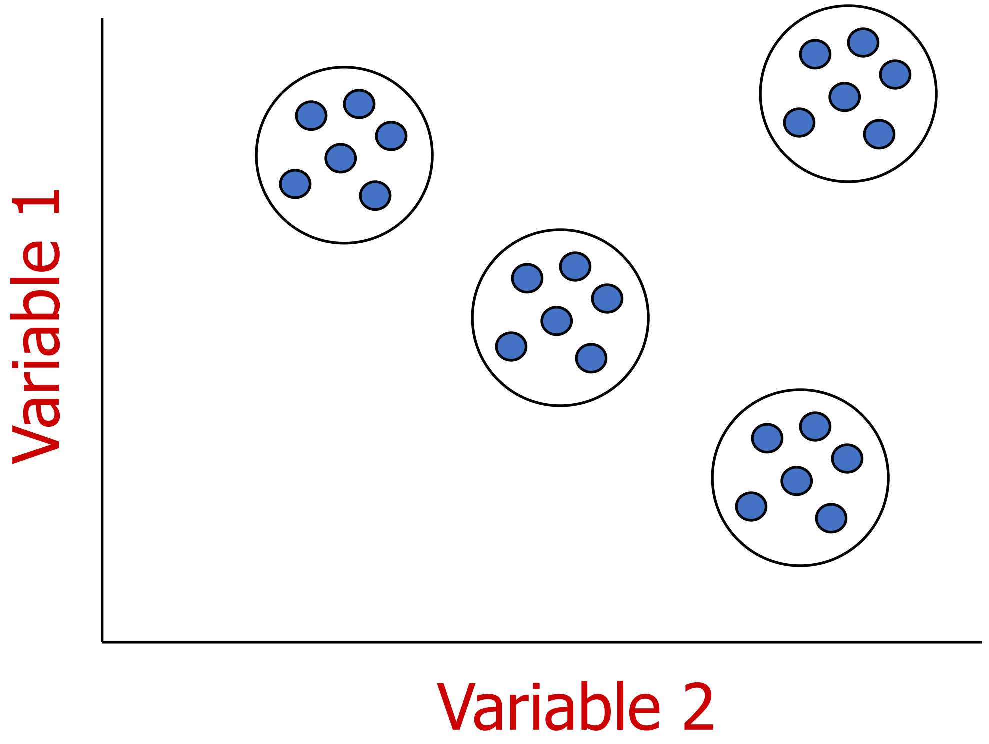
DATA MINING AND MACHINE LEARNING

Clustering Algorithms

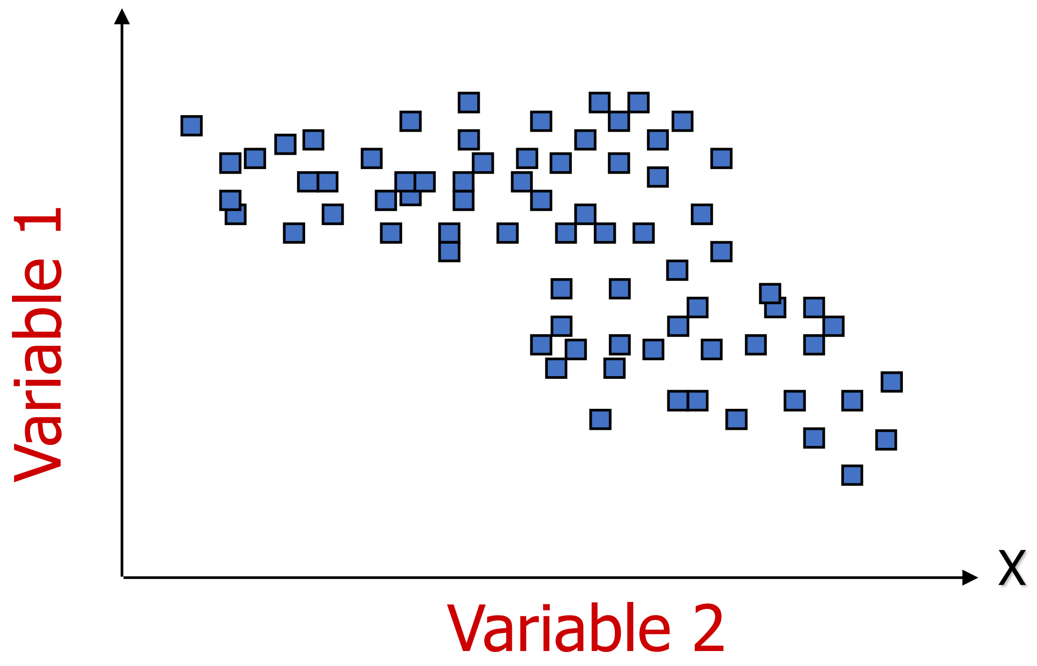
What is clustering?

