Hotel Name	Rooms	Hotel Type	Star Rating	Region	City	State	Country
1	Treehotel	300	Inn	4	Edeforsvag 2 A	Vid väg 97	Harads	Sweden
2	Kakslauttanen Arctic Resort	200	Hotel	4	Kiilopaantie 9	Kiilopaantie 9	Saariselka	Finland
3	Giraffe Manor	500	Specialty Lodging	3	Koitobos Rd	Koitobos Rd	Nairobi	Kenya
4	Fantasyland Hotel & Resort	450	Hotel	4	17700 87 Ave NW	Edmonton	Alberta	Canada
5	Ottoman Cave Suites	700	Hotel	5	AvcilarMahallesi	IlkokulSokak No 16	Goreme	Turkey
6	Helga's Folly	700	Hotel	4	70 RajaphillaMawatha	70 RajaphillaMawatha	Kandy	Sri Lanka

 # Creating Dimension Room

CREATE TABLE DimRoom(RoomID int auto_increment primary key, RoomType varchar(50), Max_Occupant int, No_of_beds int, Room_side varchar(50), AC varchar(1), Renovation_year year);

#Inserting Values in Dimension Room

INSERT INTO DimRoom (RoomType, Max_Occupant, No_of_beds, Room_side, AC, Renovation_year) VALUES ('Queen', 3, 3, 'West', 'Y', 2015),

('King', 2, 2, 'South', 'N', 2012),

('King', 2, 2, 'West', 'Y', 2015),

('Suite', 4, 4, 'East', 'N', 2011),

('King', 2, 2, 'North', 'N', 1993),

('King', 2, 2, 'South', 'Y', 1980);

 # Displaying Dimension Room

select * from DimRoom ;

Room ID	Room Type	Max_Occupant	No_of_beds	Room_side	AC	Renovation_year
1	Queen	3	3	West	Y	2015
2	King	2	2	South	N	2012
3	King	2	2	West	Y	2015
4	Suite	4	4	East	N	2011
5	King	2	2	North	N	1993
6	King	2	2	South	Y	1980

 # Create Dimension Customer

CREATE TABLE DimCustomer (CustomerID int auto_increment primary key, CustomerName varchar(50) not null, Address varchar(100), Type_of_stay varchar(50), Check_in datetime, Check_out datetime, Amount_paid decimal(19,4));

#Insert Values in Dimension Customer

INSERT INTO DimCustomer (CustomerName, Address, Type_of_stay, Check_in, Check_out, Amount_paid) VALUES ('Krish Khatri', '061, Shenoy Nagar New Delhi-214996', 'Night', '2016-03-08 01:00:22', '2016-03-18 09:54:24', 16871.9772),

('NayantaraKalita', '72/64 JaggiZilaRaebareli 623911', 'Night', '2018-06-27 22:07:35', '2018-06-30 08:18:50', 14708.9695),

('PurabRamaswamy', '67/74 Malhotra Ganj, Mau-524031', 'Day', '2019-04-30 15:15:37', '2019-05-10 02:31:12', 14394.8425),

('NehmatVerma', '52/20 Thaman Gulbarga-366179', 'Day', '2015-06-01 11:22:44', '2015-06-26 05:27:48', 18337.4186),

('ArmaanJohal', 'H.No. 71 Basu Circle, Raiganj 310096', 'Day', '2015-12-06 08:32:13', '2015-12-14 19:25:12', 13656.1320),

('KismatChada', '85/504 ShereChowkMehsana 483469', 'Day', '2012-12-21 18:06:03', '2012-12-28 05:28:15', 17314.9652);

#Displaying Dimension Customer

select * from DimCustomer;

Customer ID	Customer Name	Address	Type_of_stay	Check_in	Check_out	Amount_paid
1	Krish Khatri	061, Shenoy Nagar New Delhi-214996	Night	2016-03-08 01:00:22	2016-03-18 09:54:24	16871.9772
2	NayantaraKalita	72/64 JaggiZilaRaebareli 623911	Night	2018-06-27 22:07:35	2018-06-30 08:18:50	14708.9695

Customer ID	Customer Name	Address	Type_of_stay	Check_in	Check_out	Amount_paid
3	PurabRamaswamy	67/74 Malhotra Ganj, Mau-524031	Day	2019-04-30 15:15:37	2019-05-10 02:31:12	14394.8425
4	NehmatVerma	52/20 Thaman Gulbarga-366179	Day	2015-06-01 11:22:44	2015-06-26 05:27:48	18337.4186
5	ArmaanJohal	H.No. 71 Basu Circle, Raiganj 310096	Day	2015-12-06 08:32:13	2015-12-14 19:25:12	13656.1320
6	KismatChada	85/504 ShereChowkMehsana 483469	Day	2012-12-21 18:06:03	2012-12-28 05:28:15	17314.9652

#Create Dimension Time

CREATE TABLE DimTime (Date date primary key, Day_of_week int, Day_of_month int, Week int, Month int, Year year, Holiday varchar(1));

#Insert Values in Dimension Time

INSERT INTO DimTime VALUES ('2016-03-08', 2, 8, 10, 3, 2016, 'N'),

('2018-06-27', 3, 27, 26, 6, 2018, 'N'),

('2019-04-30', 2, 30, 18, 4, 2019, 'N'),

('2015-06-01', 1, 1, 22, 6, 2015, 'N'),

('2015-12-06', 7, 6, 49, 12, 2015, 'N'),

('2012-12-21', 5, 21, 51, 12, 2012, 'N');

 # Displaying Dimension Time

select * from DimTime;

Date	Day_of_week	Day_of_month	Week	Month	Year	Holiday
2012-12-21	5	21	51	12	2012	N
2015-06-01	1	1	22	6	2015	N
2015-12-06	7	6	49	12	2015	N
2016-03-08	2	8	10	3	2016	N
2018-06-27	3	27	26	6	2018	N
2019-04-30	2	30	18	4	2019	N

 # Create Fact Table

CREATE TABLE FactHotelOccupancy (HotelID int references DimHotel(HotelID), RoomID int references DimRoom(RoomID), CustomerID int references DimCustomer(CustomerID), Date date references DimTime(Date), No_of_occupied_rooms int, No_of_vacant_rooms int, Revenue decimal(19,4), primary key (HotelID, RoomID, CustomerID));

#Insert Values in Fact Table

```
INSERT INTO FactHotelOccupancy VALUES (1, 2, 6, '2012-12-21', 60, 240, 11161382.6162),
(2, 5, 4, '2015-06-01', 150, 50, 7696742.5189),
(3, 3, 1, '2016-03-08', 325, 175, 3904503.3812),
(4, 4, 3, '2019-04-30', 236, 214, 4981383.1735),
(5, 6, 2, '2018-06-27', 284, 416, 7735696.0160),
(6, 1, 5, '2015-12-06', 657, 43, 9987647.5030);
```

 # Display Fact Table

```
select * from FactHotelOccupancy;
```

Hotel ID	Room ID	Customer ID	Date	No_of_occupied_rooms	No_of_vacant_rooms	Revenue
1	2	6	2012-12-21	60	240	11161382.6162
2	5	4	2015-06-01	150	50	7696742.5189
3	3	1	2016-03-08	325	175	3904503.3812
4	4	3	2019-04-30	236	214	4981383.1735
5	6	2	2018-06-27	284	416	7735696.0160
6	1	5	2015-12-06	657	43	9987647.5030

▶ Experiment No. 3 : Implementation of OLAP Operations

In this experiment, we will perform the following operations on the dimension tables and fact table creating in Experiment 2. For simplicity, we have only shown 1 query with result for each operation.

Roll-up Operation

```
select Region, Country, State, City, sum(Revenue) from DimHotel inner join FactHotelOccupancy on DimHotel.HotelID = FactHotelOccupancy.HotelID group by Region, Country, State, City with rollup;
```

Region	Country	State	City	sum(Revenue)
17700 87 Ave NW	Canada	Alberta	Edmonton	4981383.1735
17700 87 Ave NW	Canada	Alberta		4981383.1735
17700 87 Ave NW	Canada			4981383.1735
70 RajaphillaMawatha	Sri Lanka	Kandy	70 RajaphillaMawatha	9987647.5030
70 RajaphillaMawatha	Sri Lanka	Kandy		9987647.5030
70 RajaphillaMawatha	Sri Lanka			9987647.5030
AvcilarMahallesi	Turkey	Goreme	IlkokulSokak No 16	7735696.0160
AvcilarMahallesi	Turkey	Goreme		7735696.0160
AvcilarMahallesi	Turkey			7735696.0160

Region	Country	State	City	sum(Revenue)
Edeforsvag 2 A	Sweden	Harads	Vid väg 97	11161382.6162
Edeforsvag 2 A	Sweden	Harads		11161382.6162
Edeforsvag 2 A	Sweden			11161382.6162
Kiilopaantie 9	Finland	Saariselka	Kiilopaantie 9	7696742.5189
Kiilopaantie 9	Finland	Saariselka		7696742.5189
Kiilopaantie 9	Finland			7696742.5189
Koitobos Rd	Kenya	Nairobi	Koitobos Rd	3904503.3812
Koitobos Rd	Kenya	Nairobi		3904503.3812
Koitobos Rd	Kenya			3904503.3812
Koitobos Rd				3904503.3812

Drill Down Operation

select Country, StarRating,sum(Revenue) from DimHotel inner join FactHotelOccupancy on DimHotel.HotelID = FactHotelOccupancy.HotelID where Country in ('Canada','Kenya') and StarRating in (1,2,3,4,5) group by StarRating with rollup;

Country	StarRating	sum(Revenue)
Kenya	3	3904503.3812
Canada	4	4981383.1735

Slicing Operation

Select Country, HotelType, sum(revenue) from DimHotel inner join FactHotelOccupancy on DimHotel.HotelID = FactHotelOccupancy.HotelID where HotelType='Hotel' GROUP BY Country ;

Country	HotelType	sum(Revenue)
Finland	Hotel	7696742.5189
Canada	Hotel	4981383.1735
Turkey	Hotel	7735696.0160
Sri Lanka	Hotel	9987647.5030

Dicing Operation

Select Country, HotelType, sum(revenue) from DimHotel inner join FactHotelOccupancy on DimHotel.HotelID = FactHotelOccupancy.HotelID where HotelType='Hotel' AND Country = 'Canada' GROUP BY Country ;

Country	HotelType	sum(Revenue)
Canada	Hotel	4981383.1735

PIVOT Operation

```
SELECT EXTRACT(YEAR  FROM Date) year
    , EXTRACT(MONTH FROM Date) month
    , SUM(Revenue)                  revenue
  FROM FactHotelOccupancy
 GROUP BY EXTRACT(YEAR  FROM Date)
       , EXTRACT(MONTH FROM Date);
SELECT year
    , SUM(CASE WHEN month =  1 THEN Revenue END) jan_revenue
    , SUM(CASE WHEN month =  2 THEN Revenue END) feb_revenue
    , SUM(CASE WHEN month =  3 THEN Revenue END) mar_revenue
    , SUM(CASE WHEN month =  4 THEN Revenue END) apr_revenue
    , SUM(CASE WHEN month =  5 THEN Revenue END) may_revenue
    , SUM(CASE WHEN month =  6 THEN Revenue END) jun_revenue
    , SUM(CASE WHEN month =  7 THEN Revenue END) jul_revenue
    , SUM(CASE WHEN month =  8 THEN Revenue END) aug_revenue
    , SUM(CASE WHEN month =  9 THEN Revenue END) sep_revenue
    , SUM(CASE WHEN month =  10 THEN Revenue END) oct_revenue
    , SUM(CASE WHEN month =  11 THEN Revenue END) nov_revenue
    , SUM(CASE WHEN month = 12 THEN Revenue END) dec_revenue
  FROM (SELECT FactHotelOccupancy.*
           , EXTRACT(YEAR  FROM Date) year
           , EXTRACT(MONTH FROM Date) month
        FROM FactHotelOccupancy
      ) invoices
 GROUP BY year
```

Year	jan_revenue	feb_revenue	mar_revenue	apr_revenue	may_revenue	jun_revenue	jul_revenue	aug_revenue	sep_revenue	oct_revenue	nov_revenue	dec_revenue
2012												11161382.6162
2015						7696742.5189						9987647.5030
2016			3904503.3812									
2019				4981383.1735								
2018						7735696.0160						

▶ Experiment No. 4 : Implementation of Bayesian algorithm

The code below applies the Naive Bayes algorithm to the Iris flowers dataset. Given an observation, it predicts the class label.

Dataset: https://raw.githubusercontent.com/jbrownlee/Datasets/master/iris.csv

```
# Make Predictions with Naive Bayes on the Iris Dataset
from csv import reader
from math import sqrt
from math import exp
from math import pi

# Load a CSV file
def load_csv(filename):
	dataset = list()
	with open(filename, 'r') as file:
		csv_reader = reader(file)
		for row in csv_reader:
			if not row:
				continue
			dataset.append(row)
	return dataset

# Convert string column to float
def str_column_to_float(dataset, column):
	for row in dataset:
		row[column] = float(row[column].strip())

# Convert string column to integer
def str_column_to_int(dataset, column):
	class_values = [row[column] for row in dataset]
	unique = set(class_values)
	lookup = dict()
	for i, value in enumerate(unique):
		lookup[value] = i
		print('[%s] => %d' % (value, i))
	for row in dataset:
		row[column] = lookup[row[column]]
	return lookup

# Split the dataset by class values, returns a dictionary
def separate_by_class(dataset):
	separated = dict()
	for i in range(len(dataset)):
		vector = dataset[i]
		class_value = vector[-1]
		if (class_value not in separated):
```

```
            separated[class_value] = list()
        separated[class_value].append(vector)
    return separated

# Calculate the mean of a list of numbers
def mean(numbers):
    return sum(numbers)/float(len(numbers))

# Calculate the standard deviation of a list of numbers
def stdev(numbers):
    avg = mean(numbers)
    variance = sum([(x-avg)**2 for x in numbers]) / float(len(numbers)-1)
    return sqrt(variance)

# Calculate the mean, stdev and count for each column in a dataset
def summarize_dataset(dataset):
    summaries = [(mean(column), stdev(column), len(column)) for column in zip(*dataset)]
    del(summaries[-1])
    return summaries

# Split dataset by class then calculate statistics for each row
def summarize_by_class(dataset):
    separated = separate_by_class(dataset)
    summaries = dict()
    for class_value, rows in separated.items():
        summaries[class_value] = summarize_dataset(rows)
    return summaries

# Calculate the Gaussian probability distribution function for x
def calculate_probability(x, mean, stdev):
    exponent = exp(-((x-mean)**2 / (2 * stdev**2 )))
    return (1 / (sqrt(2 * pi) * stdev)) * exponent

# Calculate the probabilities of predicting each class for a given row
def calculate_class_probabilities(summaries, row):
    total_rows = sum([summaries[label][0][2] for label in summaries])
    probabilities = dict()
    for class_value, class_summaries in summaries.items():
        probabilities[class_value] = summaries[class_value][0][2]/float(total_rows)
        for i in range(len(class_summaries)):
            mean, stdev, _ = class_summaries[i]
            probabilities[class_value] *= calculate_probability(row[i], mean, stdev)
    return probabilities

# Predict the class for a given row
def predict(summaries, row):
```

```
    probabilities = calculate_class_probabilities(summaries, row)
    best_label, best_prob = None, -1
    for class_value, probability in probabilities.items():
        if best_label is None or probability >best_prob:
            best_prob = probability
            best_label = class_value
    return best_label

# Make a prediction with Naive Bayes on Iris Dataset
filename = 'C:/Users/lenovo/Documents/iris.csv'
dataset = load_csv(filename)
for i in range(len(dataset[0])-1):
    str_column_to_float(dataset, i)
# convert class column to integers
str_column_to_int(dataset, len(dataset[0])-1)
# fit model
model = summarize_by_class(dataset)
# define a new record
row = [5.7,2.9,4.2,1.3]
# predict the label
label = predict(model, row)
print('Data=%s, Predicted: %s' % (row, label))
```

Output :

```
[Iris-virginica] => 0
[Iris-versicolor] => 1
[Iris-setosa] => 2
 Data=[5.7, 2.9, 4.2, 1.3], Predicted: 1
```

Running the data first summarizes the mapping of class labels to integers and then fits the model on the entire dataset.

There are three class labels. 0,1 & 2. In the output, when a new observation is defined, a class label is predicted. Here, our observation is predicted as belonging to class 1 which is "**Iris-versicolor**".

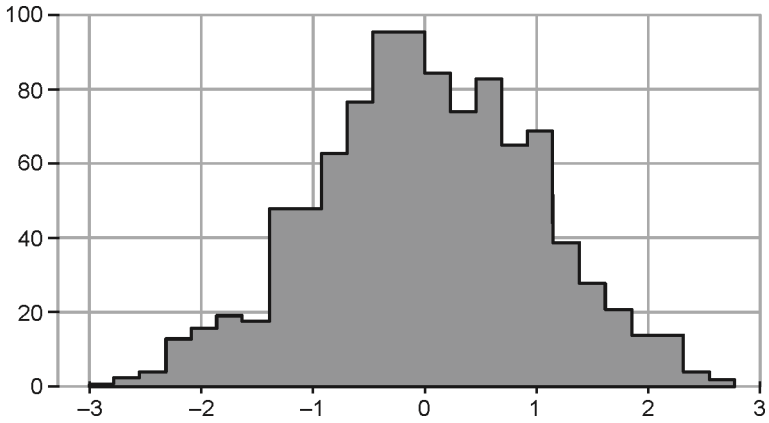
▶ Experiment No. 5 : Implementation of Data Discretization & Visualization

Data discretization will map numerical variables onto discrete values. The discretization transform is available in the scikit-learn Python machine learning library via the KBinsDiscretizer class.

```
# demonstration of the discretization transform
from numpy.random import randn
from sklearn.preprocessing import KBinsDiscretizer
from matplotlib import pyplot
# generate gaussian data sample
data = randn(1000)
# histogram of the raw data
pyplot.hist(data, bins=25)
pyplot.show()
# reshape data to have rows and columns
```

```
data = data.reshape((len(data),1))
# discretization transform the raw data
# The “strategy” argument controls the manner in which the input variable is divided, as either
# “uniform,” “quantile,” or “kmeans.” The “n_bins” argument controls the number of bins that will be
# created and must be set based on the choice of strategy. The “encode” argument controls whether
# the transform will map each value to an integer value by setting “ordinal” or a one-hot encoding
# “onehot.”
kbins = KBinsDiscretizer(n_bins=10, encode='ordinal', strategy='uniform')
data_trans = kbins.fit_transform(data)
# summarize first few rows
print(data_trans[:10, :])
# histogram of the transformed data
pyplot.hist(data_trans, bins=10)
pyplot.show()
```

Output :



(1G2)**Fig. L.2 : Histogram of data with Gaussian Distribution**

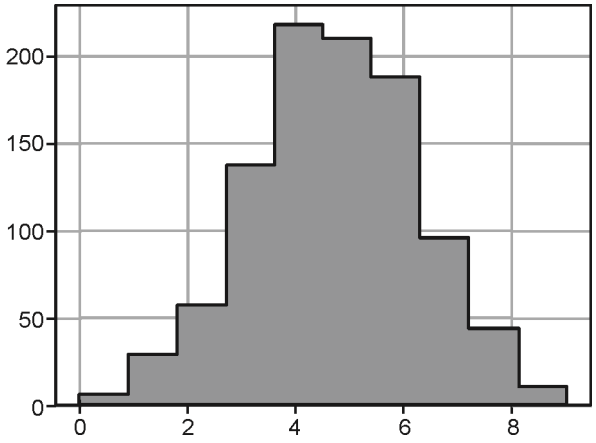
Running the example first creates a sample of 1,000 random Gaussian floating-point values and plots the data as a histogram.

Next the KBinsDiscretizer is used to map the numerical values to categorical values. We configure the transform to create 10 categories (0 to 9), to output the result in ordinal format (integers) and to divide the range of the input data uniformly.

A sample of the transformed data is printed, clearly showing the integer format of the data as expected.

```
[[5.]
 [3.]
 [2.]
 [6.]
 [7.]
 [5.]
 [3.]
 [4.]
 [4.]
 [2.]]
```

Finally, a histogram is created showing the 10 discrete categories and how the observations aredistributed across these groups, following the same pattern as the original data with a Gaussian shape.



