

NAME : DIYA JAIN

DIV : TY-3

ROLL NO : 26

BATCH : B

SUBJECT : DWM

Experiment 08

Aim: Perform data Pre-processing task and demonstrate Classification, Clustering, Association algorithm on data sets using data mining tool WEKA.

Introduction: Data mining is the process of extracting useful patterns from large datasets. WEKA is a powerful open-source tool that supports various data mining techniques through an easy-to-use interface. In this experiment, we use WEKA to demonstrate three key tasks:

- **Classification:** Predicting predefined class labels (e.g., spam detection).
- **Clustering:** Grouping similar data without prior labels.
- **Association:** Finding relationships between items (e.g., market basket analysis).

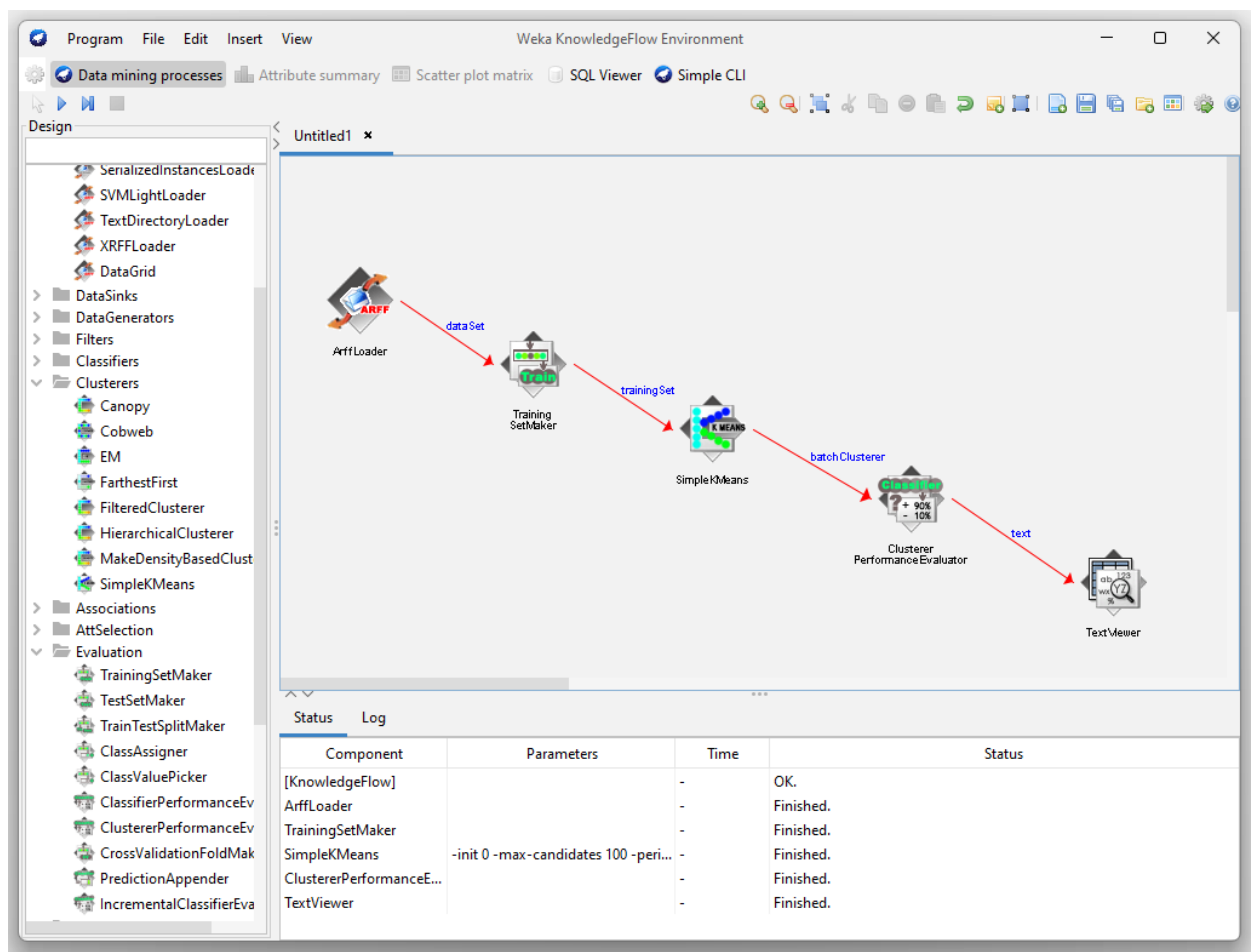
Before applying these algorithms, data preprocessing is done to clean and prepare the data for better accuracy.

