Practical No 1:

AIM:

CREATE A DATABASE WITH THE FOLLOWING TABLES AND ADD ATLEAST 5 RECORDS.

DESCRIPTION:

We need to create a database in MySQL with tables and add 5 records to a table.

PROCEDURE:

Steps :

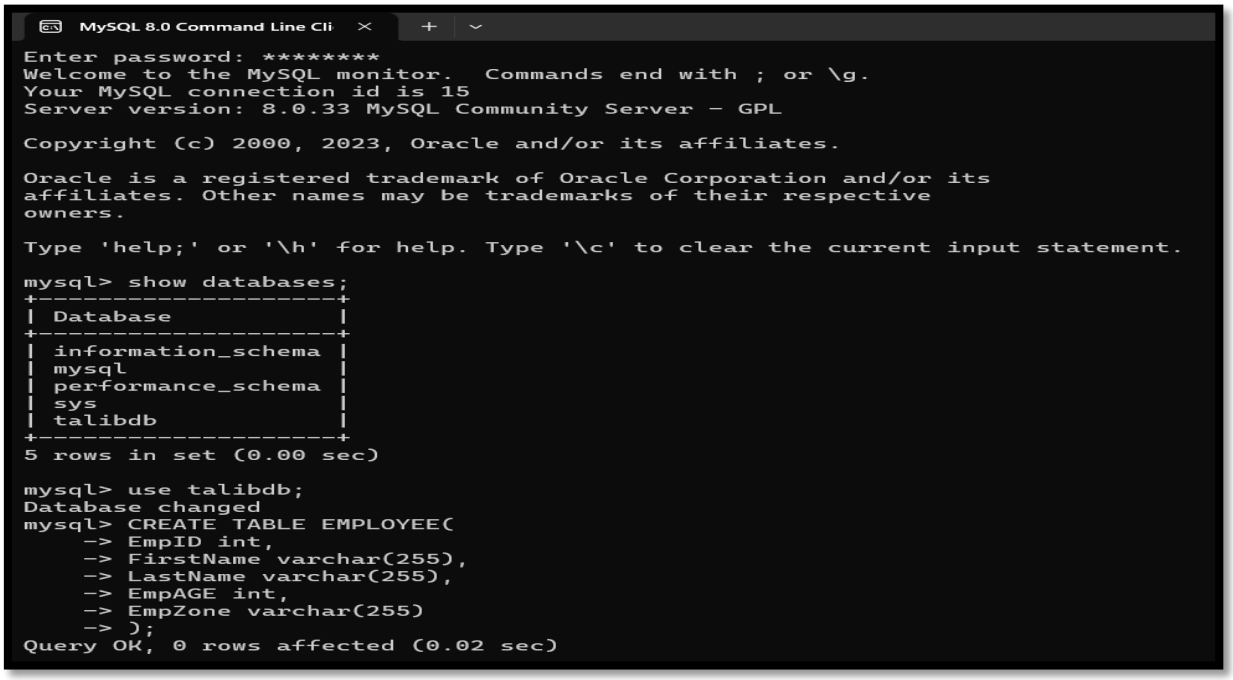
Step 1: - Install MySQL server on windows

Step 2: - Open MySQL command line

Step 3: - Creating a database

Step 4: - Creating a table in the database and inserting records

Step 5: - Inserting Records



```
mysql> DESC EMPLOYEE;
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| EmpID     | int          | YES  |     | NULL    |       |
| FirstName | varchar(255) | YES  |     | NULL    |       |
| LastName  | varchar(255) | YES  |     | NULL    |       |
| EmpAGE    | int          | YES  |     | NULL    |       |
| EmpZone   | varchar(255) | YES  |     | NULL    |       |
+-----------+--------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql>
```

MySQL 8.0 Command Line Cli

```
mysql> use talibdb;
Database changed
mysql> INSERT INTO EMPLOYEE(EmpID, FirstName, LastName, EmpAGE, EMPZONE) VALUES(1, 'Jack', 'Sparrow', 25, 'North');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM EMPLOYEE;
+-------+-----------+----------+--------+---------+
| EmpID | FirstName | LastName | EmpAGE | EmpZone |
+-------+-----------+----------+--------+---------+
|     1 | Jack      | Sparrow  |     25 | North   |
+-------+-----------+----------+--------+---------+
1 row in set (0.00 sec)

mysql> INSERT INTO EMPLOYEE(EmpID, FirstName, LastName, EmpAGE, EMPZONE) VALUES(2, 'Sejal', 'Dahake', 25, 'South');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO EMPLOYEE(EmpID, FirstName, LastName, EmpAGE, EMPZONE) VALUES(3, 'Adnan', 'Shaikh', 21, 'East');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO EMPLOYEE(EmpID, FirstName, LastName, EmpAGE, EMPZONE) VALUES(4, 'Shebel', 'Qureshi', 27, 'West');
Query OK, 1 row affected (0.03 sec)

mysql> INSERT INTO EMPLOYEE(EmpID, FirstName, LastName, EmpAGE, EMPZONE) VALUES(5, 'Aisha', 'Shaikh', 20, 'North West');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM EMPLOYEE;
+-------+-----------+----------+--------+------------+
| EmpID | FirstName | LastName | EmpAGE | EmpZone    |
+-------+-----------+----------+--------+------------+
|     1 | Jack      | Sparrow  |     25 | North      |
|     2 | Sejal     | Dahake   |     25 | South      |
|     3 | Adnan     | Shaikh   |     21 | East       |
|     4 | Shebel    | Qureshi  |     27 | West       |
|     5 | Aisha     | Shaikh   |     20 | North West |
+-------+-----------+----------+--------+------------+
5 rows in set (0.00 sec)

mysql>
```

Practical No 2:

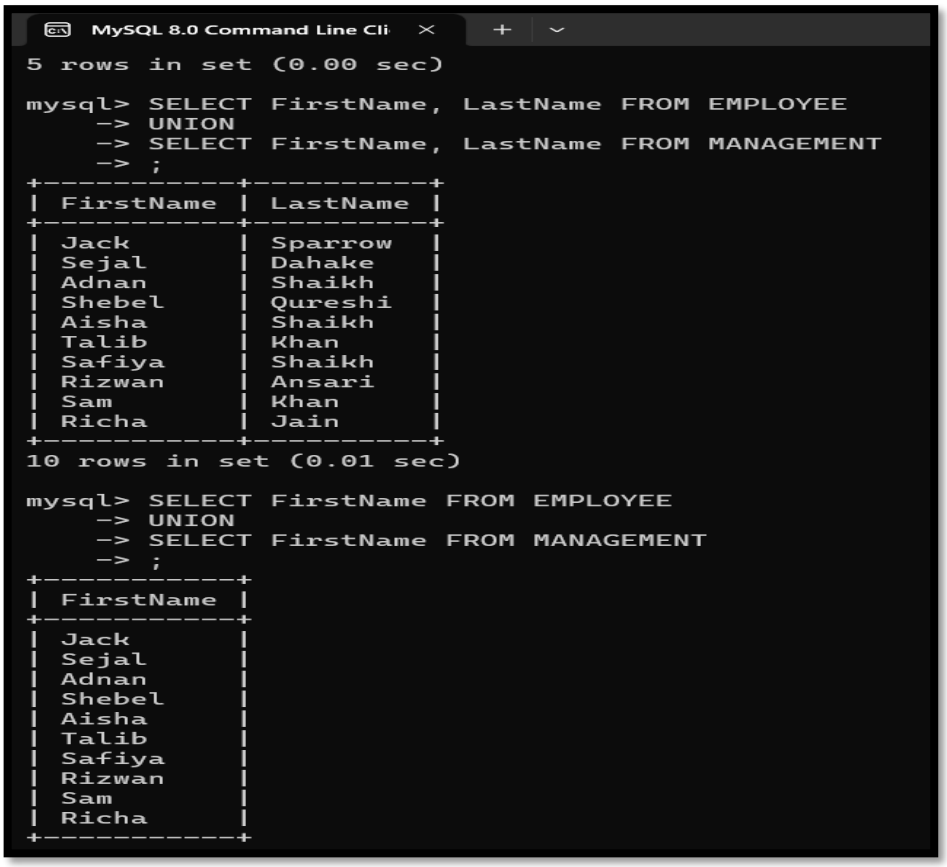
AIM:

IMPORT TABLES INTO MYSQL SERVER DATABASE USING IMPORT AND EXPORT TOOL AND TRANSFORM THE DATA AS THE SCHEMA.

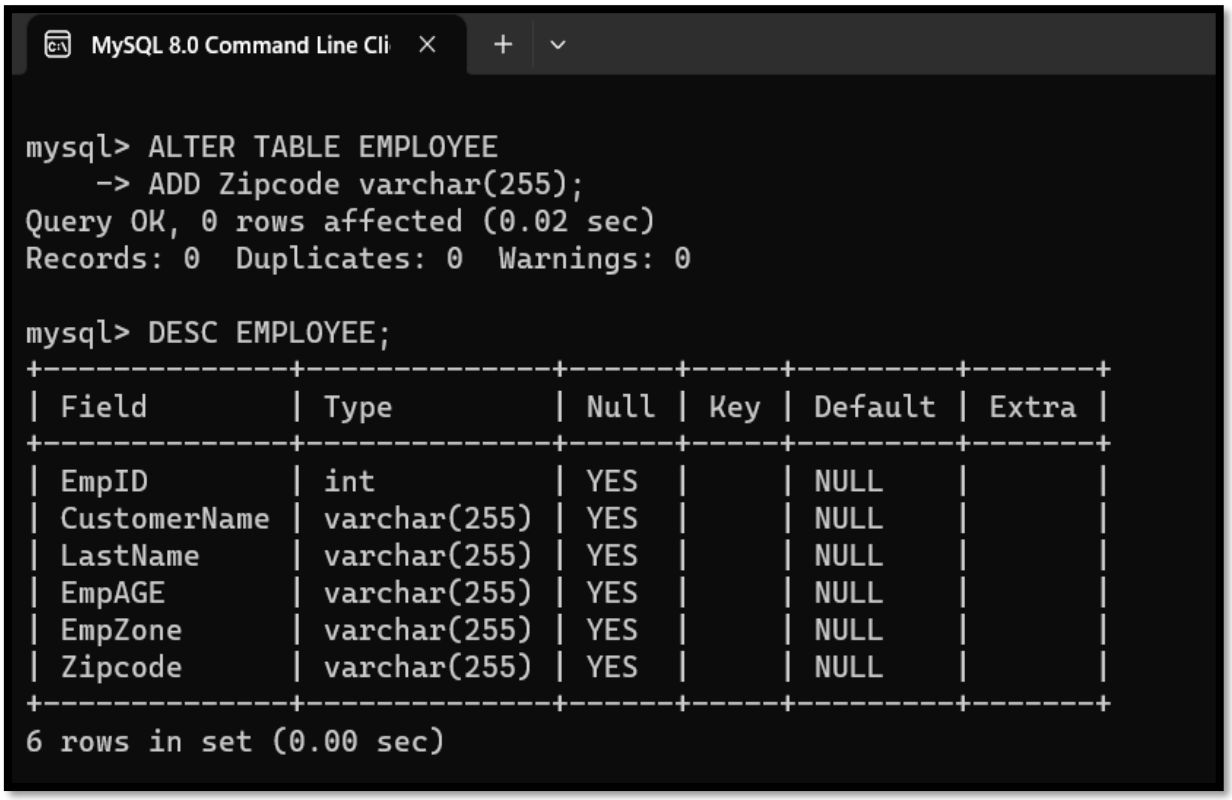
DESCRIPTION:

1. IN THE EMPLOYEE AND MANAGEMENT TABLES MERGE THE COLUMNS OF FIRST AND LAST NAME INTO ONE AND RENAME IT AS CUSTOMER NAME
2. IN MANAGEMENT TABLE CHANGE THE DATATYPE OF EMPLOYEE AGE TO VARCHAR (255)
3. IN THE STORES TABLE CHANGE THE DATATYPE ZIPCODE TO INT

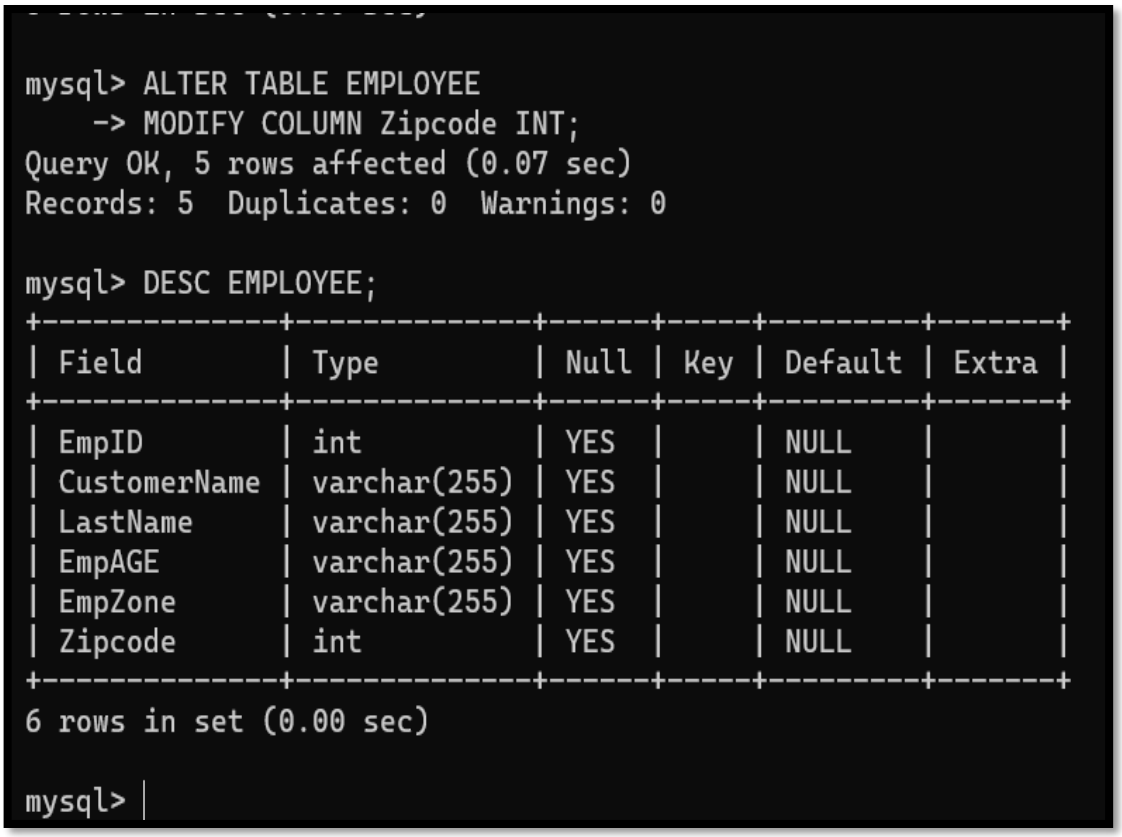
```
mysql> use talibdb;
Database changed
mysql> SELECT * FROM EMPLOYEE;
+-------+-----------+----------+--------+------------+
| EmpID | FirstName | LastName | EmpAGE | EmpZone    |
+-------+-----------+----------+--------+------------+
|     1 | Jack      | Sparrow  |     25 | North      |
|     2 | Sejal     | Dahake   |     25 | South      |
|     3 | Adnan     | Shaikh   |     21 | East       |
|     4 | Shebel    | Qureshi  |     27 | West       |
|     5 | Aisha     | Shaikh   |     20 | North West |
+-------+-----------+----------+--------+------------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM MANAGEMENT;
+--------+-----------+----------+--------+-----------+
| MngrID | FirstName | LastName | EmpAge | EmpZone   |
+--------+-----------+----------+--------+-----------+
|      1 | Talib     | Khan     |     40 | Mumbai    |
|      1 | Safiya    | Shaikh   |     35 | Mumbai    |
|      3 | Rizwan    | Ansari   |     38 | Hyderabad |
|      4 | Sam       | Khan     |     45 | Chenai    |
|      5 | Richa     | Jain     |     41 | Chenai    |
+--------+-----------+----------+--------+-----------+
5 rows in set (0.00 sec)

mysql>
```




```
mysql> ALTER TABLE EMPLOYEE
    -> RENAME COLUMN FirstName to CustomerName;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM EMPLOYEE;
+-------+--------------+----------+--------+------------+
| EmpID | CustomerName | LastName | EmpAGE | EmpZone    |
+-------+--------------+----------+--------+------------+
|     1 | Jack         | Sparrow  |     25 | North      |
|     2 | Sejal        | Dahake   |     25 | South      |
|     3 | Adnan        | Shaikh   |     21 | East       |
|     4 | Shebel       | Qureshi  |     27 | West       |
|     5 | Aisha        | Shaikh   |     20 | North West |
+-------+--------------+----------+--------+------------+
5 rows in set (0.00 sec)
```

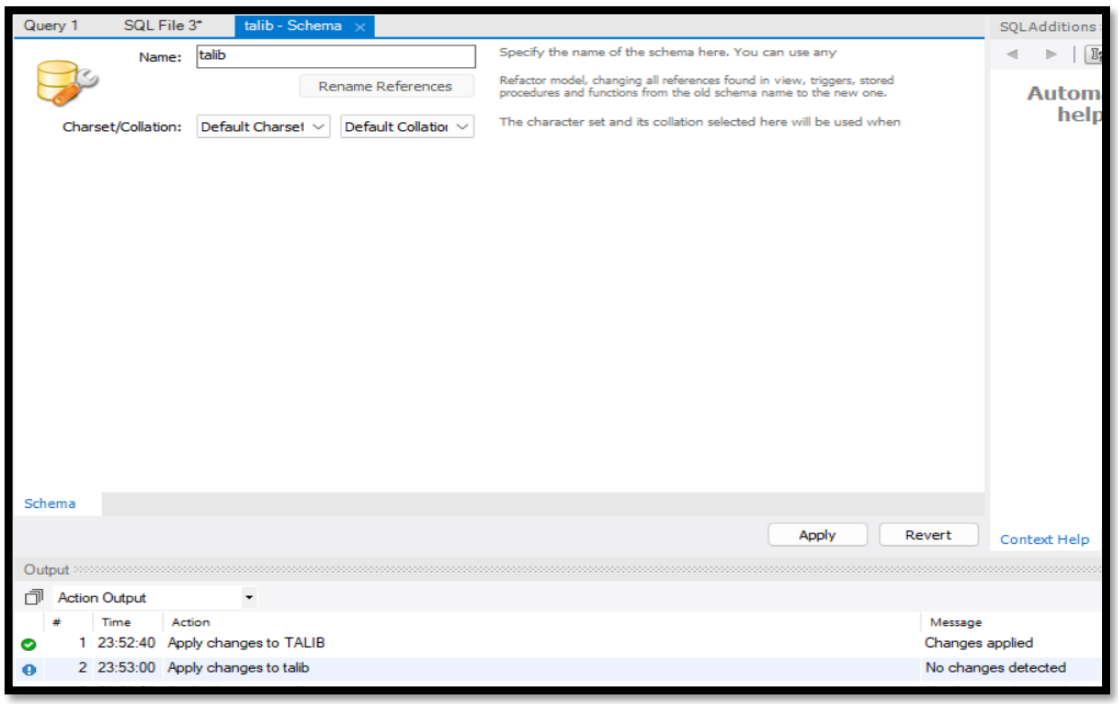


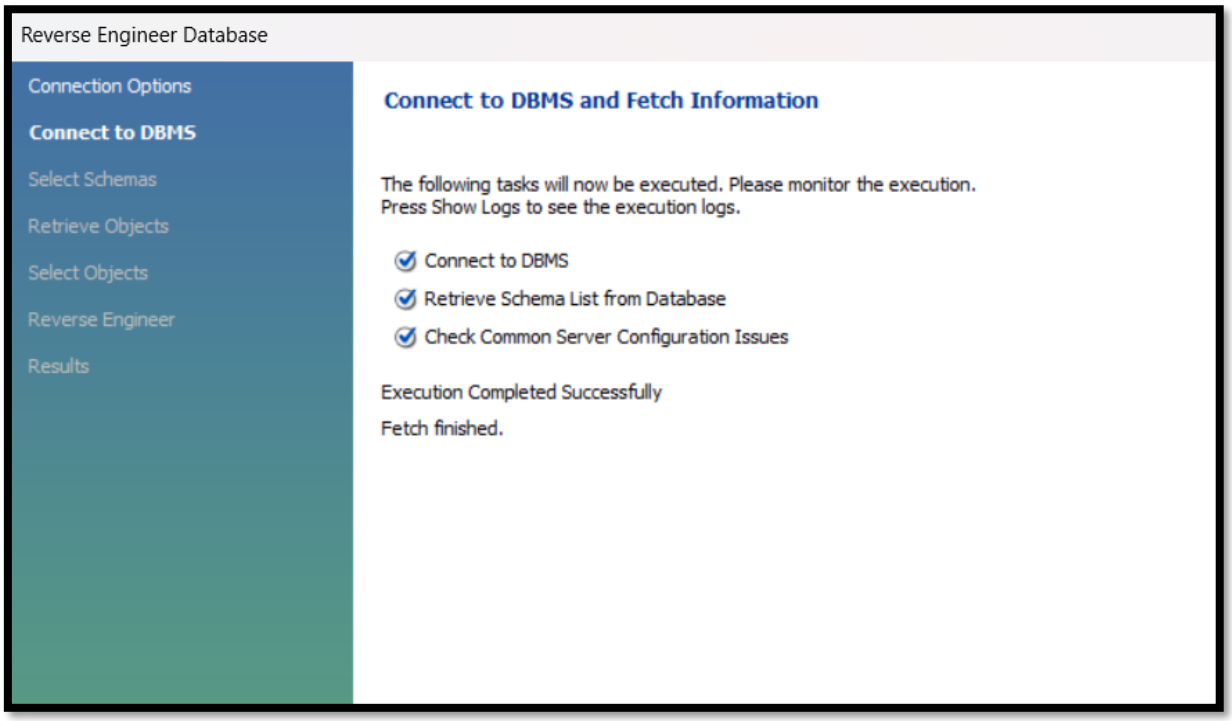

```
mysql> ALTER TABLE EMPLOYEE
    -> ADD Zipcode varchar(255);
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESC EMPLOYEE;
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| EmpID        | int          | YES  |     | NULL    |       |
| CustomerName | varchar(255) | YES  |     | NULL    |       |
| LastName     | varchar(255) | YES  |     | NULL    |       |
| EmpAGE       | varchar(255) | YES  |     | NULL    |       |
| EmpZone      | varchar(255) | YES  |     | NULL    |       |
| Zipcode      | varchar(255) | YES  |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

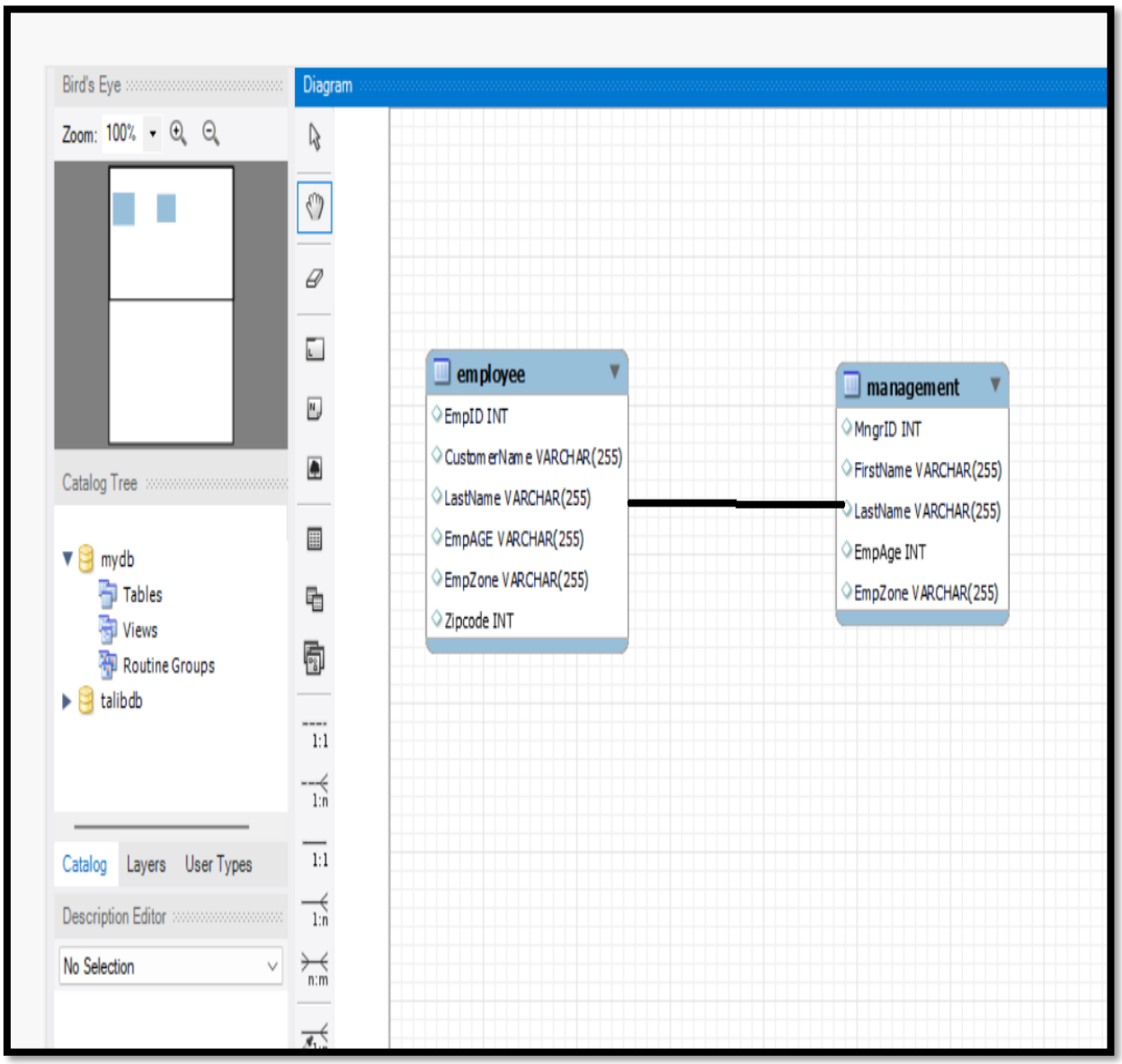


