**2. star:**

create database Star\_Schema;

use Star\_Schema;

create table Fact\_Sales(

Sales\_Rep\_id int Primary key,

Time\_id int not null,

Order\_id int null,

Customer\_id int not null,

Product\_id int not null,

Net\_amount\_per\_customer int not null,

Net\_amount\_per\_product int not null,

Net\_amount\_per\_promotion int not null

);

INSERT INTO Fact\_Sales VALUES

('1', '7','2', '3', '6', '0', '0','0'),

('2', '9','10', '5', '8','0', '0','0'),

('3', '6','5', '7', '1', '0', '0','0'),

('4', '1','8', '9', '3', '0', '0','0'),

('5', '10','7', '1', '9', '0', '0','0'),

('6', '4','1', '10', '7','0', '0','0'),

('7', '8','3', '8', '10','0', '0','0'),

('8', '2','4', '6', '2', '0', '0','0'),

('9', '5','6', '4', '5', '0', '0','0'),

('10', '3','9', '2', '4', '0', '0','0');

select \* from Fact\_Sales;

create table Dim\_Sales\_Rep(

Sales\_Rep\_id int not null,

Name varchar(30) not null,

Deal varchar(30)not null,

Discount int not null,

Primary key(Sales\_Rep\_id)

);

INSERT INTO Dim\_Sales\_Rep VALUES

('1', 'Manish','Yes', '20'),

('2', 'Shabnam','Yes', '15'),

('3', 'Pratik','No', '30'),

('4', 'Rajesh','No', '7'),

('5', 'Mitali','Yes', '10'),

('6', 'Priyal','Yes', '13'),

('7', 'Sachin','No', '19'),

('8', 'Amit','Yes', '10'),

('9', 'Rutuja','No', '8'),

('10', 'Shreyani','Yes', '15');

select \* from Dim\_Sales\_Rep;

create table Dim\_Time (

Time\_id int not null,

day varchar(30) not null,

month varchar(30) not null,

year int not null,

primary key(Time\_id)

);

INSERT INTO Dim\_Time VALUES

('1', 'Monday','Jan', '2010'),

('2', 'Tuesday','Feb', '2011'),

('3', 'Wednesday','March', '2012'),

('4', 'Thrusday','April', '2016'),

('5', 'Friday','May', '2017'),

('6', 'Monday','June', '2019'),

('7', 'Tuesday','July', '2011'),

('8', 'Wednesday','Oct', '2014'),

('9', 'Thrusday','Nov', '2016'),

('10', 'Friday','Dec', '2021');

select \* from Dim\_Time;

create table Dim\_Customer (

Customer\_id int not null,

Name varchar(30) not null,

billing\_address varchar(60) not null,

shipping\_address varchar(60) not null,

primary key(Customer\_id)

);

INSERT INTO Dim\_Customer VALUES

('1', 'Neha','abc', 'abc'),

('2', 'Ved','def', 'def'),

('3', 'Sameer','ghi', 'ghi'),

('4', 'Taniya','jki', 'jki'),

('5', 'Anvi','mno', 'mno'),

('6', 'Rohit','pqr', 'pqr'),

('7', 'Ayansh','stu', 'xyz'),

('8', 'Anjali','vu', 'stu'),

('9', 'Piyu','abc', 'xyz'),

('10', 'Anju','xyz', 'xyz');

select\*from Dim\_Customer;

create table Dim\_Product (

Product\_id int not null,

product\_number int not null,

type varchar(30) not null,

quality varchar(30) not null,

price int not null,

requested\_ship\_date date not null,

primary key(Product\_id)

);

INSERT INTO Dim\_Product VALUES

('1','101','Mouse','A1','700','2022-09-16'),

('2','102','Keyboard','A1','1500','2022-08-09'),

('3','103','RAM','A1','5000','2022-08-01'),

('4','104','USB cable','A1','600','2022-07-23'),

('5','105','Pendrive','A1','800','2022-07-11'),

('6','106','Hard Disk','A1','4000','2022-06-24'),

('7','107','Charger','A1','500','2022-04-08'),

('8','108','Ethernet Cable','A1','1200','2022-03-12'),

('9','109','Graphic Card','A1','6000','2022-02-16'),

('10','110','Printer','A1','6500','2022-01-01');

select \* from Dim\_Product;

create table Dim\_Order (

Order\_id int not null,

order\_number int not null,

order\_date date not null,

amount int not null,

primary key(Order\_id)

