

LAB PROGRAM - 2

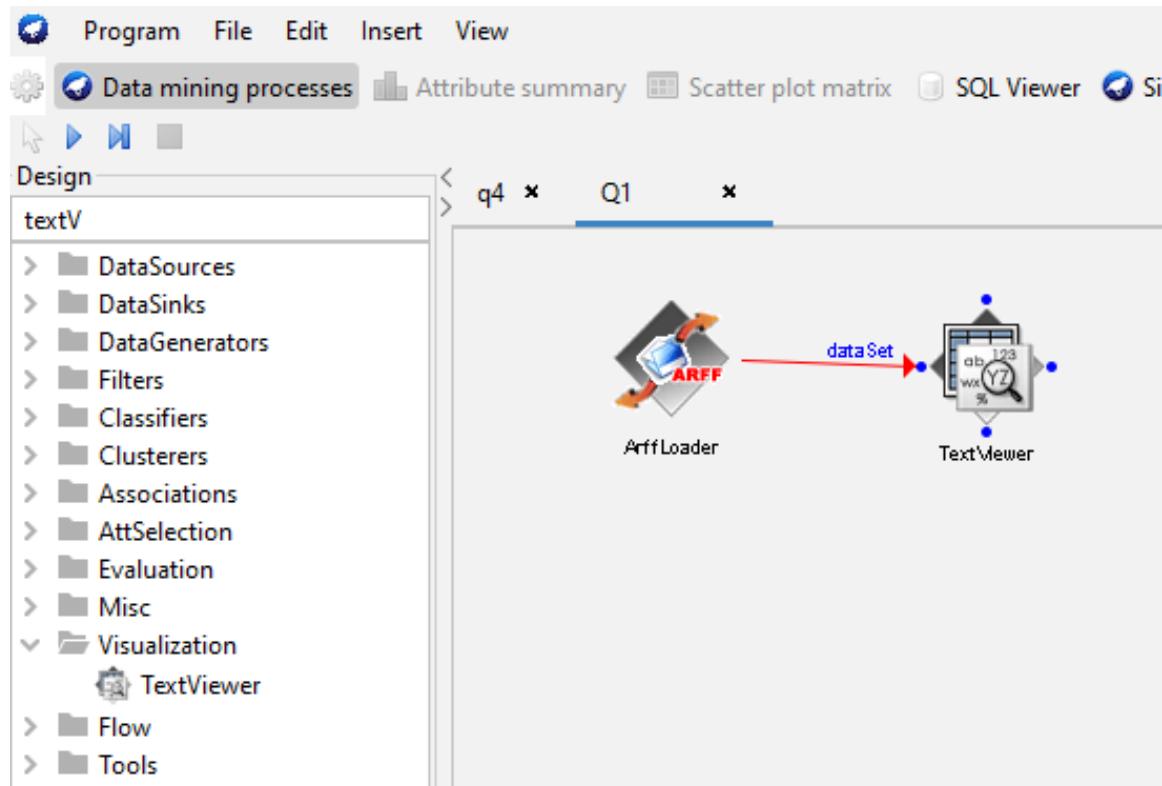
2. Consider the following data set

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

- i) Load ARFF file and explore knowledge flow interface
- ii) configure the data source , check the status area after executing the configuration
- iii) Perform operations such as Attribute Selection, Filter, Classify, Data Sink, Visualization and Evaluation
- iv) Apply incremental learning and analyze the result
- v) do clustering : use generator properties, two clustering schemes, and result panel
- vi) Generate classification Matrix and Construct Decision tree
- vii) Perform Linear Regression and Analyze , Validate and Visualize the data

i) Load ARFF File and Explore Knowledge Flow Interface

- Open Weka 3.9.6 and choose **Knowledge Flow** from the main window.
- In the Knowledge Flow interface, go to the toolbar and click on the **Data Sources** tab.
- Drag the **ARFFLoader** component from the list to the main canvas.
- Double-click the ARFFLoader icon, select the ARFF file you want to load, and click **OK**.
- Connect the **ARFFLoader** to a **DataSource** or **Dataset** component to view data flow.
- Explore the various options and panels (like visualization, preprocessing, etc.) in the interface.



