Er. PERUMAL MANIMEKALAI
COLLEGE OF ENGINEERING
ACCREDITED BY NBA & NAAC WITH 'A' GRADE
Koneripalli, HOSUR - 635 117.

YEAR/SEM/DEPT : III/VI/ CSE

ACADEMIC YEAR : 2023-2024 (EVEN SEM)

SUBJECT CODE : CCS341

SUBJECT NAME : DATAWAREHOUSING LABORATORY

FACULTY INCHARGE : Dr.N.Shunmuga Karpagam ASP/CSE

LAB MANUAL

CCS341 DATAWAREHOUSING LABORATORY

FACULTY INCHARGE HOD PRINCIPAL

PMC TECH
INSPIRE TO INNOVATE

Er. PERUMAL MANIMEKALAI COLLEGE OF ENGINEERING

Accredited by NAAC ('A' Grade) & NBA (B.E. - CSE | ECE | EEE | MECH & B.TECH. - IT)
AN AUTONOMOUS INSTITUTION

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(THIRD YEAR VI SEM)

LABORATORY SUBJECT: DATA WAREHOUSING CCS341

Faculty: Dr.N.Shunmuga Karpagam ASP/CSE

Vision and Mission of the Institute

Vision:

PMC TECH strives to achieve excellence in technical Education through innovative teaching, learning and applied Multidisciplinary research with professional and ethical practices.

Mission:

PMC TECH will Endeavour

- To become the state of art teaching and learning center for Engineering and Technology, Research and Management Studies
- To have world class infrastructure for providing quality education and research towards creativity, self discipline and ethical values
- To associate with industry, R&D and business organizations and to have connectivity with the society
- To create knowledge, based professionals to enrich their quality of life by empowering self and family.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Vision and Mission of the Department

Vision:

Envisions Computer Science Engineering Graduates with profound skills and knowledge embedded with innovative thinking for entrepreneurial, career and societal needs.

Mission:

M1: To imbibe high quality knowledge, skills and practices of computer science engineering through quality infrastructure and resources.

M2: To endow the students to solve simple solutions to complex engineering problems through software.

M3: To provide strong software development skills for lifelong learning, to meet the challenges of industry and society.

Program Educational Objectives

PEO1: Have good knowledge in fast evolving computer science engineering tools and systems, towards employability, higher studies and research.

PEO2: Develop high end software and firmware systems though technical, problem solving and soft skills with ethical standards.

PEO3: Believe in self, nurture to be a team member with leadership qualities, engineer products and systems with sustainable development, have lifelong learning attitudes.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PROGRAM SPECIFIC OUTCOMES (PSOs):

PSO1: Apply standard practices in software development using open source programming environments to deliver a high quality and cost effective products and solutions.

PSO2: Design, analyze and develop systems in the areas of networking, software engineering, artificial intelligence, machine learning, Internet of Things and Cloud computing.

PSO3: Apply programming skill, Agile and Extreme Programming to solve the industrial and societal problems.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Program Outcomes (POs)

Engineering Graduates will be able to:

1. **Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem Analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design / Development of Solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct Investigations of Complex Problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern Tool Usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The Engineer and Society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and Sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and Team Work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project Management and Finance:** Demonstrate knowledge and understanding of the Engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life - Long Learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PMC TECH
INSPIRE TO INNOVATE

Er. PERUMAL MANIMEKALAI COLLEGE OF ENGINEERING
Accredited by NAAC ('A' Grade) & NBA (B.E. - CSE | ECE | EEE | MECH & B.TECH. - IT)
AN AUTONOMOUS INSTITUTION

DEPARTMENT OF INFORMATION TECHNOLOGY

DATA WAREHOUSING (CCS341)

PRACTICAL EXERCISES:

1. Data exploration and integration with WEKA
2. Apply weka tool for data validation
3. Plan the architecture for real time application
4. Write the query for schema definition
5. Design data ware house for real time applications
6. Analyse the dimensional Modeling
7. Case study using OLAP
8. Case study using OTLP
9. Implementation of warehouse testing.

CCS341 DATA WAREHOUSING LAB

COURSE OBJECTIVES:

1	To know the details of data warehouse Architecture
2	To understand the OLAP Technology
3	To understand the partitioning strategy
4	To differentiate various schema
5	To understand the roles of process manager & system manager

COURSE OUTCOMES:

Year of study		2023-24
Course code & name		
Cos	K L	Course Outcome Statements
C314.1	K2	Students will be able to describe data warehouse architecture for various Problems
C314.2	K2	Students will be able to apply the OLAP Technology
C314.3	K2	Students will be able to analyze the partitioning strategy
C314.4	K3	Students will be able to apply the data warehouse schemas
C314.5	K2	Students will be able to frame roles of process manager & system manager

COURSE OUTCOMES VS POs MAPPING (DETAILED; HIGH:3; MEDIUM:2; LOW:1**):**

Course code & name:		CCS341 DATA WAREHOUSING LAB													
	CO Vs PO MAPPING														
CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C314.1	3	2	3	3	1	-	-	-	1	1	2	2	1	2	2
C314.2	2	3	2	3	1	-	-	-	1	1	1	2	2	2	2
C314.3	3	1	2	1	1	-	-	-	2	1	1	2	1	2	2
C314.4	3	2	3	1	1	-	-	-	2	1	2	1	2	2	2
C314.5	2	1	1	2	2	-	-	-	2	1	2	2	2	1	2
AVG	3	2	2	2	1	-	-	-	2	1	2	2	2	2	2

