Department of Computer Engineering

Vivekanand Education Society's Institute of Technology

Hashu Advani Memorial Complex, Collector's Colony,

Chembur, Mumbai – 400074. India.

(A.Y 2023-24)

Mini Project Report On
"Healthcare Management System"

Course Title: Data Warehousing and Mining

Course Code: CSL 503

Mini Project Report for

Continuous Assessment

Lab Instructor: Mrs. Priya R. L

Submitted by,

Name: Ayush Duseja

Division: D12-A

Roll no: 22

Index

MINI PROJECT REPORT

Aim:

Mini Project and Presentation.

Introduction to Project:

In today's fast-evolving healthcare landscape, efficient data management is critical for enhancing the quality of care, optimizing operations, and improving financial outcomes. A healthcare management system serves as the backbone of these processes by organizing data related to patients, treatments, and other vital operational details. However, there is immense potential to further optimize these systems by transforming traditional databases into data warehouses. Data warehouses can store and organize large amounts of structured data from multiple sources, making it easier to mine for insights that drive better decision-making.

The proposed healthcare management system aims to do just that by evolving from a simple database to a comprehensive data warehouse. This system will collect and store essential information, including patient demographics, treatments, insurance information, and billing status. Once this data is accumulated, it can be analyzed to extract meaningful insights, such as predicting patient outcomes, improving hospital operations, and identifying financial inefficiencies.

The goal is not only to ensure efficient data storage but also to enable stakeholders—such as healthcare providers, administrators, and financial managers—to make informed decisions based on advanced analytics. These insights can help optimize resource allocation, enhance patient care, and improve the financial performance of healthcare institutions. With regular updates and trend analyses, the system will provide valuable information for continuous improvement in healthcare services.

Problem Definition:

Current healthcare management systems often function solely as databases, limiting their capacity to generate actionable insights for improving patient care, operational efficiency, and financial performance. Without a robust data warehouse, valuable information remains fragmented, making it difficult for stakeholders to identify patterns, predict outcomes, and optimize processes effectively. This lack of advanced data management hinders the healthcare sector's ability to adapt quickly to changing needs and maximize resource utilization.

By transforming the existing database into a fully functional data warehouse, this project seeks to resolve these challenges. Through data mining and predictive analysis, the system will provide insights into clinical, operational, and financial aspects of healthcare management. Stakeholders can then use these insights to make strategic decisions that enhance patient care, streamline hospital operations, and control costs.

Key Points

1. **Comprehensive Data Warehouse:** Transforming the healthcare management system from a database into a data warehouse to store, organize, and mine patient and operational data.
2. **Clinical Analysis:** Predicting and identifying effective treatments for specific patient demographics and conditions to improve healthcare outcomes.
3. **Operational Analysis:** Analyzing hospital operations to identify bottlenecks, improve patient flow, and optimize resource usage.
4. **Financial Analysis:** Reducing financial inefficiencies by optimizing billing processes, detecting high-cost treatments, and improving revenue cycles.
5. **Data-Driven Decision Making:** Providing stakeholders with actionable insights through periodic analysis to enhance patient care, operational efficiency, and financial management.

Data warehousing and Mining Tools Used :

● Data Warehouse Tools :

1. Microsoft SQL Server
2. Flat File DataSet (EX : .csv)

● Data Mining Tools :

1. Tableau
2. Microsoft SQL Server Analysis Services (SSAS)
3. Python ,Java

Dataset Used:

Dataset :

https://www.kaggle.com/datasets/sanaaafrine/covid-19-dataset

Dataset(Used for 1-4 Exps):

ID	Age	AgeGroup	Gender	BloodPres	Hemoglob	BloodGluc	SugarLevel	Arthritis	Breathing	Diagnosis	TypeofDis	Medicine	TypeofMe	Locality	RecoveryTime
1	62	Adults	Female	70	12.6	122	Prediabeti	No	No	Dengue	Epidemic [	Inj C-Tri 1	Antibiotic	Rural	0 to 48 Hours
2	43	Adults	Male	60	17.8	174	High	No	No	Chikungun	Epidemic [	Inj C-Tri 1	Antibiotic	Rural	0 to 48 Hours
3	62	Adults	Female	70	12.6	122	Prediabeti	No	No	Dengue	Epidemic [	Inj C-Tri 1	Antibiotic	Rural	0 to 48 Hours
4	43	Adults	Male	60	17.8	165	High	No	No	Chikungun	Epidemic [	Inj C-Tri 1	Antibiotic	Rural	0 to 48 Hours
5	56	Adults	Female	60	13.7	97	Normal	No	No	Viral Fever	Epidemic [	Inj C-Tri 1	Antibiotic	Rural	48 to 72 Hours
6	4	Children	Female	80	12	99	Normal	No	No	Bacterial I	Communic	Inj Clavan	Antibiotic	Rural	48 to 72 Hours
7	4	Children	Female	80	12	99	Normal	Yes	No	Bacterial I	Communic	Inj Clavan	Antibiotic	Rural	48 to 72 Hours
8	29	Adults	Female	80	17.5	132	High	No	No	Rickettsial	Bacterial I	Inj C-Tri 1	Antibiotic	Rural	0 to 48 Hours
9	34	Adults	Male	70	15.3	98	Normal	No	No	Viral Fever	Epidemic [	Inj C-Tri 1	Antibiotic	Rural	48 to 72 Hours
10	29	Adults	Female	80	17.5	132	High	No	No	Rickettsial	Bacterial I	Inj C-Tri 1	Antibiotic	Rural	0 to 48 Hours
11	80	Seniors	Male	70	12.65	124	Prediabeti	No	No	Rickettsial	Bacterial I	Inj C-Tri 1	Antibiotic	Rural	48 to 72 Hours
12	59	Adults	Male	50	9.6	100	Prediabeti	No	No	Rickettsial	Bacterial I	Inj C-Tri 1	Antibiotic	Rural	48 to 72 Hours
13	6	Children	Female	50	12.65	140	High	Yes	No	Bacterial I	Communic	Inj Mikacir	Antibiotic	Urban	48 to 72 Hours
14	6	Children	Female	50	12.65	137	High	No	No	Bacterial I	Communic	Inj Mikacir	Antibiotic	Urban	48 to 72 Hours
15	50	Adults	Female	80	14.2	121	Prediabeti	No	No	Constipati	Epidemic [	Inj-Emsut I	Antiinflam	Rural	48 to 72 Hours
16	33	Adults	Female	90	11.8	186	High	No	No	Gastroenti	Epidemic [	Inj-Sompra	Proton Pur	Rural	48 to 72 Hours
17	47	Adults	Male	60	13.6	121	Prediabeti	No	No	Renalcalcı	Epidemic [	cap cerab,	Antispasm	Rural	48 to 72 Hours
18	76	Seniors	Female	70	10.3	99	Normal	No	No	Viral Fever	Epidemic [	Inj C-Tri 1	Antibiotic	Rural	0 to 48 Hours
19	76	Seniors	Female	70	10.3	99	Normal	No	No	Viral Fever	Epidemic [	Inj C-Tri 1	Antibiotic	Rural	0 to 48 Hours
20	41	Adults	Female	70	11.1	100	Prediabeti	No	No	Dengue	Epidemic [	Inj C-Tri 1	Antibiotic	Rural	48 to 72 Hours
21	47	Adults	Male	80	14.9	99	Normal	No	No	Rickettsial	Bacterial I	Inj C-Tri 1	Antibiotic	Rural	0 to 48 Hours
22	47	Adults	Male	80	14.9	99	Normal	No	No	Rickettsial	Bacterial I	Inj C-Tri 1	Antibiotic	Rural	0 to 48 Hours
23	60	Adults	Male	80	12.65	198	High	No	No	Rickettsial	Bacterial I	Inj C-Tri 1	Antibiotic	Rural	48 to 72 Hours
24	68	Seniors	Male	70	12.4	114	Prediabeti	No	No	Rickettsial	Bacterial I	Inj C-Tri 1	Antibiotic	Rural	48 to 72 Hours
25	23	Youth	Female	80	17.7	74	Normal	No	No	Viral Fever	Epidemic [	Inj C-Tri 1	Antibiotic	Rural	48 to 72 Hours