(1G3)**Fig L.3 : Histogram of Transformed Data with Discrete Categories**

▶ Experiment No. 6 : Data Pre-processing using WEKA, Classification, Clustering, Association Rule mining on data sets using WEKA

WEKA :

- Weka (Waikato Environment for Knowledge Analysis) is developed at University of Waikato, New Zealand. This tool is written in JAVA.
- Weka is open source software issued under the GNU General Public License.
- Weka is a collection of machine learning algorithms for data mining tasks. It contains tools for data pre-processing, attribute selection, classification, regression, clustering, association rules mining, and visualization.
- This tool can be downloaded for appropriate operating system and installed according to the instructions given on the following URL:

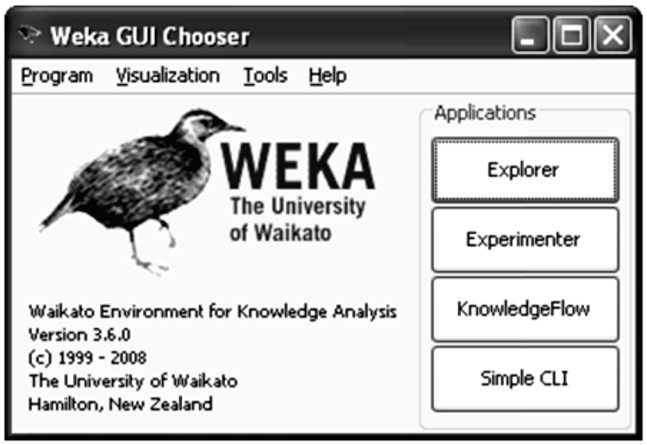
  https://waikato.github.io/weka-wiki/downloading_weka/.

WEKA Interface:

WEKA interface has four tabs:

- Explorer
- Knowledge flow
- Experimenter
- Command Line Interface (CLI)

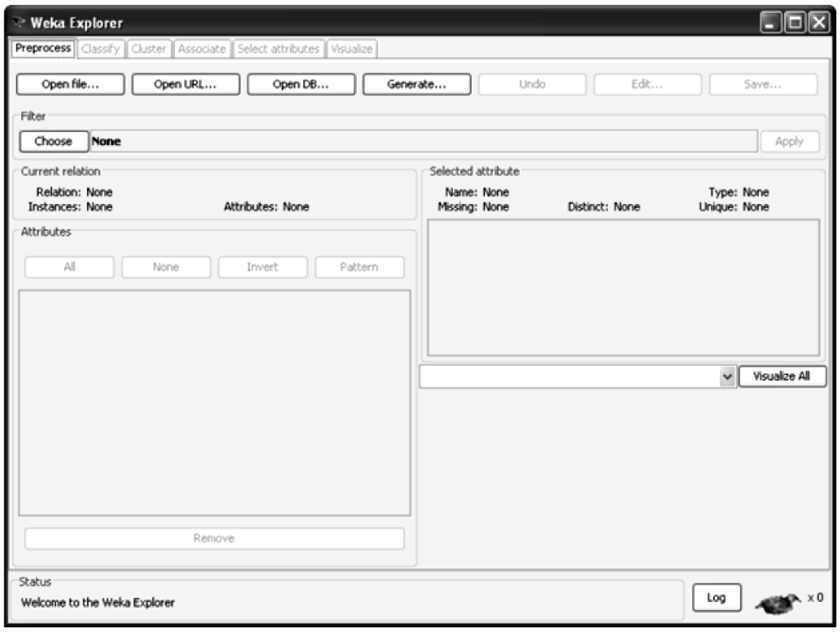
After installing and starting up Weka, you will have the options of starting up the “Simple CLI (Command-Line Interface)”, the “Explorer”, the “Experimenter” and the “Knowledge Flow”. The Experimenter interface is a more powerful interface for manipulating, tracking, and analyzing experiments and used for large datasets. Explorer interface can be used to run quick experiments on small datasets. You have to click “Explorer” to bring up the Weka Knowledge Explorer Interface.



(1G4)**Fig. L.4 : WEKA Interface**

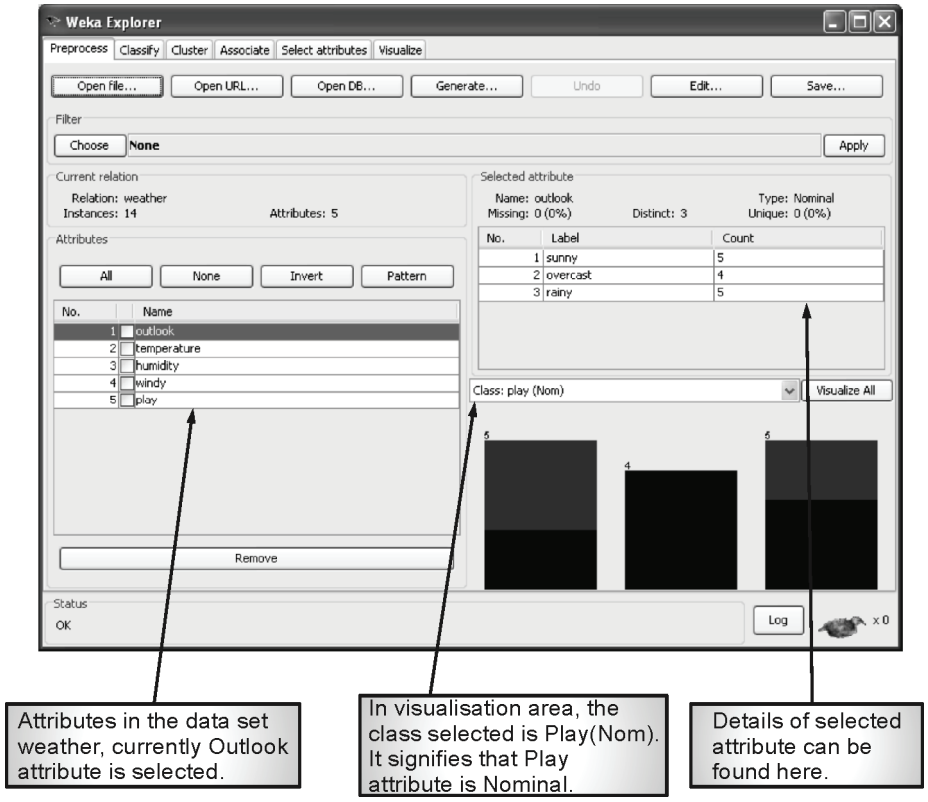
Weka Explorer

The Explorer interface has a number of tabs: "Preprocess", "Classify", "Cluster", "Associate", "Attribute selection" and "Visualize". These options are used to load and filter data that we're going to use in an experiment, to build and test a model for classification of our data, to create clusters of our data, to create association rules, to allow the automatic selection of features to create a reduced dataset and to visualize (graphically view) the data respectively. If you click on the "Explorer", the following screen would appear:



(1G5)**Fig. L.5 : WEKA Explorer**

Click on "Open file" to select a dataset. WEKA allows .csv file or .arff file as a datset. Let's select a dataset "weather.arff" which is already present in the data folder of WEKA.



(1G6)**Fig. L.6 : WEKA Preprocess tab**

Weka File Formats

File Formats like CSV (Comma SepartedValues : *.csv), Binary Serialized Instances (*.bsi) etc are supported by WEKA. The most commonly used file format is Attribute Relation File Format (ARFF). You can also convert .csv file to .arff using WEKA. Here is whether.arff file opened in notepad:

```
@relation weather
@attribute outlook {sunny, overcast, rainy}
@attribute temperature real
@attribute humidity real
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}
@data
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes
rainy,65,70,TRUE,no
overcast,64,65,TRUE,yes
sunny,72,95,FALSE,no
sunny,69,70,FALSE,yes
rainy,75,80,FALSE,yes
sunny,75,70,TRUE,yes
```

Name of the table/relation

attributes

Observations/samples

```
overcast,72,90,TRUE,yes
overcast,81,75,FALSE,yes
rainy,71,91,TRUE,no
```

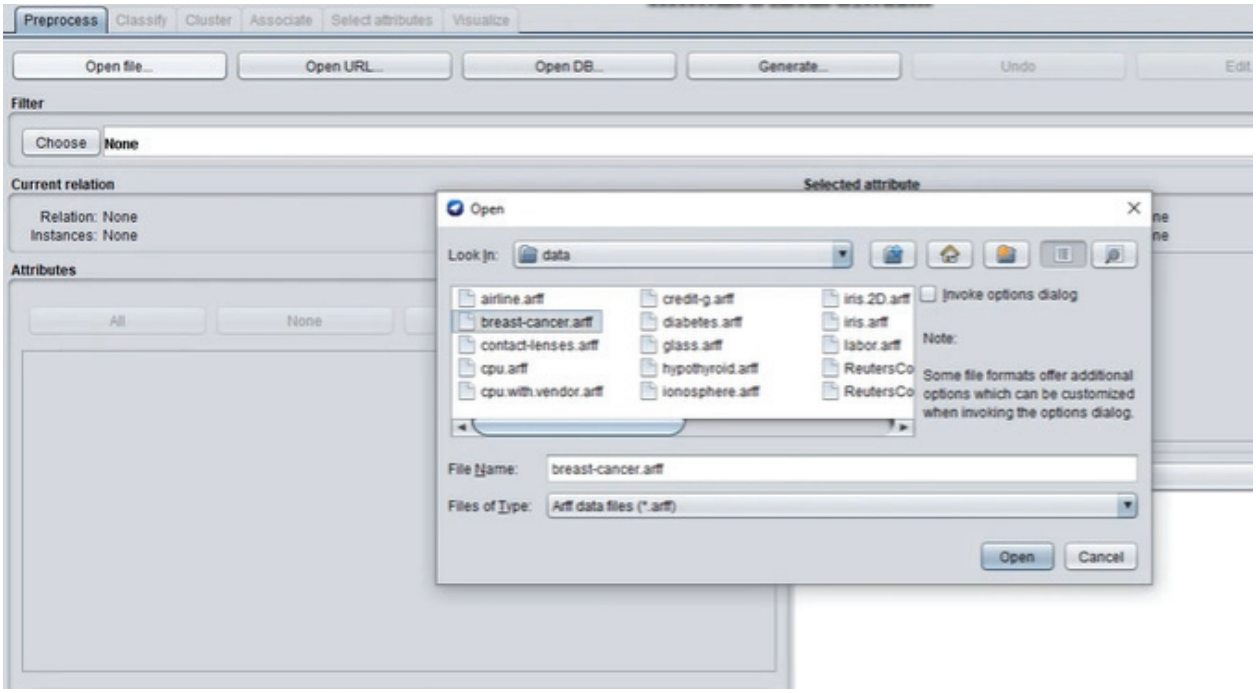
6.1 PRE-PROCESSING USING WEKA

Pre-processing the dataset is required so that the data mining tasks are efficiently applied. We will try here two approaches: Filling the missing values and Removing outliers.

6.1.1 Filling the Missing Values in the Dataset

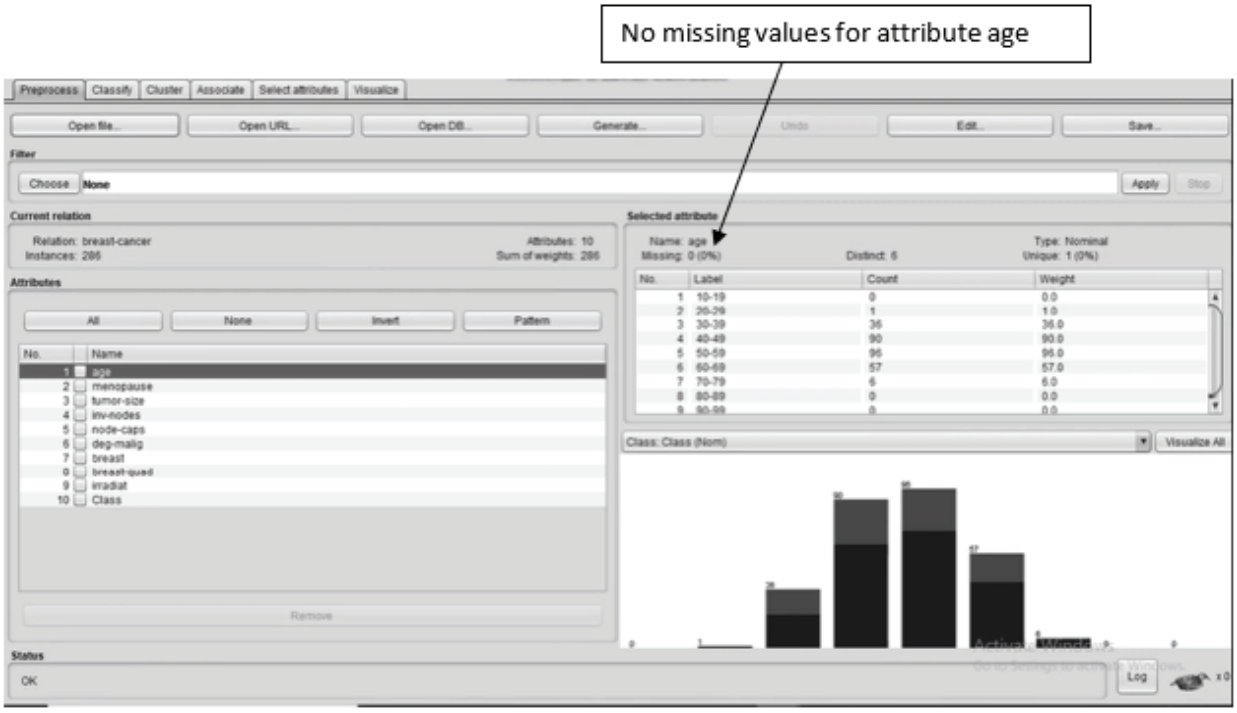
There are multiple ways to treat the missing values. You can simply ignore them and perform data mining task or else you can try them to fill with constant or mean/median value. Here, we will see, how WEKA can be used to fill in the missing values.

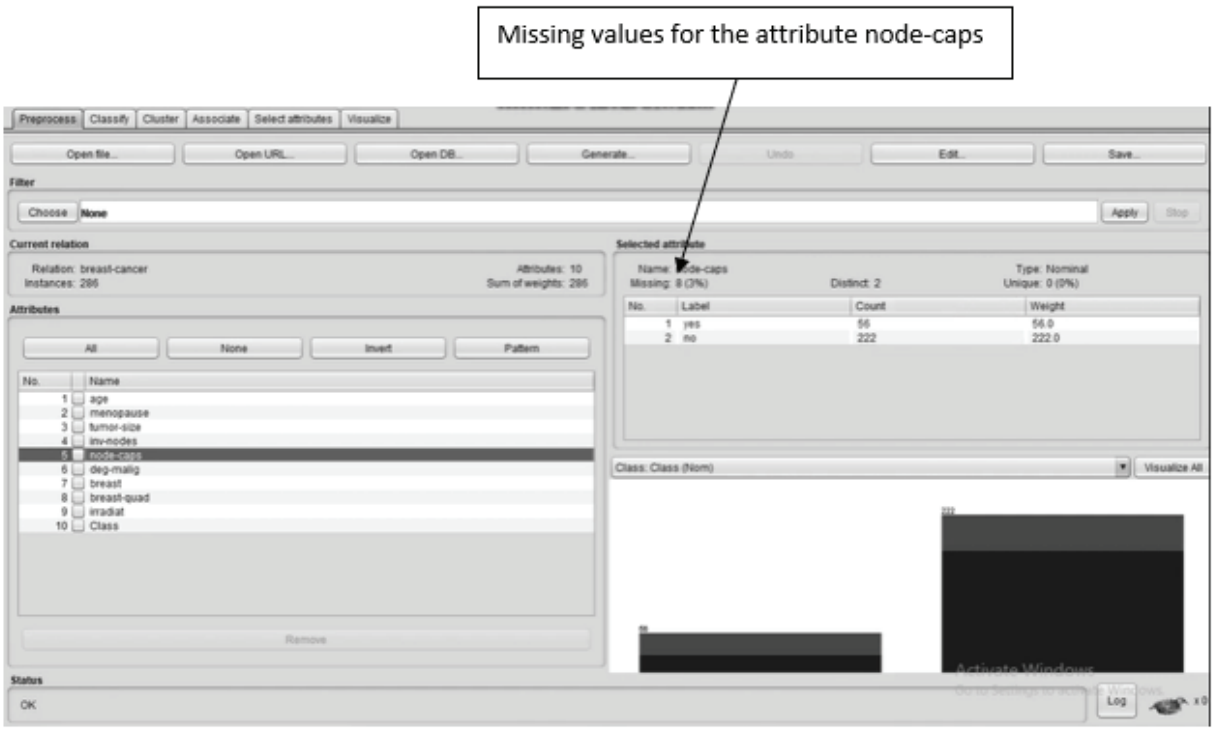
- **Step 1 : Select the dataset "breast-cancer".**