CCS341-DATAWAREHOUSING LABORATORY OUTCOMES MAPPING

Ex.NO.	TITLE	PO's	CO
1	Data exploration and integration with WEKA	PO1,PO2,PO3, PO4, PO5, ,PO9,PO10, PO11,PO12,PSO1,PSO2,PSO3	CO1
2	Apply weka tool for data validation	PO1,PO2,PO3, PO4, PO5, ,PO9,PO10, PO11,PO12,PSO1,PSO2,PSO3	CO1,C02
3	Plan the architecture for real time application	PO1,PO2,PO3, PO4, PO5, ,PO9,PO10, PO11,PO12,PSO1,PSO2,PSO3	CO3,CO4, CO5
4	Write the query for schema definition	PO1,PO2,PO3, PO4, PO5, ,PO9,PO10, PO11,PO12,PSO1,PSO2,PSO3	CO4
5	Design data ware house for real time applications	PO1,PO2,PO3, PO4, PO5, ,PO9,PO10, PO11,PO12,PSO1,PSO2,PSO3	CO5
6	Analyse the dimensional Modeling	PO1,PO2,PO3, PO4, PO5, ,PO9,PO10, PO11,PO12,PSO1,PSO2,PSO3	CO4
7	Case study using OLAP	PO1,PO2,PO3, PO4, PO5, ,PO9,PO10, PO11,PO12,PSO1,PSO2,PSO3	CO2,CO3, CO4,CO5
8	Case study using OTLP	PO1,PO2,PO3, PO4, PO5, ,PO9,PO10, PO11,PO12,PSO1,PSO2,PSO3	CO2,CO3, CO4,CO5
9	Implementation of warehouse testing.	PO1,PO2,PO3, PO4, PO5, ,PO9,PO10, PO11,PO12,PSO1,PSO2,PSO3	CO5

CONTENT BEYOND THE SYLLABUS

1	Implementation of data warehouse using pentaho
2	Run J48 and Naïve Bayes classifiers on the following datasets and determine the accuracy: 1.vehicle.arff 2.kr-vs-kp.arff 3.glass.arff 4.wave-form-5000.arff On which datasets does the Naïve Bayes perform better?
3	Load the „weather.arff" dataset in Weka and run the ID3 classification algorithm. What problem do you have and what is the solution?
4	List the attributes do you think might be crucial in making the bank assessment

RUBRICS FOR LABORATORY

Name of the Lab: **CCS341-DATAWAREHOUSING LABORATORY**

Department : **B.E(CSE)**

Year/Sem: **III/VI**

Attributes	Descriptors	Scores
Team Work (10)	Planner	8-10
	Initiator	5-7
	Follower	1-4
	NON-performer	0
Aim/Algorithm (10)	Good Level (Information, Tabulation, Drawings)	9-10
	Average	5-8
	Low	0-4
Programs (10)	Exact	10
	Within Range	1-9
	Out of range	0
Results (10)	Exact	10
	Within Range	1-9
	Out of range	0
Viva-Voce (10)	Good Level (Answering 80% to 100% Questions)	8-10
	Average (Answering 40% to 79% Questions)	4-7
	Low (Answering 0% to 39% Questions)	0-3

1. Data exploration and integration with WEKA

AIM:

To Explore Data and Integrate with WEKA

ALGORTIHM AND EXPLORES:

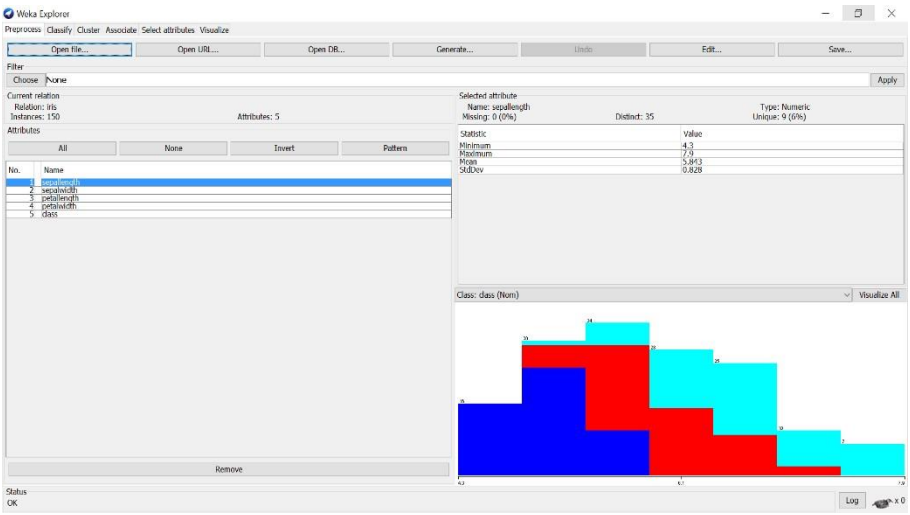
1.Download and install Weka. You can find it here: http://www.cs.waikato.ac.nz/mn/weka/downloading.html

2.Open the weka tool and select the explorer option.

3.New window will be opened which consists of different options (Preprocess, Association etc.)