);

INSERT INTO Dim\_Order VALUES

('1','1001','2022-09-16','1'),

('2','1002','2022-08-09','2'),

('3','1003','2022-08-01','6'),

('4','1004','2022-07-23','1'),

('5','1005','2022-07-11','2'),

('6','1006','2022-06-24','3'),

('7','1007','2022-04-08','1'),

('8','1008','2022-03-12','3'),

('9','1009','2022-02-16','4'),

('10','1010','2022-01-01','1');

select \* from Dim\_Order;

**3. star:**

create database Book\_Management;

use Book\_Management;

create table Fact\_BookManagement(

Author\_Id int not null,

Book\_Id int not null,

Location\_Id int not null,

Publication\_Id int not null,

Cost int not null,

Quantity int not null,

Profit int not null

);

Insert into Fact\_BookManagement

values

('1', '7','2', '3', '1200', '3','1200'),

('2', '9','10', '5','1000', '2','650'),

('3', '6','5', '7','940', '1','200'),

('4', '1','8', '9','1150', '4','1600'),

('5', '10','7', '1','1300', '2','800'),

('6', '4','1', '10','1050', '5','1550'),

('7', '8','3', '8','930', '3','1090'),

('8', '2','4', '6','870', '1','170'),

('9', '5','6', '4','800', '2','600'),

('10', '3','9', '2','900', '3','900');

select \* from Fact\_BookManagement;

create table Dim\_Author(

Author\_Id int not null,

Author\_Name varchar(30) not null,

Authour\_Age int not null,

Author\_Country varchar(30) not null,

primary key(Author\_Id)

);

INSERT INTO Dim\_Author VALUES

('1', 'Amit Garg','72','India'),

('2', 'JK Rowling','78','UK'),

('3', 'HC Verma','80','India'),

('4', 'Cao Xueqin','42','China'),

('5', 'RK Narayan','65','India'),

('6', 'Charles Dickens','59','UK'),

('7', 'CS Lewis','57','UK'),

('8', 'Sanjesh Pawale','48','India'),

('9', 'DC Pandey','76','India'),

('10', 'Dan Brown','63','USA');

select \* from Dim\_Author;

create table Dim\_Location (

Location\_Id int not null,

Location\_Country varchar(30) not null,

primary key(Location\_Id)

);

INSERT INTO Dim\_Location VALUES

('1', 'India'),

('2', 'UK'),

('3', 'USA'),

('4', 'India'),

('5', 'UK'),

('6', 'India'),

('7', 'USA'),

('8', 'USA'),

('9', 'India'),

('10', 'UK');

select \* from Dim\_Location;

create table Dim\_Book (

Book\_Id int not null,

Book\_Type varchar(30) not null,

primary key(Book\_Id)

);

INSERT INTO Dim\_Book VALUES

('1', 'Horror'),

('2', 'Fantasy'),

('3', 'Mystery'),

('4', 'Classics'),

('5', 'Fiction'),

('6', 'Adventure'),

('7', 'Educational'),

('8', 'Historical'),

('9', 'Poetry'),

('10','Comedy');

select \* from Dim\_Book;

create table Dim\_Publication (

Publication\_Id int not null,

Publication\_Name varchar(30) not null,

Publication\_Country varchar(30) not null,

Publication\_Year int not null,

primary key(Publication\_Id)

);

INSERT INTO Dim\_Publication VALUES

('1', 'Readers Zone','UK', '2004'),

('2', 'Arihant','India', '1999'),

('3', 'Art Book','India', '2006'),

('4', 'Jaico Publishing House','India', '1995'),

('5', 'Course Technology','USA', '1993'),

('6', 'Pan Macmillan','India', '1991'),

('7', 'Penguin Books','UK', '2001'),

('8', 'HarperCollins','UK', '1994'),

('9', 'Simon & Schuster','USA', '1990'),

('10', 'Scholastic','USA', '2002');

select \* from Dim\_Publication;