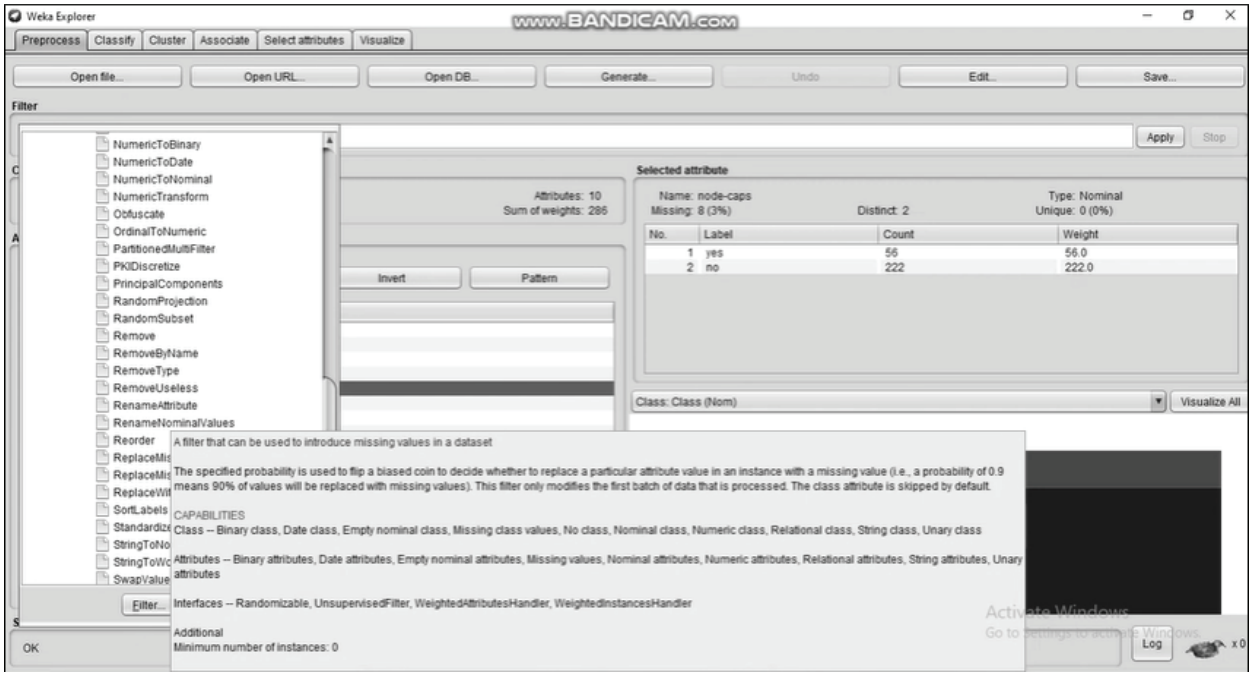
NOTES

- **Step 2: Check every attribute for missing values.**

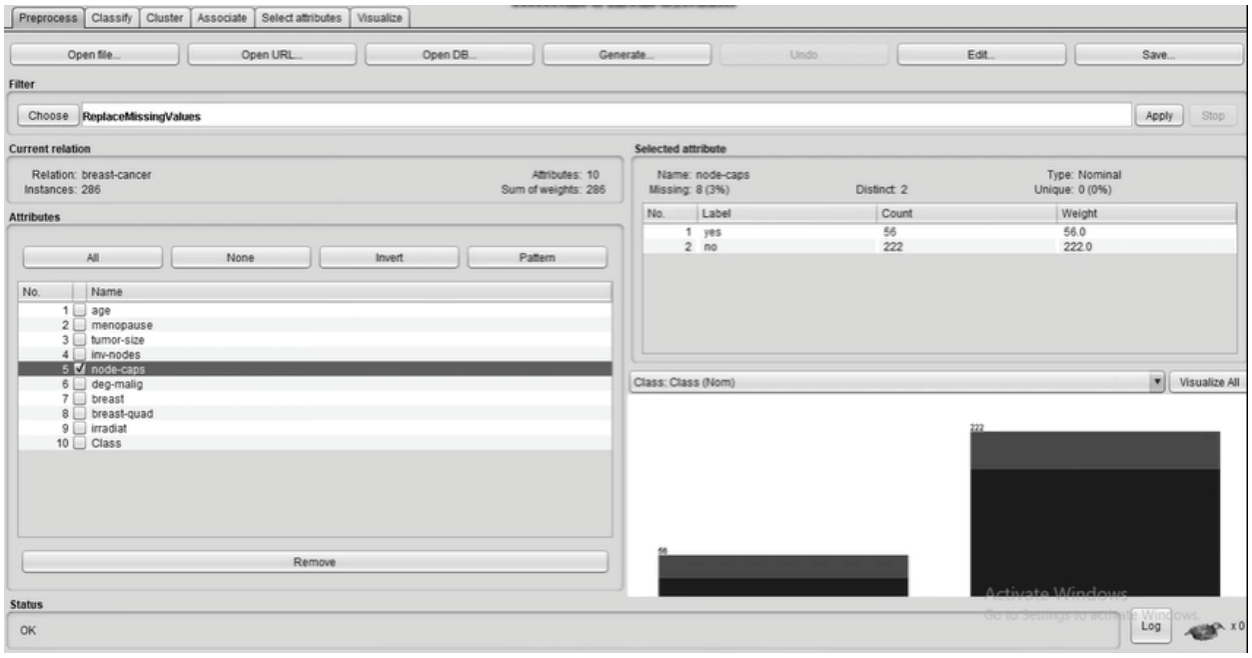




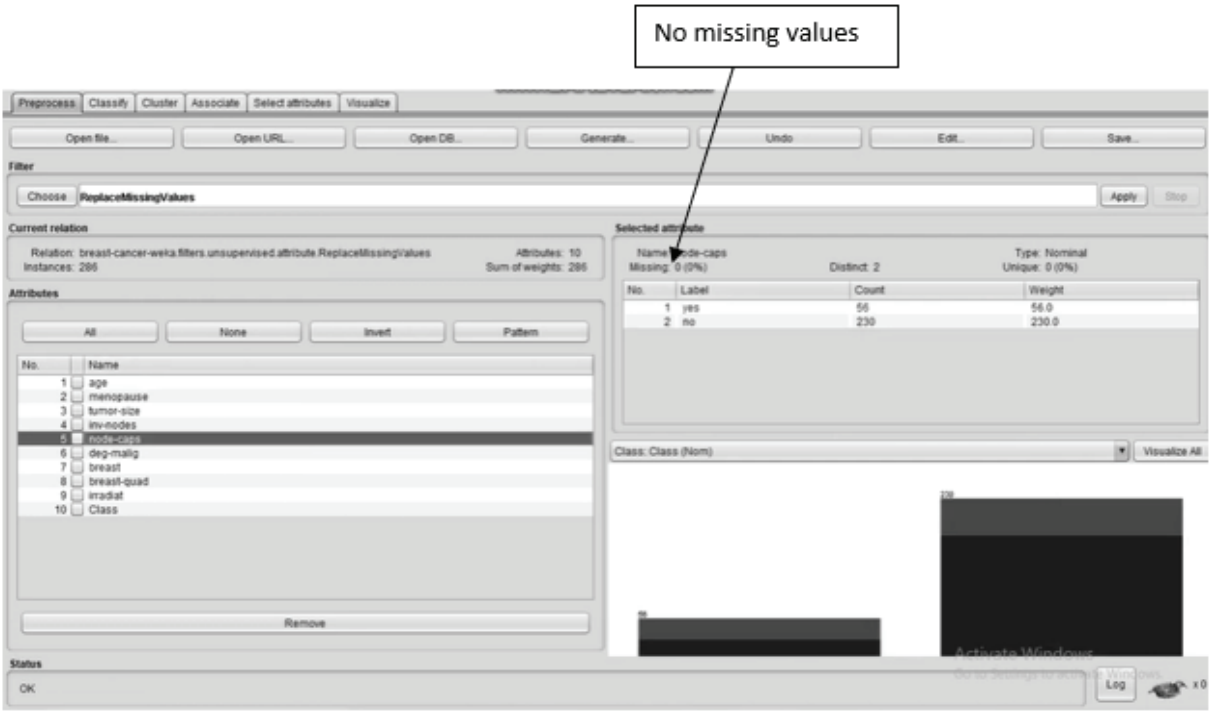
- **Step 3 : To fill the missing value with mean/median value, Select filter from unsupervised -> attribute ->ReplaceMissingValues.**



- **Step 4 : Select the attribute for which the filter is to be applied and click on Apply.**



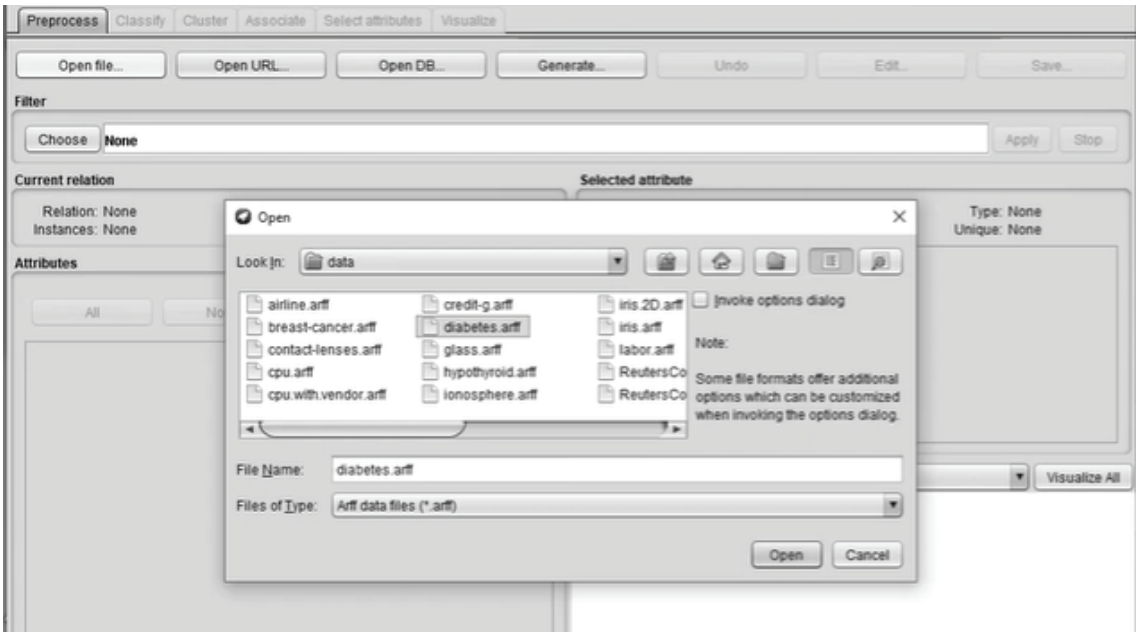
After this, we can see that all the missing values are filled by mean/median.



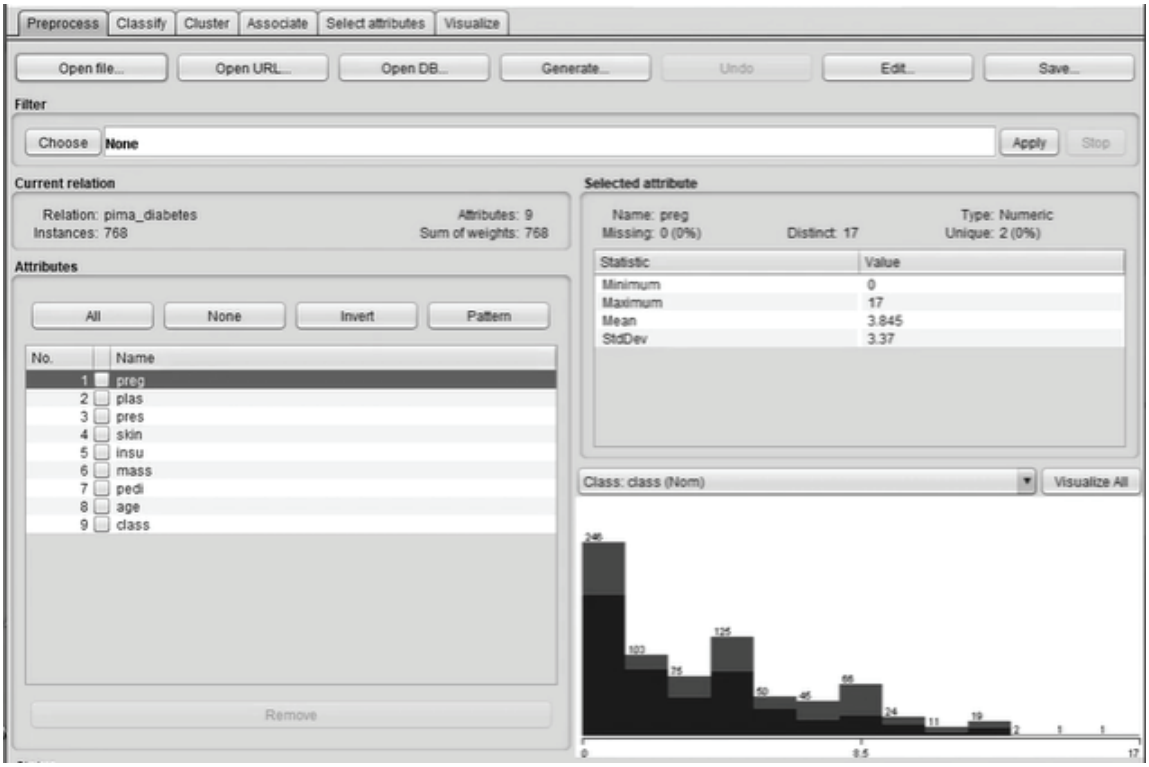
6.1.2 Removing Outliers from the Dataset

In this section, we will see outlier detection and removal. We will use Interquartile range to remove the outliers. You can check your arff file before applying the filter and after applying the filter to cross-check whether the outliers have been removed or not.

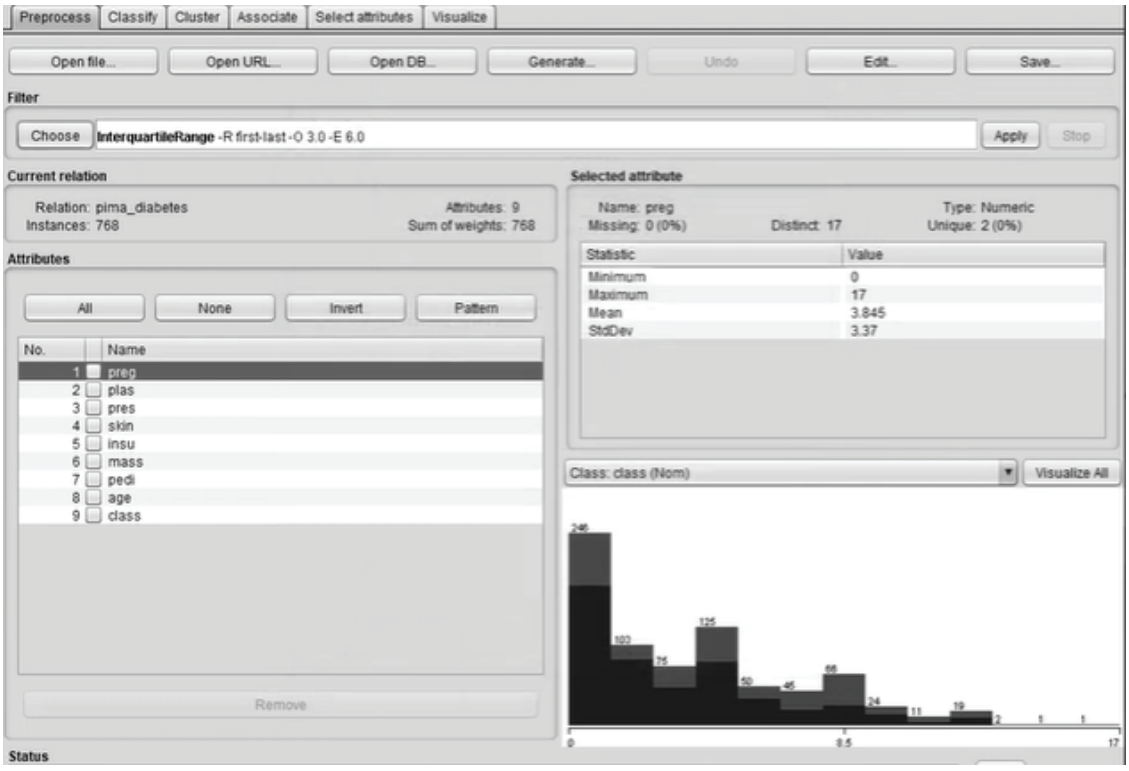
- **Step1 : Select the dataset. Here, we are choosing "diabetes.arff".**



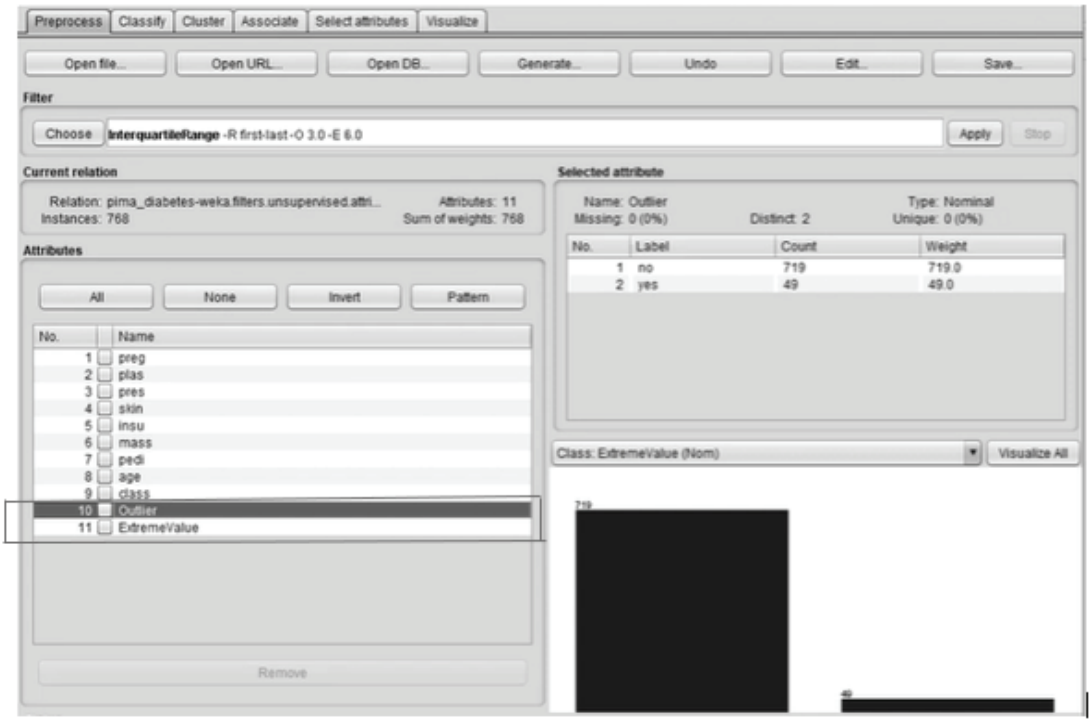
We can see, there are nine attributes in the dataset.



- **Step 2: To detect the outliers, From Filter section, go to unsupervised -> attribute ->InterquartileRange And click on Apply.**

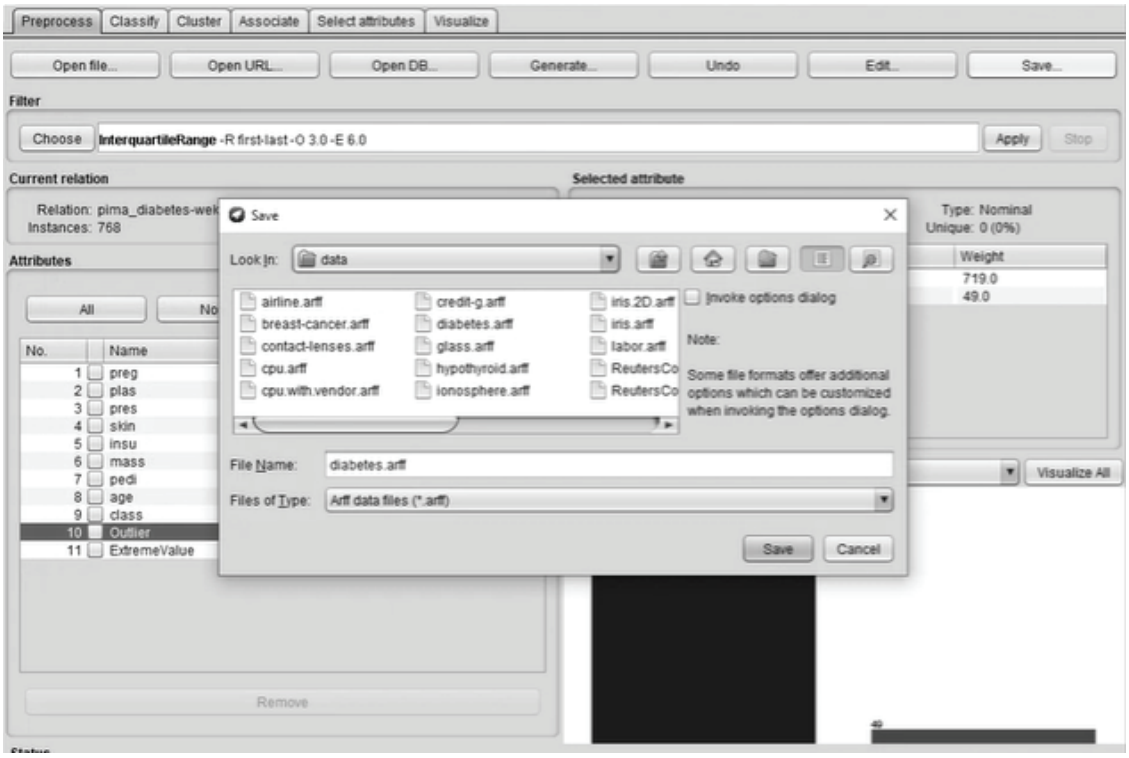


After this, we can see in the attribute section. Two more attributes are added: Outlier and ExtremeValue.

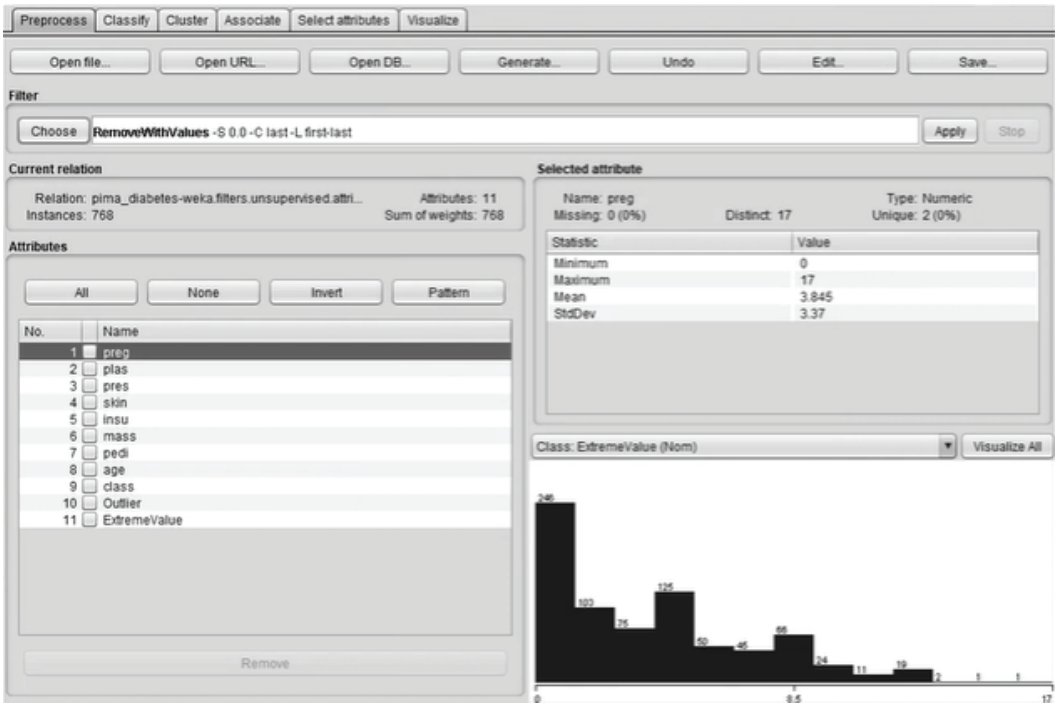


- **Step 3 : Check details of selected attribute "Outlier". There are two labels present: no and yes. Yes label tells us number of outliers present and no label tells us normal values. So, here 49 outlier values exist out of 768 observations. We need to remove these outliers.**

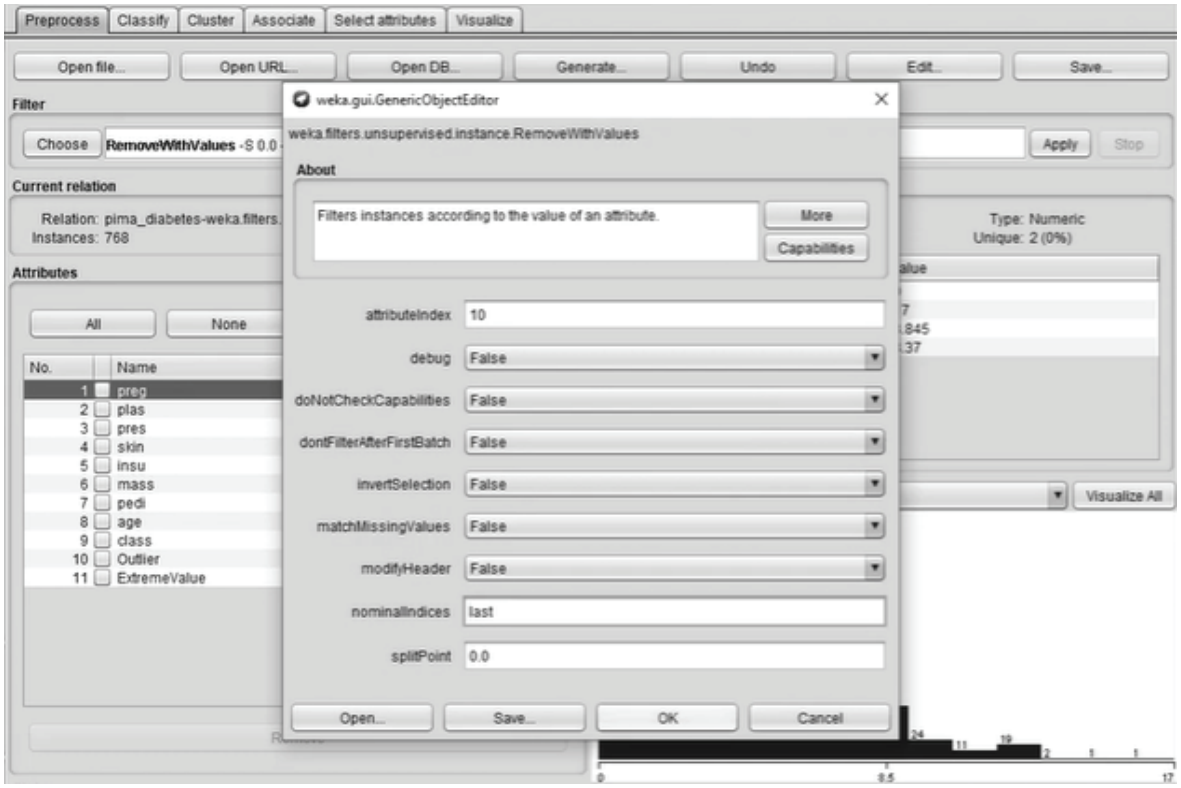
After applying InterquartileFilter, save your file first so that the changes are saved into the database. You can use different name and then you can compare the original and this file after removal of outliers.