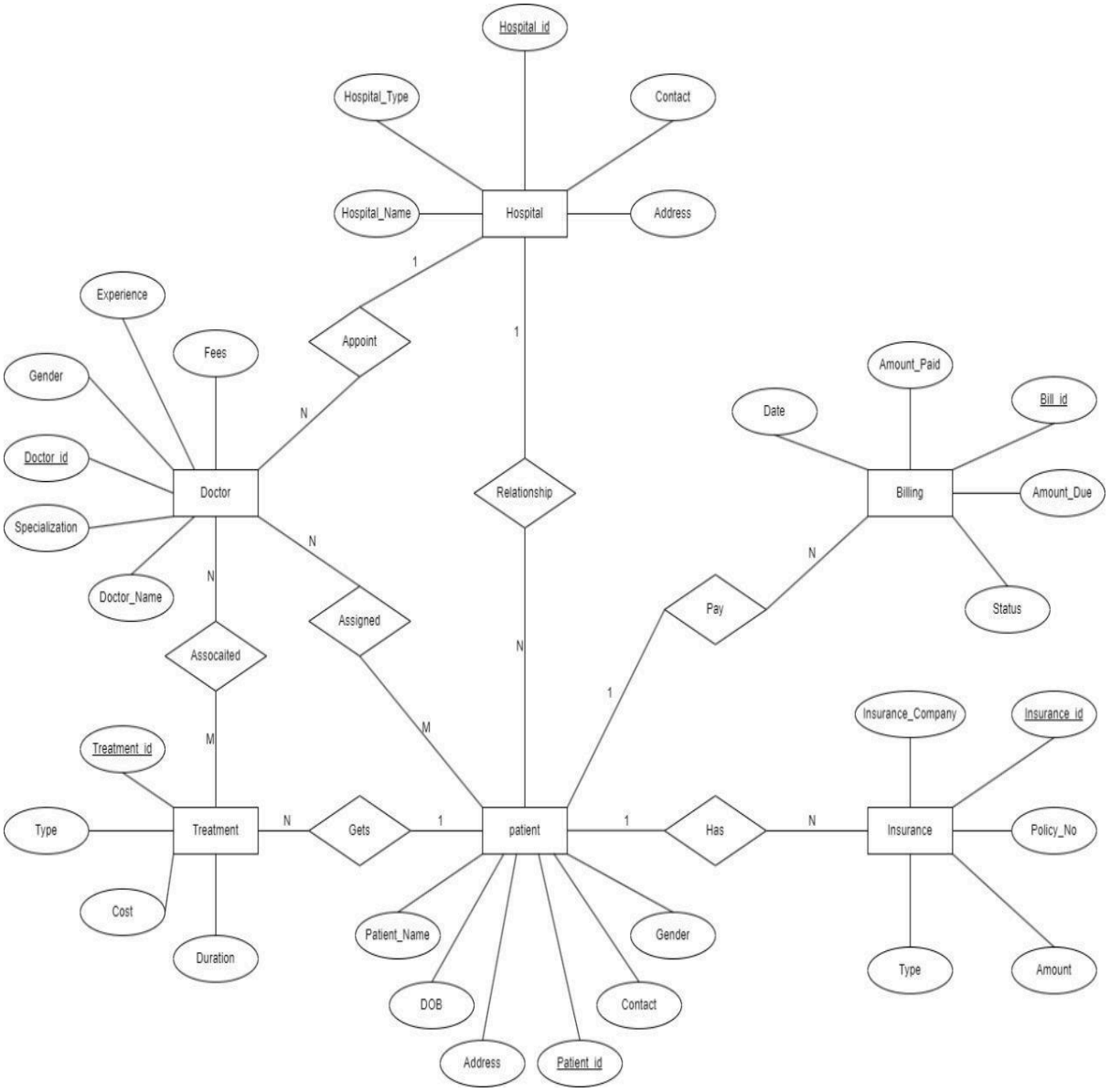
Dataset(Used for remaining exps):

country	continent	population	day	time	Cases	Recovered	Deaths	Tests
Saint-Hele	Africa	6115	########	2024-06-3	2166	2		
Falkland-Is	South-Ame	3539	########	2024-06-3	1930	1930		8632
Montserra	North-Ame	4965	########	2024-06-3	1403	1376	8	17762
Diamond-Princess			########	2024-06-3	712	699	13	
Vatican-Ci	Europe	799	########	2024-06-3	29	29		
Western-S	Africa	626161	########	2024-06-3	10	9	1	
MS-Zaandam			########	2024-06-3	9	7	2	
China	Asia	1.45E+09	########	2024-06-3	503302	379053	5272	1.6E+08
Tokelau	Oceania	1378	########	2024-06-3	80			
Saint-Pierr	North-Ame	5759	########	2024-06-3	3452	2449	2	25400
Tuvalu	Oceania	12066	########	2024-06-3	2943		1	
Niue	Oceania	1622	########	2024-06-3	1059	1056		
Nicaragua	North-Ame	6779100	########	2024-06-3	18491	4225	225	
Tajikistan	Asia	9957464	########	2024-06-3	17786	17264	125	
Djibouti	Africa	1016097	########	2024-06-3	15690	15427	189	305941
Greenland	North-Ame	56973	########	2024-06-3	11971	2761	21	164926
Eritrea	Africa	3662244	########	2024-06-3	10189	10086	103	23693
Niger	Africa	26083660	########	2024-06-3	9931	8890	312	254538
Equatorial	Africa	1496662	########	2024-06-3	17229	16907	183	365697
Liberia	Africa	5305117	########	2024-06-3	8090	7783	295	139824
Nauru	Oceania	10903	########	2024-06-3	5393	5347	1	20509
Caribbean	North-Ame	26647	########	2024-06-3	11682	10476	38	30126
Bermuda	North-Ame	61939	########	2024-06-3	18860	18685	165	1029558
Gambia	Africa	2558482	########	2024-06-3	12626	12189	372	155686
Grenada	North-Ame	113475	########	2024-06-3	19693	19358	238	182981

covid_19

Methodology employed / Algorithms Implemented:

Step 1: ER Diagram:



Step 2: Dimension Modeling :

Information Package Diagram:

Time	Patient	Hospital	Doctor	Treatment	Insurance	Billing
Year	Patient_id(PK)	Hospital_id(PK)	Doctor_id(PK)	Treat_id(PK)	Insurance_id(PK)	Bill_id(PK)
Quater	Name	Hospital_Type	Name	Treat_Type	Insurance_Company	Date
Month	DOB	Hospital_Name	Specilization	Treatment_Cost	Insurance_Type	Amount Paid
Week	Gender	Address	Experience	Duration	Policy_No	Amount Due
Date	Address	Contact_No	Fees	Discharge_Time	Coverage_Amount	Status
Day	Contact_No	Capacity	Gender			

Facts:-Number Of Patient Encountered,Reports

Step 3: Implementation of all dimension tables and Fact Tables :

1. Dimension Tables

Each dimension table stores details about different aspects of the healthcare management system. The following dimension tables are considered:

1. Patient Dimension (Parent Dimension):

Keeps track of all patients along with their details such as Patient ID, Name, Date of Birth

(DOB), Gender, Address, and Contact Number.

Sub-Dimension:

- NameDimension
- DOBDimension
- GenderDimension
- Address Dimension
- Contact_no Dimension

2. Hospital Dimension (Parent Dimension):

Keeps track of all hospitals along with their details such as Hospital ID, Hospital Type, Hospital

Name, Address, Contact Number, and Capacity.

Sub-Dimensions:

- Hospital Type Dimension:
- Hospital Name Dimension:
- Address Dimension:
- Contact Number Dimension:
- Capacity Dimension:

3. Doctor Dimension (Parent Dimension):

Keeps track of all doctors along with their details such as Doctor ID, Name, Specialization,

Experience, Fees, and Gender.

Sub-Dimensions:

- NameDimension:
- Specialization Dimension:
- Experience Dimension:
- FeesDimension:
- GenderDimension:

4. Treatment Dimension (Parent Dimension):

Keeps track of all treatments along with their details such as Treatment ID, Treatment Type,

Treatment Cost, Duration, and Discharge Time.

Sub-Dimensions:

- Treatment Type Dimension:
- Treatment Cost Dimension:
- Duration Dimension:
- Discharge Time Dimension:

5. Insurance Dimension (Parent Dimension):

Keeps track of all insurance details such as Insurance ID, Insurance Company, Insurance Type,

Policy Number, and Coverage Amount.

Sub-Dimensions:

- Insurance Company Dimension:
- Insurance Type Dimension:
- Policy Number Dimension:
- Coverage Amount Dimension:

6. Billing Dimension (Parent Dimension):

Keeps track of all billing details such as Billing ID, Date, Amount Paid, Amount Due, and

Status.

Sub-Dimensions:

- DateDimension:
- AmountPaid Dimension:
- AmountDueDimension:
- Status Dimension:

7. Time Dimension (Parent Dimension):

Keeps track of time-related details such as Year, Quarter, Month, Week, Date, and Day.

Sub-Dimensions:

- YearDimension:
- Quarter Dimension:
- MonthDimension:
- WeekDimension:
- DateDimension:
- DayDimension:

Measures :

1. **Number of Patient Encounters:** Tracks the total number of patient interactions with the healthcare system, including visits, consultations, and treatments, helping to understand patient load and resource utilization.

2. **Reports**: Represents the generation and analysis of reports related to patient care, hospital operations, treatment outcomes, insurance claims, and billing statuses, aiding in decision-making and operational efficiency.

2. Fact Table

The fact table stores key metrics for analysis related to patient encounters and surgeries.

- **Attributes:**
  - Encounter_ID (Primary Key)
  - Patient_ID (Foreign Key referencing the Patient Dimension)
  - Doctor_ID (Foreign Key referencing the Doctor Dimension)
  - Hospital_ID(Foreign Key referencing the Hospital Dimension)
  - Treatment_ID (Foreign Key referencing the Treatment Dimension)
  - Insurance_ID (Foreign Key referencing the Insurance Dimension)
  - Time_ID (Foreign Key referencing the Time Dimension)
  - Number_of_Patient_Encounters
  - Number_of_Surgeries_Done

3. Relationships and Star Schema Structure

The fact table has foreign key relationships with the dimension tables to enable multidimensional analysis:

1. Patient Dimension (Parent Dimension)

- **Fact Table**: Patient_ID in the fact table refers to the Patient_ID in the **Patient Dimension**.
- **Relationship**: This relationship allows for the analysis of patient treatments, demographics, and medical history.

2. Hospital Dimension (Parent Dimension)

- **Fact Table**: Hospital_ID in the fact table refers to the Hospital_ID in the **Hospital Dimension**.
- **Relationship**: This relationship supports analyzing patient treatments, services, and capacity at each hospital.

