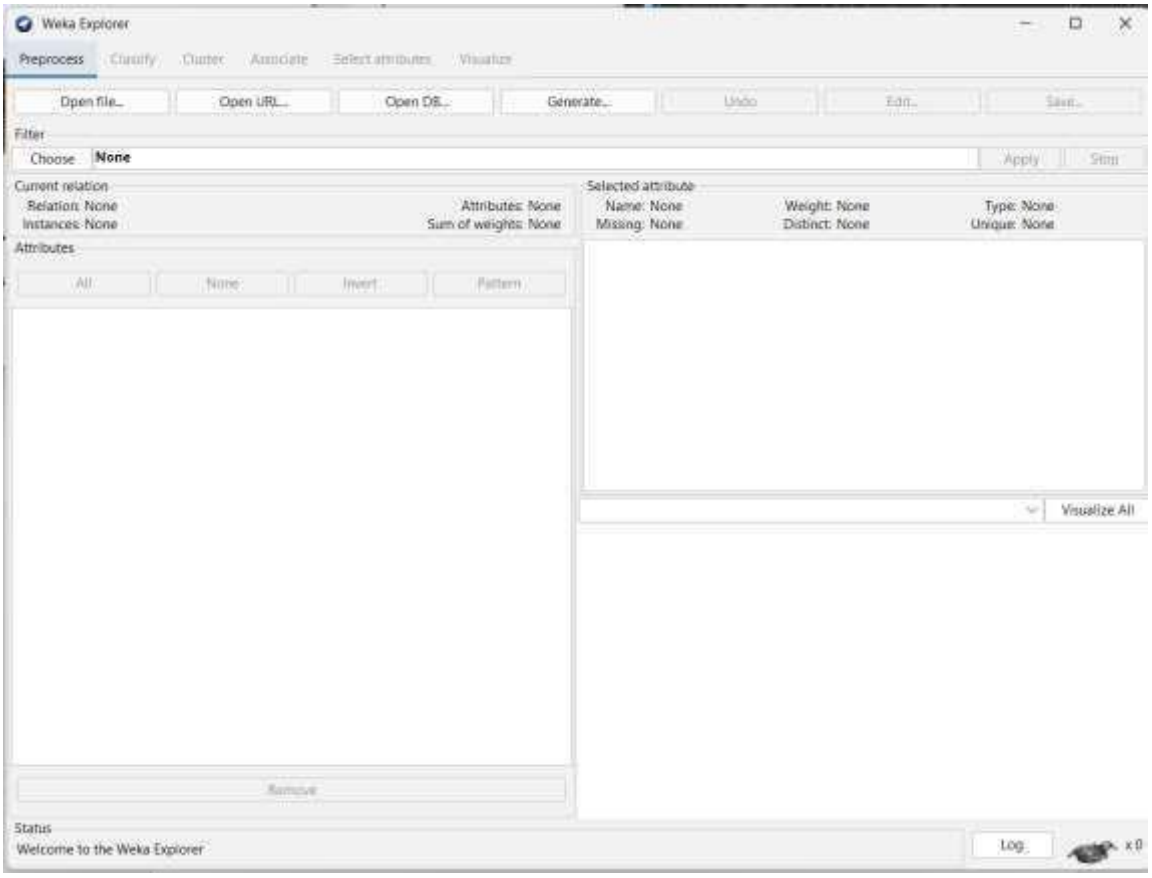
Ex No: 1&2 **Date:**	**DATA EXPLORATION AND INTEGRATION, DATA VALIDATION WITH WEKA**

INTRODUCTION:

Invoke Weka from the Windows Start menu (on Linux or the Mac, double-click weka.jar or weka.app, respectively). This starts up the Weka GUI Chooser.Click the Explorer button to enter the Weka Explorer. The Preprocess panel opens up when the Explorer interface is started. Click the open file option and starts perform the respective operations, this can be shown below the figure.



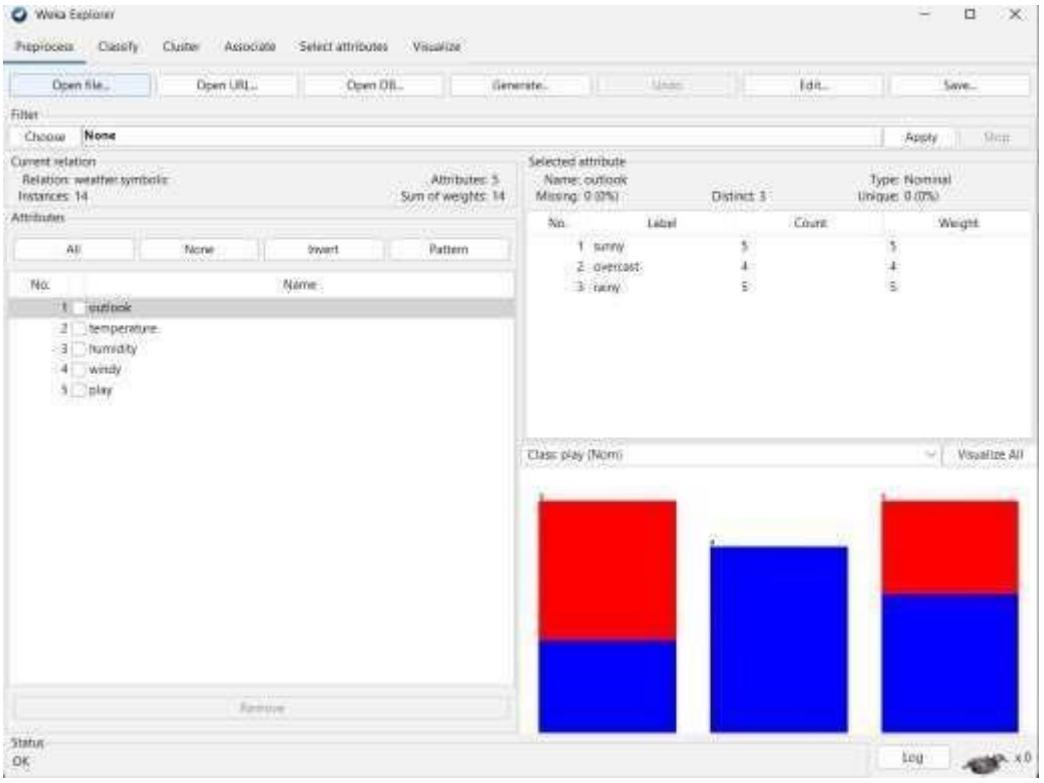
THE PANELS:

1. PREPROCESS.
2. CLASSIFY.
3. CLUSTER.
4. ASSOCIATE.
5. SELECT ATTRIBUTE
6. VISUALIZE

PREPROCESS PANEL

LOADING THE DATA-SET:

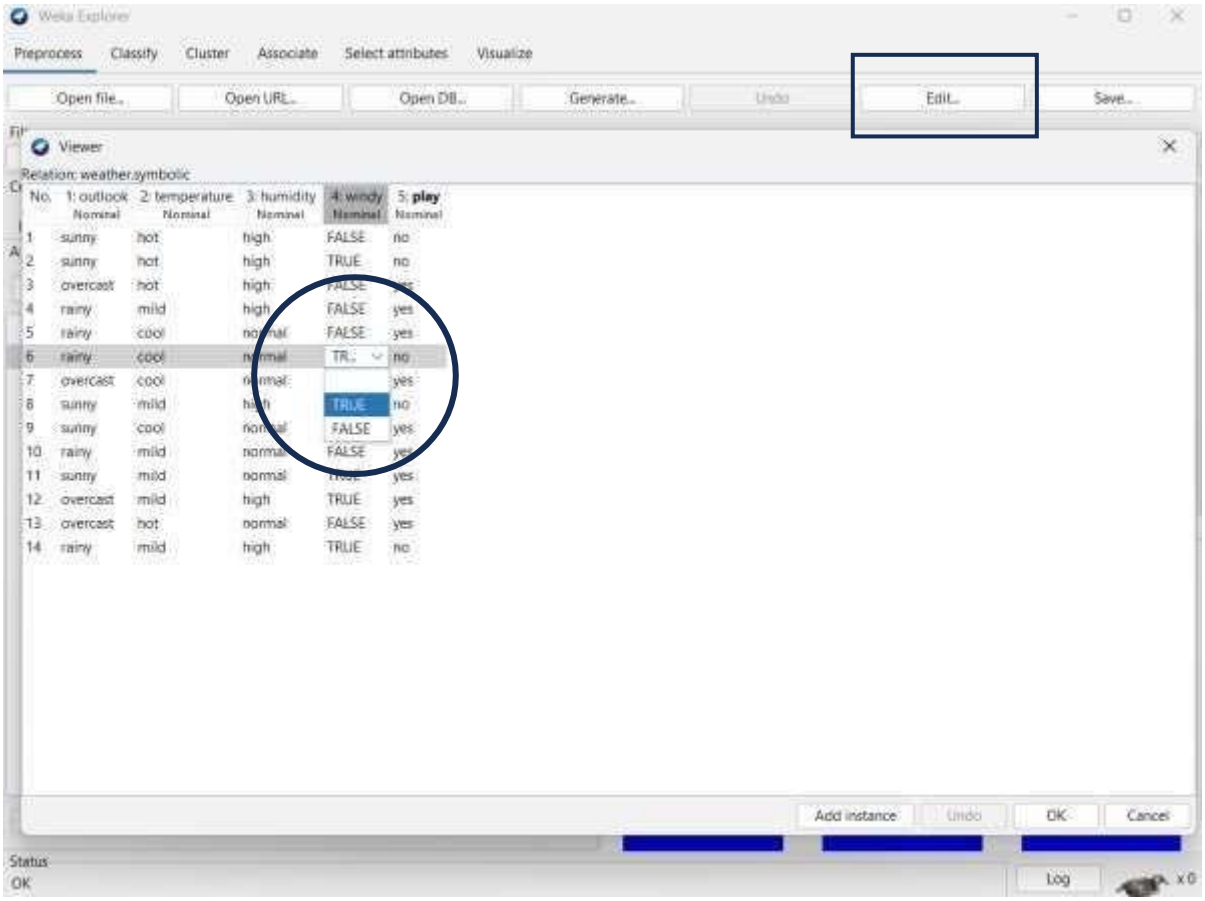
Load the dataset from the data folder at the open file option, choose the required dataset from the list of datasets. Here, for the experiment we chose *"Weather.nominal.arff"* dataset and analyse the attributes from the dataset.



As the result shows, the weather data has 14 instances, and 5 attributes called *outlook*, *temperature*, *humidity*, *windy*, and *play*. Click on the name of an attribute in the left subpanel to see information about the selected attribute on the right, such as its values and how many times an instance in the dataset has a particular value. This information is also shown in the form of a histogram. All attributes in this dataset are "nominal"—that is, they have a predefined finite set of values. The last attribute, *play*, is the "class" attribute; its value can be *yes* or *no*.

DATA SET EDITOR:

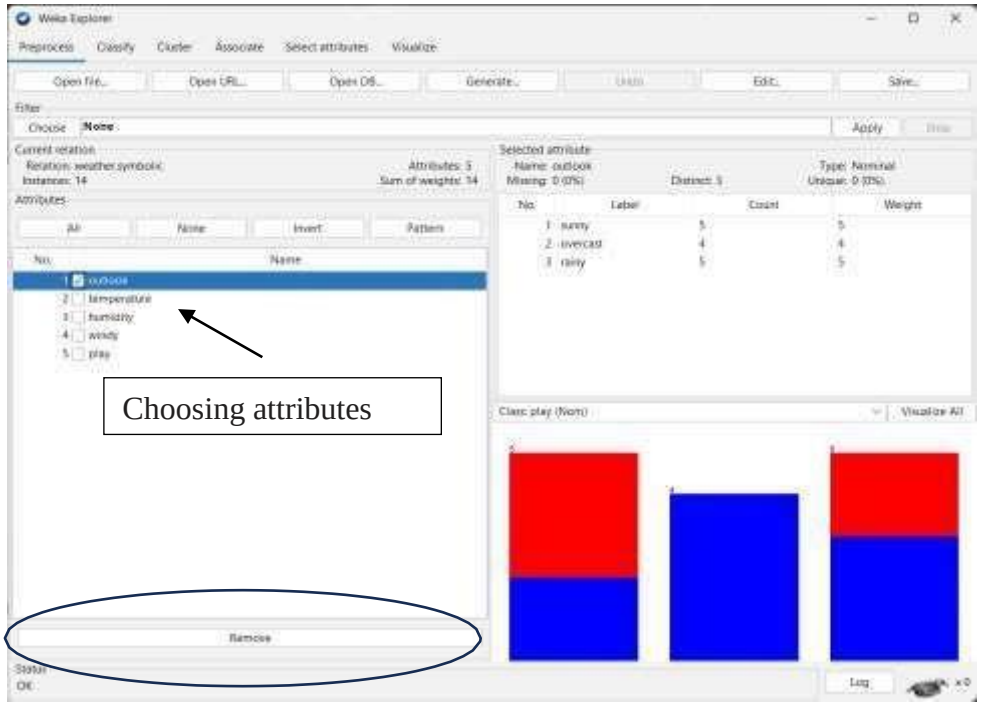
It is possible to view and edit an entire dataset from within Weka. To do this, load the *weather.nominal.arff* file again. Click the *Edit* button from the row of buttons at the top of the Preprocess panel. This opens a new window called Viewer, which lists all instances of the weather data.



APPLYING FILTER:

As you know, Weka "filters" can be used to modify datasets in a systematic fashion--that is, they are data Preprocessing tools. Reload the *weather.nominal* dataset, and let's remove an attribute from it. The appropriate filter is called *Remove*; its full name is:

```
weka.filters.unsupervised.attribute.Rem
```

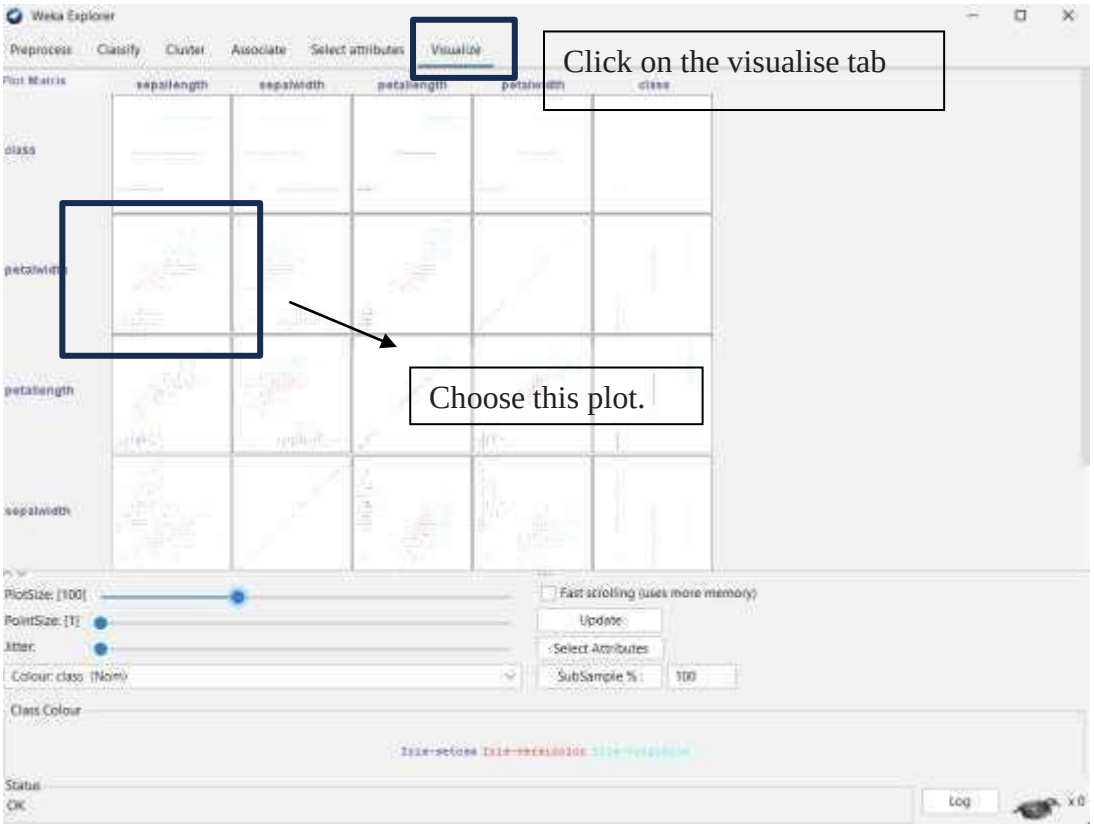


THE VISUALISE PANEL:

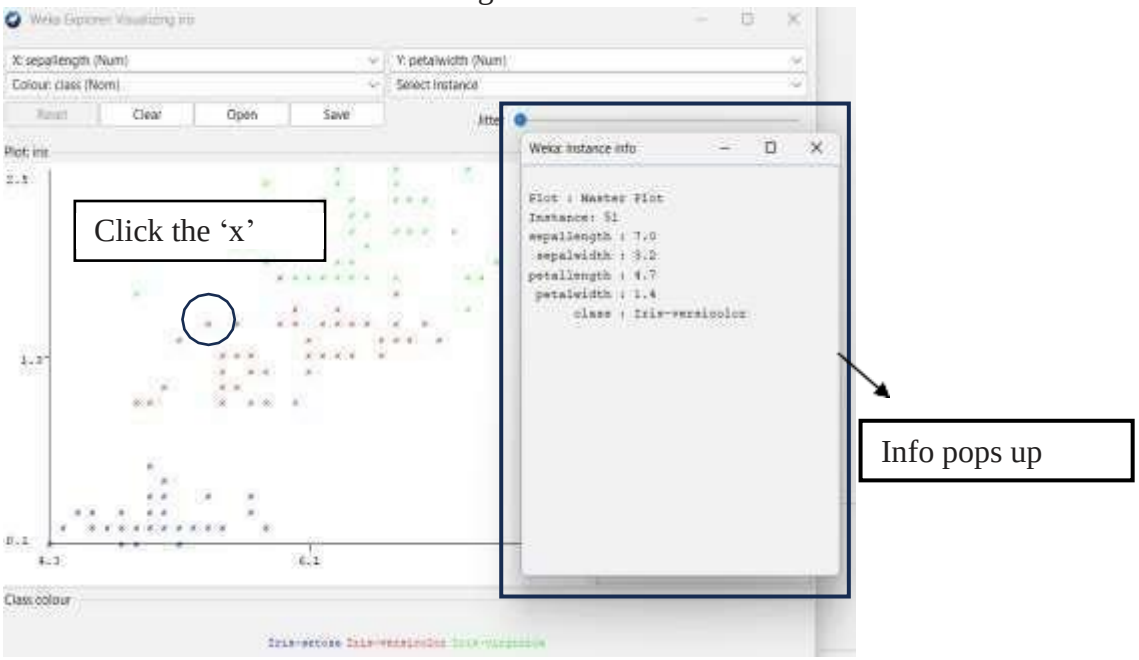
Now take a look at Weka's data visualization facilities. These work best with numeric data, so we use the iris data. Load iris.arff, which contains the iris dataset containing 50 examples of three types of Iris: Iris setosa, Iris versicolor, and Iris virginica.

1. Click the Visualize tab to bring up the Visualize panel.

2. Click the first plot in the second row to open a window showing an enlarged plot using the selected axes. Instances are shown as little crosses, the colour of which depends on the instance's class. The x-axis shows the sepal length attribute, and the y-axis shows petal width.



3. Clicking on one of the crosses opens up an Instance Info window, which lists the values of all attributes for the selected instance. Close the Instance Info window again.



The selection fields at the top of the window containing the scatter plot determine which attributes are used for the *x*- and *y*-axes. Change the *x*-axis to *petalwidth* and the *y*-axis to *petallength*. The field showing *Color: class (Num)* can be used to change the color coding.

Each of the barlike plots to the right of the scatter plot window represents a single attribute. In each bar, instances are placed at the appropriate horizontal position and scattered randomly in the vertical direction. Clicking a bar uses that attribute for the *x*-axis of the scatter plot. Right-clicking a bar does the same for the *y*-axis. Use these bars to change the *x*- and *y*-axes back to *sepallength* and *petalwidth*.

The ***Jitter*** slider displaces the cross for each instance randomly from its true position, and can reveal situations where instances lie on top of one another.

Experiment a little by moving the slider.

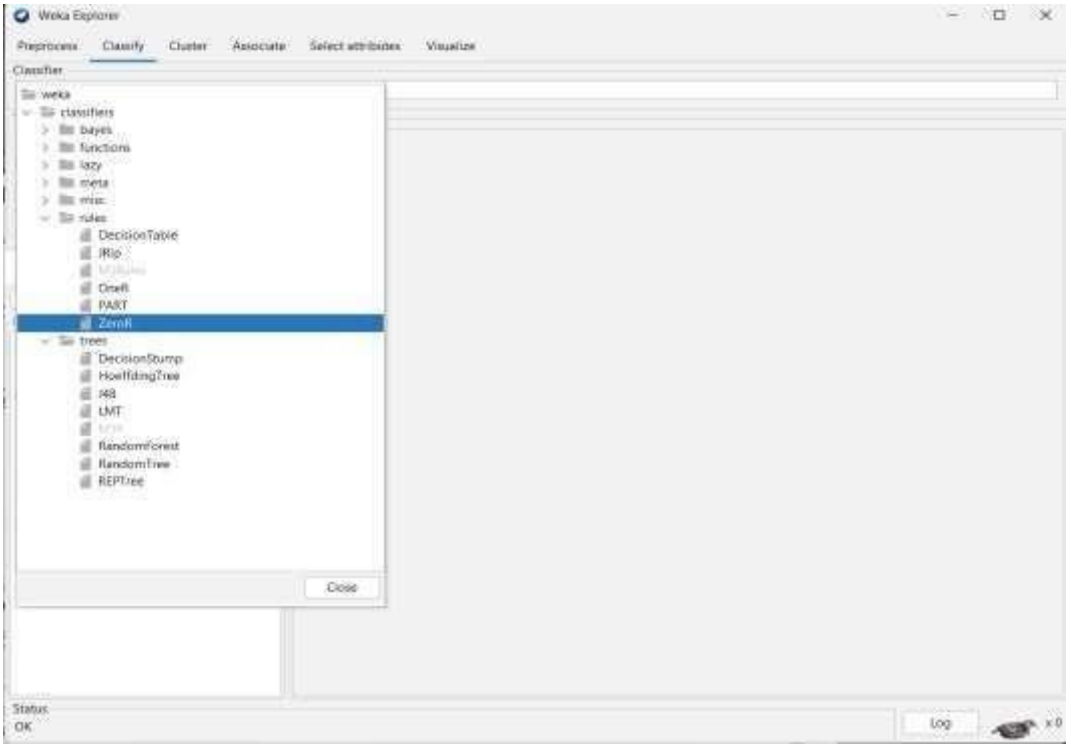
The ***Select Instance*** button and the *Reset*, *Clear*, and *Save* buttons let you modify the dataset. Certain instances can be selected and the others removed. Try the Rectangle option: Select an area by left-clicking and dragging the mouse. The *Reset* button changes into a *Submit* button. Click it, and all instances outside the rectangle are deleted. You could use *Save* to save the modified dataset to a file. *Reset* restores the original dataset.T

CLASSIFY PANEL:

Now we apply a classifier to the weather data. Load the weather data again. Go to the Preprocess panel, click the *Open file* button, and select **"*weather.nominal.arff*"** from the data directory. Then switch to the Classify panel by clicking the *Classify* tab at the top of the window.

