NAME :- Dev Parekh

ROLL NO :-42

DIV:-TY09/B

> **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.).
> 2. **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.

3. **Configure Evaluation Component:**

   ○ Add the **Evaluation** component to evaluate the clustering model. ○ Set the evaluation type to **Static** for using the dataset as-is.

4. **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.

5. **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.).

6. **Analyze Clustering Performance:**

   ○ Add the **ClustererPerformanceEvaluator** component.

   ○ Connect it to the output of the Clusterer to measure model effectiveness.

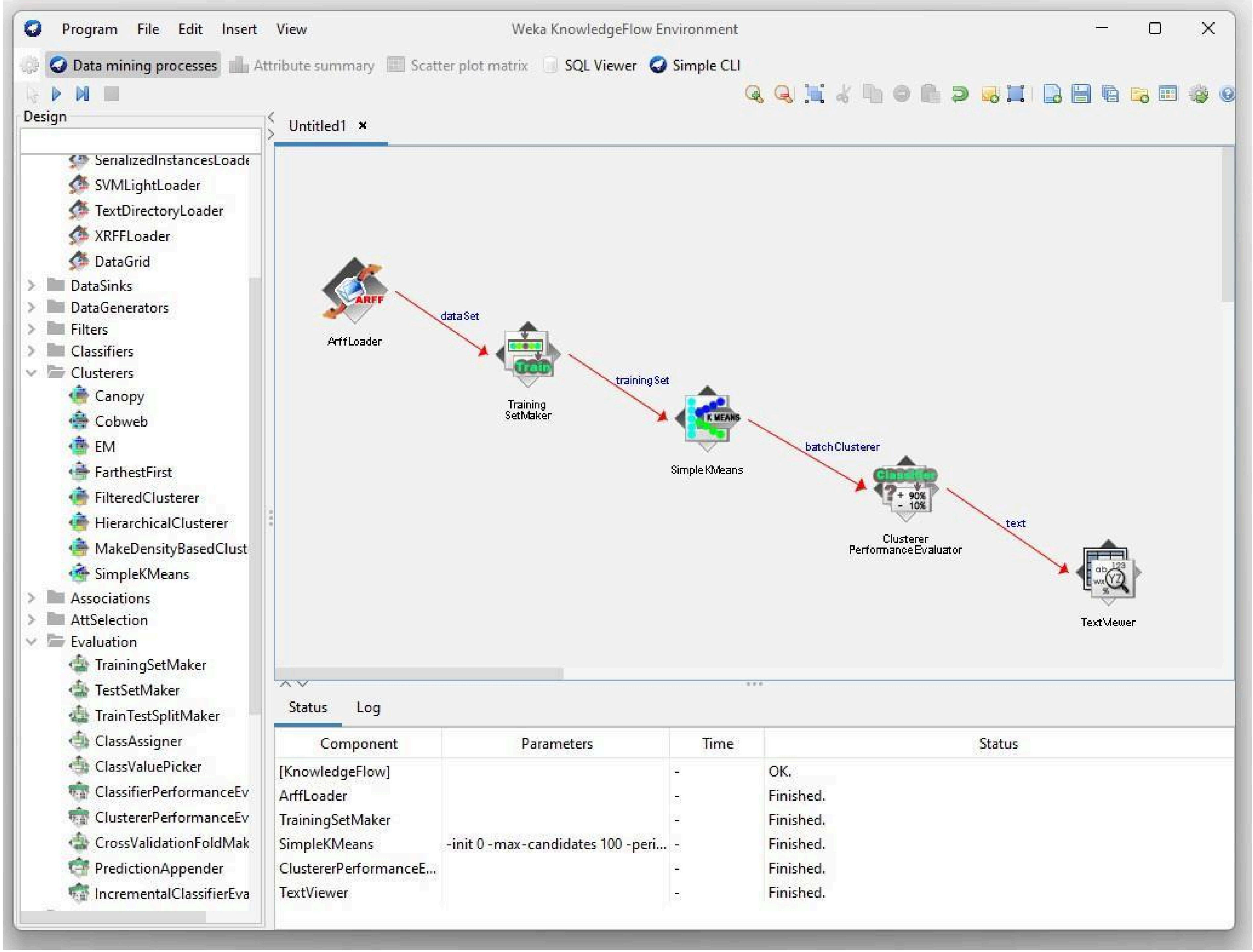
7. **Add Output Viewers:**

   ○ Drag in a **TextViewer** to view textual output (e.g., cluster assignments, summary). ○ Add a **Visualization** component for graphical display of cluster distribution.