The screenshot shows the 'Text Viewer' window. The left pane displays a 'Result list' with three entries: '08:49:20.811 - labprgm2', '08:50:03.577 - labprgm2lab', and '08:50:08.120 - labprgm2lab'. The right pane is titled 'Text' and contains the actual ARFF data:

```
@relation labprgm2

@attribute no numeric
@attribute 'age nominal' {L20,20-40,G40}
@attribute 'imcome nominal' {high,low,medium}
@attribute 'stud nominal' {no,yes}
@attribute 'creditrate nominal' {fair,excellent}
@attribute 'buyscomp nominal' {yes,no}

@data
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
```

At the bottom of the window, there are buttons for 'Close', 'Settings', and 'Clear results'.

ii) Configure the Data Source and Check the Status Area

- Once the ARFFLoader is connected, right-click it and select **Start Loading**.
- You will see updates in the **Status Area** (usually located at the bottom of the Knowledge Flow window).
- Make sure the data loads successfully by viewing the progress bar or status log.

Component	Parameters	Time	Status
[KnowledgeFlow]	-	-	OK.
ArffLoader	-	-	Finished.
TextViewer	-	-	Finished.

iii) Perform Operations: Attribute Selection, Filter, Classify, Data Sink, Visualization, and Evaluation

1. Attribute Selection:

- Add the **Attribute Selection** component from the list (under "Filter").
- Connect the ARFFLoader or Dataset to the **Attribute Selection** component.
- Double-click to configure and select the desired selection method (e.g., CFS, InfoGain, etc.).
- Start the process and view the selected attributes.

2. Filter:

- Add a **Filter** component (such as **Normalize** or **Discretize**) and connect it to the Dataset.
- Configure the filter and apply it.

3. Classify:

- Add a **Classifier** (e.g., J48 for decision trees, or NaiveBayes).
- Connect the **Filtered Dataset** to the **Classifier**.
- Configure the classifier, run it, and check the results.

4. Data Sink:

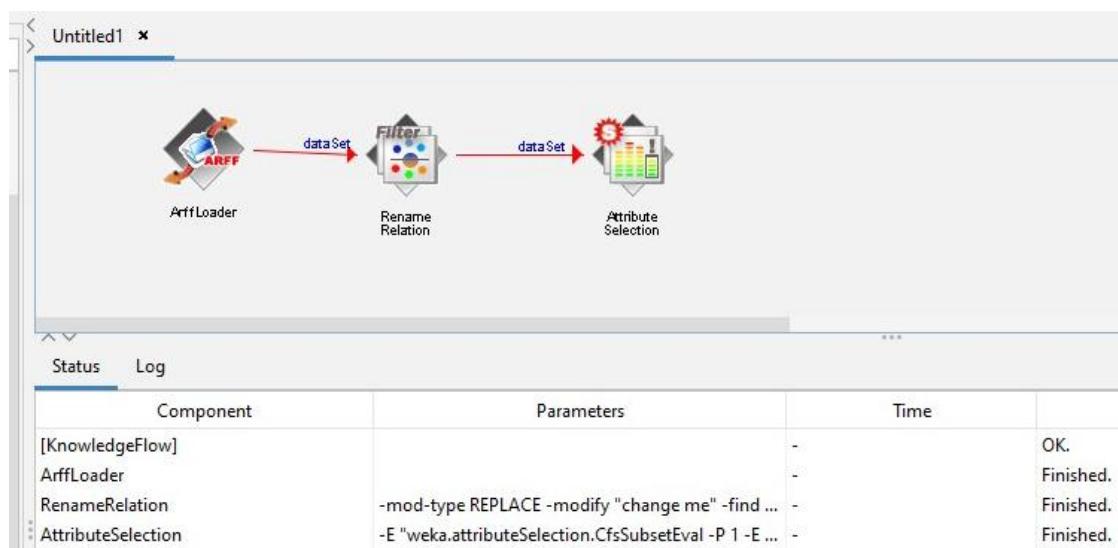
- Use **DataSink** to output results to a file.
- Connect the Classifier's output to **DataSink**, specify the output file, and run the process.