3. In the preprocess, click the ―open file‖ option.

4.Go to C:\Program Files\Weka-3-6\data for finding different existing. arff datasets. Click on any dataset for loading the data then the data will be displayed as shown below



). Load each dataset and observe the following:

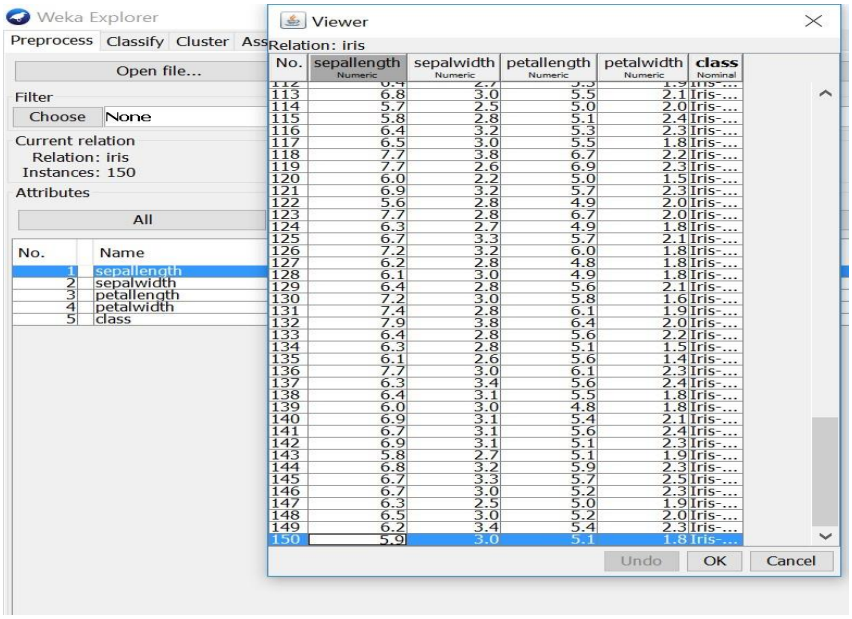
Here we have taken **IRIS.arff** dataset as sample for observing all the below things.

i. List the attribute names and they types

There are 5 attributes& its datatype present in the above loaded dataset (IRIS.arff) sepallength – Numeric sepalwidth – Numeric petallength – Numeric petallength – Numeric Class – Nominal

ii. Number of records in each dataset

There are total <u>150</u> records (Instances) in dataset (IRIS.arff).

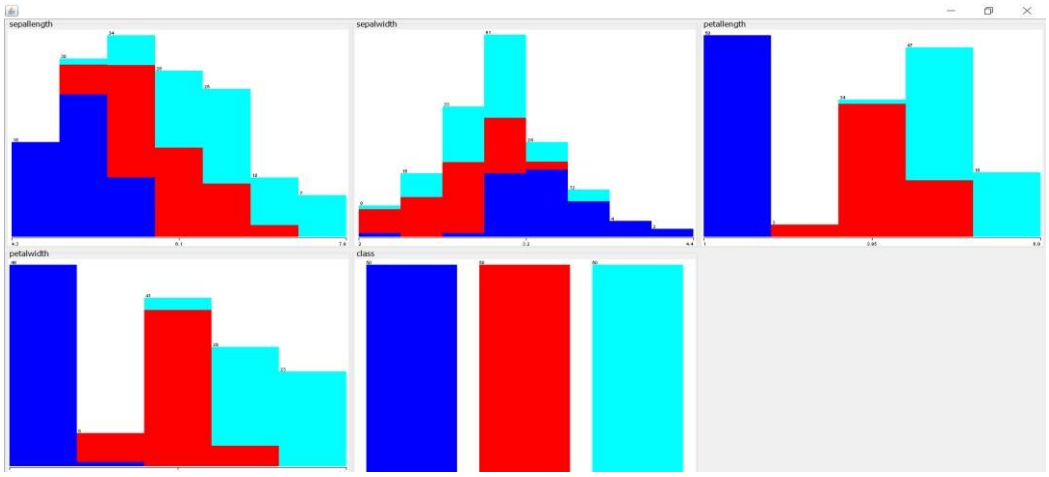


iii. Identify the class attribute (if any)

There is one class attribute which consists of 3 labels. They are:

1. Iris-setosa
2. Iris-versicolor
3. Iris-virginica

iv. Plot Histogram

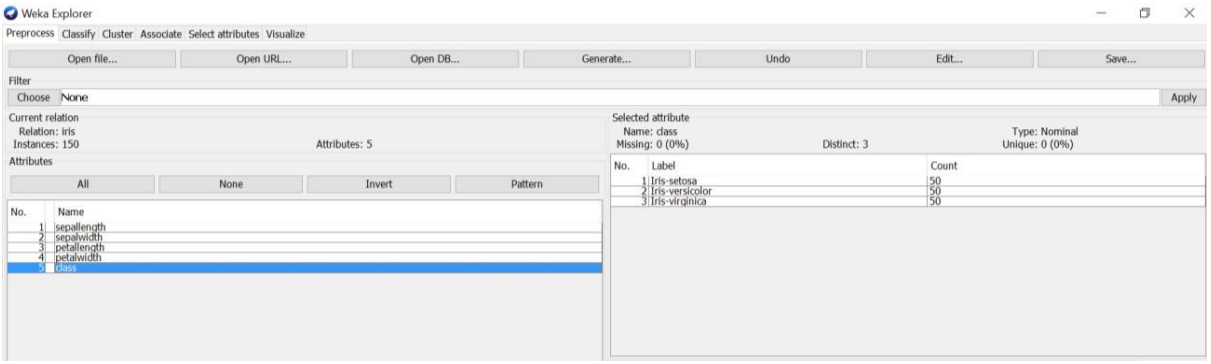


v. Determine the number of records for each class.