SELECT Book\_Type as 'Book Type', Quantity, Profit as 'Profit per Book Type'

from Fact\_BookManagement

join Dim\_Book on Fact\_BookManagement.Book\_Id =

Dim\_Book.Book\_Id

group by Book\_Type, Quantity, Profit;

SELECT Location\_Country as 'Location- Country', Quantity, Profit 'Profit per Book Location'

from Fact\_BookManagement

join Dim\_Location on Fact\_BookManagement.Location\_Id =

Dim\_Location.Location\_Id

group by Location\_Country, Quantity, Profit;

SELECT Author\_Name as 'Author Name', Quantity, Profit 'Profit per Book Author'

from Fact\_BookManagement

join Dim\_Author on Fact\_BookManagement.Author\_Id =

Dim\_Author.Author\_Id

group by Author\_Name, Quantity, Profit;

SELECT Publication\_Name as

'Publication Name', Quantity, Profit 'Profit per Book Publication'

from Fact\_BookManagement

join Dim\_Publication on Fact\_BookManagement.Publication\_Id

= Dim\_Publication.Publication\_Id group by Publication\_Name, Quantity, Profit;

**4. snowflake:**

create database bank;

use bank;

CREATE TABLE Sub\_Dim\_Address (

Address\_Key INT PRIMARY KEY,

Street\_Address VARCHAR(255),

City VARCHAR(100),

State VARCHAR(100),

Zip\_Code VARCHAR(20),

Country VARCHAR(100)

);

CREATE TABLE Sub\_Dim\_Product\_Term (

Product\_Term\_Key INT PRIMARY KEY,

Product\_Term VARCHAR(20),

Term\_Description VARCHAR(255)

);

CREATE TABLE Dim\_Date (

Date\_Key INT PRIMARY KEY,

Full\_Date DATE,

Month INT,

Quarter INT,

Year INT,

Day\_Of\_Week VARCHAR(50)

);

CREATE TABLE Dim\_Customer (

Customer\_Key INT PRIMARY KEY,

Customer\_ID VARCHAR(50),

Customer\_Name VARCHAR(255),

Date\_Of\_Birth DATE,

Gender VARCHAR(10),

Address\_Key INT,

Phone\_Number VARCHAR(20),

Email VARCHAR(100),

Marital\_Status VARCHAR(50),

Employment\_Status VARCHAR(50),

FOREIGN KEY (Address\_Key) REFERENCES Sub\_Dim\_Address(Address\_Key)

);

CREATE TABLE Dim\_Account (

Account\_Key INT PRIMARY KEY,

Account\_ID VARCHAR(50),

Account\_Type VARCHAR(50),

Account\_Open\_Date DATE,

Account\_Status VARCHAR(50),

Customer\_Key INT,

FOREIGN KEY (Customer\_Key) REFERENCES Dim\_Customer(Customer\_Key)

);

CREATE TABLE Dim\_Branch (

Branch\_Key INT PRIMARY KEY,

Branch\_ID VARCHAR(50),

Branch\_Name VARCHAR(255),

Branch\_Location INT,

Branch\_Manager VARCHAR(255),

Branch\_Contact VARCHAR(50),

FOREIGN KEY (Branch\_Location) REFERENCES Sub\_Dim\_Address(Address\_Key)

);

CREATE TABLE Dim\_Product (

Product\_Key INT PRIMARY KEY,

Product\_ID VARCHAR(50),

Product\_Name VARCHAR(255),

Product\_Type VARCHAR(50),

Interest\_Rate DECIMAL(5,2),

Product\_Term VARCHAR(50)

);

CREATE TABLE Fact\_Transactions (

Transaction\_ID INT PRIMARY KEY,

Date\_Key INT,

Customer\_Key INT,

Account\_Key INT,

Branch\_Key INT,

Product\_Key INT,

Transaction\_Amount DECIMAL(18,2),

Transaction\_Type VARCHAR(50),

Transaction\_Fee DECIMAL(18,2),

FOREIGN KEY (Date\_Key) REFERENCES Dim\_Date(Date\_Key),

FOREIGN KEY (Customer\_Key) REFERENCES Dim\_Customer(Customer\_Key),

FOREIGN KEY (Account\_Key) REFERENCES Dim\_Account(Account\_Key),

FOREIGN KEY (Branch\_Key) REFERENCES Dim\_Branch(Branch\_Key),

FOREIGN KEY (Product\_Key) REFERENCES Dim\_Product(Product\_Key)

