# **CHA2555 Artificial Intelligence**

# **Practical 18: Clustering 1**

**Activity 1:**

Given the following 1D data: {6, 8, 18, 28, 12, 32, 24}, choose your own initial centroids and perform *k*-means. E.g., you may start with a pair of initial centroids (18, 20), perform the k-means procedures until it converges. The following table can help you find the final centroids following the k-means algorithm, and you can create more when you work on your initial centroids. Try other centroids of your own choice (Stay within the range 6 to 32, round up/down averages if needed) and compare if the final results are consistent.

Iterate until converged:

1. Compute distance from all data points to all *k* centroids
2. For each **data point**, assign it to the cluster whose current centroid it is nearest
3. For each **centroid**, compute the average (mean) of all points assigned to it
4. Replace the *k* centroids with the new averages

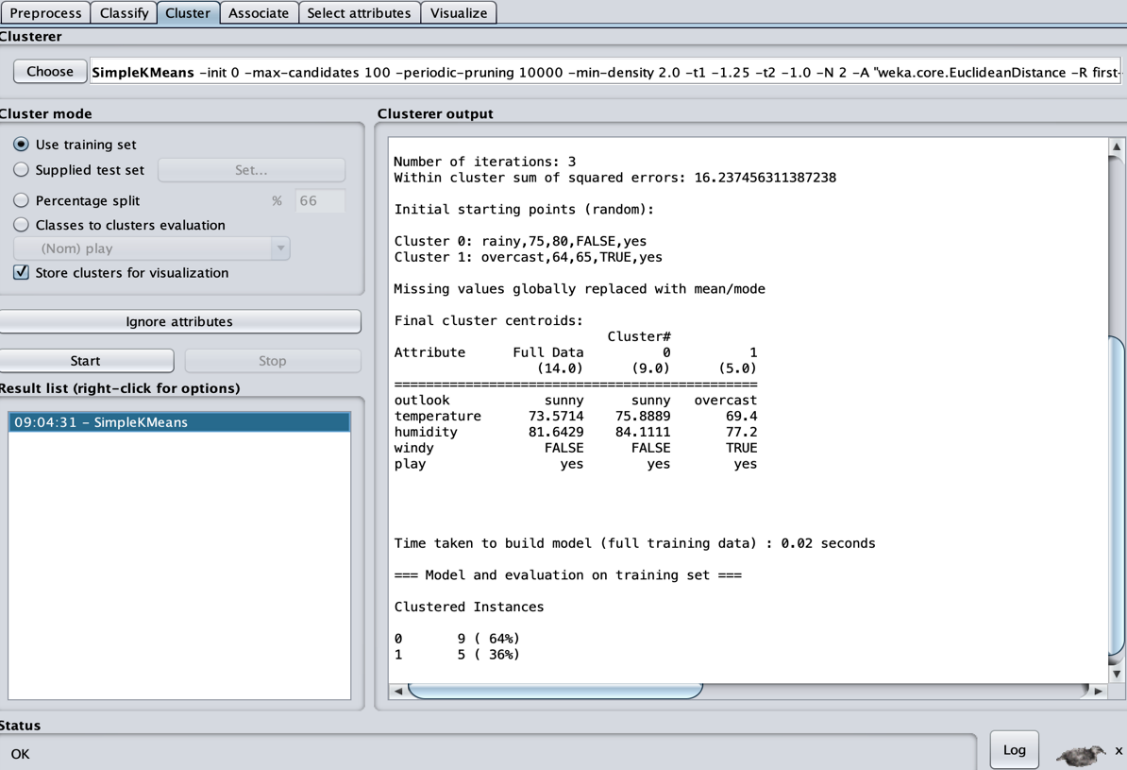
With initial centroids (18, 20)

|  |  |  |
| --- | --- | --- |
| Centroids | Number of iterations | Cluster assignment |
| 18, 20 | 1 | cluster 1 (18): 6, 8, 12, 18  cluster 2 (20): 24, 28, 32 |
|  |  |  |
|  |  |  |
|  |  |  |