```
mysql> ALTER TABLE EMPLOYEE
    -> MODIFY COLUMN Zipcode INT;
Query OK, 5 rows affected (0.07 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> DESC EMPLOYEE;
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| EmpID        | int          | YES  |     | NULL    |       |
| CustomerName | varchar(255) | YES  |     | NULL    |       |
| LastName     | varchar(255) | YES  |     | NULL    |       |
| EmpAGE       | varchar(255) | YES  |     | NULL    |       |
| EmpZone      | varchar(255) | YES  |     | NULL    |       |
| Zipcode      | int          | YES  |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql>
```

CREATING SCHEMA






Bird's Eye
Zoom: 100%
Catalog Tree
mydb
Tables
Views
Routine Groups
talibdb
Catalog
Layers
User Types
Description Editor
No Selection
Diagram
1:1
1:n
1:1
1:n
n:m
employee
EmpID INT
CustomerName VARCHAR(255)
LastName VARCHAR(255)
EmpAGE VARCHAR(255)
EmpZone VARCHAR(255)
Zipcode INT
management
MngrID INT
FirstName VARCHAR(255)
LastName VARCHAR(255)
EmpAge INT
EmpZone VARCHAR(255)

Practical No 3:

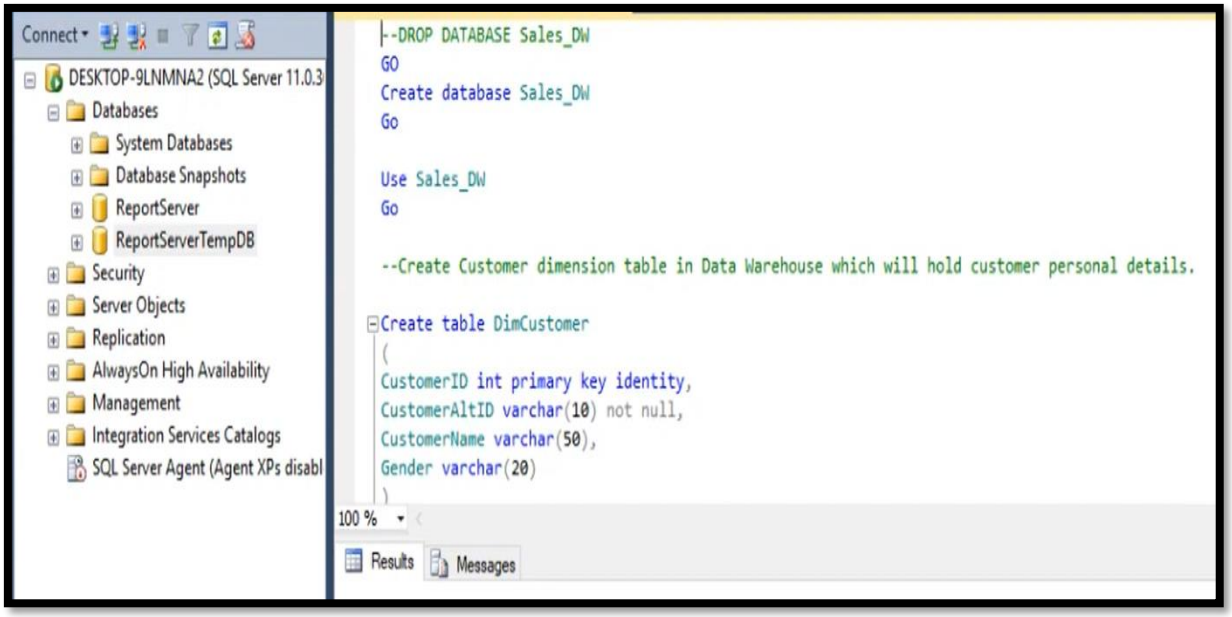
AIM:
Create an OLAP Cube [Online Analytical Processing Model]

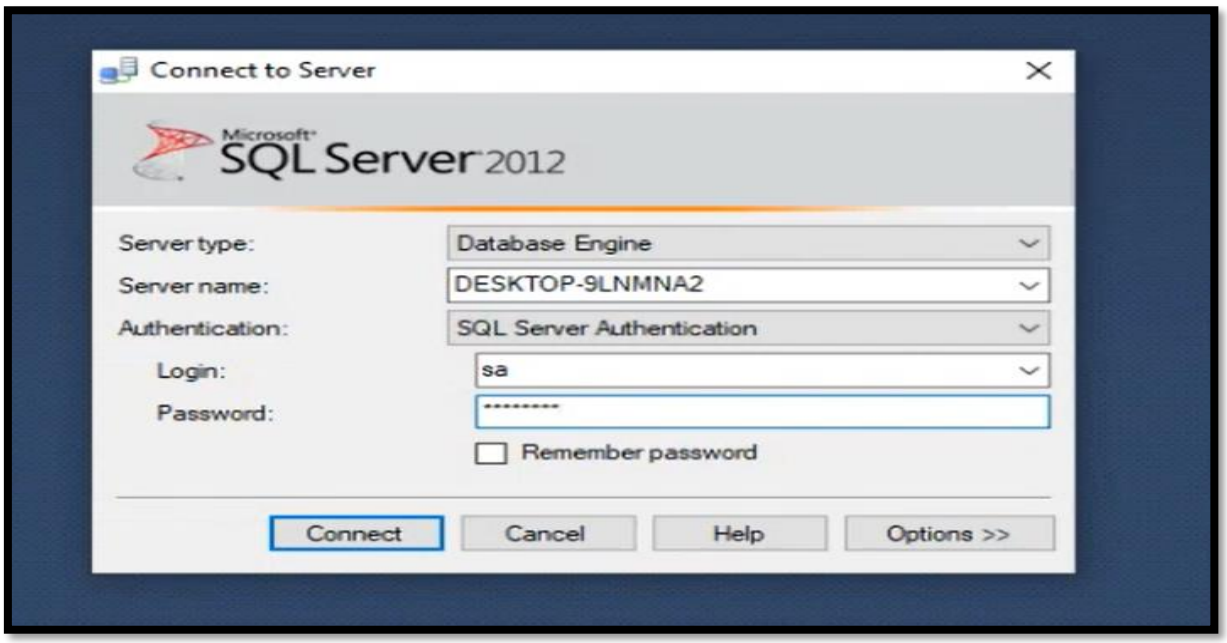
DESCRIPTION:
Create an OLAP Cube an Online Analytical Processing Model with the help of MySQL Server and Visual Studio.

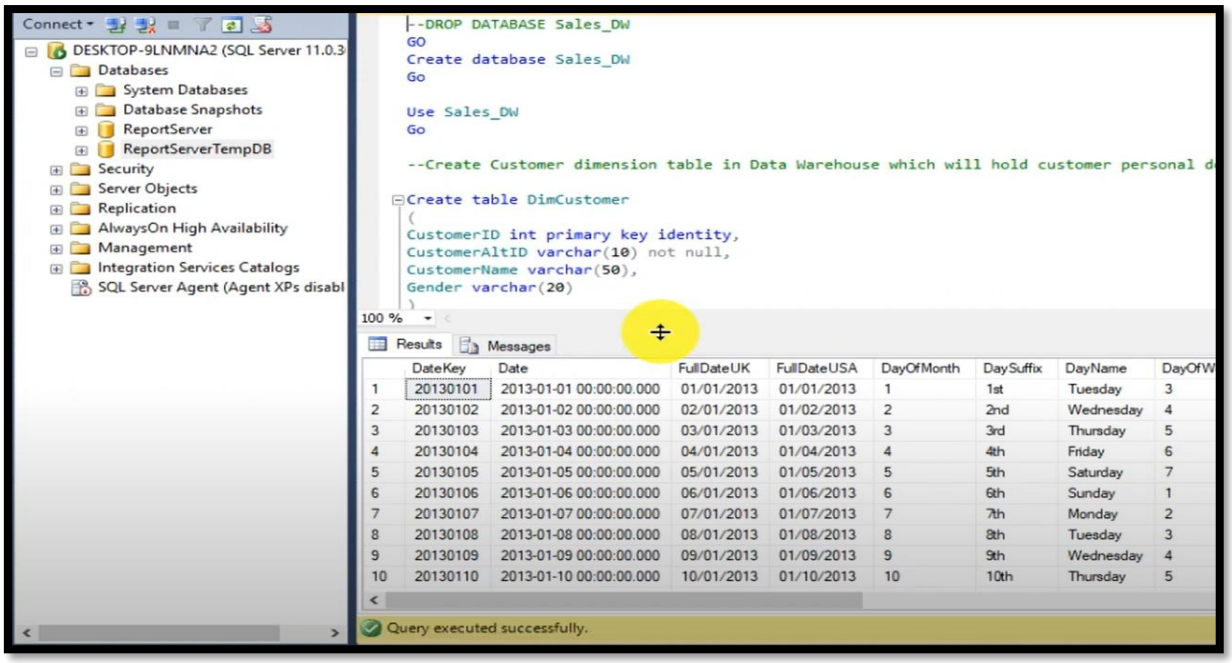
Procedure:

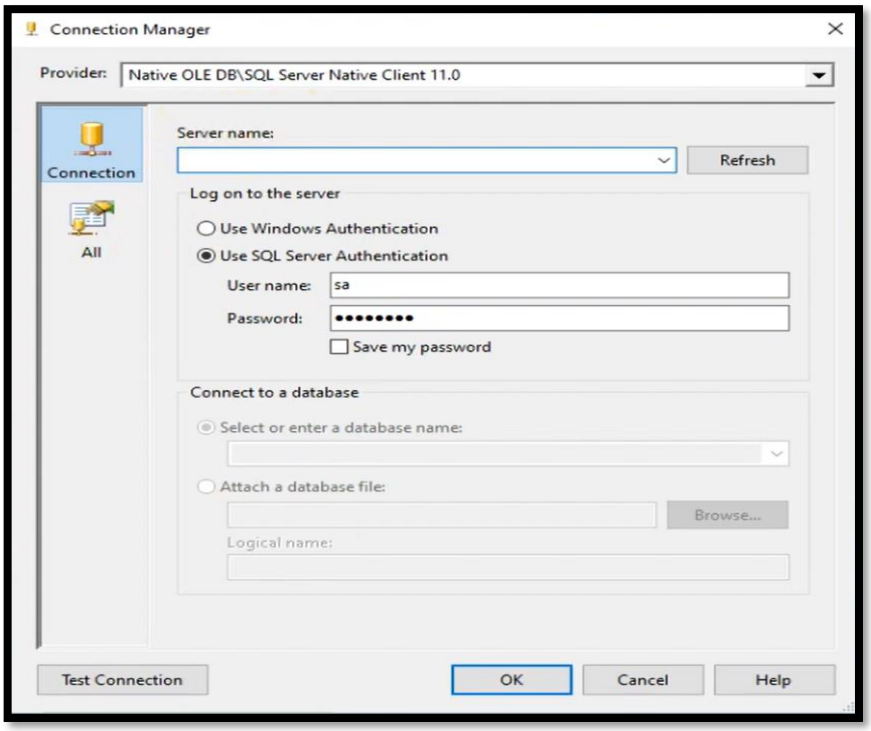
Steps:

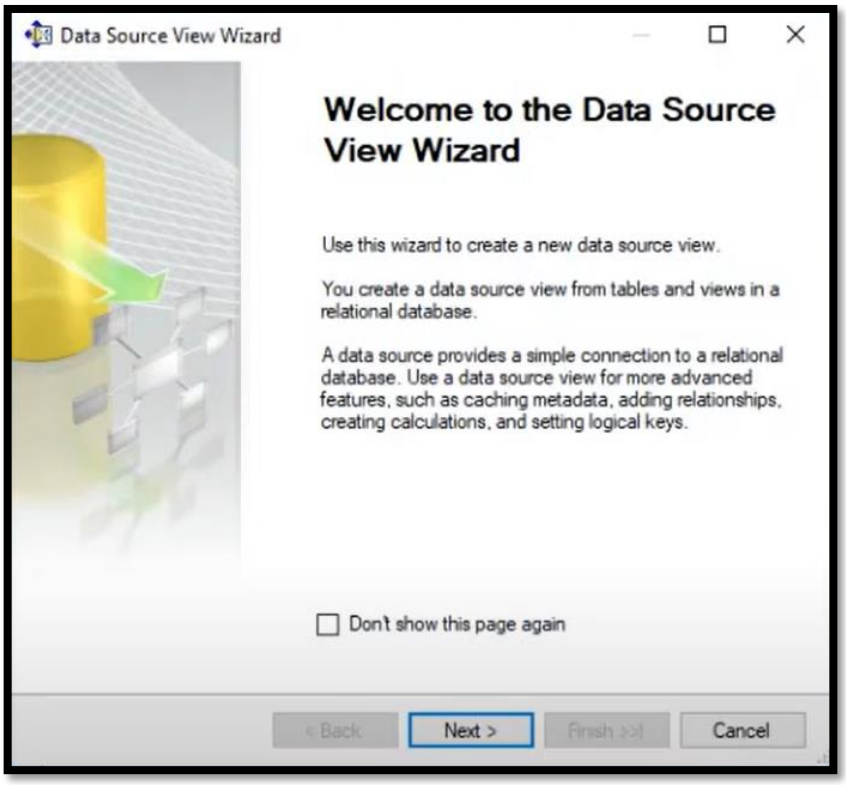
1. Open Start , MySQL Server

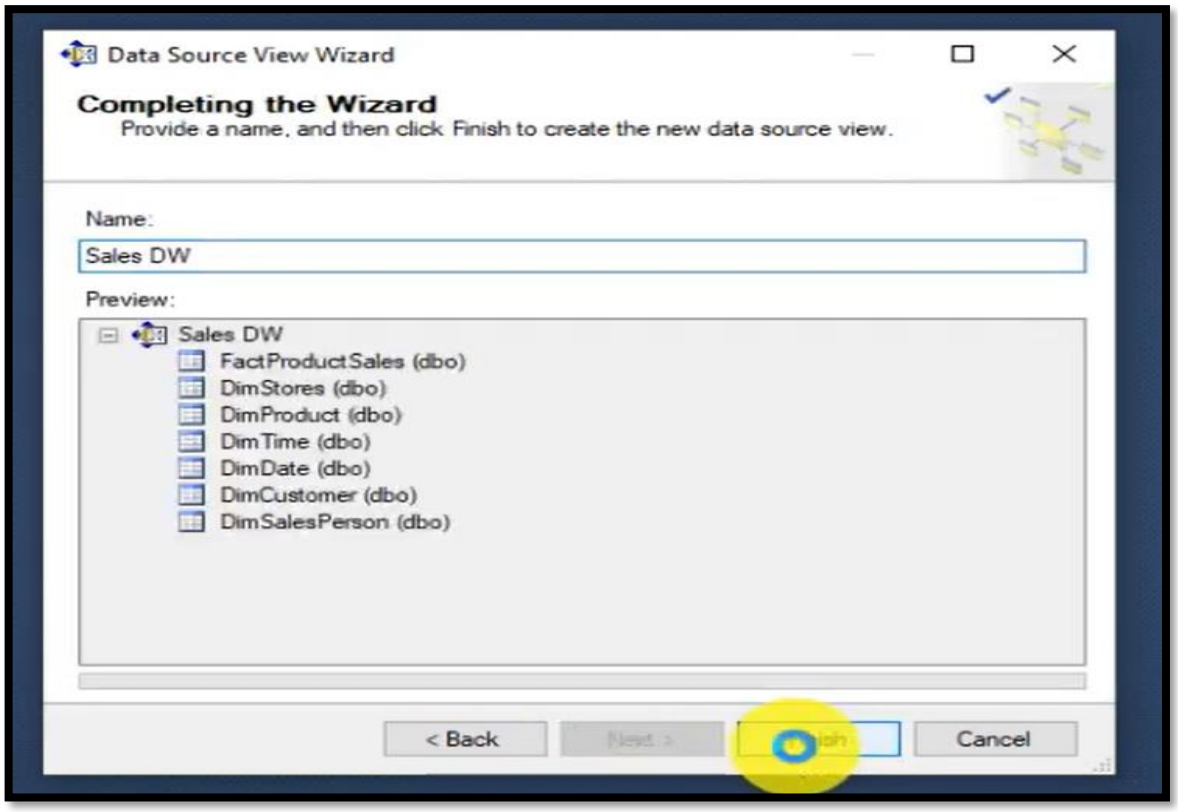


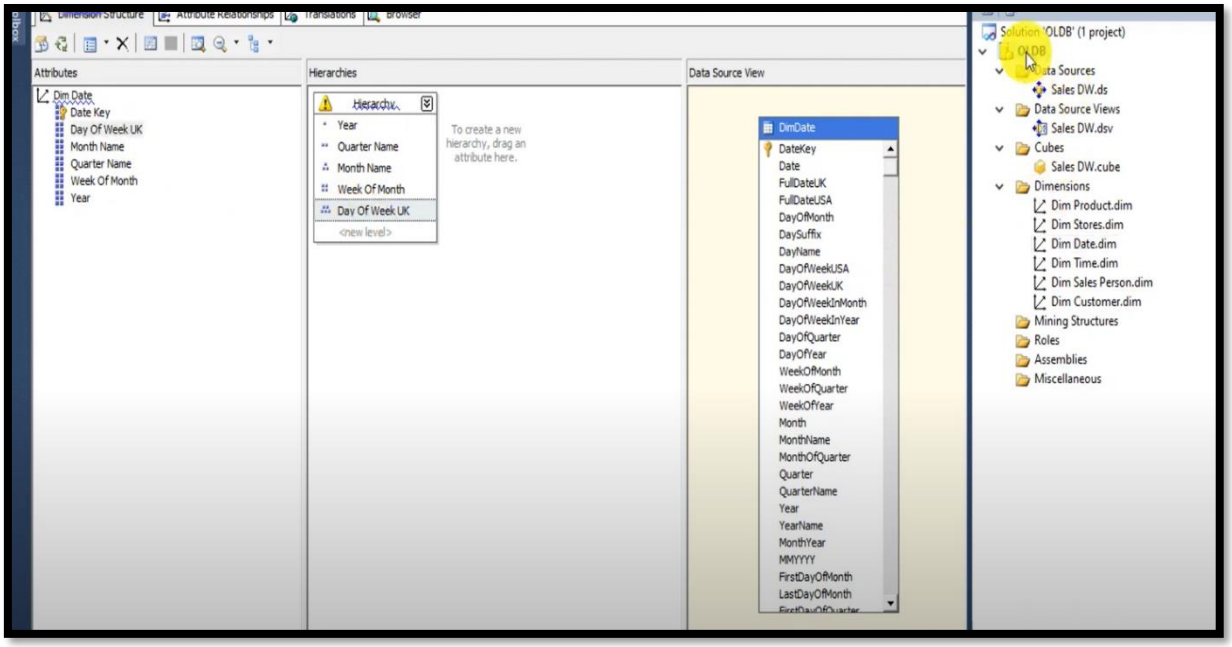

Connect to Server
Microsoft SQL Server 2012
Server type:
Database Engine
Server name:
DESKTOP-9LNMNA2
Authentication:
SQL Server Authentication
Login:
sa
Password:
Remember password
Connect
Cancel
Help
Options >>

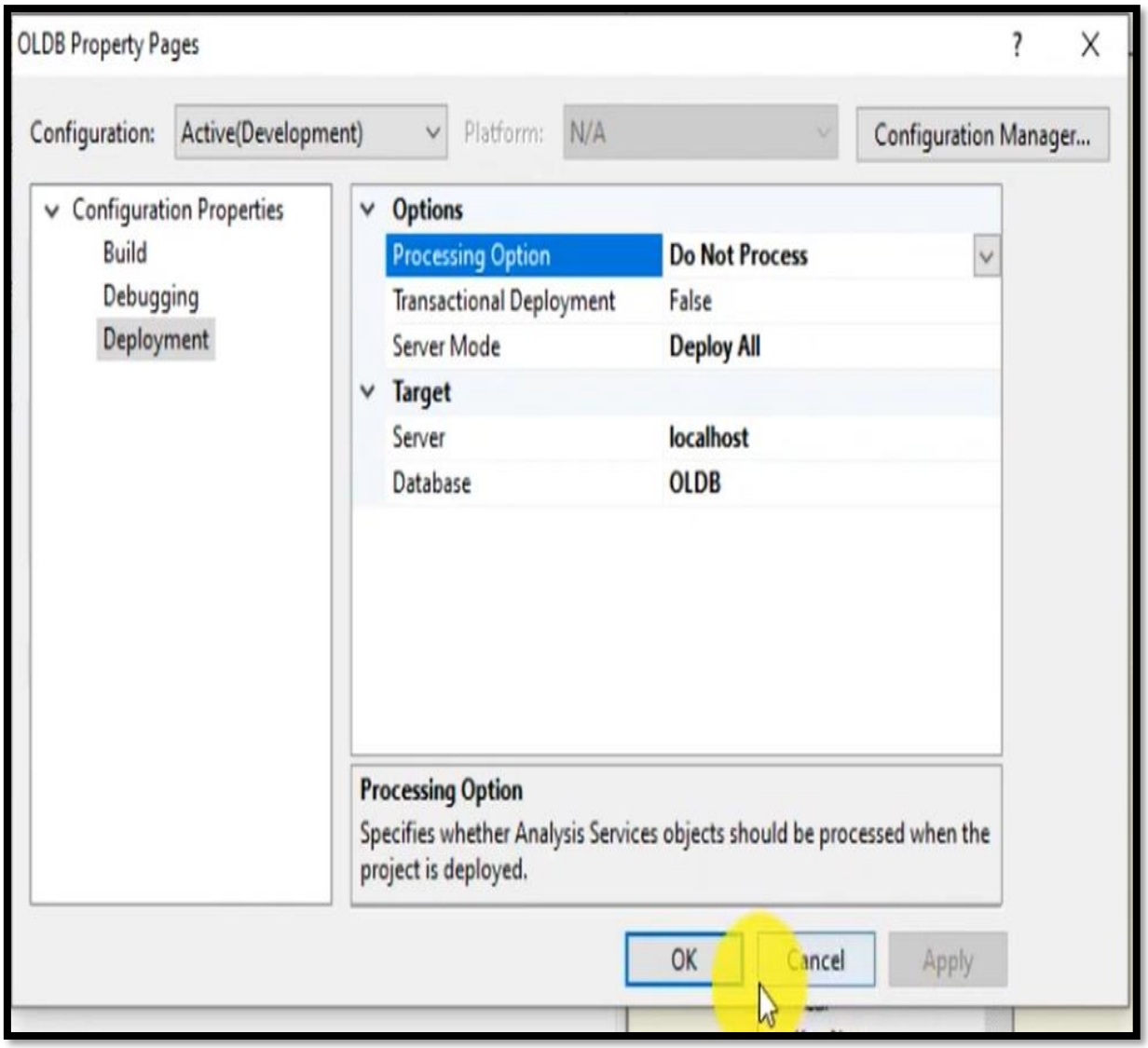

Connect
DESKTOP-9LNMNA2 (SQL Server 11.0.3
Databases
System Databases
Database Snapshots
ReportServer
ReportServerTempDB
Security
Server Objects
Replication
AlwaysOn High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs disabl
--DROP DATABASE Sales_DW
GO
Create database Sales_DW
Go
Use Sales_DW
Go
--Create Customer dimension table in Data Warehouse which will hold customer personal d
Create table DimCustomer
(
CustomerID int primary key identity,
CustomerAltID varchar(10) not null,
CustomerName varchar(50),
Gender varchar(20)
)
100 %
Results
Messages
DateKey Date FullDateUK FullDateUSA DayOfMonth DaySuffix DayName DayOfW
1 20130101 2013-01-01 00:00:00.000 01/01/2013 01/01/2013 1 1st Tuesday 3
2 20130102 2013-01-02 00:00:00.000 02/01/2013 01/02/2013 2 2nd Wednesday 4
3 20130103 2013-01-03 00:00:00.000 03/01/2013 01/03/2013 3 3rd Thursday 5
4 20130104 2013-01-04 00:00:00.000 04/01/2013 01/04/2013 4 4th Friday 6
5 20130105 2013-01-05 00:00:00.000 05/01/2013 01/05/2013 5 5th Saturday 7
6 20130106 2013-01-06 00:00:00.000 06/01/2013 01/06/2013 6 6th Sunday 1
7 20130107 2013-01-07 00:00:00.000 07/01/2013 01/07/2013 7 7th Monday 2
8 20130108 2013-01-08 00:00:00.000 08/01/2013 01/08/2013 8 8th Tuesday 3
9 20130109 2013-01-09 00:00:00.000 09/01/2013 01/09/2013 9 9th Wednesday 4
10 20130110 2013-01-10 00:00:00.000 10/01/2013 01/10/2013 10 10th Thursday 5
Query executed successfully.


Connection Manager
Provider:
Native OLE DB\SQL Server Native Client 11.0
Connection
All
Server name:
Refresh
Log on to the server
Use Windows Authentication
Use SQL Server Authentication
User name:
sa
Password:
Save my password
Connect to a database
Select or enter a database name:
Attach a database file:
Browse...
Logical name:
Test Connection
OK
Cancel
Help


Data Source View Wizard
Welcome to the Data Source View Wizard
Use this wizard to create a new data source view.
You create a data source view from tables and views in a relational database.
A data source provides a simple connection to a relational database. Use a data source view for more advanced features, such as caching metadata, adding relationships, creating calculations, and setting logical keys.
Don't show this page again
< Back
Next >
Finish >>|
Cancel


Data Source View Wizard
Completing the Wizard
Provide a name, and then click Finish to create the new data source view.
Name:
Sales DW
Preview:
Sales DW
FactProductSales (dbo)
DimStores (dbo)
DimProduct (dbo)
DimTime (dbo)
DimDate (dbo)
DimCustomer (dbo)
DimSalesPerson (dbo)
< Back
Next >
Finish
Cancel


Attributes
Hierarchies
Data Source View
Dim Date
Date Key
Day Of Week UK
Month Name
Quarter Name
Week Of Month
Year
Hierarchy
Year
Quarter Name
Month Name
Week Of Month
Day Of Week UK
<new level>
To create a new hierarchy, drag an attribute here.
DimDate
DateKey
Date
FullDateUK
FullDateUSA
DayOfMonth
DaySuffix
DayName
DayOfWeekUSA
DayOfWeekUK
DayOfWeekInMonth
DayOfWeekInYear
DayOfQuarter
DayOfYear
WeekOfMonth
WeekOfQuarter
WeekOfYear
Month
MonthName
MonthOfQuarter
Quarter
QuarterName
Year
YearName
MonthYear
MMYYYY
FirstDayOfMonth
LastDayOfMonth
Solution 'OLDB' (1 project)
OLDB
Data Sources
Sales DW.ds
Data Source Views
Sales DW.dsv
Cubes
Sales DW.cube
Dimensions
Dim Product.dim
Dim Stores.dim
Dim Date.dim
Dim Time.dim
Dim Sales Person.dim
Dim Customer.dim
Mining Structures
Roles
Assemblies
Miscellaneous


OLDB Property Pages
Configuration: Active(Development)
Platform: N/A
Configuration Manager...
Configuration Properties
Build
Debugging
Deployment
Options
Processing Option
Do Not Process
Transactional Deployment
False
Server Mode
Deploy All
Target
Server
localhost
Database
OLDB
Processing Option
Specifies whether Analysis Services objects should be processed when the project is deployed.
OK
Cancel
Apply

Practical No 4:

AIM:
Create a Weather Table with the help of Data Mining Tool WEKA.

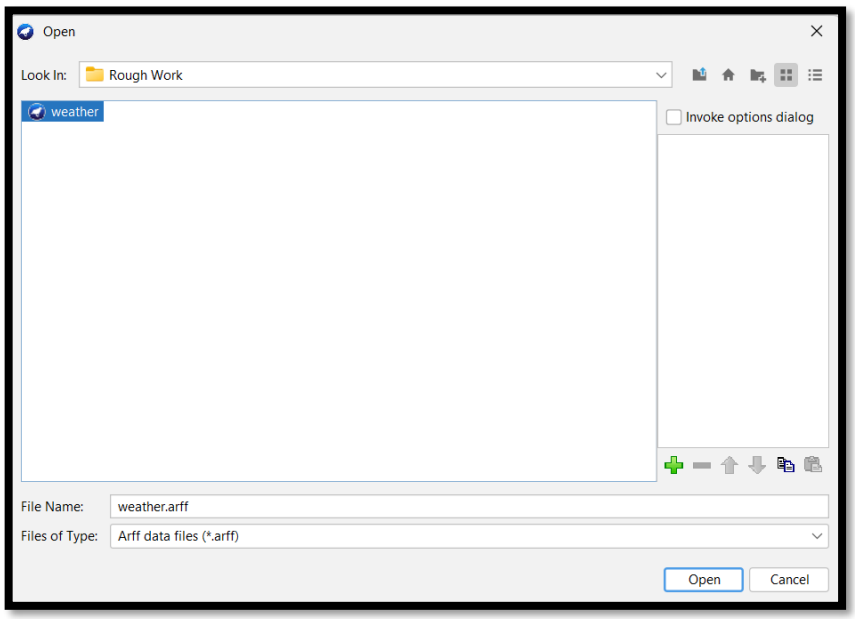
DESCRIPTION:
We need to create a Weather Table with training data set which includes attributes like outlook, temperature, humidity, windy, play.

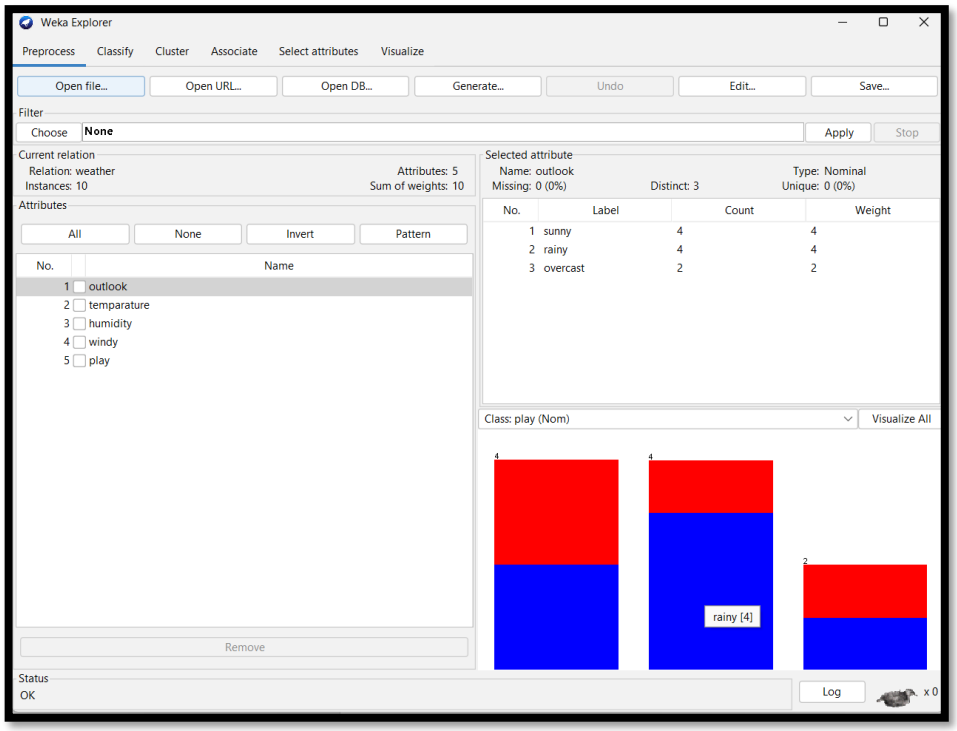
PROCEDURE:
Steps:
1. Open Start, Programs, Accessories. Notepad
2. Type the following training data set with the help of Notepad for Weather Table.

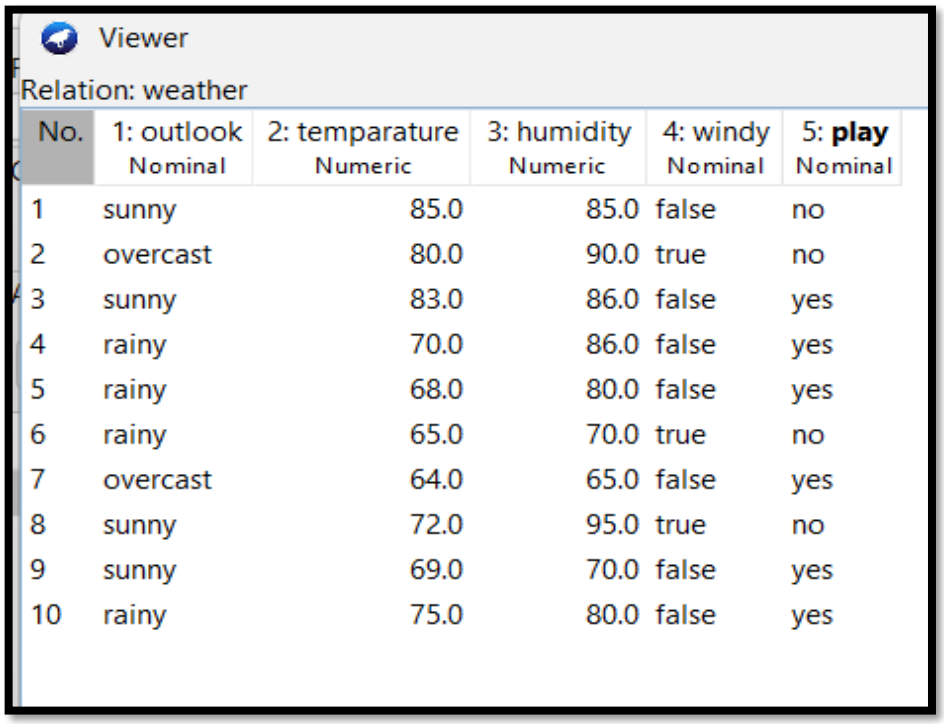
```
@relation weather
@attribute outlook {sunny,rainy,overcast}
@attribute temparature numeric
 @attribute humidity numeric
 @attribute windy {true,false}
@attribute play {yes,no}