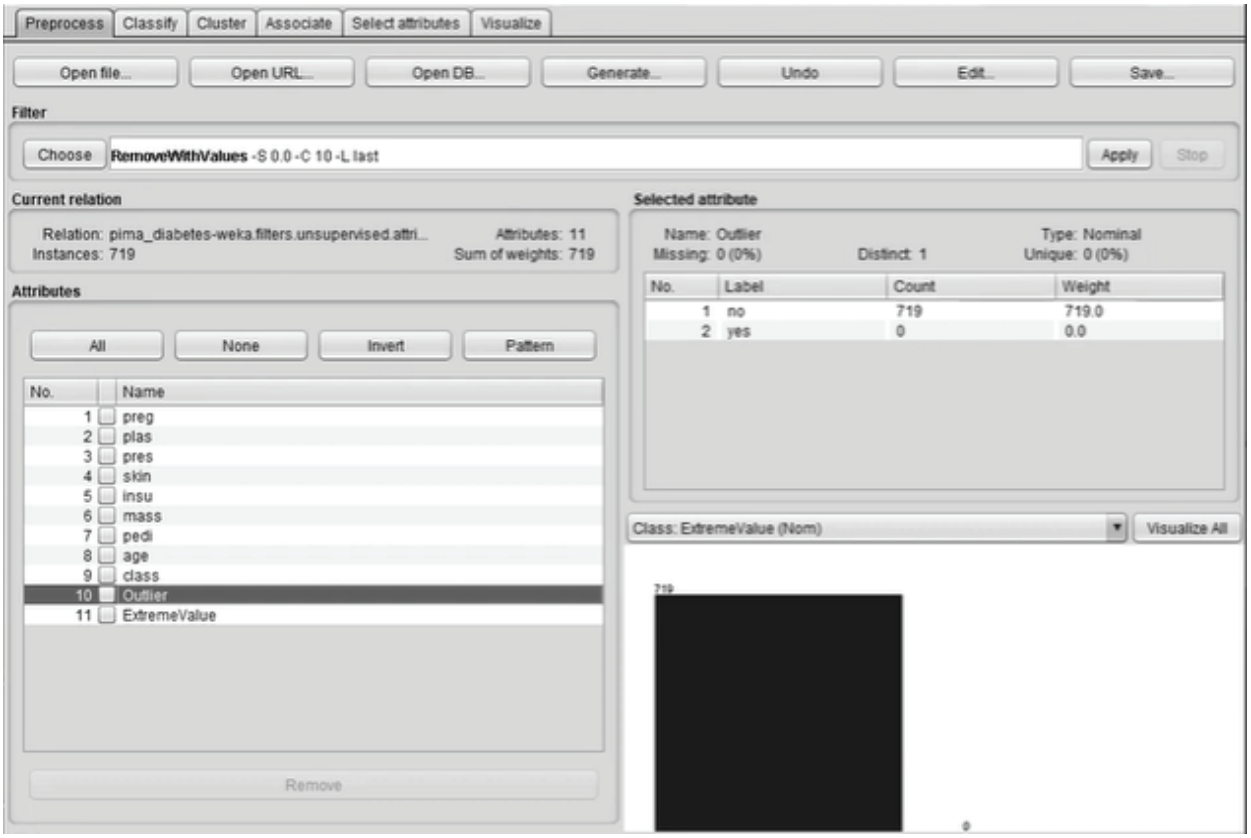
- **Step 4 : To remove the outliers, Select the newly saved file where we have added Outlier and ExtremValue attributes.**
- **Step 5 : In filter section, go to unsupervised -> instance ->RemovewithValues.**



Rightclick on RemovewithValues. You will see these settings. Select the attribute index as 10 as, Outlier is the 10th attribute and set nominalIndices to last. Click on Ok. Apply the filter.



- **Step 6 : Check the attribute "Outlier" again. We can see that there are no attributes present in Yes label now. This indicates, we have successfully removed the outliers.**



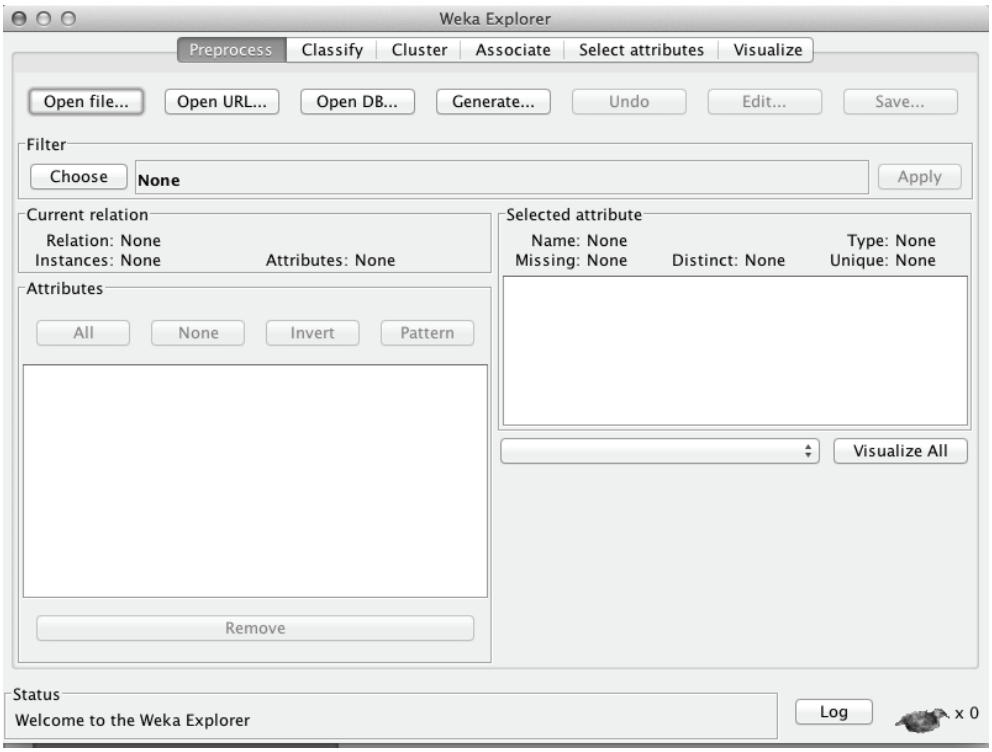
6.2 CLASSIFICATION USING WEKA

WEKA allows to perform classification using number of classifiers. In this section, we will see decision tree and Naïve Bayes approach.

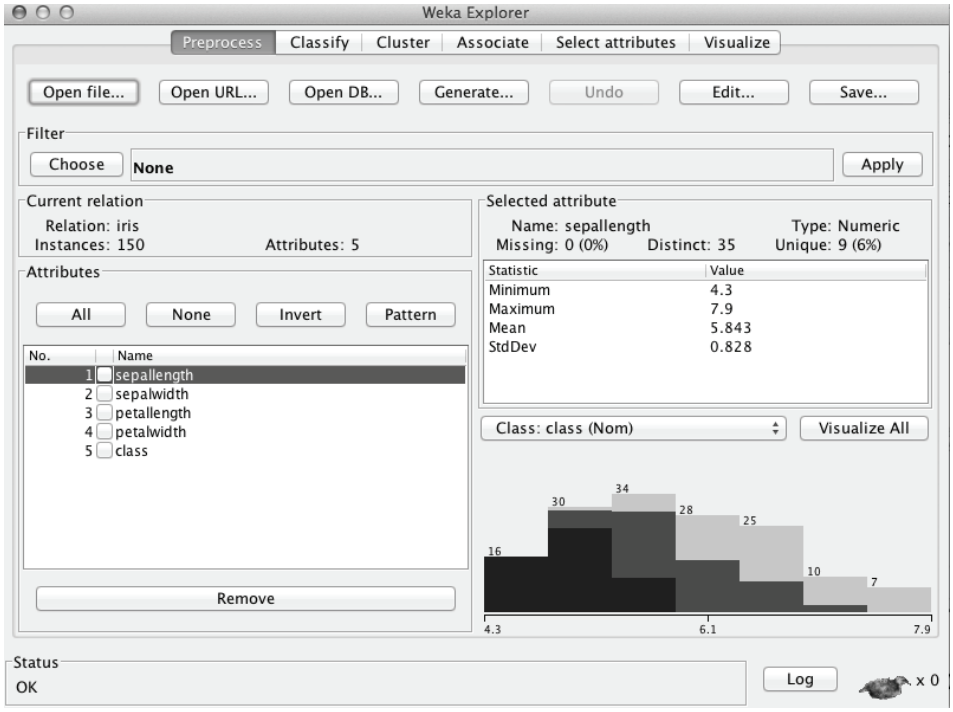
6.2.1 Decision Tree Classifier

- **Step 1: Open Weka, the following GUI should appear on your screen.**

- **Step 2: Click on the Explorer.**

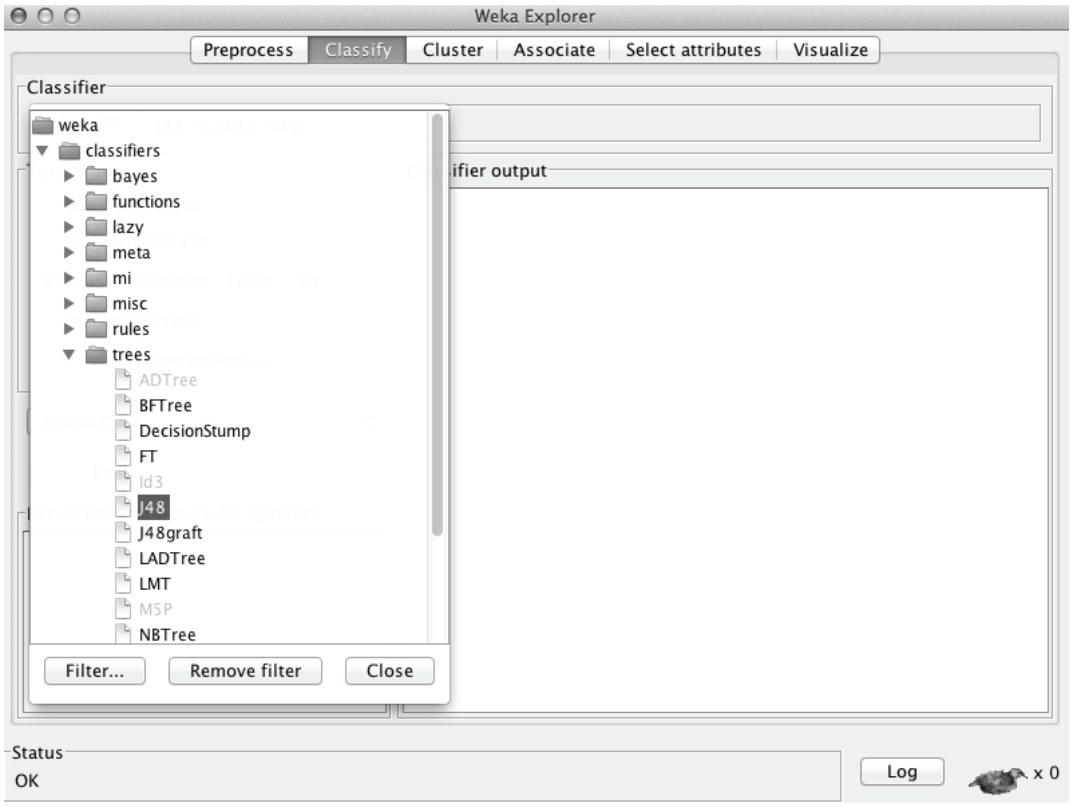


- **Step 3: Select iris.arffdataset from Open file section. The following screen would appear.**



You can observe that the Iris flower dataset contains 150 rows and 4 attributes: sepallength, sepalwidth, petallength, petalwidth and a class attribute for the species of iris flower (setosa, versicolor, virginica).

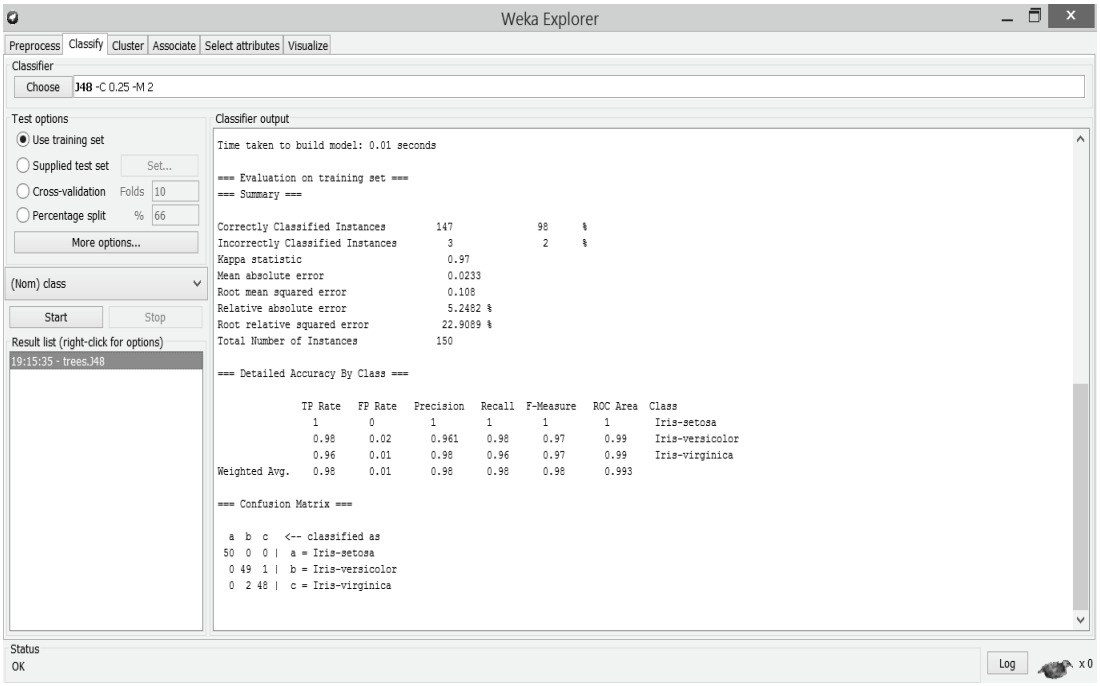
- **Step 4 :** Click on "Classify". Select the classification algorithm to be executed on the dataset from "Choose" option. Here, we have selected J48 algorithm under trees which creates decision tree. This classifier is actually C4.8 and is an extension to C4.5 algorithm.



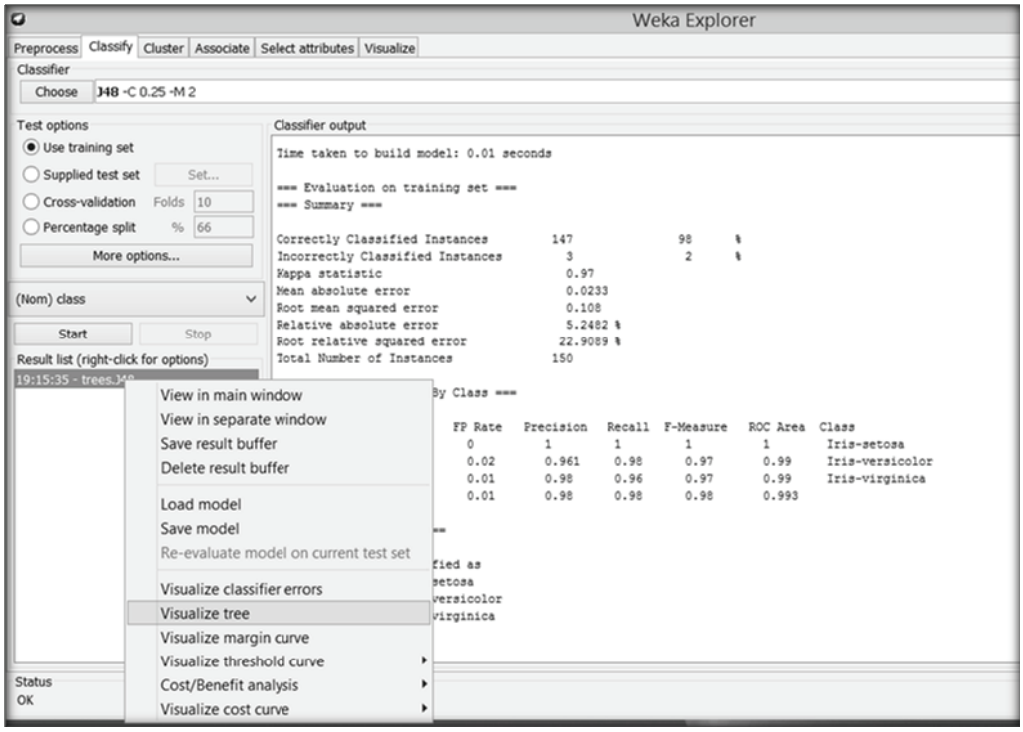
click on start to execute the algorithm. The output can be seen in the output window. From the output, following things can be observed:

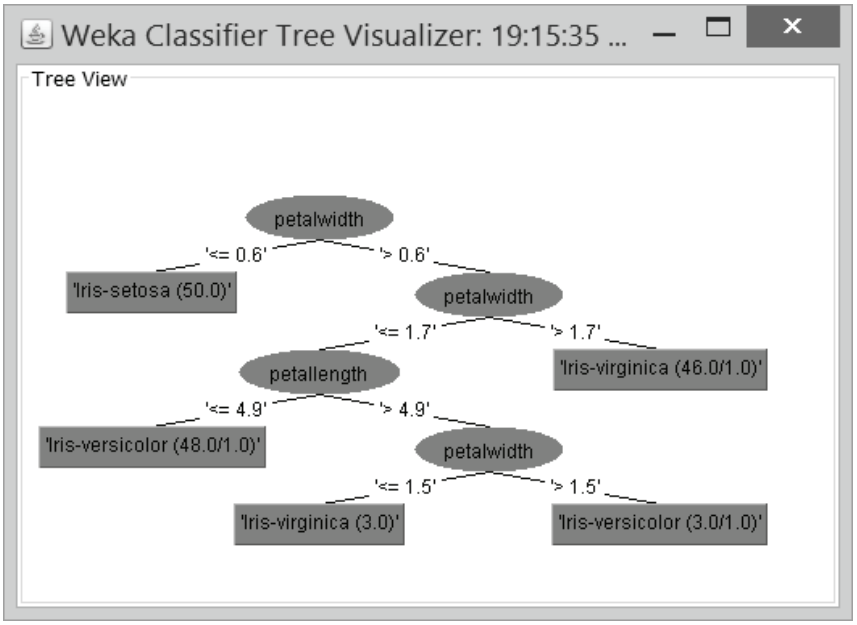
- We can see that there are 147 correctly classified instances out of 150 i.e. the accuracy of the model is 98%.
- The confusion matrix shows the table of actual classes compared with predicted classes. There are 50 true positives for Iris -Setosa and none of the observation is placed in other classes, so 0 errors. There is one observation where Iris-versicolor was classified as a Iris-virginica and two observations where a Iris-virginica was classified as a Iris-versicolor. So, there are total three errors.