**Activity 2:**

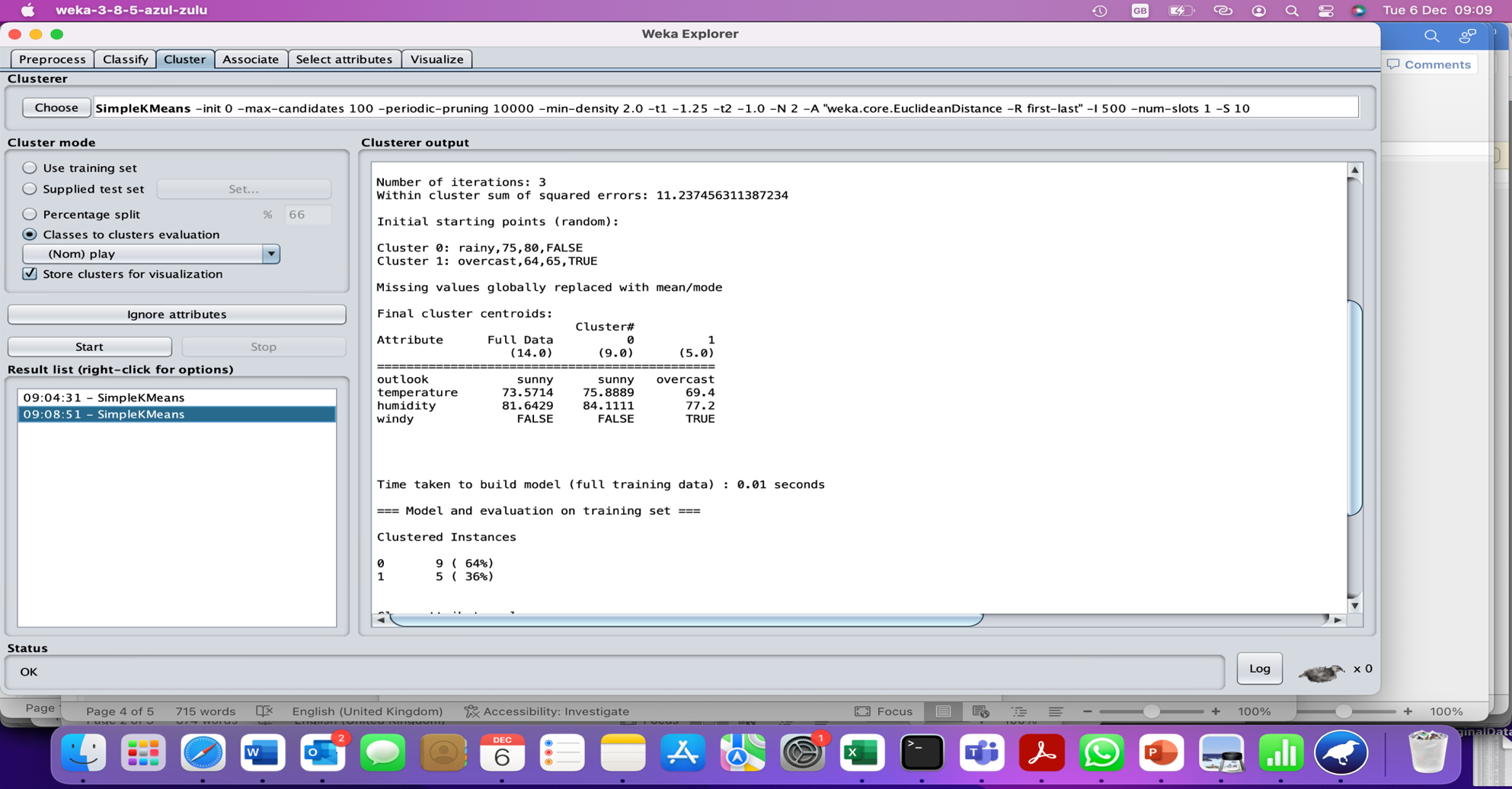
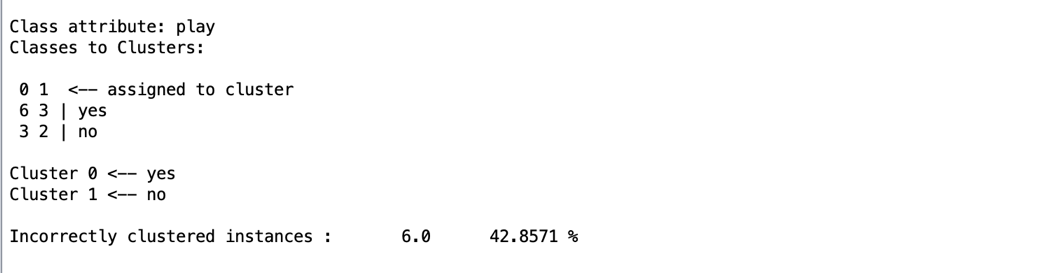
In this lab session we are going to use k-means, k-means clustering splits ***n*** observations into ***k clusters*** in which each observation belongs to the cluster with the nearest mean. The mean is used as a prototype of the cluster.