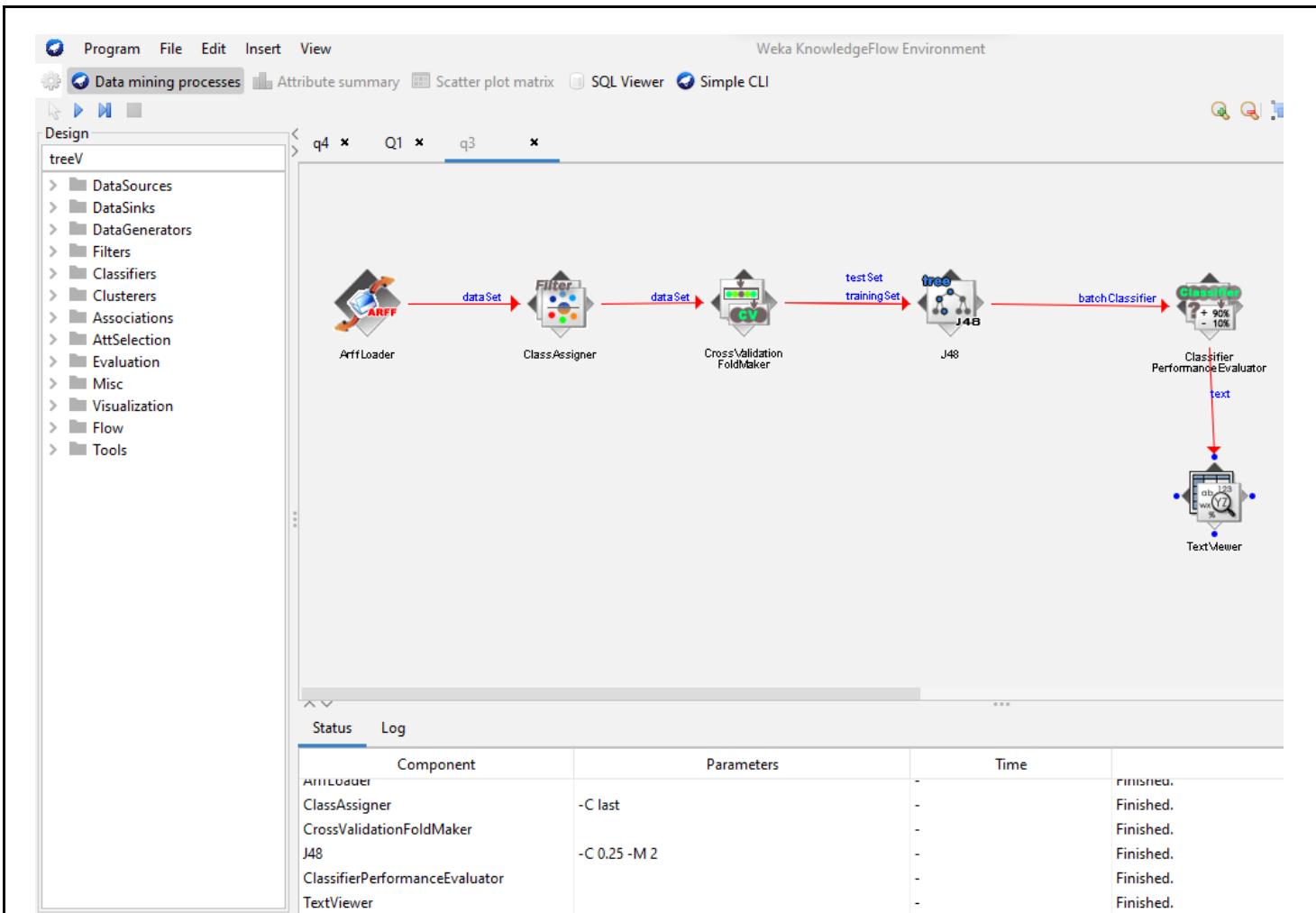
5. Visualization:

- Use **Graph Viewer** to view data plots.
- Connect a **DataSource** or **Classifier** output to it for visualization.

6. Evaluation:

- Add an **Evaluation** component (e.g., **Cross Validation** or **Percentage Split**).
- Connect the classifier to it, configure, and evaluate the model.
- View results such as confusion matrix and performance metric.





