PRACTICAL FILE

DATA MINING AND PREDICTIVE ANALYSIS LAB

MLE6806

Submitted By:
Khushi Tanwar
I.B.Tech + M.Tech (AIML)
A501132620001

Submitted To:
Dr. Shivani Sharma

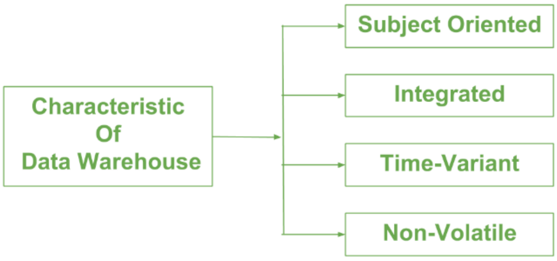
INDEX

PROGRAM 1:

To understand the basic features of Data Warehousing.

THEORY:

A data warehouse is a centralized repository for storing and managing large amounts of data from various sources for analysis and reporting. It is optimized for fast querying and analysis, enabling organizations to make informed decisions by providing a single source of truth for data. Data warehousing typically involves transforming and integrating data from multiple sources into a unified, organized, and consistent format. Below are major **characteristics** of data warehouse :



- **Subject-oriented:** A data warehouse focuses on specific themes like sales or marketing, delivering information tailored to these themes rather than current operations. It eliminates irrelevant data to facilitate precise decision-making.
- **Integrated:** Data integration ensures that data from different sources is combined into a reliable format, allowing effective analysis. Integration involves establishing shared entities and adhering to consistent naming conventions and encoding structures.
- **Time-Variant:** Data in a warehouse is maintained over different time intervals, providing historical perspectives for analysis. Once stored, data cannot be modified, ensuring consistency, and facilitating analysis over time.
- **Non-Volatile:** Data in a warehouse is permanent and not erased or updated when new data is inserted. It is read-only and refreshed at intervals, enabling analysis of historical data without the need for transaction processing or concurrency control.

Functions of Data warehouse:

- **Data Consolidation:** Combining multiple data sources into a single repository in a data warehouse for consistency.
- **Data Cleaning:** Identifying and removing errors and inconsistencies from data sources to ensure accuracy.
- **Data Integration:** Combining data from various sources into a unified repository with consistent formatting.
- **Data Storage:** Storing large volumes of historical data for easy access and analysis.
- **Data Transformation:** Cleaning and transforming data to remove inconsistencies and irrelevant information.
- **Data Analysis:** Analysing and visualizing data to extract insights and inform decision-making.
- **Data Reporting**: Providing reports and dashboards for different departments and stakeholders.
- **Data Mining:** Extracting patterns and trends from data to support strategic planning.
- **Performance Optimization:** Optimizing data warehouse systems for fast querying and analysis.

PROGRAM 2:

Explore WEKA Data Mining/Machine Learning Toolkit:

(c) Downloading and/or installation of WEKA data mining toolkit.
(d) Understand the features of WEKA tool kit such as Explorer, Knowledge flow interface, Experimenter, command-line interface.
(e) Navigate the options available in the WEKA (deselect attributes panel, preprocess panel, classify panel, cluster panel, associate panel and visualize).
(f) Study the ARFF file format.

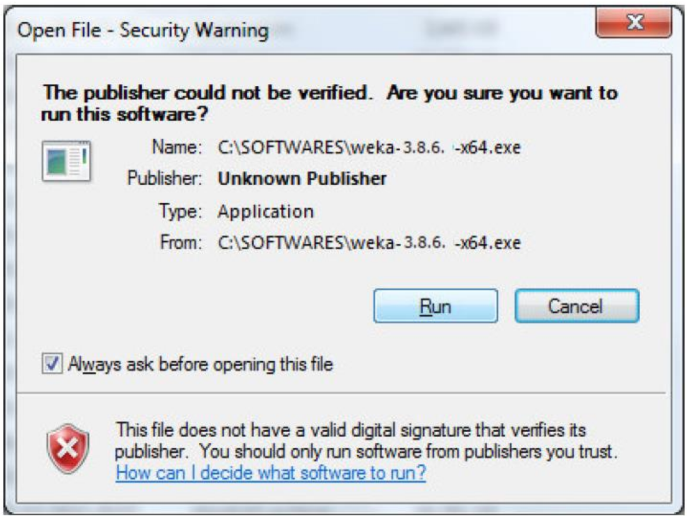
THEORY:

(a) WEKA - an open-source software provides tools for data preprocessing, implementation of several Machine Learning algorithms, and visualization tools so that you can develop machine learning techniques and apply them to real-world data mining problems. How to Download :-

1. **Download the software**(Weka Tool Kit) from the website.

2. The **Java is mandatory** for installation of WEKA so if you have already Java on your machine then download only WEKA else download the software with JVM.

3. Then open the file location and **double click** on the file. Click on **Run**.



Open File - Security Warning

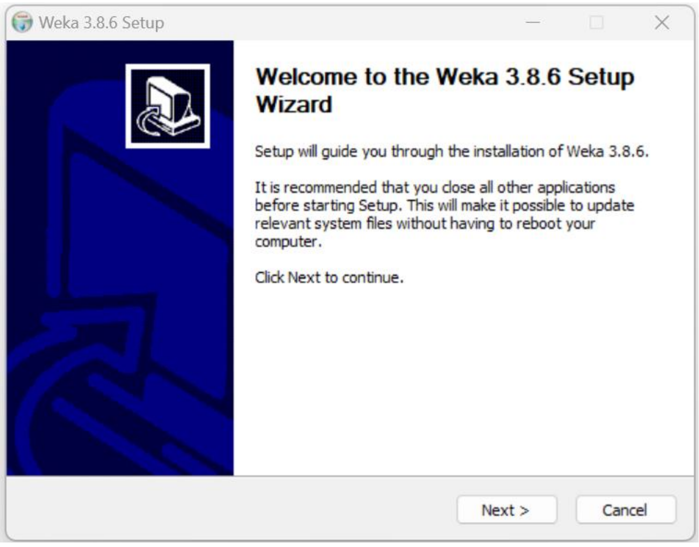
The publisher could not be verified. Are you sure you want to run this software?

Name: C:\SOFTWARES\weka-3.8.6. -x64.exe
Publisher: Unknown Publisher
Type: Application
From: C:\SOFTWARES\weka-3.8.6. -x64.exe

☑ Always ask before opening this file

This file does not have a valid digital signature that verifies its publisher. You should only run software from publishers you trust. How can I decide what software to run?

4. Click **Next** to continue.



Weka 3.8.6 Setup

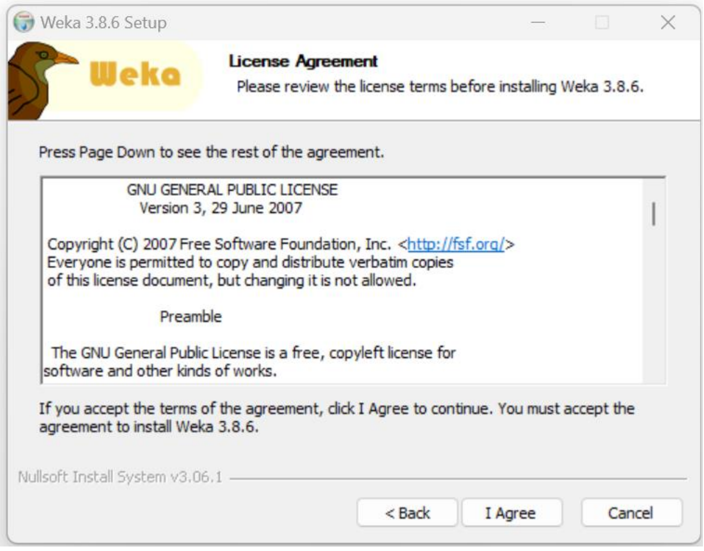
Welcome to the Weka 3.8.6 Setup Wizard

Setup will guide you through the installation of Weka 3.8.6.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

5. If you accept the terms of the agreement, click **I Agree** to continue. You must accept the agreement to install Weka.



License Agreement

Please review the license terms before installing Weka 3.8.6.

Press Page Down to see the rest of the agreement.

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

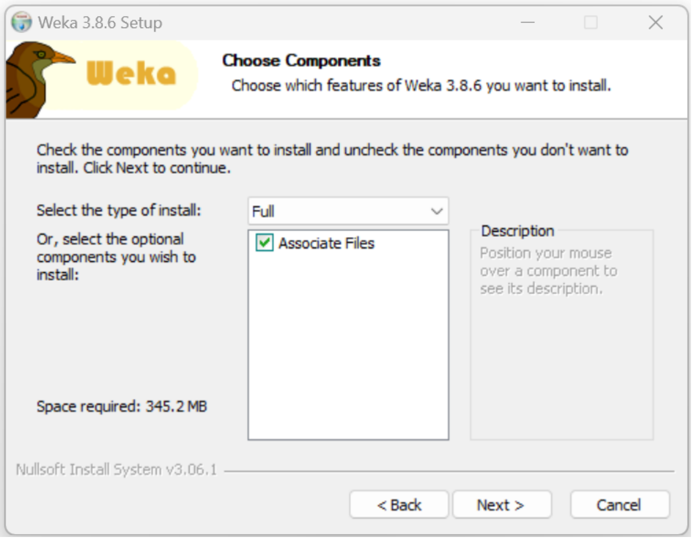
Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for
software and other kinds of works.

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Weka 3.8.6.

6. Check the components you want to install and uncheck the components you do not want to install. Click **Next** to continue.



Choose Components

Choose which features of Weka 3.8.6 you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select the type of install: Full

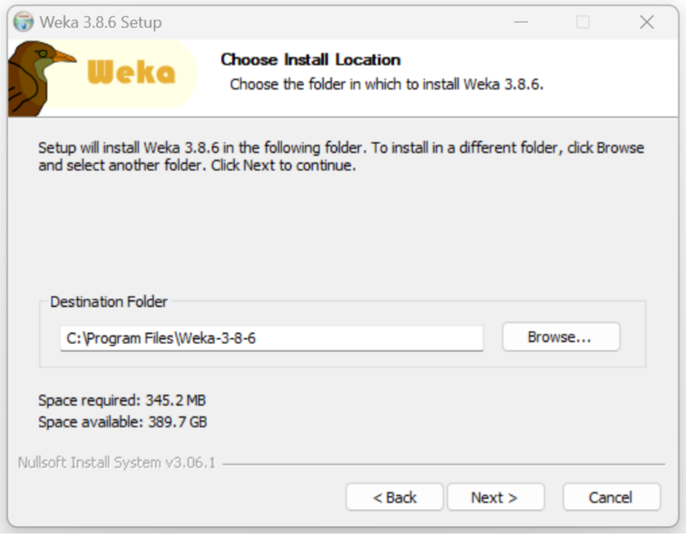
Or, select the optional components you wish to install: ☑ Associate Files

Description

Position your mouse over a component to see its description.

Space required: 345.2 MB

7. Setup will install Weka 3.8.6 in the following folder. To install in a different folder, click Browse and select another folder. Click **Next** to continue.



Choose Install Location

Choose the folder in which to install Weka 3.8.6.

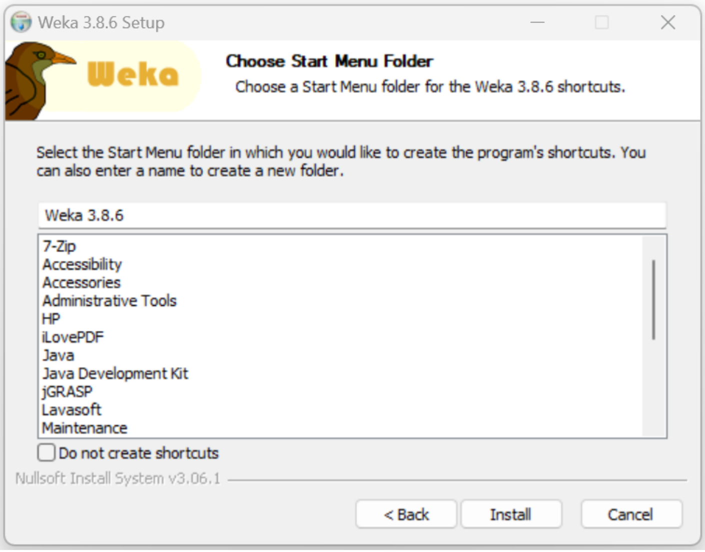
Setup will install Weka 3.8.6 in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder

C:\Program Files\Weka-3-8-6

Space required: 345.2 MB
Space available: 389.7 GB

8. Choose a Start Menu folder in which you would like to create the Weka 3.8.6 shortcuts. You can also enter a name to create a new folder. And then click on **Install** to continue.



Choose Start Menu Folder

Choose a Start Menu folder for the Weka 3.8.6 shortcuts.

Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.

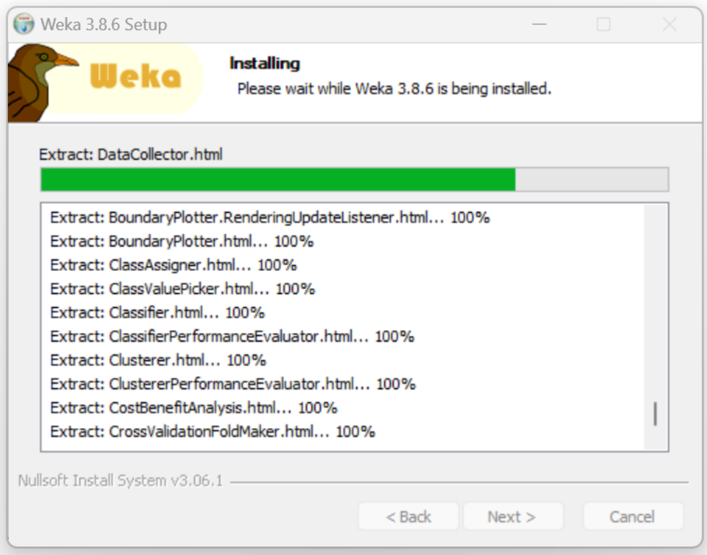
Weka 3.8.6

7-Zip
Accessibility
Accessories
Administrative Tools
HP
iLovePDF
Java
Java Development Kit
jGRASP
Lavasoft
Maintenance

□ Do not create shortcuts

Nullsoft Install System v3.06.1

9.**Installing:** Please wait while Weka 3.8.6 is being installed.