NOTES



- **Step 5 : To visualize the tree, Right click on result list as shown below.**

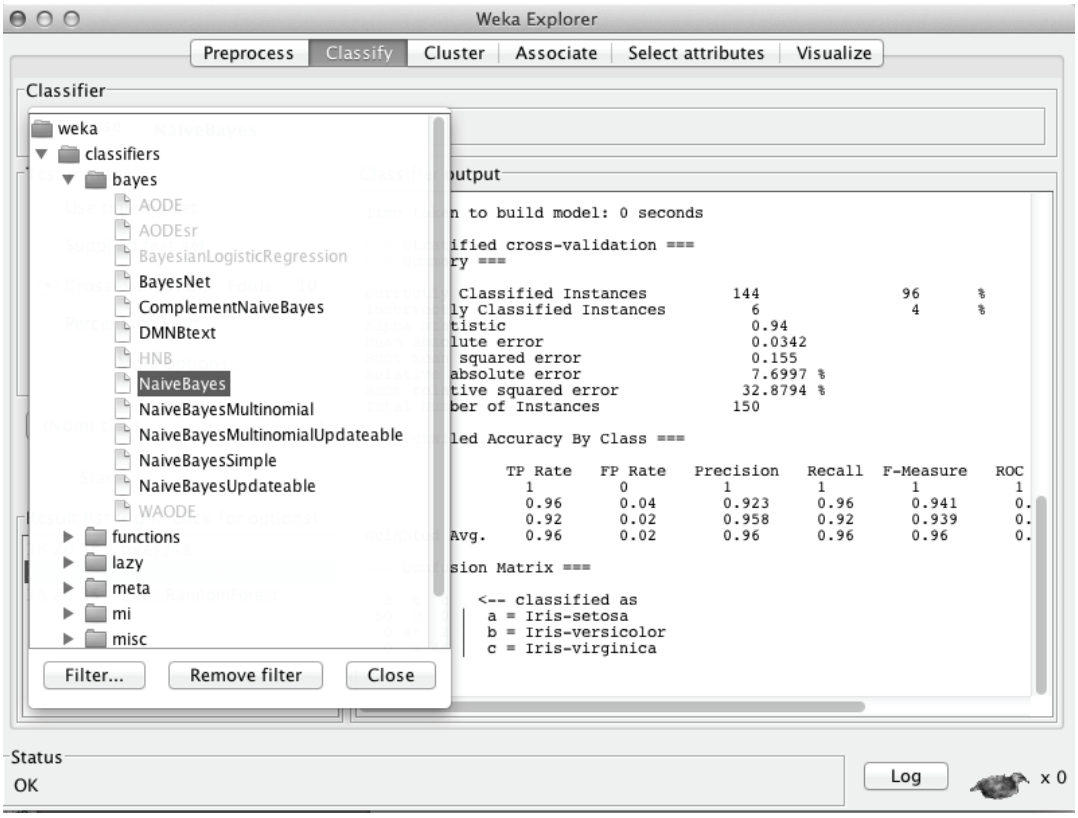




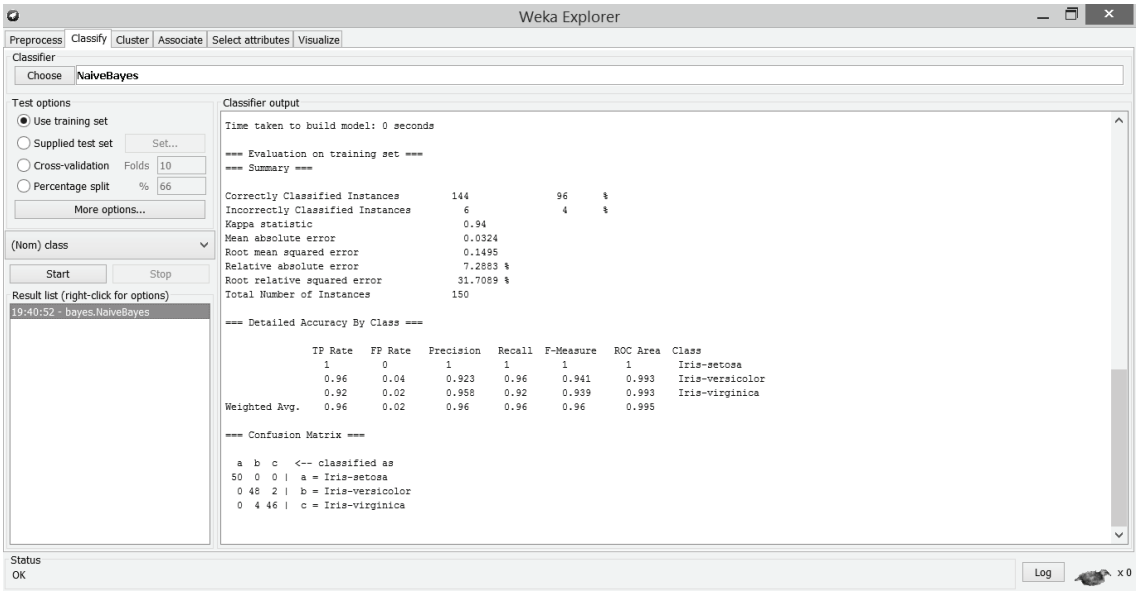
6.2.2 Naïve Bayes Classifier

Apply the first three steps we have applied in Decision Tree Classifier.

- **Step 4 : Select the Naïve Bayes classifier, under Classifiers –>Bayes -> Naïve Bayes and Click on Apply.**



The following output would appear:

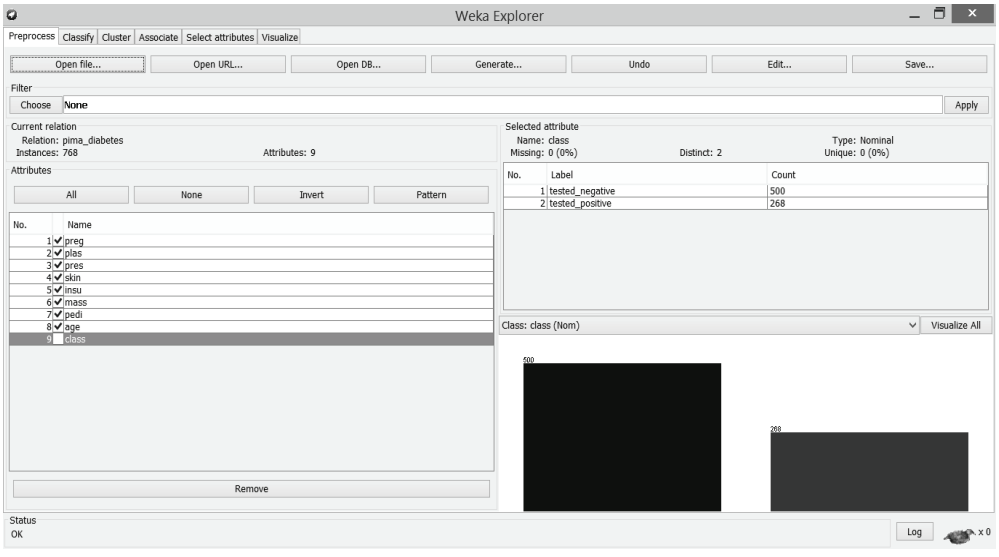


You can run multiple classifiers in this way and compare their accuracies.

⏭ 6.3 CLUSTERING USING WEKA

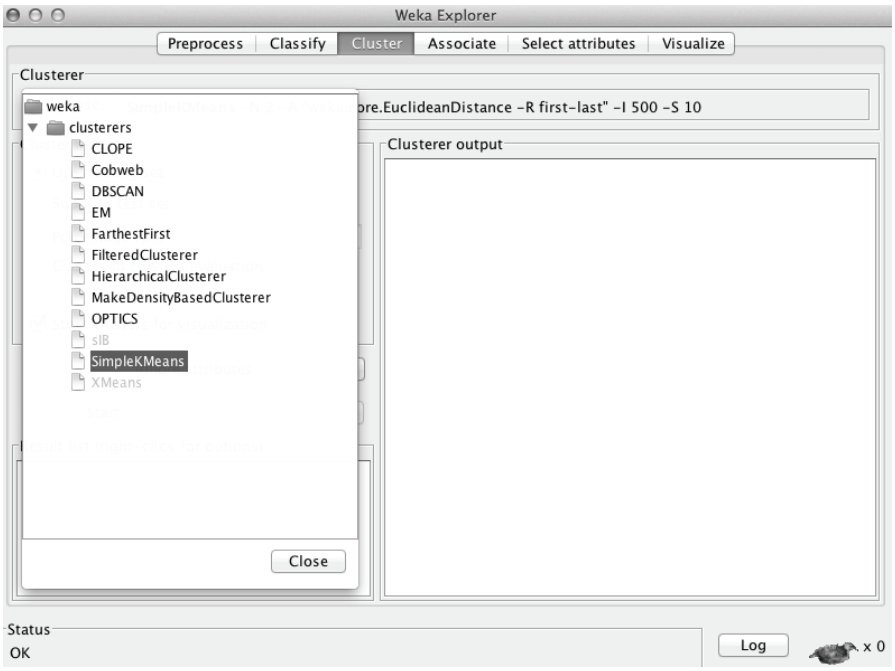
In this section, we will see Cluster tab of WEKA. There are multiple clustering algorithms which WEKA supports. We will apply simple k-means algorithm for clustering.

- **Step 1: Click on Cluster tab after selecting the dataset. We have selected diabetes.arff here.**

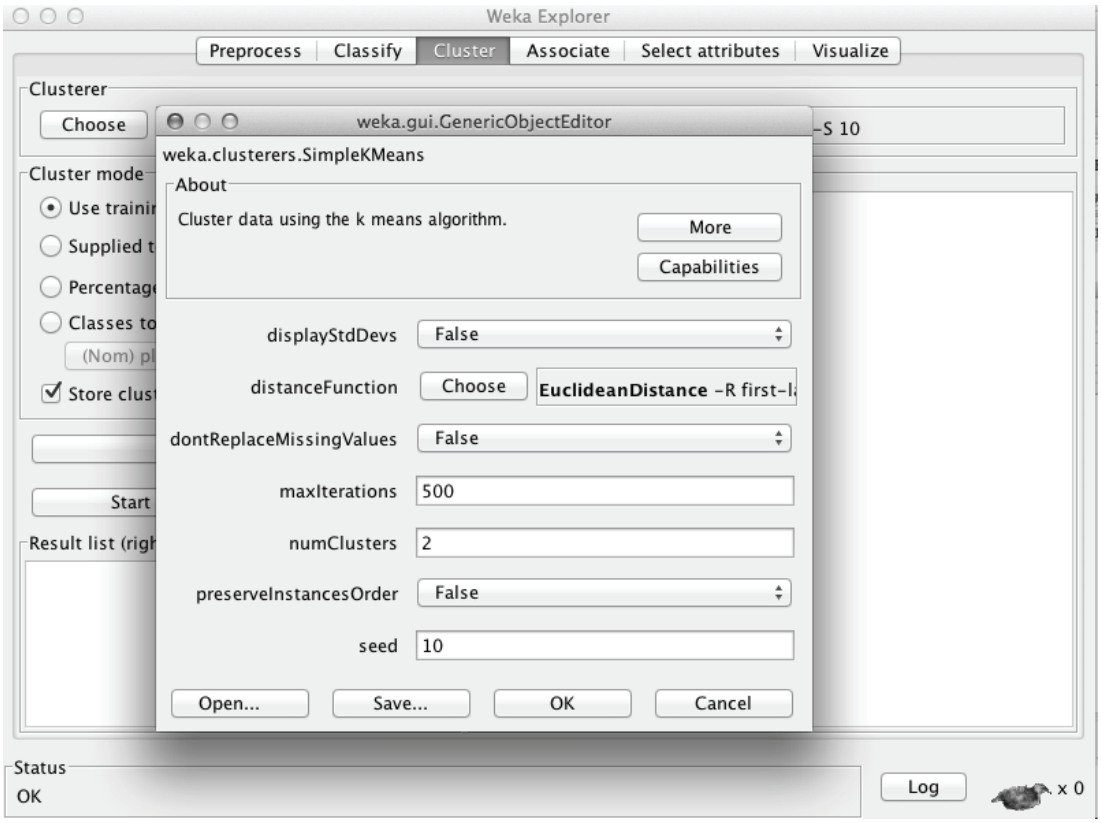


NOTES

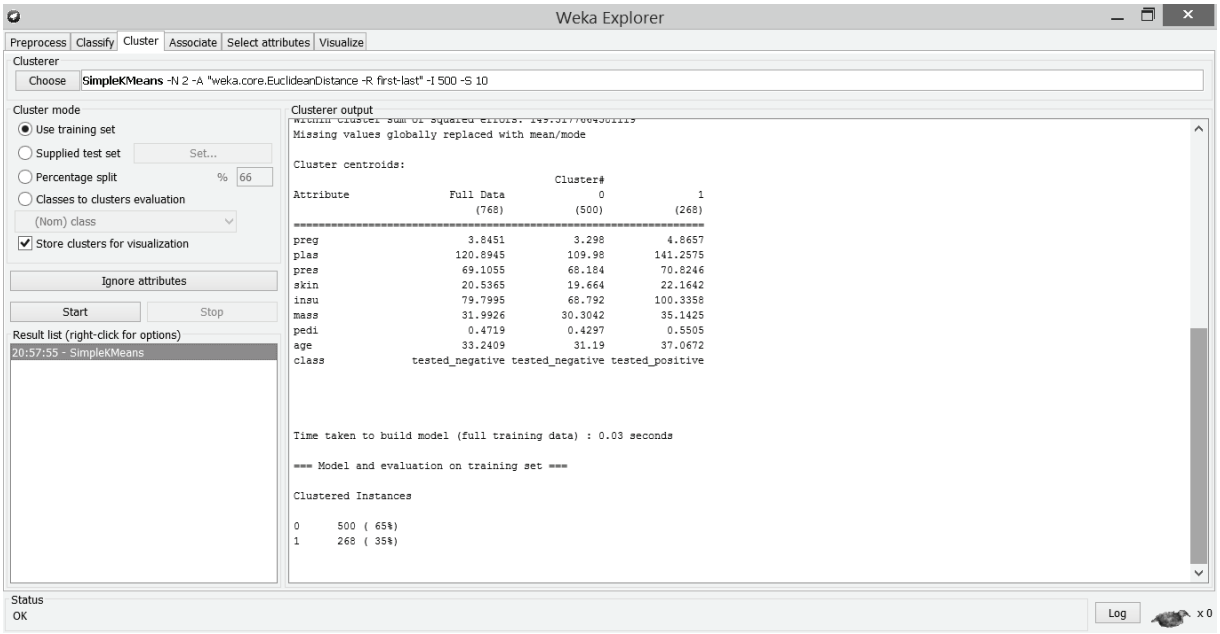
- **Step 2: Select the algorithm to be applied. Clusters ->SimplekMeans.**



- **Step 3: Right Click on SimpleKMeans once selected if you need to change the parameters as input like the number of iterations, number of clusters and so on as shown below.**

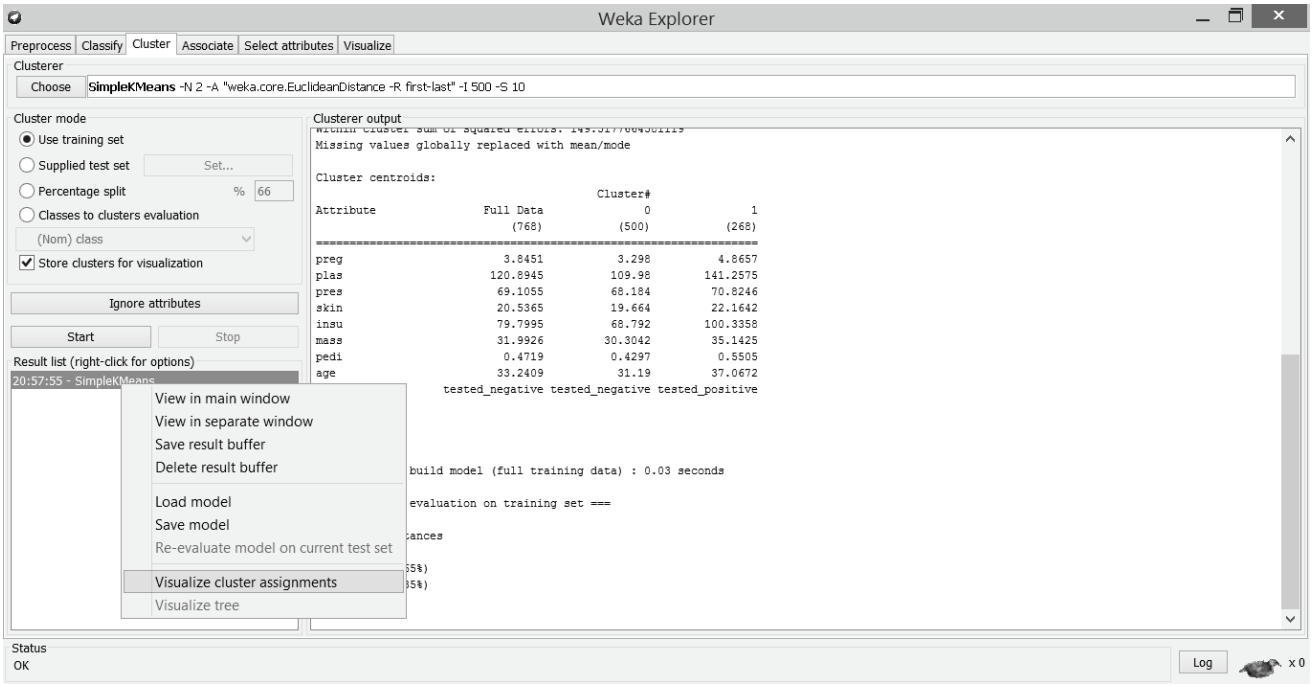


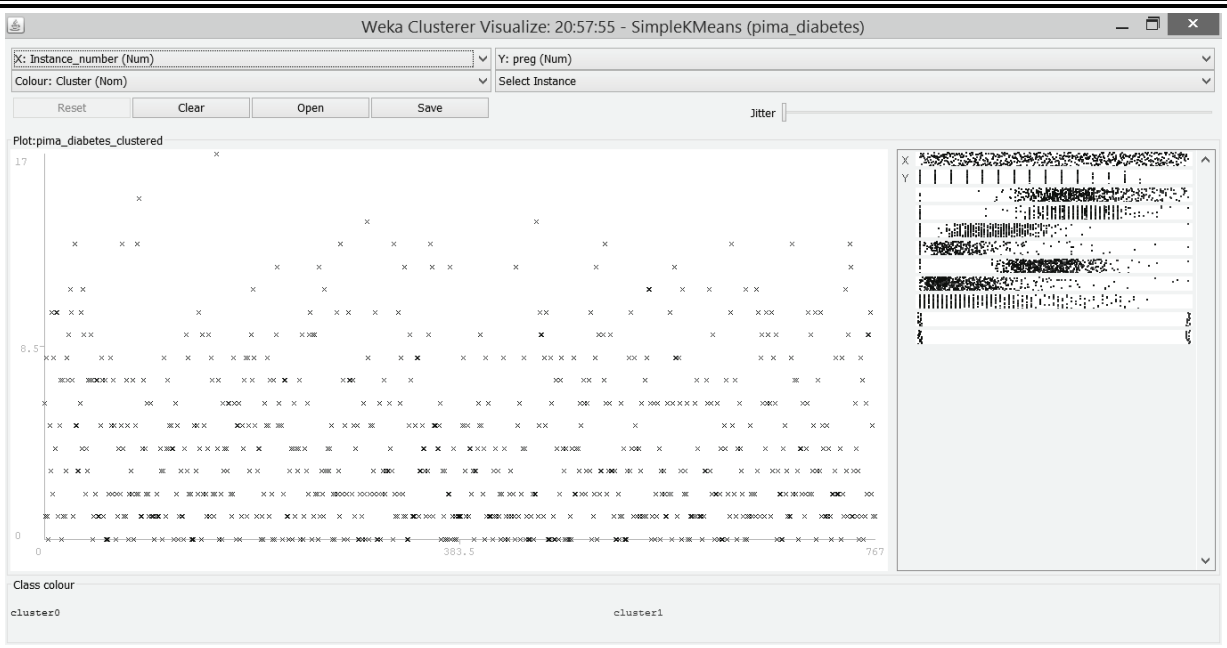
- **Step 4 : Once the parameters are set, click Ok and then click on start, to apply simple k-means on the data set. Following output would appear:**



Since we have set the number of clusters as 2, we can observe that the number of clusters formed are two. There are total 768 instances out of which 500 instances are in cluster 0 and 268 instances in cluster 1.