);

INSERT INTO Sub\_Dim\_Address (Address\_Key, Street\_Address, City, State, Zip\_Code, Country) VALUES

(1, '123 Elm St', 'Springfield', 'IL', '62701', 'USA'),

(2, '456 Oak St', 'Shelbyville', 'IL', '62702', 'USA');

INSERT INTO Sub\_Dim\_Product\_Term (Product\_Term\_Key, Product\_Term, Term\_Description) VALUES

(1, '5 years', 'Short-term product'),

(2, '10 years', 'Medium-term product');

INSERT INTO Dim\_Date (Date\_Key, Full\_Date, Month, Quarter, Year, Day\_Of\_Week) VALUES

(1, '2024-01-01', 1, 1, 2024, 'Monday'),

(2, '2024-02-01', 2, 1, 2024, 'Thursday');

INSERT INTO Dim\_Customer (Customer\_Key, Customer\_ID, Customer\_Name, Date\_Of\_Birth, Gender, Address\_Key, Phone\_Number, Email, Marital\_Status, Employment\_Status) VALUES

(1, 'CUST001', 'John Doe', '1980-05-15', 'Male', 1, '555-1234', 'johndoe@example.com', 'Married', 'Employed'),

(2, 'CUST002', 'Jane Smith', '1985-09-20', 'Female', 2, '555-5678', 'janesmith@example.com', 'Single', 'Self-employed');

INSERT INTO Dim\_Account (Account\_Key, Account\_ID, Account\_Type, Account\_Open\_Date, Account\_Status, Customer\_Key) VALUES

(1, 'ACC001', 'Savings', '2020-01-01', 'Active', 1),

(2, 'ACC002', 'Checking', '2021-02-01', 'Active', 2);

INSERT INTO Dim\_Branch (Branch\_Key, Branch\_ID, Branch\_Name, Branch\_Location, Branch\_Manager, Branch\_Contact) VALUES

(1, 'BR001', 'Main Branch', 1, 'Alice Johnson', '555-0001'),

(2, 'BR002', 'Downtown Branch', 2, 'Bob Williams', '555-0002');

INSERT INTO Dim\_Product (Product\_Key, Product\_ID, Product\_Name, Product\_Type, Interest\_Rate, Product\_Term) VALUES

(1, 'PRD001', 'Home Loan', 'Loan', 3.5, '5 years'),

(2, 'PRD002', 'Car Loan', 'Loan', 4.0, '10 years');

INSERT INTO Fact\_Transactions (Transaction\_ID, Date\_Key, Customer\_Key, Account\_Key, Branch\_Key, Product\_Key, Transaction\_Amount, Transaction\_Type, Transaction\_Fee) VALUES

(1, 1, 1, 1, 1, 1, 10000.00, 'Debit', 10.00),

(2, 2, 2, 2, 2, 2, 20000.00, 'Credit', 20.00);

SELECT

f.Transaction\_ID,

f.Transaction\_Amount,

f.Transaction\_Type,

f.Transaction\_Fee,

d.Full\_Date,

c.Customer\_Name,

c.Gender,

c.Email,

a.Account\_Type,

b.Branch\_Name,

b.Branch\_Manager,

p.Product\_Name,

p.Product\_Type,

p.Interest\_Rate,

ad.Street\_Address,

ad.City,

ad.State,

ad.Zip\_Code,

pt.Product\_Term

FROM

Fact\_Transactions f

JOIN

Dim\_Date d ON f.Date\_Key = d.Date\_Key

JOIN

Dim\_Customer c ON f.Customer\_Key = c.Customer\_Key

JOIN

Dim\_Account a ON f.Account\_Key = a.Account\_Key

JOIN

Dim\_Branch b ON f.Branch\_Key = b.Branch\_Key

JOIN

Dim\_Product p ON f.Product\_Key = p.Product\_Key

JOIN

Sub\_Dim\_Address ad ON c.Address\_Key = ad.Address\_Key

JOIN

Sub\_Dim\_Product\_Term pt ON p.Product\_Term = pt.Product\_Term;