Installing

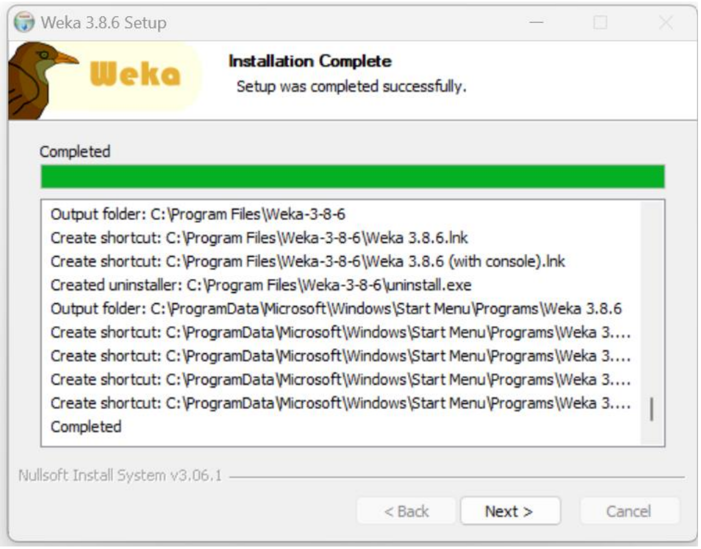
Please wait while Weka 3.8.6 is being installed.

Extract: DataCollector.html

```
Extract: BoundaryPlotter.RenderingUpdateListener.html... 100%
Extract: BoundaryPlotter.html... 100%
Extract: ClassAssigner.html... 100%
Extract: ClassValuePicker.html... 100%
Extract: Classifier.html... 100%
Extract: ClassifierPerformanceEvaluator.html... 100%
Extract: Clusterer.html... 100%
Extract: ClustererPerformanceEvaluator.html... 100%
Extract: CostBenefitAnalysis.html... 100%
Extract: CrossValidationFoldMaker.html... 100%
```

Nullsoft Install System v3.06.1

10.**Installation Complete:** Setup was completed successfully. Click **Next** to continue.



Installation Complete

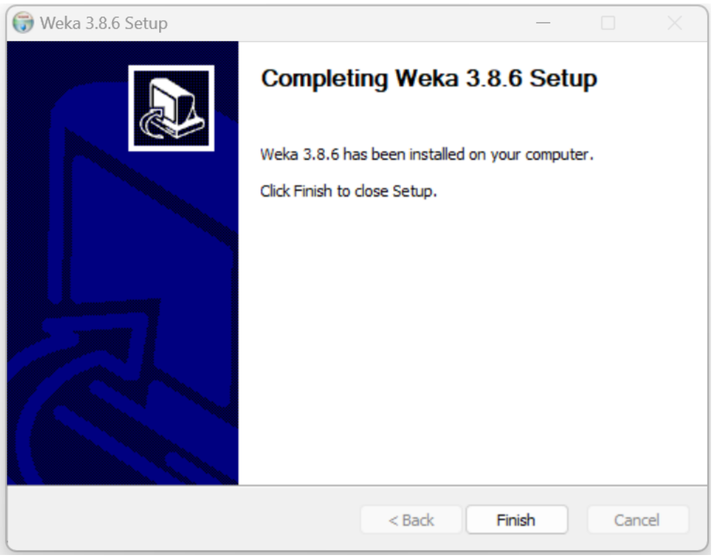
Setup was completed successfully.

Completed

```
Output folder: C:\Program Files\Weka-3-8-6
Create shortcut: C:\Program Files\Weka-3-8-6\Weka 3.8.6.lnk
Create shortcut: C:\Program Files\Weka-3-8-6\Weka 3.8.6 (with console).lnk
Created uninstaller: C:\Program Files\Weka-3-8-6\uninstall.exe
Output folder: C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Weka 3.8.6
Create shortcut: C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Weka 3....
Create shortcut: C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Weka 3....
Create shortcut: C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Weka 3....
Create shortcut: C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Weka 3....
Completed
```

Nullsoft Install System v3.06.1

11. Weka 3.8.6 has been installed on your computer. Click **Finish** to close Setup.


Weka 3.8.6 Setup

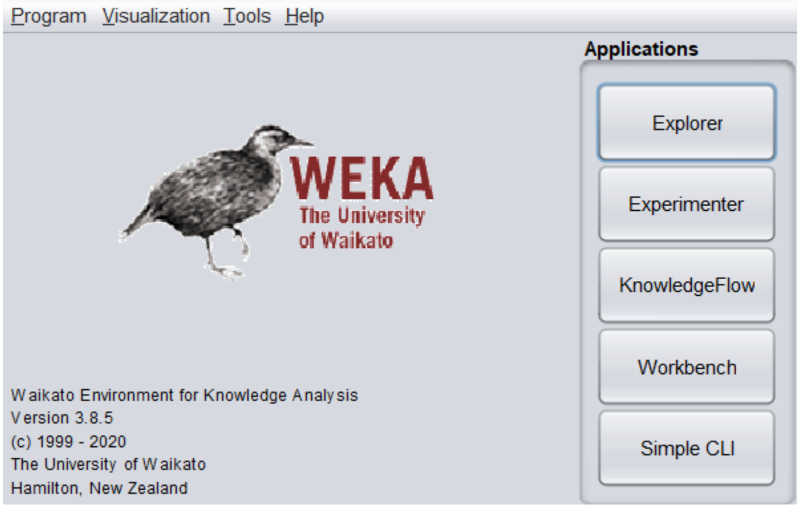
Completing Weka 3.8.6 Setup

Weka 3.8.6 has been installed on your computer.

Click Finish to close Setup.

(b) Here are the features of some of its key components:

- **Explorer:** WEKA's graphical user interface (GUI) for data preprocessing, modelling, and evaluation, offering tools for visualization, classification, clustering, and more.
- **Knowledge Flow Interface:** Enables users to design and execute machine learning workflows graphically, creating data processing pipelines with components like loaders, filters, classifiers, and evaluators.
- **Experimenter:** Facilitates systematic comparison and evaluation of machine learning algorithms across multiple datasets, automating experiment execution, result collection, and statistical analysis.
- **Command-Line Interface (CLI):** Allows users to perform WEKA operations via text-based commands, catering to scripting, batch processing, and integration into workflows for advanced users.

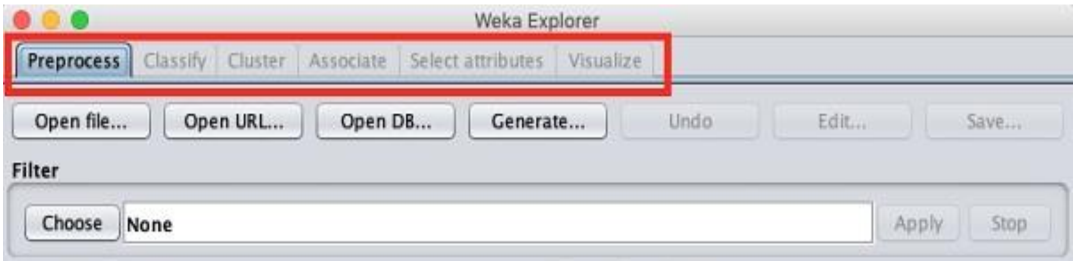


Waikato Environment for Knowledge Analysis
Version 3.8.5
(c) 1999 - 2020
The University of Waikato
Hamilton, New Zealand

(c) Here's a brief overview of the options available in the WEKA Explorer for each panel:

- **Deselect Attributes Panel:** Allows users to manually select or deselect specific attributes from the dataset based on criteria like relevance or importance.
- **Preprocess Panel:** Enables various data preprocessing tasks such as handling missing values, normalization, discretization, and attribute selection to prepare the dataset for analysis.
- **Classify Panel:** Allows users to apply machine learning algorithms for classification tasks, selecting classifiers, setting parameters, and evaluating performance metrics.

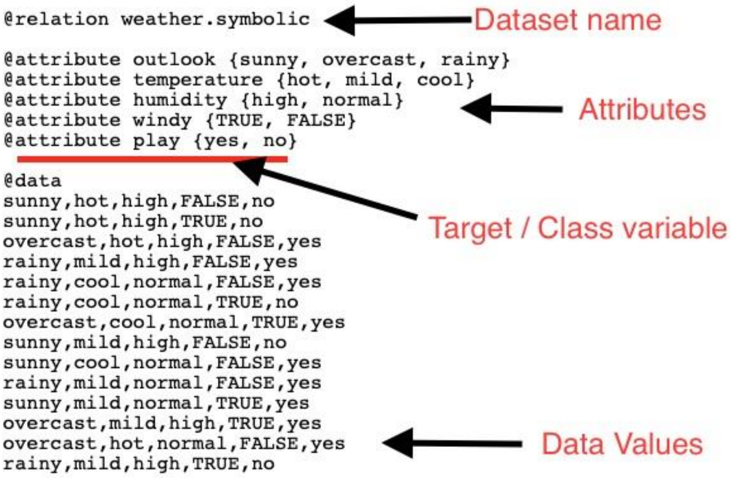
- **Cluster Panel**: Enables the application of clustering algorithms to group similar instances in the dataset, with options to select algorithms, specify parameters, and visualize resulting clusters.
- **Associate Panel:** Used for association rule mining, discovering relationships between variables in large datasets, with options to specify parameters and view generated rules and metrics.
- **Visualize Panel:** Provides visualization tools for exploring and understanding the dataset and analysis results, including attribute distributions, scatter plots, decision boundaries, cluster assignments, and association rules.



(d) An Arff file contains two sections - header and data.

- The header describes the attribute types.
- The data section contains a comma separated list of data.

As an example for Arff format, the Weather data file loaded from the WEKA sample databases is shown below:



```
@relation weather.symbolic

@attribute outlook {sunny, overcast, rainy}
@attribute temperature {hot, mild, cool}
@attribute humidity {high, normal}
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,hot,high,FALSE,no
sunny,hot,high,TRUE,no
overcast,hot,high,FALSE,yes
rainy,mild,high,FALSE,yes
rainy,cool,normal,FALSE,yes
rainy,cool,normal,TRUE,no
overcast,cool,normal,TRUE,yes
sunny,mild,high,FALSE,no
sunny,cool,normal,FALSE,yes
rainy,mild,normal,FALSE,yes
sunny,mild,normal,TRUE,yes
overcast,mild,high,TRUE,yes
overcast,hot,normal,FALSE,yes
rainy,mild,high,TRUE,no
```

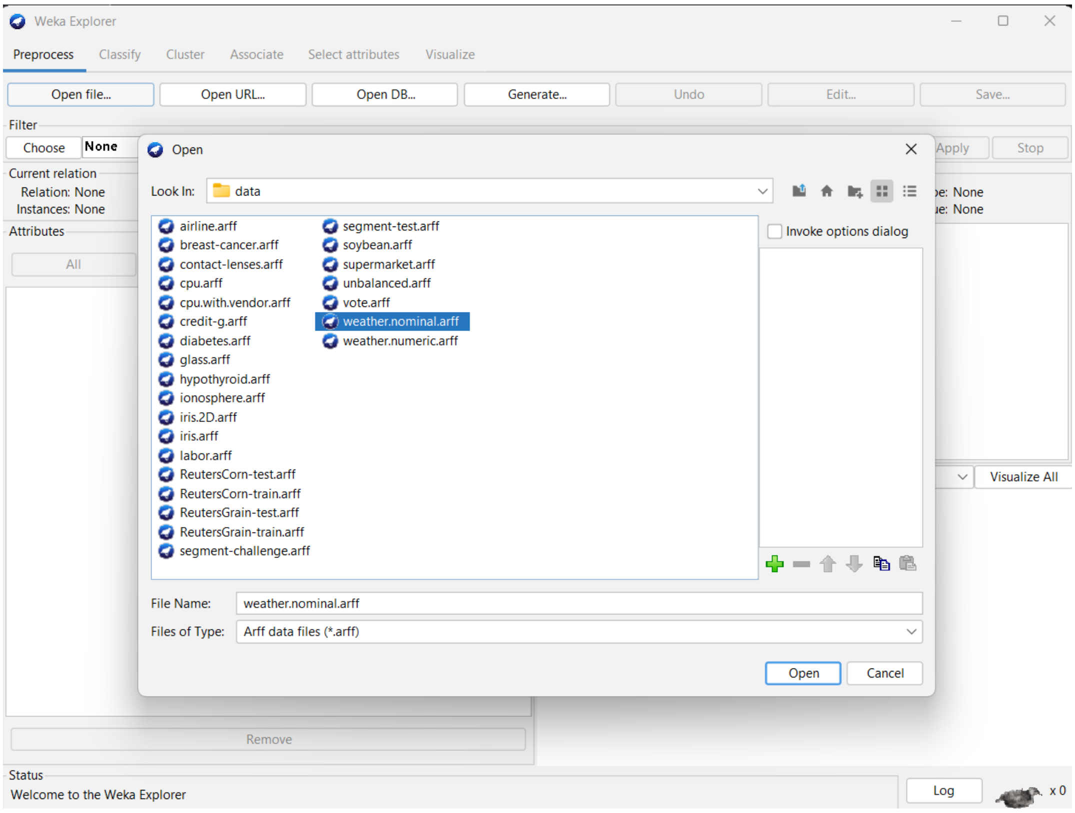
- The @relation tag defines the name of the database.
- The @attribute tag defines the attributes.
- The @data tag starts the list of data rows each containing the comma separated fields.