A graphical output of the cluster can also be viewed by right clicking the Result list and choosing visualizing the cluster assignment as shown below.



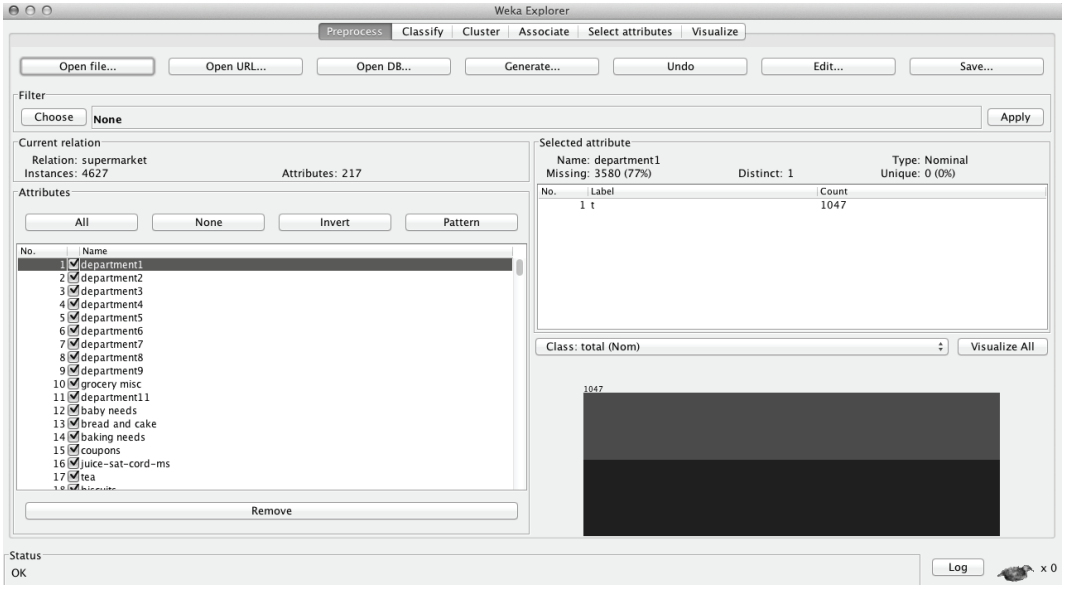


6.4 ASSOCIATION RULE MINING USING WEKA

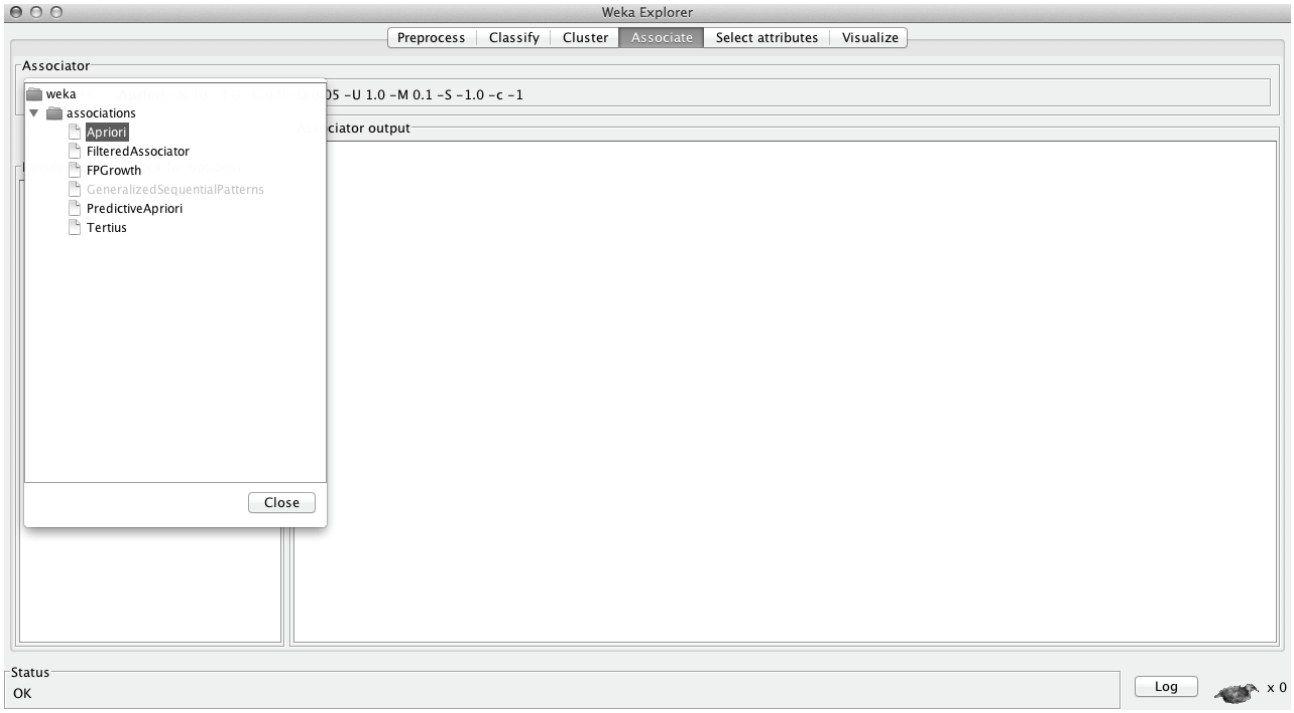
WEKA supports multiple algorithms to find association rules among the attributes. In this section we will observe Apriori and FPGrowth algorithm to generate association rules.

6.4.1 Apriori Algorithm

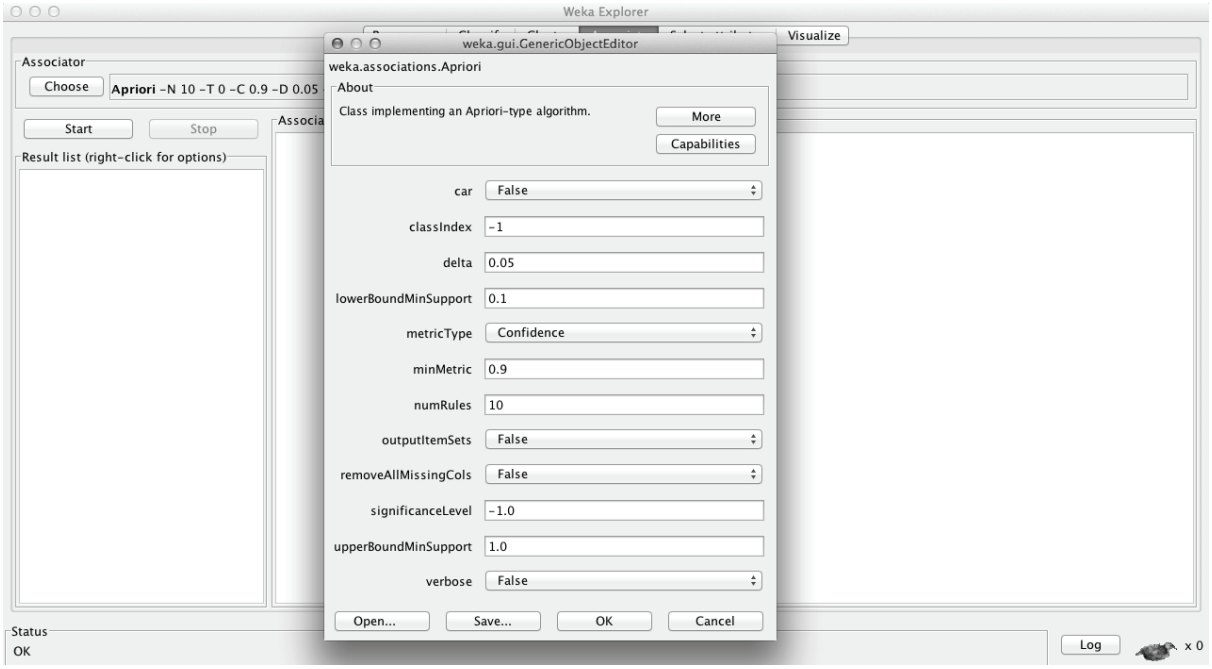
Step 1: Select the data set using the preprocess tab. Here, we are selecting supermarket.arff.



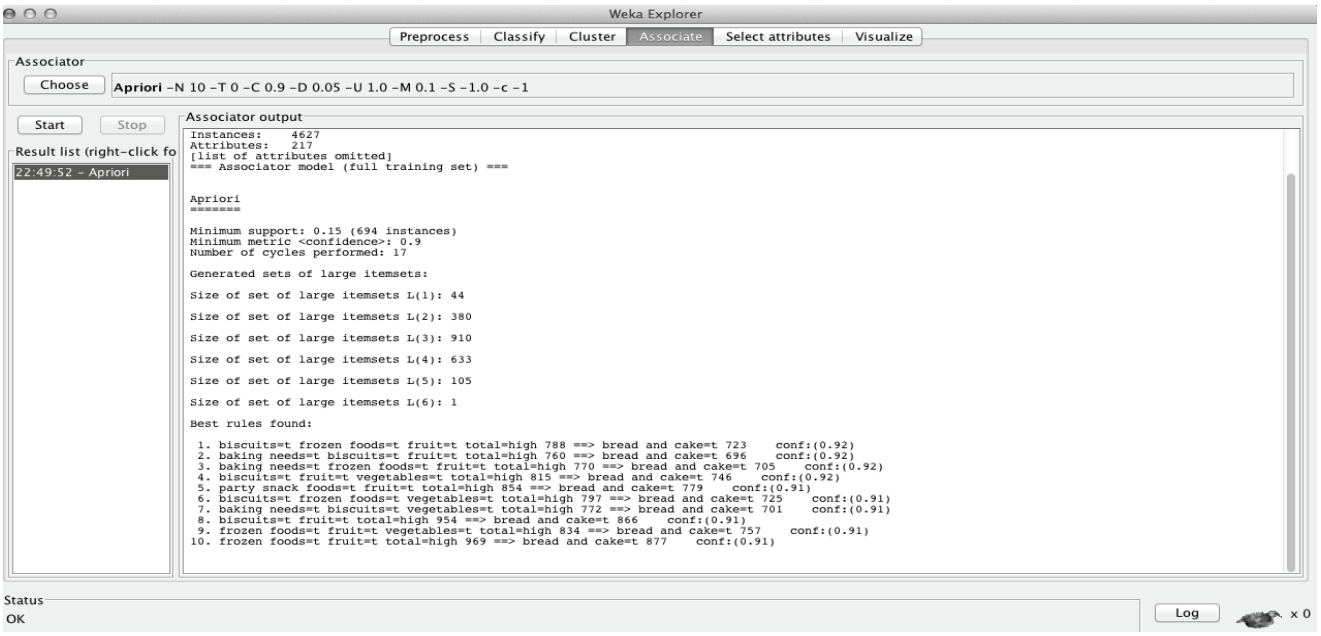
- **Step 2 :Click on Associate tab and select the algorithm for rule generation. Here we are selecting Apriori algorithm.**



- **Step 3: Right click on Apriori and set the required parameters. Click on Ok.**

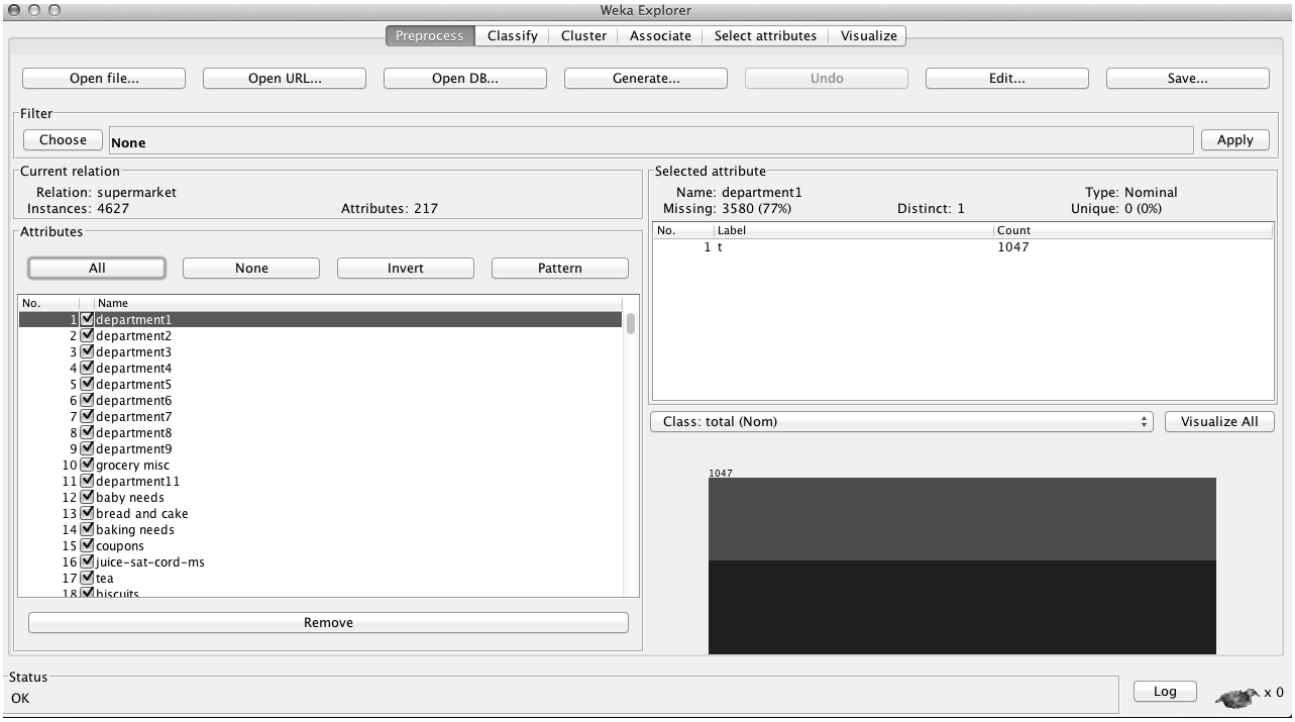


- **Step 4 :Click on start and observe the output. The output displays best rules and their confidence.**

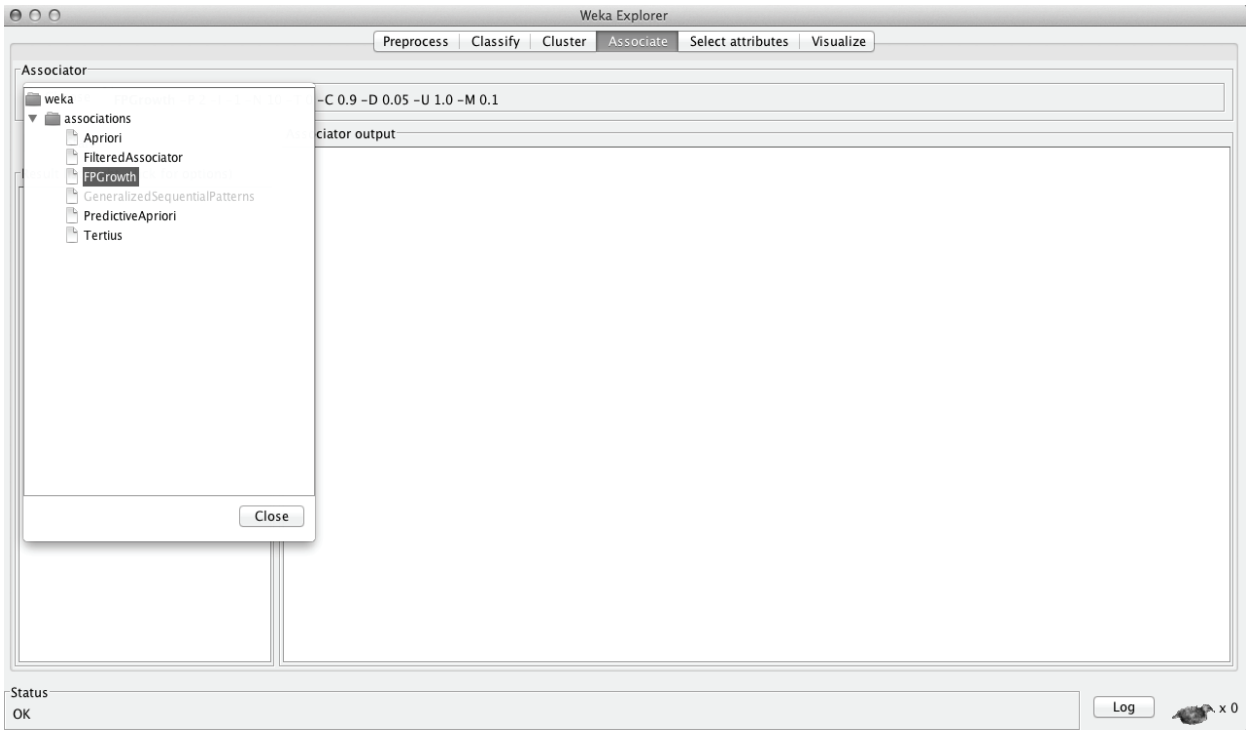


6.4.2 FP Growth Algorithm

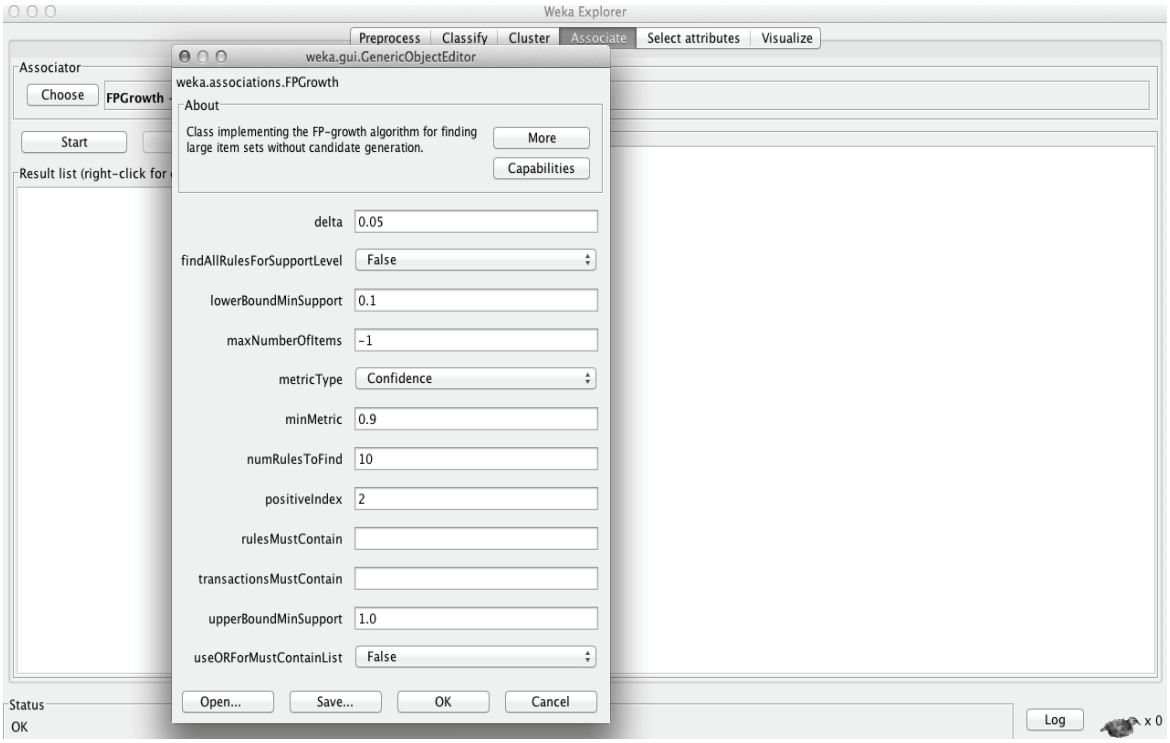
- **Step 1: Select the data set using the preprocess tab. Here, we are selecting supermarket.arff.**



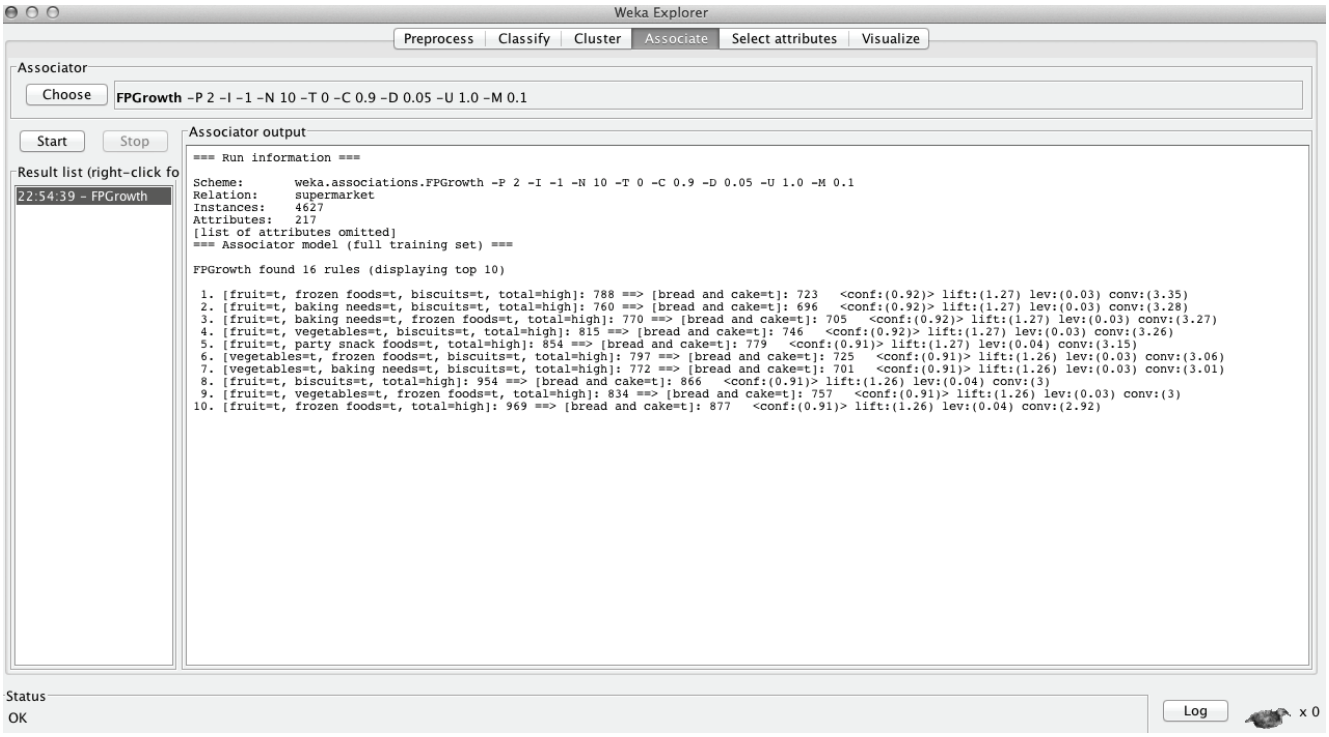
- **Step 2:** Click on Associate tab and select the algorithm for rule generation. Here we are selecting FPGrowth algorithm.