USING THE C4.5 CLASSIFIER:

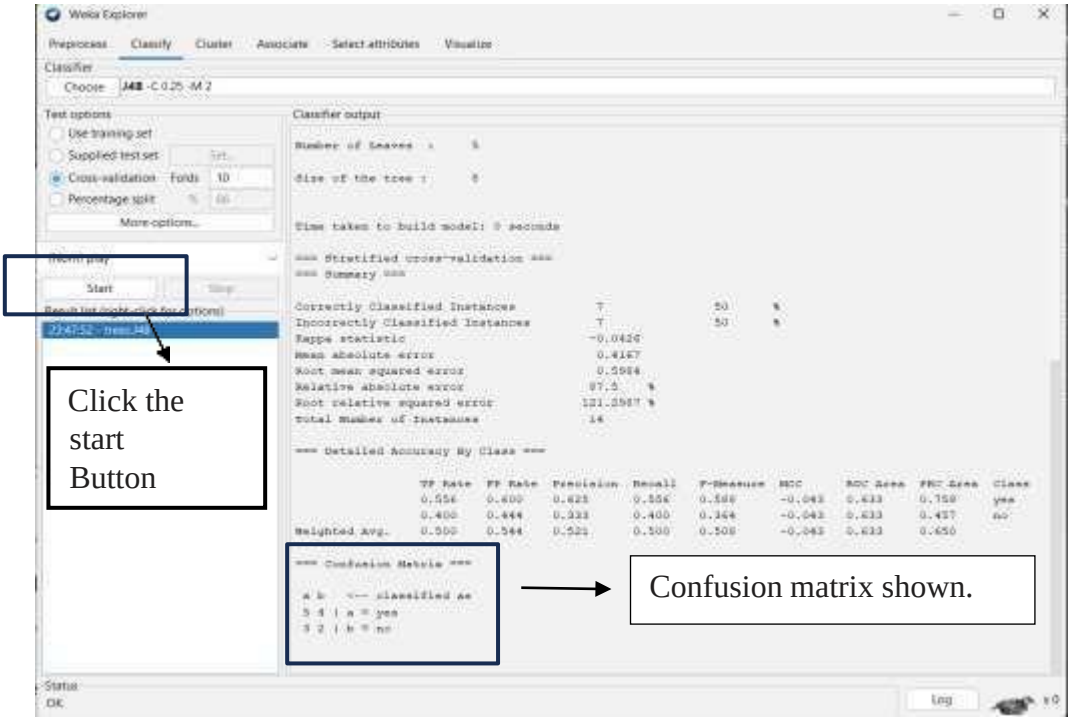
The C4.5 algorithm for building decision trees is implemented in Weka as a classifier called *J48*. Select it by clicking the *Choose* button near the top of the *Classify* tab. A dialog window appears showing various types of classifier. Click the *trees* entry to reveal its subentries, and click *J48* to choose that classifier. Classifiers, like filters, are organized in a hierarchy: *J48* has the full name *weka.classifiers.trees.J48*.



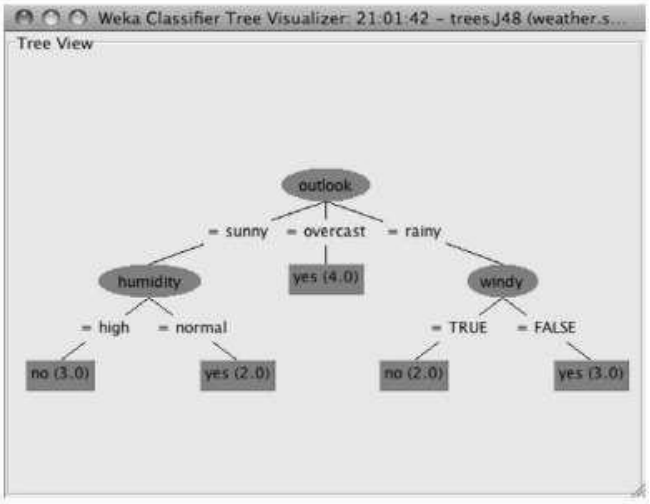
OUTPUT:

The outcome of training and testing appears in the Classifier Output box on the right. Scroll through the text and examine it. First, look at the part that describes the decision tree, reproduce in image below. This represents the decision tree that was built, including the number of instances that fall under each leaf. The textual representation is clumsy to interpret, but Weka can generate an equivalent graphical version.

Here's how to get the graphical tree. Each time the *Start* button is pressed and a new classifier is built and evaluated, a new entry appears in the Result List panel in the lower left corner.



BUILTING THE DECISION TREE:



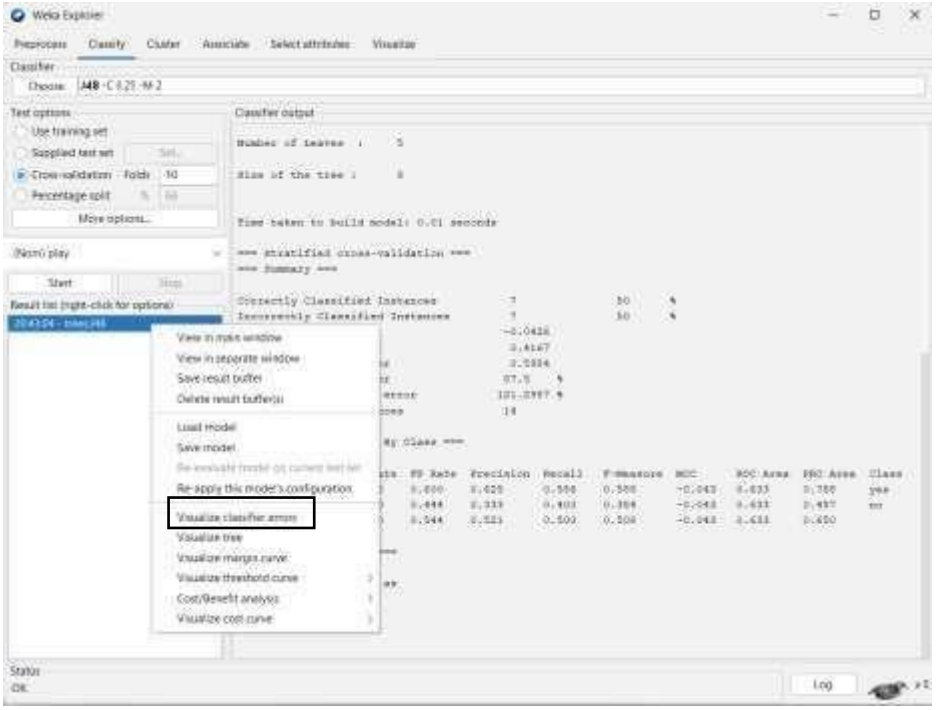
Setting the Test Method:

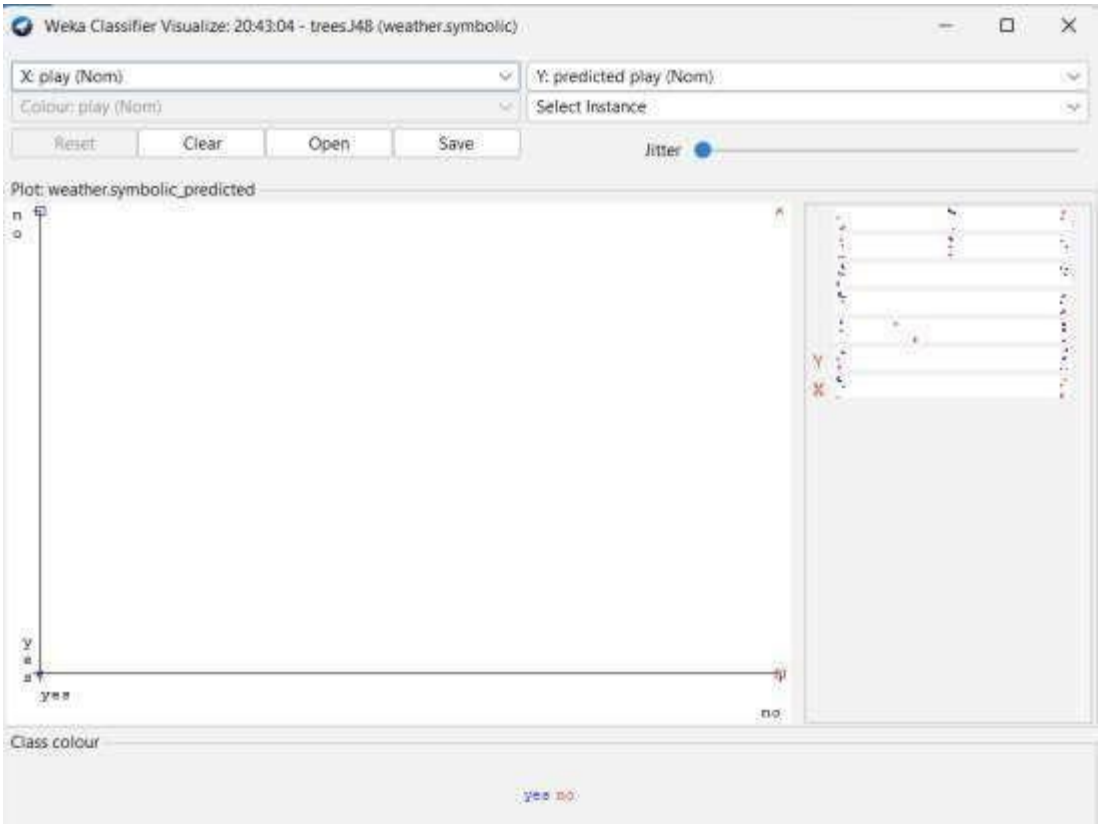
When the *Start* button is pressed, the selected learning algorithm is run and the dataset that was loaded in the Preprocess panel is used with the selected test protocol.