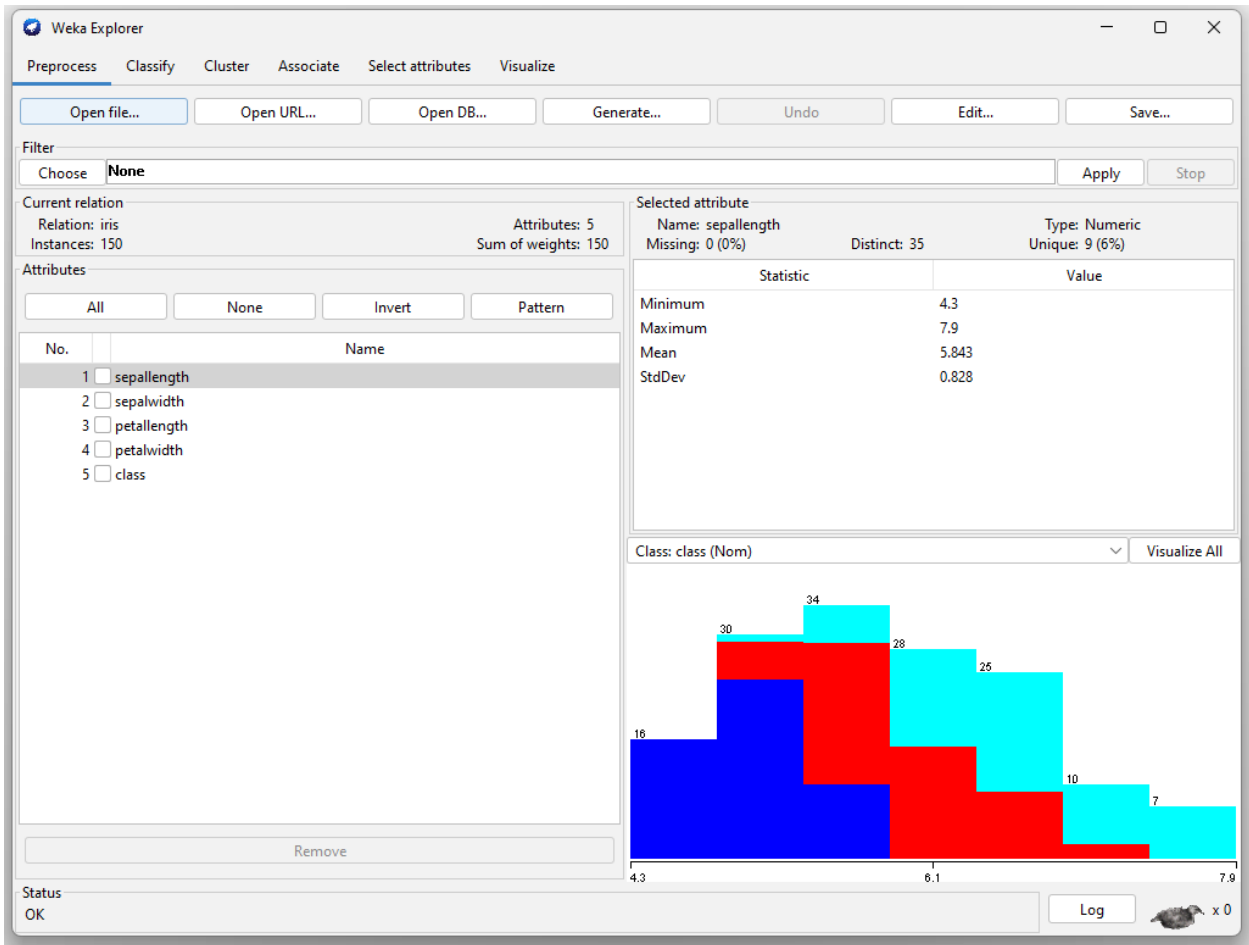
Procedure:

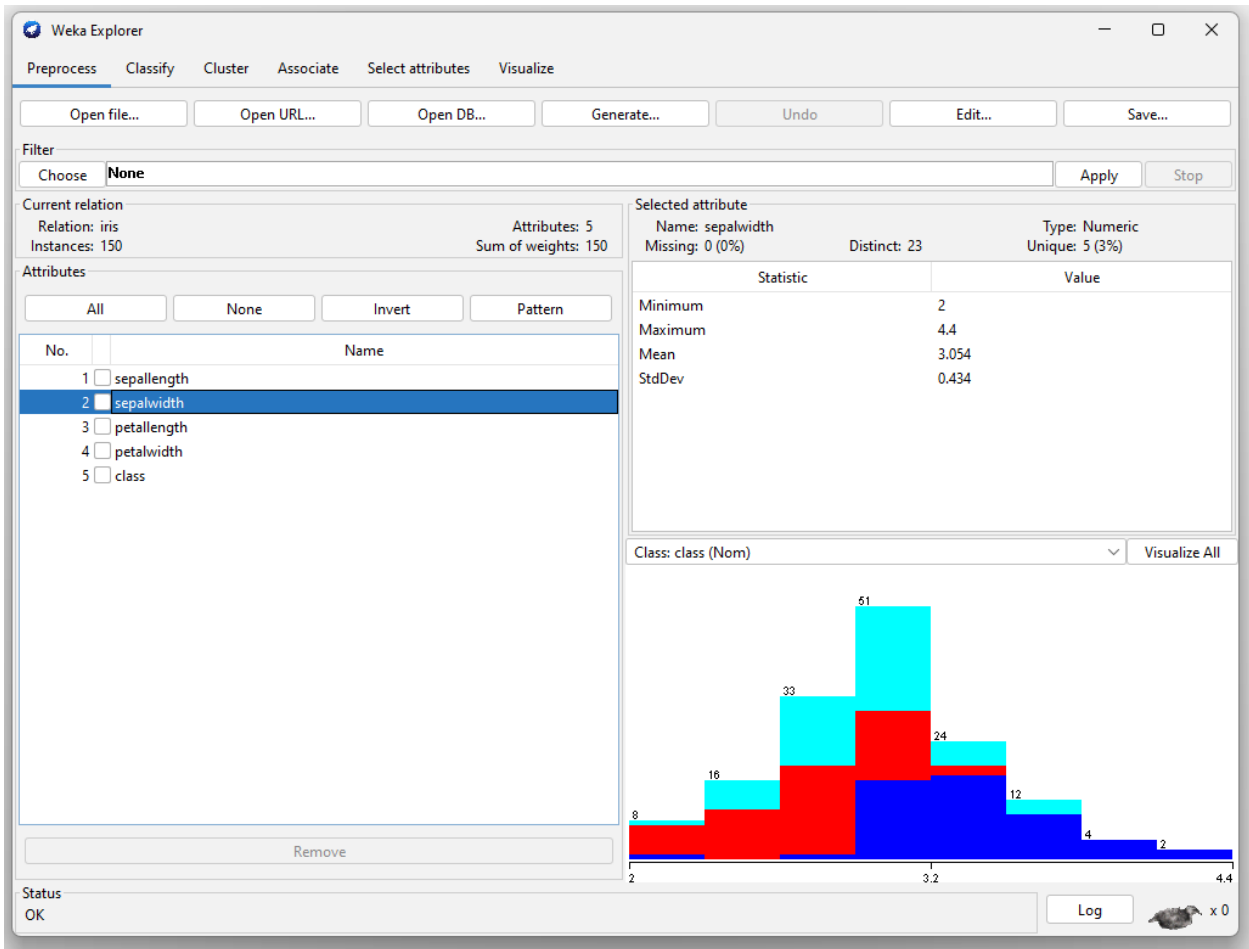
1. **Open Weka Knowledge Flow:**
 - Go to **Program Files** on your **PC** and launch **Weka 3.6**.
 - Choose the **Knowledge Flow** environment from the initial menu (Explorer, Experimenter, Knowledge Flow, etc.).
1. **Load Dataset Using Arff Loader:**
 - Drag the **ArffLoader** from the "Data Sources" section into the canvas.
 - Right-click → **Configure**, then click **Browse** and select a dataset (e.g., from the **Data** folder like `iris.arff`).
 - This loads your data into the flow.
1. **Configure Evaluation Component:**
 - Add the **Evaluation** component to evaluate the clustering model.
 - Set the evaluation type to **Static** for using the dataset as-is.
1. **Prepare the Training Format:**
 - Add a **TrainingSetMaker** component.
 - This prepares your data in a format suitable for training.
 - Connect it to the output of the ArffLoader.
1. **Add and Configure Clusterer:**
 - Drag the **Clusterer** component into the workspace.
 - Choose **SimpleKMeans** as the clustering algorithm.
 - Configure it (e.g., set number of clusters, distance function, etc.).
1. **Analyze Clustering Performance:**