Text Viewer

Result list

09:16:43.268 - J48

Text

```

==== Evaluation result ===

Scheme: J48
Options: -C 0.25 -M 2
Relation: labprgm2-weka.filters.unsupervised.attribute.ClassAssigner-Clast

==== Summary ===

Correctly Classified Instances      10          66.6667 %
Incorrectly Classified Instances   5           33.3333 %
Kappa statistic                   -0.1194
Mean absolute error               0.4339
Root mean squared error          0.4938
Relative absolute error          104.4801 %
Root relative squared error     107.8336 %
Total Number of Instances        15

==== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC  ROC Area  PRC Area  Class
0.909    1.000    0.714     0.909    0.800    -0.161  0.182    0.621    yes
0.000    0.091    0.000     0.000    0.000    -0.161  0.182    0.267    no
Weighted Avg.  0.667    0.758    0.524     0.667    0.587    -0.161  0.182    0.526

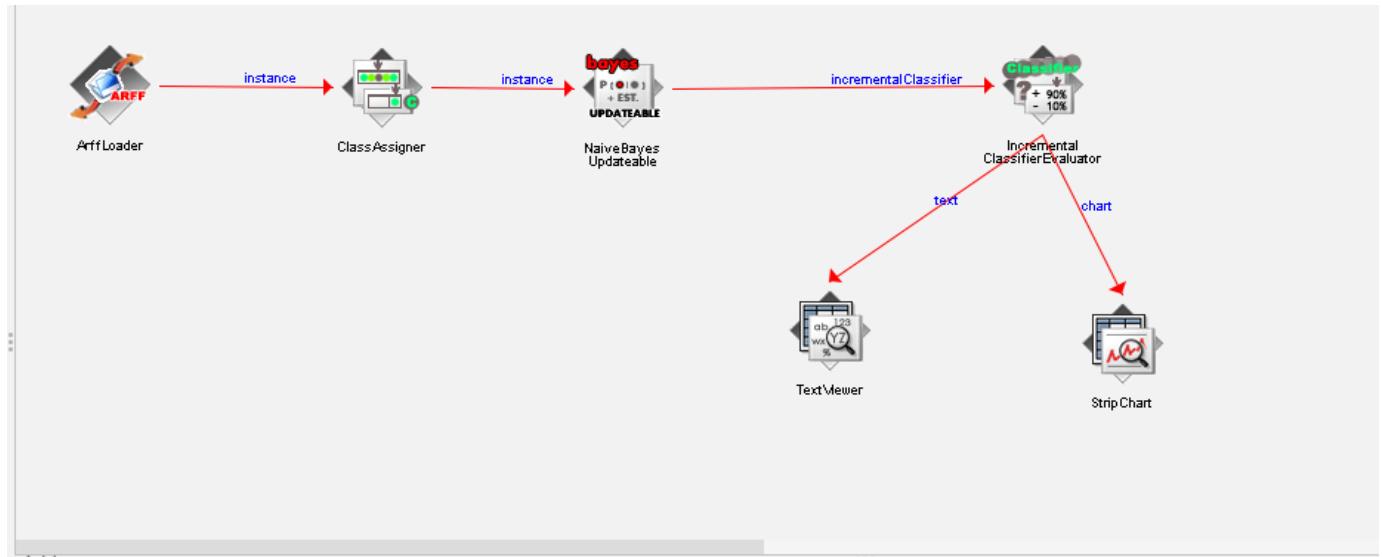
==== Confusion Matrix ===

a  b  <-- classified as
10 1 | a = yes
 4 0 | b = no
  
```

Close **Settings** **Clear results**

iv) Apply Incremental Learning and Analyze the Result

- Use an Incremental Classifier (like NaiveBayesUpdateable).
- Add the **IncrementalClassifier** component and connect it to the data source.
- Configure the classifier and enable incremental updates.
- Execute the process and monitor how the model is updated with new data chunks.
- Analyze performance by using an evaluation component to track learning over time.



Component	Parameters	Time	Status
AffLoader	Overall flow throughput -	-	Finished - 15 insts @ 1505 insts/sec
ClassAssigner		-	Finished - 15 insts @ 535 insts/sec
NaiveBayesUpdateable		-	Finished - 15 insts @ 833 insts/sec
IncrementalClassifierEvaluator		-	Finished - 15 insts @ 15000 insts/sec
StripChart		-	Finished - 15 insts @ 15000 insts/sec
TextViewer		-	Finished.