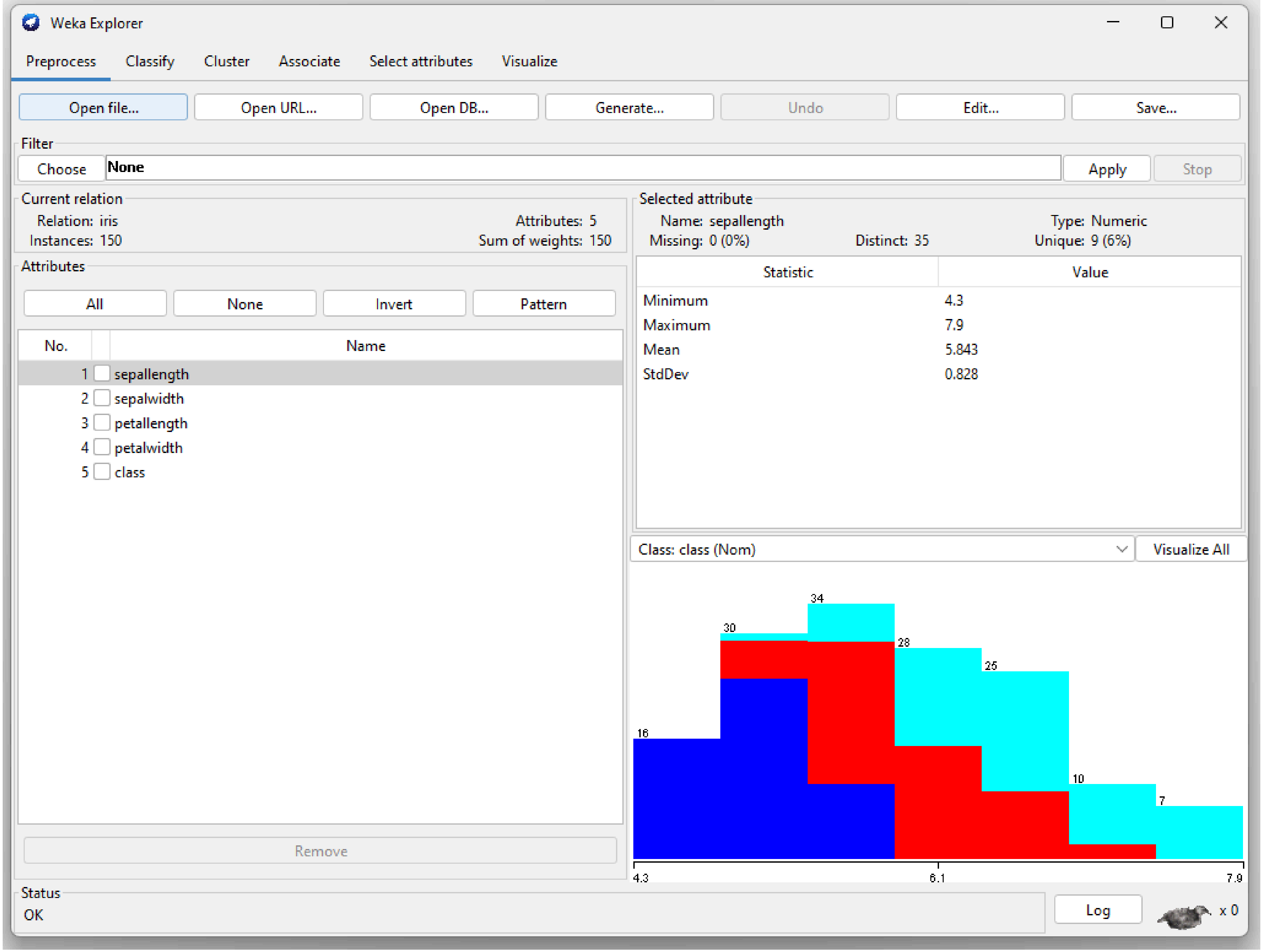
8. **Connect Components and Run Flow:**

   - 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:



Component	Parameters	Time	Status
[KnowledgeFlow]		-	OK.
ArffLoader		-	Finished.
TrainingSetMaker		-	Finished.
SimpleKMeans	-init 0 -max-candidates 100 -peri...	-	Finished.
ClustererPerformanceE...		-	Finished.
TextViewer		-	Finished.