- Clustering: the process of grouping a set of objects into classes of similar objects
- Most common form of *unsupervised learning*
    - Unsupervised learning = learning from raw data, as opposed to supervised data where a classification of examples is given

An Ideal Clustering Situation



Common Clustering Situation



Clustering considerations

- What does it mean for objects to be similar?
- What algorithm and approach do we take?
  - *Top-down: k-means*
  - *Bottom-up: hierarchical agglomerative clustering*
- Do we need a hierarchical arrangement of clusters?
- How many clusters?
- Can we label or name the clusters?
- How do we make it efficient and scalable?

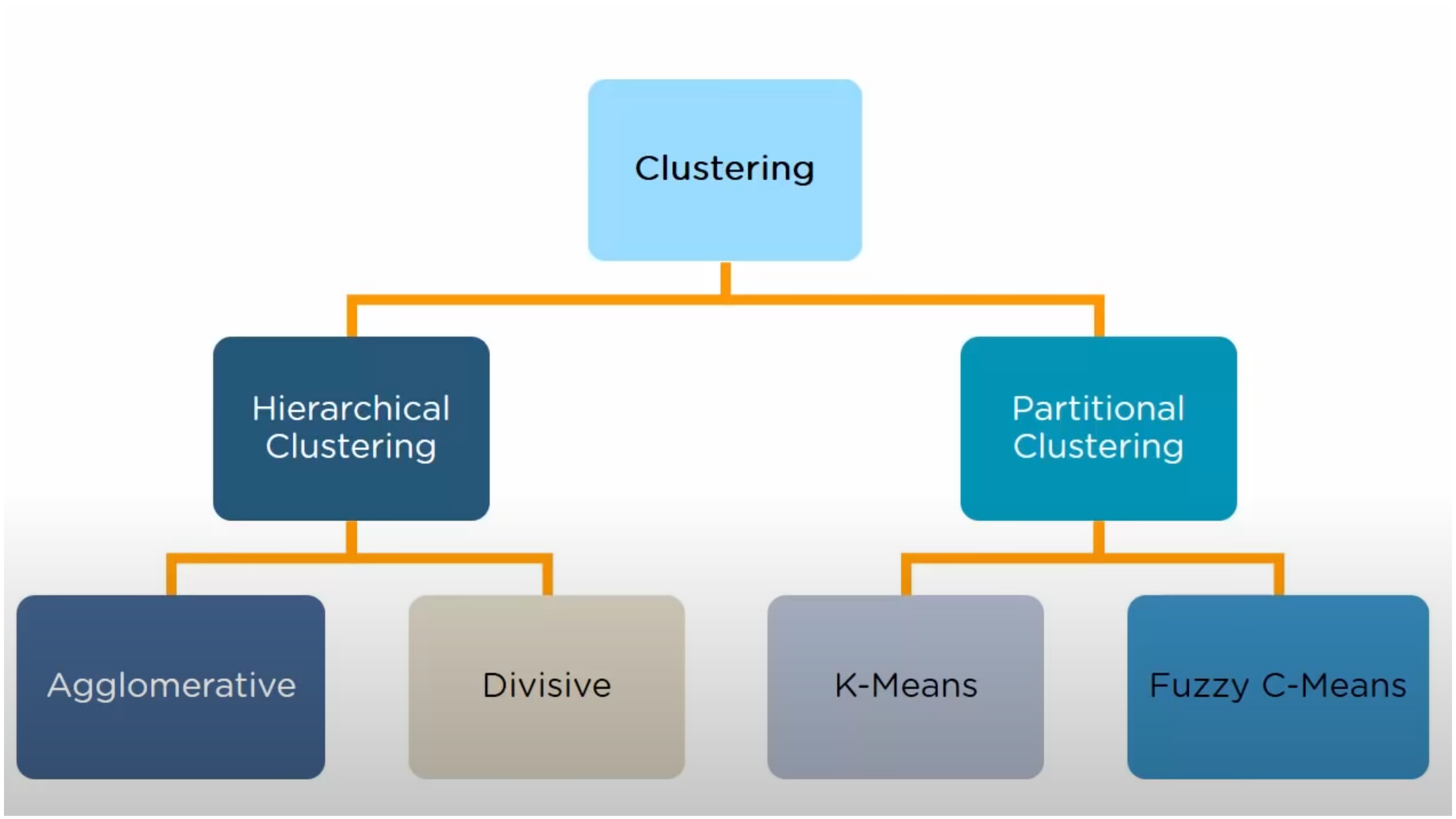
What makes docs “related”?

