EX.NO:10 ANALYSIS OF BENCHMARK DATASET USING RAPIDMINER TOOL, KNIME TOOL

AIM:

To analyse the benchmark dataset using RapidMiner tool and Knime tool in data mining.

PROCEDURE:

Analysis of benchmark datasets using RapidMiner Tool, KNIME Tool.

Benchmark datasets are used both for method training and testing. We can divide testing approaches into three categories. The most reliable are systematic benchmark studies. Quite often the initial method performance assessment is done on somewhat limited test data or does not report all necessary measures. The third group includes studies for initial method and hypothesis testing typically with a limited amount of data.

Development and testing of computational methods are dependent on experimental data. Only in comparison to existing knowledge can method performance be assessed. For that purpose, benchmark datasets with known and verified outcome are needed. During the last few years, such datasets have been collected for a number of applications in the field of variation interpretation. VariBench (1) and VariSNP (2) are the two existing databases for variation benchmark datasets for variation interpretation. VariBench contains all kinds of datasets while VariSNP is a dedicated resource for variation sets from dbSNP database for short variations .

Criteria for Benchmarks:

Step 1: Installing and Opening Knime

The first step is to download and install Knime from their official website. Once installed, open Knime, and you will see a blank workflow space. We can start by importing our benchmark dataset.

Step 2: Importing the Dataset

To import the dataset, click on the "File" tab, select "Import KNIME Workflow," and then navigate to the location of your benchmark dataset and select it. Once imported, the dataset will appear in the workflow space.

Step 3: Data Cleaning and Manipulation
The next step is to clean and manipulate the data. Knime offers a wide range of data manipulation tools, including:

- Data cleaning: We can use the "Data Manipulation" node to remove duplicates, missing values, and unnecessary columns or rows.

- Data transformation: We can use the "Column Expressions" node to create new columns, modify existing ones, or apply mathematical operations.

- Data aggregation: We can use the "Group By" node to group data by a certain category and perform aggregation functions such as sum, count, mean, and median.

- Data filtering: We can use the "Row Filter" node to filter the data based on a specific criterion.

Step 4: Data Analysis
Once the data is cleaned and manipulated, we can start analyzing it using Knime's various analysis tools. Some of the most common analysis nodes include:

- Statistics: We can use the "Statistics" node to calculate various statistical measures such as mean, median, standard deviation, correlation coefficient, and regression analysis.

- Classification: We can use the "Decision Tree" node, "Random Forest," or "Naive Bayes" node to classify data into different categories based on certain attributes.

- Clustering: We can use the "k-Means" node or "Hierarchical Clustering" node to group data points into different clusters based on their similarity.

- Association Rules: We can use the "Association Rule Learner" node to discover frequent itemsets and generate association rules among them.

Step 5: Data Visualization
Once we have analyzed the data, we can use Knime's visualization tools to create informative charts and graphs. Some of the most commonly used nodes for data visualization include:

- Scatter Plot: We can use the "Scatter Plot" node to create a scatter plot of our dataset to visualize the relationship between two variables.

- Bar Chart: We can use the "Bar Chart" node to create a bar chart of categorical data to compare different categories.

- Line Chart: We can use the "Line Chart" node to create a line chart to visualize trends over time.
- Heatmap: We can use the "Heatmap" node to create a heatmap to visualize the distribution of values across different categories.

Step 6: Exporting the Results
Once we have completed our analysis and created informative charts and graphs, we can export our results using the "File" tab and selecting "Export KNIME Workflow."

This will create a report that summarizes our analysis and includes any charts and graphs we created.

RESULT:

Thus to analyse the benchmark dataset using RapidMiner tool and Knime tool in data mining has been executed successfully.

EX.NO:11 CROSS-VALIDATION USING J48 ALGORITHM

AIM:

To write a procedure for cross-validation using J48 Algorithm for weather table.