@data
sunny,85.0,85.0,false,no
overcast,80.0,90.0,true,no
sunny,83.0,86.0,false,yes
rainy,70.0,86.0,false,yes
 rainy,68.0,80.0,false,yes
rainy,65.0,70.0,true,no
overcast,64.0,65.0,false,yes
 sunny,72.0,95.0,true,no
sunny,69.0,70.0,false,yes
rainy,75.0,80.0,false,yes
```

3. After that the file is saved with .arff file format.

4. Minimize the arff file and then open Start ◊ Programs ◊ weka-3-4.
5. Click on weka-3-4, then Weka dialog box is displayed on the screen.
6. In that dialog box there are four modes, click on explorer.
7. Explorer shows many options. In that click on 'open file' and select the arff file
8. Click on edit button which shows weather table on weka.







Viewer

Relation: weather

No.	1: outlook Nominal	2: temparature Numeric	3: humidity Numeric	4: windy Nominal	5: **play** Nominal
1	sunny	85.0	85.0	false	no
2	overcast	80.0	90.0	true	no
3	sunny	83.0	86.0	false	yes
4	rainy	70.0	86.0	false	yes
5	rainy	68.0	80.0	false	yes
6	rainy	65.0	70.0	true	no
7	overcast	64.0	65.0	false	yes
8	sunny	72.0	95.0	true	no
9	sunny	69.0	70.0	false	yes
10	rainy	75.0	80.0	false	yes

Practical No 5:

Aim:
Apply Pre-Processing techniques to the training data set of Weather Table

Description:
Real world databases are highly influenced to noise, missing and inconsistency due to their queue size so the data can be pre-processed to improve the quality of data and missing results and it also improves the efficiency. There are 3 pre-processing techniques they are:

1) Add
2) Remove
3) Normalization

PROCEDURE:

Steps:

1) Open Start, Programs, Accessories. Notepad

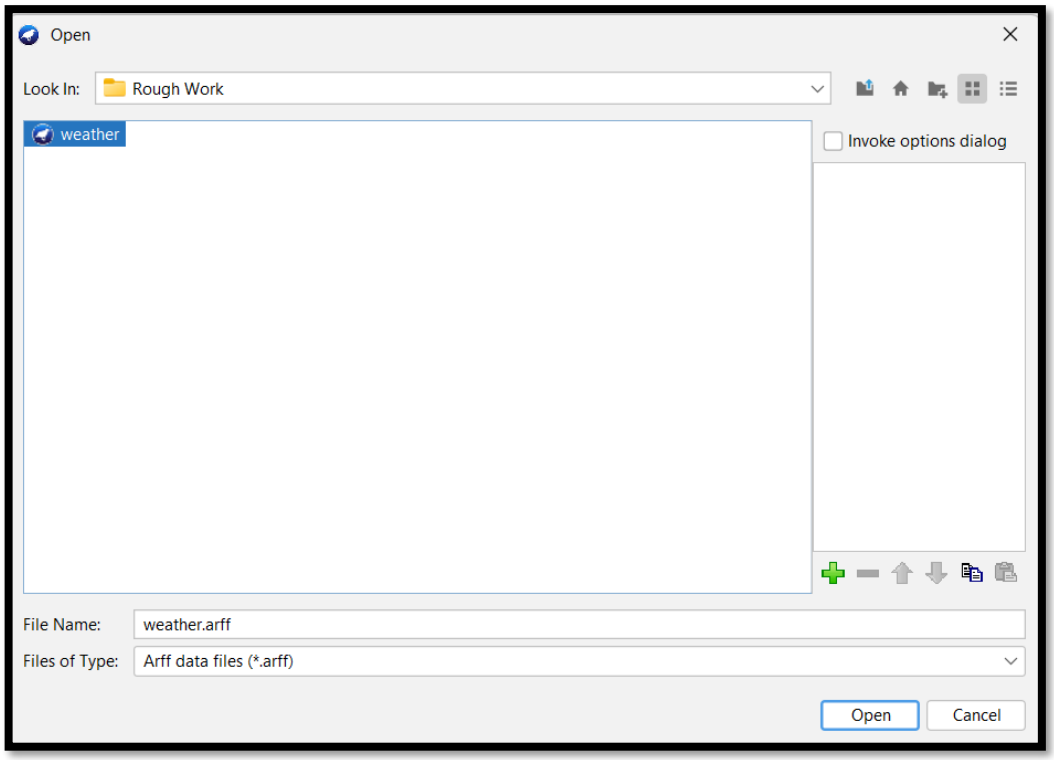
2) Type the following training data set with the help of Notepad for Weather Table.

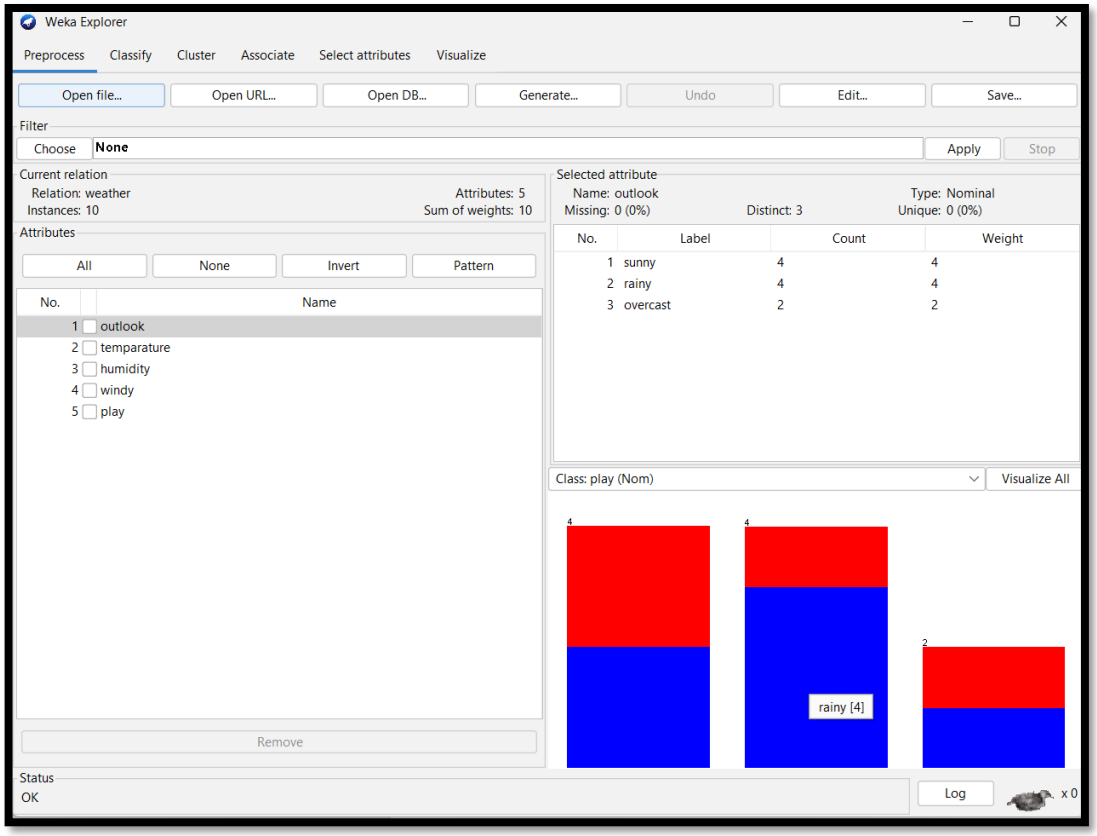
```
@relation weather
@attribute outlook {sunny,rainy,overcast}
@attribute temparature numeric
@attribute humidity numeric
@attribute windy {true,false}
@attribute play {yes,no}

@data
sunny,85.0,85.0,false,no
overcast,80.0,90.0,true,no
sunny,83.0,86.0,false,yes
rainy,70.0,86.0,false,yes
rainy,68.0,80.0,false,yes
rainy,65.0,70.0,true,no
overcast,64.0,65.0,false,yes
```

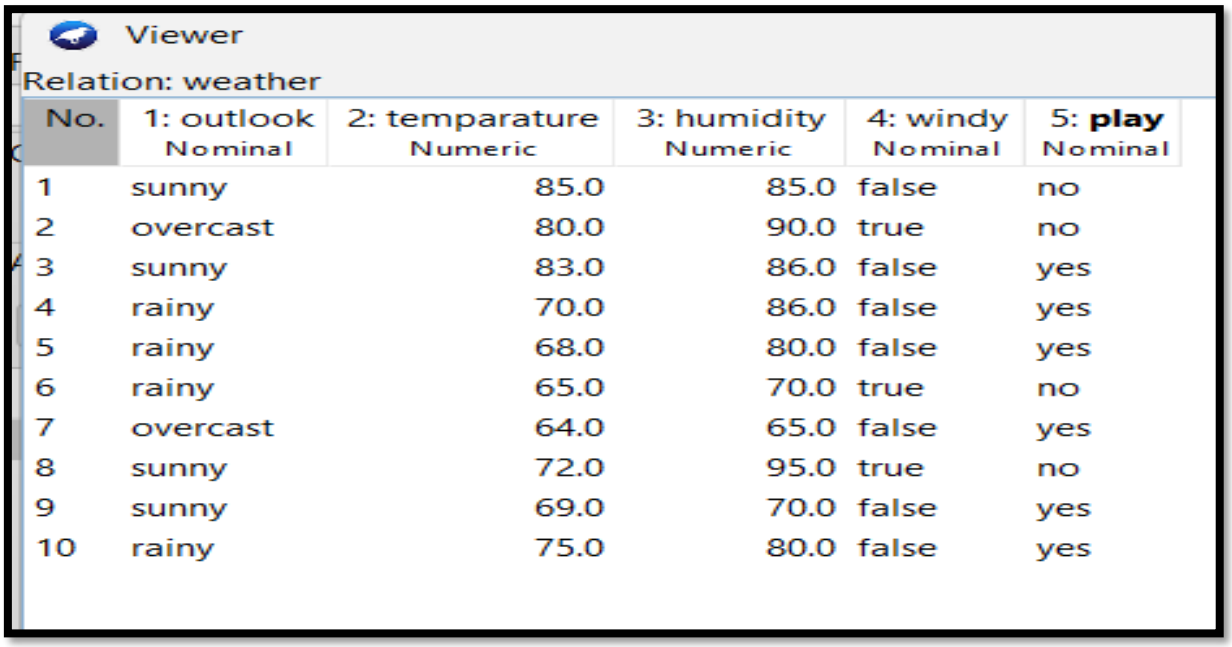
sunny,72.0,95.0,true,no
sunny,69.0,70.0,false,yes
rainy,75.0,80.0,false,yes

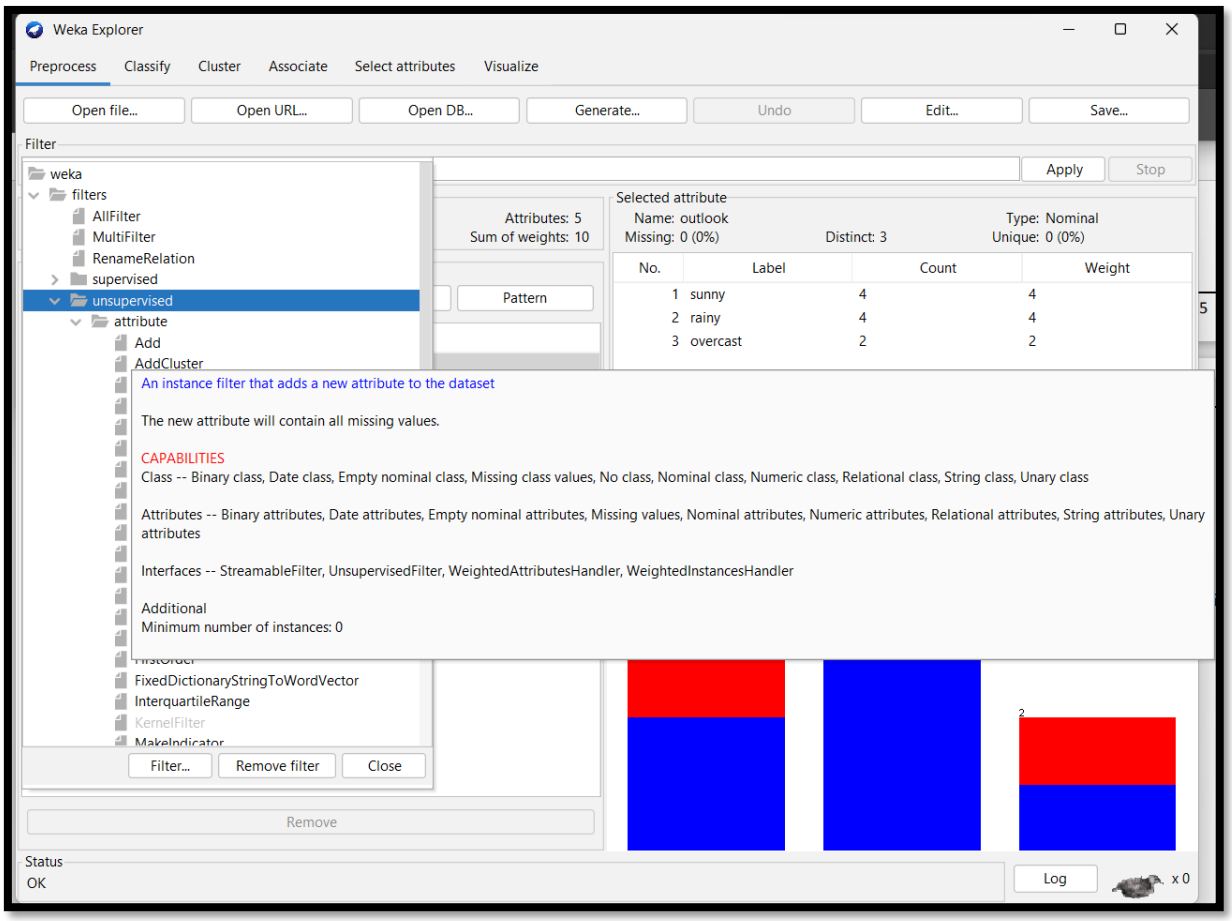
3) After that the file is saved with .arff file format.
4) Minimize the arff file and then open Start ◊ Programs ◊ weka-3-4.
5) Click on weka-3-4, then Weka dialog box is displayed on the screen.
6) In that dialog box there are four modes, click on explorer.
7) Explorer shows many options. In that click on 'open file' and select the arff file
8) Click on edit button which shows weather table on weka.

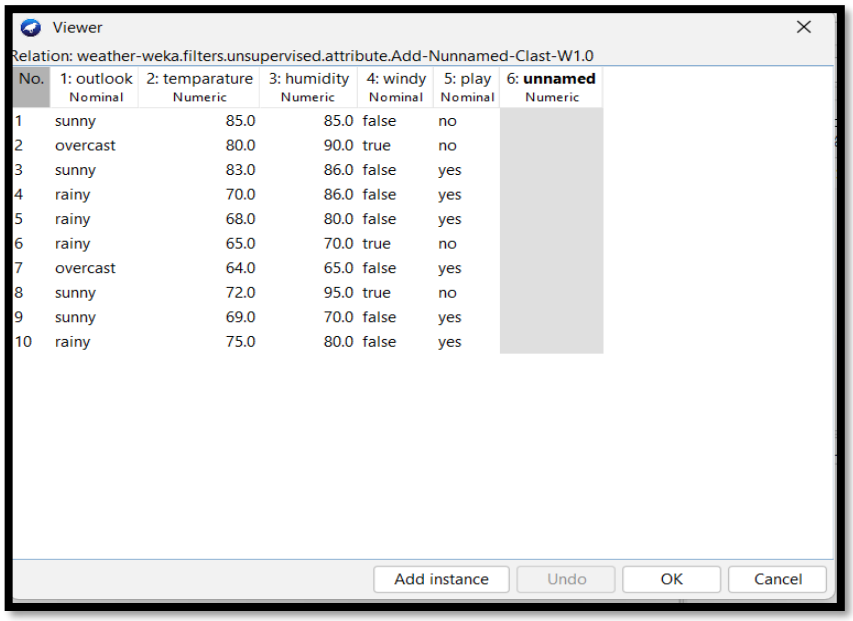
ADD PRE-PROCESSING TECHNIQUES


Open
Look In:
Rough Work
weather
Invoke options dialog
File Name:
weather.arff
Files of Type:
Arff data files (*.arff)
Open
Cancel

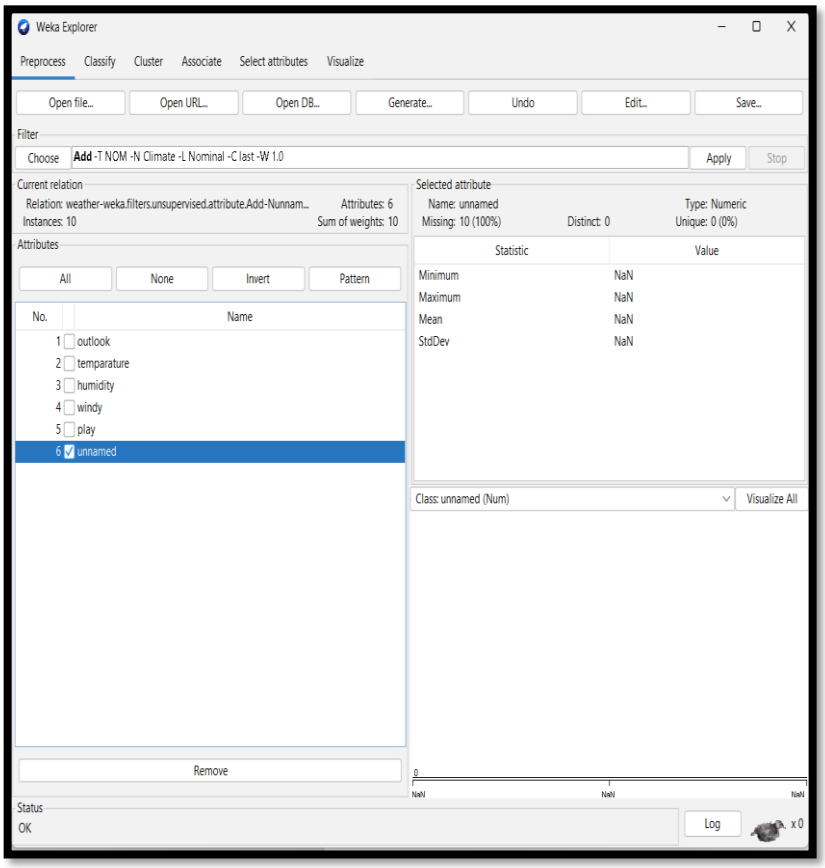

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Stop
Current relation
Relation: weather
Instances: 10
Attributes: 5
Sum of weights: 10
Attributes
All
None
Invert
Pattern
No.
Name
1 outlook
2 temparature
3 humidity
4 windy
5 play
Remove
Selected attribute
Name: outlook
Missing: 0 (0%)
Distinct: 3
Type: Nominal
Unique: 0 (0%)
No.
Label
Count
Weight
1 sunny 4 4
2 rainy 4 4
3 overcast 2 2
Class: play (Nom)
Visualize All
rainy [4]
Status
OK
Log
x 0

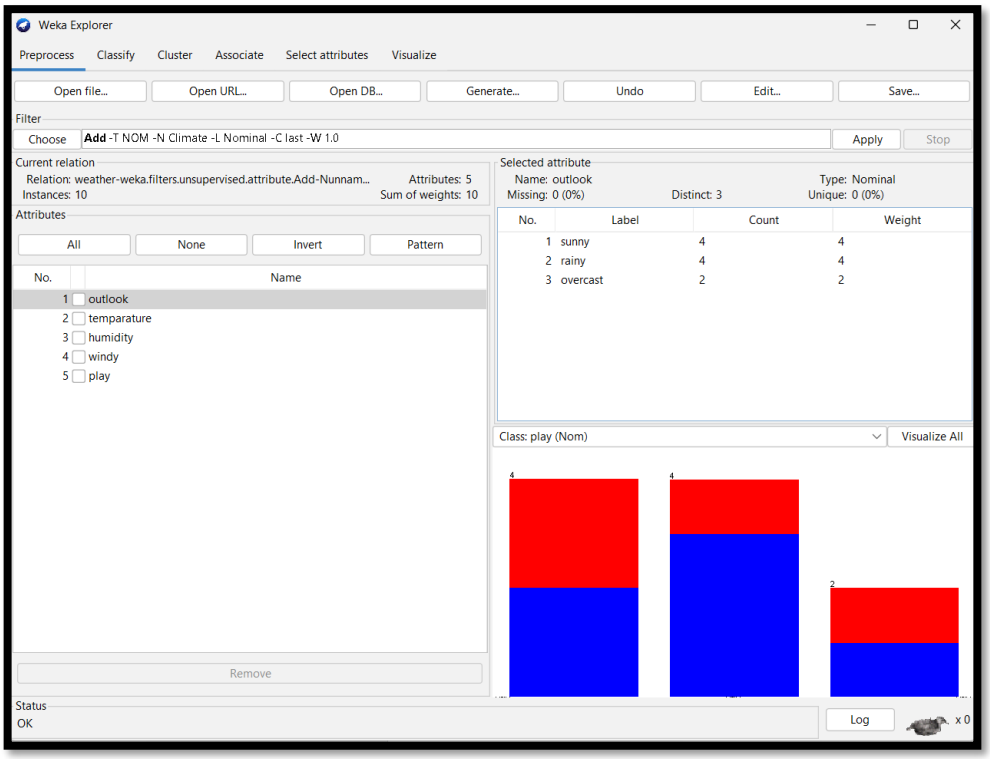

Viewer
Relation: weather
No. 1: outlook Nominal 2: temparature Numeric 3: humidity Numeric 4: windy Nominal 5: play Nominal
1 sunny 85.0 85.0 false no
2 overcast 80.0 90.0 true no
3 sunny 83.0 86.0 false yes
4 rainy 70.0 86.0 false yes
5 rainy 68.0 80.0 false yes
6 rainy 65.0 70.0 true no
7 overcast 64.0 65.0 false yes
8 sunny 72.0 95.0 true no
9 sunny 69.0 70.0 false yes
10 rainy 75.0 80.0 false yes

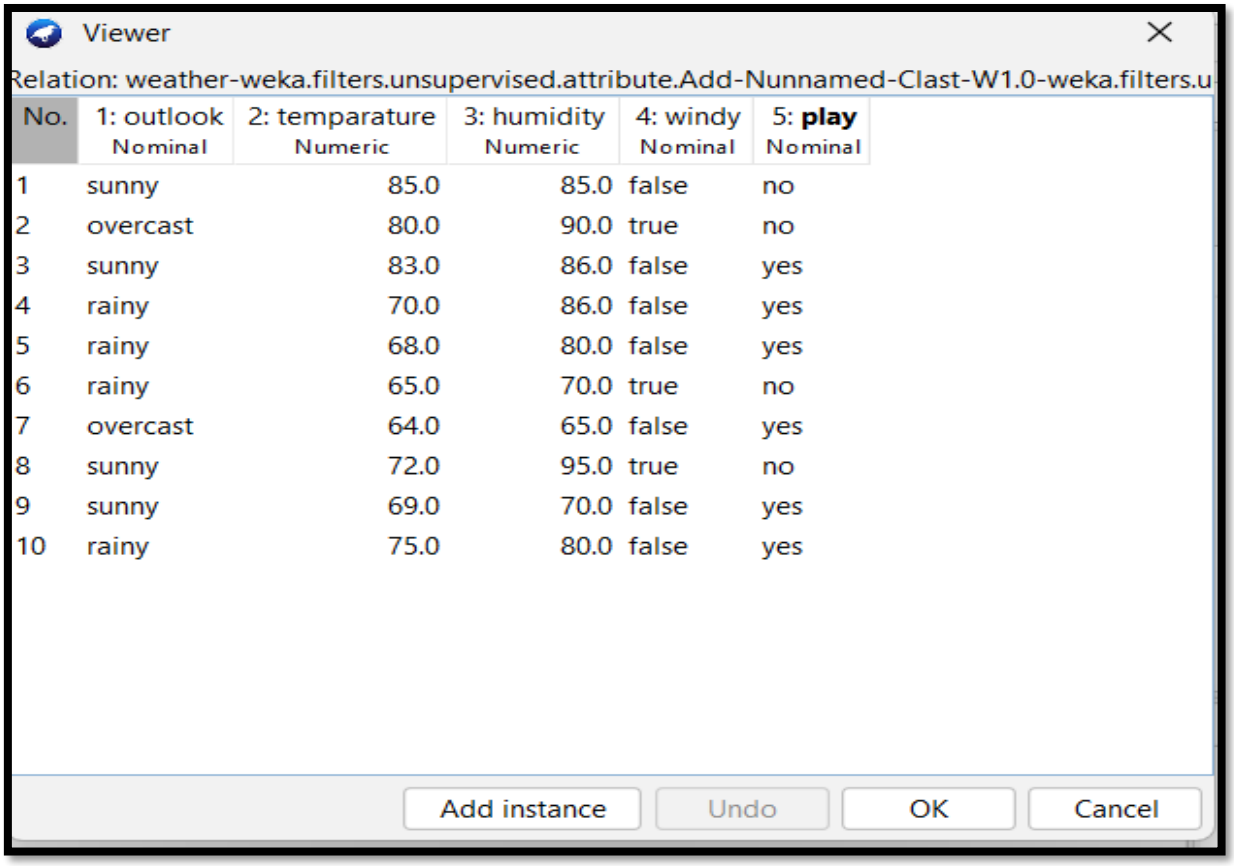

Weka Explorer
Preprocess Classify Cluster Associate Select attributes Visualize
Open file... Open URL... Open DB... Generate... Undo Edit... Save...
Filter
Apply Stop
weka
filters
AllFilter
MultiFilter
RenameRelation
supervised
unsupervised
attribute
Add
AddCluster
Attributes: 5
Sum of weights: 10
Pattern
Selected attribute
Name: outlook Type: Nominal
Missing: 0 (0%) Distinct: 3 Unique: 0 (0%)
No. Label Count Weight
1 sunny 4 4
2 rainy 4 4
3 overcast 2 2
An instance filter that adds a new attribute to the dataset
The new attribute will contain all missing values.
CAPABILITIES
Class -- Binary class, Date class, Empty nominal class, Missing class values, No class, Nominal class, Numeric class, Relational class, String class, Unary class
Attributes -- Binary attributes, Date attributes, Empty nominal attributes, Missing values, Nominal attributes, Numeric attributes, Relational attributes, String attributes, Unary attributes
Interfaces -- StreamableFilter, UnsupervisedFilter, WeightedAttributesHandler, WeightedInstancesHandler
Additional
Minimum number of instances: 0
FixedDictionaryStringToWordVector
InterquartileRange
KernelFilter
Filter... Remove filter Close
Remove
Status
OK
Log
x 0



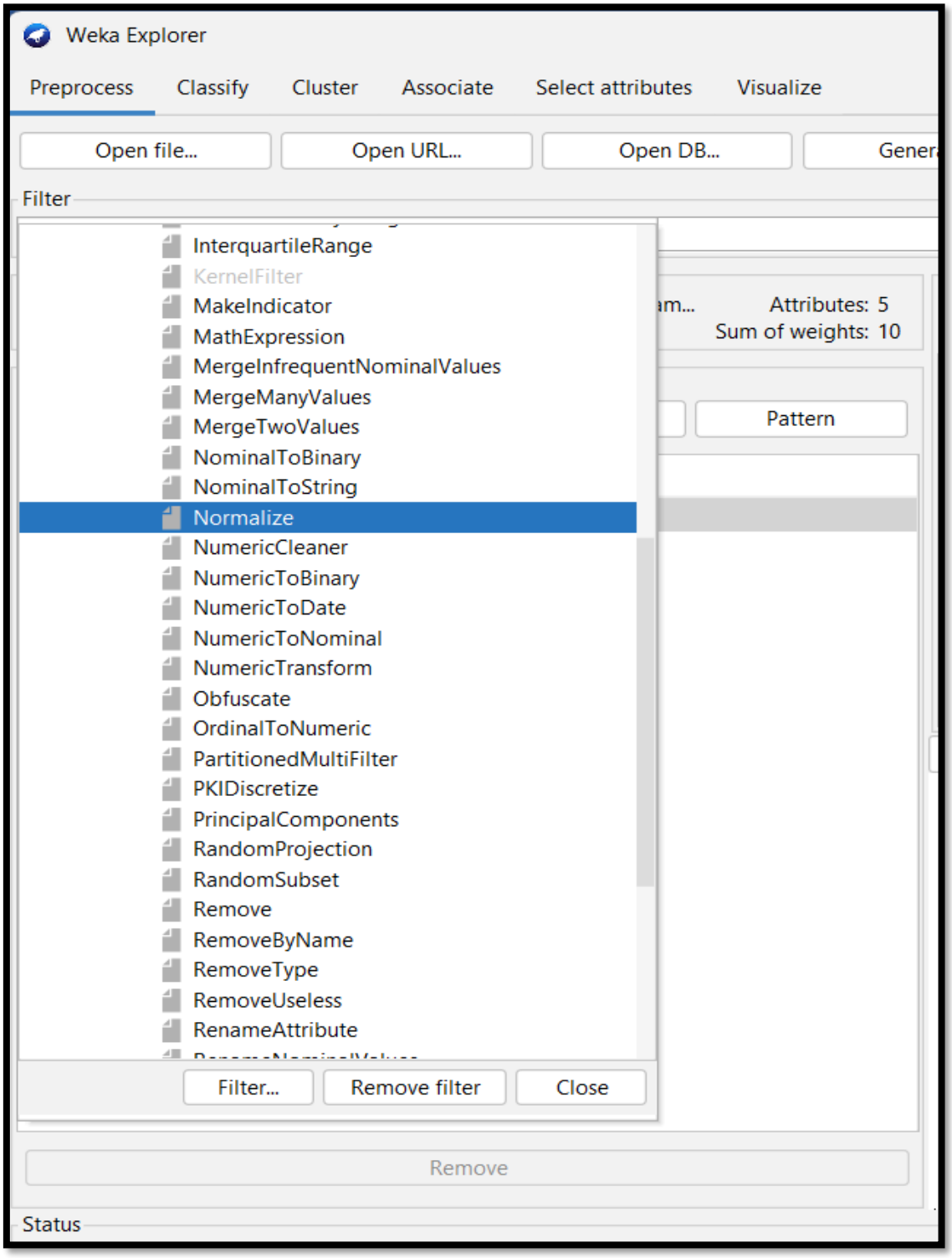
REMOVE PRE-PROCESSING TECHNIQUES

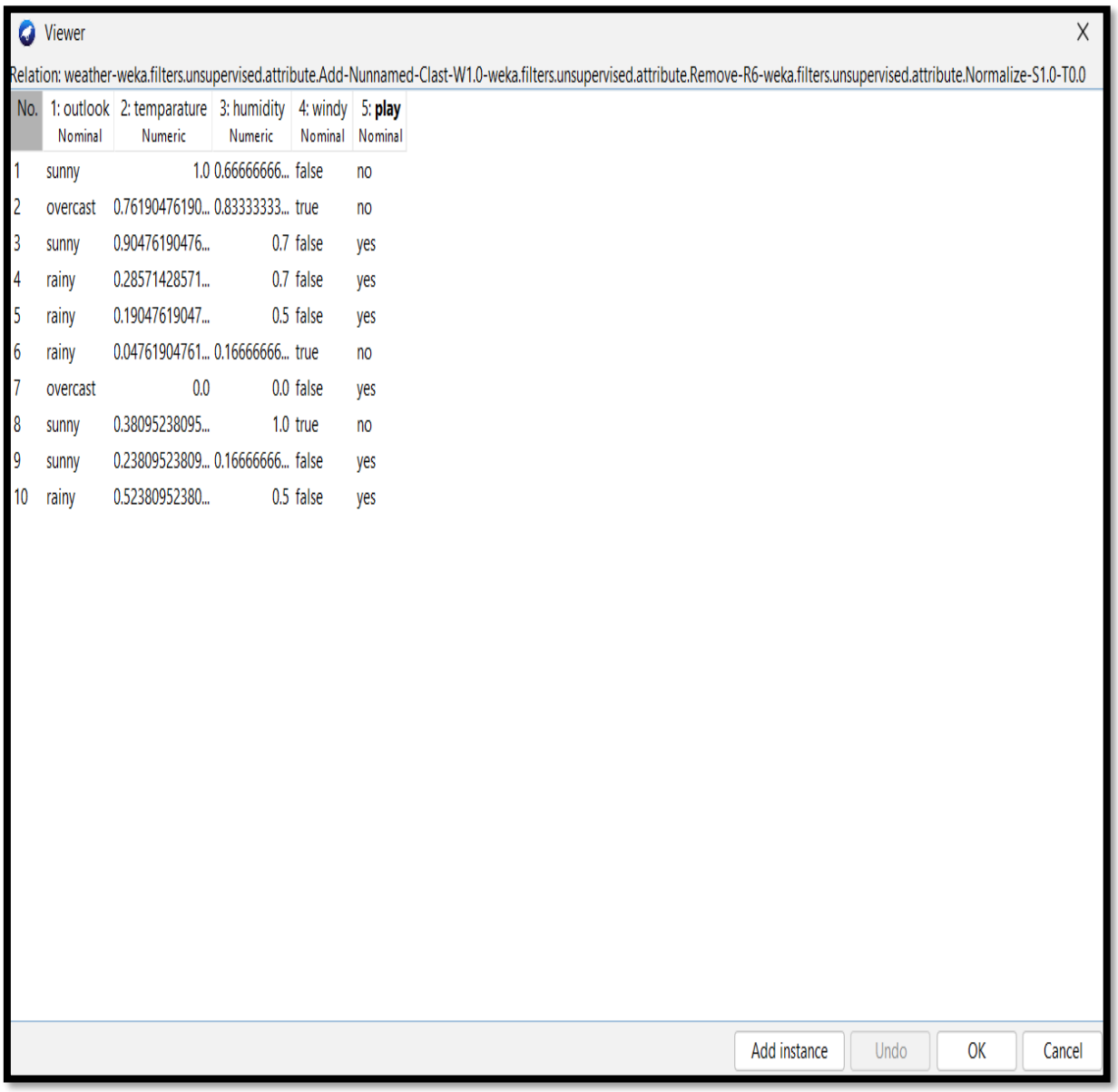



Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
Add -T NOM -N Climate -L Nominal -C last -W 1.0
Apply
Stop
Current relation
Relation: weather-weka.filters.unsupervised.attribute.Add-Nunnam...
Attributes: 5
Instances: 10
Sum of weights: 10
Attributes
All
None
Invert
Pattern
No.
Name
1 outlook
2 temparature
3 humidity
4 windy
5 play
Remove
Selected attribute
Name: outlook
Missing: 0 (0%)
Distinct: 3
Type: Nominal
Unique: 0 (0%)
No.
Label
Count
Weight
1 sunny 4 4
2 rainy 4 4
3 overcast 2 2
Class: play (Nom)
Visualize All
Status
OK
Log
x 0


Viewer
Relation: weather-weka.filters.unsupervised.attribute.Add-Nunnamed-Clast-W1.0-weka.filters.u
No.
1: outlook Nominal
2: temparature Numeric
3: humidity Numeric
4: windy Nominal
5: play Nominal
1 sunny 85.0 85.0 false no
2 overcast 80.0 90.0 true no
3 sunny 83.0 86.0 false yes
4 rainy 70.0 86.0 false yes
5 rainy 68.0 80.0 false yes
6 rainy 65.0 70.0 true no
7 overcast 64.0 65.0 false yes
8 sunny 72.0 95.0 true no
9 sunny 69.0 70.0 false yes
10 rainy 75.0 80.0 false yes
Add instance
Undo
OK
Cancel

NORMALIZING PREPROCESSING TECHNIQUE





RESULT:

This Program has been finally executed.

Practical No.-6

Aim:
Normalize Weather Table data using Knowledge Flow.

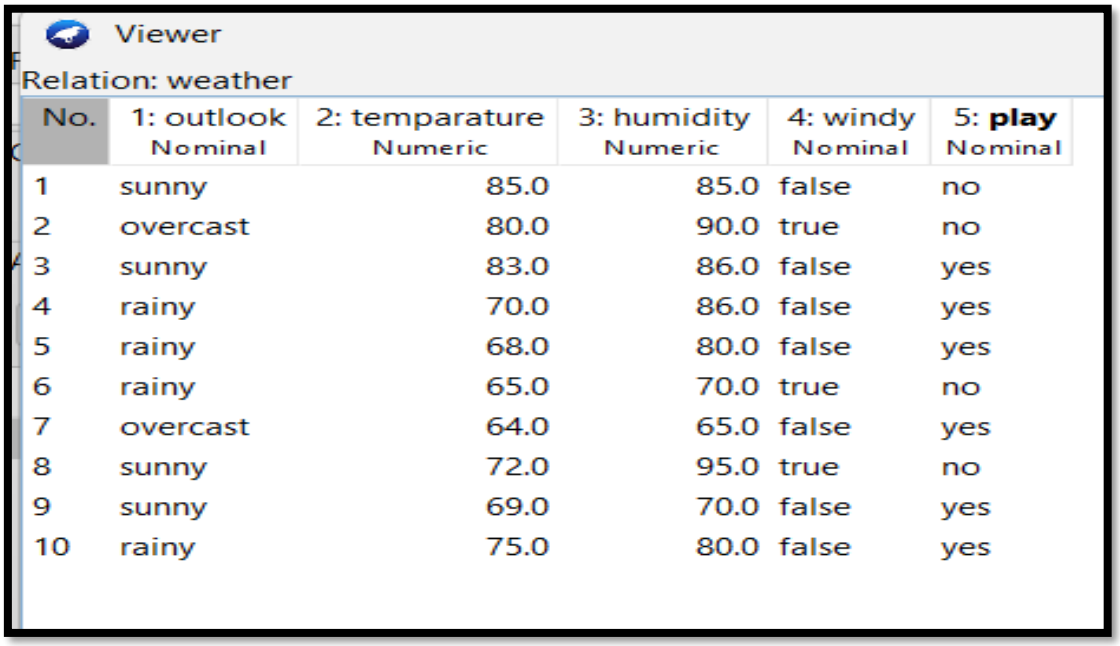
Description:
The knowledge flow provides an alternative way to the explorer as a graphical front end to WEKA's algorithm. Knowledge flow is a working progress. So, some of the functionality from explorer is not yet available. So, on the other hand there are the things that can be done in knowledge flow, but not in explorer. Knowledge flow presents a dataflow interface to WEKA. The user can select WEKA components from a toolbar placed them on a layout campus and connect them together in order to form a knowledge flow for processing and analyzing the data.

Creation of Weather Table:
Procedure:
Steps:-
1) Open Start ◊ Programs ◊ Accessories ◊ Notepad
2) Type the following training data set with the help of Notepad for Weather Table.