SELECT \* FROM Sub\_Dim\_Address;

SELECT \* FROM Sub\_Dim\_Product\_Term;

SELECT \* FROM Dim\_Date;

SELECT \* FROM Dim\_Customer;

SELECT \* FROM Dim\_Account;

SELECT \* FROM Dim\_Branch;

SELECT \* FROM Dim\_Product;

**5. galaxy constellation:**

-- Create the database and use it

CREATE DATABASE healthcare;

USE healthcare;

-- Creating Dimension Tables

CREATE TABLE DimPatients (

PatientID INT PRIMARY KEY,

Name VARCHAR(100),

Age INT,

Gender CHAR(1),

Address VARCHAR(255),

PhoneNumber VARCHAR(15)

);

INSERT INTO DimPatients VALUES

(1, 'John Doe', 35, 'M', '123 Main St', '123-456-7890'),

(2, 'Jane Smith', 29, 'F', '456 Oak Ave', '987-654-3210'),

(3, 'Bob Johnson', 50, 'M', '789 Pine Rd', '555-666-7777'),

(4, 'Alice Green', 40, 'F', '101 Maple St', '321-654-0987'),

(5, 'Michael Brown', 60, 'M', '202 Birch St', '222-333-4444'),

(6, 'Nancy Wilson', 45, 'F', '303 Cedar St', '444-555-6666'),

(7, 'Kevin Black', 55, 'M', '404 Spruce St', '777-888-9999');

CREATE TABLE DimDoctors (

DoctorID INT PRIMARY KEY,

DoctorName VARCHAR(100),

Specialization VARCHAR(50),

ContactNumber VARCHAR(15),

Email VARCHAR(100)

);

INSERT INTO DimDoctors VALUES

(1, 'Dr. Alice Brown', 'Cardiology', '111-222-3333', 'alice.brown@hospital.com'),

(2, 'Dr. Chris White', 'Neurology', '444-555-6666', 'chris.white@hospital.com'),

(3, 'Dr. David Green', 'Orthopedics', '555-666-7777', 'david.green@hospital.com'),

(4, 'Dr. Emily Black', 'Pediatrics', '666-777-8888', 'emily.black@hospital.com'),

(5, 'Dr. Frank Blue', 'Oncology', '777-888-9999', 'frank.blue@hospital.com'),

(6, 'Dr. Grace Yellow', 'Dermatology', '888-999-0000', 'grace.yellow@hospital.com'),

(7, 'Dr. Henry Orange', 'Endocrinology', '999-000-1111', 'henry.orange@hospital.com');

CREATE TABLE DimDepartments (

DepartmentID INT PRIMARY KEY,

DepartmentName VARCHAR(100)

);

INSERT INTO DimDepartments VALUES

(1, 'Cardiology'),

(2, 'Neurology'),

(3, 'Orthopedics'),

(4, 'Pediatrics'),

(5, 'Oncology'),

(6, 'Dermatology'),

(7, 'Endocrinology');

CREATE TABLE DimTreatments (

TreatmentCode VARCHAR(10) PRIMARY KEY,

TreatmentName VARCHAR(100),

TreatmentDescription VARCHAR(255)

);

INSERT INTO DimTreatments VALUES

('T001', 'Bypass Surgery', 'Cardiac Bypass Surgery'),

('T002', 'MRI Scan', 'Magnetic Resonance Imaging'),

('T003', 'Knee Replacement', 'Surgical Knee Replacement'),

('T004', 'Vaccination', 'Routine Immunization'),

('T005', 'Chemotherapy', 'Cancer Treatment'),

('T006', 'Skin Biopsy', 'Dermatological Skin Examination'),

('T007', 'Insulin Therapy', 'Diabetes Treatment');

CREATE TABLE DimDiagnosis (

DiagnosisCode VARCHAR(10) PRIMARY KEY,

DiagnosisName VARCHAR(100),

DiagnosisDescription VARCHAR(255)

);

INSERT INTO DimDiagnosis VALUES

('D001', 'Heart Attack', 'Myocardial Infarction'),

('D002', 'Migraine', 'Severe Headache'),

('D003', 'Arthritis', 'Joint Inflammation'),

('D004', 'Flu', 'Influenza Virus'),

('D005', 'Lung Cancer', 'Malignant Lung Tumor'),

('D006', 'Eczema', 'Chronic Skin Condition'),

('D007', 'Diabetes', 'Metabolic Disease');