PROGRAM 3:

To understand the working of datasets in WEKA and to perform demonstration of preprocessing on dataset weather.arff.

WORKING:

Using the Open file ... option under the Preprocess tag select the weather-nominal.arff file.



Relation: None
Instances: None

Look In: data

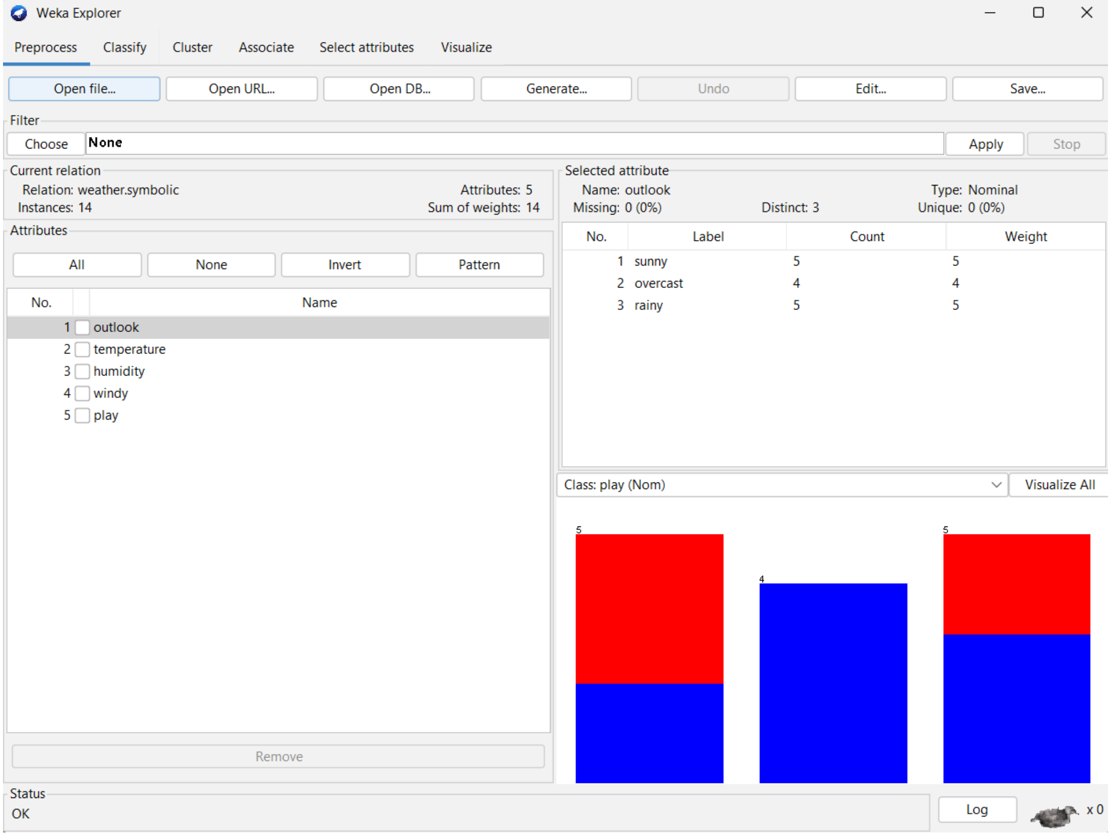
airline.arff	segment-test.arff
breast-cancer.arff	soybean.arff
contact-lenses.arff	supermarket.arff
cpu.arff	unbalanced.arff
cpu.with.vendor.arff	vote.arff
credit-g.arff	weather.nominal.arff
diabetes.arff	weather.numeric.arff
glass.arff	
hypothyroid.arff	
ionosphere.arff	
iris.2D.arff	
iris.arff	
labor.arff	
ReutersCorn-test.arff	
ReutersCorn-train.arff	
ReutersGrain-test.arff	
ReutersGrain-train.arff	
segment-challenge.arff	

File Name: weather.nominal.arff

Files of Type: Arff data files (*.arff)

Status
Welcome to the Weka Explorer

When you open the file, your screen looks like as shown here –



Current relation
Relation: weather.symbolic
Instances: 14
Attributes: 5
Sum of weights: 14

Attributes

No.		Name
1	□	outlook
2	□	temperature
3	□	humidity
4	□	windy
5	□	play

Selected attribute
Name: outlook
Missing: 0 (0%)
Distinct: 3
Type: Nominal
Unique: 0 (0%)

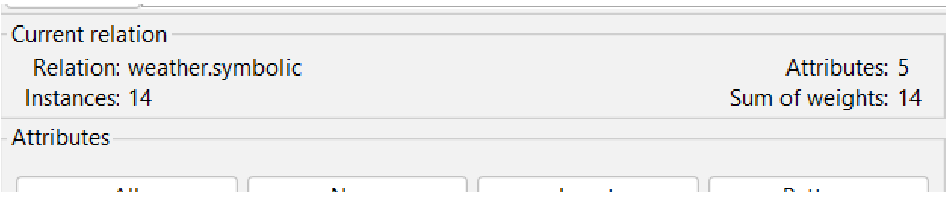
No.	Label	Count	Weight
1	sunny	5	5
2	overcast	4	4
3	rainy	5	5

Class: play (Nom)

Status
OK

Let us first look at the highlighted **Current relation** sub window. It shows the name of the database that is currently loaded. You can infer two points from this sub window −

- There are 14 instances - the number of rows in the table.
- The table contains 5 attributes - the fields, which are discussed in the upcoming sections.

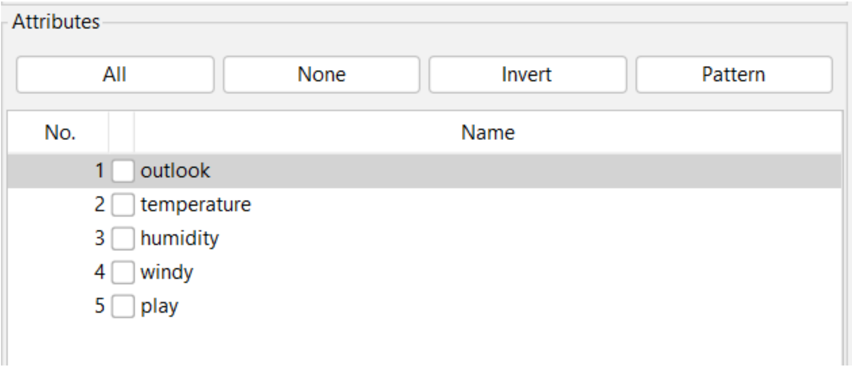


Current relation

Relation: weather.symbolic   Attributes: 5

Instances: 14   Sum of weights: 14

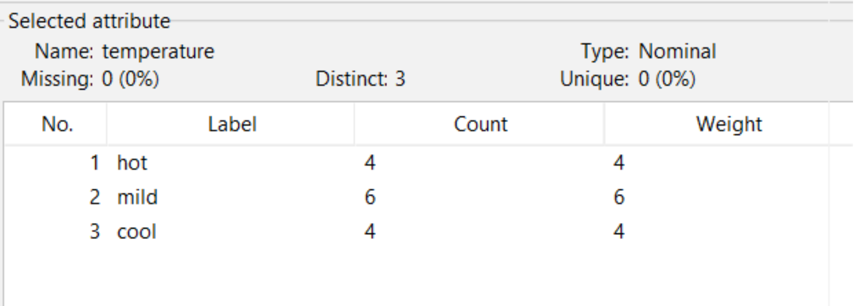
On the left side, notice the **Attributes** sub window that displays the various fields in the database.



No.		Name
1	□	outlook
2	□	temperature
3	□	humidity
4	□	windy
5	□	play

The **weather** database contains five fields - outlook, temperature, humidity, windy and play. When you select an attribute from this list by clicking on it, further details on the attribute itself are displayed on the right-hand side.

Let us select the temperature attribute first. When you click on it, you would see the following screen −



Selected attribute

Name: temperature   Type: Nominal

Missing: 0 (0%)   Distinct: 3   Unique: 0 (0%)

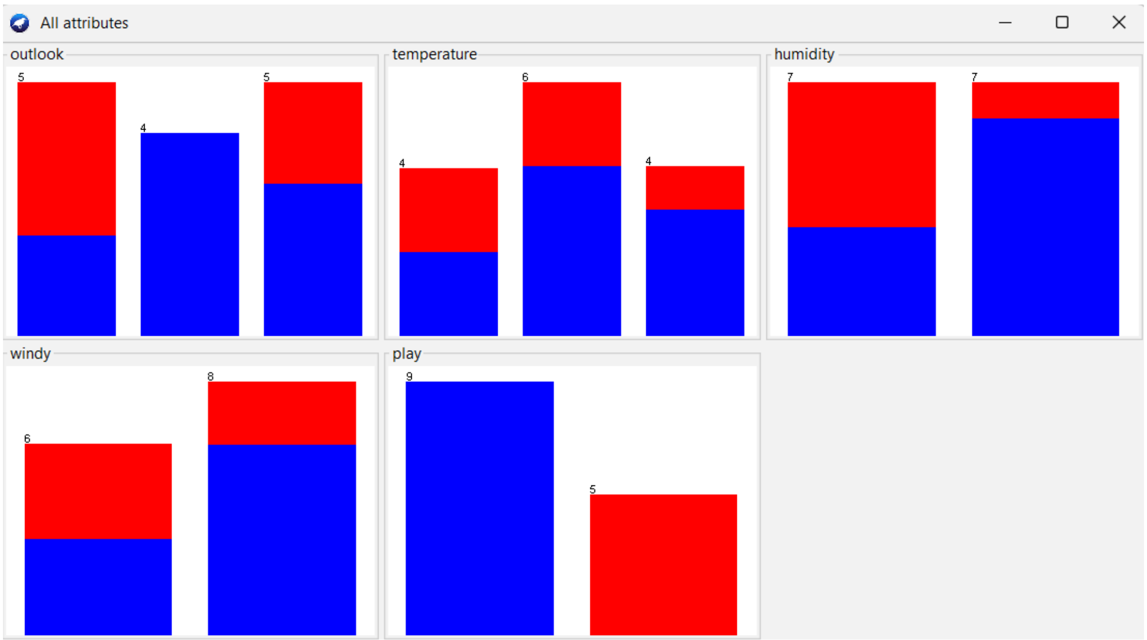
No.	Label	Count	Weight
1	hot	4	4
2	mild	6	6
3	cool	4	4

In the **Selected Attribute** sub window, you can observe the following −

- The name and the type of the attribute are displayed.
- The type for the **temperature** attribute is **Nominal**.
- The number of **Missing** values is zero.
- There are three distinct values with no unique value.
- The table underneath this information shows the nominal values for this field as hot, mild, and cold.
- It also shows the count and weight in terms of a percentage for each nominal value.

At the bottom of the window, you see the visual representation of the **class** values.

If you click on the **Visualize All** button, you will be able to see all features in one single window as shown here −


All attributes
outlook
5
4
5
temperature
4
6
4
humidity
7
7
windy
6
8
play
9
5

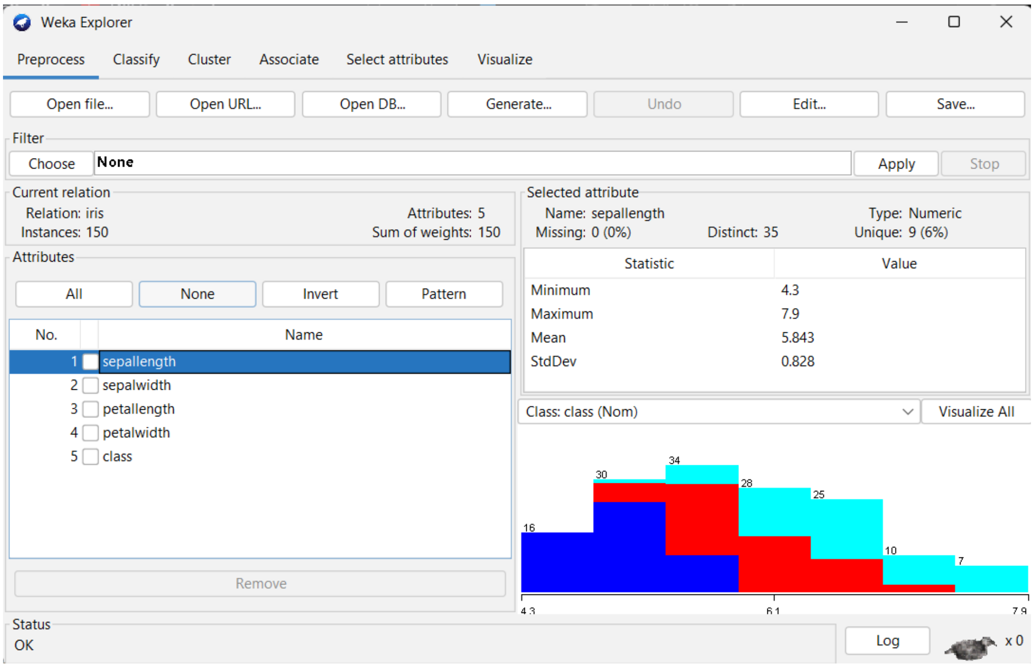
PROGRAM 4:

To apply Numeric Transform (data preprocessing step) on Iris Dataset.

WORKING:

To apply Numeric Transform as a data preprocessing step on the Iris dataset in WEKA, you can follow these steps:

1. **Load the Dataset**: Open WEKA and navigate to the "Explorer" interface.
2. **Open the Iris Dataset**: Click on the "Open file..." button and select the Iris dataset file. The Iris dataset is a commonly used dataset available in the "data" folder of WEKA's installation directory.
3. **Explore the Dataset**: Switch to the "Preprocess" panel to view the dataset attributes and instances. Ensure that you have a clear understanding of the dataset's structure and contents before proceeding with preprocessing.