```
@relation weather
@attribute outlook {sunny,rainy,overcast}
@attribute temparature numeric
 @attribute humidity numeric
 @attribute windy {true,false}
@attribute play {yes,no}

@data
sunny,85.0,85.0,false,no
overcast,80.0,90.0,true,no
sunny,83.0,86.0,false,yes
rainy,70.0,86.0,false,yes
 rainy,68.0,80.0,false,yes
rainy,65.0,70.0,true,no
```

overcast,64.0,65.0,false,yes
sunny,72.0,95.0,true,no
sunny,69.0,70.0,false,yes
rainy,75.0,80.0,false,yes

3) After that the file is saved with .arff file format.
4) Minimize the arff file and then open Start ◊ Programs ◊ weka-3-4.
5) Click on weka-3-4, then Weka dialog box is displayed on the screen.
6) In that dialog box there are four modes, click on explorer.
7) Explorer shows many options. In that click on 'open file' and select the arff file
8) Click on edit button which shows Weather table on weka


Viewer

Relation: weather

No.	1: outlook Nominal	2: temparature Numeric	3: humidity Numeric	4: windy Nominal	5: **play** Nominal
1	sunny	85.0	85.0	false	no
2	overcast	80.0	90.0	true	no
3	sunny	83.0	86.0	false	yes
4	rainy	70.0	86.0	false	yes
5	rainy	68.0	80.0	false	yes
6	rainy	65.0	70.0	true	no
7	overcast	64.0	65.0	false	yes
8	sunny	72.0	95.0	true	no
9	sunny	69.0	70.0	false	yes
10	rainy	75.0	80.0	false	yes

Procedure for Knowledge Flow:
1) Open Start ◊ Programs ◊ Weka-3-4 ◊ Weka-3-4
2) Open the Knowledge Flow.
3) Select the Data Source component and add Arff Loader into the knowledge layout canvas.
4) Select the Filters component and add Attribute Selection and Normalize into the knowledge layout canvas.
5) Select the Data Sinks component and add Arff Saver into the knowledge layout canvas.

6) Right click on Arff Loader and select Configure option then the new window will be opened and select Weather.arff

7) Right click on Arff Loader and select Dataset option then establish a link between Arff Loader and Attribute Selection.

8) Right click on Attribute Selection and select Dataset option then establish a link between Attribute Selection and Normalize.

9) Right click on Attribute Selection and select Configure option and choose the best attribute for Weather data.

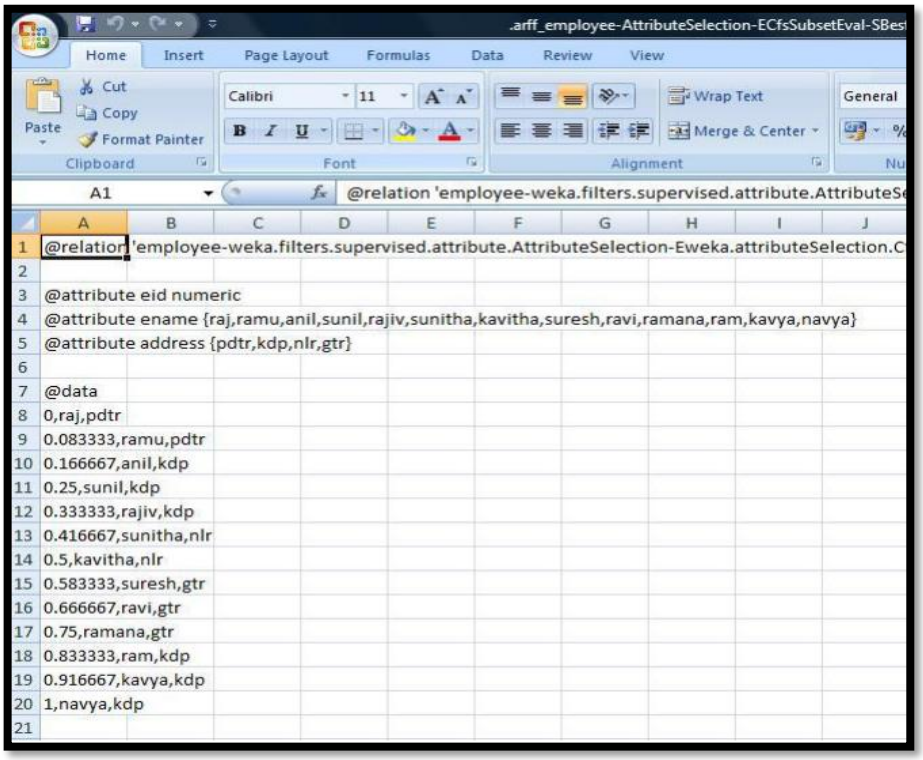
10) Right click on Normalize and select Dataset option then establish a link between Normalize and Arff Saver.

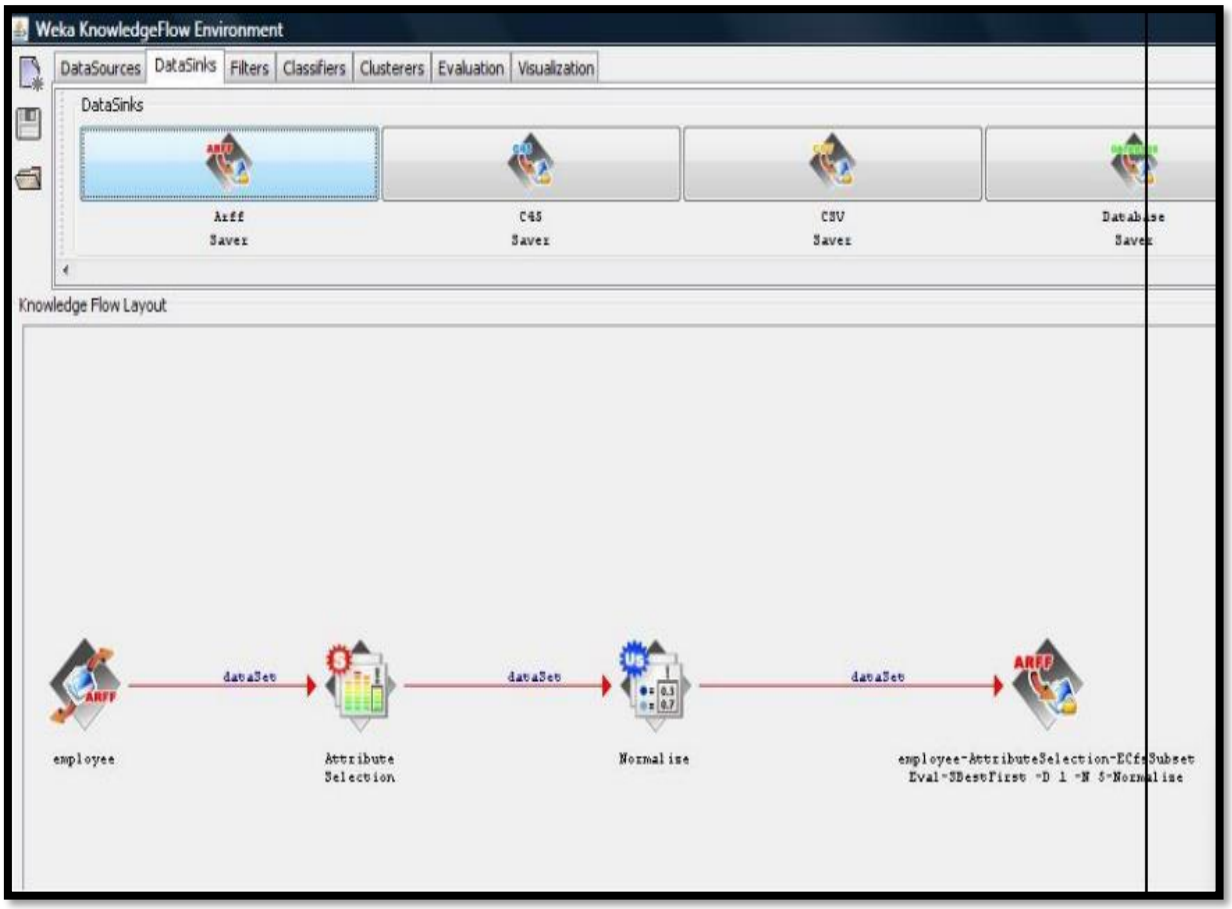
11) Right click on Arff Saver and select Configure option then new window will be opened and set the path, enter .arff in look in dialog box to save normalize data.

12) Right click on Arff Loader and click on Start Loading option then everything will be executed one by one.

13) Check whether output is created or not by selecting the preferred path.

14) Rename the data name as a.arff 15) Double click on a.arff then automatically the output will be opened in MS-Excel.




Weka KnowledgeFlow Environment
DataSources
DataSinks
Filters
Classifiers
Clusterers
Evaluation
Visualization
DataSinks
Arff Saver
C45 Saver
CSV Saver
Database Saver
Knowledge Flow Layout
dataSet
dataSet
dataSet
employee
Attribute Selection
Normalize
employee-AttributeSelection-ECfsSubset Eval-SBestFirst -D 1 -N 5-Normalize

Practical No 7:

Aim:
Finding Association Rules for Buying data.

Description:
In data mining, association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases. It can be described as analyzing and presenting strong rules discovered in databases using different measures of interestingness. In market basket analysis association rules are used and they are also employed in many application areas including Web usage mining, intrusion detection and bioinformatics.

Creation of Buying Table:

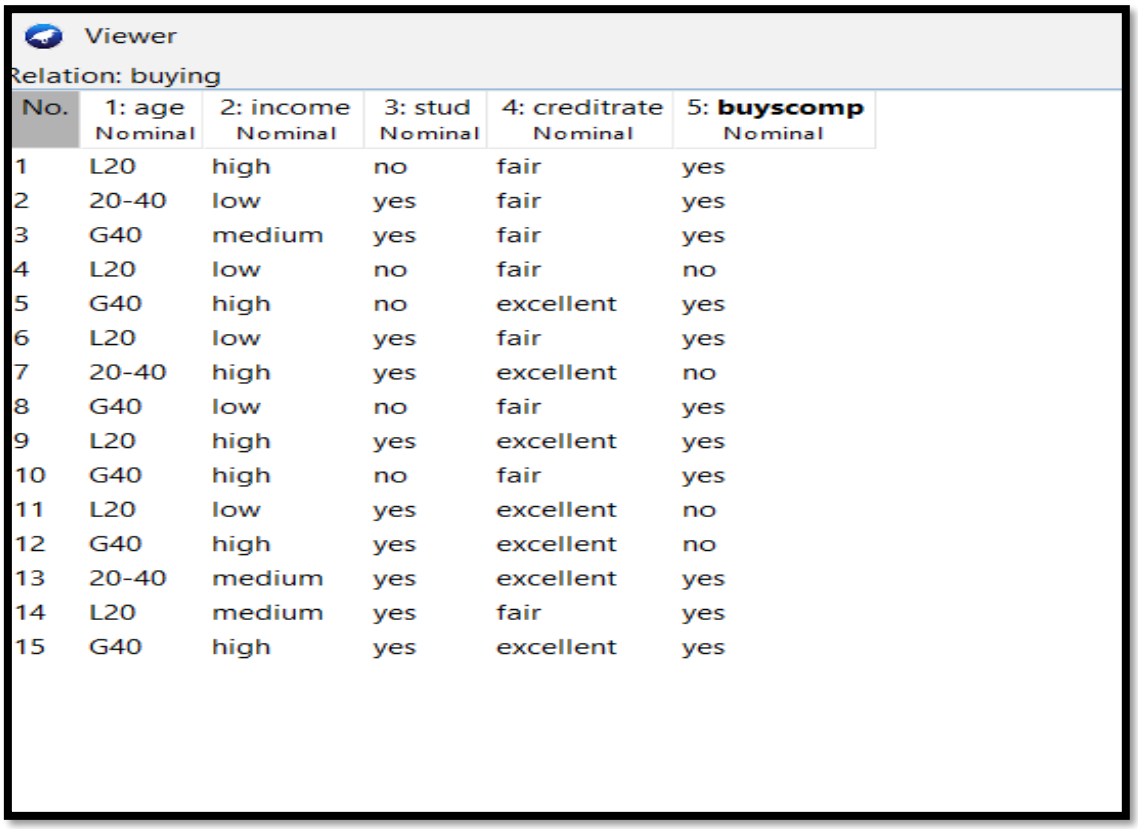
Procedure:

1) Open Start ◊ Programs ◊ Accessories ◊ Notepad
2) Type the following training data set with the help of Notepad for Buying Table.