3. Doctor Dimension (Parent Dimension)

- **Fact Table**: Doctor_ID in the fact table refers to the Doctor_ID in the **Doctor Dimension**.
- **Relationship**: This relationship supports the analysis of patient treatments based on the doctor's specialization, experience, and fees.

4. Treatment Dimension (Parent Dimension)

- **Fact Table**: Treatment_ID in the fact table refers to the Treatment_ID in the **Treatment Dimension**.
- **Relationship**: This allows for detailed analysis of treatments administered, including costs and duration.

5. Insurance Dimension (Parent Dimension)

- **Fact Table**: Insurance_ID in the fact table refers to the Insurance_ID in the **Insurance Dimension**.
- **Relationship**: This relationship enables analysis of insurance claims and coverage associated with patient treatments.

6. Billing Dimension (Parent Dimension)

- **Fact Table**: Billing_ID in the fact table refers to the Billing_ID in the **Billing Dimension**.
- **Relationship**: This supports the analysis of billing details, payment status, and financial tracking for patients.

7. Time Dimension (Parent Dimension)

- **Fact Table**: Time_ID in the fact table refers to the Time_ID in the **Time Dimension**.
- **Relationship**: This relationship allows for time-based analysis of patient treatments, hospital admissions, and billing cycles.

Step 4: Implementation of OLAP Operations

In this step, we implemented **Online Analytical Processing (OLAP)** operations to analyze healthcare data across multiple dimensions. First, we wrote a Python script to populate the **Fact_Healthcare_Analysis** table with **10,000 rows of data**. This was done by generating combinations of values from key dimensions: **Patient (e.g., demographics, medical history)**, **Doctor (e.g., specialty, experience)**, **Room (e.g., type, occupancy)**, **Surgery (e.g.,**

type, time), and **Time (e.g., year, month)**. Each combination was assigned random values for **Number of Patient Encounters**, **Number of Surgeries Done**, and **Room Occupancy**, ensuring that the fact table had sufficient data for in-depth analysis.

The Python script used nested loops to iterate over all possible combinations of 10 rows from each dimension, producing **10x10x10x10 rows** in total. After generating the data, we inserted these 10,000 rows into the **Fact_Healthcare_Analysis** table in **Microsoft SQL Server**. This allowed us to simulate real-world healthcare transactions and perform OLAP operations efficiently.

Next, we carried out four major types of OLAP operations—**Roll-Up**, **Drill-Down**, **Slice**, and **Dice**—using SQL queries. Each of these operations provided a different perspective on the data, allowing for deep, multidimensional analysis:

1. Roll-Up Operations:

Roll-up queries aggregate the data by grouping higher levels in the dimension hierarchy. For example, one query rolled up the data by **Year**, **Doctor Speciality**, and **Room Type** to provide insights into the overall performance of different doctors and room types across various years.

2. Drill-Down Operations:

Drill-down queries provided more granular views of the data. For example, one query drilled down to **monthly totals for patient encounters and surgery counts per doctor and room type**. Another query focused on drilling down by **room occupancy**, showing the impact of different room types (e.g., ICU, General Ward) on the **overall hospital utilization** and patient distribution.

3. Slice Operations:

Slice operations focus on filtering the data based on specific criteria. For instance, one query sliced the dataset to analyze the **surgery performance in 2023 for cardiovascular surgeries performed in ICU rooms**. Another query sliced the dataset to focus on **patient encounters with male doctors in the emergency ward**, providing insights into the correlation between doctor demographics and patient distribution.

4. Dice Operations:

Dice operations applied multiple filters to provide a multidimensional view of the data. For instance, one query diced the data to analyze the performance of **surgeries performed in ICU rooms for patients aged 60 and above in 2022 and 2023**, focusing on **doctor specialties** and the **patient's recovery rate**. Another query diced the data to evaluate the impact of **doctor experience and room type** on the **length of stay for pediatric patients** in hospitals across different months.

Step 5: Naive Bayes Algorithm for Classification

The methodology for building a Naive Bayes Medical Classifier involves loading health-related data from a CSV file, sanitizing the dataset to ensure proper formatting, and filling missing values with defaults. Each medical record contains attributes like age, blood pressure, hemoglobin, diagnosis, and more, which serve as features. The dataset is then split into training (80%) and testing (20%) sets. Using the training data, the model calculates the prior probabilities for each recovery time class (Low, Medium, High) and conditional probabilities for the observed feature values. During prediction, the classifier applies the Naive Bayes formula, combining prior probabilities with conditional probabilities to estimate the likelihood of each class. Laplace smoothing is used to handle zero probabilities. The model is tested on unseen data, and its accuracy is measured based on correct predictions. Once trained, the classifier can predict recovery times for new medical records, helping to anticipate patient recovery based on their health data.

Step 6: Transforming Continuous Variables (Data Discretization Using Equal Frequency) and Visualization

This methodology involves transforming continuous medical features into categorical bins using equal-width discretization. After loading the dataset from a CSV file, the continuous feature "Blood Pressure" is discretized into five equal-width bins. The minimum and maximum values of "Blood Pressure" are calculated, and the range is divided into equal intervals. Each data point is assigned to one of these intervals, which are labeled as "Very Low," "Low," "Normal," "High," and "Very High." This transformation helps in categorizing continuous data for better analysis and visualization. Additionally, the "Blood Pressure" values are normalized to a scale of 0 to 1, which facilitates comparison across datasets. The final dataset, containing both the discretized and normalized values of "Blood Pressure," is then saved to a CSV file for further analysis or visualization in tools such as Tableau.

Step 7: Data Clustering Using K-Means Algorithm

This methodology applies the K-means clustering algorithm to group countries based on similar COVID-19-related attributes. The process begins by loading the dataset, which contains features such as population, total cases, recovered cases, deaths, and tests performed in each country. Missing values in the dataset are handled by imputing the mean values, ensuring the data is complete for analysis. Key features are then standardized to ensure that each attribute is on the same scale, thus allowing a fair comparison across countries.

K-means clustering is applied with a specified number of clusters (in this case, three). K-means assigns each data point (i.e., country) to the nearest centroid, which represents the center of a cluster, and iteratively updates these centroids to minimize the distance between data points and their respective centroids. After clustering, the centroids (cluster centers) are analyzed to understand the key characteristics of each group, such as average population, total cases, recovered cases, deaths, and tests conducted.

Finally, a scatter plot is generated to visualize the clustering results, highlighting the countries' positions relative to the centroids based on standardized values for population and cases. The countries are labeled by their respective cluster assignments, allowing for a clear visual interpretation of the clustering results. This approach provides insights into how countries group together based on their COVID-19 profiles, which can help identify trends in case severity, recovery rates, and testing efforts across different clusters.

Step 8: Data Classification Using C4.5 Algorithm

This methodology utilizes a decision tree classifier, similar to the C4.5 algorithm, to classify patients based on medical features. The dataset is loaded, containing features like age, blood pressure, and gender, with the target variable being the type of disease. Categorical data such as gender is transformed into numerical values using one-hot encoding, while numeric features like age and blood pressure are standardized. Missing values are handled by imputing the mean for numeric features and a constant placeholder for categorical ones. The dataset is split into training and testing sets, and a decision tree is trained with a simplified C4.5 approach, limiting the tree depth to 3 to prevent overfitting. The decision tree uses information gain to select the best features for classification. After training, the decision tree model is visualized, showing how different attributes contribute to disease classification. This approach helps in predicting disease types for new patients based on their features and provides insights into the key factors influencing the classification.

Step 9: Data Clustering Using DBSCAN Algorithm

The methodology for clustering using DBSCAN (Density-Based Spatial Clustering of Applications with Noise) involved analyzing a COVID-19 dataset to identify groups of similar observations based on specific features. Initially, the dataset was loaded, and the features continent and population were selected for clustering. Categorical data, such as continent, was converted into numerical format using Label Encoding, ensuring fair treatment of variables. The DBSCAN algorithm was then applied with parameters for radius (eps) and minimum samples required to form a dense region, allowing it to uncover clusters while identifying noise points as outliers. A function was created to optimize these parameters, ensuring at least five clusters were formed. Finally, the clustering results were visualized through a scatter plot, illustrating the relationships between continent and population, and highlighting any noise points that indicated anomalous observations.