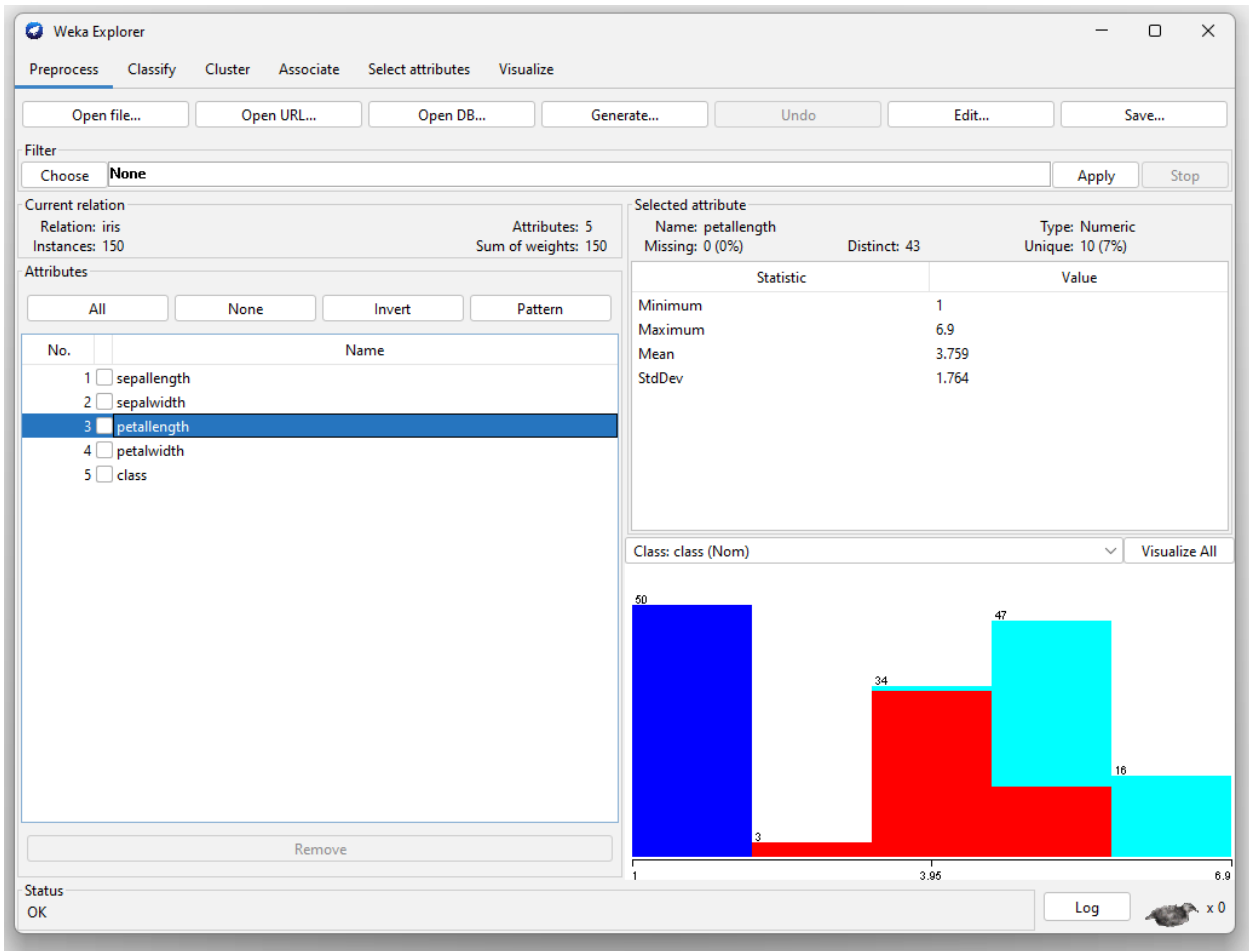
- Add the **ClustererPerformanceEvaluator** component.
 - Connect it to the output of the Clusterer to measure model effectiveness.
1. **Add Output Viewers:**
 - Drag in a **TextViewer** to view textual output (e.g., cluster assignments, summary).
 - Add a **Visualization** component for graphical display of cluster distribution.
 1. **Connect Components and Run Flow:**
 - Right-click on each component to **Connect** them in order: ArffLoader → TrainingSetMaker → Clusterer → ClustererPerformanceEvaluator → TextViewer/Visualization
 - Finally, right-click the **last component** and choose **Start Execution** to run the workflow.