Filter: None

Current relation
Relation: iris
Instances: 150
Attributes: 5
Sum of weights: 150

Attributes

No.	Name
1	sepallength
2	sepalwidth
3	petallength
4	petalwidth
5	class

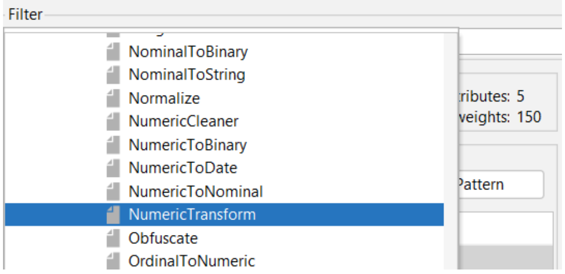
Selected attribute
Name: sepallength
Missing: 0 (0%)
Distinct: 35
Type: Numeric
Unique: 9 (6%)

Statistic	Value
Minimum	4.3
Maximum	7.9
Mean	5.843
StdDev	0.828

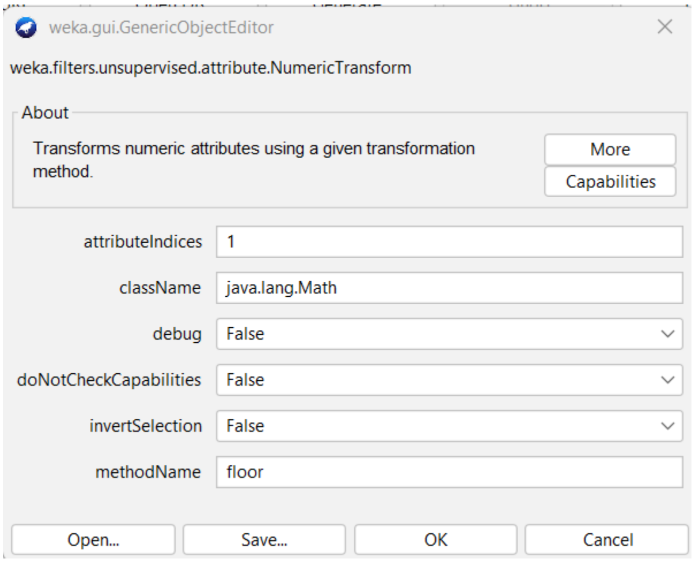
Class: class (Nom)

Status
OK

4. **Apply Numeric Transform**: To apply a Numeric Transform, follow these steps:
   - Select the "NumericTransform" filter from the "Filters" panel on the left-hand side.
   - Drag and drop the NumericTransform filter onto the preprocessing pipeline area in the middle of the interface.



   - Configure the NumericTransform filter by double-clicking on it in the preprocessing pipeline. Specify the transformation options you want to apply to the numeric attributes in the dataset.
   - Common transformations include standardization (mean normalization and scaling), normalization to a specific range, or any custom transformation you may require.



weka.filters.unsupervised.attribute.NumericTransform

About

Transforms numeric attributes using a given transformation method.

attributeIndices: 1

className: java.lang.Math

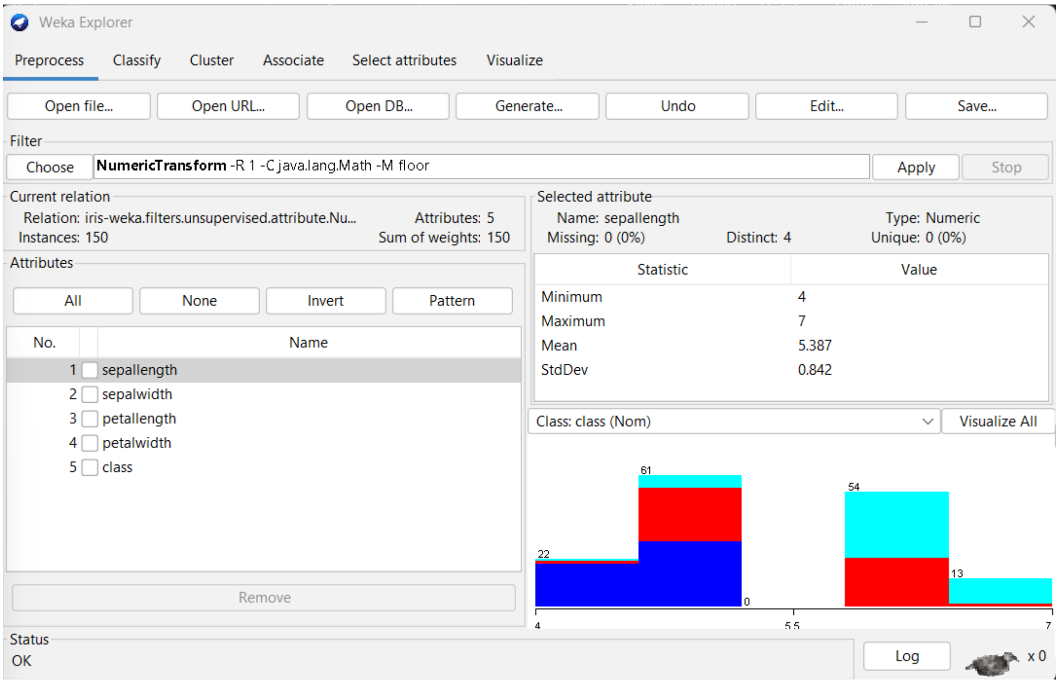
debug: False

doNotCheckCapabilities: False

invertSelection: False

methodName: floor

- Click on the "Apply" button to apply the Numeric Transform filter to the dataset.



NumericTransform -R 1 -C java.lang.Math -M floor

Current relation

Relation: iris-weka.filters.unsupervised.attribute.Nu... Attributes: 5

Instances: 150 Sum of weights: 150

Selected attribute

Name: sepallength Type: Numeric

Missing: 0 (0%) Distinct: 4 Unique: 0 (0%)

Statistic	Value
Minimum	4
Maximum	7
Mean	5.387
StdDev	0.842

Attributes

No.	Name
1	sepallength
2	sepalwidth
3	petallength
4	petalwidth
5	class

PROGRAM 5:

To understand the importance of CSV data and then load student academic record (CSV format) in Weka.

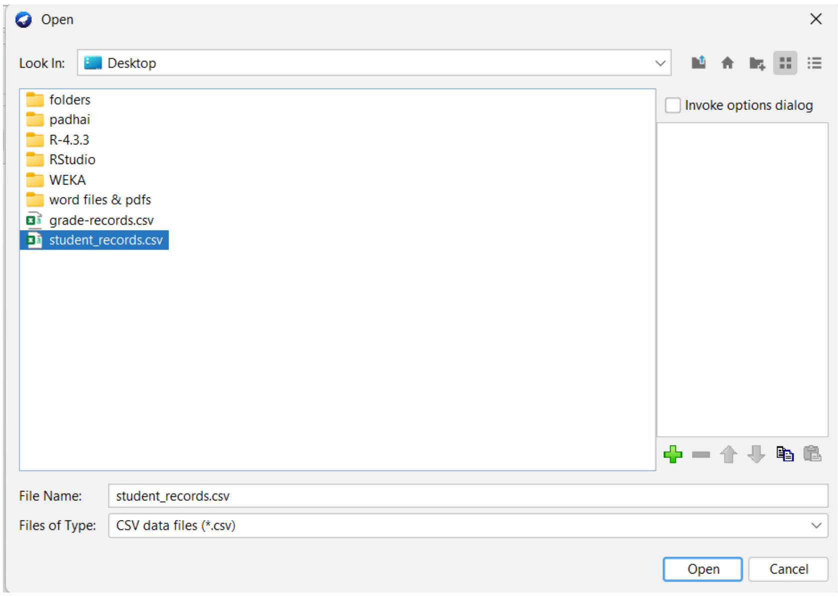
THOERY:

CSV (comma-separated values) files are a common way to store tabular data, such as student academic records. They are easy to read and write and can be opened by a variety of software programs. CSV data is important because it is a standard format that can be used to exchange data between different programs. This makes it easy to share data with others, and to collaborate on projects. CSV data is also easy to store and archive, making it a good choice for long-term data storage.

WORKING:

To load a student academic record (CSV format) in Weka, you can use the following steps:

1. Open the Weka Explorer, Click the "Open file" button, Select the CSV file that you want to load, and click the "Open" button.



Look In: Desktop

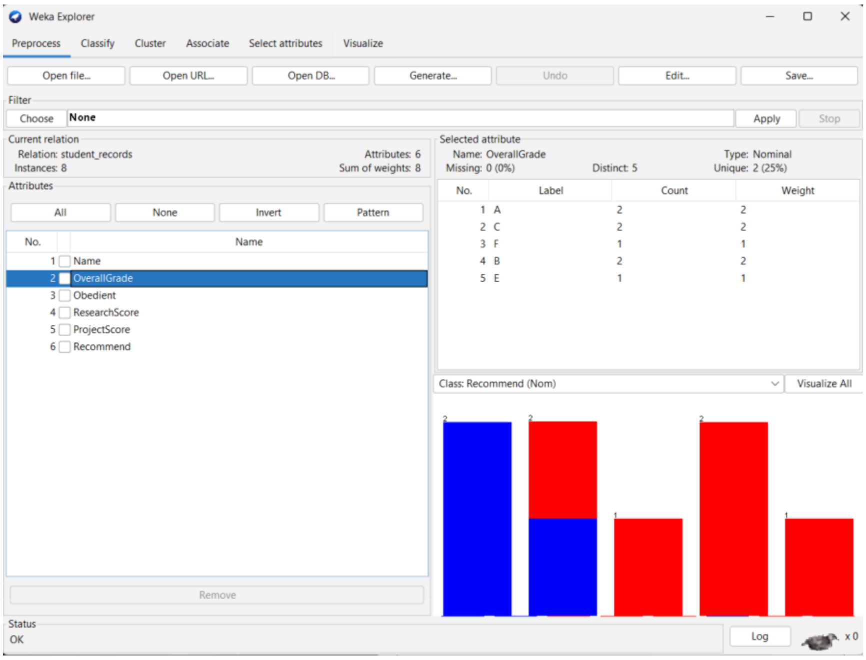
folders
padhai
R-4.3.3
RStudio
WEKA
word files & pdfs
grade-records.csv
student_records.csv

□ Invoke options dialog

File Name: student_records.csv

Files of Type: CSV data files (*.csv)

2. The CSV file will be loaded into Weka, and you can view it in the "Preprocess" panel. You can then use the data in the CSV file to create a machine learning model, or to perform other data analysis tasks.



Filter: Choose None

Current relation
Relation: student_records   Attributes: 6
Instances: 8   Sum of weights: 8

Attributes

No.	Name
1	□ Name
2	□ OverallGrade
3	□ Obedient
4	□ ResearchScore
5	□ ProjectScore
6	□ Recommend

Selected attribute
Name: OverallGrade   Type: Nominal
Missing: 0 (0%)   Distinct: 5   Unique: 2 (25%)

No.	Label	Count	Weight
1	A	2	2
2	C	2	2
3	F	1	1
4	B	2	2
5	E	1	1

Class: Recommend (Nom)

Status
OK

PROGRAM 6:

To perform decision tree classification using J48 algorithm on weather.arff.

THOERY:

Decision trees are also known as **Classification And Regression Trees (CART)**. They work by learning answers to a hierarchy of if/else questions leading to a decision. These questions form a tree-like structure, and hence the name.

Each node in the tree represents a question derived from the features present in your dataset. Your dataset is split based on these questions until the maximum depth of the tree is reached. The last node does not ask a question but represents which class the value belongs to.

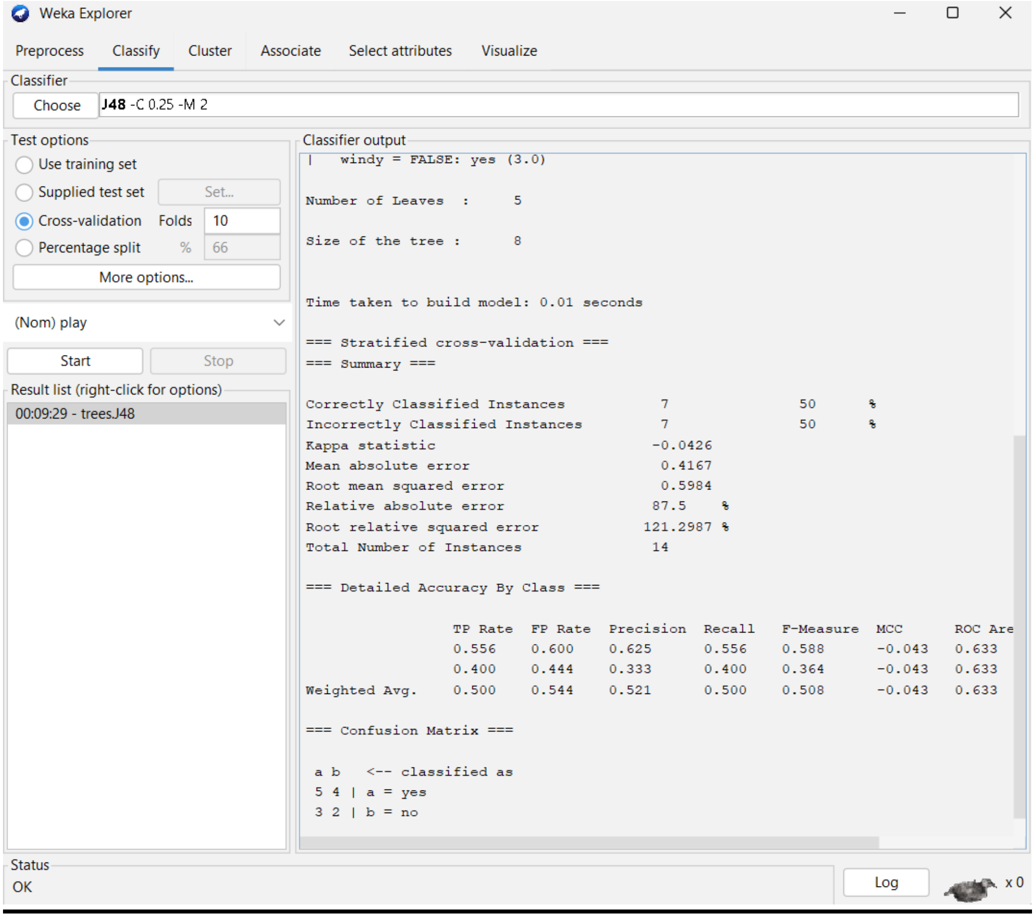
- The topmost node in the Decision tree is called the **Root node.**
- The bottom-most node is called the **Leaf node.**
- A node divided into sub-nodes is called a **Parent node.** The sub-nodes are called **Child nodes.**

J48 is a decision tree algorithm implemented in WEKA, which is based on the C4.5 algorithm developed by Ross Quinlan.

