**Roll No:** 60 **Name:** Subrato Tapaswi **Practical:** 6

## Experiment 6

**Aim:** Using open-source tools to implement clustering algorithms (k-means).

**Theory**:

The GUI Chooser consists of four buttons—one for each of the four major Weka applications—and four menus.


The buttons can be used to start the following applications:

**Explorer** An environment for exploring data with WEKA

**Experimenter** An environment for performing experiments and conduct- ing statistical tests between learning schemes.

**KnowledgeFlow** This environment supports essentially the same functions as the Explorer but with a drag-and-drop interface. One advantage is that it supports incremental learning.

**SimpleCLI** Provides a simple command-line interface that allows direct execution of WEKA commands for operating systems that do not provide their own command line interface.

**Roll No:** 60 **Name:** Subrato Tapaswi **Practical:** 6

## Explorer:

At the very top of the window, just below the title bar, is a row of tabs. The tabs are as follows:

- **Preprocess**. Choose and modify the data being acted on.
- **Classify**. Train and test learning schemes that classify or perform regression.
- **Cluster**. Learn clusters for the data.
- **Associate**. Learn association rules for the data.
- **Select attributes**. Select the most relevant attributes in the data.
- **Visualize**. View an interactive 2D plot of the data.

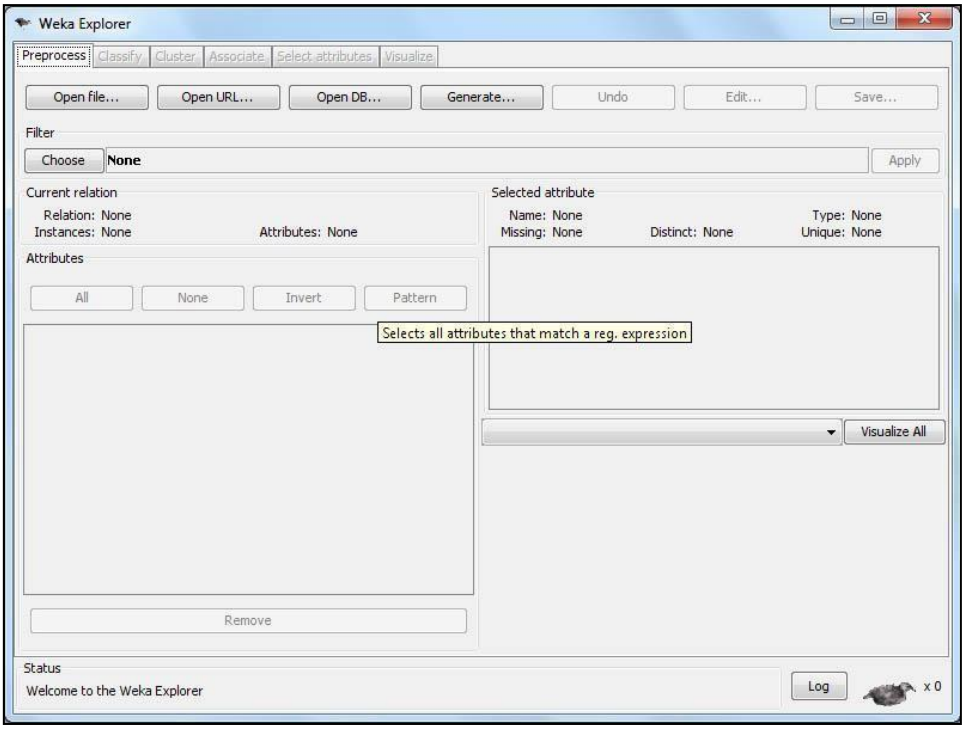
## Loading Data:

The first four buttons at the top of the preprocess section enable you to load data into WEKA:

**Open file....** Brings up a dialog box allowing you to browse for the data file on the local file system.

**Open URL....** Asks for a Uniform Resource Locator address for where the data is stored.

**Open DB.....** Reads data from a database. (Note that to make this work you might have to edit the file in weka/experiment/DatabaseUtils.props.)