Filter: None

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

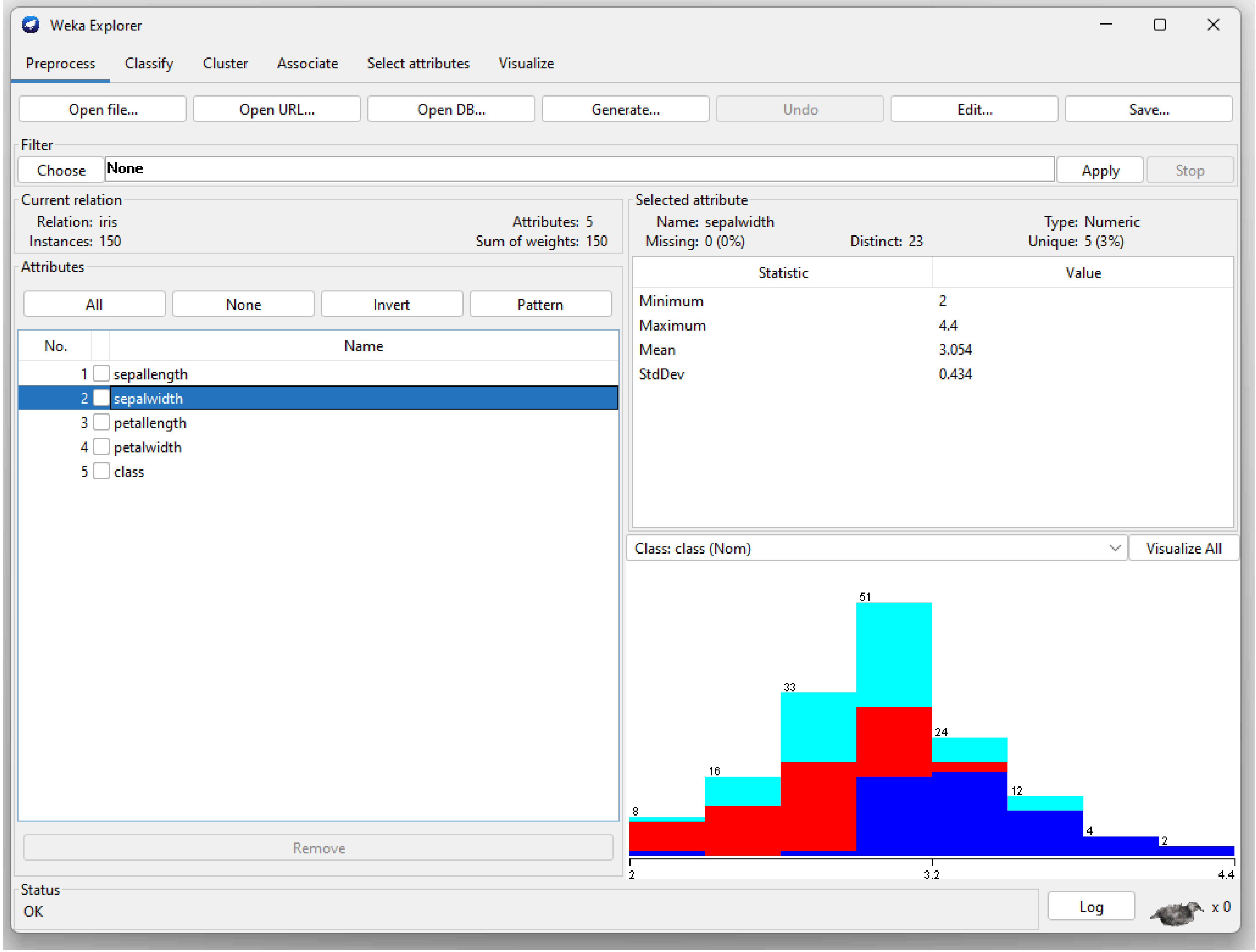
Selected attribute

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



Filter

None