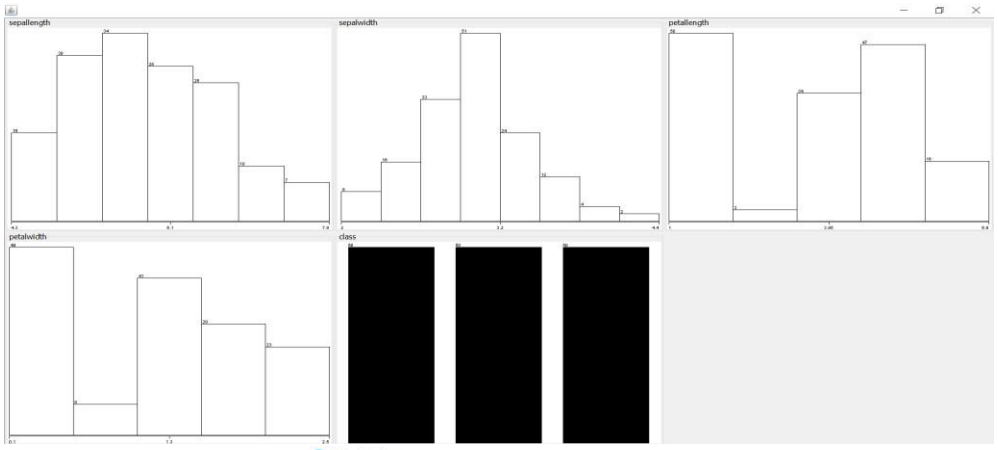
There is one class attribute (150 records) which consists of 3 labels. They are shown below 1. Iris-setosa - 50 records

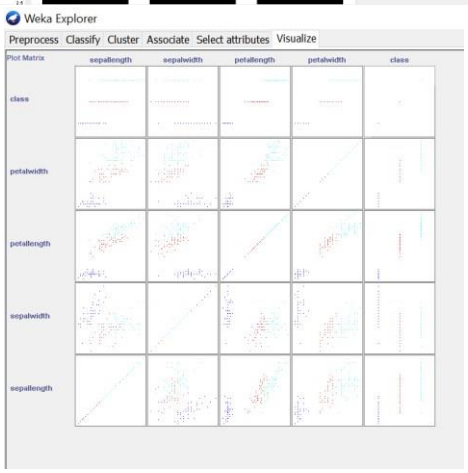
2. Iris-versicolor – 50 records

3. Iris-virginica – 50 records



vi. Visualize the data in various dimensions





RESULT:

Thus the data exploration and integration with WEKA executed successfully.

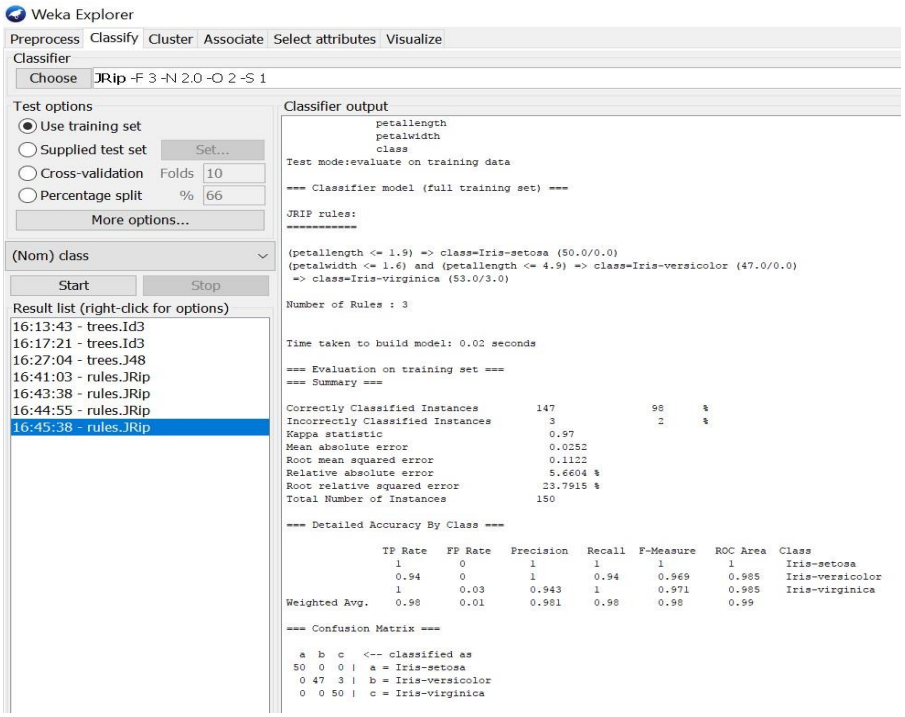
2. Apply WEKA tool for Data Validation

AIM:

To Apply WEKA tool for Data Validation

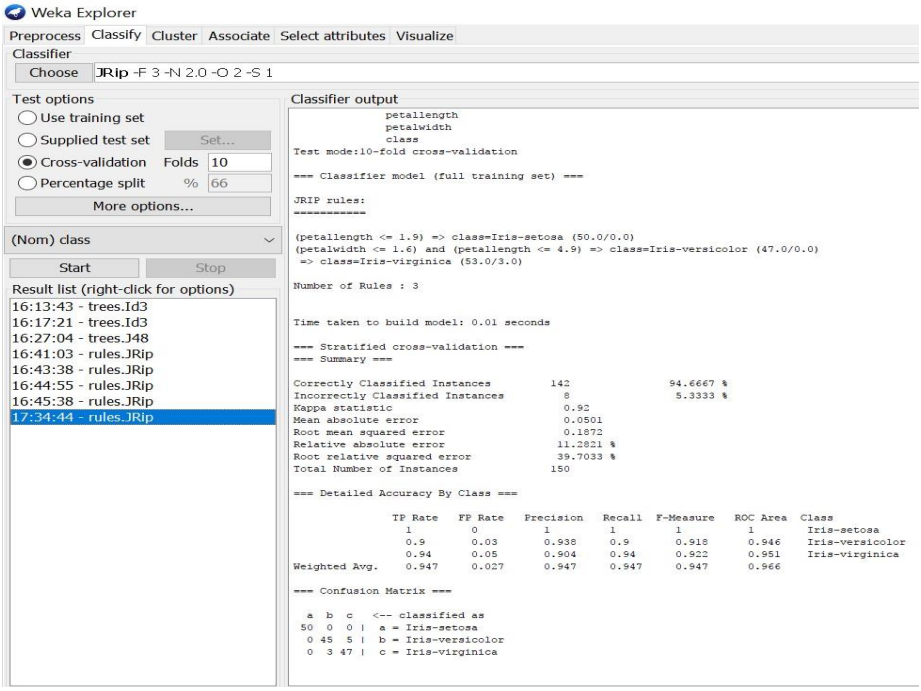
Steps and Apply:

1. Load the dataset (Iris-2D. arff) into weka tool
2. Go to classify option & in left-hand navigation bar we can see differentclassification algorithms under rules section.
3. In which we selected JRip (If-then) algorithm & click on start option with —use training set‖ test option enabled.
4. Then we will get detailed accuracy by class consists ofF-measure, TP rate, FP rate, Precision, Recall values& Confusion Matrix as represented below.



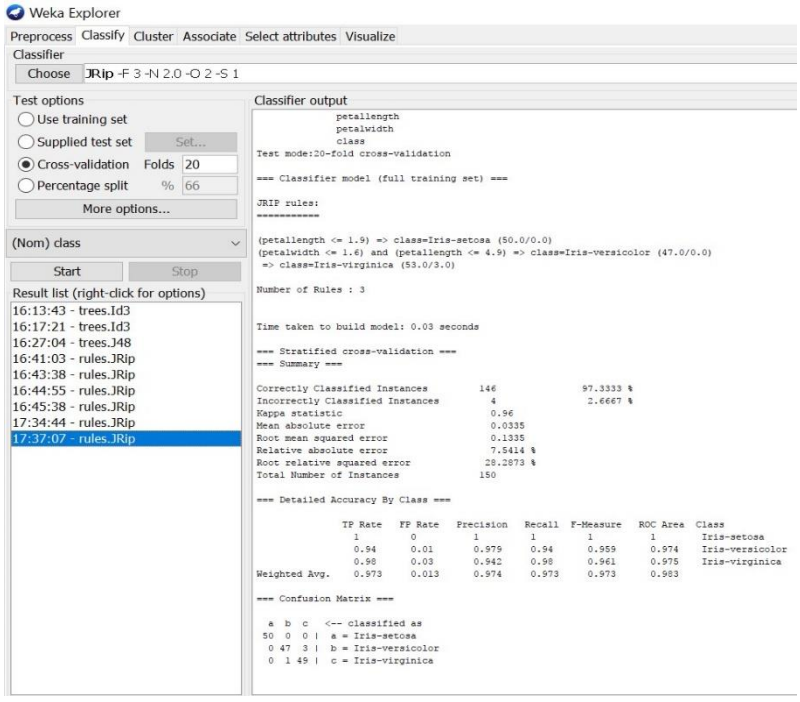
Using Cross-Validation Strategy with 10 folds:

Here, we enabled cross-validation test option with 10 folds & clicked start button as represented below.



Using Cross-Validation Strategy with 20 folds:

Here, we enabled cross-validation test option with 20 folds & clicked start button as represented below.



If we see the above results of cross validation with 10 folds & 20 folds. As per our observation the error rate is lesser with 20 folds got 97.3% correctness when compared to 10 folds got 94.6% correctness.

RESULT: Thus the WEKA tool for Data Validation done Successfully.

3.Plan the architecture for real time application

Aim:

To plan the architecture for a real-time application using Weka, you need to consider several factors. Weka is a popular machine learning library that provides various algorithms for data mining and predictive modelling.

Here are the steps to plan the architecture:

1. **Define the problem**: Clearly understand the problem you are trying to solve with your real-time application. Identify the specific tasks and goals you want to achieve using Weka.
2. **Data collection and preprocessing**: Determine the data sources and collect the required data for your application. Preprocess the data to clean, transform, and prepare it for analysis using Weka. This may involve tasks like data cleaning, feature selection, normalization, and handling missing values.
3. **Choose the appropriate Weka algorithms**: Weka offers a wide range of machine learning algorithms. Select the algorithms that are suitable for your problem and data. Consider factors like the type of data (classification, regression, clustering), the size of the dataset, and the computational requirements.
4. **Real-time data streaming**: If your application requires real-time data processing, you need to set up a mechanism to stream the data continuously. This can be done using technologies like Apache Kafka, Apache Flink, or Apache Storm. Ensure that the data streaming infrastructure is integrated with Weka for seamless processing.
5. **Model training and evaluation**: Train the selected Weka algorithms on your training dataset. Evaluate the performance of the models using appropriate evaluation metrics like accuracy, precision, recall, or F1-score. Fine-tune the models if necessary.
6. **Integration and deployment**: Integrate the trained models into your real-time application. This may involve developing APIs or microservices to expose the models' functionality. Ensure that the application can handle real-time requests and provide predictions or insights in a timely manner.
7. **Monitoring and maintenance**: Set up monitoring mechanisms to track the performance of your real-time application. Monitor the accuracy and performance of the models over time. Update the models periodically to adapt to changing data patterns or to improve performance.

Remember to document your architecture design and implementation decisions for future reference. Regularly review and update your architecture as your application evolves and new requirements arise.