**Roll No:** 60 **Name:** Subrato Tapaswi **Practical:** 6

## K-means Clustering

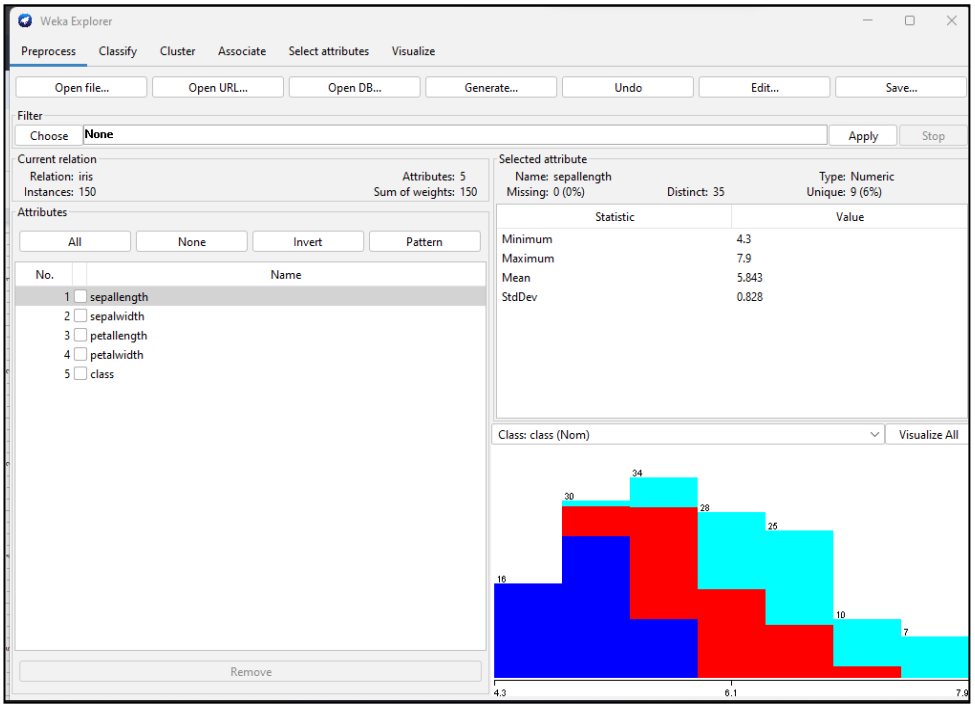
The algorithm will categorize the items into k groups or clusters of similarity. To calculate that similarity, we will use the euclidean distance as measurement.

The algorithm works as follows:

First, we initialize k points, called means or cluster centroids, randomly.
We categorize each item to its closest mean and we update the mean's coordinates, which are the averages of the items categorized in that cluster so far.
We repeat the process for a given number of iterations and at the end, we have our clusters.
The "points" mentioned above are called means because they are the mean values of the items categorized in them. To initialize these means, we have a lot of options. An intuitive method is to initialize the means at random items in the data set. Another method is to initialize the means at random values between the boundaries of the data set (if for a feature x the items have values in [0,3], we will initialize the means with values for x at [0,3]).

To update a mean, we need to find the average value for its feature, for all the items in the mean/cluster. We can do this by adding all the values and then dividing by the number of items, or we can use a more elegant solution. We will calculate the new average without having to re-add all the values, by doing the following:
$m = (m*(n-1)+x)/n$

where m is the mean value for a feature, n is the number of items in the cluster, and x is the feature value for the added item. We do the above for each feature to get the new mean.