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

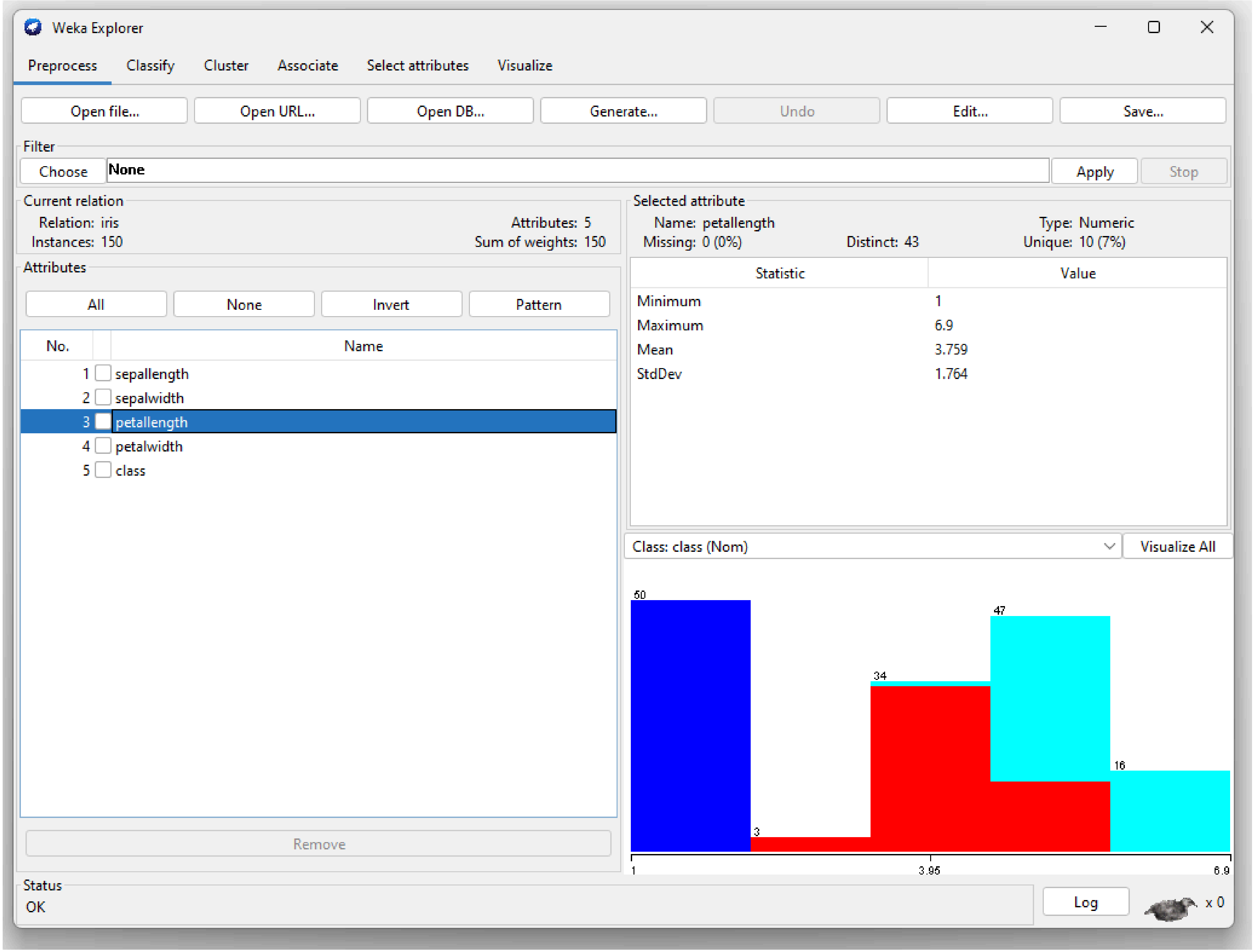
Name: sepalwidth
Type: Numeric
Missing: 0 (0%)
Distinct: 23
Unique: 5 (3%)

Statistic	Value
Minimum	2
Maximum	4.4
Mean	3.054
StdDev	0.434

Class: class (Nom)