RESULT: Thus architecture for real time applications was Planned.

4.Write the query for schema definition

AIM:

To Write the query for schema definition

ALGORITHM:

1. Create a new database
2. Switch to the newly created database
3. Define the schema for each table
4. Define relationships between tables (if needed)
5. Execute the schema definition queries

PROGRAM:

```
CREATE TABLE books (book_id INT,title VARCHAR(255),author
VARCHAR(100),publication_year INT,isbn VARCHAR(20),PRIMARY KEY (book_id));
CREATE TABLE members (member_id INT,name VARCHAR(100),email
VARCHAR(255),phone_number VARCHAR(20),address VARCHAR(255),PRIMARY KEY
(member_id));
CREATE TABLE checkouts (checkout_id INT,book_id INT, member_id INT,checkout_date
DATE,return_date DATE);

DESC books;
DESC members;
DESC checkouts;


INSERT INTO books (book_id, title, author, publication_year, isbn)
VALUES (1, 'To Kill a Mockingbird', 'Harper Lee', 1960, '9780061120084');


INSERT INTO books VALUES (2, '1984', 'George Orwell', 1949, '9780451524935');
INSERT INTO books VALUES (3, 'Pride and Prejudice', 'Jane Austen', 1813,
'9780141439518');
INSERT INTO books VALUES (4, 'The Great Gatsby', 'F. Scott Fitzgerald', 1925,
'9780743273565');


INSERT INTO members (member_id, name, email, phone_number, address)
```

```
VALUES (1, 'John Smith', 'john@example.com', '+1 (555) 123-4567', '123 Main Street, Anytown, USA');

INSERT INTO members (member_id, name, email, phone_number, address)
VALUES (2, 'Alice Johnson', 'alice@example.com', '+1 (555) 987-6543', '456 Elm Street, Springfield, USA');

INSERT INTO members (member_id, name, email, phone_number, address)
VALUES (3, 'Emily Davis', 'emily@example.com', '+1 (555) 789-0123', '789 Oak Avenue, Metropolis, USA');

INSERT INTO members (member_id, name, email, phone_number, address)
VALUES (4, 'Michael Brown', 'michael@example.com', '+1 (555) 321-8765', '987 Pine Road, Gotham City, USA');



INSERT INTO checkouts (checkout_id, book_id, member_id, checkout_date, return_date)
VALUES (1, 1, 1, '2024-02-07', '2024-03-07');

INSERT INTO checkouts (checkout_id, book_id, member_id, checkout_date, return_date)
VALUES (2, 2, 2, '2024-02-08', '2024-03-08');

INSERT INTO checkouts (checkout_id, book_id, member_id, checkout_date, return_date)
VALUES (3, 3, 3, '2024-02-09', '2024-03-09');

INSERT INTO checkouts (checkout_id, book_id, member_id, checkout_date, return_date)
VALUES (4, 4, 4, '2024-02-10', '2024-03-10');



SELECT * FROM books;
SELECT * FROM members;
SELECT * FROM checkouts;
```

OUTPUT:

Database 'library' created.

Database changed to 'library'.

Table 'books' created successfully.

Table 'members' created successfully.

Table 'checkouts' created successfully

```
SQL> DESC books;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BOOK_ID                                            NUMBER(38)
 TITLE                                              VARCHAR2(255)
 AUTHOR                                             VARCHAR2(100)
 PUBLICATION_YEAR                                   NUMBER(38)
 ISBN                                               VARCHAR2(20)

SQL> DESC members;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 MEMBER_ID                                          NUMBER(38)
 NAME                                               VARCHAR2(100)
 EMAIL                                              VARCHAR2(255)
 PHONE_NUMBER                                       VARCHAR2(20)
 ADDRESS                                            VARCHAR2(255)

SQL> DESC checkouts;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CHECKOUT_ID                                        NUMBER(38)
 BOOK_ID                                            NUMBER(38)
 MEMBER_ID                                          NUMBER(38)
 CHECKOUT_DATE                                      DATE
 RETURN_DATE                                        DATE

SQL>
```

RESULT:

Thus Schema Definition was written and executed Successfully.

5. Design data ware house for real time applications

AIM:

To Design data ware house for real time applications

ALGORITHM AND PROGRAM:

```
import sqlite3

con = sqlite3.connect("tutorial.db")

cur = con.cursor()

cur.execute("CREATE TABLE movie(title, year, score)")


res = cur.execute("SELECT name FROM sqlite_master")

res.fetchone()



import sqlite3

conn = sqlite3.connect('data_warehouse.db')


cursor = conn.cursor()cursor.execute('''CREATE TABLE IF NOT EXISTS orders
(order_id INTEGER PRIMARY KEY,customer_id INTEGER,order_date
DATE,total_amount FLOAT)''')

cursor.execute('''CREATE TABLE IF NOT EXISTS customers (customer_id
INTEGER PRIMARY KEY,name TEXT,email TEXT)''')


conn.commit()

conn.close()
```

```
print("Data warehouse schema created successfully.")


import sqlite3


# Connect to the SQLite database

conn = sqlite3.connect('data_warehouse.db')

cursor = conn.cursor()


# Insert data into the 'customers' table

cursor.execute("INSERT INTO customers (customer_id, name, email) VALUES
(1, 'John Doe', 'john@example.com')")

cursor.execute("INSERT INTO customers (customer_id, name, email) VALUES
(2, 'Jane Smith', 'jane@example.com')")

cursor.execute("INSERT INTO customers (customer_id, name, email) VALUES
(3, 'Alice Johnson', 'alice@example.com')")


# Insert data into the 'orders' table

cursor.execute("INSERT INTO orders (order_id, customer_id, order_date,
total_amount) VALUES (1, 1, '2024-02-07', 100.50)")

cursor.execute("INSERT INTO orders (order_id, customer_id, order_date,
total_amount) VALUES (2, 2, '2024-02-08', 200.75)")

cursor.execute("INSERT INTO orders (order_id, customer_id, order_date,
total_amount) VALUES (3, 3, '2024-02-09', 150.25)")


# Commit the transaction and close the connection

conn.commit()

conn.close()
```

```
print("Data inserted successfully.")



import sqlite3

# Connect to the SQLite database
conn = sqlite3.connect('data_warehouse.db')
cursor = conn.cursor()

# Retrieve data from the 'customers' table
cursor.execute("SELECT * FROM customers")
customers_data = cursor.fetchall()

# Display the data from the 'customers' table
print("Customers Table:")
for row in customers_data:
    print(row)
# Retrieve data from the 'orders' table
cursor.execute("SELECT * FROM orders")
orders_data = cursor.fetchall()

# Display the data from the 'orders' table
print("\nOrders Table:")
for row in orders_data:
    print(row)
conn.close()
```