For example, in the case of tenfold cross-validation this involves running the learning algorithm 10 times to build and evaluate 10 classifiers. A model built from the *full* training set is then printed into the Classifier **Output area:** This may involve running the learning algorithm one final time. The remainder of the output depends on the test protocol that was chosen using test options.

VISUALISE THE ERRORS:

Right-click the *trees.J48* entry in the result list and choose *Visualize classifier errors*. A scatter plot window pops up. Instances that have been classified correctly are marked by little crosses; ones that are incorrect are marked by little squares.





CLUSTER PANEL :

Clustering Data :

WEKA contains "clusterers" for finding groups of similar instances in a dataset. The clustering schemes available in WEKA are,

- ✓ *k*-Means,
- ✓ EM,
- ✓ Cobweb,
- ✓ *X*-means,
- ✓ Farthest First.

Clusters can be visualized and compared to "true" clusters (if given). Evaluation is based on log likelihood if clustering scheme produces a probability distribution.

For this exercise we will use customer data that is contained in "customers.arff" file and analyze it with "***k-means"*** clustering scheme.

Steps:

(i) Select the file from WEKA

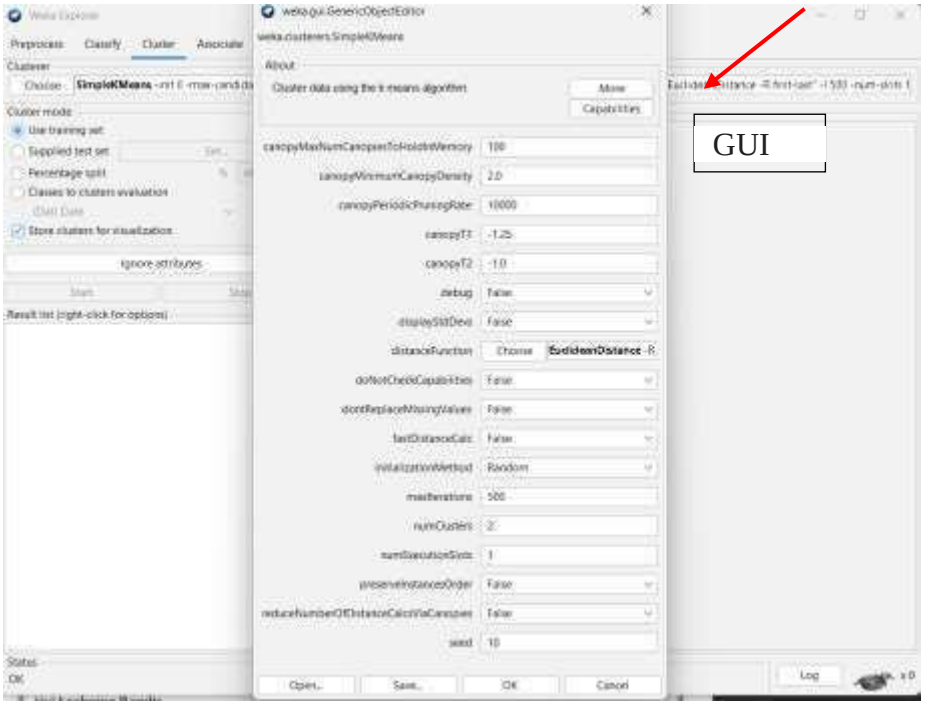
In 'Preprocess' window click on 'Open file…' button and select ***"weather.arff"*** file. Click 'Cluster' tab at the top of WEKA Explorer window.

(ii) Choose the Cluster Scheme.

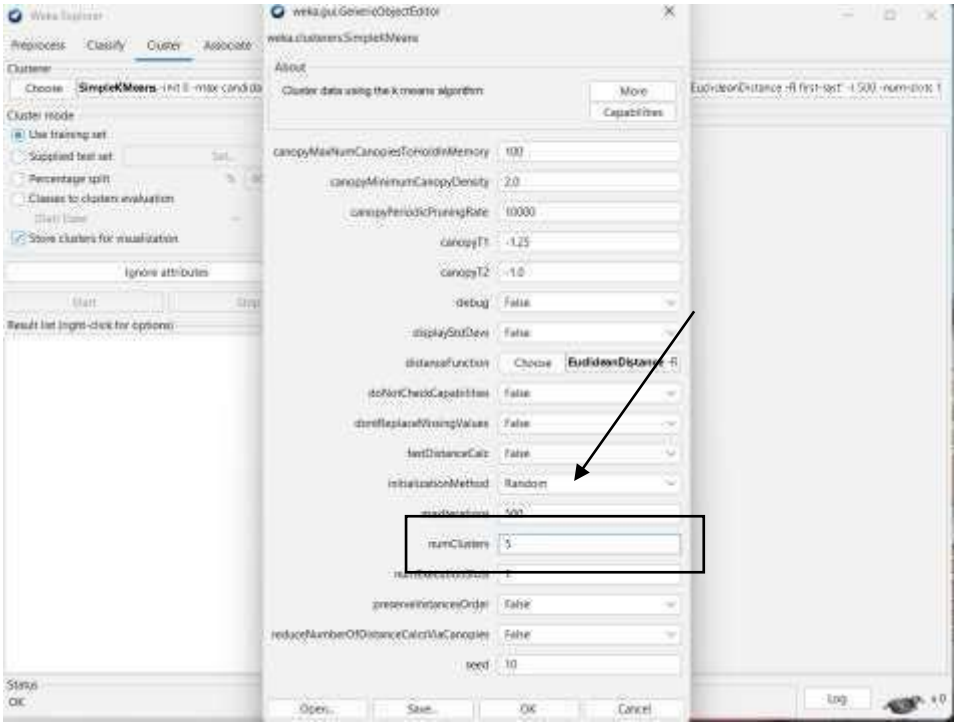
1. In the 'Clusterer' box click on 'Choose' button. In pull-down menu select WEKA Clusterers, and select the cluster scheme ***'SimpleKMeans'***. Some implementations of K-means only allow numerical values for attributes; therefore, we do not need to use a filter.

2. Once the clustering algorithm is chosen, right-click on the algorithm,***"weak.gui.GenericObjectEditor"*** comes up to the screen.

.