Status

OK



Filter
None

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: petallength Type: Numeric

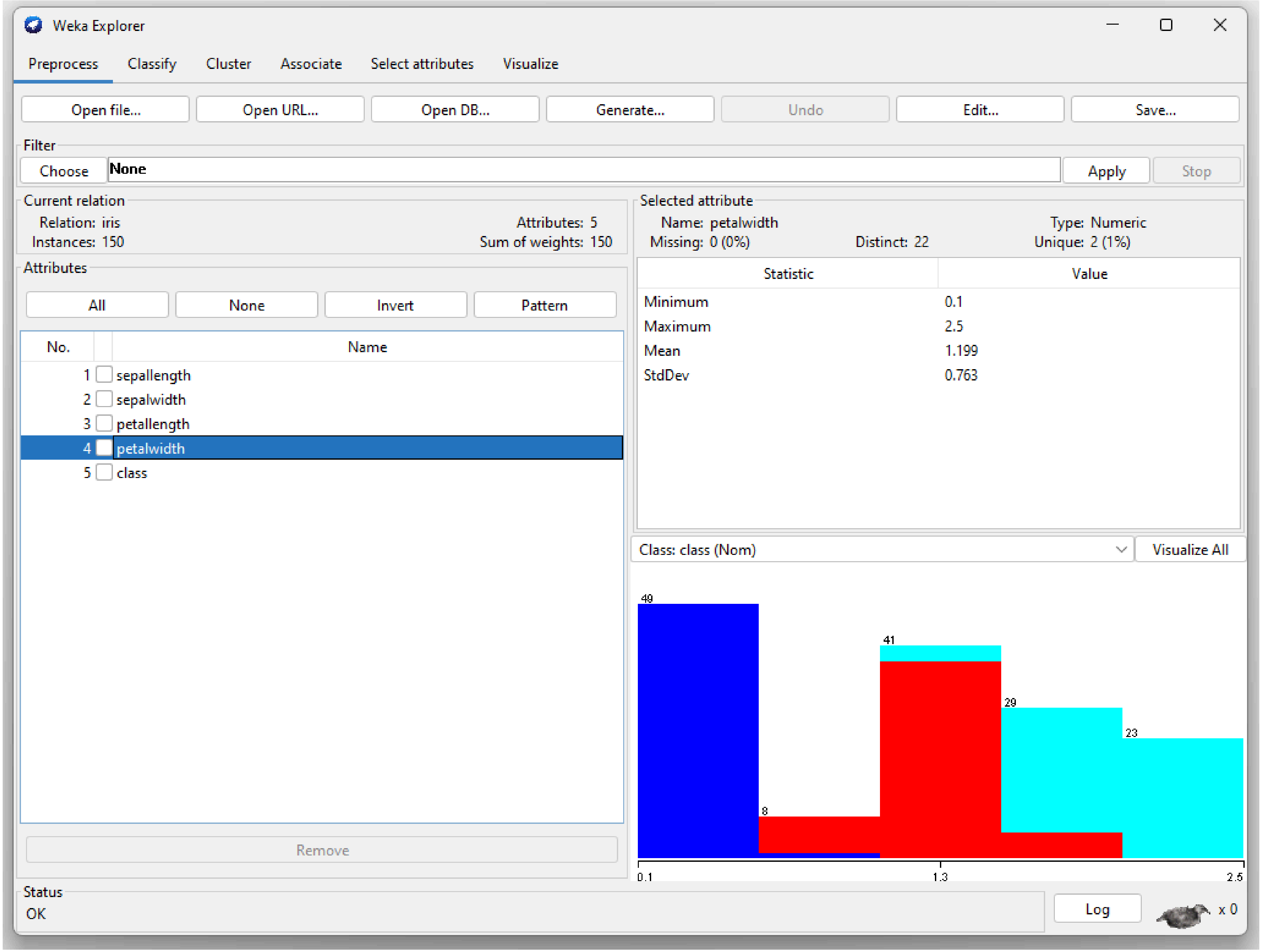
Missing: 0 (0%) Distinct: 43 Unique: 10 (7%)

Statistic	Value
Minimum	1
Maximum	6.9
Mean	3.759
StdDev	1.764

Class: class (Nom)