```
@relation buying
@attribute age {L20,20-40,G40}
@attribute income {high,medium,low}
@attribute stud {yes,no}
@attribute creditrate {fair,excellent}
@attribute buyscomp {yes,no}
@data
L20,high,no,fair,yes
20-40,low,yes,fair,yes
G40,medium,yes,fair,yes
L20,low,no,fair,no
G40,high,no,excellent,yes
L20,low,yes,fair,yes
20-40,high,yes,excellent,no
G40,low,no,fair,yes
L20,high,yes,excellent,yes
```

G40,high,no,fair,yes
L20,low,yes,excellent,no
G40,high,yes,excellent,no
20-40,medium,yes,excellent,yes
L20,medium,yes,fair,yes
G40,high,yes,excellent,yes

3) After that the file is saved with .arff file format.
4) Minimize the arff file and then open Start ◊ Programs ◊ weka-3-4.
5) Click on weka-3-4, then Weka dialog box is displayed on the screen.
6) In that dialog box there are four modes, click on explorer.
7) Explorer shows many options. In that click on 'open file' and select the arff file
8) Click on edit button which shows buying table on weka.


Viewer

Relation: buying

No.	1: age Nominal	2: income Nominal	3: stud Nominal	4: creditrate Nominal	5: **buyscomp** Nominal
1	L20	high	no	fair	yes
2	20-40	low	yes	fair	yes
3	G40	medium	yes	fair	yes
4	L20	low	no	fair	no
5	G40	high	no	excellent	yes
6	L20	low	yes	fair	yes
7	20-40	high	yes	excellent	no
8	G40	low	no	fair	yes
9	L20	high	yes	excellent	yes
10	G40	high	no	fair	yes
11	L20	low	yes	excellent	no
12	G40	high	yes	excellent	no
13	20-40	medium	yes	excellent	yes
14	L20	medium	yes	fair	yes
15	G40	high	yes	excellent	yes

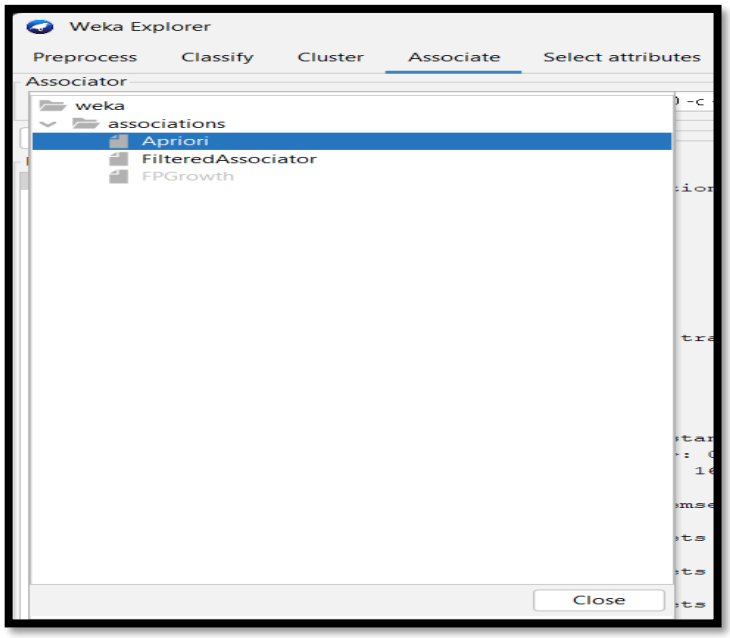
Procedure for Association Rules:
1) Open Start ◊ Programs ◊ Weka-3-4 ◊ Weka-3-4
2) Open explorer.

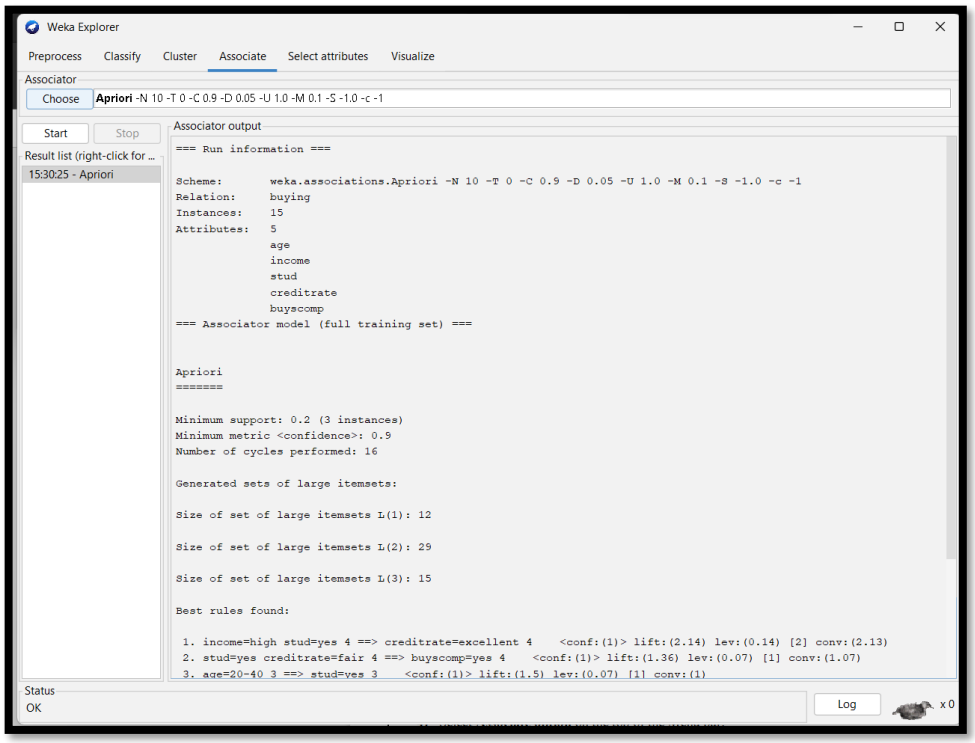
3) Click on open file and select buying.arff

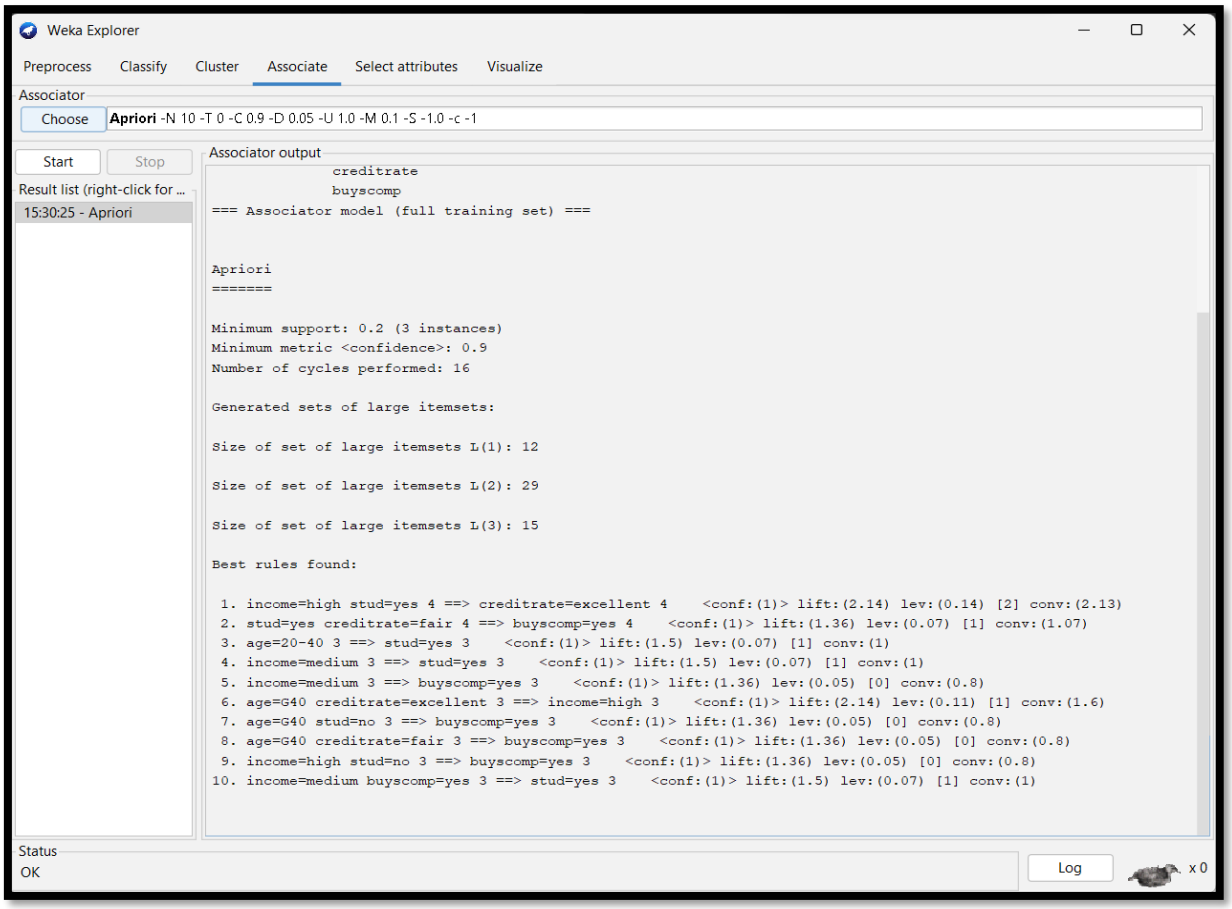
4) Select Associate option on the top of the Menu bar.

5) Select Choose button and then click on Apriori Algorithm.

6) Click on Start button and output will be displayed on the right side of the window






Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Associator
Choose
Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1
Start
Stop
Result list (right-click for ...
15:30:25 - Apriori
Associator output
creditrate
buyscomp
=== Associator model (full training set) ===
Apriori
=======
Minimum support: 0.2 (3 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 16
Generated sets of large itemsets:
Size of set of large itemsets L(1): 12
Size of set of large itemsets L(2): 29
Size of set of large itemsets L(3): 15
Best rules found:
1. income=high stud=yes 4 ==> creditrate=excellent 4 <conf:(1)> lift:(2.14) lev:(0.14) [2] conv:(2.13)
2. stud=yes creditrate=fair 4 ==> buyscomp=yes 4 <conf:(1)> lift:(1.36) lev:(0.07) [1] conv:(1.07)
3. age=20-40 3 ==> stud=yes 3 <conf:(1)> lift:(1.5) lev:(0.07) [1] conv:(1)
4. income=medium 3 ==> stud=yes 3 <conf:(1)> lift:(1.5) lev:(0.07) [1] conv:(1)
5. income=medium 3 ==> buyscomp=yes 3 <conf:(1)> lift:(1.36) lev:(0.05) [0] conv:(0.8)
6. age=G40 creditrate=excellent 3 ==> income=high 3 <conf:(1)> lift:(2.14) lev:(0.11) [1] conv:(1.6)
7. age=G40 stud=no 3 ==> buyscomp=yes 3 <conf:(1)> lift:(1.36) lev:(0.05) [0] conv:(0.8)
8. age=G40 creditrate=fair 3 ==> buyscomp=yes 3 <conf:(1)> lift:(1.36) lev:(0.05) [0] conv:(0.8)
9. income=high stud=no 3 ==> buyscomp=yes 3 <conf:(1)> lift:(1.36) lev:(0.05) [0] conv:(0.8)
10. income=medium buyscomp=yes 3 ==> stud=yes 3 <conf:(1)> lift:(1.5) lev:(0.07) [1] conv:(1)
Status
OK
Log
x 0

Practical No 8:

Aim:
To Construct Decision Tree for Weather data and classify it.

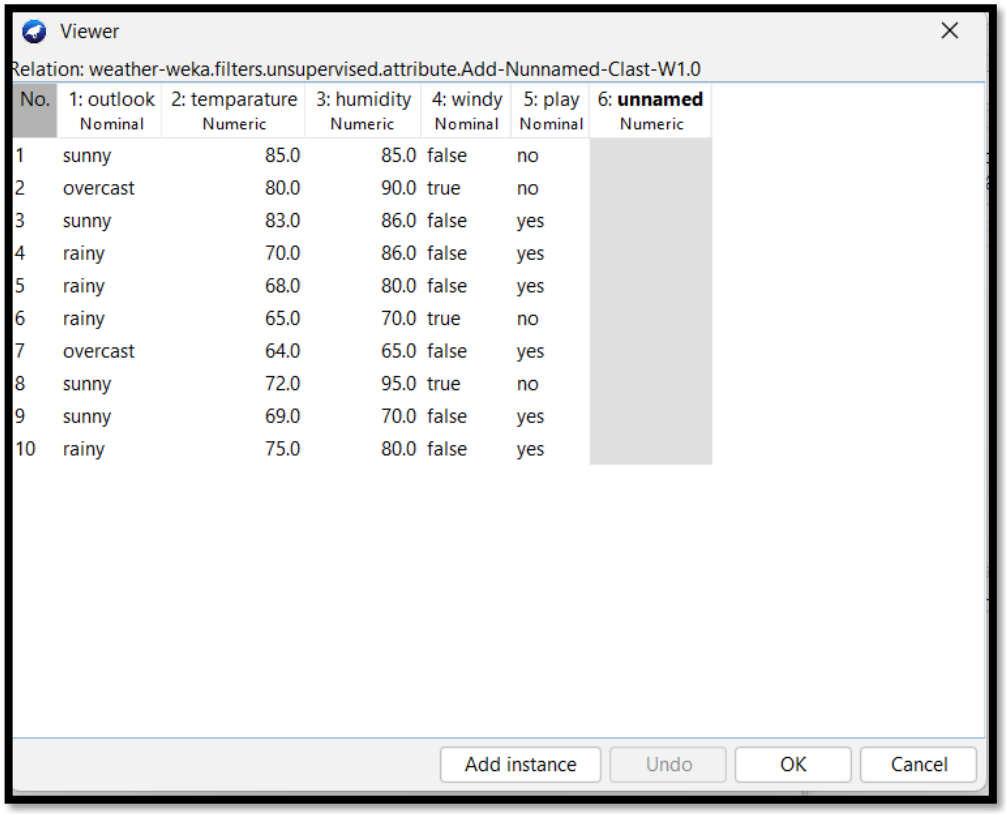
Description:
Classification & Prediction: Classification is the process for finding a model that describes the data values and concepts for the purpose of Prediction.
Decision Tree: A decision Tree is a classification scheme to generate a tree consisting of root node, internal nodes and external nodes. Root nodes representing the attributes. Internal nodes are also the attributes. External nodes are the classes and each branch represents the values of the attributes Decision Tree also contains set of rules for a given data set; there are two subsets in Decision Tree. One is a Training data set and second one is a Testing data set. Training data set is previously classified data. Testing data set is newly generated data.

Creation of Weather Table:
Procedure:
1) Open Start ◊ Programs ◊ Accessories ◊ Notepad
2) Type the following training data set with the help of Notepad for Weather Table.

```
@relation weather
@attribute outlook {sunny,rainy,overcast}
@attribute temparature numeric
 @attribute humidity numeric
 @attribute windy {true,false}
@attribute play {yes,no}

@data
sunny,85.0,85.0,false,no
overcast,80.0,90.0,true,no
sunny,83.0,86.0,false,yes
```

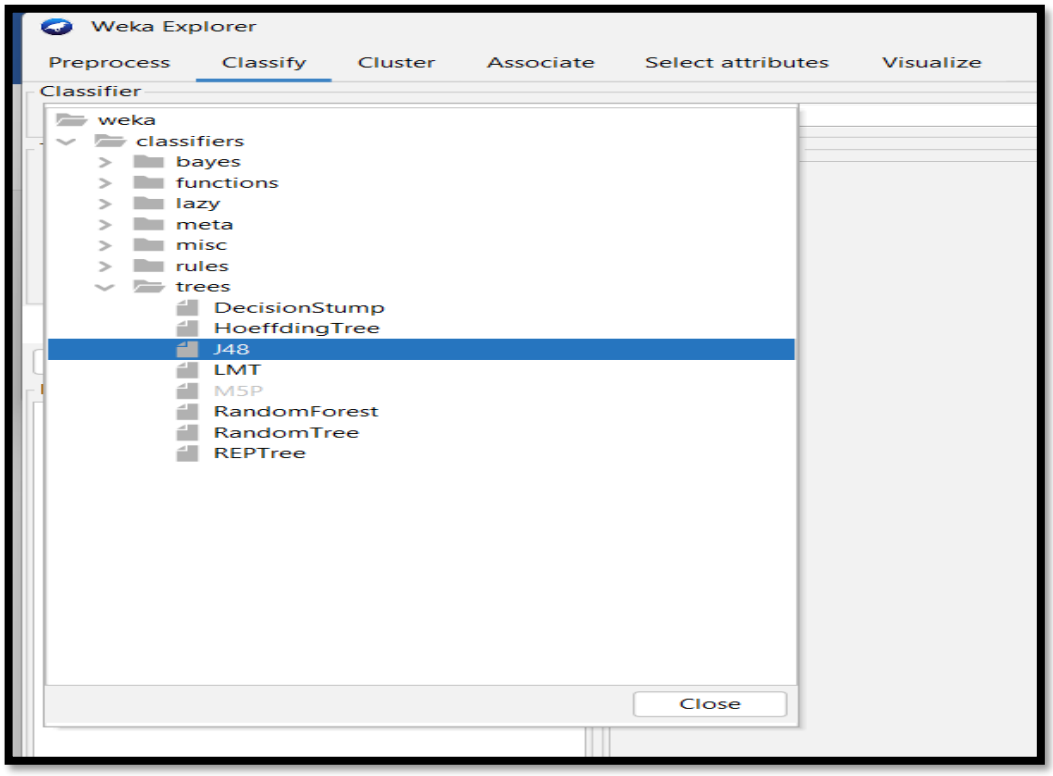
rainy,70.0,86.0,false,yes
rainy,68.0,80.0,false,yes
rainy,65.0,70.0,true,no
overcast,64.0,65.0,false,yes
sunny,72.0,95.0,true,no
sunny,69.0,70.0,false,yes
rainy,75.0,80.0,false,yes

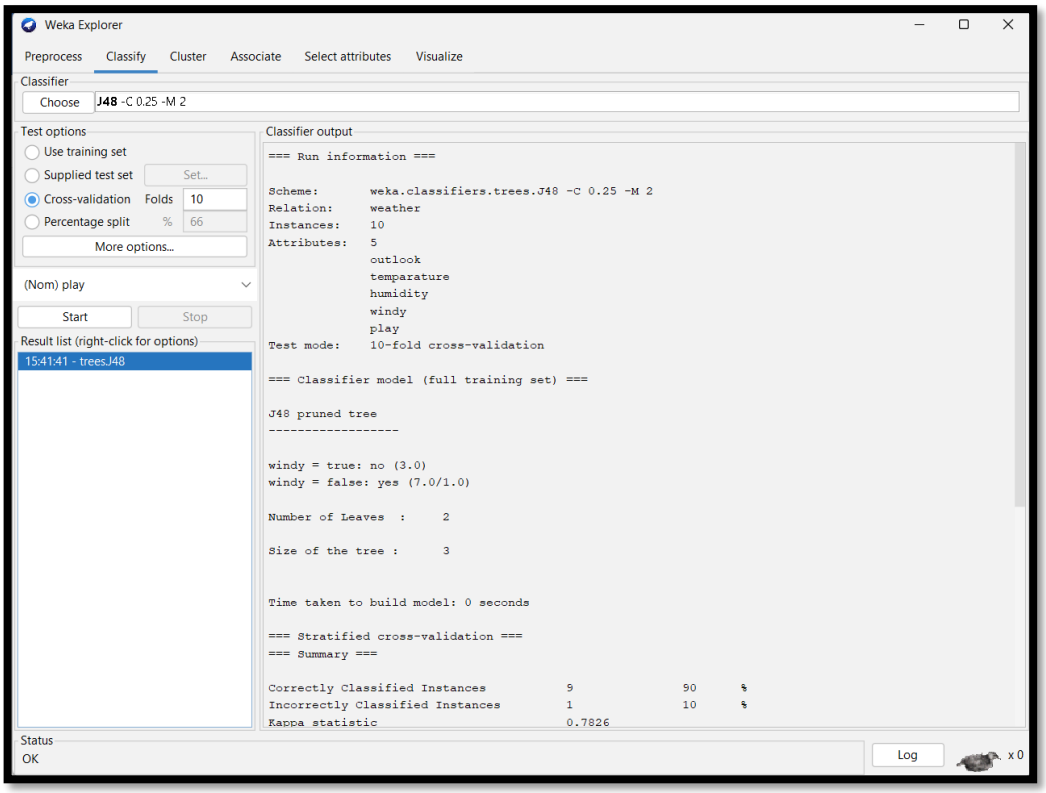
3) After that the file is saved with .arff file format.
4) Minimize the arff file and then open Start ◊ Programs ◊ weka-3-4. 32
5) Click on weka-3-4, then Weka dialog box is displayed on the screen.
6) In that dialog box there are four modes, click on explorer.
7) Explorer shows many options. In that click on 'open file' and select the arff file
8) Click on edit button which shows weather table on weka.



Procedure for Decision Trees:

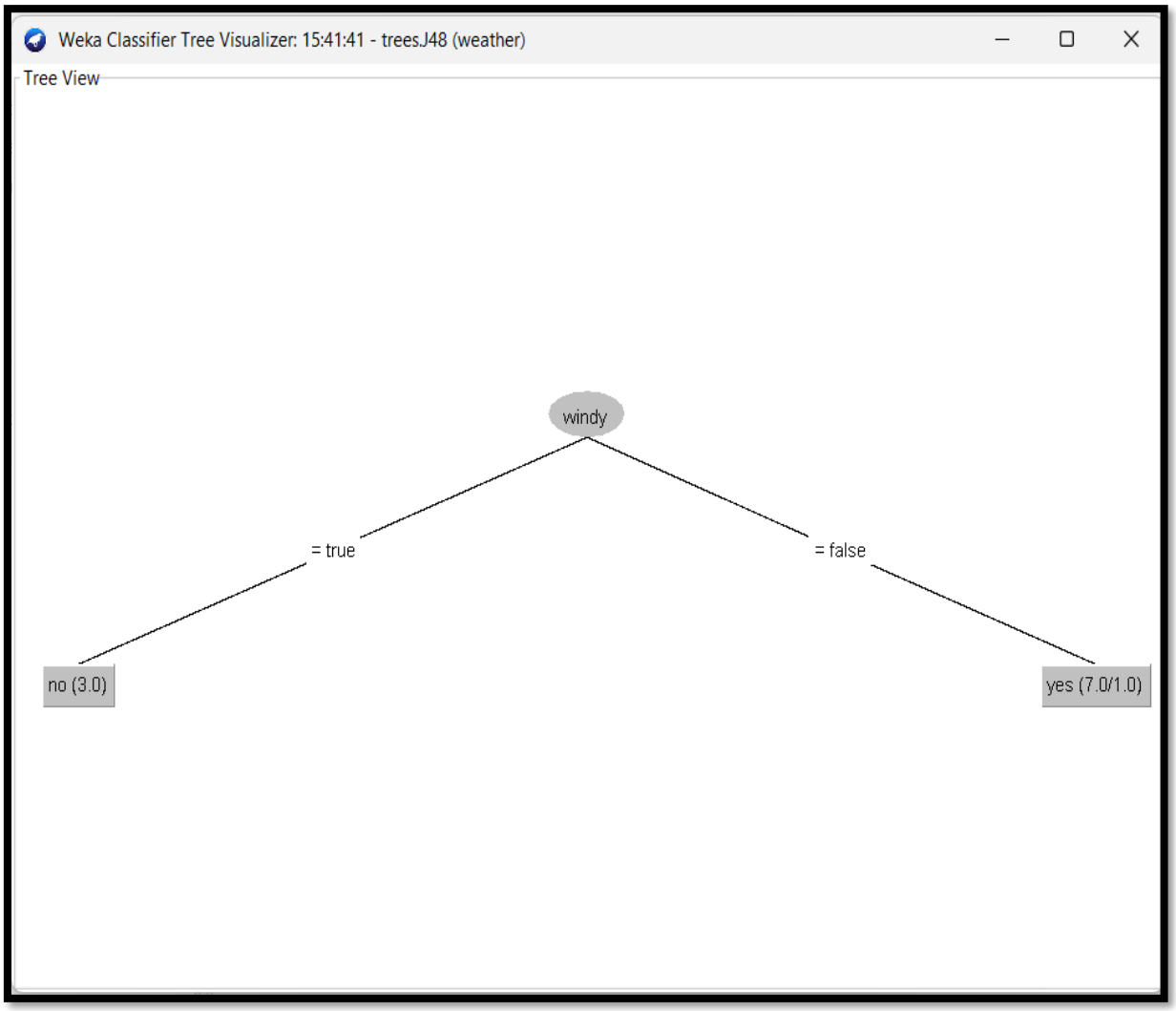
1) Open Start ◊ Programs ◊ Weka-3-4 ◊ Weka-3-4
2) Open explorer.
3) Click on open file and select weather.arff
4) Select Classifier option on the top of the Menu bar.
5) Select Choose button and click on Tree option.
6) Click on J48.
7) Click on Start button and output will be displayed on the right side of the window.
8) Select the result list and right click on result list and select Visualize Tree option.
9) Then Decision Tree will be displayed on new window.






```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances           9               90      %
Incorrectly Classified Instances         1               10      %
Kappa statistic                          0.7826
Mean absolute error                      0.2
Root mean squared error                  0.3416
Relative absolute error                 37.931  %
Root relative squared error             63.8746 %
Total Number of Instances               10

=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000     0.250     0.857       1.000   0.923      0.802    0.750     0.857     yes
                 0.750     0.000     1.000       0.750   0.857      0.802    0.750     0.850     no
Weighted Avg.    0.900     0.150     0.914       0.900   0.897      0.802    0.750     0.854

=== Confusion Matrix ===

 a b   <-- classified as
 6 0 | a = yes
 1 3 | b = no
```

Log x



Result:

This program has been successfully executed.

Practical No 9:

Aim:

Write a procedure for Visualization for Weather Table.

Description:

This program calculates and has comparisons on the data set selection of attributes and methods of manipulations have been chosen. The Visualization can be shown in a 2-D representation of the information.

Creation of Weather Table: Procedure:

1) Open Start ◊ Programs ◊ Accessories ◊ Notepad

2) Type the following training data set with the help of Notepad for Weather Table.