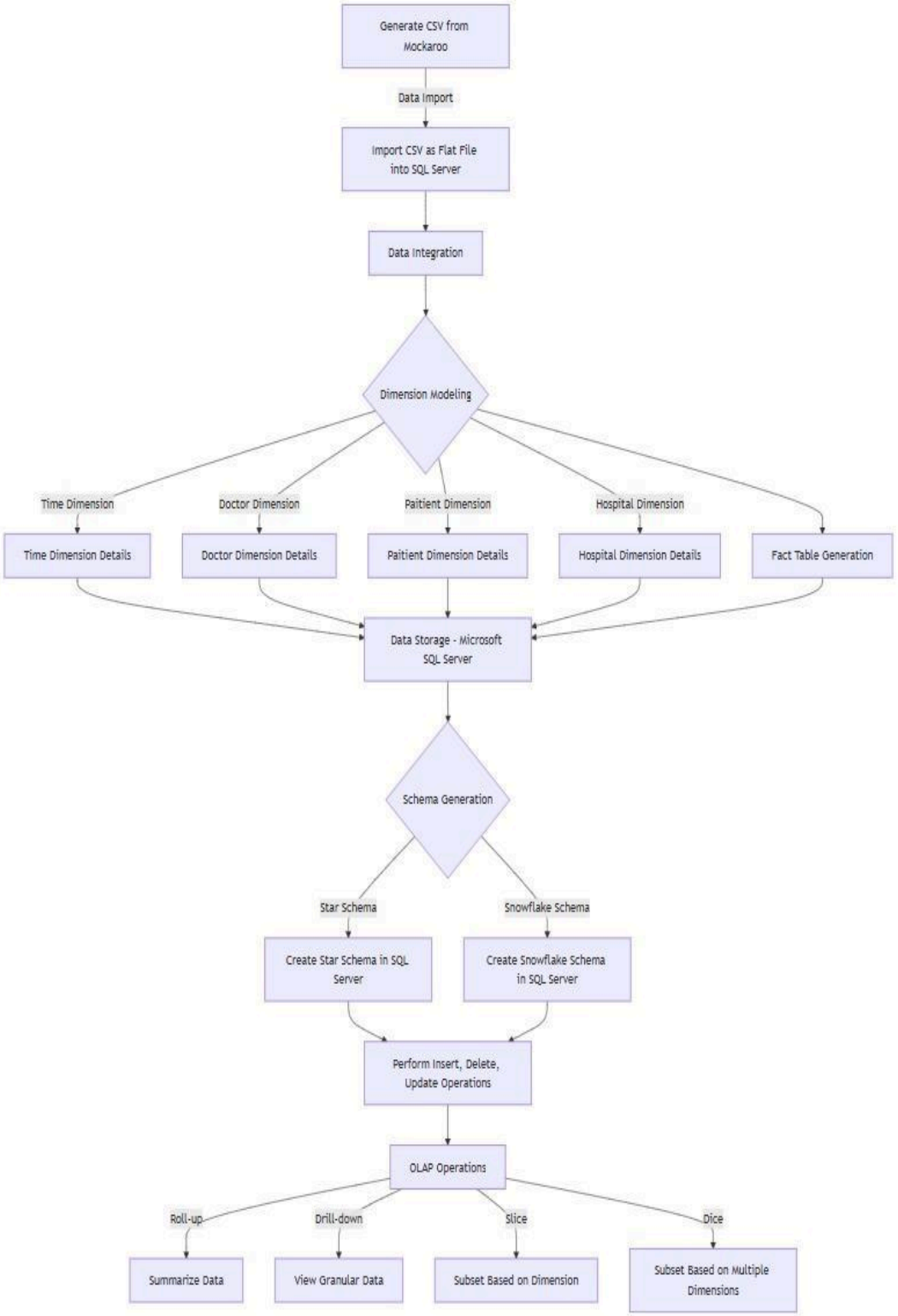
Step 10: Apriori Algorithm

The implementation employs a Decision Tree Classifier to analyze COVID-19 case data, transforming it into a binary classification problem. Initially, irrelevant columns are dropped, and missing values are addressed through median imputation. A new target variable, `case_category`, is created based on a threshold of 1000 cases, categorizing data into high and low cases. The numerical features are then standardized using the StandardScaler before splitting the dataset into training and testing sets. After training the model, predictions are made, and performance is evaluated using accuracy metrics and a confusion matrix, which is visualized to provide insights into the model's classification effectiveness.

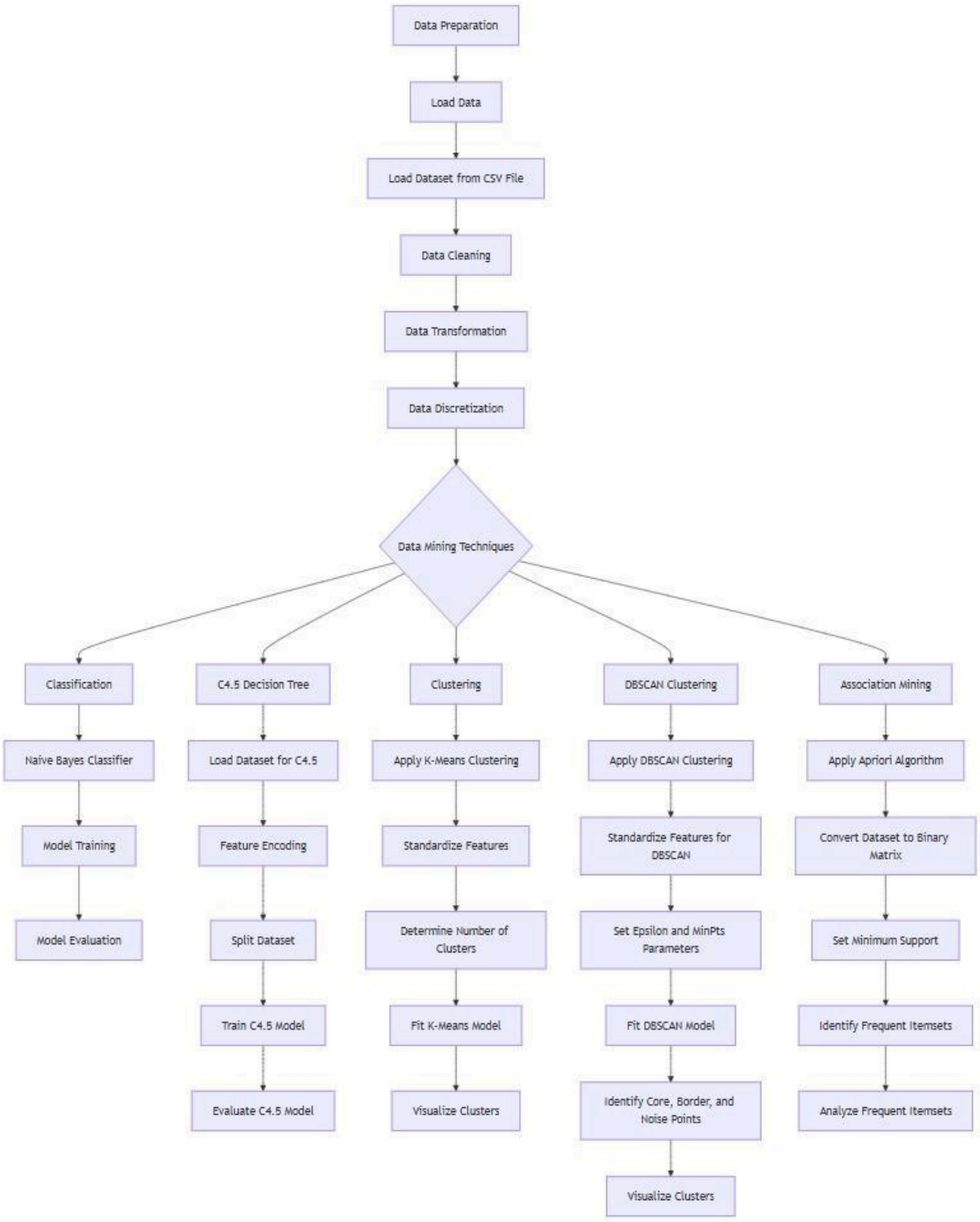
Additionally, the Apriori algorithm is utilized to discover frequent itemsets within the dataset by first binning numeric columns into categorical ranges. This preprocessing step enables effective association rule mining, which identifies relationships between different COVID-19 metrics. The algorithm operates with a minimum support threshold of 5%, resulting in frequent itemsets that are analyzed further to generate association rules based on a confidence threshold of 60%. Key metrics such as support, confidence, and lift are calculated for these rules, allowing for a detailed evaluation of the associations discovered. This analysis aids in understanding trends and patterns in COVID-19 data, facilitating informed decision-making based on the identified relationships.

Flow Diagram:

Data Warehouse Diagram :

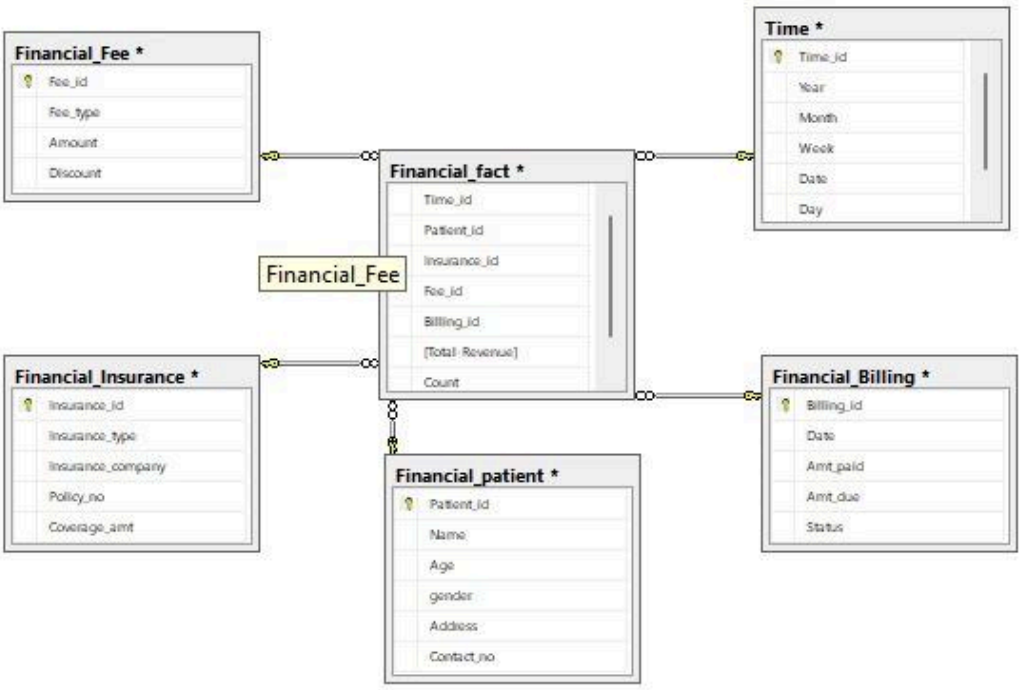


Data Mining Diagram :