1. Clustering - Open Weka Explorer environment and load the training file using the Preprocess mode. Try first with ***weather.arff.*** Get to the Cluster mode (by clicking on the Cluster tab) and select a clustering algorithm, for example SimpleKMeans. Then click on Start and you get the clustering result in the output window. The actual clustering for this algorithm is shown as one instance for each cluster representing the cluster centroid.

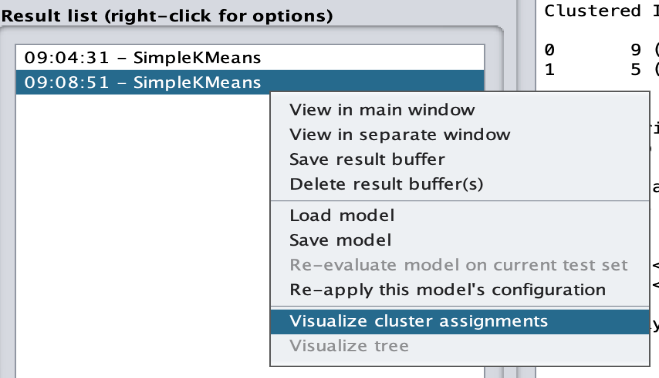


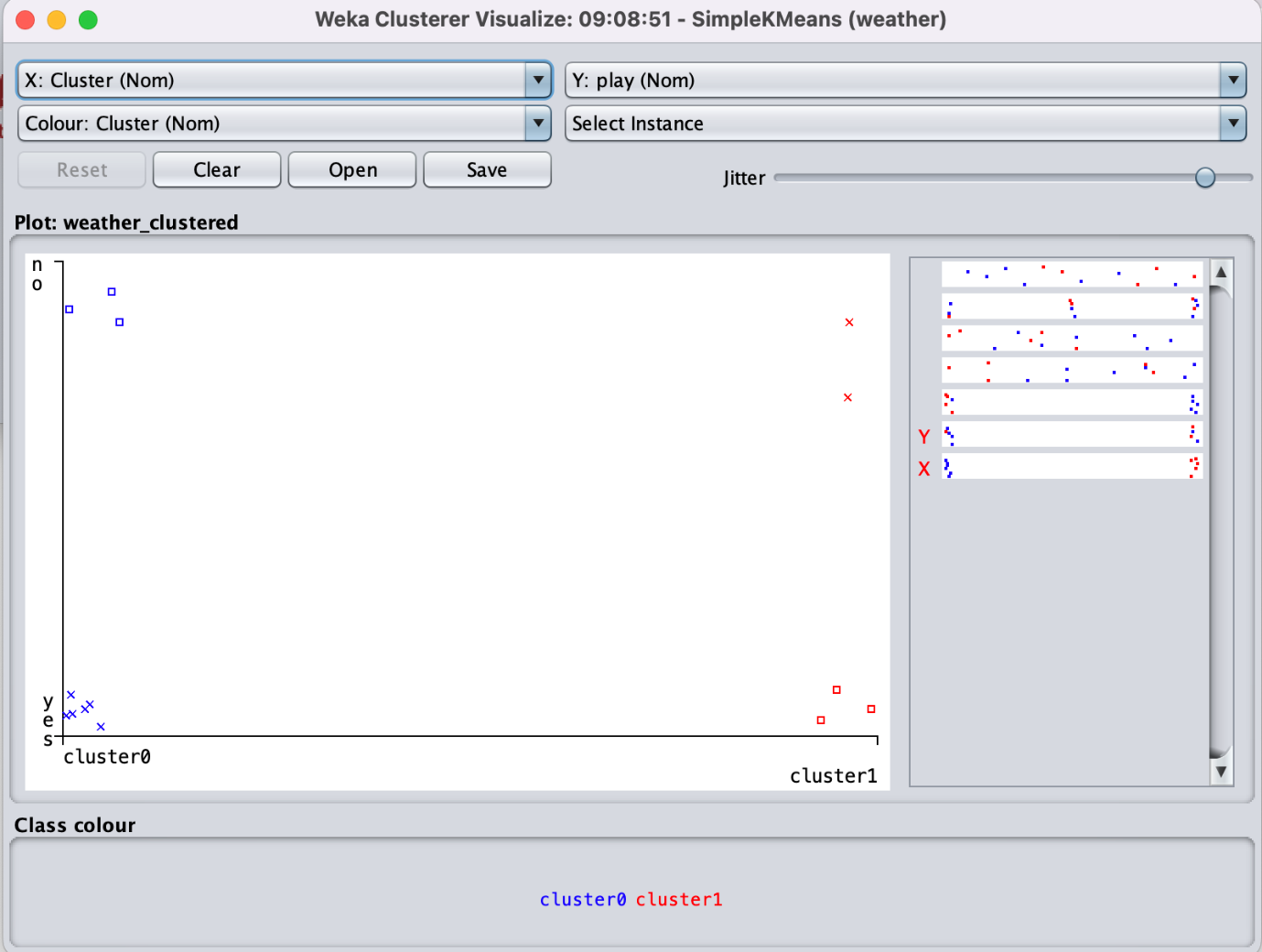
2. Evaluation - The way Weka evaluates the clustering depends on the cluster mode you select. Four different cluster modes are available (as buttons in the Cluster mode panel):