WORKING:

To perform decision tree classification using the J48 algorithm on the "weather.arff" dataset in WEKA, you can follow these steps:

1. Load the Dataset: Open WEKA and navigate to the "Explorer" interface.
2. Open the "weather.arff" Dataset: Click on the "Open file..." button and select the "weather.arff" dataset file.
3. Choose the Class Attribute: In the "Preprocess" panel, ensure that the class attribute (the attribute you want to predict) is correctly set. For the "weather.arff" dataset, the class attribute is likely to be something like "play," indicating whether or not to play outside based on weather conditions.
4. Select the Classify Panel: Switch to the "Classify" panel in WEKA.
5. Choose J48 Algorithm: In the "Classifier" section of the panel, click on the dropdown menu to select the J48 algorithm. J48 is WEKA's implementation of the C4.5 decision tree algorithm.
6. Set Options (if needed): Optionally, you can set specific options for the J48 algorithm by clicking on the "More options..." button. This allows you to customize parameters such as the confidence factor, minimum number of instances per leaf, and pruning options.
7. Run the Classifier: Once you have selected the J48 algorithm and set any desired options, click on the "Start" button to run the classifier. WEKA will train the decision tree classifier on the "weather.arff" dataset using the J48 algorithm.
8. Evaluate the Model: After the classifier has been trained, WEKA will display evaluation results in the "Classifier output" section. This includes metrics such as accuracy, precision, recall, and F-measure. You can use these metrics to assess the performance of the decision tree classifier on the dataset.



Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Classifier

Choose **J48** -C 0.25 -M 2

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)

00:09:29 - trees.J48

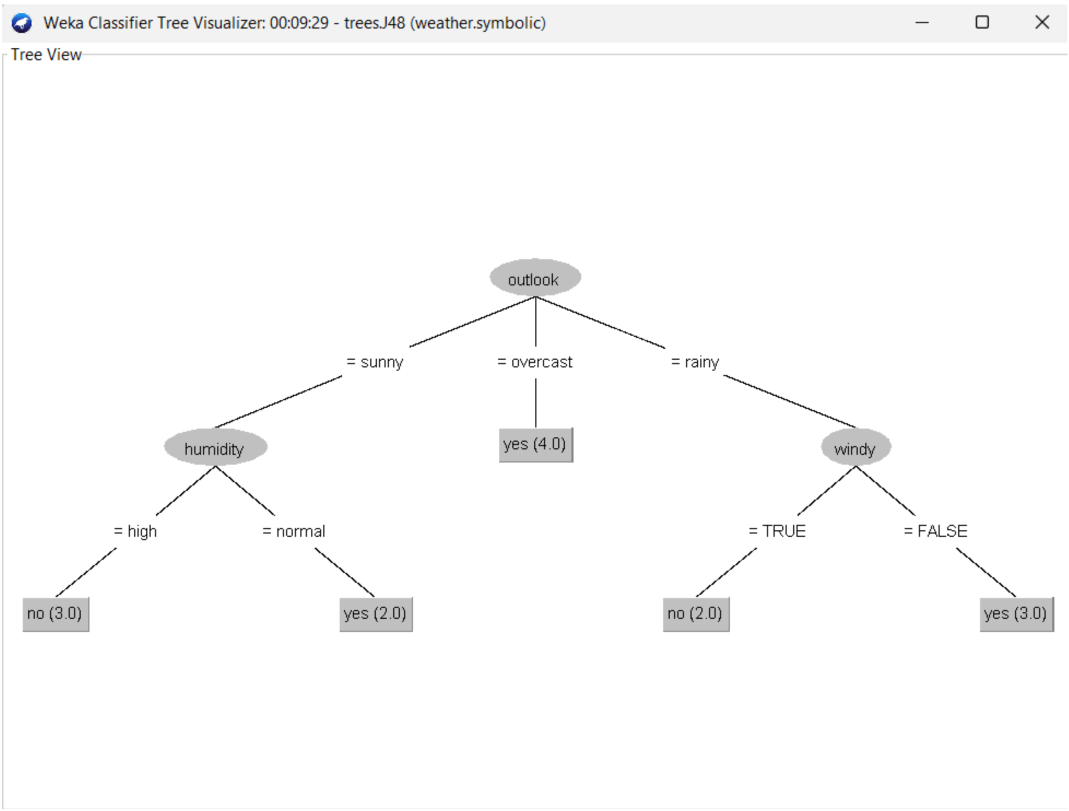
Classifier output

```
|   windy = FALSE: yes (3.0)

Number of Leaves  :     5

Size of the tree :      8


Time taken to build model: 0.01 seconds

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

Correctly Classified Instances           7               50      %
Incorrectly Classified Instances         7               50      %
Kappa statistic                         -0.0426
Mean absolute error                      0.4167
Root mean squared error                  0.5984
Relative absolute error                 87.5    %
Root relative squared error            121.2987 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Are
                 0.556    0.600    0.625      0.556    0.588      -0.043   0.633
                 0.400    0.444    0.333      0.400    0.364      -0.043   0.633
Weighted Avg.    0.500    0.544    0.521      0.500    0.508      -0.043   0.633

=== Confusion Matrix ===

 a b   <-- classified as
 5 4 | a = yes
 3 2 | b = no
```

Status

OK

9. Visualize the Decision Tree (Optional): If you want to visualize the decision tree that was learned by the J48 algorithm, you can click on the "Visualize tree" button. This will display a graphical representation of the decision tree, allowing you to explore its structure and decision rules.



Weka Classifier Tree Visualizer: 00:09:29 - trees.J48 (weather.symbolic)

PROGRAM 7:

Demonstration of classification rule process on dataset employee.arff using naïve bayes algorithm.

THOERY:

Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem, with an assumption of feature independence.

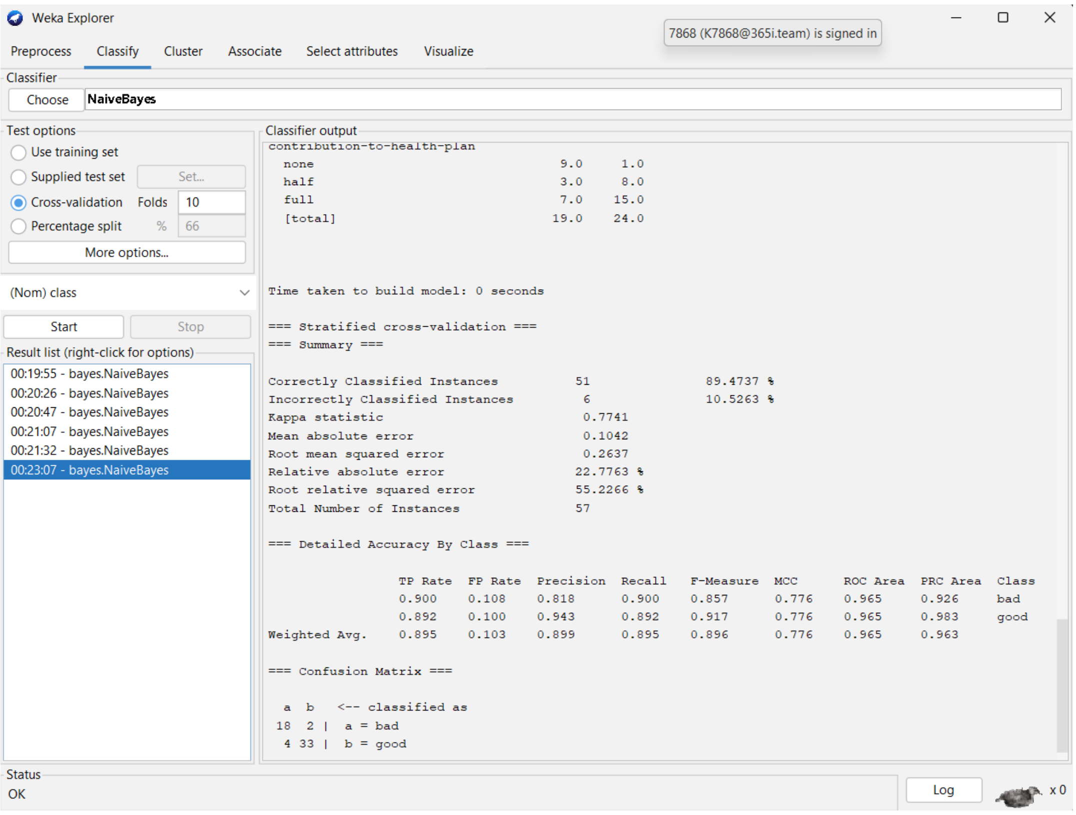
It calculates the probability of each class given a set of input features and selects the class with the highest probability as the predicted class. Despite its simplicity, Naive Bayes often performs well in practice, especially on text **classification** tasks and datasets with high-dimensional features.

It is computationally efficient and requires minimal parameter tuning, making it suitable for large datasets and real-time applications. However, its assumption of feature independence may not hold true in all datasets, potentially leading to suboptimal performance in some cases.

WORKING:

To demonstrate the classification rule process using the Naïve Bayes algorithm on the "employee.arff" dataset in WEKA, follow these steps:

1. Load the Dataset: Open WEKA and navigate to the "Explorer" interface.
2. Open the "employee.arff" Dataset: Click on the "Open file..." button and select the "employee.arff" dataset file.
3. Choose the Class Attribute: In the "Preprocess" panel, ensure that the class attribute (the attribute you want to predict) is correctly set. For the "employee.arff" dataset, the class attribute might be something like "left," indicating whether an employee left the company or not.
4. Select the Classify Panel: Switch to the "Classify" panel in WEKA.
5. Choose Naïve Bayes Algorithm: In the "Classifier" section of the panel, click on the dropdown menu and select the Naïve Bayes algorithm. Naïve Bayes is a probabilistic classifier based on Bayes' theorem, assuming independence between features.
6. Run the Classifier: Once you have selected the Naïve Bayes algorithm, click on the "Start" button to run the classifier. WEKA will train the Naïve Bayes classifier on the "employee.arff" dataset.
7. Evaluate the Model: After the classifier has been trained, WEKA will display evaluation results in the "Classifier output" section. This includes metrics such as accuracy, precision, recall, and F-measure, which assess the performance of the Naïve Bayes classifier on the dataset.
8. Interpret the Classification Rules: While Naïve Bayes does not explicitly generate decision rules like some other algorithms, you can interpret the classification process based on probability estimates. Naïve Bayes calculates the probability of each class label given the input features and assigns the class label with the highest probability as the predicted class. You can analyze the conditional probabilities of features given each class to understand how the classifier makes decisions.