- Ideal: semantic similarity.
- Practical: statistical similarity
  - *Treat documents as vectors*
  - *For many algorithms, easier to think in terms of a distance (rather than <u>similarity</u>) between docs.*
  - *Think of either cosine similarity or Euclidean distance*

Clustering Algorithms

- Partitional algorithms
  - *Usually start with a random (partial) partitioning*
  - *Refine it iteratively*
    - *K* means clustering
    - Model based clustering
- Hierarchical algorithms
  - *Bottom-up, agglomerative*
  - *Top-down, divisive*

Types of Clustering



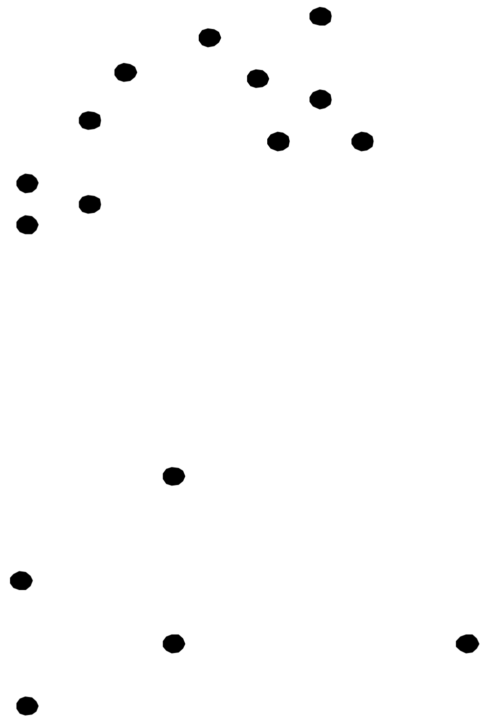
Types of Clustering

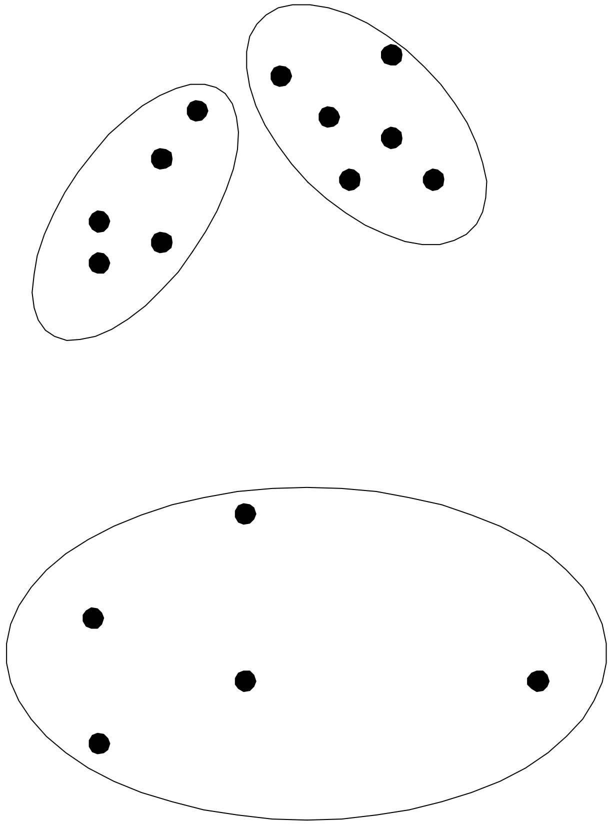
- A clustering is a set of clusters
- Important distinction between hierarchical and partitional sets of clusters
- Partitional Clustering
  - *A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset*
- Hierarchical clustering
  - *A set of nested clusters organized as a hierarchical tree*