-- Creating Fact Tables

CREATE TABLE FactPatientVisits (

VisitID INT PRIMARY KEY,

PatientID INT,

DoctorID INT,

DepartmentID INT,

VisitDate DATE,

DiagnosisCode VARCHAR(10),

TreatmentCode VARCHAR(10),

FOREIGN KEY (PatientID) REFERENCES DimPatients(PatientID),

FOREIGN KEY (DoctorID) REFERENCES DimDoctors(DoctorID),

FOREIGN KEY (DepartmentID) REFERENCES DimDepartments(DepartmentID),

FOREIGN KEY (DiagnosisCode) REFERENCES DimDiagnosis(DiagnosisCode),

FOREIGN KEY (TreatmentCode) REFERENCES DimTreatments(TreatmentCode)

);

INSERT INTO FactPatientVisits VALUES

(1, 1, 1, 1, '2024-08-01', 'D001', 'T001'),

(2, 2, 2, 2, '2024-08-03', 'D002', 'T002'),

(3, 3, 3, 3, '2024-08-05', 'D003', 'T003'),

(4, 4, 4, 4, '2024-08-07', 'D004', 'T004'),

(5, 5, 5, 5, '2024-08-09', 'D005', 'T005'),

(6, 6, 6, 6, '2024-08-11', 'D006', 'T006'),

(7, 7, 7, 7, '2024-08-13', 'D007', 'T007');

CREATE TABLE FactBilling (

BillingID INT PRIMARY KEY,

PatientID INT,

VisitID INT,

TotalAmount DECIMAL(10, 2),

PaymentMethod VARCHAR(50),

BillingDate DATE,

FOREIGN KEY (PatientID) REFERENCES DimPatients(PatientID),

FOREIGN KEY (VisitID) REFERENCES FactPatientVisits(VisitID)

);

INSERT INTO FactBilling VALUES

(1, 1, 1, 50000.00, 'Credit Card', '2024-08-02'),

(2, 2, 2, 3000.00, 'Cash', '2024-08-04'),

(3, 3, 3, 45000.00, 'Insurance', '2024-08-06'),

(4, 4, 4, 200.00, 'Debit Card', '2024-08-08'),

(5, 5, 5, 150000.00, 'Bank Transfer', '2024-08-10'),

(6, 6, 6, 2500.00, 'Cash', '2024-08-12'),

(7, 7, 7, 10000.00, 'Insurance', '2024-08-14');

SHOW TABLES;

SELECT \* FROM DimPatients;

SELECT \* FROM DimDoctors;

SELECT \* FROM DimDepartments;

SELECT \* FROM DimTreatments;

SELECT \* FROM DimDiagnosis;

SELECT \* FROM FactPatientVisits;

SELECT \* FROM FactBilling;

**6. apriori:**

import numpy as np

import pandas as pd

from mlxtend.frequent\_patterns import apriori,association\_rules

#from mlxtend.frequent\_patterns import association\_rules

import warnings

warnings.filterwarnings('ignore')

-

data = pd.read\_csv("D:\\SEM V\\DWDM\\LAB\\6\Online Retail.csv", encoding="latin-1")

-

data.head()

-

data.columns

-

data.shape

-

data.isnull().values.any()

-

data.isnull().sum()

-

data['Description'] = data['Description'].str.strip()

data.dropna(axis=0,subset=['InvoiceNo'],inplace=True)

data['InvoiceNo'] = data['InvoiceNo'].astype('str')

data = data[~data['InvoiceNo'].str.contains('C')]

data.Country.unique()

-

basket\_France = (data[data['Country']=="France"]

          .groupby(['InvoiceNo', 'Description'])['Quantity']

          .sum().unstack().reset\_index().fillna(0)

          .set\_index('InvoiceNo'))

-

def hot\_encode(x):

  if(x<=0):

    return 0

  if(x>=1):

    return 1

-

basket\_encoded = basket\_France.applymap(hot\_encode)

basket\_France=basket\_encoded

basket\_France.head()