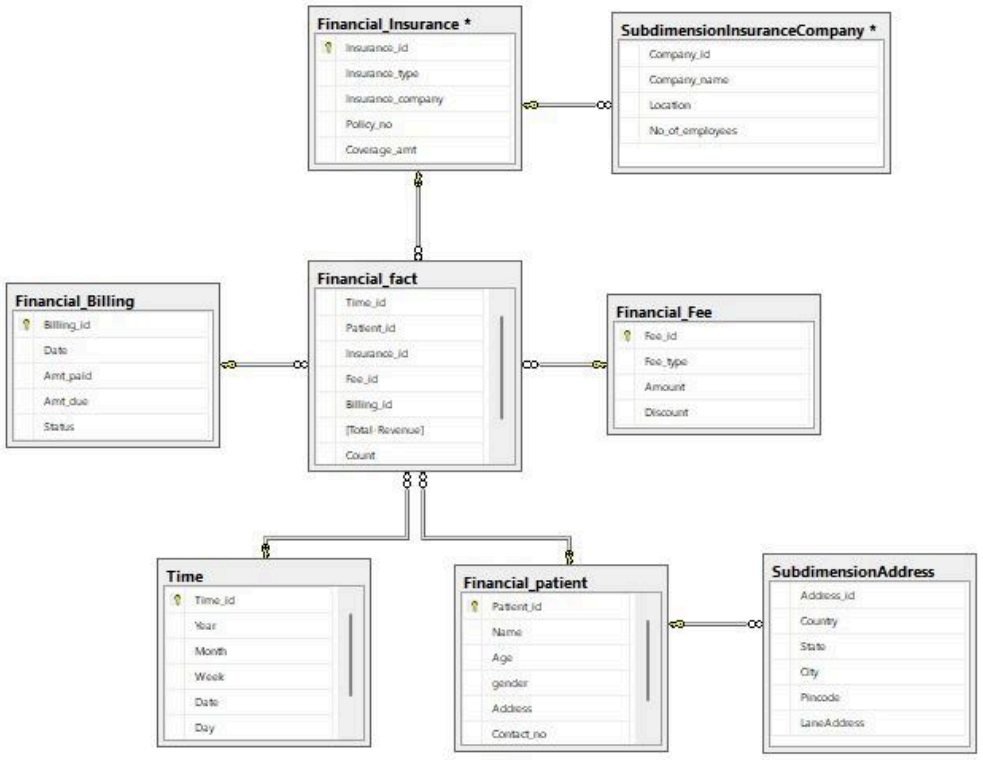


Results (Screenshots):

Star Diagram:



Snowflake Diagram



Implementation of all dimensions table and fact table

```
SELECT TOP 1000 [Doctor_id]
      ,[Name]
      ,[Speciality]
      ,[Gender]
      ,[Fee]
      ,[Experience]
  FROM [D12A4].[dbo].[Operational_Doctor]
```

	Doctor_id	Name	Speciality	Gender	Fee	Experience
1	1	Kelly Trusty	Cardiologist	F	73284	8
2	2	Leonidas Barthot	Neurologist	F	25591	11
3	3	Maggi Horley	MD	F	45491	12
4	4	Trixi De L'Isle	Cardiologist	M	52017	4
5	5	Steame Weld	MD	M	66240	16
6	6	Ynez Bonnell	MD	M	50670	38
7	7	Ben Byne	Gynaecologist	F	9425	41
8	8	Goldina Harp	MD	M	3145	12
9	9	Sena Innerstone	MBBS	F	1868	31
10	10	Joni Trotman	MD	F	21557	19
11	11	Onfroi Tregale	Dentist	M	26553	20

Insert Statements :

```
insert into [Fact(Operation)] values(1,1,1,1,1,2,5,3);
```

```
) %
Messages

(1 row(s) affected)
```

	Doctor_id	Patient_id	Room_id	Surgery_id	Time_id	Total_Surgery	Total_Patient	Total_Room_occupied
1	1	1	1	1	1	2	5	3
2	1	1	1	1	1	2	5	3
3	2	2	2	2	2	5	6	4

Delete Statements :

```
Delete from Operational_Doctor where Doctor_id=3;
```

```
100 %
Results  Messages
```

	Doctor_id	Name	Speciality	Gender	Fee	Experience
1	1	Kelly Trusty	Cardiologist	F	73284	8
2	2	Leonidas Barthot	Neurologist	F	25591	11
3	4	Trixi De L'Isle	Cardiologist	M	52017	4
4	5	Steame Weld	MD	M	66240	16
5	6	Ynez Bonnell	MD	M	50670	38
6	7	Ben Byne	Gynaecologist	F	9425	41
7	8	Goldina Harp	MD	M	3145	12
8	9	Sena Innerstone	MBBS	F	1868	31
9	10	Joni Trotman	MD	F	21557	19
10	11	Onfroi Tregale	Dentist	M	26553	20
11	12	Arley Greenless	Neurologist	M	77849	7

Update Statement :

```
UPDATE Operational_Doctor
SET Name='Ashok Kumar'
where Doctor_id=2;
```

```
(1 row(s) affected)
```

	Doctor_id	Name	Speciality	Gender	Fee	Experience
1	1	Kelly Trusty	Cardiologist	F	73284	8
2	2	Ashok Kumar	Neurologist	F	25591	11

Implementation of OLAP Operations (Roll-up,Drill-down,Slice&Dice):

```
SELECT TOP (1000) [Patient_id]
      ,[Doctor_id]
      ,[Room_id]
      ,[Surgery_id]
      ,[Time_id]
      ,[Total_Surgery]
      ,[Total_Patient]
      ,[Total_Room_Occupied]
  FROM [HealthcareSyatem].[dbo].[Operational_Fact]
```

Results Messages

Patient_id	Doctor_id	Room_id	Surgery_id	Time_id	Total_Surgery	Total_Patient	Total_Room_Occupied
138	539	181	476	15	609086	157	17
239	67	747	19	375	636065	219	356
392	33	723	423	118	765923	166	72
398	55	686	946	504	232714	92	184
535	975	475	98	463	152014	48	239
909	88	741	912	406	938773	295	238
177	335	112	703	752	620661	297	195
125	987	657	73	421	488502	161	499
670	693	639	629	941	480324	211	21
248	323	266	861	633	642671	377	220
661	351	942	941	357	460341	213	152
521	790	113	974	115	513988	290	393
472	403	966	16	325	896344	463	119
200	444	45	938	469	667619	298	464
848	888	513	477	420	271835	465	376
363	684	23	405	737	435861	190	451

Roll-up:

QUERY 1 : Aggregate by Year, Insurance Type, and Patient Gender and finding
Total Revenue and Count

```
SELECT
T.Year,
I.Insurance_type,
P.gender,
SUM(F.Revenue) AS Total_Revenue,
SUM(F.Count) AS Total_Count
FROM
Financial_fact F
JOIN
Time T ONF.Time_id = T.Time_id
JOIN
Financial_Insurance I ON F.Insurance_id = I.Insurance_id
JOIN
Financial_patient P ON F.Patient_id = P.Patient_id
GROUPBY
T.Year,
I.Insurance_type,
P.gender;
```

	Year	Insurance_type	gender	Total_Revenue	Total_Count
1	1959	Accident	F	805684	314
2	1959	Accident	M	760835	856
3	1959	Cancer	F	1257633	94
4	1959	Mediclaim	M	557730	346
5	1959	MediPlus	M	474185	22
6	1959	Termlife	M	171748	428
7	1964	Accident	F	1039041	613
8	1964	Accident	M	408158	404
9	1964	Cancer	M	934165	114

Drill-down:

Query:Break Down by Month to find total revenue month wise and year wise and of particular insurance type

```
SELECT
T.Year,
T.Month,
I.Insurance_type,
SUM(F.Revenue) AS Total_Revenue,
```

```
SUM(F.Count) AS Total_Count
FROM
Financial_Fact F
JOIN
Time T ONF.Time_id = T.Time_id
JOIN
Financial_Insurance I ON F.Insurance_id = I.Insurance_id
GROUPBY
T.Year,
T.Month,
I.Insurance_type
```

	Patient_id	Billing_id	Fee_id	Insurance_id	Time_id	Revenue	Count
1	425	97	514	489	1	206584	153
2	1000	636	691	481	480	541081	184
3	854	88	96	33	848	638168	153
4	577	128	466	405	744	245842	255
5	722	505	677	205	417	598352	438
6	845	756	757	925	647	297803	282

	Year	Month	Insurance_type	Total_Revenue	Total_Count
1	1959	December	Accident	1143994	696
2	1959	December	Cancer	483869	12
3	1959	November	Accident	422525	474
4	1959	November	Cancer	773764	82
5	1959	November	Mediclaim	557730	346
6	1959	November	MediPlus	474185	22
7	1959	November	Termlife	171748	428
8	1964	March	Accident	1447199	1017
9	1964	March	Cancer	934165	114
10	1964	March	MediPlus	257195	374