Classifier

NaiveBayes

Test options

○ Use training set

○ Supplied test set

● Cross-validation Folds 10

○ Percentage split % 66

(Nom) class

Result list (right-click for options)

00:19:55 - bayes.NaiveBayes
00:20:26 - bayes.NaiveBayes
00:20:47 - bayes.NaiveBayes
00:21:07 - bayes.NaiveBayes
00:21:32 - bayes.NaiveBayes
00:23:07 - bayes.NaiveBayes

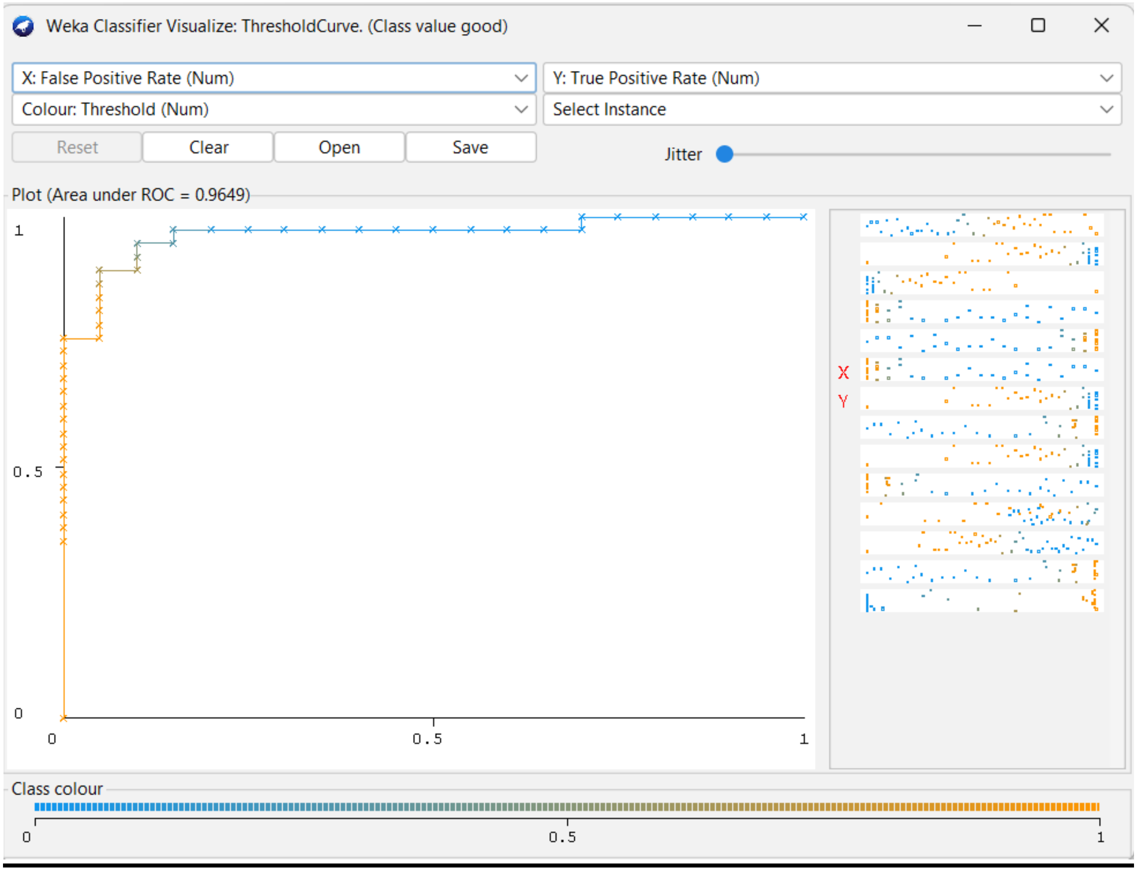
Classifier output

```
contribution-to-health-plan
  none                              9.0     1.0
  half                              3.0     8.0
  full                              7.0    15.0
  [total]                          19.0    24.0



Time taken to build model: 0 seconds

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

Correctly Classified Instances          51               89.4737 %
Incorrectly Classified Instances         6               10.5263 %
Kappa statistic                          0.7741
Mean absolute error                      0.1042
Root mean squared error                  0.2637
Relative absolute error                 22.7763 %
Root relative squared error             55.2266 %
Total Number of Instances               57

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.900    0.108    0.818      0.900    0.857      0.776    0.965     0.926     bad
                 0.892    0.100    0.943      0.892    0.917      0.776    0.965     0.983     good
Weighted Avg.    0.895    0.103    0.899      0.895    0.896      0.776    0.965     0.963

=== Confusion Matrix ===

  a  b   <-- classified as
 18  2 |  a = bad
  4 33 |  b = good
```

Status

OK

9. Visualize the Classifier Output (Optional): If you want to visualize the classification results or explore the probabilities assigned by the Naïve Bayes classifier, you can use visualization tools provided by WEKA.


Weka Classifier Visualize: ThresholdCurve. (Class value good)
X: False Positive Rate (Num)
Y: True Positive Rate (Num)
Colour: Threshold (Num)
Select Instance
Reset
Clear
Open
Save
Jitter
Plot (Area under ROC = 0.9649)
1
0.5
0
0
0.5
1
X
Y
Class colour
0
0.5
1

PROGRAM 8:

To understand the concept of discretization and to perform discretization on the dataset airline.arff.

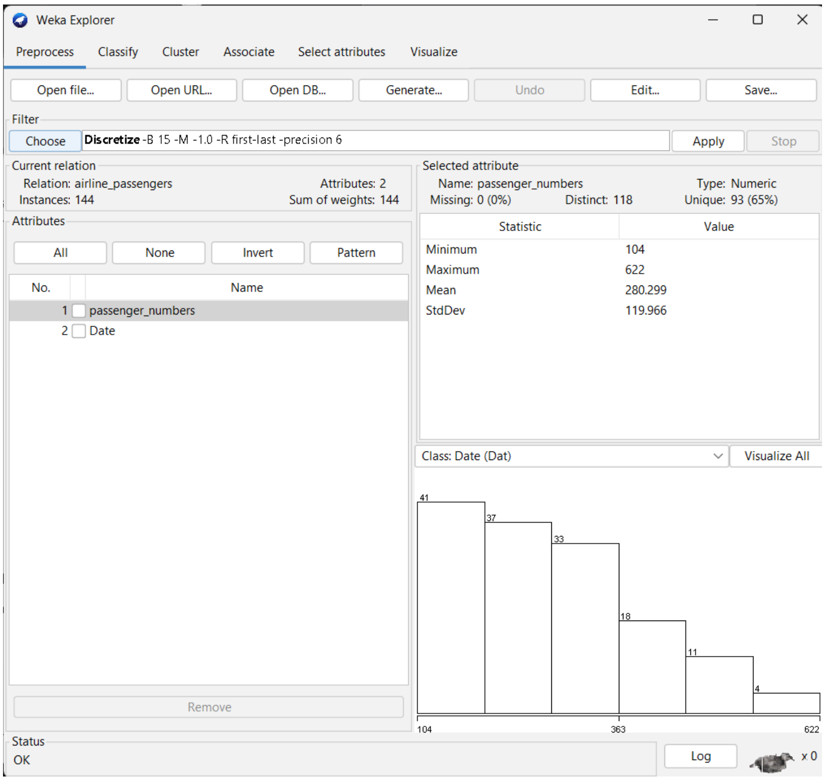
THOERY:

Discretization is the process of converting continuous data into discrete intervals or categories. This is often done to simplify analysis or to prepare data for certain types of algorithms that require categorical input. There are several methods for discretization, including equal-width binning, equal-frequency binning, and clustering-based methods.

WORKING:

Step-by-step guide to discretizing the dataset **airline.arff** in WEKA:

1. **Open WEKA**: Launch the WEKA application.
2. **Load Dataset**: Go to the "Explorer" tab, click on the "Open file" button, and select the **airline.arff** dataset.
3. **Preprocess Tab**: Switch to the "Preprocess" tab.
4. **Choose Filter**: In the Preprocess tab, you'll see a list of preprocessing filters on the left. Scroll down or type "discretize" in the search bar to find the "Discretize" filter.
5. **Apply Discretize Filter**: Drag the "Discretize" filter from the left panel to the right panel, where the preprocessing pipeline is constructed.



Filter

Discretize -B 15 -M -1.0 -R first-last -precision 6

Current relation

Relation: airline_passengers Attributes: 2
Instances: 144 Sum of weights: 144

Attributes

No.		Name
1	□	passenger_numbers
2	□	Date

Selected attribute

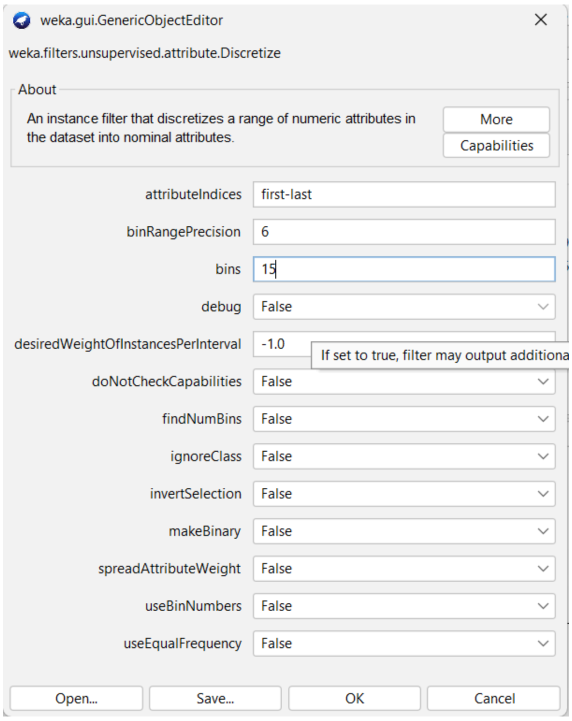
Name: passenger_numbers Type: Numeric
Missing: 0 (0%) Distinct: 118 Unique: 93 (65%)

Statistic	Value
Minimum	104
Maximum	622
Mean	280.299
StdDev	119.966

Class: Date (Dat)

Status
OK

6. **Configure Discretize Filter (Optional)**: You can click on the "Discretize" filter in the right panel to configure its parameters. For example, you can specify the number of bins or select different discretization methods.



weka.gui.GenericObjectEditor

weka.filters.unsupervised.attribute.Discretize

About

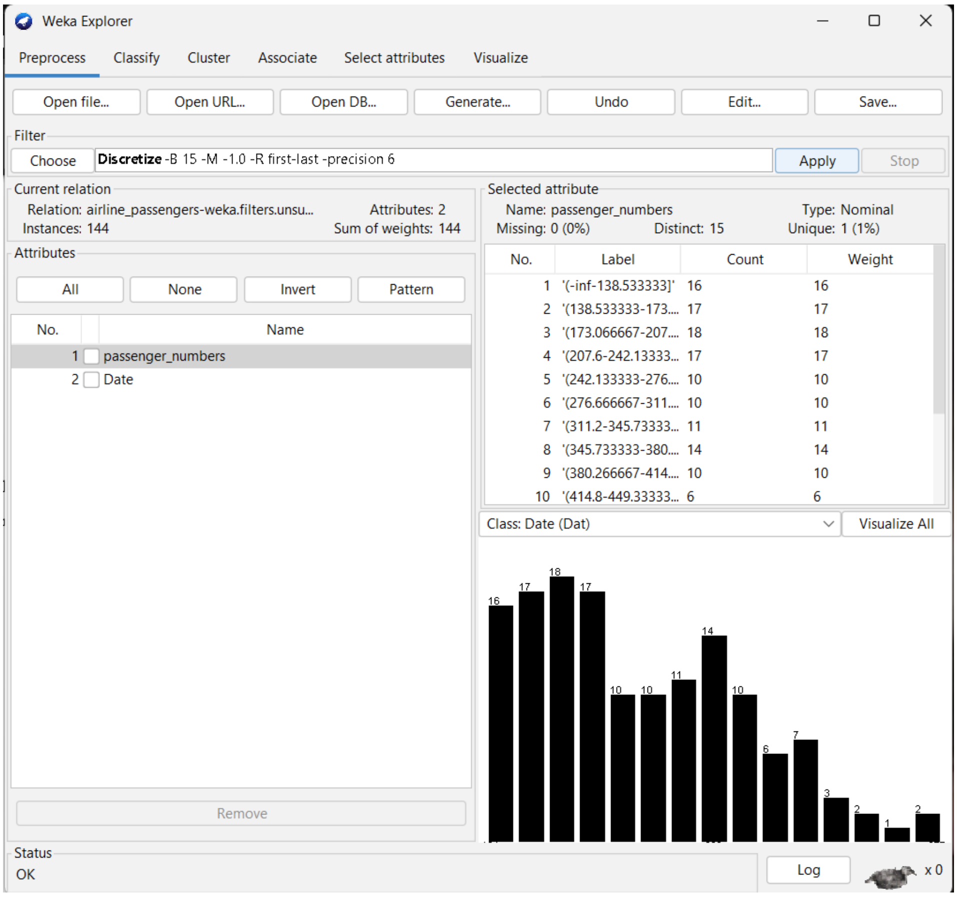
An instance filter that discretizes a range of numeric attributes in the dataset into nominal attributes.

attributeIndices: first-last
binRangePrecision: 6
bins: 15
debug: False
desiredWeightOfInstancesPerInterval: -1.0
doNotCheckCapabilities: False
findNumBins: False
ignoreClass: False
invertSelection: False
makeBinary: False
spreadAttributeWeight: False
useBinNumbers: False
useEqualFrequency: False

If set to true, filter may output additiona

7. **Run**: Click on the "Start" button to run the preprocessing pipeline with the Discretize filter.

8. **Save Results**: After the preprocessing is complete, you can save the discretized dataset by clicking on the "Save" button.



Weka Explorer

Filter: Discretize -B 15 -M -1.0 -R first-last -precision 6

Current relation
Relation: airline_passengers-weka.filters.unsu... Attributes: 2
Instances: 144 Sum of weights: 144

Attributes

No.	Name
1	□ passenger_numbers
2	□ Date

Selected attribute
Name: passenger_numbers Type: Nominal
Missing: 0 (0%) Distinct: 15 Unique: 1 (1%)

No.	Label	Count	Weight
1	'(-inf-138.533333]'	16	16
2	'(138.533333-173....	17	17
3	'(173.066667-207....	18	18
4	'(207.6-242.13333...	17	17
5	'(242.133333-276....	10	10
6	'(276.666667-311....	10	10
7	'(311.2-345.73333...	11	11
8	'(345.733333-380....	14	14
9	'(380.266667-414....	10	10
10	'(414.8-449.33333...	6	6

Class: Date (Dat)

Status
OK

PROGRAM 9:

To create Training, Validation and Test dataset for iris.arff.

THOERY:

Brief overview of training, validation, and test datasets and their roles in machine learning:

1. **Training Dataset**:

The training dataset is used to train machine learning models. It contains a labelled set of examples that the model learns from. The model adjusts its parameters based on the patterns and relationships present in the training data.

2. **Validation Dataset**:

The validation dataset is used to fine-tune the model's hyperparameters and assess its performance during training. It helps in preventing overfitting by providing an independent dataset to evaluate the model's generalization performance. After training on the training dataset, the model's performance is evaluated on the validation dataset, and hyperparameters are adjusted accordingly.