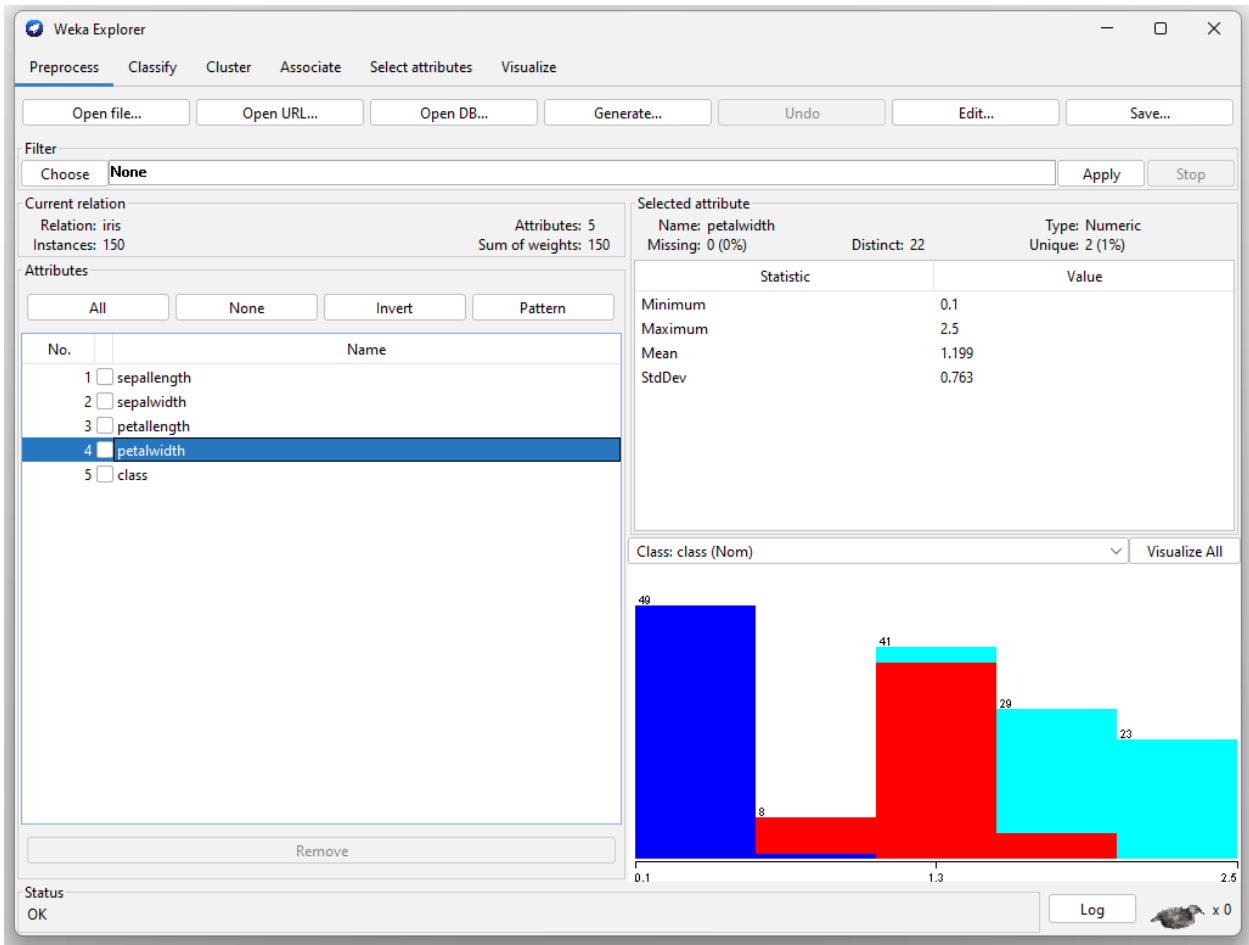
Implementation/Outputs:

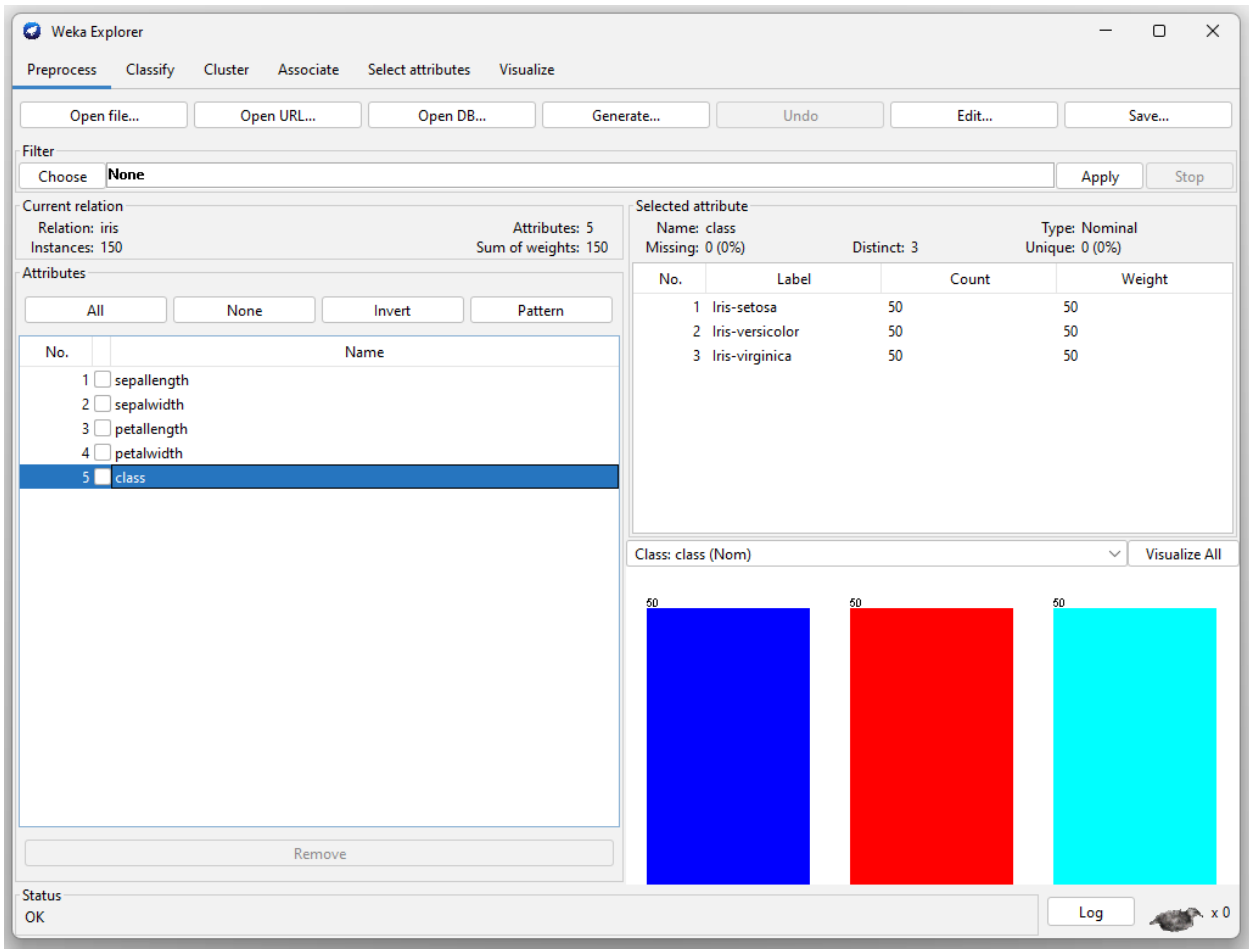












Weka Explorer

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Clusterer

ChooseEM -I 100 -N -1 -X 10 -max -1 -ll-cv 1.0E-6 -ll-iter 1.0E-6 -M 1.0E-6 -K 10 -num-slots 1 -S 100

Cluster mode

☒ Use training set

☐ Supplied test set

☐ Percentage split

☐ Classes to clusters evaluation

☒ Store clusters for visualization

Set...

%66

(Nom) class

Ignore attributes

Start

Stop

Result list (right-click for options)

11:28:21 - EM

Clusterer output

=== Run information ===

Scheme: weka.clusterers.EM -I 100 -N -1 -X 10 -max -1 -ll-cv 1.0E-6 -ll-iter 1.0E-6

Relation: iris

Instances: 150

Attributes: 5

Test mode: evaluate on training data

=== Clustering model (full training set) ===

EM

==

Number of clusters selected by cross validation: 4

Number of iterations performed: 16

AttributeCluster

0123

(0.32)(0.33)(0.2)(0.14)