OUTPUT:

```
Customers Table:
(1, 'John Doe', 'john@example.com')
(2, 'Jane Smith', 'jane@example.com')
(3, 'Alice Johnson', 'alice@example.com')

Orders Table:
(1, 1, '2024-02-07', 100.5)
(2, 2, '2024-02-08', 200.75)
(3, 3, '2024-02-09', 150.25)
```

RESULT:

Thus Data Warehouse for real time application Designed.

6.Analyse the dimensional Modeling

AIM:

To Analyse the dimensional Modeling

ALGORITHM:

1. Identify the business process
2. Identify dimensional and facts
3. Design the dimensional model
4. Define relationships
5. Optimize for query performance

PROGRAM:

1. *Sales Fact Table:*

sql

```
CREATE TABLE SalesFact (
   SaleID INT PRIMARY KEY,
   DateID INT,
   ProductID INT,
   QuantitySold INT,
   AmountSold DECIMAL(10, 2)
);
```

2. *Date Dimension:*

sql

```
CREATE TABLE DateDim (
   DateID INT PRIMARY KEY,
   CalendarDate DATE,
   Day INT,
```

```
    Month INT,
    Year INT
);

-- Populate Date Dimension (sample data)
INSERT INTO DateDim (DateID, CalendarDate, Day, Month, Year)
VALUES
    (1, '2024-01-01', 1, 1, 2024),
    (2, '2024-01-02', 2, 1, 2024),
    -- Add more dates as needed
    ;
```

3. *Product Dimension:*

```
sql
CREATE TABLE ProductDim (
    ProductID INT PRIMARY KEY,
    ProductName VARCHAR(255),
    Category VARCHAR(50),
    -- Additional attributes as needed
);

-- Populate Product Dimension (sample data)
INSERT INTO ProductDim (ProductID, ProductName, Category)
VALUES
    (101, 'Product A', 'Electronics'),
```

```
    (102, 'Product B', 'Clothing'),
    -- Add more products as needed
    ;
```

4. *Query to retrieve sales with date and product details:*

```
sql
SELECT
    s.SaleID,
    d.CalendarDate,
    p.ProductName,
    s.QuantitySold,
    s.AmountSold
FROM
    SalesFact s
JOIN DateDim d ON s.DateID = d.DateID
JOIN ProductDim p ON s.ProductID = p.ProductID;
```

This query retrieves sales information along with corresponding date and product details, leveraging the dimensional model.

OUTPUT:

SaleID	CalendarDate	ProductName	QuantitySold	AmountSold
1	2024-01-01	Product A	10	100.00
2	2024-01-02	Product B	5	50.00
3	2024-01-02	Product A	8	80.00

RESULT:

Thus the dimensional modelling Analysed Successfully.

7. Case study using OLAP

AIM:

To study case using OLAP

Introduction:

In this case study, we will explore how Online Analytical Processing (OLAP) technology was implemented in a retail data warehousing environment to improve data analysis capabilities and support decision-making processes. The case study will focus on a fictional retail company, XYZ Retail, and the challenges they faced in managing and analyzing their vast amounts of transactional data.

Background:

XYZ Retail is a large chain of stores with locations across the country. The company has been experiencing rapid growth in recent years, leading to an increase in the volume of data generated from sales transactions, inventory management, customer interactions, and other operational activities. The existing data management system was struggling to keep up with the demand for timely and accurate data analysis, hindering the company's ability to make informed business decisions.

Challenges:

1. Lack of real-time data analysis: The existing data warehouse system was unable to provide real-time insights into sales trends, inventory levels, and customer preferences.

2. Limited scalability: The data warehouse infrastructure was reaching its limits in terms of storage capacity and processing power, making it difficult to handle the growing volume of data.

3. Complex data relationships: The data stored in the warehouse was highly normalized, making it challenging to perform complex queries and analyze data across multiple dimensions.

Solution:

To address these challenges, XYZ Retail decided to implement an OLAP solution as part of their data warehousing strategy. OLAP technology allows for multidimensional analysis of data, enabling users to easily slice and dice information across various dimensions such as time, product categories, geographic regions, and customer segments.

Implementation:

1. Data modeling: The data warehouse was redesigned using a star schema model, which simplifies data relationships and facilitates OLAP cube creation.

2. OLAP cube creation: OLAP cubes were created to store pre-aggregated data for faster query performance. The cubes were designed to support various dimensions and measures relevant to the retail business.