3. **Test Dataset**:

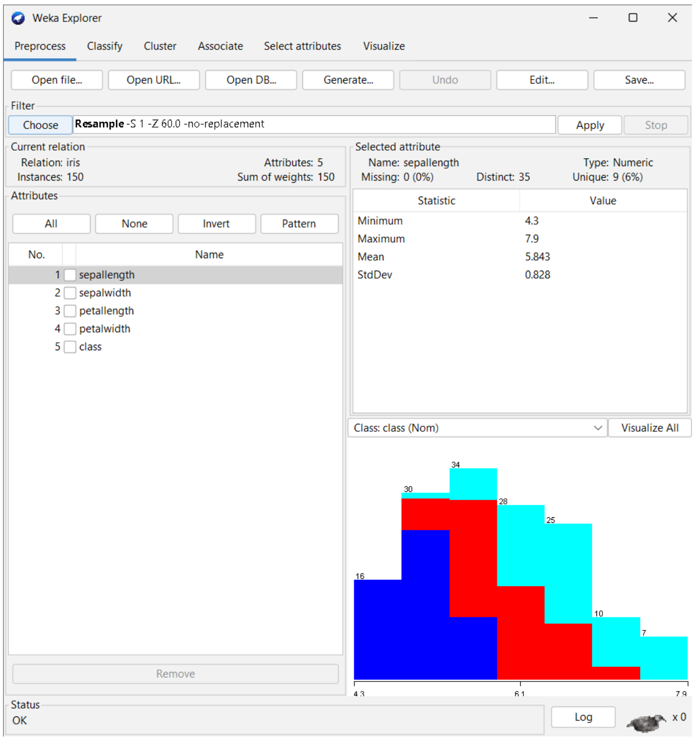
The test dataset is used to evaluate the final performance of the trained model. It provides an unbiased estimate of the model's performance on unseen data. The test dataset should be kept separate from the training and validation datasets to ensure an objective evaluation.

WORKING:

Creating training, validation, and test datasets from the iris.arff dataset in WEKA involves splitting the dataset into three separate subsets. Here's how you can do it:

- Training Set:

1. Open WEKA: Launch the WEKA application.
2. Load Dataset: Go to the "Explorer" tab, click on the "Open file" button, and select the iris.arff dataset.



Filter: Resample -S 1 -Z 60.0 -no-replacement

Current relation
Relation: iris Attributes: 5
Instances: 150 Sum of weights: 150

Attributes

No.	Name
1	□ sepallength
2	□ sepalwidth
3	□ petallength
4	□ petalwidth
5	□ class

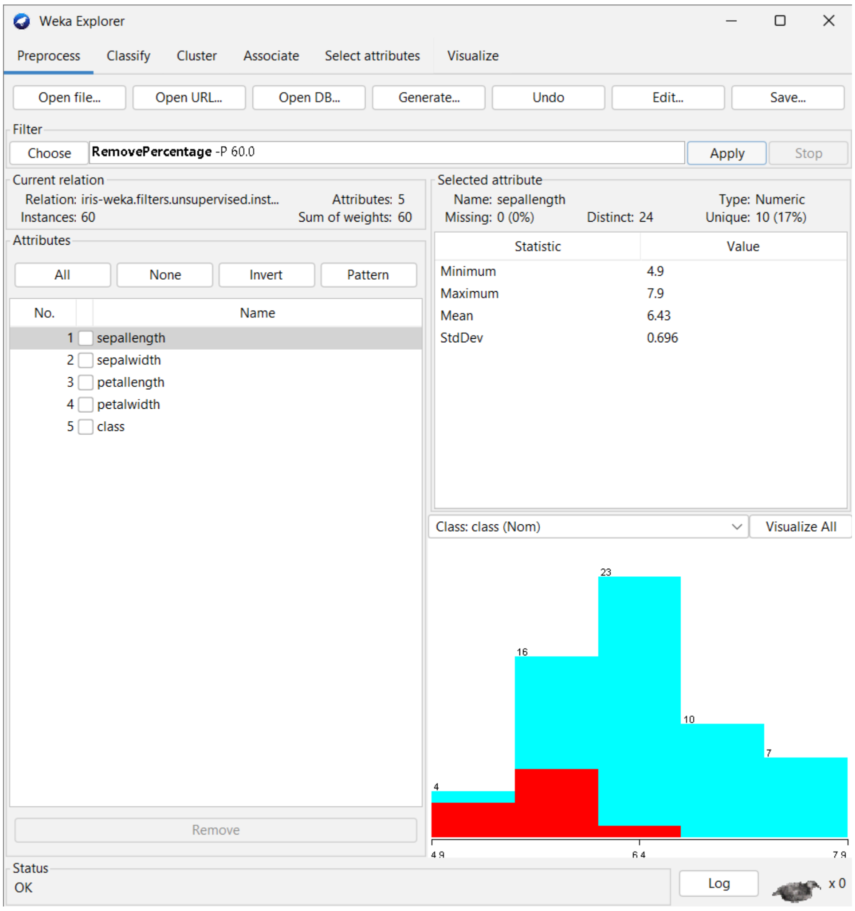
Selected attribute
Name: sepallength Type: Numeric
Missing: 0 (0%) Distinct: 35 Unique: 9 (6%)

Statistic	Value
Minimum	4.3
Maximum	7.9
Mean	5.843
StdDev	0.828

Class: class (Nom)

Status
OK

3. Select the **RemovePercentage** filter in the preprocess panel.
4. Set the correct percentage for the split.
5. Apply the filter.
6. Save the generated data as a new file.



Filter: **RemovePercentage** -P 60.0

Current relation
Relation: iris-weka.filters.unsupervised.inst... Attributes: 5
Instances: 60 Sum of weights: 60

Selected attribute
Name: sepallength Type: Numeric
Missing: 0 (0%) Distinct: 24 Unique: 10 (17%)

Statistic	Value
Minimum	4.9
Maximum	7.9
Mean	6.43
StdDev	0.696

Attributes

No.		Name
1	□	sepallength
2	□	sepalwidth
3	□	petallength
4	□	petalwidth
5	□	class

Class: class (Nom)

Status
OK

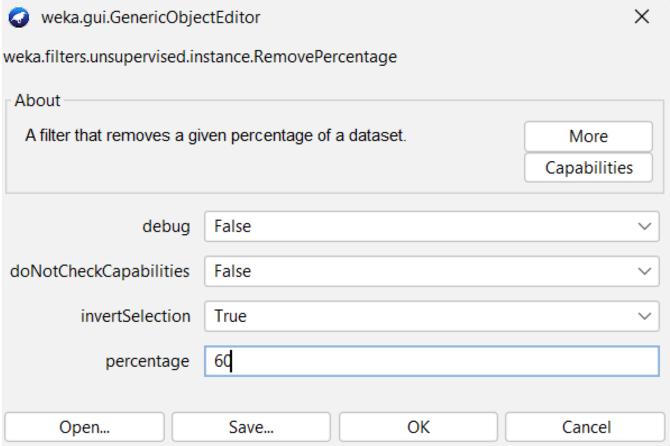
- Training Set:

1. Open WEKA: Launch the WEKA application.

2. Load Dataset: Go to the "Explorer" tab, click on the "Open file" button, and select the iris.arff dataset.

3. select the RemovePercentage filter if not yet selected.

4. set the invertSelection property to true.



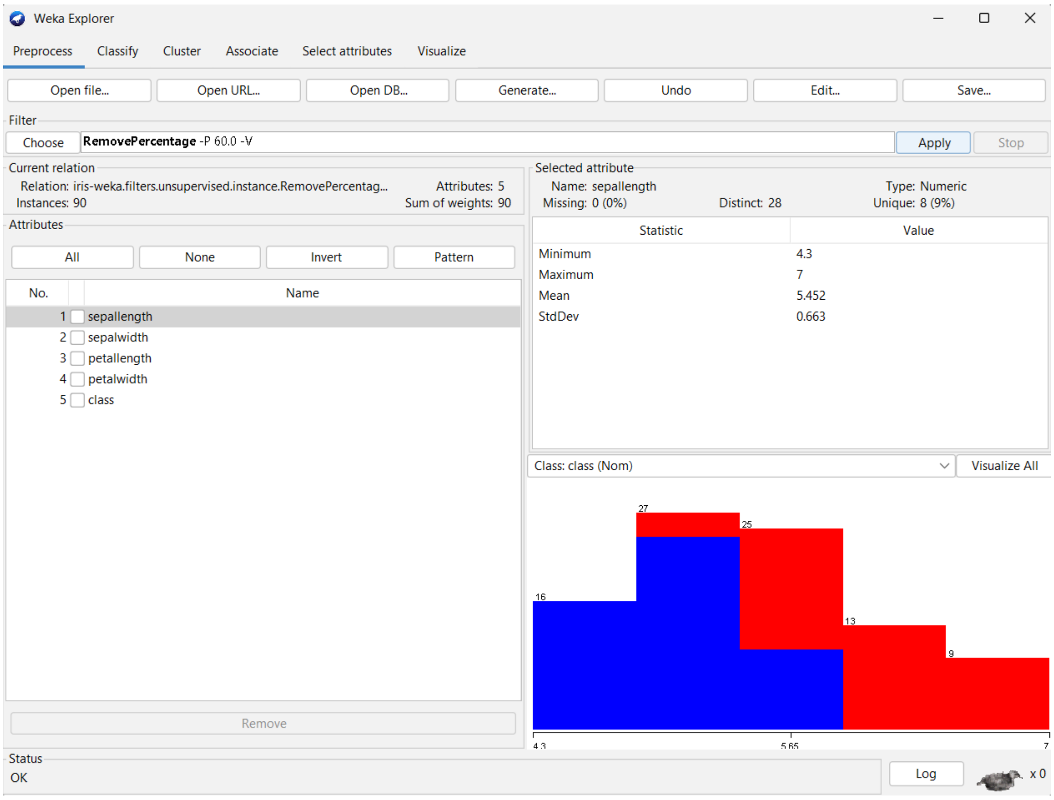
weka.filters.unsupervised.instance.RemovePercentage

About
A filter that removes a given percentage of a dataset.

debug: False
doNotCheckCapabilities: False
invertSelection: True
percentage: 60

5. apply the filter.

6. save the generated data as new file.



Filter: RemovePercentage -P 60.0 -V

Current relation

Relation: iris-weka.filters.unsupervised.instance.RemovePercentag... Attributes: 5

Instances: 90 Sum of weights: 90

No.	Name
1	□ sepallength
2	□ sepalwidth
3	□ petallength
4	□ petalwidth
5	□ class

Selected attribute

Name: sepallength Type: Numeric

Missing: 0 (0%) Distinct: 28 Unique: 8 (9%)

Statistic	Value
Minimum	4.3
Maximum	7
Mean	5.452
StdDev	0.663

Class: class (Nom)

Status

OK

PROGRAM 10:

To apply Apriori technique on the dataset and to generate association rules.

THOERY:

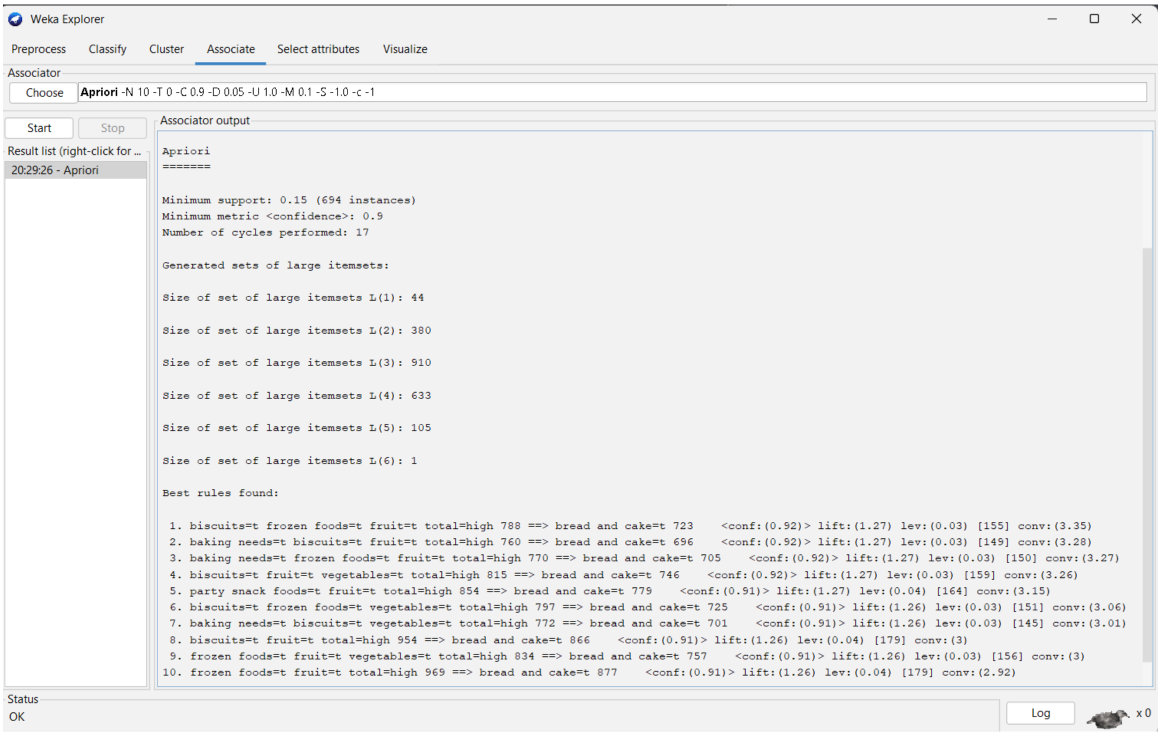
The Apriori algorithm is a common method for association rule mining. It works by generating all possible itemsets in a dataset, finding the frequent itemsets using the "join and prune" technique, and generating association rules from the frequent itemsets.

The Apriori algorithm is a powerful tool for association rule mining. It can be used to find interesting and useful patterns in data. However, it can also be computationally expensive, especially for large datasets.