-

freq\_items = apriori(basket\_France,min\_support=0.1, use\_colnames=True)

rules = association\_rules(freq\_items,metric='lift',min\_threshold=1)

rules= rules.sort\_values(['confidence','lift'], ascending =[False, False])

print(rules.head())

----

**7. k means clustering:**

import pandas as pd

import numpy as np

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler, MinMaxScaler

import matplotlib.pyplot as plt

import seaborn as sns

import warnings

warnings.filterwarnings('ignore')

-

df= pd.read\_csv("D:\\SEM V\\DWDM\\LAB\\7\\Country\_clusters.csv", encoding="latin-1")

-

df

-

df1= pd.get\_dummies(df)

df1

-

df.info()

-

sns.scatterplot(x = df['Latitude'], y=df['Longitude'])

-

numeric\_df=df.select\_dtypes(include=[int,float])

std\_scalar = StandardScaler()

x=numeric\_df

array = std\_scalar.fit\_transform(x)

x = pd.DataFrame(array, columns=x.columns)

x

-

from sklearn.preprocessing import StandardScaler

# Creating an instance of StandardScaler

std\_scalar = StandardScaler()

# Applying the scaler to the 'Latitude' and 'Longitude' columns

array = std\_scalar.fit\_transform(df1[['Latitude', 'Longitude']])

# Creating a DataFrame with the scaled values

x\_scaled = pd.DataFrame(array, columns=['Latitude', 'Longitude'])

# Concatenating the scaled columns back with the rest of the dataset

x = pd.concat([x\_scaled, df1.drop(['Latitude', 'Longitude'], axis=1)], axis=1)

# Display the final DataFrame

x.head()

-

kmean\_model = KMeans(n\_clusters=3)

y\_pred = kmean\_model.fit\_predict(x)

y\_pred

-

x['Traget'] = y\_pred

x

-

plt.scatter(x = df['Latitude'], y= df['Longitude'], c= y\_pred, cmap='rainbow')

---

**8. hierarchical and agglomerative clustering:**

import matplotlib.pyplot as plt

import pandas as pd

%matplotlib inline

import numpy as np

-

customer\_data = pd.read\_csv("D:\\SEM V\\DWDM\\LAB\\8\\shopping-data.csv")

-

customer\_data.shape

-

customer\_data.head()

-

data = customer\_data.iloc[:, 3:5].values

data

-

import scipy.cluster.hierarchy as shc

plt.figure(figsize=(10,7))

plt.title("Customer Dendograms")

dend = shc.dendrogram(shc.linkage(data, method='ward'))

-

from sklearn.cluster import AgglomerativeClustering

cluster = AgglomerativeClustering(n\_clusters=5, metric='euclidean', linkage='ward')

lables\_=cluster.fit\_predict(data)

lables\_

----

**9. naive Bayesian:**

import numpy as np

import pandas as pd

from sklearn.feature\_extraction.text import CountVectorizer, TfidfVectorizer

from sklearn.model\_selection import train\_test\_split

from sklearn.naive\_bayes import GaussianNB

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report

from sklearn.naive\_bayes import GaussianNB, MultinomialNB, BernoulliNB

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score

import warnings

warnings.filterwarnings("ignore")

-

df=pd.read\_excel("D:\\SEM V\\DWDM\\LAB\\weathers.xlsx")

-

df

-

df1=df.dropna()

df1

-

df1["outlook"].replace({"overcast":1, "rainy":2, "sunny":3},inplace=True)

-

df1["temperature"].replace({"mild":0,"hot":1,"cool":2},inplace=True)

-

df1["humidity"].replace({"high":0,"normal":1},inplace=True)

-

df1["play"].replace({"no":0,"yes":1},inplace=True)

-

x = df1.drop('play', axis=1)

y = df1["play"]

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.2, random\_state=11,stratify=y)

-

gnb\_model = GaussianNB()

gnb\_model.fit(x\_train, y\_train)

-

y\_pred= gnb\_model.predict(x\_test)

cnf\_matrix =confusion\_matrix(y\_pred, y\_test)

print("Confusion Matrix:\n", cnf\_matrix)

print("-"\*60)

accuracy = accuracy\_score (y\_pred, y\_test)

print("Accuracy:", accuracy)

print("-"\*60)

clf\_report = classification\_report(y\_pred, y\_test)

print("Classification Report:\n",clf\_report)