=====

sepalwidth

mean5.8975.0066.94266.1304

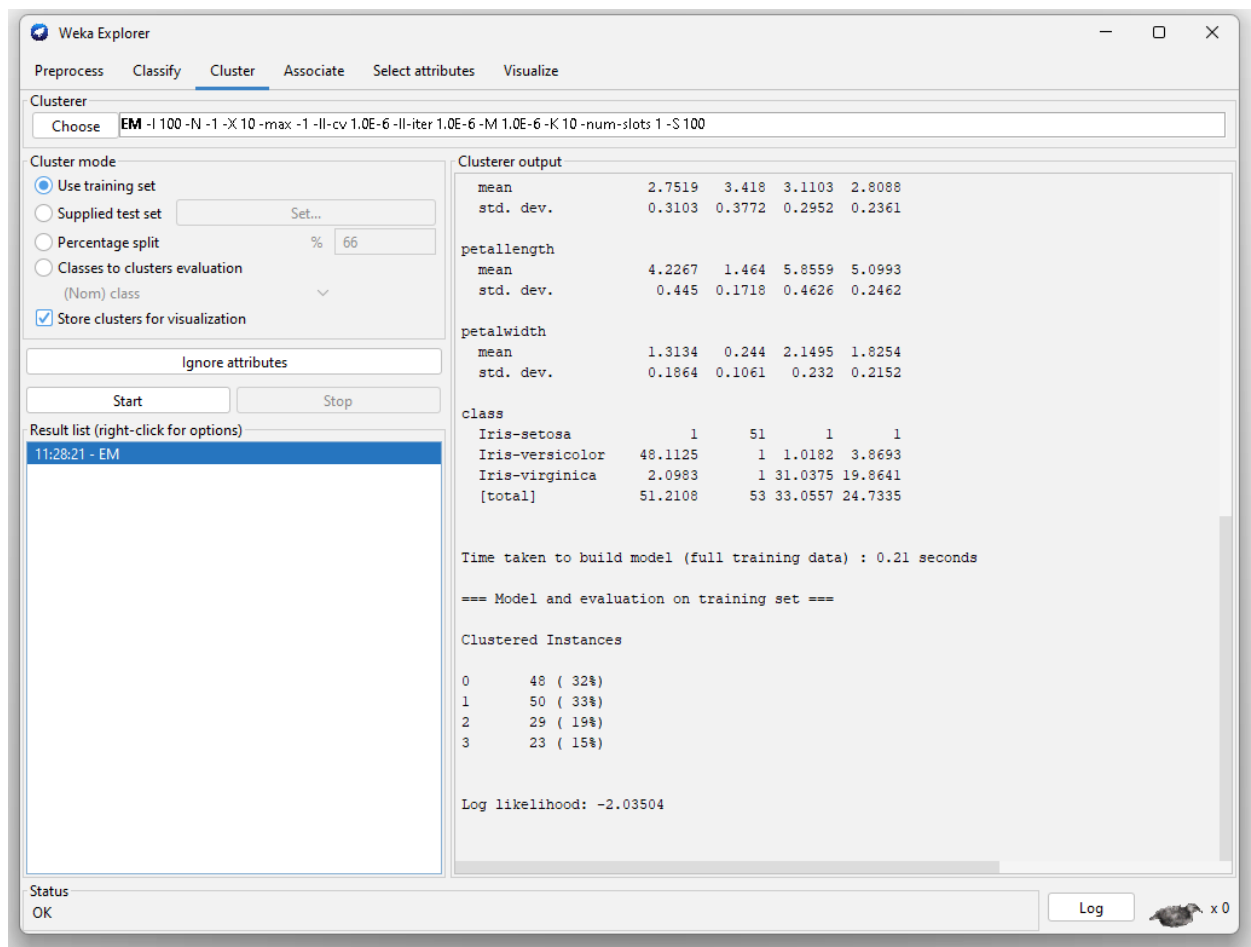
std. dev.0.52790.34890.4980.2943

Status

OK

Log

x 0



Conclusion: We successfully demonstrated data preprocessing and applied key data mining techniques—Classification, Clustering, and Association—using the WEKA tool. WEKA's intuitive interface and built-in algorithms made it easy to load datasets, configure models, and visualize results. Through this practical approach, we understood how to classify data, group it into clusters, and discover hidden associations, all of which are essential in real-world data analysis and decision-making.

GITHUB LINK : <https://github.com/DiyaJain09/DWM-EXPERIMENTS.git>