```
@relation weather
@attribute outlook {sunny, rainy, overcast}
@attribute temperature numeric
@attribute humidity numeric
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}
@data
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes
rainy,65,70,TRUE,no
overcast,64,65,TRUE,yes
sunny,72,95,FALSE,no
sunny,69,70,FALSE,yes
rainy,75,80,FALSE,yes
sunny,75,70,TRUE,yes
overcast,72,90,TRUE,yes
overcast,81,75,FALSE,yes
rainy,71,91,TRUE,no
```

3) After that the file is saved with .arff file format.
4) Minimize the arff file and then open Start ◊ Programs ◊ weka-3-4.
5) Click on weka-3-4, then Weka dialog box is displayed on the screen.
6) In that dialog box there are four modes, click on explorer.
7) Explorer shows many options. In that click on 'open file' and select the arff file
8) Click on edit button which shows weather table on weka.

Viewer

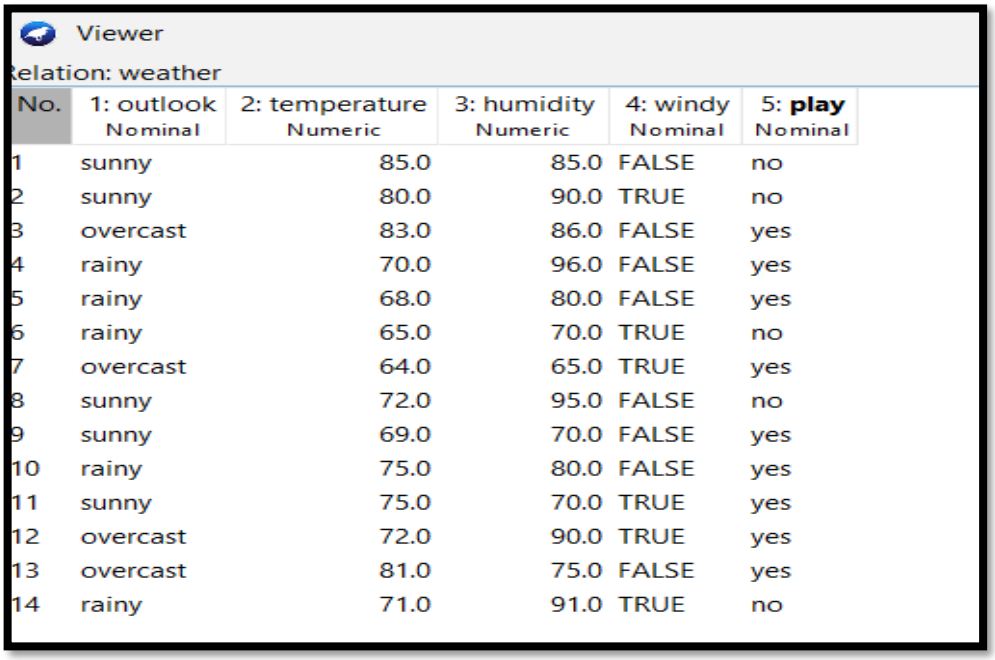
Relation: weather

No.	1: outlook Nominal	2: temperature Numeric	3: humidity Numeric	4: windy Nominal	5: **play** Nominal
1	sunny	85.0	85.0	FALSE	no
2	sunny	80.0	90.0	TRUE	no
3	overcast	83.0	86.0	FALSE	yes
4	rainy	70.0	96.0	FALSE	yes
5	rainy	68.0	80.0	FALSE	yes
6	rainy	65.0	70.0	TRUE	no
7	overcast	64.0	65.0	TRUE	yes
8	sunny	72.0	95.0	FALSE	no
9	sunny	69.0	70.0	FALSE	yes
10	rainy	75.0	80.0	FALSE	yes
11	sunny	75.0	70.0	TRUE	yes
12	overcast	72.0	90.0	TRUE	yes
13	overcast	81.0	75.0	FALSE	yes
14	rainy	71.0	91.0	TRUE	no

Procedure:

1) Open Start ◊ Programs ◊ Weka-3-4 ◊ Weka-3-4
2) Open the explorer and click on Preprocess, then a new window will appear. In that window select weather.arff file then the data will be displayed.
3) After that click on the Visualize tab on the top of the Menu bar.
4) When we select Visualize tab then Plot Matrix is displayed on the screen.
5) After that we select the Select Attribute button, then select Outlook attribute and clock OK.
6) Click on the Update button to display the output.

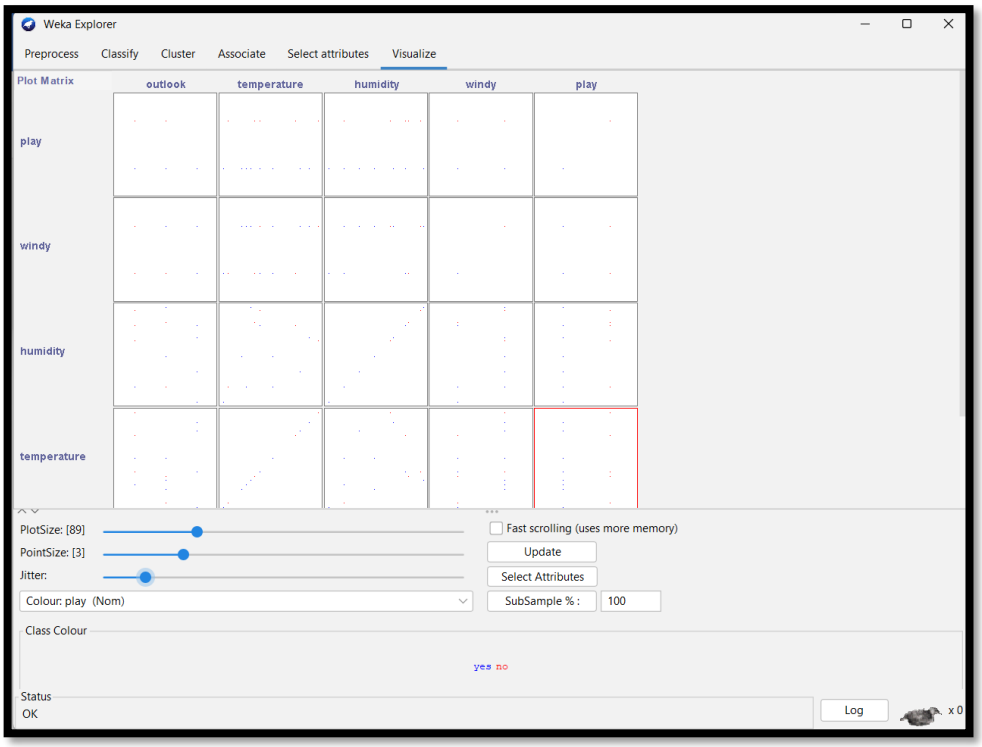
7) After that select the Select Attribute button and select Temperature attribute and then click OK.
8) Increase the Plot Size and Point Size.
9) Click on the Update button to display the output.
10) After that we select the Select Attribute button, then select Humidity attribute and clock OK.
11) Click on the Update button to display the output.
12) After that select the Select Attribute button and select Windy attribute and then click OK.
13) Increase the Jitter Size.
14) Click on the Update button to display the output.
15) After that we select the Select Attribute button, then select Play attribute and clock OK.
16) Click on the Update button to display the output.

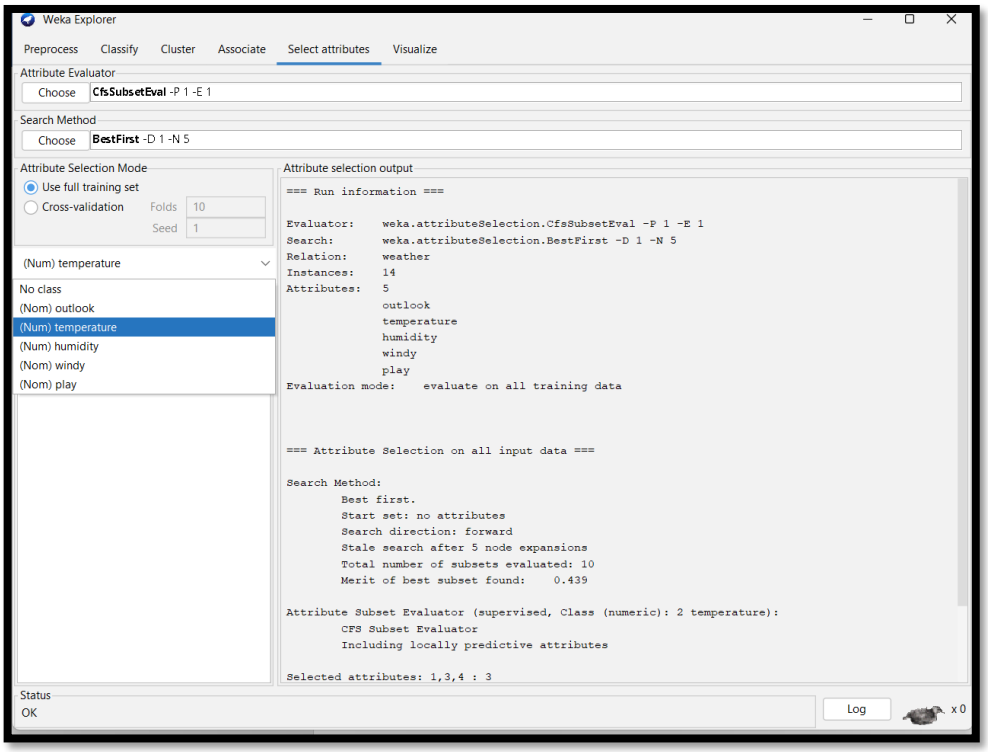


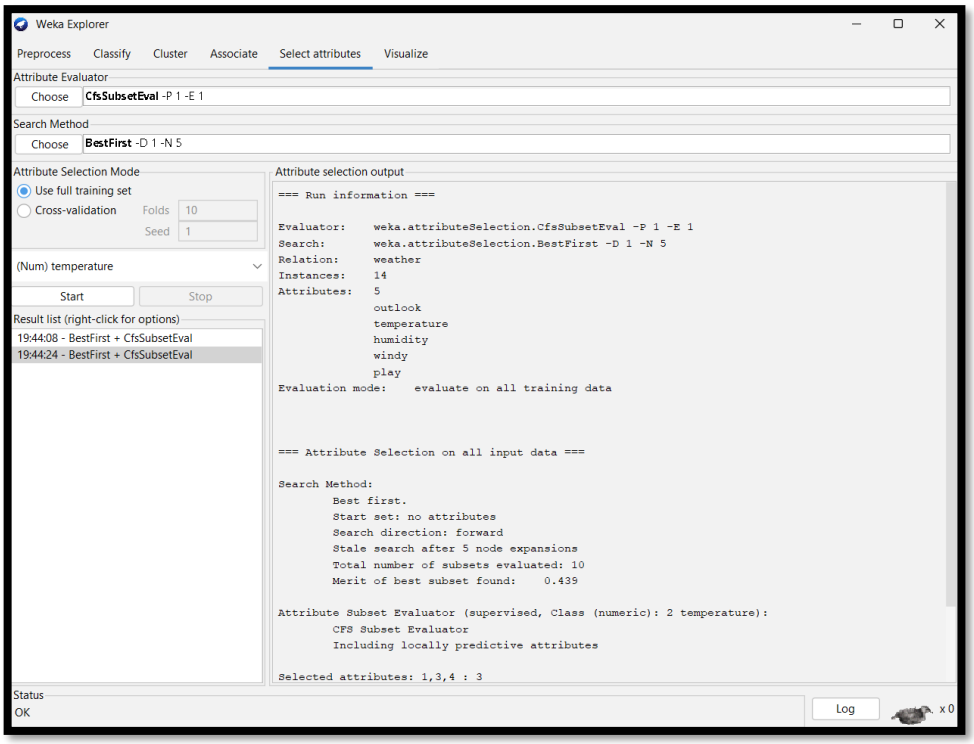
Viewer

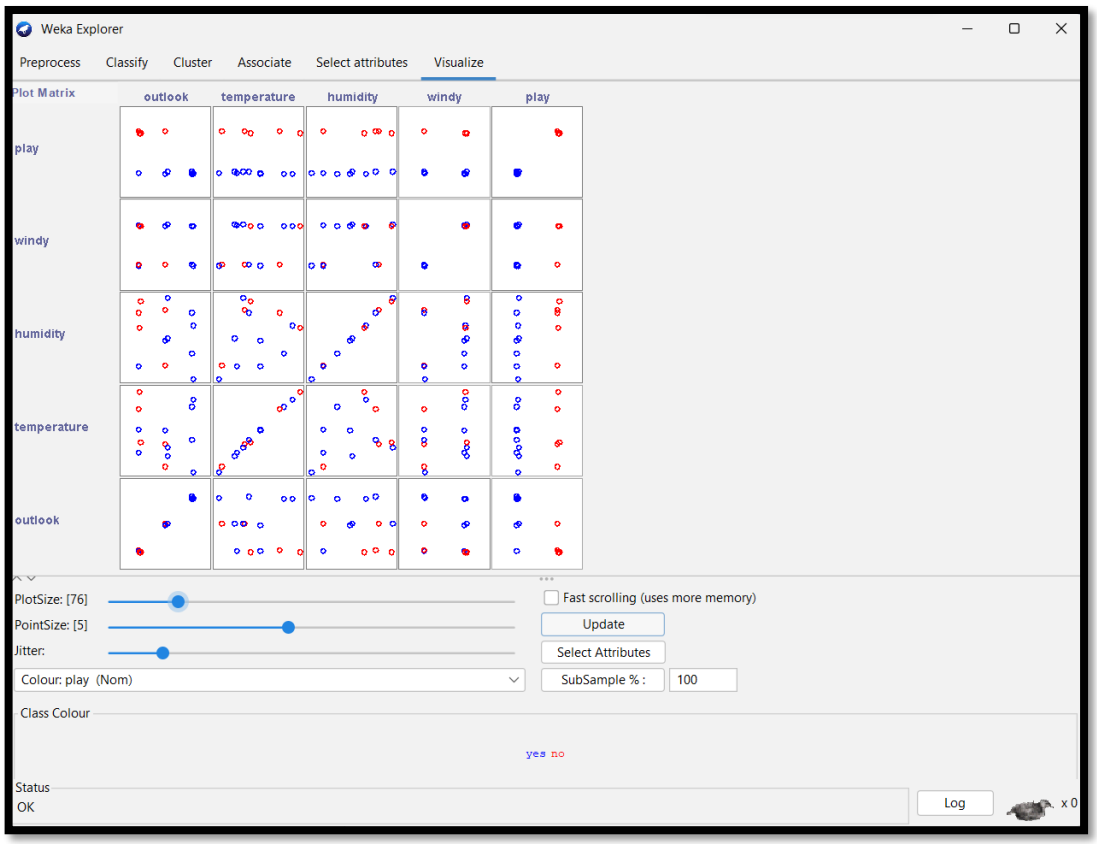
Relation: weather

No.	1: outlook Nominal	2: temperature Numeric	3: humidity Numeric	4: windy Nominal	5: **play** Nominal
1	sunny	85.0	85.0	FALSE	no
2	sunny	80.0	90.0	TRUE	no
3	overcast	83.0	86.0	FALSE	yes
4	rainy	70.0	96.0	FALSE	yes
5	rainy	68.0	80.0	FALSE	yes
6	rainy	65.0	70.0	TRUE	no
7	overcast	64.0	65.0	TRUE	yes
8	sunny	72.0	95.0	FALSE	no
9	sunny	69.0	70.0	FALSE	yes
10	rainy	75.0	80.0	FALSE	yes
11	sunny	75.0	70.0	TRUE	yes
12	overcast	72.0	90.0	TRUE	yes
13	overcast	81.0	75.0	FALSE	yes
14	rainy	71.0	91.0	TRUE	no


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Plot Matrix
outlook
temperature
humidity
windy
play
play
windy
humidity
temperature
PlotSize: [89]
PointSize: [3]
Jitter:
Colour: play (Nom)
Fast scrolling (uses more memory)
Update
Select Attributes
SubSample % :
100
Class Colour
yes no
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Attribute Evaluator
Choose
CfsSubsetEval -P 1 -E 1
Search Method
Choose
BestFirst -D 1 -N 5
Attribute Selection Mode
Use full training set
Cross-validation
Folds
10
Seed
1
(Num) temperature
No class
(Nom) outlook
(Num) temperature
(Num) humidity
(Nom) windy
(Nom) play
Attribute selection output
=== Run information ===
Evaluator: weka.attributeSelection.CfsSubsetEval -P 1 -E 1
Search: weka.attributeSelection.BestFirst -D 1 -N 5
Relation: weather
Instances: 14
Attributes: 5
outlook
temperature
humidity
windy
play
Evaluation mode: evaluate on all training data
=== Attribute Selection on all input data ===
Search Method:
Best first.
Start set: no attributes
Search direction: forward
Stale search after 5 node expansions
Total number of subsets evaluated: 10
Merit of best subset found: 0.439
Attribute Subset Evaluator (supervised, Class (numeric): 2 temperature):
CFS Subset Evaluator
Including locally predictive attributes
Selected attributes: 1,3,4 : 3
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Attribute Evaluator
Choose
CfsSubsetEval -P 1 -E 1
Search Method
Choose
BestFirst -D 1 -N 5
Attribute Selection Mode
Use full training set
Cross-validation
Folds
10
Seed
1
(Num) temperature
Start
Stop
Result list (right-click for options)
19:44:08 - BestFirst + CfsSubsetEval
19:44:24 - BestFirst + CfsSubsetEval
Attribute selection output
=== Run information ===
Evaluator: weka.attributeSelection.CfsSubsetEval -P 1 -E 1
Search: weka.attributeSelection.BestFirst -D 1 -N 5
Relation: weather
Instances: 14
Attributes: 5
outlook
temperature
humidity
windy
play
Evaluation mode: evaluate on all training data
=== Attribute Selection on all input data ===
Search Method:
Best first.
Start set: no attributes
Search direction: forward
Stale search after 5 node expansions
Total number of subsets evaluated: 10
Merit of best subset found: 0.439
Attribute Subset Evaluator (supervised, Class (numeric): 2 temperature):
CFS Subset Evaluator
Including locally predictive attributes
Selected attributes: 1,3,4 : 3
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Plot Matrix
outlook
temperature
humidity
windy
play
play
windy
humidity
temperature
outlook
PlotSize: [76]
PointSize: [5]
Jitter:
Fast scrolling (uses more memory)
Update
Select Attributes
Colour: play (Nom)
SubSample % :
100
Class Colour
yes no
Status
OK
Log
x 0

Practical No 10:

Aim:
Write a procedure for cross-validation using J48 Algorithm for weather table.

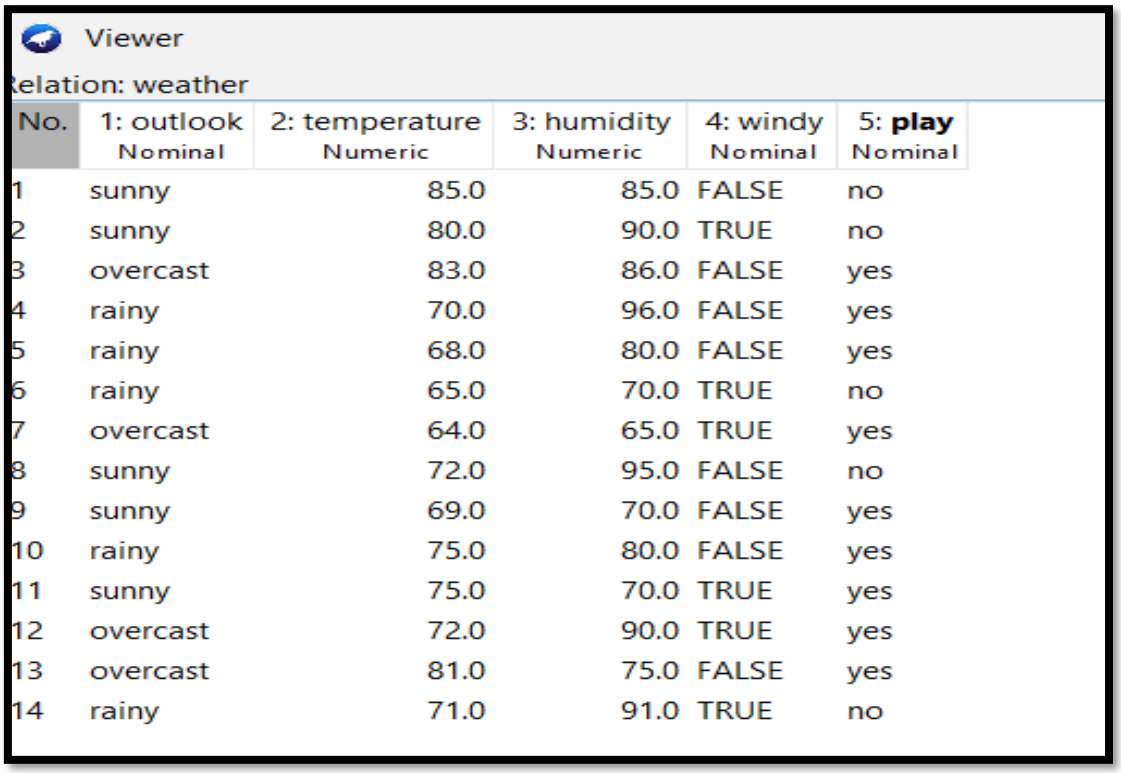
Description:
Cross-validation, sometimes called rotation estimation, is a technique for assessing how the results of a statistical analysis will generalize to an independent data set. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice. One round of cross-validation involves partitioning a sample of data into complementary subsets, performing the analysis on one subset (called the training set), and validating the analysis on the other subset (called the validation set or testing set).

Creation of Weather Table: Procedure:
1) Open Start ◊ Programs ◊ Accessories ◊ Notepad
2) Type the following training data set with the help of Notepad for Weather Table.