Status

OK



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

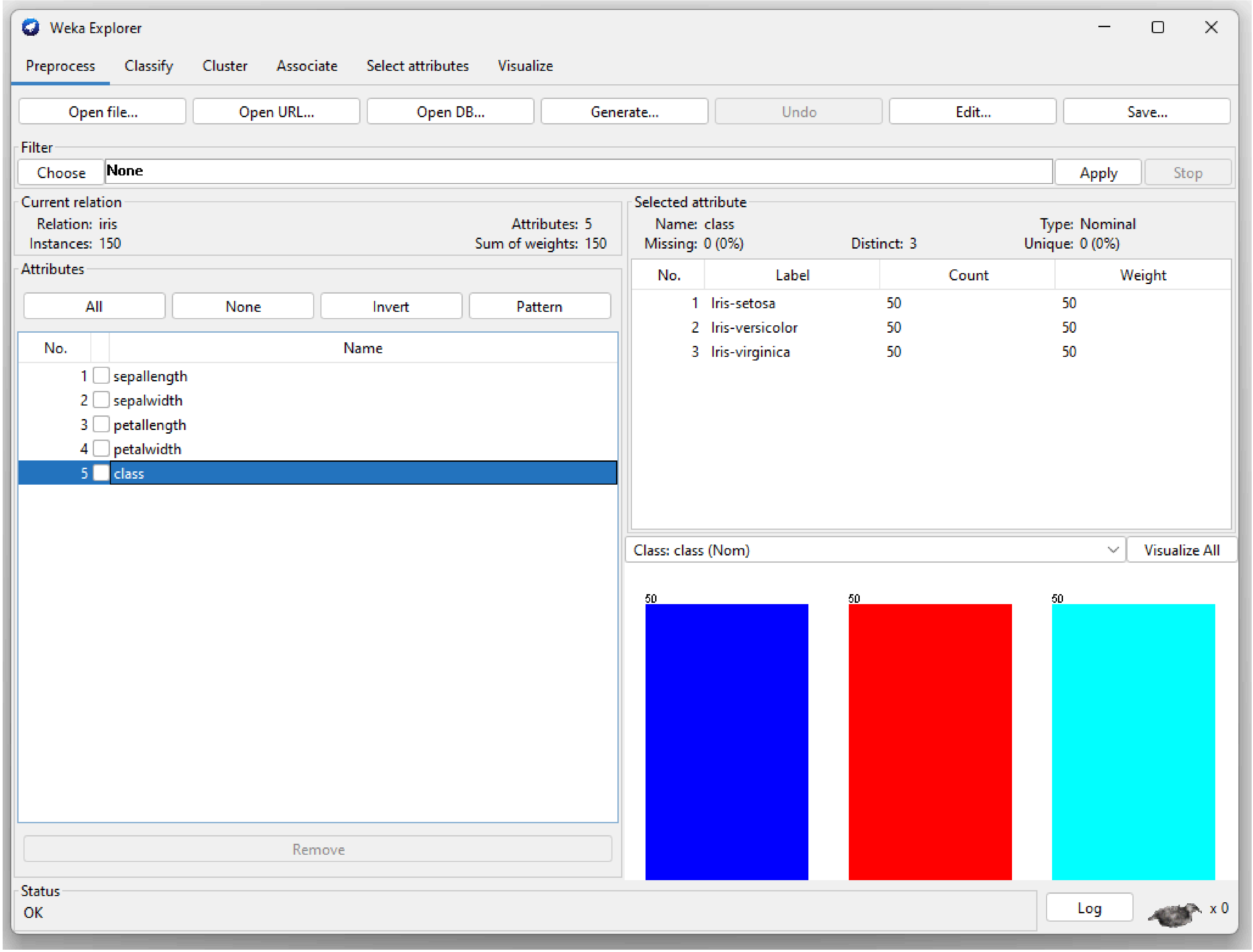
Selected attribute

Name: petalwidth
Type: Numeric
Missing: 0 (0%)
Distinct: 22
Unique: 2 (1%)

Statistic	Value
Minimum	0.1
Maximum	2.5
Mean	1.199
StdDev	0.763

Class: class (Nom)