Slice:

Query: find Total Revenue and count filter by year i.e 2000, Insurance_type i.e Mediclaim and Fee type i.e Claim Fee

```
SELECT
T.Year,
I.Insurance_type,
Fe.Fee_id,
Fe.Fee_type,
SUM(F.Revenue) AS Total_Revenue,
SUM(F.Count) AS Total_Count
FROM
Financial_fact F
JOIN
Time T ONF.Time_id = T.Time_id
```

```
JOIN
Financial_Insurance I ON F.Insurance_id = I.Insurance_id
JOIN
Financial_Fee Fe ON f.Fee_id = Fe.Fee_id
WHERE
T.Year = 2000
ANDI.Insurance_type = 'Mediclaim'
ANDFe.Fee_type = 'Claim Fee'
GROUPBY
T.Year,
I.Insurance_type,
Fe.Fee_id,
Fe.Fee_type
```

	Year	Insurance_type	Fee_id	Fee_type	Total_Revenue	Total_Count
1	2000	Mediclaim	440	Claim Fee	524504	404
2	2000	Mediclaim	768	Claim Fee	985480	292
3	2000	Mediclaim	809	Claim Fee	326354	427

Dice:

Query 1: Revenue and Payments for Specific Insurance Types, Gender and Time Periods(Year wise)

```
SELECT
I.Insurance_type,
T.Year,
P.gender,
SUM(F.Revenue) AS Total_Revenue,
SUM(F.Count) AS Total_Payments
FROM
Financial_fact F
JOIN
Financial_Insurance I ON F.Insurance_id = I.Insurance_id
JOIN
Time T ONF.Time_id = T.Time_id
JOIN
Financial_patient P ON F.Patient_id = P.Patient_id
WHERE
I.Insurance_type IN ('Accident', 'Cancer')
ANDT.Year IN (2022, 2023,2008,2003,2000)
ANDp.gender IN ('F','M')
GROUPBY
I.Insurance_type,
T.Year,
P.gender;
```

	Insurance_type	Year	gender	Total_Revenue	Total_Payments
1	Accident	2000	F	3131795	1141
2	Accident	2000	M	5538601	2159
3	Accident	2003	F	5938648	2264
4	Accident	2003	M	6898019	2306
5	Accident	2008	F	4034859	1057
6	Accident	2008	M	6791739	3060
7	Cancer	2000	F	3818240	1839
8	Cancer	2000	M	3500207	1358
9	Cancer	2003	F	2755484	989
10	Cancer	2003	M	5520987	1816
11	Cancer	2008	F	2029771	849
12	Cancer	2008	M	4383724	2242

Naïve Bayes Classifier:

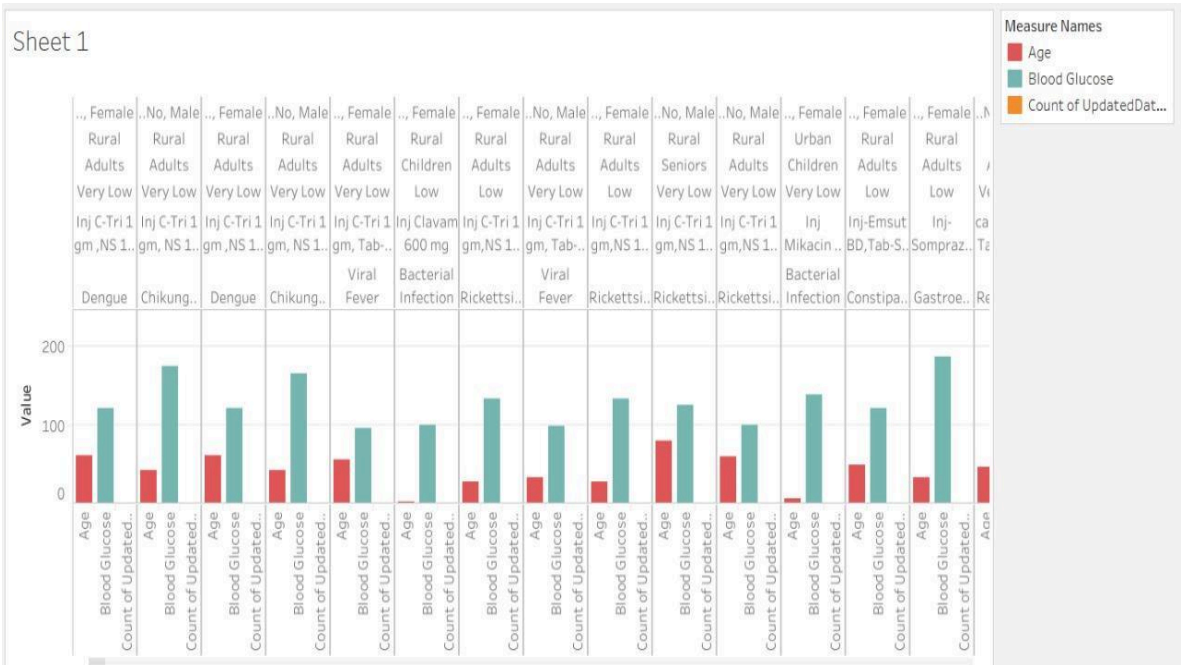
```
Actual: Medium, Predicted: Low
Actual: High, Predicted: High
Actual: High, Predicted: High
Accuracy: 0.8765133171912833

Process finished with exit code 0
```

Transforming Continuous Variables and Visualizing the Patterns: Data Discretization (Equal-Frequency) and Visualization

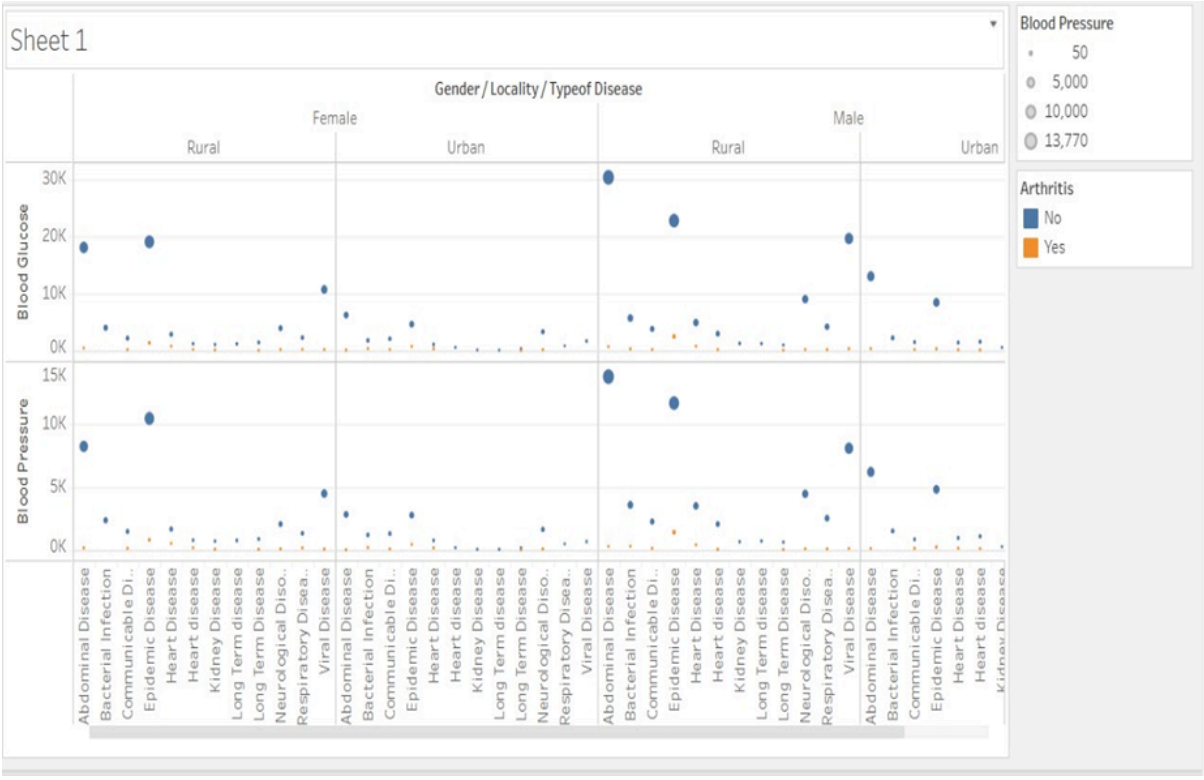
```
   ID  Age AgeGroup  Gender  ...                              Medicine  TypeofMedicine  Locality    RecoveryTime
0   1   62   Adults  Female  ...                 Inj C-Tri 1 gm ,NS 100ml      Antibiotic     Rural   0 to 48 Hours
1   2   43   Adults    Male  ...  Inj C-Tri 1 gm, NS 100ml ,Tab HCQS 200      Antibiotic     Rural   0 to 48 Hours
2   3   62   Adults  Female  ...                 Inj C-Tri 1 gm ,NS 100ml      Antibiotic     Rural   0 to 48 Hours
3   4   43   Adults    Male  ...  Inj C-Tri 1 gm, NS 100ml ,Tab HCQS 200      Antibiotic     Rural   0 to 48 Hours
4   5   56   Adults  Female  ...       Inj C-Tri 1 gm, Tab-Montegress LC      Antibiotic     Rural  48 to 72 Hours

[5 rows x 16 columns]

Number of Intervals (Bins): 5

Blood Pressure Range: (50, 180)
Interval Width: 26.0

Created Intervals and Discrete Values:
Interval 1: (50.00, 76.00) - Value: Very Low
Interval 2: (76.00, 102.00) - Value: Low
Interval 3: (102.00, 128.00) - Value: Normal
Interval 4: (128.00, 154.00) - Value: High
Interval 5: (154.00, 180.00) - Value: Very High

Discretized 'BloodPressure' Column:
   BloodPressure BloodPressure (Discretized)
0             70                    Very Low
1             60                    Very Low
2             70                    Very Low
3             60                    Very Low
4             60                    Very Low

Normalized 'BloodPressure' Column:
   BloodPressure  BloodPressure (Normalized)
0             70                    0.153846
1             60                    0.076923
2             70                    0.153846
3             60                    0.076923
4             60                    0.076923
```