- **Step 3: Right click on FP Growth and set the required parameters. Click on Ok.**



- **Step 4 : Click on start and observe the output. The output displays best rules and their confidence.**



▶ Experiment No. 7 : Implementation of K-means clustering algorithm

Dataset:https://raw.githubusercontent.com/timurista/data-analysis/master/python-jupyter/Cluster%20Analysis/3.01.%20Country%20clusters.csv

```
# importing the relevant libraries
import numpy as np
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
from sklearn.cluster import KMeans
# loading the data
data = pd.read_csv('Countryclusters.csv')
data
#plotting the data
plt.scatter(data['Longitude'],data['Latitude'])
plt.xlim(-180,180)
plt.ylim(-90,90)
plt.show()
# selecting the feature
x = data.iloc[:,1:3] # 1t for rows and second for columns
x
# clustering
kmeans = KMeans(3)
```

```
means.fit(x)
# clustering results
identified_clusters = kmeans.fit_predict(x)
identified_clusters
array([1, 1, 0, 0, 0, 2])
data_with_clusters = data.copy()
data_with_clusters['Clusters'] = identified_clusters
plt.scatter(data_with_clusters['Longitude'],data_with_clusters['Latitude'],c=data_with_clusters['Clusters'],cmap='rainbow')
```

Output :

	Country	Latitude	Longitude	Language
0	USA	44.97	– 103.77	English
1	Canada	62.40	– 96.80	English
2	France	46.75	2.40	French
3	UK	54.01	– 2.53	English
4	Germany	51.15	10.40	German
5	Australia	– 25.45	133.11	English

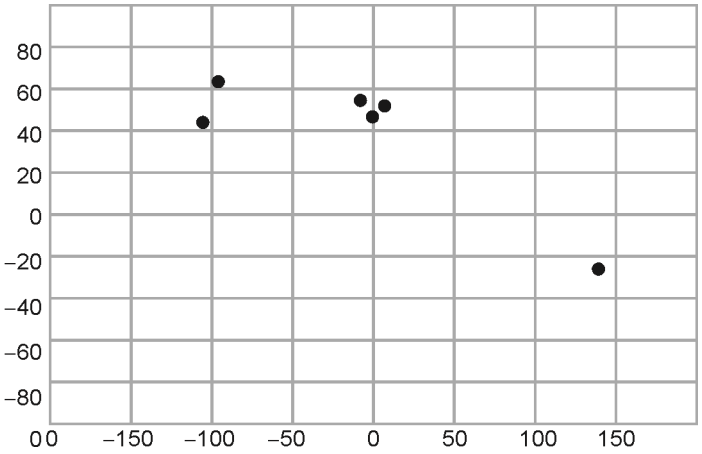


Fig. L.7

	Latitude	Longitude
0	44.97	– 103.77
1	62.40	– 96.80
2	46.75	2.40
3	54.01	– 2.53
4	51.15	10.40
5	– 25.45	133.11

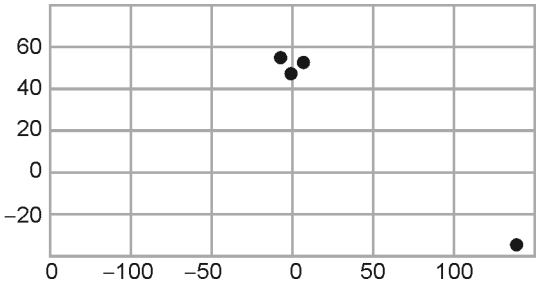


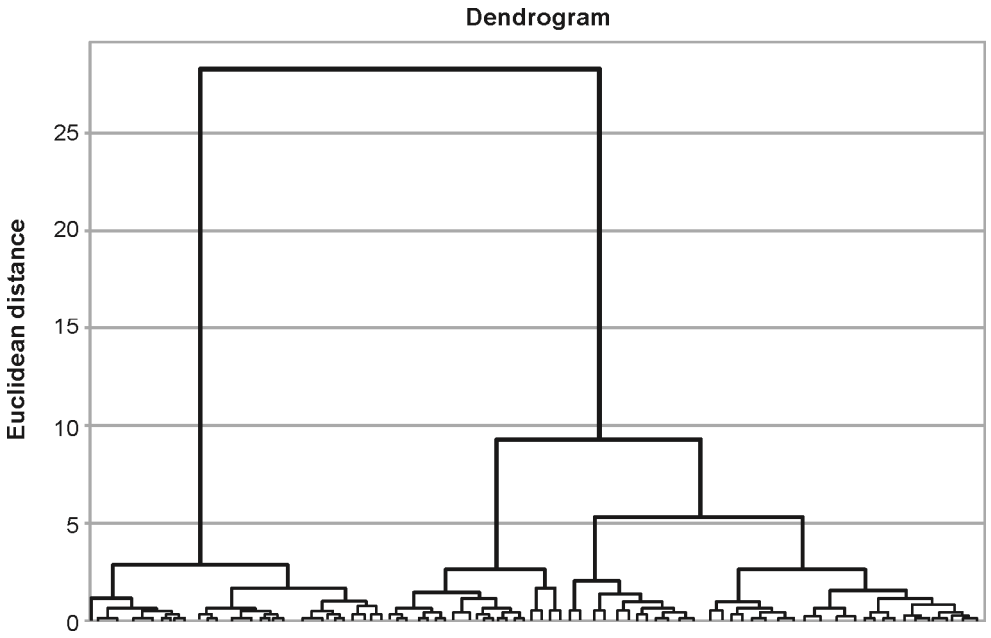
Fig. L.8

▶ Experiment No. 8 : Implementation of Hierarchical clustering algorithm

Dataset: https://raw.githubusercontent.com/jbrownlee/Datasets/master/iris.csv

```
#Importing libraries
from sklearn.datasets import load_iris
from sklearn.cluster import AgglomerativeClustering
import numpy as np
import matplotlib.pyplot as plt
from scipy.cluster.hierarchy import dendrogram , linkage

#Getting the data ready

data = load_iris()
df = data.data
#Selecting certain features based on which clustering is done
df = df[:,1:3]

#Linkage Matrix
Z = linkage(df, method = 'ward')
 # Replace ward by single for single linkage
# Replace ward by complete for complete linkage
#Replace ward by average for average linkage

#plotting dendrogram
dendro = dendrogram(Z)
plt.title('Dendrogram')
plt.ylabel('Euclidean distance')
plt.show()
```

Output :



(1G7)**Fig. L. 9**

▶ Experiment No. 9 : Implementation of Apriori algorithm

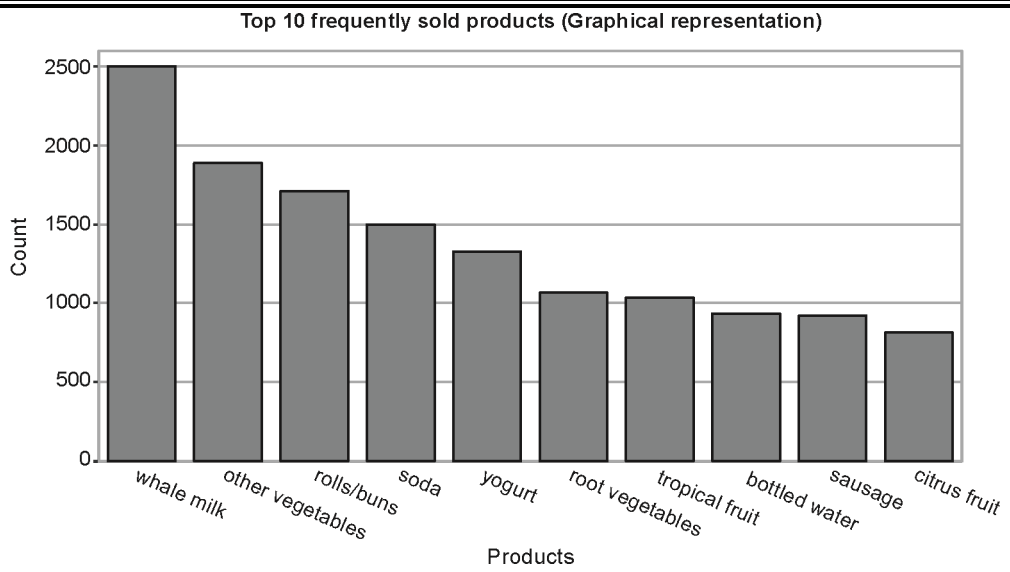
Dataset: https://github.com/amankharwal/Website-data/blob/master/Groceries_dataset.csv

First install **apyori package** using: pip install apyori

```
import numpy as np # linear algebra
import pandas as pd # data processing
import plotly.express as px
import apyori
from apyori import apriori
data = pd.read_csv("Groceries_dataset.csv")
data.head()
```

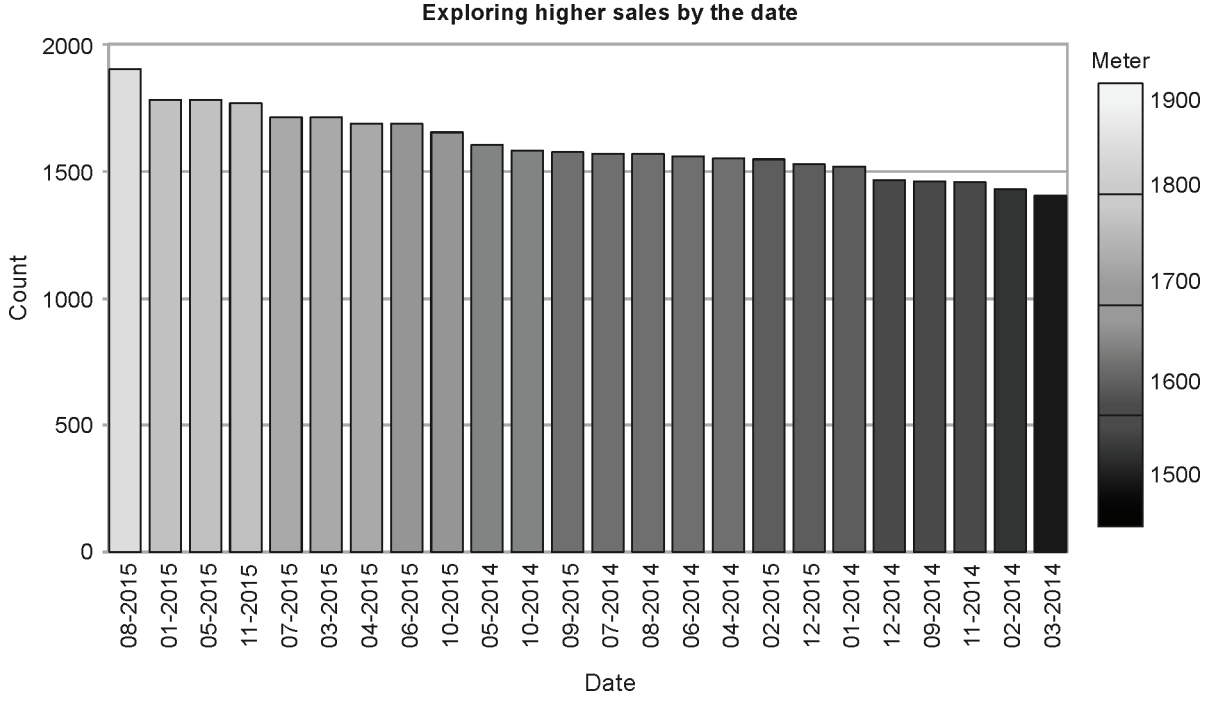
	Member_number	Date	item Description
0	1808	21-07-2015	tropical fruit
1	2552	05-01-2015	whole milk
2	2300	19-09-2015	pip fruit
3	1187	12-12-2015	other vegetables
4	3037	01-02-2015	whole milk

```
print("Top 10 frequently sold products(Tabular Representation)")
x = data['itemDescription'].value_counts().sort_values(ascending=False)[:10]
fig = px.bar(x= x.index, y= x.values)
fig.update_layout(title_text= "Top 10 frequently sold products (Graphical Representation)", xaxis_title= "Products",
yaxis_title="Count")
fig.show()
```



(1G8)**Fig. L. 10**

```
data["Year"] = data['Date'].str.split("-").str[-1]
data["Month-Year"] = data['Date'].str.split("-").str[1] + "-" + data['Date'].str.split("-").str[-1]
fig1 = px.bar(data["Month-Year"].value_counts(ascending=False),
          orientation= "v",
          color = data["Month-Year"].value_counts(ascending=False),
           labels={'value':'Count', 'index':'Date','color':'Meter'})

fig1.update_layout(title_text="Exploring higher sales by the date")
fig1.show()
```



(1G9)**Fig. L. 11**

```
rules = apriori(transactions, min_support = 0.00030, min_confidence = 0.05, min_lift = 3, max_length = 2, target = "rules")
association_results = list(rules)
print(association_results[0])
```

```
RelationRecord(items=frozenset({'liver loaf', 'fruit/vegetable juice'}), support=0.0004009891064626077 5, ordered_statistics=[OrderedStatistic(items_base=frozenset({'liver loaf'}), items_add=frozenset({'fruit/vegetable juice'}), confidence=0.12, lift=3.5276227897838903)])
```

```
for item in association_results:

    pair = item[0]
    items = [x for x in pair]

print("Rule : ", items[0], " -> " + items[1])
print("Support : ", str(item[1]))
print("Confidence : ",str(item[2][0][2]))
print("Lift : ", str(item[2][0][3]))

print("=============================")
```

Output :

```
Rule :  liver loaf  -> fruit/vegetable juice
Support :  0.0004009891064626077 5
Confidence :  0.12
Lift :  3.5276227897838903
=============================
Rule :  ham  -> pickled vegetables
Support :  0.0005346521419501437
Confidence :  0.05970149253731344
Lift :  3.4895055970149254
=============================
Rule :  roll products   -> meat
Support :  0.0003341575887188398
Confidence :  0.06097560975609757
Lift :  3.620547812620984
=============================
Rule :  misc. beverages  -> salt
Support :  0.0003341575887188398
Confidence :  0.05617977528089888
Lift :  3.5619405827461437
=============================
Rule :  spread cheese  -> misc. beverages
Support :  0.0003341575887188398
Confidence :  0.05
Lift :  3.170127118644068
=============================
Rule :  soups  -> seasonal products
```

```
Support :  0.0003341575887188398
Confidence :  0.10416666666666667
Lift :  14.704205974842768
==============================
Rule :  spread cheese  -> sugar
Support :  0.00040098910646260775
Confidence :  0.06
Lift :  3.3878490566037733
==============================
```

▶ Experiment No. 10 : Implementation of Page Rank/HITS algorithm

Page Rank Algorithm in Python

```
Import numpy as np
Import scipy as sc
import pandas as pd
from fractions import Fraction
  def display_format(my_vector,my_decimal):
    return np.round((my_vector).astype(np.float), decimals=my_decimal)
    my_dp=Fraction(1,3)
    Mat=np.matrix([[0,0,1],
    [Fraction(1,2),0,0],
    [Fraction(1,2),1,0]])
    Ex=np.zeros((3,3))
    Ex[:]=my_dp
    beta =0.7
    Al= beta *Mat+((1-beta)*Ex)
    r =np.matrix([my_dp,my_dp,my_dp])
    r =np.transpose(r)
    previous_r= r
  for i in range(1,100):
    r =Al* r
    print(display_format(r,3))
if(previous_r==r).all():
  break
previous_r= r
print("Final:\n",display_format(r,3))
print("sum",np.sum(r))
```

Output :

```
[[0.333]
[0.217]
[0.45 ]]
[[0.415]
[0.217]
[0.368]]
[[0.358]
[0.245]
[0.397]]
[[0.378]
```

```
[0.225]
[0.397]]
[[0.378]
[0.232]
[0.39 ]]
[[0.373]
[0.232]
[0.395]]
[[0.376]
[0.231]
[0.393]]
[[0.375]
[0.232]
[0.393]]
[[0.375]
[0.231]
[0.394]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
```

```
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
```

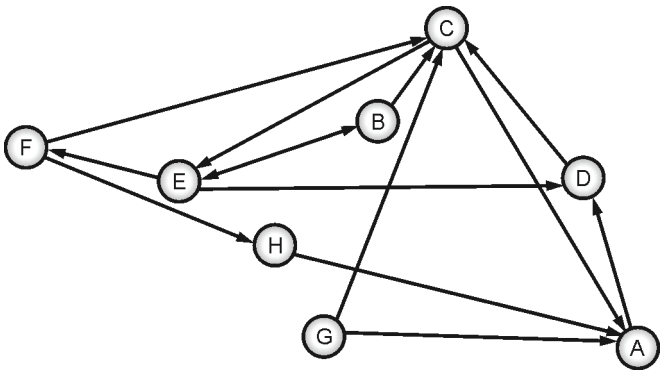
```
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
```

```
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
[[0.375]
[0.231]
[0.393]]
Final:
[[0.375]
[0.231]
[0.393]]
sum 0.9999999999999951
```

HITS algorithm in Python:

```
# importing modules
import networkx as nx
import matplotlib.pyplot as plt

G = nx.DiGraph()

G.add_edges_from([('A', 'D'), ('B', 'C'), ('B', 'E'), ('C', 'A'),
                  ('D', 'C'), ('E', 'D'), ('E', 'B'), ('E', 'F'),
                  ('E', 'C'), ('F', 'C'), ('F', 'H'), ('G', 'A'),
                  ('G', 'C'), ('H', 'A')])

plt.figure(figsize =(10, 10))
nx.draw_networkx(G, with_labels = True)

hubs, authorities = nx.hits(G, max_iter = 50, normalized = True)
# The in-built hits function returns two dictionaries keyed by nodes
# containing hub scores and authority scores respectively.

print("Hub Scores: ", hubs)
print("Authority Scores: ", authorities)
```

Output :



(1G10)**Fig. L. 12**

```
Hub Scores: {'A': 0.04642540386472174, 'D': 0.133660375232863, 'B': 0.15763599440595596, 'C': 0.037389132480584515,
'E': 0.2588144594158868, 'F': 0.15763599440595596, 'H': 0.037389132480584515, 'G': 0.17104950771344754}
Authority Scores: {'A': 0.10864044085687284, 'D': 0.13489685393050574, 'B': 0.11437974045401585, 'C':
0.3883728005172019, 'E': 0.06966521189369385, 'F': 0.11437974045401585, 'H': 0.06966521189369385, 'G': 0.0}
```

Lab Manual Ends…

❑❑❑

ViVa-Questions

Q. 1 What is data warehouse?

Ans. : A data warehouse is an electronic storage of an Organization's historical data for the purpose of reporting, analysis and data mining or knowledge discovery.