Status
OK



Filter: None

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

Attributes

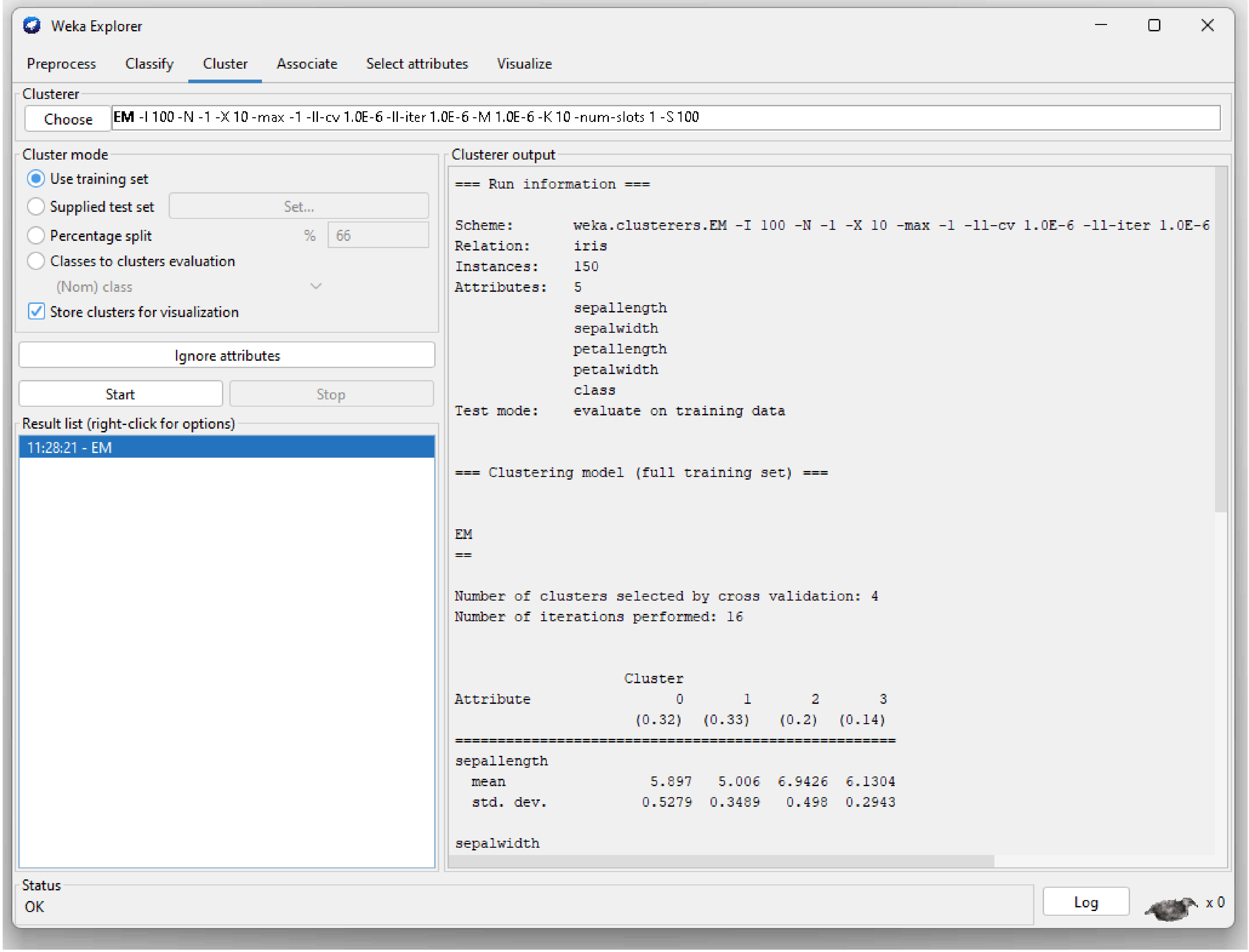
No.		Name
1	□	sepallength
2	□	sepalwidth
3	□	petallength
4	□	petalwidth
5	□	class

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

No.	Label	Count	Weight
1	Iris-setosa	50	50
2	Iris-versicolor	50	50
3	Iris-virginica	50	50

Class: class (Nom)