3. Reporting and analysis: Business users were trained on how to use OLAP tools to create ad-hoc reports, perform trend analysis, and drill down into detailed data.

Results:

1. Improved data analysis: With OLAP technology in place, XYZ Retail was able to perform complex analyses on sales data, identify trends, and make informed decisions based on real-time insights.

2. Faster query performance: OLAP cubes enabled faster query performance compared to traditional relational databases, allowing users to retrieve data more efficiently.

3. Enhanced decision-making: The ability to analyze data across multiple dimensions helped XYZ Retail gain a deeper understanding of their business operations and customer behavior, leading to more strategic decision-making.

Conclusion:

By leveraging OLAP technology in their data warehousing environment, XYZ Retail was able to overcome the challenges of managing and analyzing vast amounts of data. The implementation of OLAP not only improved data analysis capabilities but also empowered business users to make informed decisions based on real-time insights. This case study demonstrates the value of OLAP in enhancing data analysis and decision-making processes in a retail environment.

RESULT:

Thus case study using OLAP done successfully.

8. Case study using OTLP

AIM:

To study case using OTLP

Introduction:

This case study explores the implementation of the Operational Data Layer Pattern (OTLP) in a data warehousing environment to improve data integration, processing, and analytics capabilities. The case study focuses on a fictional company, Tech Solutions Inc., and how they leveraged OTLP to enhance their data warehousing operations.

Background:

Tech Solutions Inc. is a technology consulting firm that provides IT solutions to various clients. The company collects a vast amount of data from different sources, including customer interactions, sales transactions, and operational activities. The existing data warehouse infrastructure was struggling to handle the growing volume of data and provide real-time insights for decision-making.

Challenges:

1. Data silos: Data from different sources were stored in separate silos, making it difficult to integrate and analyze data effectively.

2. Real-time data processing: The existing data warehouse was not capable of processing real-time data streams, leading to delays in data analysis and decision-making.

3. Scalability: The data warehouse infrastructure was reaching its limits in terms of storage capacity and processing power, hindering the company's ability to scale with the growing data volume.

Solution:

To address these challenges, Tech Solutions Inc. decided to implement the OTLP pattern in their data warehousing environment. OTLP combines elements of both Operational Data Store (ODS) and Traditional Data Warehouse (TDW) architectures to enable real-time data processing, data integration, and analytical capabilities.

Implementation:

1. Data integration: Tech Solutions Inc. integrated data from various sources into the operational data layer, where data transformations and cleansing processes were applied.

2. Real-time processing: The OTLP architecture allowed for real-time data processing, enabling the company to analyze streaming data and generate insights in near real-time.

3. Analytics and reporting: Business users were provided with self-service analytics tools to create ad-hoc reports, perform trend analysis, and gain actionable insights from the integrated data.

Results:

1. Improved data integration: The OTLP architecture facilitated seamless integration of data from multiple sources, breaking down data silos and enabling a unified view of the company's operations.

2. Real-time analytics: With OTLP in place, Tech Solutions Inc. was able to analyze streaming data in real-time, allowing for faster decision-making and response to market trends.

3. Scalability: The OTLP architecture provided scalability to handle the growing volume of data, ensuring that the company's data warehousing operations could support future growth.

Conclusion:

By implementing the Operational Data Layer Pattern (OTLP) in their data warehousing environment, Tech Solutions Inc. was able to overcome the challenges of data silos, real-time data processing, and scalability. The adoption of OTLP not only improved data integration and analytics capabilities but also empowered business users to make informed decisions based on real-time insights. This case study highlights the benefits of leveraging OTLP in enhancing data warehousing operations for improved business outcomes.

RESULT:

Thus case study using OTLP done successfully.

9. Implementation of warehouse testing.

AIM:

To implement warehouse testing

Steps with program:

1. Install necessary libraries:

pip install pytest pandas

2. Create a Python script for data transformation and loading:

```
# data_transformation.py
import pandas as pd
def transform_data(input_data):
# Perform data transformation logic here
transformed_data = input_data.apply(lambda x: x * 2)
return transformed_data

def load_data(transformed_data):
 # Load transformed data into the operational data layer
 transformed_data.to_csv('transformed_data.csv', index=False)
```

3. Create test cases using pytest:

```
# test_data_integration.py
```

```
import pandas as pd
import data_transformation

def test_transform_data():
    input_data = pd.DataFrame({'A': [1, 2, 3], 'B': [4, 5, 6]})
    expected_output = pd.DataFrame({'A': [2, 4, 6], 'B': [8, 10, 12]})
    transformed_data = data_transformation.transform_data(input_data)
    assert transformed_data.equals(expected_output)

def test_load_data():
    input_data = pd.DataFrame({'A': [1, 2, 3], 'B': [4, 5, 6]})
    data_transformation.load_data(input_data)
    loaded_data = pd.read_csv('transformed_data.csv')
    assert input_data.equals(loaded_data)
```

4. Run the tests using pytest:

```
pytest test_data_integration.py
```

5. Analyze the test results to ensure that the data transformation and loading processes are functioning correctly in the operational data layer.

By implementing automated tests for data integration processes in the data warehousing environment, you can ensure the accuracy and reliability of the data transformation and loading operations. This approach helps in identifying any issues or discrepancies early on in the development cycle, leading to a more robust and efficient data warehousing system.

OUTPUT:

```
============================
test session starts
============================
platform linux -- Python
3.8.10, pytest-6.2.4,
pluggy-0.13.1
rootdir: /path/to/your/project
collected 2 items

test_data_integration.py ..
[100%]

============================
2 passed in 0.12s
============================
```

RESULT:

Thus implementation of warehouse testing done successfully.