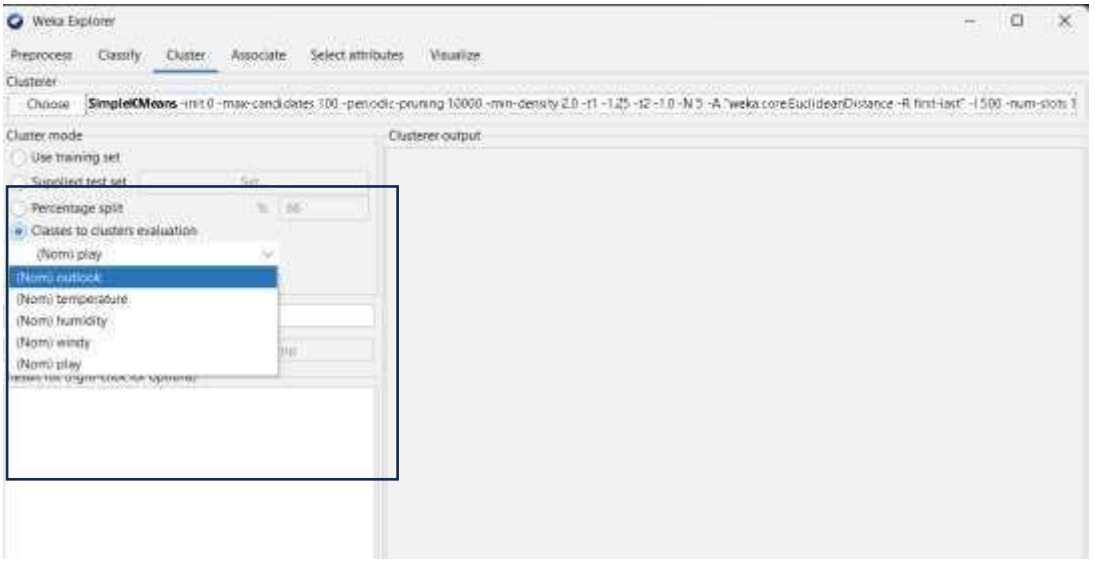


3. Set the value in ***"numClusters"*** box to 5 (Instead of default 2) because you have five clusters in your .arff file. Leave the value of 'seed' as is. The seed value is used in generating a random number, which is used for making the initial assignment of instances to clusters. Note that, in general, K-means is quite sensitive to how clusters are initially assigned. Thus, it is often necessary to try different values and evaluate the results.



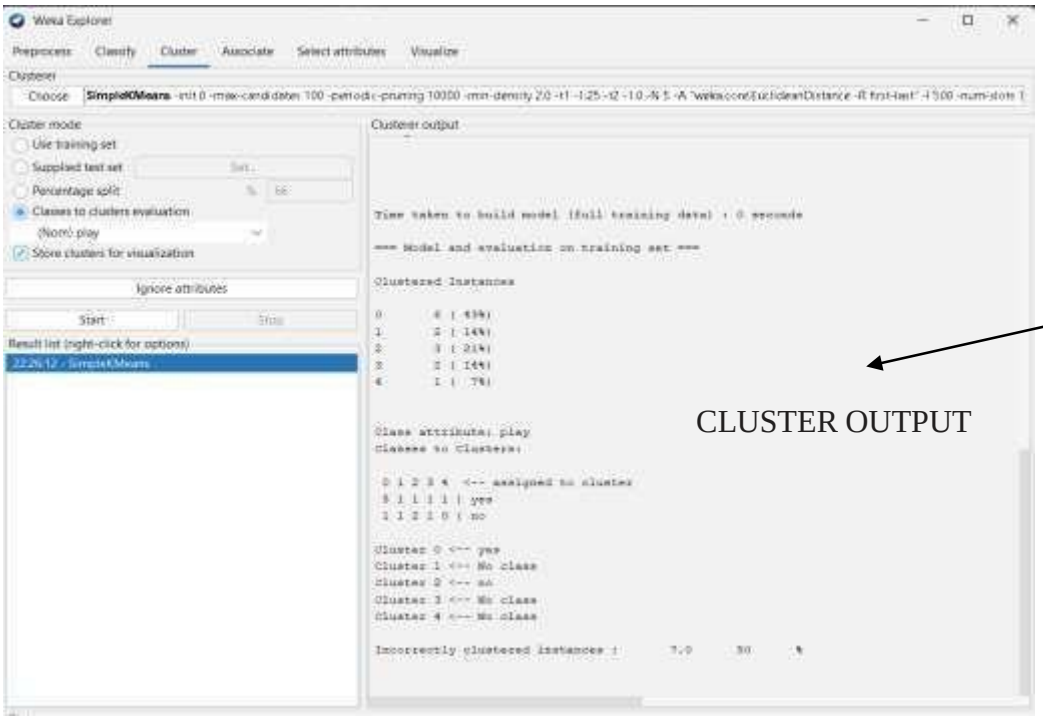
(iii) Setting the test options.

1. Before you run the clustering algorithm, you need to choose 'Cluster mode'.
2. Click on 'Classes to cluster evaluation' radio-button in 'Cluster mode' box and select 'Play' in the pull-down box below. It means that you will compare how well the chosen clusters match up with a pre-assigned class (''Play'') in the data.
3. Once the options have been specified, you can run the clustering algorithm. Click on the 'Start' button to execute the algorithm.



4. When training set is complete, the 'Cluster' output area on the right panel of 'Cluster'

window is filled with text describing the results of training and testing. A new entry appears in the ‘Result list’ box on the left of the result. These behave just like their classification counterparts.



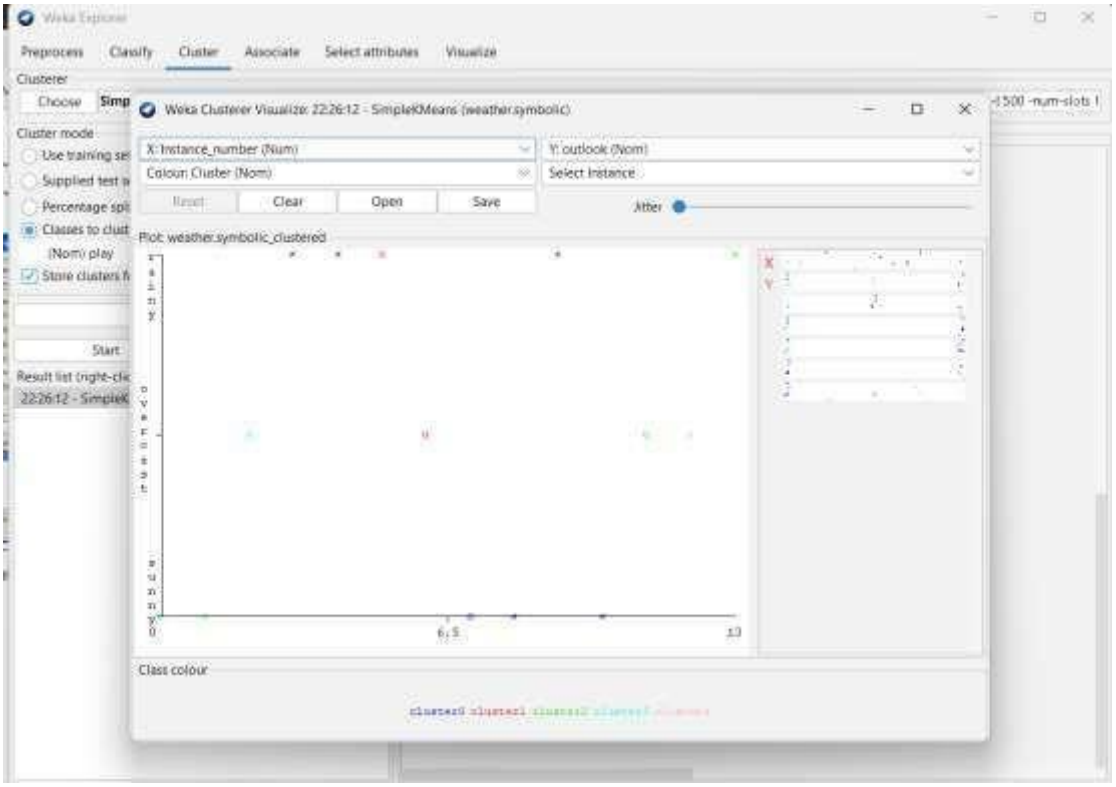
CLUSTER OUTPUT

(iv) Analysing Results.