-

y\_pred\_train = gnb\_model.predict(x\_train)

cnf\_matrix = confusion\_matrix(y\_pred\_train, y\_train)

print("Confusion Matrix:\n", cnf\_matrix)

print("-"\*60)

accuracy= accuracy\_score (y\_pred\_train, y\_train)

print("Accuracy: ", accuracy)

print("-"\*60)

clf\_report = classification\_report (y\_pred\_train, y\_train)

print("Classification Report: \n",clf\_report)

-

column\_names=x.columns

column\_names

-

outlook\_value={"overcast":1,"rainy":2,"sunny":3}

temperature\_value={"mild": 0, "hot":1,"cool":2}

humidity\_value={"high":0,"normal":1}

-

json\_data={"outlook": outlook\_value,

"temperature":temperature\_value,

"humidity": humidity\_value,

"columns":list (column\_names)}

json\_data

-

import json

with open("Project\_data\_NB.json", "w") as f:

   json.dump(json\_data, f)

-

outlook="rainy"

temperature="mild"

humidity="high"

windy=1.0

-

test\_array = np.zeros(len(column\_names))

test\_array[0] = json\_data['outlook'][outlook]

test\_array[1] = json\_data['temperature'][temperature]

test\_array[2] = json\_data['humidity'][humidity]

test\_array[3] = windy

test\_array

-

play=gnb\_model.predict([test\_array])[0]

if play==1:

 print("yes, Play is happening")

else:

 print("No, Play is not happening")

**10. decision tree:**

import numpy as np

import pandas as pd

from sklearn.tree import DecisionTreeRegressor, plot\_tree

from sklearn.model\_selection import train\_test\_split, GridSearchCV

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score

import seaborn as sns

import matplotlib.pyplot as plt

import warnings

# Ignore warnings for cleaner output

warnings.filterwarnings("ignore")

-

# Load the dataset

df = pd.read\_csv("D:\\SEM V\\DWDM\\LAB\\medical\_insurance.csv")

-

df

-

print(df.info())

-

print(df.isna().sum())

-

df['sex'].replace({"male": 0, "female": 1}, inplace=True)

sex\_value = {"female": 1, "male": 0}

print("Sex Value Encoding:", sex\_value)

-

df['smoker'].replace({"yes": 0, "no": 1}, inplace=True)

smoker\_value = {"no": 1, "yes": 0}

print("Smoker Value Encoding:", smoker\_value)

-

df = pd.get\_dummies(df, columns=['region'])

print(df.head())

-

X = df.drop("charges", axis=1)  # Dropping the target variable 'charges'

y = df["charges"]  # Target variable

-

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

-

dt\_reg = DecisionTreeRegressor()

dt\_reg.fit(X\_train, y\_train)

# Make predictions on the test set

y\_pred = dt\_reg.predict(X\_test)

-

mae = mean\_absolute\_error(y\_test, y\_pred)

mse = mean\_squared\_error(y\_test, y\_pred)

rmse = np.sqrt(mse)

r2 = r2\_score(y\_test, y\_pred)

-

print(f"Mean Absolute Error (MAE): {mae}")

print(f"Mean Squared Error (MSE): {mse}")

print(f"Root Mean Squared Error (RMSE): {rmse}")

print(f"R² Score: {r2}")

-

plt.figure(figsize=(45, 40))

plot\_tree(dt\_reg, filled=True, feature\_names=X.columns)

plt.title("Decision Tree")

plt.show()

-

y\_pred\_train = dt\_reg.predict(X\_train)

mse\_train = mean\_squared\_error(y\_train, y\_pred\_train)

print("Mean Squared Error (Train) --->", mse\_train)

mae\_train = mean\_absolute\_error(y\_train, y\_pred\_train)

print("Mean Absolute Error (Train) --->", mae\_train)

rmse\_train = np.sqrt(mse\_train)

print("Root Mean Square Error (Train) --->", rmse\_train)

r2\_train = r2\_score(y\_train, y\_pred\_train)

print("R² Score (Train) --->", r2\_train)

# Display feature column names

column\_names = X.columns

print(column\_names)

------