```
@relation weather
@attribute outlook {sunny, rainy, overcast}
@attribute temperature numeric
@attribute humidity numeric
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}
@data
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes
rainy,65,70,TRUE,no
overcast,64,65,TRUE,yes
sunny,72,95,FALSE,no
```

sunny,69,70,FALSE,yes
rainy,75,80,FALSE,yes
sunny,75,70,TRUE,yes
overcast,72,90,TRUE,yes
overcast,81,75,FALSE,yes
rainy,71,91,TRUE,no

3) After that the file is saved with .arff file format.
4) Minimize the arff file and then open Start ◊ Programs ◊ weka-3-4.
5) Click on weka-3-4, then Weka dialog box is displayed on the screen.
6) In that dialog box there are four modes, click on explorer.
7) Explorer shows many options. In that click on 'open file' and select the arff file
8) Click on edit button which shows weather table on weka.



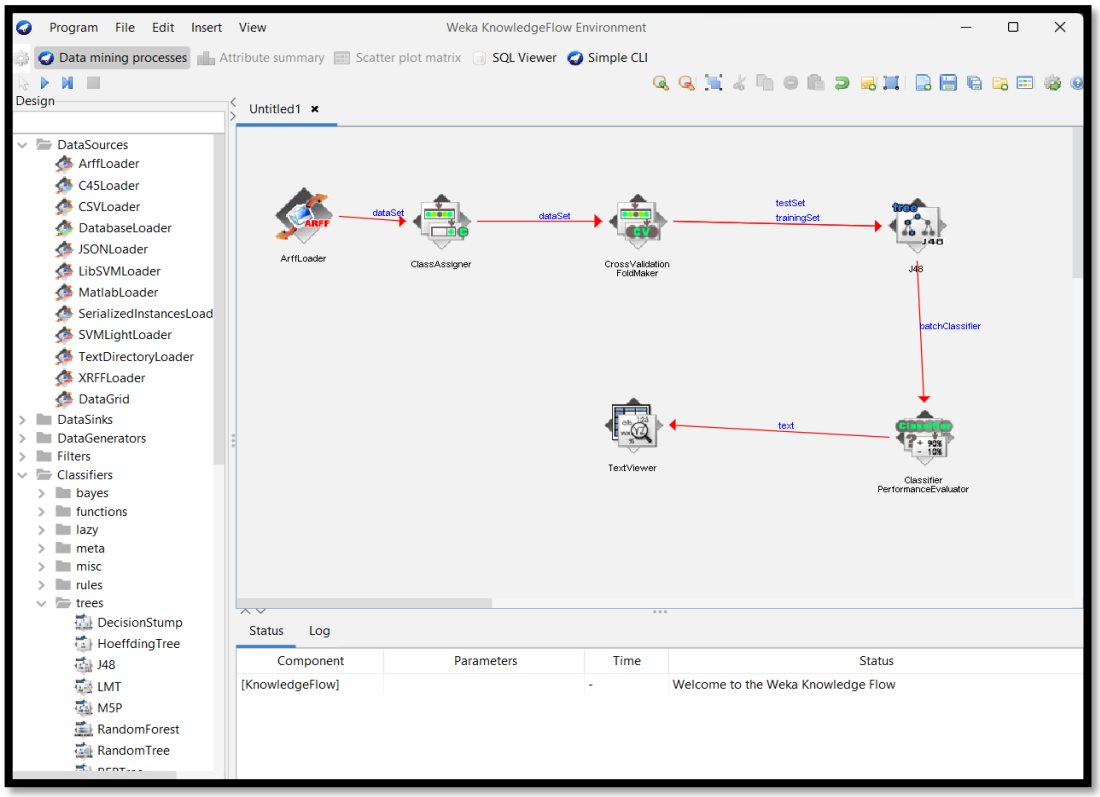
Viewer

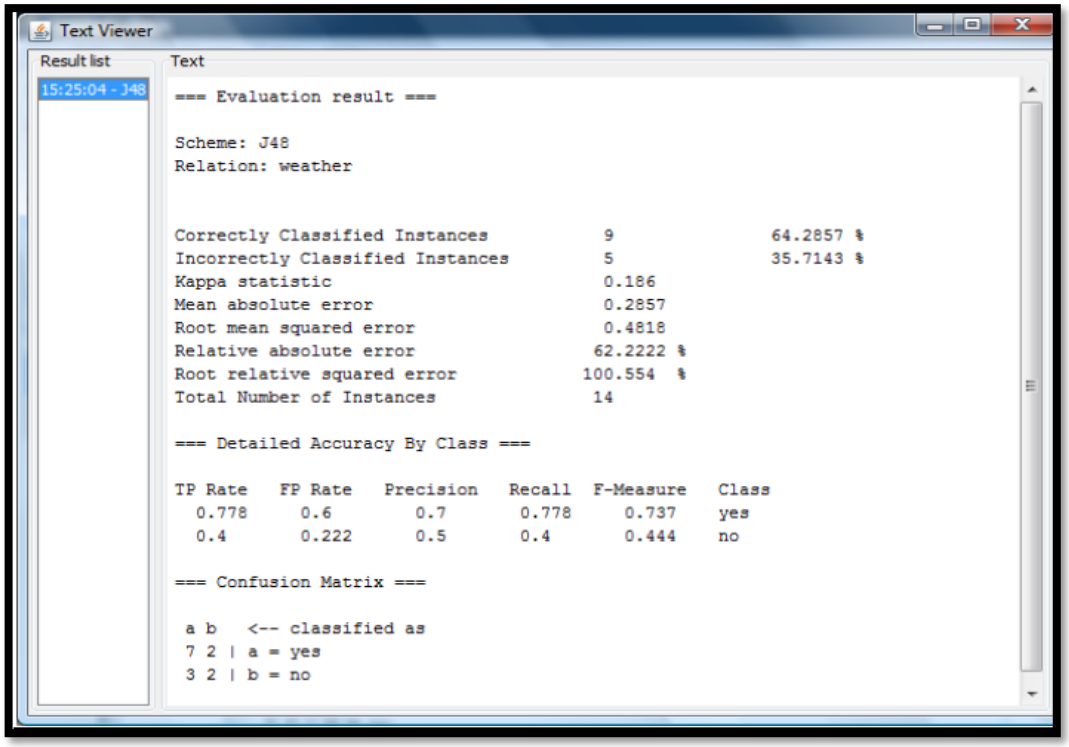
Relation: weather

No.	1: outlook Nominal	2: temperature Numeric	3: humidity Numeric	4: windy Nominal	5: **play** Nominal
1	sunny	85.0	85.0	FALSE	no
2	sunny	80.0	90.0	TRUE	no
3	overcast	83.0	86.0	FALSE	yes
4	rainy	70.0	96.0	FALSE	yes
5	rainy	68.0	80.0	FALSE	yes
6	rainy	65.0	70.0	TRUE	no
7	overcast	64.0	65.0	TRUE	yes
8	sunny	72.0	95.0	FALSE	no
9	sunny	69.0	70.0	FALSE	yes
10	rainy	75.0	80.0	FALSE	yes
11	sunny	75.0	70.0	TRUE	yes
12	overcast	72.0	90.0	TRUE	yes
13	overcast	81.0	75.0	FALSE	yes
14	rainy	71.0	91.0	TRUE	no

Procedure:

1) Start -> Programs -> Weka 3.4
2) Open Knowledge Flow.
3) Select Data Source tab & choose Arff Loader.
4) Place Arff Loader component on the layout area by clicking on that component.
5) Specify an Arff file to load by right clicking on Arff Loader icon, and then a pop-up menu will appear. In that select Configure & browse to the location of weather.arff
6) Click on the Evaluation tab & choose Class Assigner & place it on the layout.
7) Now connect the Arff Loader to the Class Assigner by right clicking on Arff Loader, and then select Data Set option, now a link will be established.
8) Right click on Class Assigner & choose Configure option, and then a new window will appear & specify a class to our data.
9) Select Evaluation tab & select Cross-Validation Fold Maker & place it on the layout.
10) Now connect the Class Assigner to the Cross-Validation Fold Maker.
11) Select Classifiers tab & select J48 component & place it on the layout.
12) Now connect Cross-Validation Fold Maker to J48 twice; first choose Training Data Set option and then Test Data Set option.
13) Select Evaluation Tab & select Classifier Performance Evaluator component & place it on the layout.
14) Connect J48 to Classifier Performance Evaluator component by right clicking on J48 & selecting Batch Classifier.
15) Select Visualization tab & select Text Viewer component & place it on the layout.
16) Connect Text Viewer to Classifier Performance Evaluator by right clicking on Text Viewer & by selecting Text option.
17) Start the flow of execution by selecting Start Loading from Arff Loader. 54
18) For viewing result, right click on Text Viewer & select the Show Results, and then the result will be displayed on the new window.


Program File Edit Insert View
Weka KnowledgeFlow Environment
Data mining processes
Attribute summary
Scatter plot matrix
SQL Viewer
Simple CLI
Design
Untitled1
DataSources
ArffLoader
C45Loader
CSVLoader
DatabaseLoader
JSONLoader
LibSVMLoader
MatlabLoader
SerializedInstancesLoad
SVMLightLoader
TextDirectoryLoader
XRFFLoader
DataGrid
DataSinks
DataGenerators
Filters
Classifiers
bayes
functions
lazy
meta
misc
rules
trees
DecisionStump
HoeffdingTree
J48
LMT
M5P
RandomForest
RandomTree
dataSet
dataSet
testSet
trainingSet
batchClassifier
text
ArffLoader
ClassAssigner
CrossValidation FoldMaker
J48
TextViewer
Classifier PerformanceEvaluator
Status
Log
Component
Parameters
Time
Status
[KnowledgeFlow]
-
Welcome to the Weka Knowledge Flow


Text Viewer
Result list
15:25:04 - J48
Text
=== Evaluation result ===
Scheme: J48
Relation: weather
Correctly Classified Instances 9 64.2857 %
Incorrectly Classified Instances 5 35.7143 %
Kappa statistic 0.186
Mean absolute error 0.2857
Root mean squared error 0.4818
Relative absolute error 62.2222 %
Root relative squared error 100.554 %
Total Number of Instances 14
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure Class
0.778 0.6 0.7 0.778 0.737 yes
0.4 0.222 0.5 0.4 0.444 no
=== Confusion Matrix ===
a b <-- classified as
7 2 | a = yes
3 2 | b = no

Practical 11A:

Aim:
Write a procedure for Employee data using Make Density Based Cluster Algorithm.

Description:
Cluster analysis or clustering is the task of assigning a set of objects into groups (called clusters) so that the objects in the same cluster are more similar (in some sense or another) to each other than to those in other clusters. Clustering is a main task of explorative data mining, and a common technique for statistical data analysis used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics.

Creation of Employee Table:
Procedure:
1) Open Start Programs Accessories Notepad
2) Type the following training data set with the help of Notepad for Employee Table

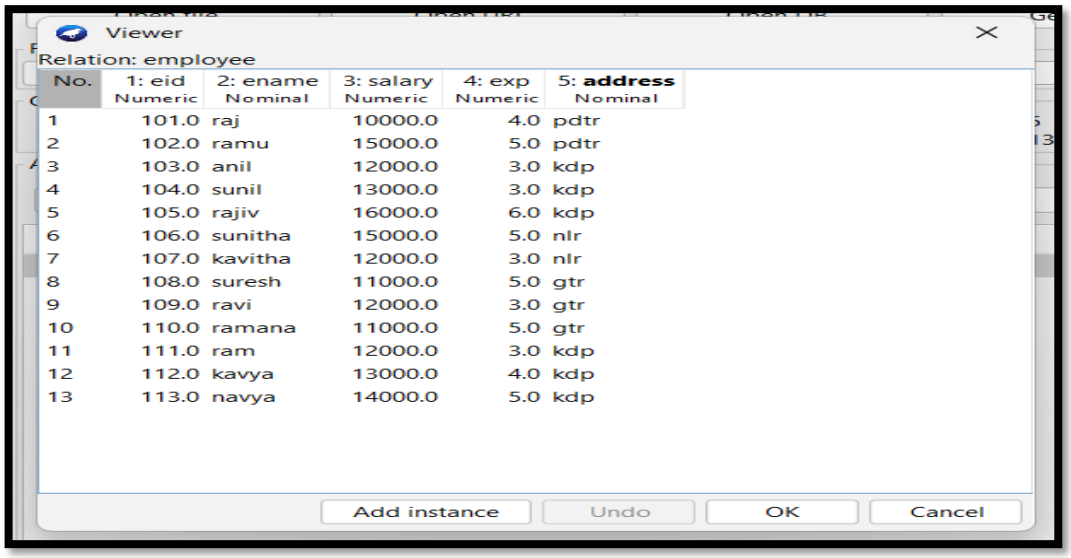
@relation employee
@attribute eid numeric
@attribute ename
{raj,ramu,anil,sunil,rajiv,sunitha,kavitha,suresh,ravi,ramana,ram,kavya,navya}
@attribute salary numeric
@attribute exp numeric
@attribute address {pdtr,kdp,nlr,gtr}
@data
101,raj,10000,4,pdtr
102,ramu,15000,5,pdtr
103,anil,12000,3,kdp
104,sunil,13000,3,kdp
105,rajiv,16000,6,kdp
106,sunitha,15000,5,nlr

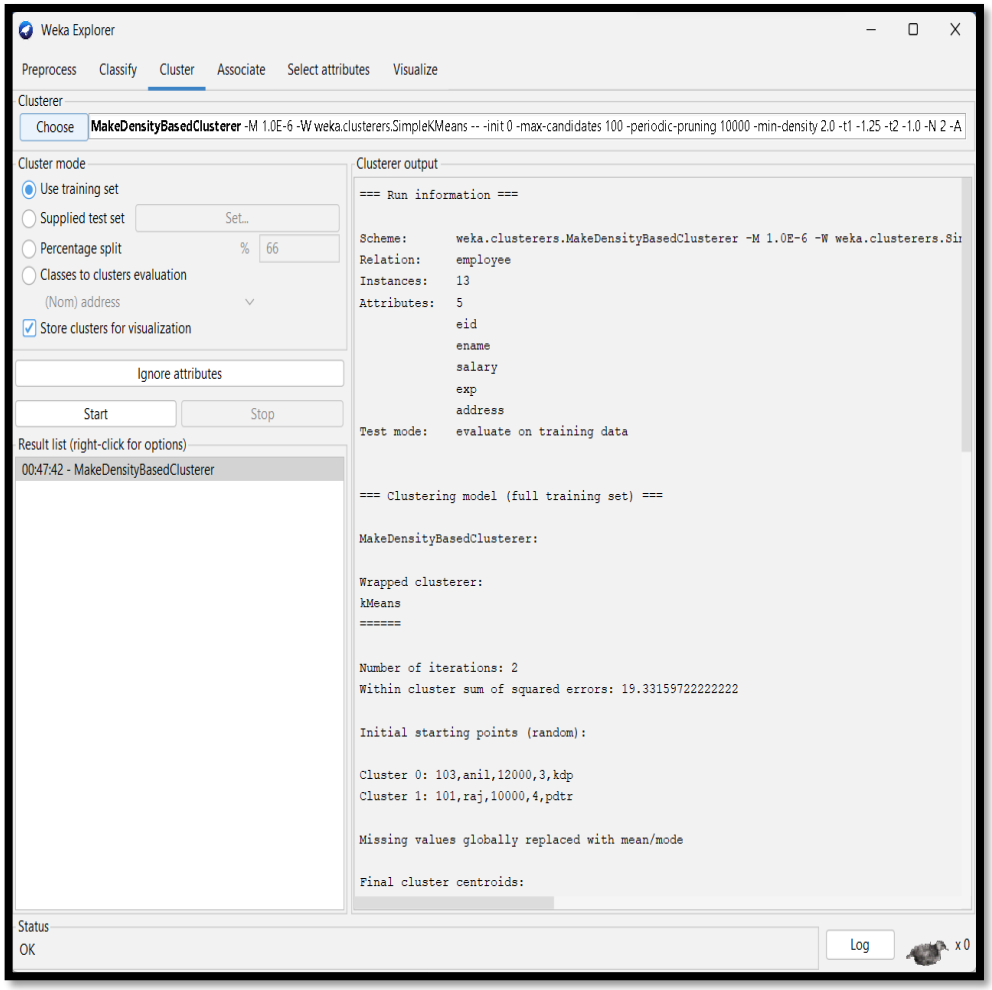
107,kavitha,12000,3,nlr
108,suresh,11000,5,gtr
109,ravi,12000,3,gtr
110,ramana,11000,5,gtr
111,ram,12000,3,kdp
112,kavya,13000,4,kdp
113,navya,14000,5,kdp

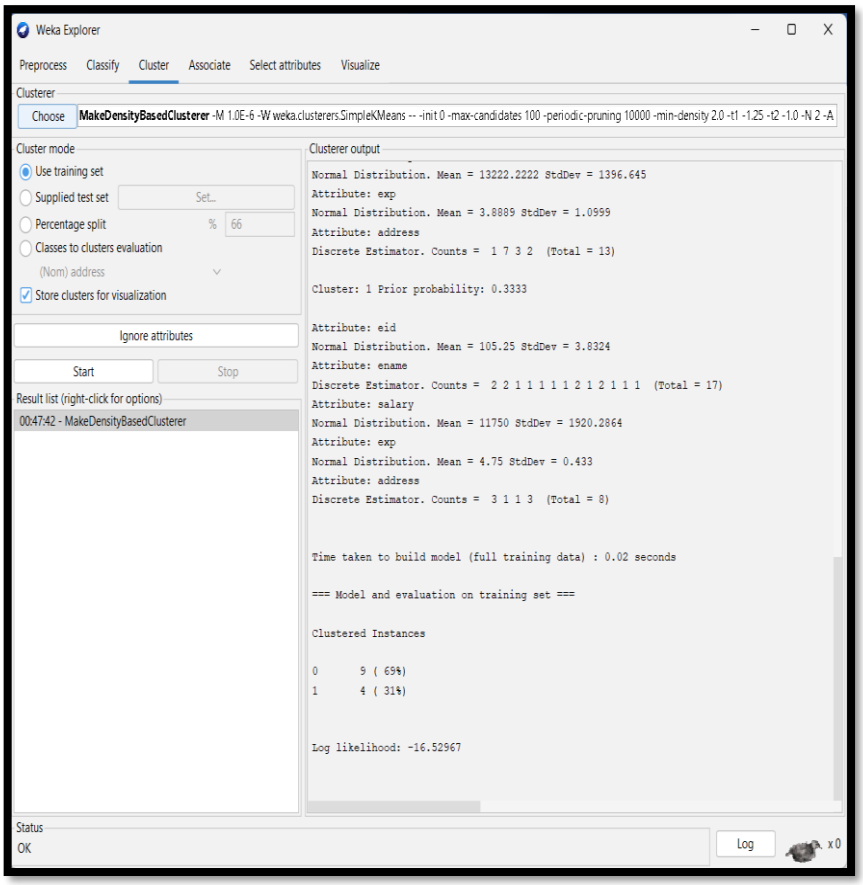
1) After that the file is saved with .arff file format.
2) Minimize the arff file and then open Start Programs weka-3-4.
3) Click on weka-3-4, then Weka dialog box is displayed on the screen.
4) In that dialog box there are four modes, click on explorer.
5) Explorer shows many options. In that click on 'open file' and select the arff file
6) Click on edit button which shows employee table on weka.

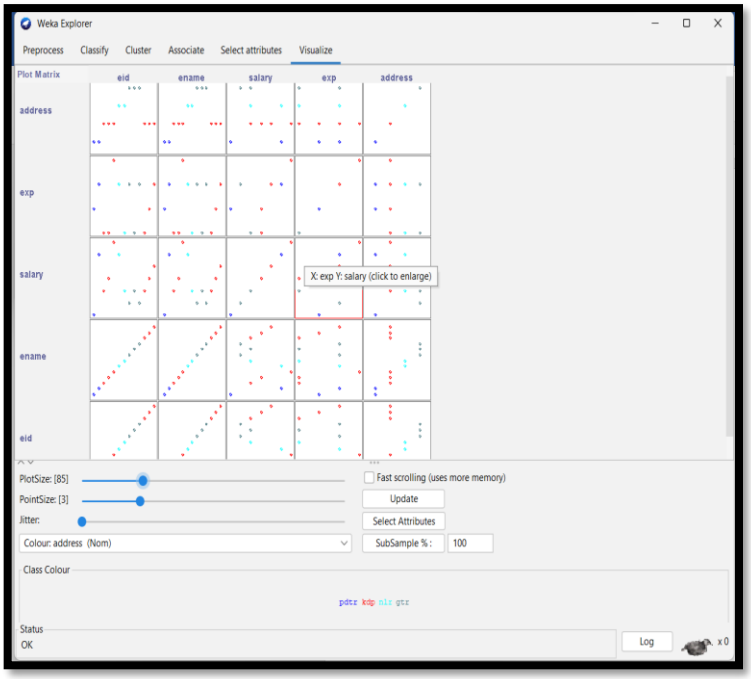
Procedure:

1) Click Start -> Programs -> Weka 3.4
2) Click on Explorer.
3) Click on open file & then select Employee.arff file.
4) Click on Cluster menu. In this there are different algorithms are there.
5) Click on Choose button and then select MakeDensityBasedClusterer algorithm.
6) Click on Start button and then output will be displayed on the screen.


Viewer
Relation: employee
No. 1: eid Numeric 2: ename Nominal 3: salary Numeric 4: exp Numeric 5: address Nominal
1 101.0 raj 10000.0 4.0 pdtr
2 102.0 ramu 15000.0 5.0 pdtr
3 103.0 anil 12000.0 3.0 kdp
4 104.0 sunil 13000.0 3.0 kdp
5 105.0 rajiv 16000.0 6.0 kdp
6 106.0 sunitha 15000.0 5.0 nlr
7 107.0 kavitha 12000.0 3.0 nlr
8 108.0 suresh 11000.0 5.0 gtr
9 109.0 ravi 12000.0 3.0 gtr
10 110.0 ramana 11000.0 5.0 gtr
11 111.0 ram 12000.0 3.0 kdp
12 112.0 kavya 13000.0 4.0 kdp
13 113.0 navya 14000.0 5.0 kdp
Add instance
Undo
OK
Cancel


Weka Explorer
Preprocess Classify Cluster Associate Select attributes Visualize
Clusterer
Choose MakeDensityBasedClusterer -M 1.0E-6 -W weka.clusterers.SimpleKMeans -- -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A
Cluster mode
Use training set
Supplied test set Set...
Percentage split % 66
Classes to clusters evaluation
(Nom) address
Store clusters for visualization
Ignore attributes
Start Stop
Result list (right-click for options)
00:47:42 - MakeDensityBasedClusterer
Clusterer output
=== Run information ===
Scheme: weka.clusterers.MakeDensityBasedClusterer -M 1.0E-6 -W weka.clusterers.Si
Relation: employee
Instances: 13
Attributes: 5
eid
ename
salary
exp
address
Test mode: evaluate on training data
=== Clustering model (full training set) ===
MakeDensityBasedClusterer:
Wrapped clusterer:
kMeans
======
Number of iterations: 2
Within cluster sum of squared errors: 19.33159722222222
Initial starting points (random):
Cluster 0: 103,anil,12000,3,kdp
Cluster 1: 101,raj,10000,4,pdtr
Missing values globally replaced with mean/mode
Final cluster centroids:
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
MakeDensityBasedClusterer -M 1.0E-6 -W weka.clusterers.SimpleKMeans -- -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Nom) address
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
00:47:42 - MakeDensityBasedClusterer
Clusterer output
Normal Distribution. Mean = 13222.2222 StdDev = 1396.645
Attribute: exp
Normal Distribution. Mean = 3.8889 StdDev = 1.0999
Attribute: address
Discrete Estimator. Counts = 1 7 3 2 (Total = 13)
Cluster: 1 Prior probability: 0.3333
Attribute: eid
Normal Distribution. Mean = 105.25 StdDev = 3.8324
Attribute: ename
Discrete Estimator. Counts = 2 2 1 1 1 1 1 2 1 2 1 1 1 (Total = 17)
Attribute: salary
Normal Distribution. Mean = 11750 StdDev = 1920.2864
Attribute: exp
Normal Distribution. Mean = 4.75 StdDev = 0.433
Attribute: address
Discrete Estimator. Counts = 3 1 1 3 (Total = 8)
Time taken to build model (full training data) : 0.02 seconds
=== Model and evaluation on training set ===
Clustered Instances
0 9 (69%)
1 4 (31%)
Log likelihood: -16.52967
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Plot Matrix
eid
ename
salary
exp
address
X: exp Y: salary (click to enlarge)
PlotSize: [85]
PointSize: [3]
Jitter:
Colour: address (Nom)
Fast scrolling (uses more memory)
Update
Select Attributes
SubSample % :
100
Class Colour
pdtr kdp nlr gtr
Status
OK
Log
x 0

Practical 11B:

Aim:
Write a procedure for Clustering Customer data using Simple KMeans Algorithm.

Description:
Cluster analysis or clustering is the task of assigning a set of objects into groups (called clusters) so that the objects in the same cluster are more similar (in some sense or another) to each other than to those in other clusters. Clustering is a main task of explorative data mining, and a common technique for statistical data analysis used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics.
Creation of Customer Table:

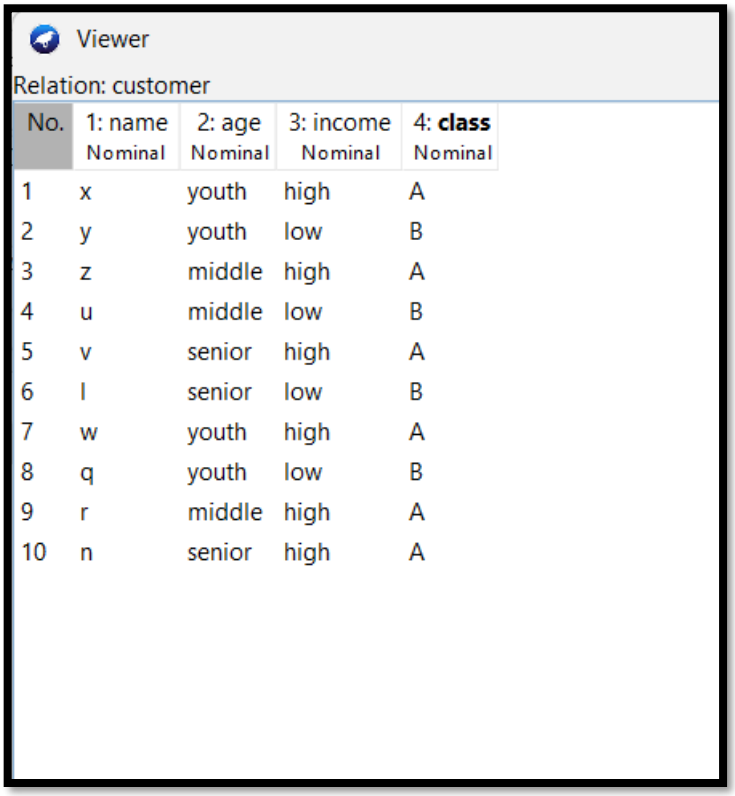
Procedure:
1) Open Start Programs Accessories Notepad
2) Type the following training data set with the help of Notepad for Buying Table.