PROCEDURE:

CREATION OF WEATHER TABLE:

1.Open Start -> Programs -> Accessories -> Notepad

2.Type the following training data set with the help of Notepadfor Weather Table.

```
@relation weather

@attribute outlook {sunny, overcast, rainy}
@attribute temperature real
@attribute humidity real @attribute
windy {TRUE, FALSE}@attribute play
{yes, no}

@data sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes
rainy,65,70,TRUE,no
overcast,64,65,TRUE,yes
sunny,72,95,FALSE,no
sunny,69,70,FALSE,yes
rainy,75,80,FALSE,yes
sunny,75,70,TRUE,yes
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 & then open Start -> Programs -> weka .
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:

Training dataset – Weather table:

- **Start** -> **Programs** -> **Weka 3.4**
- Open **Knowledge Flow**.
- Select **Data Source** tab & choose **Arff Loader**.
- Place **Arff Loader** component on the **layout area** by clicking on that component.
- 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.**
- Click on the **Evaluation** tab & choose **Class Assigner** & place it on the layout.
- Now **connect** the **Arff Loader** to the **Class Assigner** by right clicking on Arff Loader, and then select.
- Right click on **Class Assigner** & choose **Configure** option, and then a new window will appear & specifya class to our data.
- Select **Evaluation** tab & select **Cross-Validation Fold Maker** & place it on the layout.
- Now **connect** the **Class Assigner** to the **Cross-Validation Fold Maker**.
- Select **Classifiers** tab & select **J48** component & place it on the layout.
- Now **connect Cross-Validation Fold Maker** to **J48 twice**; **first** choose **Training Data Set** option and
- Select **Evaluation Tab** & select **Classifier Performance Evaluator** component & place it on the layout.
- Connect **J48** to **Classifier Performance Evaluator** component by right clicking on J48 & selecting
- Select **Visualization** tab & select **Text Viewer** component & place it on the layout.
- Connect **Text Viewer** to **Classifier Performance Evaluator** by right clicking on Text Viewer & byselecting **Text** option.
- Start the flow of execution by selecting **Start Loading** from **Arff Loader**.
- For viewing **result**, **right click** on **Text Viewer** & select the **Show Results**, and then the result will bedisplayed on the new window

OUTPUT:

Viewer

Relation: weather

No.	outlook Nominal	temperature Numeric	humidity Numeric	windy Nominal	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

Undo OK Cancel

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
```

RESULT:

Thus the program for cross-validation using J48 Algorithm for weather table has been executed successfully.

EX.NO:12 CLUSTERING USING COBWEB ALGORITHM

AIM:

To write a procedure for Clustering Buying data using Cobweb Algorithm.

PROCEDURE:

CREATION OF BUYING TABLE:

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,y
es 20-
40,low,y
es,fair,ye
s
G40,medium,yes,fair,
yes
L20,low,no,fair,no
G40,high,no,excellent
,yes
```

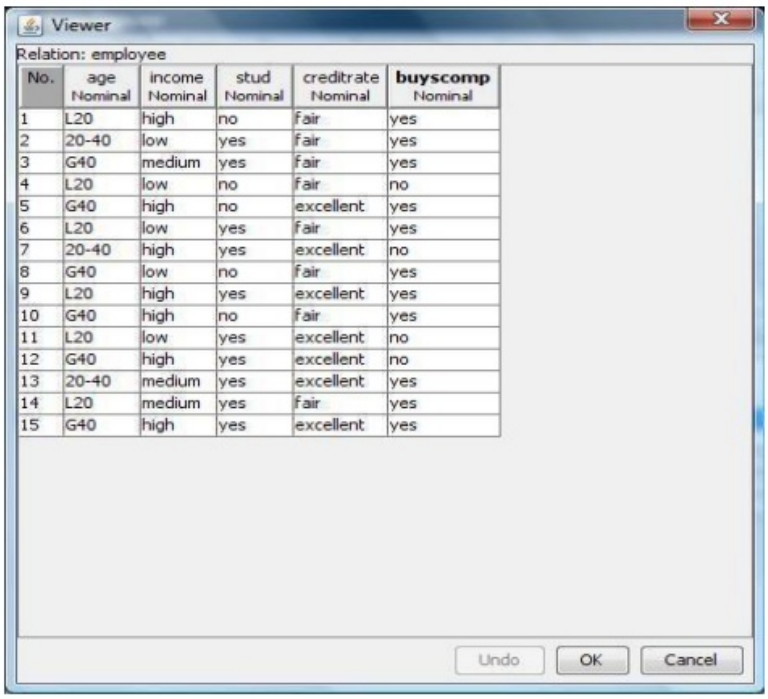
L20,low,yes,fair,yes
20-
40,high,yes,ex
cellent,no
G40,low,no,fa
ir,yes
L20,high,yes,
excellent,yes
G40,high,no,f
air,yes
L20,low,yes,e
xcellent,no
G40,high,yes,
excellent,no
20-
40,medium,yes,excellent,yes
L20,medium,yes,fair,yes
G40,high,yes,excellent,yes

3) After that the file is saved with **.arff** file format
4) Minimize the arff file and then open Start -> Programs -> weka .
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.

Training dataset – Buying table:

- Click **Start -> Programs -> Weka 3.4.**
- Click on **Explorer**.
- Click on **open file** & then select **Buying.arff** file.
- Click on **Cluster menu**. In this there are different algorithms are there.
- Click on **Choose button** and then select **cobweb** algorithm.
- Click on **Start button** and then **output** will be displayed on the screen

OUTPUT:

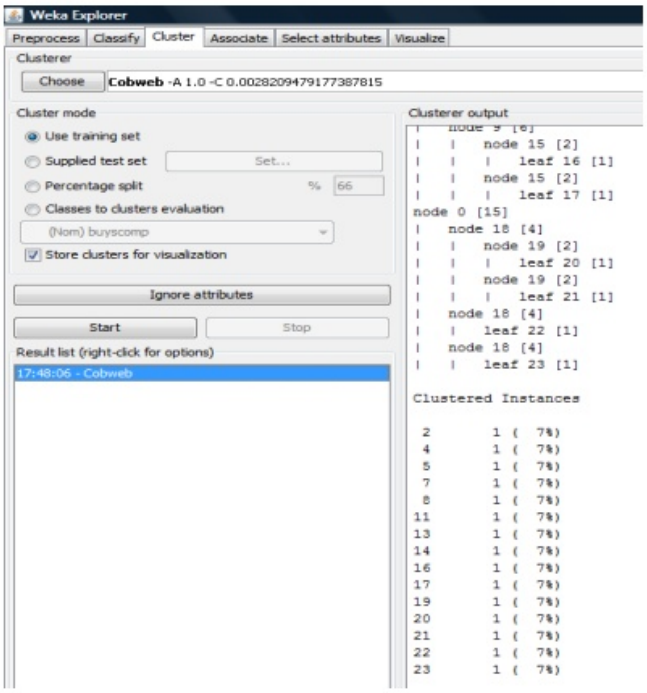


Viewer

Relation: employee

No.	age Nominal	income Nominal	stud Nominal	creditrate Nominal	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

Undo OK Cancel



Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Clusterer

Choose Cobweb -A 1.0 -C 0.0028209479177387815

Cluster mode

Use training set

Supplied test set Set...

Percentage split % 66

Classes to clusters evaluation

(Nom) buyscomp

Store clusters for visualization

Ignore attributes

Start Stop

Result list (right-click for options)

17:48:06 - Cobweb

Clusterer output

```
|   |   node 15 [2]
|   |   |   leaf 16 [1]
|   |   node 15 [2]
|   |   |   leaf 17 [1]
node 0 [15]
|   node 18 [4]
|   |   node 19 [2]
|   |   |   leaf 20 [1]
|   |   node 19 [2]
|   |   |   leaf 21 [1]
|   node 18 [4]
|   |   leaf 22 [1]
|   node 18 [4]
|   |   leaf 23 [1]

Clustered Instances

 2       1 (  7%)
 4       1 (  7%)
 5       1 (  7%)
 7       1 (  7%)
 8       1 (  7%)
11       1 (  7%)
13       1 (  7%)
14       1 (  7%)
16       1 (  7%)
17       1 (  7%)
19       1 (  7%)
20       1 (  7%)
21       1 (  7%)
22       1 (  7%)
23       1 (  7%)
```

RESULT:

Thus the program for Clustering Buying data using Cobweb Algorithm has been executed successfully.