Status
OK



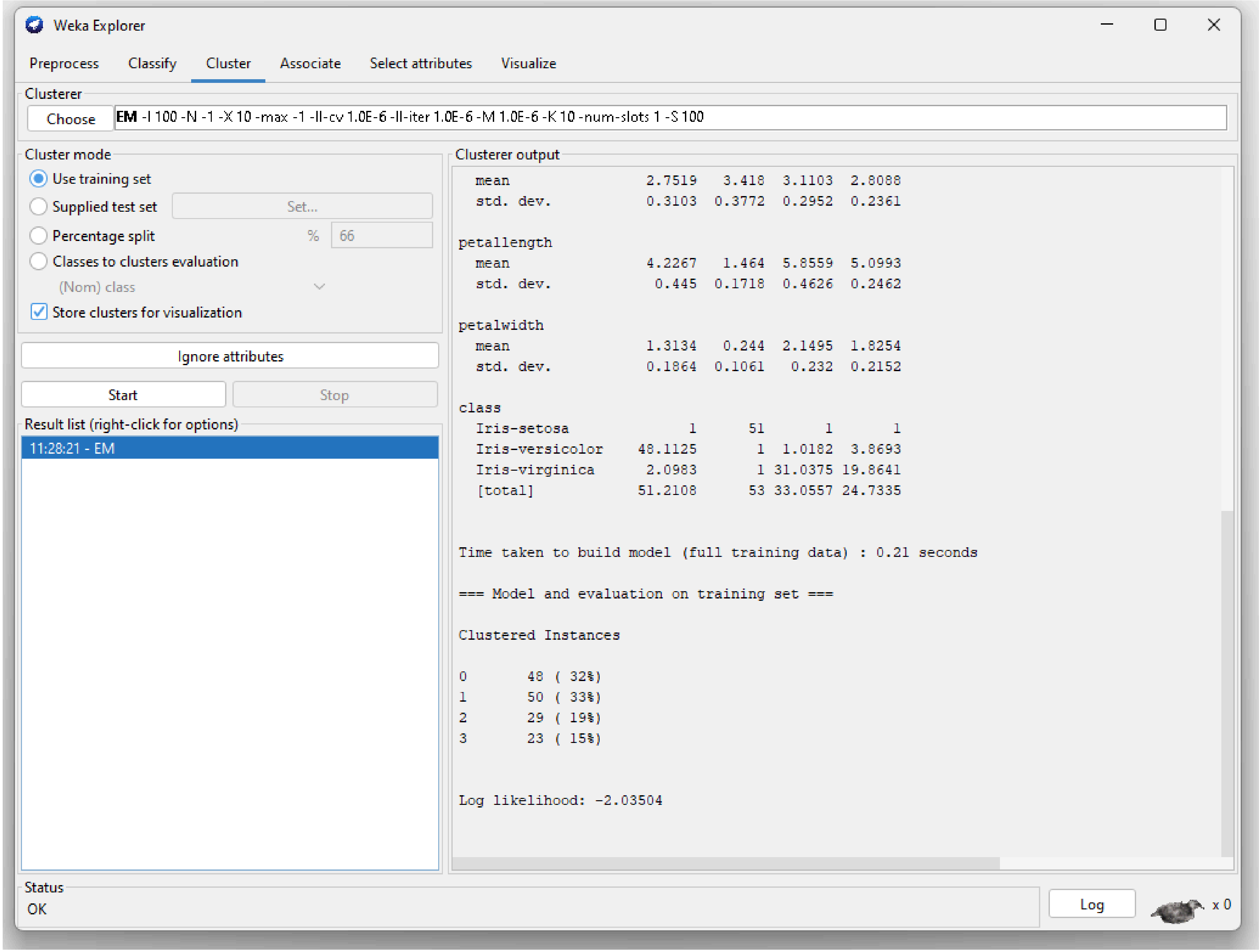
Clusterer: EM -I 100 -N -1 -X 10 -max -1 -ll-cv 1.0E-6 -ll-iter 1.0E-6 -M 1.0E-6 -K 10 -num-slots 1 -S 100

- ● Use training set
- ○ Supplied test set
- ○ Percentage split % 66
- ○ Classes to clusters evaluation
  (Nom) class
- ☑ Store clusters for visualization

Result list: 11:28:21 - EM

```
=== 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
              sepallength
              sepalwidth
              petallength
              petalwidth
              class
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


                    Cluster
Attribute                 0       1       2       3
                     (0.32)  (0.33)   (0.2)  (0.14)
====================================================
sepallength
  mean                5.897   5.006  6.9426  6.1304
  std. dev.          0.5279  0.3489   0.498  0.2943

sepalwidth
```

Status: OK



Clusterer

EM -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 % 66
- ○ Classes to clusters evaluation
  (Nom) class
- ☑ Store clusters for visualization

Result list (right-click for options)

11:28:21 - EM

Clusterer output

```
  mean                2.7519   3.418  3.1103  2.8088
  std. dev.           0.3103  0.3772  0.2952  0.2361

petallength
  mean                4.2267   1.464  5.8559  5.0993
  std. dev.            0.445  0.1718  0.4626  0.2462

petalwidth
  mean                1.3134   0.244  2.1495  1.8254
  std. dev.           0.1864  0.1061   0.232  0.2152

class
  Iris-setosa              1      51       1       1
  Iris-versicolor    48.1125       1  1.0182  3.8693
  Iris-virginica      2.0983       1 31.0375 19.8641
  [total]            51.2108      53 33.0557 24.7335


Time taken to build model (full training data) : 0.21 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       48 ( 32%)
1       50 ( 33%)
2       29 ( 19%)
3       23 ( 15%)


Log likelihood: -2.03504
```

Status

OK

> **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/Devp71/DWM