```
@relation customer
@attribute name {x,y,z,u,v,l,w,q,r,n}
@attribute age {youth,middle,senior}
@attribute income {high,medium,low}
@attribute class {A,B}
@data
x,youth,high,A
y,youth,low,B
z,middle,high,A
u,middle,low,B
v,senior,high,A
l,senior,low,B
w,youth,high,A
q,youth,low,B
r,middle,high,A
n,senior,high,A
```

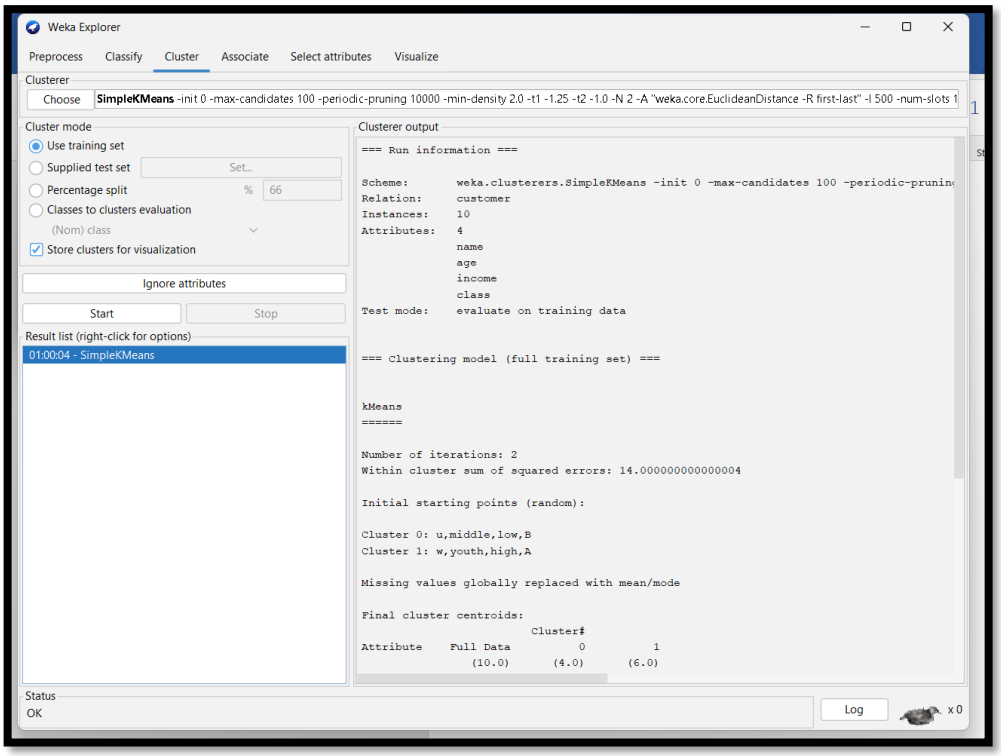
3) After that the file is saved with .arff file format.
4) Minimize the arff file and then open Start Programs weka-3-4.
5) Click on weka-3-4, then Weka dialog box is displayed on the screen.
6) In that dialog box there are four modes, click on explorer.
7) Explorer shows many options. In that click on 'open file' and select the arff file
8) Click on edit button which shows buying table on weka.

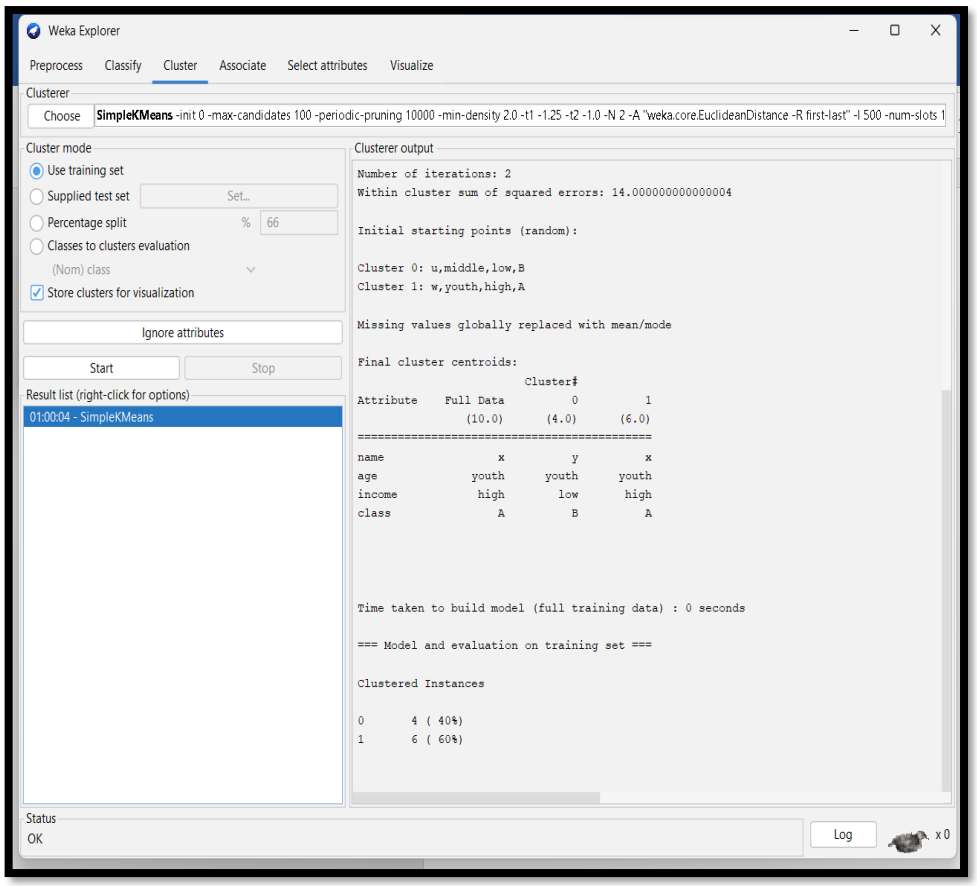
Procedure:

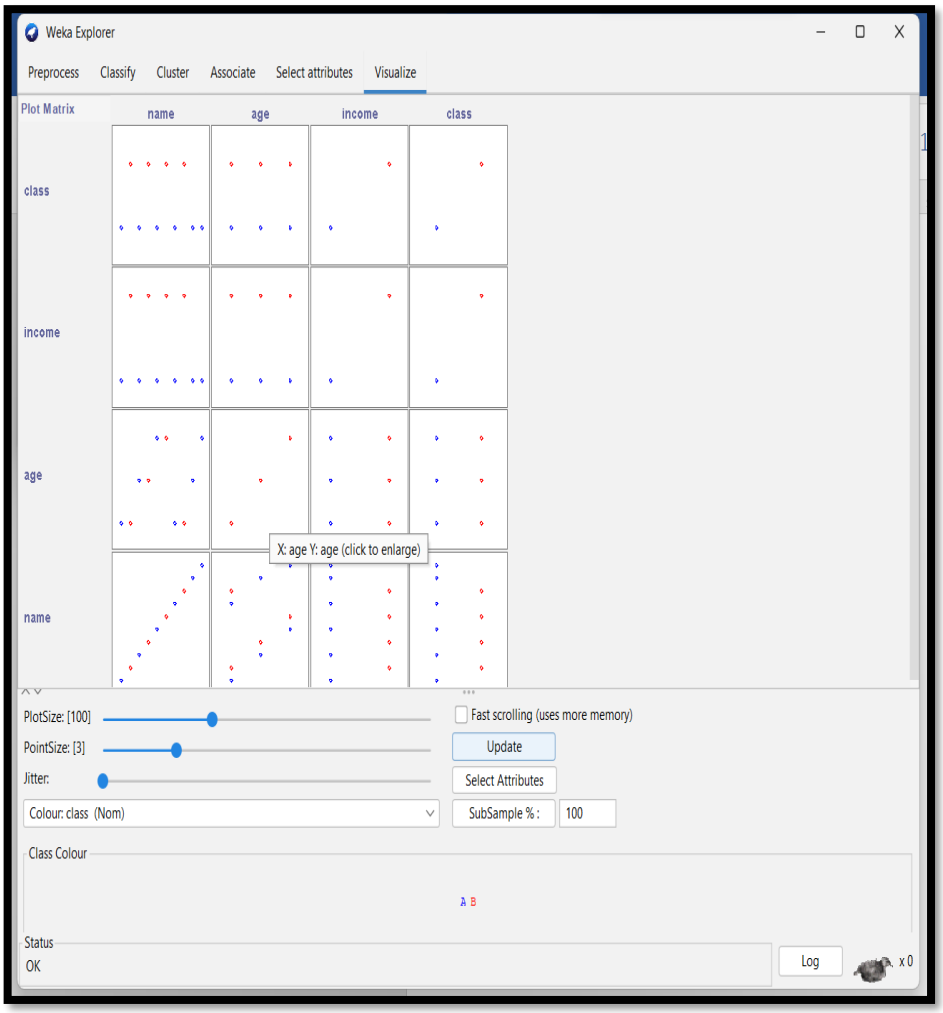
1) Click Start -> Programs -> Weka 3.4
2) Click on Explorer. 3) Click on open file & then select Customer.arff file.
3) Click on Cluster menu. In this there are different algorithms are there.
4) Click on Choose button and then select SimpleKMeans algorithm.
5) Click on Start button and then output will be displayed on the screen



No.	1: name Nominal	2: age Nominal	3: income Nominal	4: **class** Nominal
1	x	youth	high	A
2	y	youth	low	B
3	z	middle	high	A
4	u	middle	low	B
5	v	senior	high	A
6	l	senior	low	B
7	w	youth	high	A
8	q	youth	low	B
9	r	middle	high	A
10	n	senior	high	A


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Nom) class
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
01:00:04 - SimpleKMeans
Clusterer output
=== Run information ===
Scheme: weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning
Relation: customer
Instances: 10
Attributes: 4
name
age
income
class
Test mode: evaluate on training data
=== Clustering model (full training set) ===
kMeans
======
Number of iterations: 2
Within cluster sum of squared errors: 14.000000000000004
Initial starting points (random):
Cluster 0: u,middle,low,B
Cluster 1: w,youth,high,A
Missing values globally replaced with mean/mode
Final cluster centroids:
Cluster#
Attribute Full Data 0 1
(10.0) (4.0) (6.0)
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Nom) class
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
01:00:04 - SimpleKMeans
Clusterer output
Number of iterations: 2
Within cluster sum of squared errors: 14.000000000000004
Initial starting points (random):
Cluster 0: u,middle,low,B
Cluster 1: w,youth,high,A
Missing values globally replaced with mean/mode
Final cluster centroids:
Cluster#
Attribute Full Data 0 1
(10.0) (4.0) (6.0)
name x y x
age youth youth youth
income high low high
class A B A
Time taken to build model (full training data) : 0 seconds
=== Model and evaluation on training set ===
Clustered Instances
0 4 (40%)
1 6 (60%)
Status
OK
Log
x 0



Result:

The program has been successfully executed.

Practical No 12A:

Aim:
Write a Procedure for Classification on Weather data using KNN Classification.

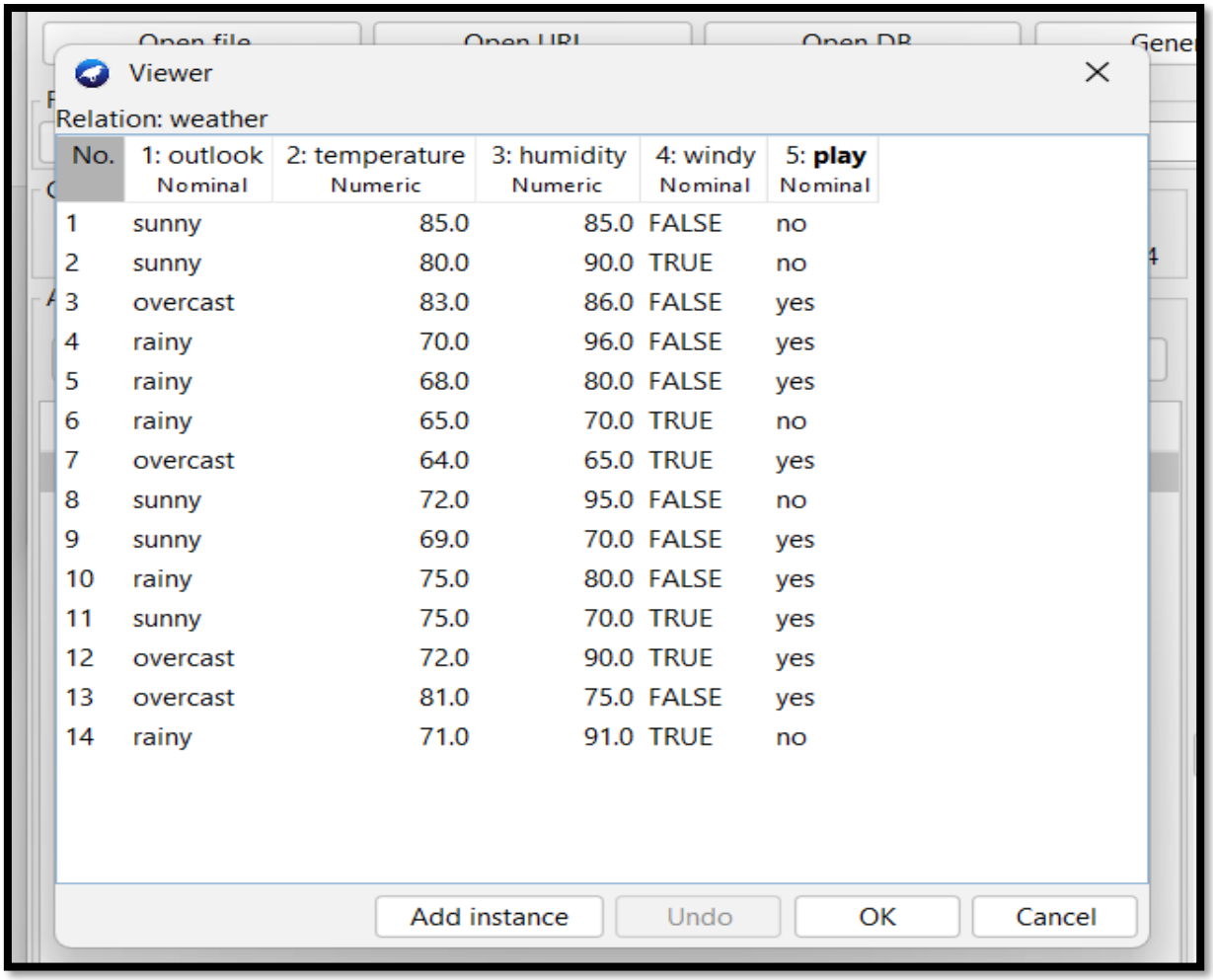
Description:
K-Nearest Neighbours is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection.
It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as GMM, which assume a Gaussian distribution of the given data).
We are given some prior data (also called training data), which classifies coordinates into groups identified by an attribute.

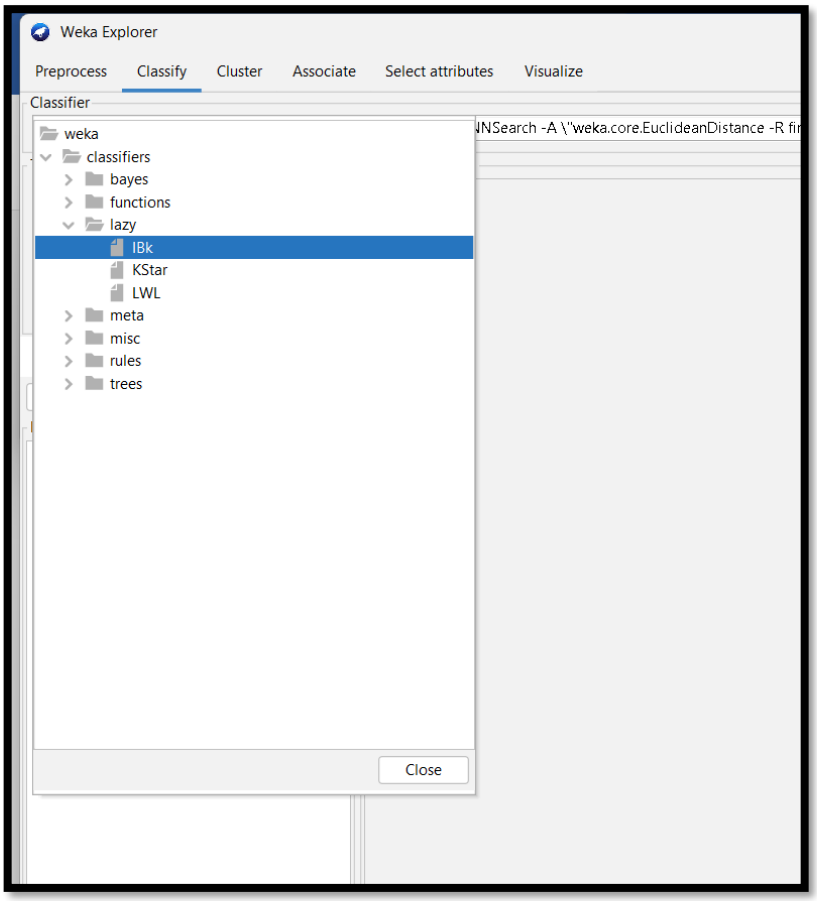
Creation of Weather Table:
Procedure:
1) Open Start Programs Accessories Notepad
2) Type the following training data set with the help of Notepad for Weather Table.

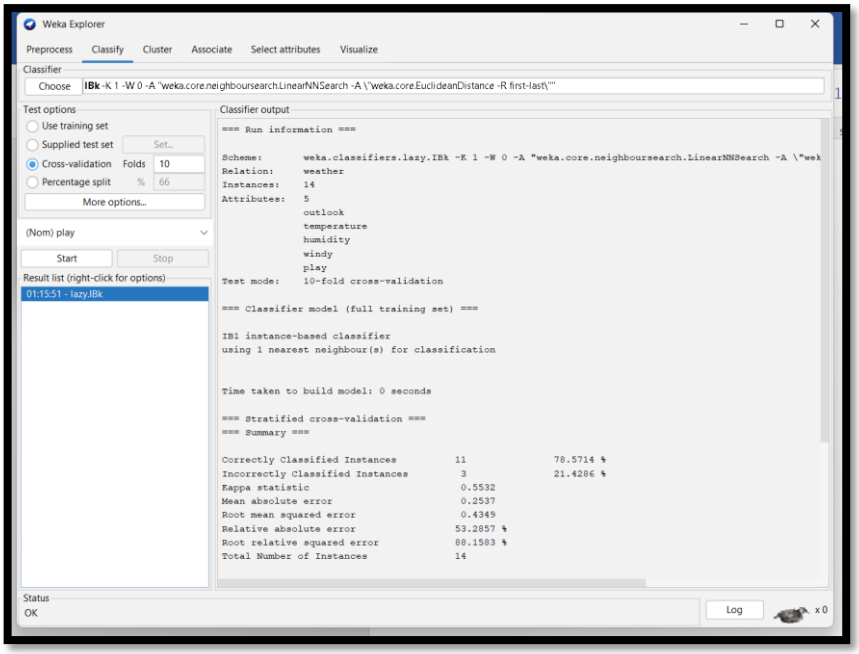
```
@relation weather
@attribute outlook {sunny,rainy,overcast}
@attribute temparature numeric
@attribute humidity numeric
@attribute windy {true,false}
@attribute play {yes,no}
 @data
sunny,85.0,85.0,false,no
overcast,80.0,90.0,true,no
sunny,83.0,86.0,false,yes
rainy,70.0,86.0,false,yes
rainy,68.0,80.0,false,yes
rainy,65.0,70.0,true,no
```

overcast,64.0,65.0,false,yes
sunny,72.0,95.0,true,no
sunny,69.0,70.0,false,yes
rainy,75.0,80.0,false,yes

1) 3)After that the file is saved with .arff file format.
3) Minimize the arff file and then open Start Programs weka-3-4.
4) Click on weka-3-4, then Weka dialog box is displayed on the screen.
5) In that dialog box there are four modes, click on explorer.
6) Explorer shows many options. In that click on 'open file' and select the arff file
7) Click on edit button which shows weather table on weka.

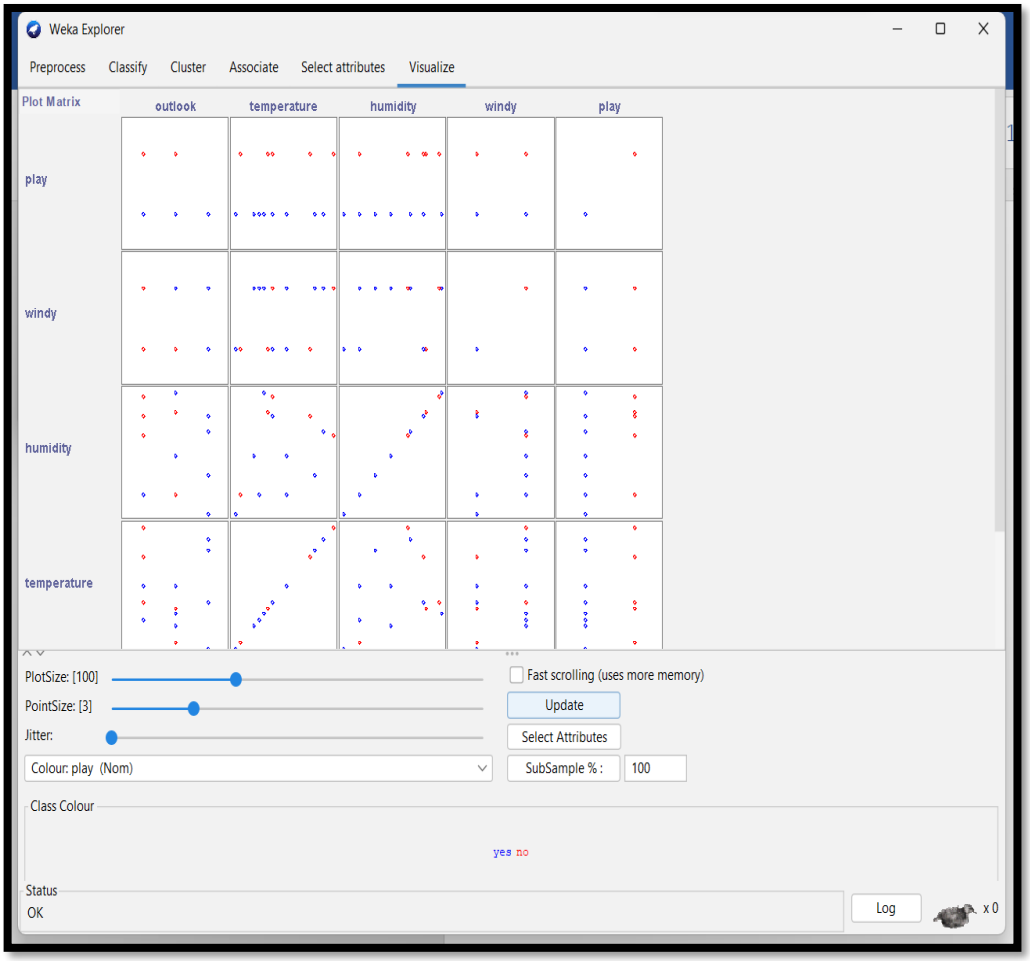



Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
weka
classifiers
bayes
functions
lazy
IBk
KStar
LWL
meta
misc
rules
trees
Close


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) play
Start
Stop
Result list (right-click for options)
01:15:51 - lazy.IBk
Classifier output
=== Run information ===
Scheme: weka.classifiers.lazy.IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"wek
Relation: weather
Instances: 14
Attributes: 5
outlook
temperature
humidity
windy
play
Test mode: 10-fold cross-validation
=== Classifier model (full training set) ===
IB1 instance-based classifier
using 1 nearest neighbour(s) for classification
Time taken to build model: 0 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 11 78.5714 %
Incorrectly Classified Instances 3 21.4286 %
Kappa statistic 0.5532
Mean absolute error 0.2537
Root mean squared error 0.4349
Relative absolute error 53.2857 %
Root relative squared error 88.1583 %
Total Number of Instances 14
Status
OK
Log
x 0

```
=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                 0.778     0.200     0.875       0.778   0.824      0.559    0.811     0.882     yes
                 0.800     0.222     0.667       0.800   0.727      0.559    0.811     0.600     no
Weighted Avg.    0.786     0.208     0.801       0.786   0.789      0.559    0.811     0.781

=== Confusion Matrix ===

 a b   <-- classified as
 7 2 | a = yes
 1 4 | b = no
```

Log x 0



Result:

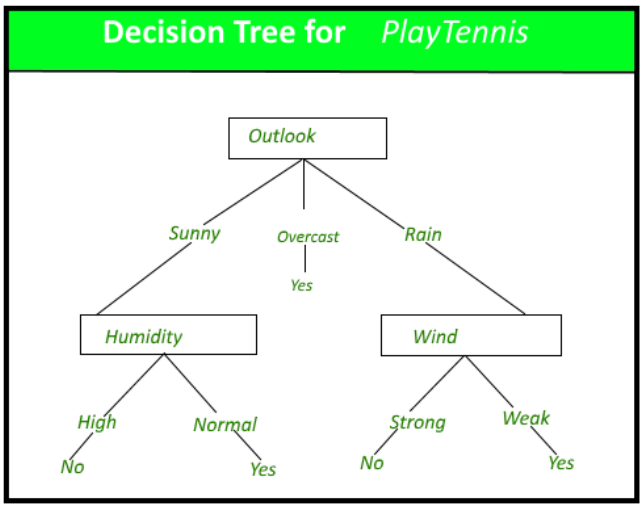
This Experiment is finally executed.

Practical-12B

Aim:
Write a procedure for Classification on Weather Data using Decision Tree

Description:

Decision Tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.



A decision tree for the concept PlayTennis.

Construction of Decision Tree: A tree can be "learned" by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions. The construction of a decision tree classifier does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery. Decision trees can handle high-dimensional data. In general decision

tree, classifier has good accuracy. Decision tree induction is a typical inductive approach to learn knowledge on classification.

Creation of Weather Table:

Procedure:

1) Open Start Programs Accessories Notepad
2) Type the following training data set with the help of Notepad for Weather Table.

```
@relation weather
@attribute outlook {sunny, rainy, overcast}
@attribute temperature numeric
@attribute humidity numeric
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}
@data
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes
rainy,65,70,TRUE,no
overcast,64,65,TRUE,yes
sunny,72,95,FALSE,no
sunny,69,70,FALSE,yes
rainy,75,80,FALSE,yes
sunny,75,70,TRUE,yes
overcast,72,90,TRUE,yes
overcast,81,75,FALSE,yes
rainy,71,91,TRUE,no
```

3) After that the file is saved with .arff file format.
4) Minimize the arff file and then open Start Programs weka-3-4. 32
5) Click on weka-3-4, then Weka dialog box is displayed on the screen.
6) In that dialog box there are four modes, click on explorer.

7) Explorer shows many options. In that click on 'open file' and select the arff file

8) Click on edit button which shows weather table on weka.

1) Procedure for Decision Trees:

1) Open Start Programs Weka-3-4 Weka-3-4

2) Open explorer.

3) Click on open file and select weather.arff

4) Select Classifier option on the top of the Menu bar.

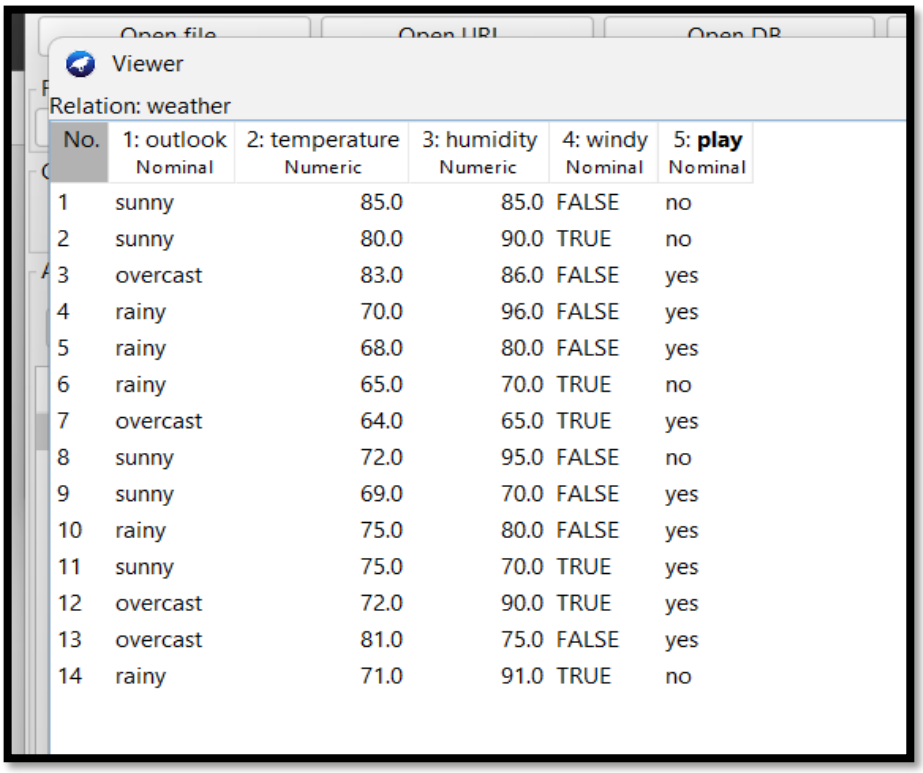
5) Select Choose button and click on Tree option.

6) Click on J48.

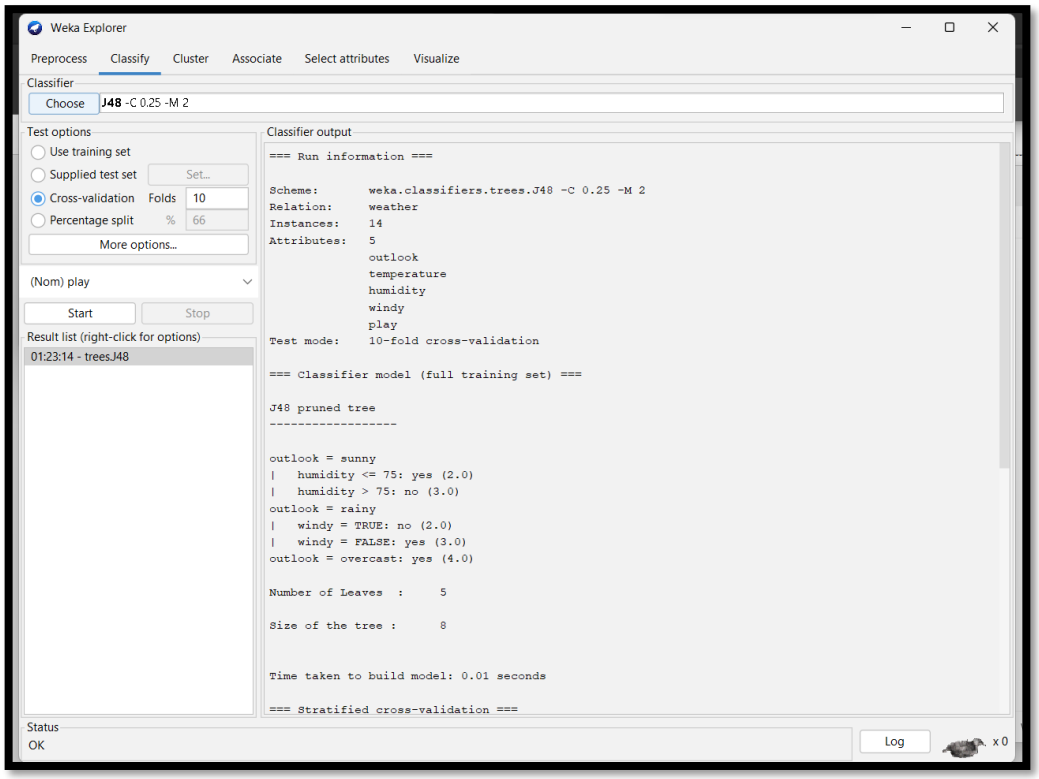
7) Click on Start button and output will be displayed on the right side of the window.

8) Select the result list and right click on result list and select Visualize Tree option.

9) Then Decision Tree will be displayed on new window.

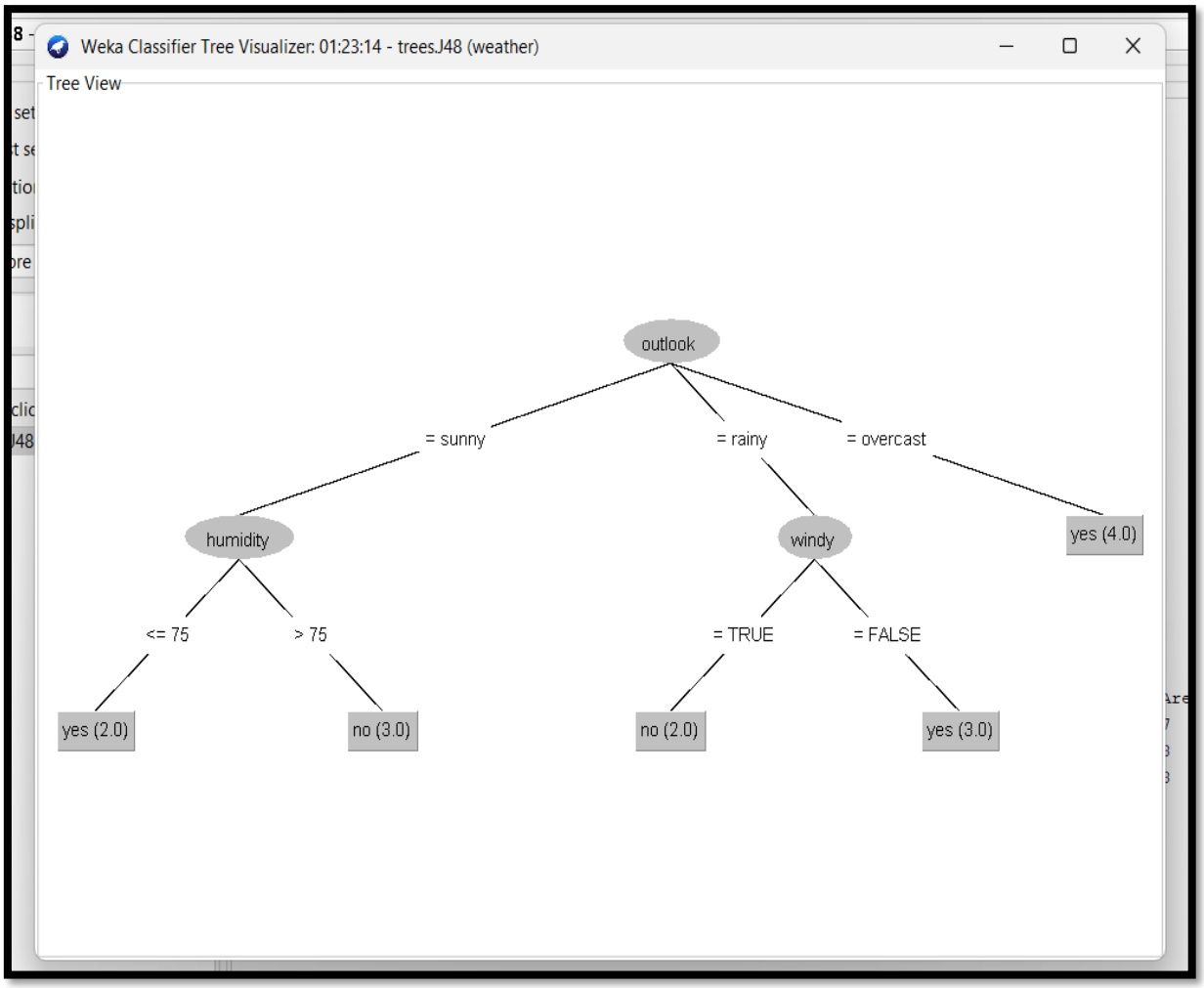


No.	1: outlook Nominal	2: temperature Numeric	3: humidity Numeric	4: windy Nominal	5: **play** Nominal
1	sunny	85.0	85.0	FALSE	no
2	sunny	80.0	90.0	TRUE	no
3	overcast	83.0	86.0	FALSE	yes
4	rainy	70.0	96.0	FALSE	yes
5	rainy	68.0	80.0	FALSE	yes
6	rainy	65.0	70.0	TRUE	no
7	overcast	64.0	65.0	TRUE	yes
8	sunny	72.0	95.0	FALSE	no
9	sunny	69.0	70.0	FALSE	yes
10	rainy	75.0	80.0	FALSE	yes
11	sunny	75.0	70.0	TRUE	yes
12	overcast	72.0	90.0	TRUE	yes
13	overcast	81.0	75.0	FALSE	yes
14	rainy	71.0	91.0	TRUE	no




```
Time taken to build model: 0.01 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances           9               64.2857 %
Incorrectly Classified Instances         5               35.7143 %
Kappa statistic                          0.186
Mean absolute error                      0.2857
Root mean squared error                  0.4818
Relative absolute error                 60      %
Root relative squared error             97.6586 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.778    0.600    0.700      0.778    0.737      0.189    0.789     0.847     yes
                 0.400    0.222    0.500      0.400    0.444      0.189    0.789     0.738     no
Weighted Avg.    0.643    0.465    0.629      0.643    0.632      0.189    0.789     0.808

=== Confusion Matrix ===

 a b   <-- classified as
 7 2 | a = yes
 3 2 | b = no
```

Log x 0



Result:

This program has been successfully executed