Q. 2 What is the benefits of data warehouse?

Ans. : A data warehouse helps to integrate data and store them historically so that we can analyze different aspects of business including, performance analysis, trend, prediction etc. over a given time frame and use the result of our analysis to improve the efficiency of business processes.

Q. 3 What is the difference between OLTP and OLAP?

Ans. : OLTP is the transaction system that collects business data. Whereas OLAP is the reporting and analysis system on that data. OLTP systems are optimized for INSERT, UPDATE operations and therefore highly normalized. On the other hand, OLAP systems are deliberately denormalized for fast data retrieval through SELECT operations.

Q. 4 What is data mart?

Ans. : Data marts are generally designed for a single subject area. An organization may have data pertaining to different departments like Finance, HR, Marketing etc. stored in data warehouse and each department may have separate data marts. These data marts can be built on top of the data warehouse.

Q. 5 What is dimension?

Ans. : A dimension is something that qualifies a quantity (measure).

For an example, consider this: If I just say… "20kg", it does not mean anything. But if I say, "20kg of Rice (Product) is sold to Ramesh (customer) on 5th April (date)", then that gives a meaningful sense. These product, customer and dates are some dimension that qualified the measure - 20kg.

Dimensions are mutually independent. Technically speaking, a dimension is a data element that categorizes each item in a data set into non-overlapping regions.

Q. 6 What is Fact?

Ans. : A fact is something that is quantifiable (or measurable). Facts are typically (but not always) numerical values that can be aggregated.

Q. 7 Briefly state different between data ware house & data mart?

Ans. : Data warehouse is made up of many datamarts. DWH contain many subject areas. but data mart focuses on one subject area generally. e.g. If there will be DHW of bank then there can be one data mart for accounts, one for Loans etc. This is high level definitions. Metadata is data about data. e.g. if in data mart we are receiving any file. then metadata will contain information like how many columns, file is of fix width or variable, ordering of fields, datatypes of field etc...

Q. 8 What are the storage models of OLAP?

Ans. : ROLAP, MOLAP and HOLAP

Q. 9 What are CUBES?

Ans. : A data cube stores data in a summarized version which helps in a faster analysis of data. The data is stored in such a way that it allows reporting easily.

E.g. using a data cube, a user may want to analyze weekly, monthly performance of an employee. Here, month and week could be considered as the dimensions of the cube.

Q. 10 What is MODEL in Data mining world?

Ans. : Models in Data mining help the different algorithms in decision making or pattern matching. The second stage of data mining involves considering various models and choosing the best one based on their predictive performance.

Q. 11 Explain how to mine an OLAP cube.

Ans. : A data mining extension can be used to slice the data the source cube in the order as discovered by data mining. When a cube is mined the case table is a dimension.

Q. 12 Define Rollup and cube.

Ans. : Custom rollup operators provide a simple way of controlling the process of rolling up a member to its parent

values. The rollup uses the contents of the column as custom rollup operator for each member and is used to evaluate the value of the member's parents.

If a cube has multiple custom rollup formulas and custom rollup members, then the formulas are resolved in the order in which the dimensions have been added to the cube.

Q. 13 Differentiate between Data Mining and Data warehousing.

Ans. : Data warehousing is merely extracting data from different sources, cleaning the data and storing it in the warehouse where as data mining aims to examine or explore the data using queries. These queries can be fired on the data warehouse. Explore the data in data mining helps in reporting, planning strategies, finding meaningful patterns etc.

E.g. a data warehouse of a company stores all the relevant information of projects and employees. Using Data mining, one can use this data to generate different reports like profits generated etc.

Q. 14 What is a Decision Tree Algorithm?

Ans. : A decision tree is a tree in which every node is either a leaf node or a decision node. This tree takes an input an object and outputs some decision. All Paths from root node to the leaf node are reached by either using AND or ORor BOTH. The tree is constructed using the regularities of the data. The decision tree is not affected by Automatic Data Preparation.

Q. 15 What is Naïve Bayes Algorithm?

Ans. : Naïve Bayes Algorithm is used to generate mining models. These models help to identify relationships between input columns and the predictable columns. This algorithm can be used in the initial stage of exploration. The algorithm calculates the probability of every state of each input column given predictable columns possible states. After the model is made, the results can be used for exploration and making predictions.

Q. 16 Explain clustering algorithm.

Ans. : Clustering algorithm is used to group sets of data with similar characteristics also called as clusters. These clusters help in making faster decisions, and exploring data. The algorithm first identifies relationships in a dataset following which it generates a series of clusters based on the relationships. The process of creating clusters is iterative. The algorithm redefines the groupings to create clusters that better represent the data.

Q. 17 Explain Association algorithm in Data mining?

Ans. :

– Association algorithm is used for recommendation engine that is based on a market based analysis. This engine suggests products to customers based on what they bought earlier. The model is built on a dataset containing identifiers. These identifiers are both for individual cases and for the items that cases contain.

– These groups of items in a data set are called as an item set. The algorithm traverses a data set to find items that appear in a case. MINIMUM_SUPPORT parameter is used any associated items that appear into an item set.

Q. 18 What are the goals of data mining?

Ans. : Prediction, identification, classification and optimization

Q. 19 Is data mining independent subject?

Ans. : No, it is interdisciplinary subject. includes, database technology, visualization, machine learning, pattern recognition, algorithm etc.

Q. 20 What are data mining functionality?

Ans. : Mining frequent pattern, association rules, classification and prediction, clustering, evolution analysis and outlier Analysis.

Q. 21 What are issues in data mining ?

Ans. : Issues in mining methodology, performance issues, user interactive issues, different source of data types issues etc.

Q. 22 List some applications of data mining.

Ans. : Agriculture, biological data analysis, call record analysis, DSS, Business intelligence system etc

Q. 23 What do you mean by interesting pattern?

Ans. : A pattern is said to be interesting if it is 1. easily understood by human 2. valid 3. potentially useful 4. novel

Q. 24 Why do we pre-process the data ?

Ans. : To ensure the data quality. [accuracy, completeness, consistency, timeliness, believability, interpret-ability]

Q. 25 What are the steps involved in data pre-processing?

Ans. : Data cleaning, data integration, data reduction, data transformation.

Q. 26 What are the forms of multidimensional model?

Ans. : Star schema, Snow flake schema, Fact constellation Schema

Q. 27 What are frequent pattern?

Ans. : A set of items that appear frequently together in a transaction data set.E.g. milk, bread, sugar

Q. 28 Compare K-mean and K-mediods algorithm.

Ans. : K-mediods is more robust than k-mean in presence of noise and outliers. K-Mediods can be computationally costly.

Q. 29 What is Baye's Theorem?

Ans. : $P(H/X) = P(X/H)* P(H)/P(X)$

Q. 30 What is decision tree classifier?

Ans. : A decision tree is a hierarchically based classifier which compares data with a range of properly selected features.

Q. 31 If there are n dimensions, how many cuboids are there?

Ans. : There would be 2^n cuboids.

Q. 32 What do you mean by web content mining?

Ans. : Web content mining refers to the discovery of useful information from Web contents, including text, images, audio, video, etc.

Q. 33 Define web structure mining and web usage mining.

Ans. :

- Web structure mining studies the model underlying the link structures of the Web. It has been used for search engine result ranking and other Web applications.
- Web usage mining focuses on using data mining techniques to analyze search logs to find interesting patterns. One of the main applications of Web usage mining is its use to learn user profiles.

Q. 34 What are frequent patterns?

Ans. : These are the patterns that appear frequently in a data set, item-set, sub sequence, etc.

Q. 35 What can business analysts gain from having a data warehouse?

Ans. :

- First, having a data warehouse may provide a competitive advantage by presenting relevant information from which to measure performance and make critical adjustments in order to help win over competitors.
- Second, a data warehouse can enhance business productivity because it is able to quickly and efficiently gather information that accurately describes the organization.
- Third, a data warehouse facilitates customer relationship management because it provides a consistent view of customers and item across all lines of business, all departments and all markets.
- Finally, a data warehouse may bring about cost reduction by tracking trends, patterns, and exceptions over long periods in a consistent and reliable manner.

Q. 36 Why is association rule necessary?

Ans. :

- In data mining, association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases.
- It is intended to identify strong rules discovered in database using different measures of interesting.

Q. 37 What are two types of data mining tasks?

Ans. :

- Descriptive task
- Predictive task

Q. 38 Define classification.

Ans. : Classification is the process of finding a model (or function) that describes and distinguishes data classes or concepts.

Q. 39 What are outliers?

Ans. : A database may contain data objects that do not comply with the general behavior or model of the data. These data objects are called outliers.

Q. 40 Define KDD.

Ans. : The process of finding useful information and patterns in data.

Q. 41 What are the components of data mining?

Ans. :

- Database, Data Warehouse, World Wide Web, or other information repository

- Database or Data Warehouse Server
- Knowledge Based
- Data Mining Engine
- Pattern Evaluation Module
- User Interface

Q. 42 Define metadata.

Ans. : A database that describes various aspects of data in the warehouse is called metadata.

Q. 43 What are the usage of metadata?

Ans. :

- Map source system data to data warehouse tables
- Generate data extract, transform, and load procedures for import jobs
- Help users discover what data are in the data warehouse
- Help users structure queries to access data they need

Q. 44 Define HOLAP.

Ans. : The hybrid OLAP approach combines ROLAP and MOLAP technology.

Q. 45 What are data mining techniques?

Ans. :

- Association rules
- Classification and prediction
- Clustering
- Deviation detection
- Similarity search
- Sequence Mining

Q. 46 List the typical OLAP operations.

Ans. : ROLL UP, DRILL DOWN, ROTATE, SLICE AND DICE

Q. 47 If there are 3 dimensions, how many cuboids are there in cube?

Ans. : 2^3 = 8 cuboids

Q. 48 Differentiate between star schema and snowflake schema.

Ans. :

- Star Schema is a multi-dimension model where each of its disjoint dimension is represented in single table.
- Snow-flake is normalized multi-dimension schema when each of disjoint dimension is represent in multiple tables.
- Star schema can become a snow-flake
- Both star and snowflake schemas are dimensional models; the difference is in their physical implementations.
- Snowflake schemas support ease of dimension maintenance because they are more normalized.
- Star schemas are easier for direct user access and often support simpler and more efficient queries.
- It may be better to create a star version of the snowflaked dimension for presentation to the users

Q. 49 List the advantages of star schema.

Ans. :

- Star Schema is very easy to understand, even for non-technical business manager.
- Star Schema provides better performance and smaller query times
- Star Schema is easily extensible and will handle future changes easily

Q. 50 What are the characteristics of data warehouse?

Ans. : Integrated, Non-volatile, Subject oriented, Time variant

Q. 51 Define support and confidence.

Ans. :

- The support for a rule R is the ratio of the number of occurrences of R, given all occurrences of all rules.
- The confidence of a rule $X \rightarrow Y$, is the ratio of the number of occurrences of Y given X, among all other occurrences given X

Q. 52 What is ETL?

Ans. :

- ETL stands for extraction, transformation, and loading process. ETL is software that allows the business to develop their disparate records while moving it from place to place, and it doesn't really matter that data is in several forms or formats. The data can come from any source. ETL is powerful enough to manage such data disparities.

- First, the extract function reads data from a particular source database and extracts a desired subset of data.
- Second, the transform function works with the acquired record using rules or lookup tables, or creating a combination with other records to convert it to the desired state.
- Finally, the load function is used to write the resulting information to a target database.

Q. 53 What are conformed dimensions?

Ans. : Conformed dimension defines the exact same thing with every possible fact table to which they are joined. They are simple to the cubes.

Q. 54 What is a surrogate key?

Ans. : A surrogate key is a substitution for the essential primary key. It is just a unique identifier or statistic for each row that can be used for the primary key to the table. The only requirement for a surrogate primary key is that it is unique for each row in the table. It is useful because the essential primary key can change, and this makes updates more difficult. Surrogate keys are always integer or numeric.

Q. 55 What is a junk dimension?

Ans. : A number of very small dimension may be lumped together to form a single dimension; a junk dimension is the attributes are not closely related. Grouping of random flags and text attributes in dimensions and changing them to a separate sub-dimension is called the junk dimension.

Q. 56 How many fact tables are there in a star schema?

Ans. : There is only one fact table in a star Schema.

Q. 57 What is Normalization?

Ans. : Normalization splits up the data into additional tables.

Q. 58 Out of star schema and snowflake schema, whose dimension table is normalized?

Ans. : Snowflake schema uses the concept of normalization.

Q. 59 What is the benefit of normalization?

Ans. : Normalization helps in reducing data redundancy.

Q. 60 Which language is used for defining Schema Definition?

Ans. : Data Mining Query Language (DMQL) is used for Schema Definition.

Q. 61 What language is the base of DMQL?

Ans. : DMQL is based on Structured Query Language (SQL).

Q. 62 What are the reasons for partitioning?

Ans. : Partitioning is done for various reasons such as easy management, to assist backup recovery, to enhance performance.

Q. 63 What is factless fact tables?

Ans. : A factless fact tables are the fact table which doesn't contain numeric fact column in the fact table.

Q. 64 How can we load the time dimension?

Ans. : Time dimensions are usually loaded through all possible dates in a year and it can be done through a program. Here, 100 years can be represented with one row per day.

Q. 65 What is SCD?

Ans. : SCD is defined as slowly changing dimensions, and it applies to the cases where record changes over time.

Q. 66 What are the types of SCD?

Ans. : There are three types of SCD and they are as follows :

- SCD 1 – The new record replaces the original record
- SCD 2 – A new record is added to the existing customer dimension table
- SCD 3 – A original data is modified to include new data

Q. 67 What is the difference between ER Modeling and Dimensional Modeling?

Ans. :

- ER modeling will have logical and physical model but Dimensional modeling will have only Physical model.
- ER Modeling is used for normalizing the OLTP database design whereas Dimensional Modeling is used for de-normalizing the ROLAP and MOLAP design.

Q. 68 What are the steps to build the datawarehouse?

Ans. : Following are the steps to be followed to build the datawaerhouse:

(1) Gathering business requirements

(2) Identifying the necessary sources

(3) Identifying the facts

(4) Defining the dimensions

(5) Defining the attributes

(6) Redefine the dimensions and attributes if required

(7) Organize the Attribute hierarchy

(8) Define Relationships

(9) Assign unique Identifiers

Q. 69 What are the different types of datawarehosuing?

Ans. : Following are the different types of Datawarehousing:

(1) Enterprise Datawarehousing

(2) Operational Data Store

(3) Data Mart

Q. 70 What is the difference between metadata and data dictionary?

Ans. : Metadata is defined as data about the data. But, Data dictionary contain the information about the project information, graphs, commands and server information.

Q. 71 Define Pre Pruning?

Ans. : A tree is pruned by halting its construction early. Upon halting, the node becomes a leaf. The leaf may hold the most frequent class among the subset samples.

Q. 72 What Are Interval Scaled Variables?

Ans. : Interval scaled variables are continuous measurements of linear scale. For example, height and weight, weather temperature or coordinates for any cluster. These measurements can be calculated using Euclidean distance or Minkowski distance.

Q. 73 What is Smoothing?

Ans. : Smoothing is an approach that is used to remove the non-systematic behavior found in time series. It usually takes the form of finding moving averages of attribute values. It is used to filter out noise and outliers.

Q. 74 What Are the advantages Data Mining over Traditional Approaches?

Ans. : Data Mining is used for the estimation of future. For example, if we take a company/business organization by using the concept of Data Mining we can predict the future of business interms of Revenue (or) Employees (or) Customers (or) Orders etc.

Traditional approaches use simple algorithms for estimating the future. But it does not give accurate results when compared to Data Mining.

Q. 75 Define Binary Variables? And What are the two types of Binary Variables?

Ans. : Binary variables are understood by two states 0 and 1, when state is 0, variable is absent and when state is 1, variable is present. There are two types of binary variables, symmetric and asymmetric binary variables. Symmetric variables are those variables that have same state values and weights. Asymmetric variables are those variables that have not same state values and weights.

Q. 76 What Are Non-Additive Facts?

Ans. : Non-additive facts are facts that cannot be summed up for any of the dimensions present in the fact table.

Q. 77 What Is Attribute Selection Measure?

Ans. : The information Gain measure is used to select the test attribute at each node in the decision tree. Such a measure is referred to as an attribute selection measure or a measure of the goodness of split.

Q. 78 What is the Scope of Data Mining?

Ans. : It helps automate the process of analyzing and identifying predictive information in a huge amount of databases and datasets. Data Mining tools can help scrape and sweep through a diverse range of data in order to identify a pattern that was previously hidden.

Q. 79 In Data Mining, what are "Continuous" and "Discrete" data?

Ans. : "Continuous data" is the data that changes continuously in a well-structured manner. The perfect example of this is age. "Discrete data" is when data is finite and has a specific meaning present in it. The most suitable example of this is gender.

Q. 80 What are a few data mining basic issues?

Ans. : A few issues of data mining are:

(1) Uncertainty handling

(2) Dealing with noisy data

(3) Dealing with missing values

(4) Data selection

Q. 81 Explain ID3 Algorithm?

Ans. : Generally, the ID3 calculation starts with the original set as the root hub. Also, on every cycle, it emphasizes through every unused attribute of the set and figures. Moreover, the entropy of attribute. Furthermore, at that point chooses the attribute. Also, it has the smallest entropy value.

Q. 82 What is a subject-oriented data warehouse?

Ans. : Subject-oriented data warehouses are those that store data around a particular "subject" such as customer, sales, product, among others.

Q. 83 What is data aggregation?

Ans. : Data aggregation is the broad definition for any process that enables information gathering expression in a summary form, for statistical analysis.

Q. 84 What is summary information?

Ans. : Summary Information is the location within data warehouse where predefined aggregations are stored.

Q. 85 What Does PageRank Mean?

Ans. : PageRank is an algorithm used by the Google search engine to measure the authority of a webpage. While the details of PageRank are proprietary, it is generally believed that the number and importance of inbound links to that page are a significant factor.

❑❑❑

Note

REFERENCES

1. Paulraj Ponniah, “Data Warehousing: Fundamentals for IT Professionals”, Wiley India.
2. Han, Kamber, “Data Mining Concepts and Techniques”, Morgan Kaufmann 2nd Edition.
3. M.H. Dunham, “Data Mining Introductory and Advanced Topics”, Pearson Education.
4. Reema Thareja, “Data warehousing”, Oxford University Press 2009.
5. Pang-Ning Tan, Michael Steinbach and Vipin Kumar, “Introduction to Data Mining”, Pearson Publisher 2nd Edition.
6. Ian H. Witten, Eibe Frank and Mark A. Hall, “Data Mining”, Morgan Kaufmann 3rd Edition.
7. Ramkrishnan, Gehrke, “Database Management Systems”,TMH, 3rd Edition.
8. Michael Berry and Gordon Linoff “Data Mining Techniques”, 2nd Edition Wiley Publications.
9. Michael Berry and Gordon Linoff “Mastering Data Mining- Art & science of CRM”, Wiley Student Edition.
10. Vikram Pudi & Radha Krishna, “Data Mining”, Oxford Higher Education.
11. Pratik Bhatia, “Data Mining and Data Warehousing: Principles and Practical Techniques”, Cambridge
12. Mehmed Kantardzic, “Data Mining: Concepts, Models, Methods, and Algorithms”, 3rd Edition, Wiley IEEE Press.
13. Ralph Kimball, “The Data Warehouse Toolkit”, Wiley.
14. Eibe Frank, “Data Mining: Practical Machine Learning Tools and Techniques”,
15. https://onlinecourses.nptel.ac.in/noc20_cs12/preview
16. https://onlinecourses.nptel.ac.in/noc21_cs06/preview
17. https://www.coursera.org/specializations/data-mining
18. https://www.examveda.com/database/practice-mcq-question-on-data-warehousing/
19. https://www.javatpoint.com/data-mining-mcq
20. https://t4tutorials.com/data-mining-mcqs/
21. https://www.sanfoundry.com/
22. https://erdplus.com/
23. https://explainextended.com/2010/09/30/olap-in-mysql-four-ways-to-filter-on-higher-level-dimensions/
24. http://web.cs.wpi.edu/~cs561/s14/Lectures/W4/OLAP.pdf
25. https://www.cs.waikato.ac.nz/ml/weka/

❑❑❑

Note