Visualization using Graphs –

1. Bar Graph

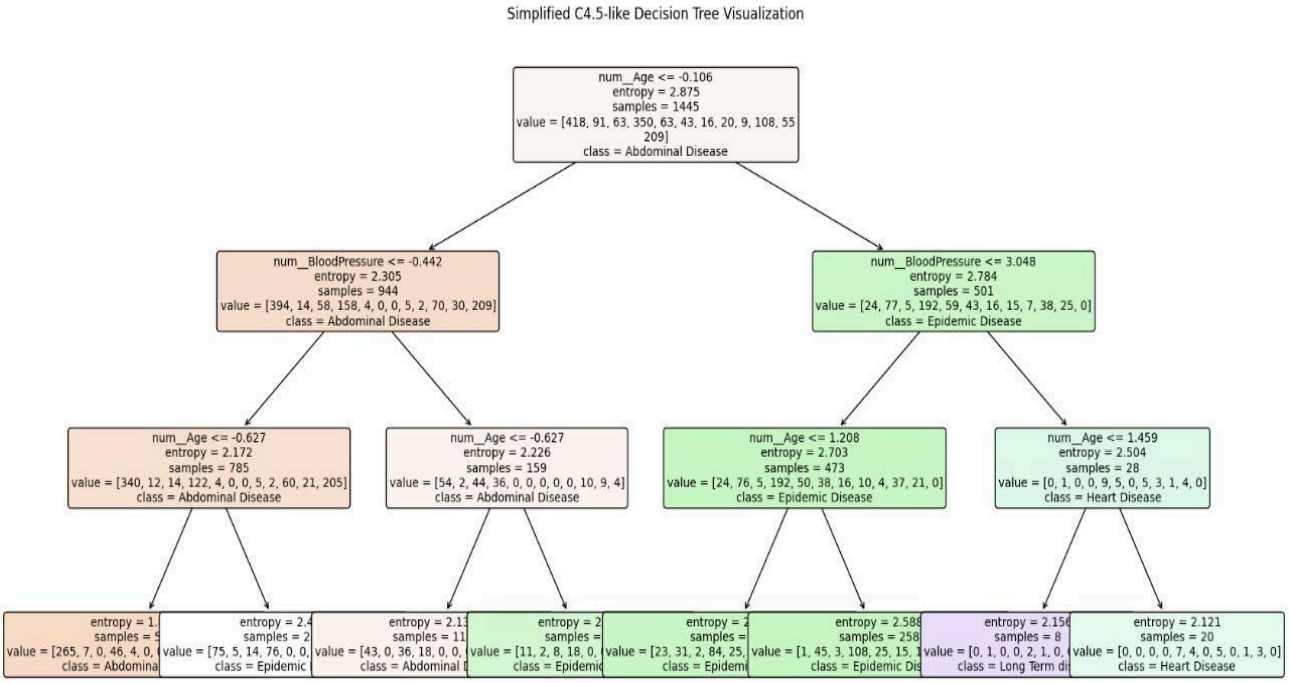


2. Pie Chart

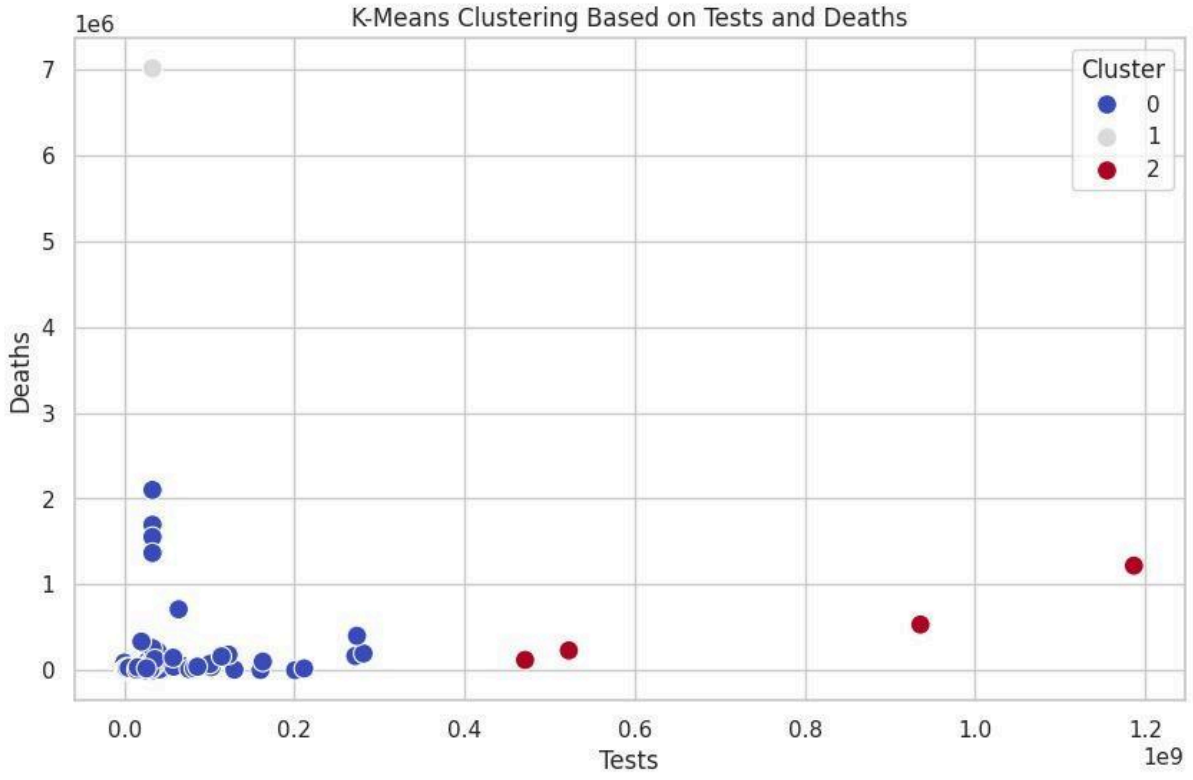


-

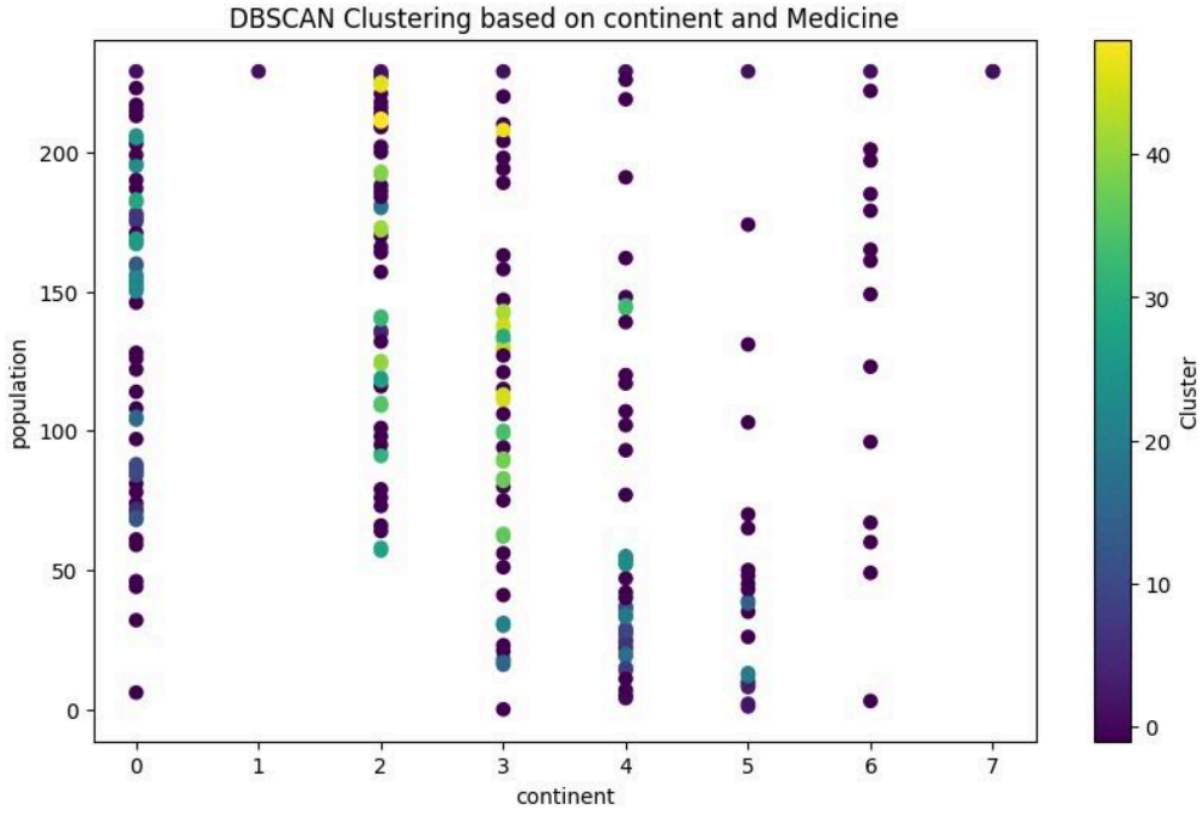
Implementation of Decision tree classifier using C4.5 algorithm.