WORKING:

To apply the Apriori algorithm and generate association rules in WEKA, follow these steps:

1. **Load the Dataset**: Open WEKA and load your dataset.
2. **Explorer Tab**: Go to the "Explorer" tab.
3. **Choose Apriori Algorithm**: In the "Choose" section, click on the "Associate" button.
4. **Select Apriori**: From the list of association algorithms, select "Apriori" by clicking on it. This will open the configuration panel for the Apriori algorithm.
5. **Configure Apriori Parameters**:
   - You can adjust various parameters such as minimum support, minimum confidence, and other settings based on your dataset and requirements.
   - Set the minimum support and minimum confidence thresholds according to your desired levels of significance.
6. **Run Apriori**: Click on the "Start" button to run the Apriori algorithm on your dataset.
7. **View Association Rules**: Once the algorithm finishes running, you'll see the generated association rules in the "Association rules" tab below the configuration panel.



Associator
Choose Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

Result list (right-click for ...
20:29:26 - Apriori

Associator output

```
Apriori
=======

Minimum support: 0.15 (694 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 17

Generated sets of large itemsets:

Size of set of large itemsets L(1): 44

Size of set of large itemsets L(2): 380

Size of set of large itemsets L(3): 910

Size of set of large itemsets L(4): 633

Size of set of large itemsets L(5): 105

Size of set of large itemsets L(6): 1

Best rules found:

 1. biscuits=t frozen foods=t fruit=t total=high 788 ==> bread and cake=t 723    <conf:(0.92)> lift:(1.27) lev:(0.03) [155] conv:(3.35)
 2. baking needs=t biscuits=t fruit=t total=high 760 ==> bread and cake=t 696    <conf:(0.92)> lift:(1.27) lev:(0.03) [149] conv:(3.28)
 3. baking needs=t frozen foods=t fruit=t total=high 770 ==> bread and cake=t 705    <conf:(0.92)> lift:(1.27) lev:(0.03) [150] conv:(3.27)
 4. biscuits=t fruit=t vegetables=t total=high 815 ==> bread and cake=t 746    <conf:(0.92)> lift:(1.27) lev:(0.03) [159] conv:(3.26)
 5. party snack foods=t fruit=t total=high 854 ==> bread and cake=t 779    <conf:(0.91)> lift:(1.27) lev:(0.04) [164] conv:(3.15)
 6. biscuits=t frozen foods=t vegetables=t total=high 797 ==> bread and cake=t 725    <conf:(0.91)> lift:(1.26) lev:(0.03) [151] conv:(3.06)
 7. baking needs=t biscuits=t vegetables=t total=high 772 ==> bread and cake=t 701    <conf:(0.91)> lift:(1.26) lev:(0.03) [145] conv:(3.01)
 8. biscuits=t fruit=t total=high 954 ==> bread and cake=t 866    <conf:(0.91)> lift:(1.26) lev:(0.04) [179] conv:(3)
 9. frozen foods=t fruit=t vegetables=t total=high 834 ==> bread and cake=t 757    <conf:(0.91)> lift:(1.26) lev:(0.03) [156] conv:(3)
10. frozen foods=t fruit=t total=high 969 ==> bread and cake=t 877    <conf:(0.91)> lift:(1.26) lev:(0.04) [179] conv:(2.92)
```

Status
OK

PROGRAM 11:

(a) Demonstration of classification rule process on dataset student.arff using J48 algorithm.
(b) Demonstration of classification rule process on dataset employee.arff using J48 algorithm.

THOERY:

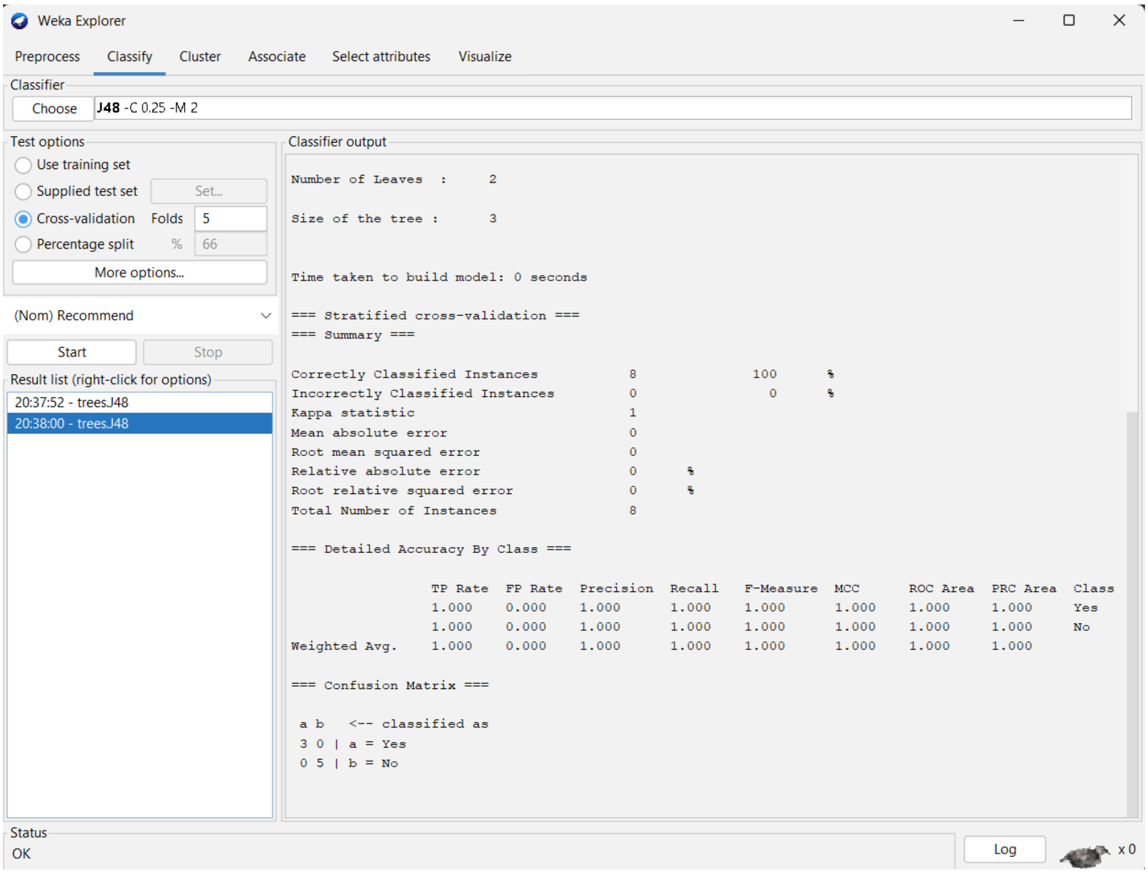
The Apriori algorithm is a common method for association rule mining. It works by generating all possible itemsets in a dataset, finding the frequent itemsets using the "join and prune" technique, and generating association rules from the frequent itemsets.

The Apriori algorithm is a powerful tool for association rule mining. It can be used to find interesting and useful patterns in data. However, it can also be computationally expensive, especially for large datasets.

WORKING:

(a) To demonstrate the classification rule process using the J48 algorithm on the dataset **student.arff** in WEKA, follow these steps:

1. **Load the Dataset**: Open WEKA and load the **student.arff** dataset.
2. **Explorer Tab**: Go to the "Explorer" tab.
3. **Choose J48 Algorithm**: In the "Choose" section, click on the "Classify" button.
4. **Select J48**: From the list of classification algorithms, select "J48" by clicking on it. This will open the configuration panel for the J48 algorithm.
5. **Configure J48 Parameters**:
   - You can adjust various parameters such as confidence factor, minimum number of instances per leaf, and other settings based on your dataset and requirements.
6. **Run J48**: Click on the "Start" button to run the J48 algorithm on your dataset.



J48 -C 0.25 -M 2

○ Use training set
○ Supplied test set Set...
● Cross-validation Folds 5
○ Percentage split % 66

(Nom) Recommend

20:37:52 - trees.J48
20:38:00 - trees.J48

Status: OK

```
Number of Leaves  :     2

Size of the tree :      3


Time taken to build model: 0 seconds

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

Correctly Classified Instances           8              100      %
Incorrectly Classified Instances         0                0      %
Kappa statistic                          1
Mean absolute error                      0
Root mean squared error                  0
Relative absolute error                  0      %
Root relative squared error              0      %
Total Number of Instances                8

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Yes
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     No
Weighted Avg.    1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000

=== Confusion Matrix ===

 a b   <-- classified as
 3 0 | a = Yes
 0 5 | b = No
```

(b) To demonstrate the classification rule process using the J48 algorithm on the dataset **employee.arff** in WEKA, you can follow a similar process as described earlier. Here's a step-by-step guide:

1. **Load the Dataset**: Open WEKA and load the **employee.arff** dataset.

2. **Explorer Tab**: Go to the "Explorer" tab.

3. **Choose J48 Algorithm**: In the "Choose" section, click on the "Classify" button.

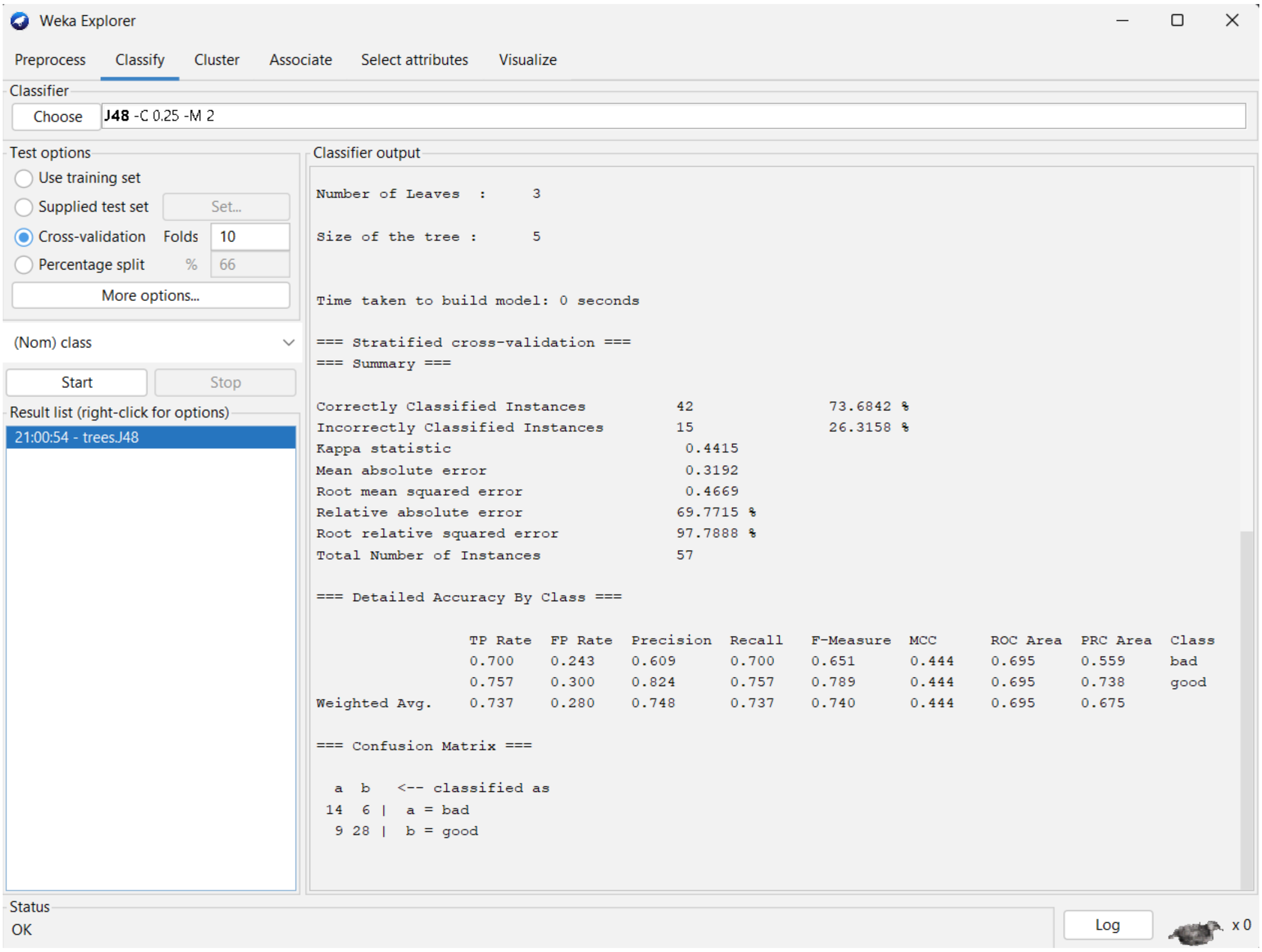
4. **Select J48**: From the list of classification algorithms, select "J48" by clicking on it. This will open the configuration panel for the J48 algorithm.

5. **Configure J48 Parameters**:

   - Adjust various parameters such as confidence factor, minimum number of instances per leaf, and other settings based on your dataset and requirements.

   - Set the parameters according to your desired model complexity and performance requirements.

6. **Run J48**: Click on the "Start" button to run the J48 algorithm on your dataset.



J48 -C 0.25 -M 2

Test options: ○ Use training set; ○ Supplied test set (Set...); ● Cross-validation Folds 10; ○ Percentage split % 66

(Nom) class

21:00:54 - trees.J48

Status: OK

```
Number of Leaves  :     3

Size of the tree :      5


Time taken to build model: 0 seconds

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

Correctly Classified Instances          42               73.6842 %
Incorrectly Classified Instances        15               26.3158 %
Kappa statistic                          0.4415
Mean absolute error                      0.3192
Root mean squared error                  0.4669
Relative absolute error                 69.7715 %
Root relative squared error             97.7888 %
Total Number of Instances               57

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.700    0.243    0.609      0.700    0.651      0.444    0.695     0.559     bad
                 0.757    0.300    0.824      0.757    0.789      0.444    0.695     0.738     good
Weighted Avg.    0.737    0.280    0.748      0.737    0.740      0.444    0.695     0.675

=== Confusion Matrix ===

  a  b   <-- classified as
 14  6 |  a = bad
  9 28 |  b = good
```