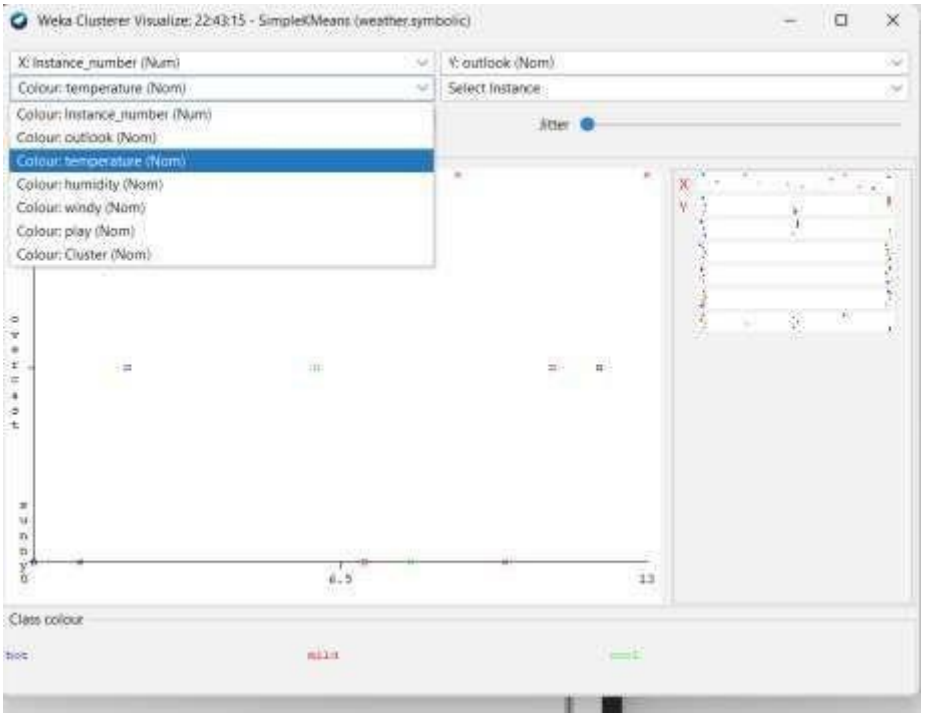
The clustering model shows the centroid of each cluster and statistics on the number and percentage of instances assigned to different clusters. Cluster centroids are the mean vectors foreach cluster; so, each dimension value and the centroid represent the mean value for that dimension in the cluster.

(v) Visualisation of Results.

1. Another way of representation of results of clustering is through visualization.
2. Right-click on the entry in the ‘Result list’ and select ‘Visualize cluster assignments’ in the pull-down window. This brings up the ‘Weka Clusterer Visualize’ window.

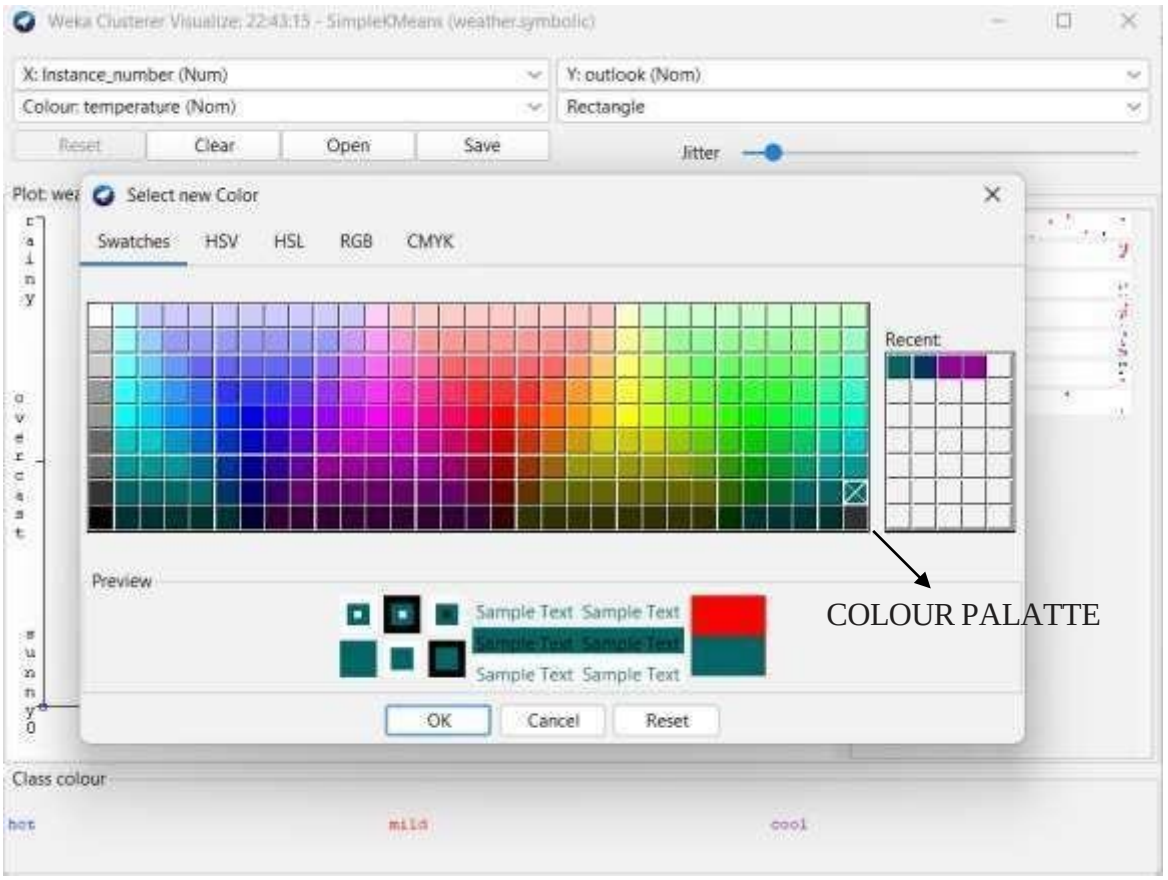


3. On the 'Weka Clusterer Visualize' window, beneath the X-axis selector there is a drop down list, ***'Colour'***, for choosing the color scheme. This allows you to choose the colour of points based on the attribute selected.

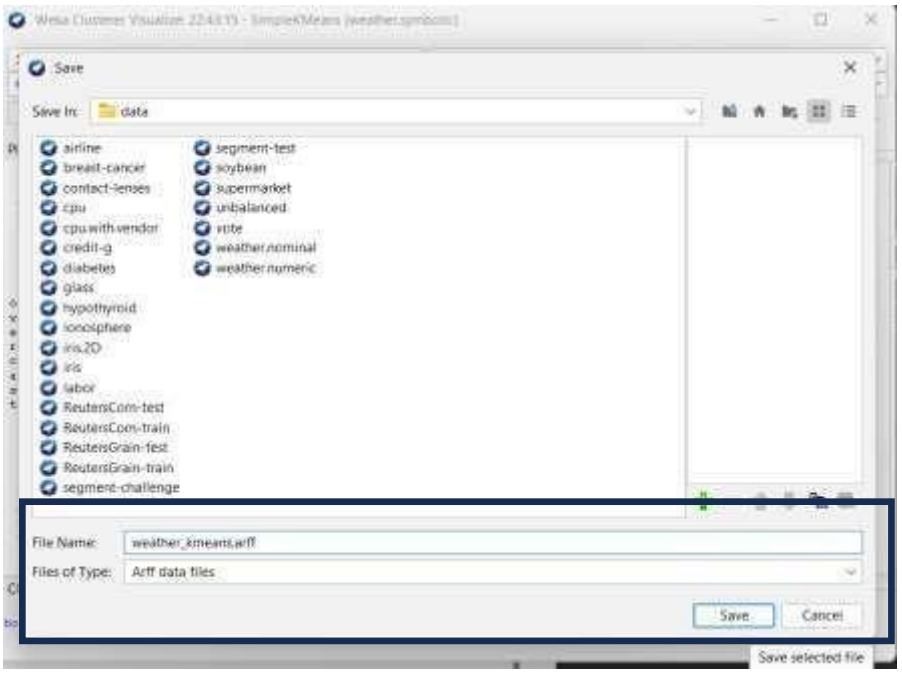


4. Below the plot area, there is a legend that describes what values the colours correspond to. In your example, Seven different colours represent Seven numbers (number of children). For better visibility you should change the colour of label '3'.