Implementation of Clustering using the K-Means algorithm:



Implementation of DBSCAN algorithm:

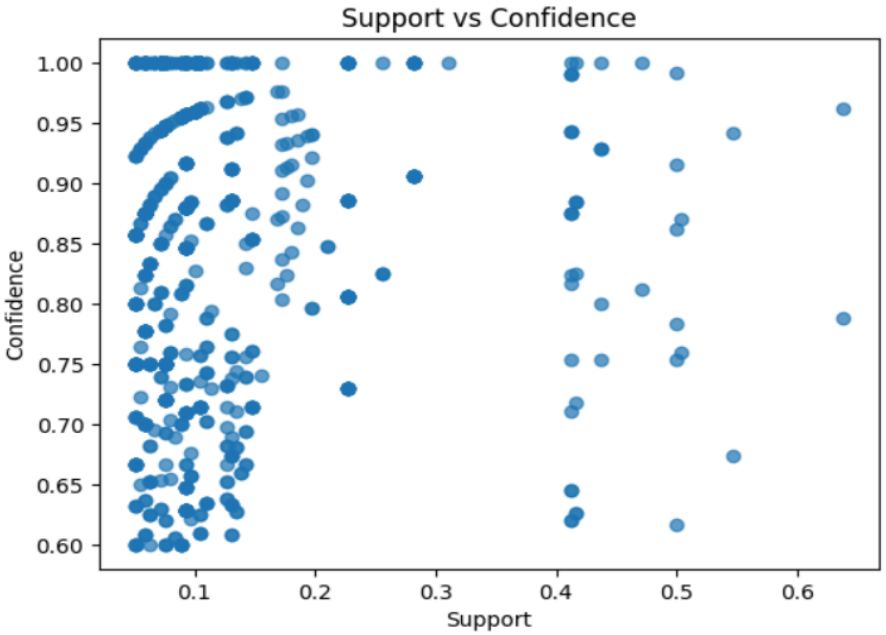


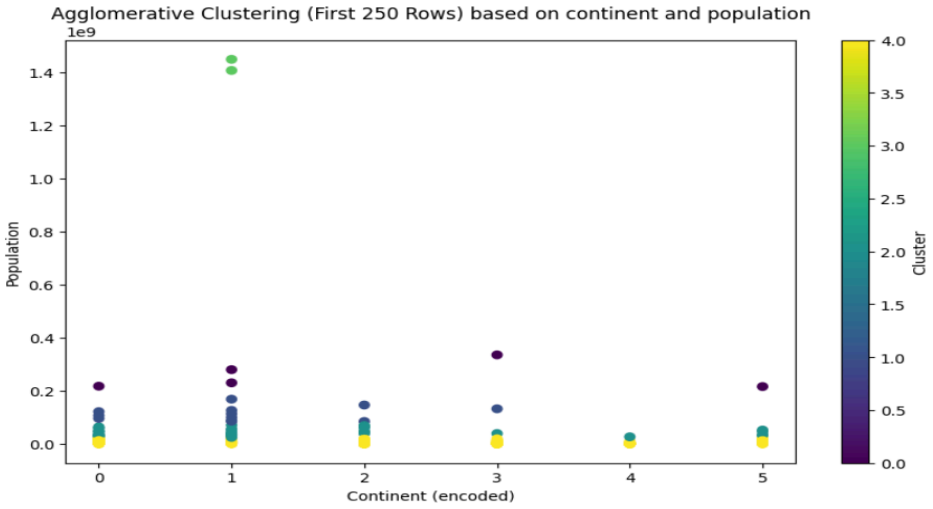
Implementation of Association Rule Mining algorithm (Apriori / FP Growth):

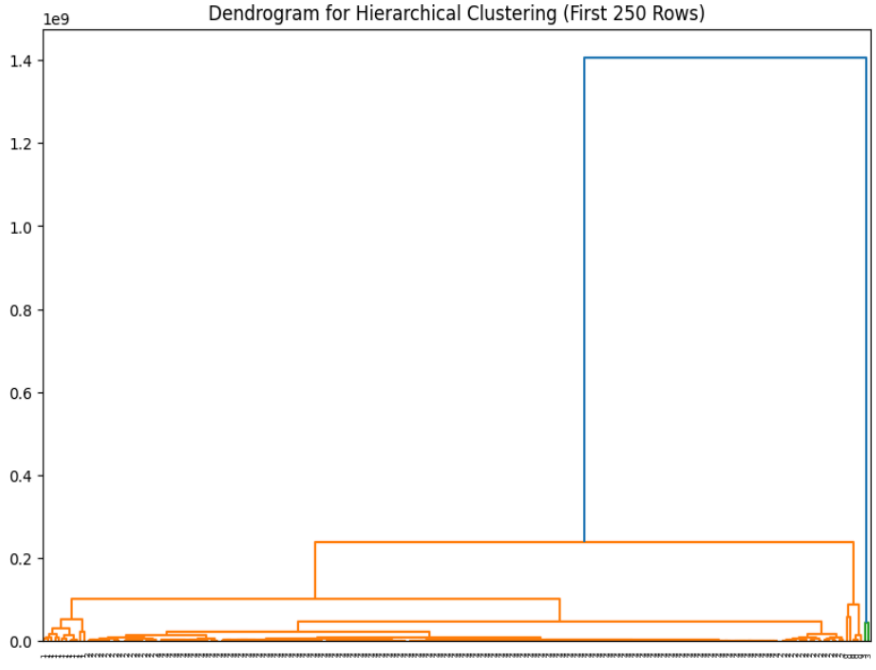
Candidate Sets (Frequent Itemsets) with Support and Support Count:

	support	itemsets	support_count
0	0.600000	(Milk)	9.0
1	0.533333	(Bread)	8.0
2	0.600000	(Eggs)	9.0
3	0.533333	(Cheese)	8.0
4	0.533333	(Butter)	8.0
5	0.466667	(Yogurt)	7.0
6	0.466667	(Apples)	7.0
7	0.533333	(Ice Cream)	8.0
8	0.400000	(Eggs, Milk)	6.0
9	0.400000	(Bread, Cheese)	6.0
10	0.400000	(Eggs, Ice Cream)	6.0
11	0.400000	(Butter, Ice Cream)	6.0

```
Strong Associations (Confidence >= 0.6) with Support and Support Count:
   antecedents  consequents  support  support_count  confidence     lift
2      (Bread)     (Cheese)      0.4            6.0        0.75  1.40625
3     (Cheese)      (Bread)      0.4            6.0        0.75  1.40625
5  (Ice Cream)       (Eggs)      0.4            6.0        0.75  1.25000
6     (Butter)  (Ice Cream)      0.4            6.0        0.75  1.40625
7  (Ice Cream)     (Butter)      0.4            6.0        0.75  1.40625
```


Support vs Confidence
Confidence
Support
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.1
0.2
0.3
0.4
0.5
0.6


Agglomerative Clustering (First 250 Rows) based on continent and population
1e9
Population
Continent (encoded)
Cluster
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0


Dendrogram for Hierarchical Clustering (First 250 Rows)
1e9
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0

Conclusion:

The Healthcare Management System project represents a significant leap forward in the application of data analytics to healthcare operations. By transforming a traditional database into a comprehensive data warehouse, this system enables healthcare institutions to harness the full potential of their data. The implementation of dimension tables for patients, doctors, rooms, surgeries, and time, along with a central fact table, creates a robust structure for multidimensional analysis. This star schema design facilitates complex queries and insights that were previously difficult or impossible to obtain.

The integration of advanced analytical techniques, including OLAP operations, Naive Bayes classification, K-means clustering, and the C4.5 algorithm, demonstrates the system's capacity for sophisticated data analysis. These tools allow healthcare providers and administrators to uncover hidden patterns, predict outcomes, and make data-driven decisions. From forecasting patient recovery times to identifying trends in COVID-19 data, the system provides a wide range of analytical capabilities that can significantly improve patient care, resource allocation, and operational efficiency.

Perhaps most importantly, this Healthcare Management System sets the stage for continuous improvement in healthcare delivery. By providing a platform for ongoing data collection, analysis, and insight generation, it enables healthcare institutions to adapt quickly to changing circumstances and evolving best practices. The ability to visualize data, discover associations through techniques like the Apriori algorithm, and classify data points using decision trees offers a comprehensive toolkit for addressing complex healthcare challenges. As this system is implemented and refined, it has the potential to drive substantial improvements in healthcare quality, accessibility, and cost-effectiveness, ultimately benefiting patients, providers, and the healthcare system as a whole.

References:

1. Paulraj Ponniah, “Data Warehousing: Fundamentals for IT Professionals”, Wiley India.
2. Han, Kamber, "Data Mining Concepts and Techniques", Morgan Kaufmann 3rd Edition.
3. Dataset References:
    - kaggle.com