**Name: - WEKA Clustering K means**

**=== Run information ===**

- Scheme: weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10
- Relation: iris
- Instances: 150
- Attributes: 5
  sepallength
  sepalwidth
  petallength
  petalwidth
  class
- Test mode: evaluate on training data

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

**======k-Means======**

Number of iterations: 7
Within cluster sum of squared errors: 62.1436882815797

Initial starting points (random):

Cluster 0: 6.1,2.9,4.7,1.4,Iris-versicolor
Cluster 1: 6.2,2.9,4.3,1.3,Iris-versicolor

Missing values globally replaced with mean/mode

Final cluster centroids:

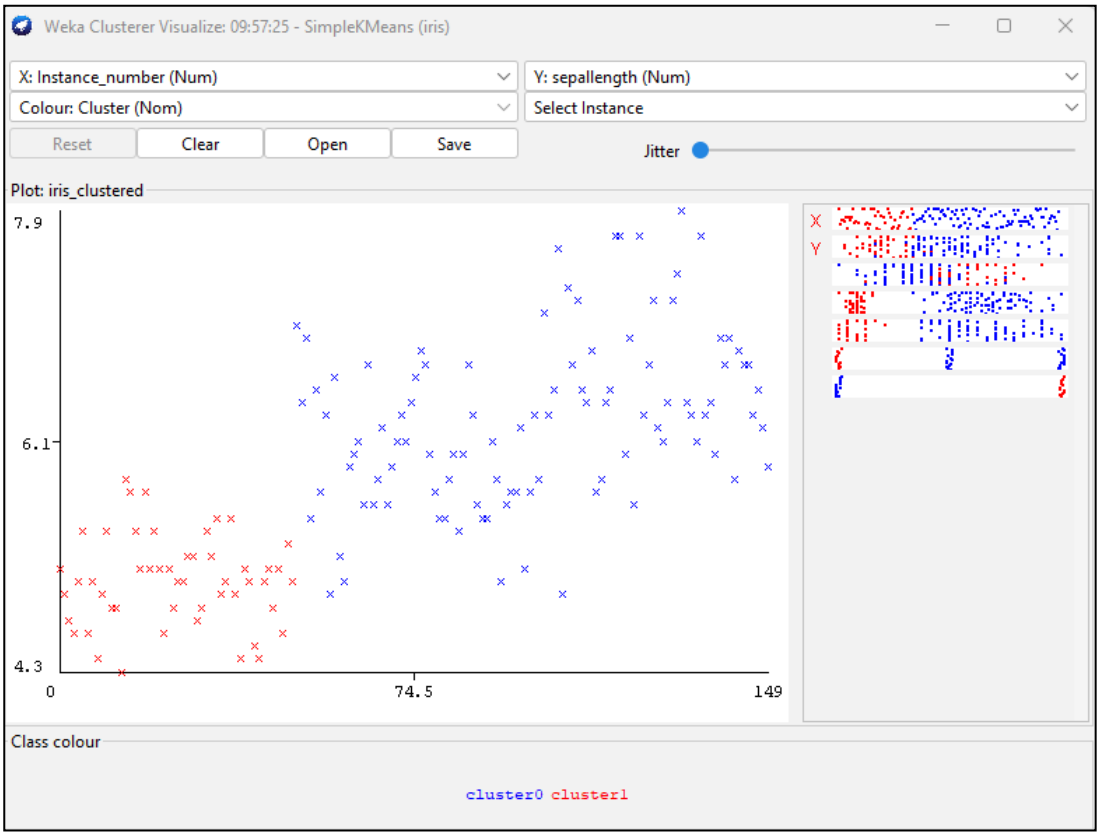
| | | Cluster# | |
|---|---|---|---|
| Attribute | Full Data | 0 | 1 |
| | (150.0) | (100.0) | (50.0) |
| sepallength | 5.8433 | 6.262 | 5.006 |
| sepalwidth | 3.054 | 2.872 | 3.418 |
| petallength | 3.7587 | 4.906 | 1.464 |
| petalwidth | 1.1987 | 1.676 | 0.244 |
| class | Iris-setosa | Iris-versicolor | Iris-setosa |

Time taken to build model (full training data) : 0 seconds

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

Clustered Instances

0 100 ( 67%)
1 50 ( 33%)




**Conclusion:**
Thus we learned and successfully implemented the K means clustering algorithm in weka open source in the iris dataset. We can easy visualize the the cluster in above graph.