5. Left click on ‘3’ in the ‘Class colour’ box and select lighter color from the color palette.



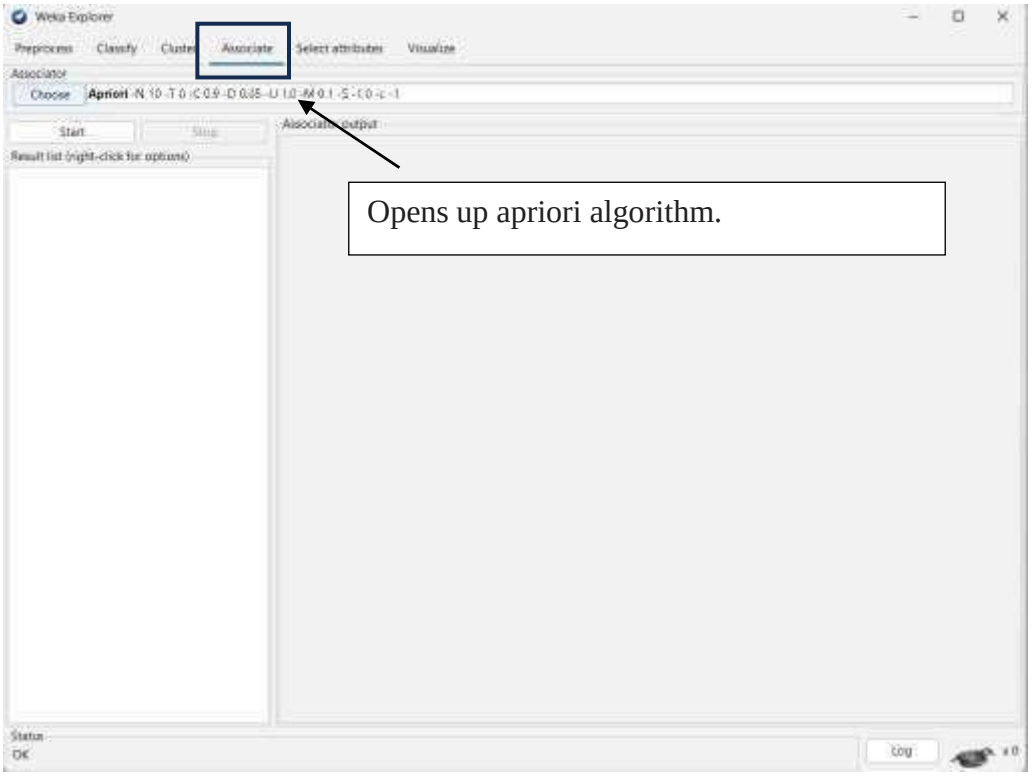
6. You may want to save the resulting data set, which included each instance along with its assigned cluster. To do so, click ‘Save’ button in the visualization window and save the result as the file ***“weather_kmeans.arff”.***



ASSOCIATION PANEL

(i) opening the file

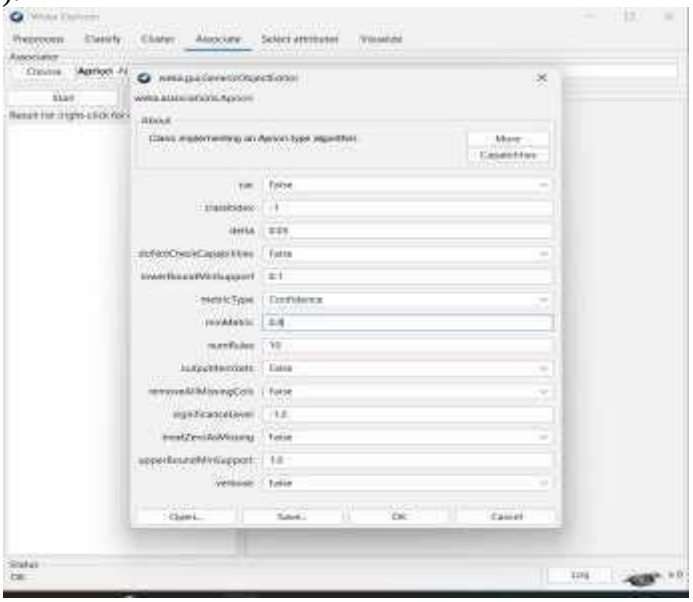
1. Click ‘Associate’ tab at the top of ‘WEKA Explorer’ window. It brings up interface for the Apriori algorithm.



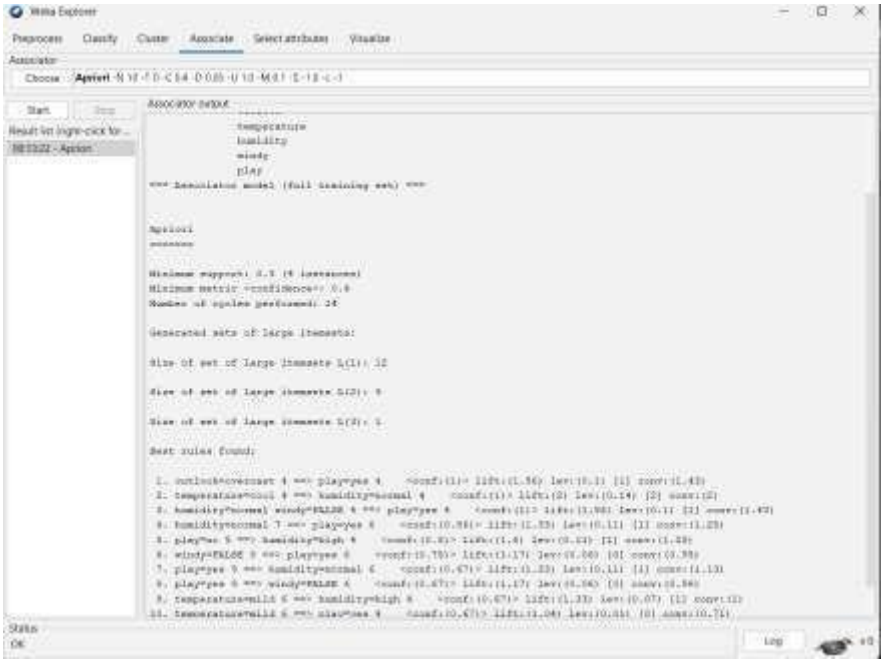
2. The association rule scheme cannot handle numeric values; therefore, for this exercise you will use grocery store data from the ***“weather.arff”*** file where all values are nominal.Open ***“weather.arff”*** file.

(ii) setting the test-options

1. Right-click on the ‘Associator’ box, ‘GenericObjectEditor’ appears on your screen. In the dialog box, change the value in ‘minMetric’ to 0.4 for confidence = 40%. Make sure that the default value of rules is set to 100. The upper bound for minimum support ‘upperBoundMinSupport’ should be set to 1.0 (100%) and ‘lowerBoundMinSupport’ to 0.1. Apriori in WEKA starts with the upper bound support and incrementally decreases support (by delta increments, which by default is set to 0.05 or 5%).



4. The algorithm halts when either the specified number of rules is generated, or the lower bound for minimum support is reached. The 'significanceLevel' testing option is only applicable in the case of confidence and is (-1.0) by default (not used).
5. Once the options have been specified, you can run Apriori algorithm. Click on the 'Start' button to execute the algorithm.



(iii) Analysing the results

```
=== Run information ===

Scheme:       weka.associations.Apriori -N 10 -T 0 -C 0.4 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1
Relation:     weather.symbolic
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
```

The results for Apriori algorithm are the following:

-> First, the program generated the sets of large itemsets found for each support size considered. In this case five item sets of three items were found to have the required minimum support.

->By default, Apriori tries to generate ten rules. It begins with a minimum support of 100% of the data items and decreases this in steps of 5% until there are at least ten rules with the required minimum confidence, or until the support has reached a lower bound of 10% whichever occurs first. The minimum confidence is set 0.4 (40%).

->As you can see, the minimum support decreased to 0.3 (30%), before the required number of rules can be generated. Generation of the required number of rules involved a total of 14 iterations.

->The last part gives the association rules that are found. The number preceding = => symbol indicates the rule's support, that is, the number of items covered by its premise. Following the rule is the number of those items for which the rule's consequent holds as well. In the parentheses there is a confidence of the rule.