1. **Use training set (default).** After generating the clustering Weka classifies the training instances into clusters according to the cluster representation and computes the percentage of instances falling in each cluster. For example, the above clustering produced by k-means shows 64% (9 instances) in cluster 0 and 36% (5 instances) in cluster 1.
2. **In Supplied test set or Percentage split** Weka can evaluate clustering on separate test data if the cluster representation is probabilistic (e.g. for EM).
3. **Classes to clusters evaluation.** In this mode Weka first ignores the class attribute and generates the clustering. Then during the test phase it assigns classes to the clusters, based on the majority value of the class attribute within each cluster. Then it computes the classification error, based on this assignment and also shows the corresponding confusion matrix. An example of this for k-means is shown below.



3. **Visualize the cluster group.** To do this, right-click on the cluster in the **Result** list panel and select **Visualize cluster assignments**. Plot Class against Cluster. All the data points will lie on top of each other, so increase the Jitter slide bar to about halfway to add random noise to each point. This allows us to see more clearly where the bulk of the datapoints lies. In this scatter plot each row represents a class and each column a cluster.

****

****

1. You could save the clustering results by clicking Save button on the Visualization panel. The results are saved in **.arff** file. You could use Weka to open it and view the results.

**Activity 3:**

1. We are familiar with the spam dataset by now. To load the **Spam.arff** dataset (available Week-11\_unit-1) select the preprocess tab and then select Open file
2. Go to the Cluster tab: Select the ***SimpleKMeans*** cluster, bring up its options window and set numClusters to 2.
3. In the Cluster mode panel, select Classes to clusters evaluation and hit Start. This option evaluates clusters with respect to a class. More specifically, in the mode Classes to clusters evaluation Weka first ignores the class attribute and generates the clustering. Then during the test phase it assigns classes to the clusters, based on the majority value of the class attribute within each cluster. Then it computes the classification error, based on this assignment and also shows the corresponding confusion matrix.
4. Ideally, we would hope to see all instances from a single class assigned to a single cluster, and no instances from different classes assigned to the same cluster.
5. Look at the Classes to Clusters confusion matrix. Clearly, we don't have a perfect correspondence between classes and clusters.
   1. How successful has the clustering been in this regard?
   2. Looking at each class individually, can you spot the particular class that is well identified by the clustering? Classes that are poorly identified?
   3. Which classes are mostly confused with each other?
   4. Compare with the performance of the supervised classifiers we used for practical 15 or 16 Decision TREE or Bayesian Naïve Bayes Classifier .
6. Visualize the cluster assignments. To do this, right-click on the cluster in the Result list panel and select Visualize cluster assignments. Plot Class against Cluster. All the data points will lie on top of each other, so increase the Jitter slide bar to about half way to add random noise to each point. This allows us to see more clearly where the bulk of the datapoints lies. In this scatter plot each row represents a class and each column a cluster.
7. Can you draw any conclusion based on this visualization?