TextViewer

Result list

```

08:32:58.631 - Results: Na
08:40:43.989 - Results: Na
  
```

Text

```

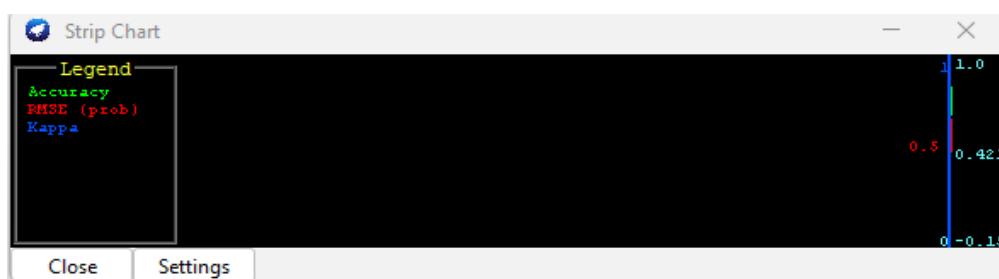
==== Performance information ===

Scheme: NaiveBayesUpdateable
Relation: labprgm2

Correctly Classified Instances      9          60      %
Incorrectly Classified Instances   6          40      %
Kappa statistic                   -0.2162
Mean absolute error               0.4936
Root mean squared error          0.6116
Total Number of Instances         15

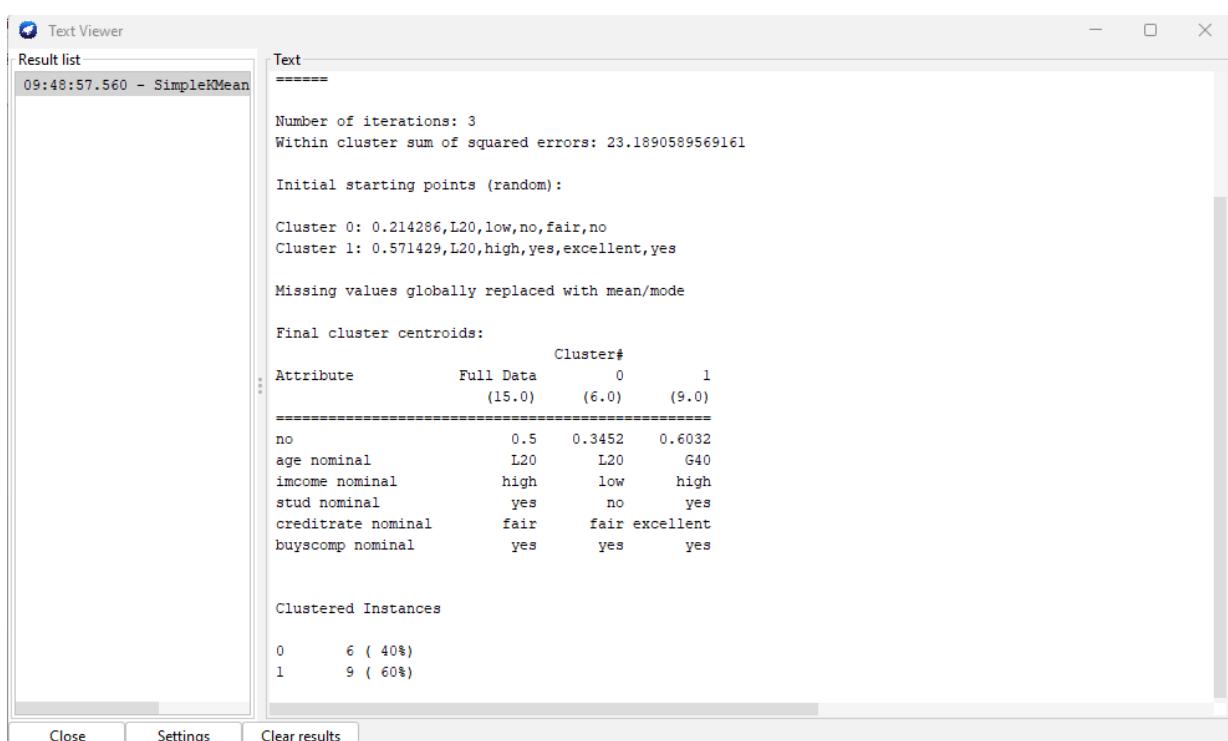
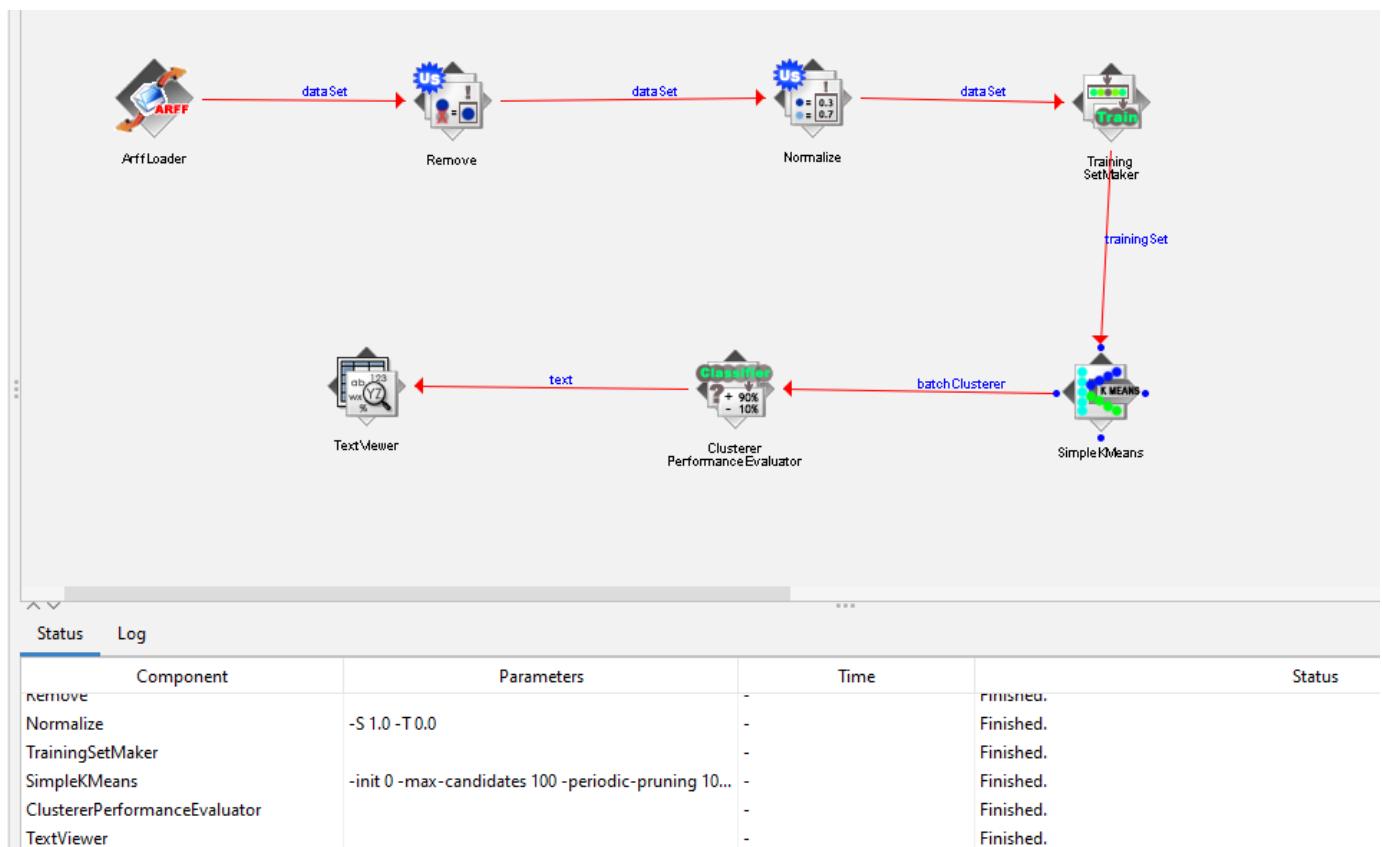
==== Confusion Matrix ===

 a b  <- classified as
9 2 | a = yes
4 0 | b = no
  
```



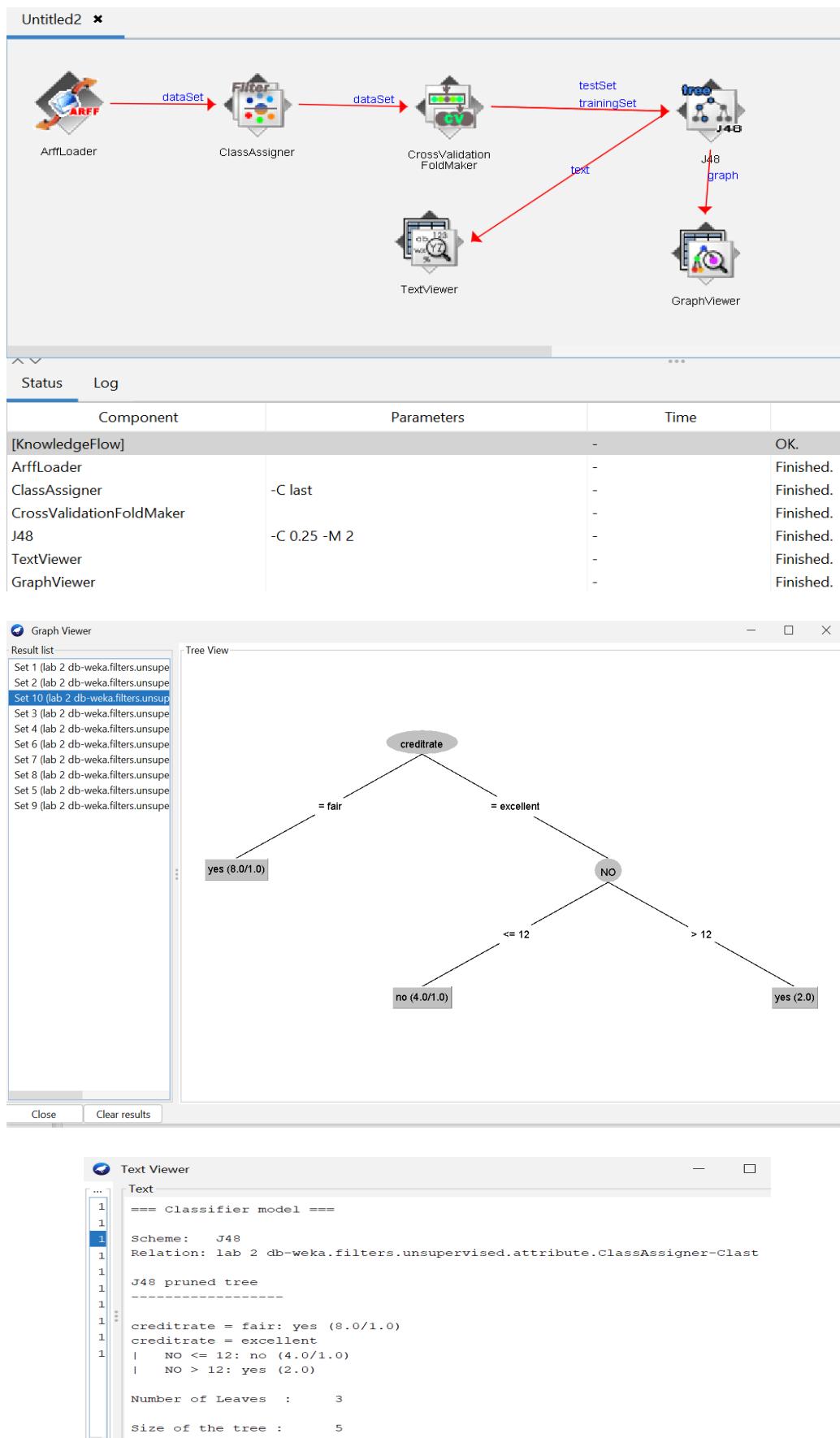
v) Do Clustering: Use Generator Properties, Two Clustering Schemes, and Result Panel

- **Remove:** Likely used to remove unwanted attributes.
- **Normalize:** Applies normalization to the data.
- **TrainingSetMaker:** Prepares the dataset for training.
- **SimpleKMeans:** This is where you are applying a clustering algorithm (K-Means).
- **ClusterPerformanceEvaluator:** Evaluates the results of clustering.
- **TextViewer:** Outputs the evaluation result.



vi) Generate classification Matrix and Construct Decision tree

- **ArffLoader** : Load your dataset.
- **J48** :Construct the decision tree classifier using the training data.
- **GraphViewer** :Visualize the results of the classification.
- **TextViewer** : Output the evaluation results for review.



vii) Perform Linear Regression and Analyze , Validate and Visualize the data

- **ArffLoader** : Load your dataset.
- **Remove** : Clean the dataset by removing unwanted attributes.
- **Normalize** : Apply normalization to prepare the data for regression analysis.
- **TrainingSetMaker**: Create a training set from your data.
- **LinearRegression** : Perform linear regression on the training data.
- **Evaluation** : Evaluate the regression model's performance.
- **TextViewer** : Output the evaluation results for review.