Partitional Clustering

Partitional Clustering

A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset


Original Points

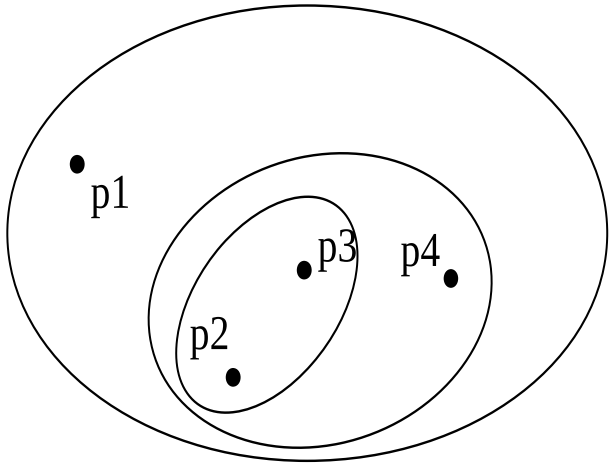

A Partitional Clustering

Clustering Approaches

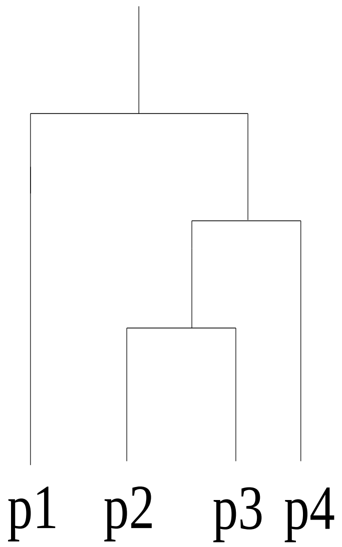
- Bottom up: at each step join the two closest clusters (starting with single-instance clusters)
  - *Design decision: distance between clusters*
    - E.g. two closest instances in clusters vs. distance between means
- Top down: find two clusters and then proceed recursively for the two subsets
  - *Can be very fast*
- Both methods produce a dendrogram (tree structure)

Hierarchical Clustering

A set of nested clusters organized as a hierarchical tree



Traditional Hierarchical Clustering



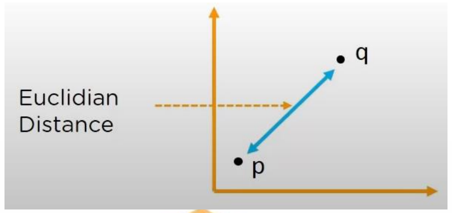
Traditional Dendrogram

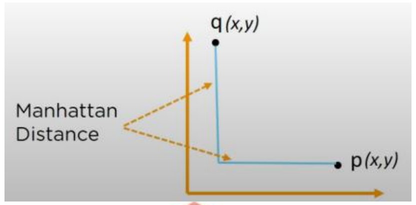
K-means Clustering

- Partitional clustering approach
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K, must be specified
- The basic algorithm is very simple

```
1: Select $K$ points as the initial centroids.
2: repeat
3:     Form $K$ clusters by assigning all points to the closest centroid.
4:     Recompute the centroid of each cluster.
5: until The centroids don't change
```

Distance Measure





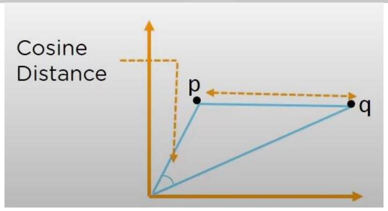
Euclidean distance measure

Manhattan distance measure

Distance measure will determine the similarity between two elements and it will influence the shape of the clusters

Squared Euclidean distance measure

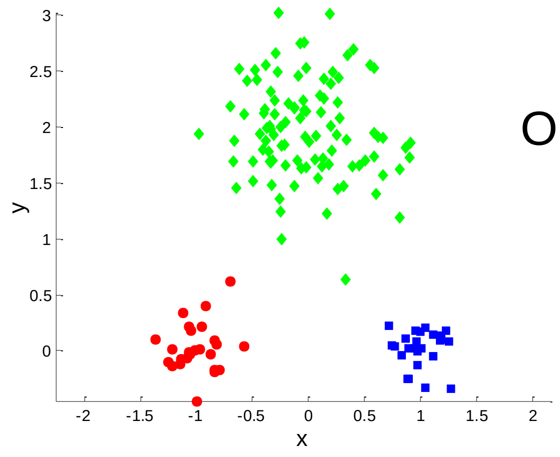
Cosine distance measure



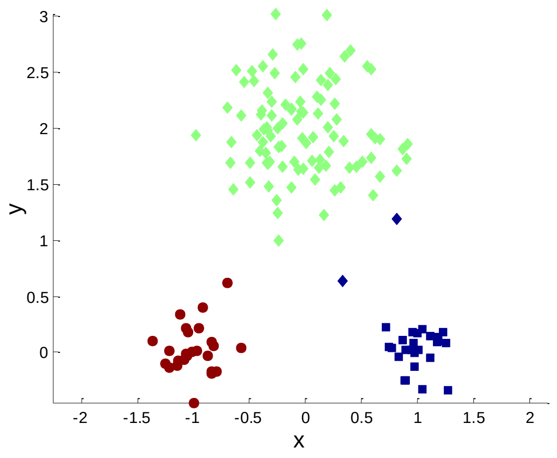
K-means Clustering – Details

- Initial centroids are often chosen randomly.
  - *Clusters produced vary from one run to another.*
- The centroid is (typically) the mean of the points in the cluster.
- ‘Closeness’ is measured by Euclidean distance
- Most of the convergence happens in the first few iterations.
  - *Often the stopping condition is changed to ‘Until relatively few points change clusters’*
- Complexity is O(n * K * I * d)
  - *n = number of points, K = number of clusters,*
    I = number of iterations, d = number of attributes

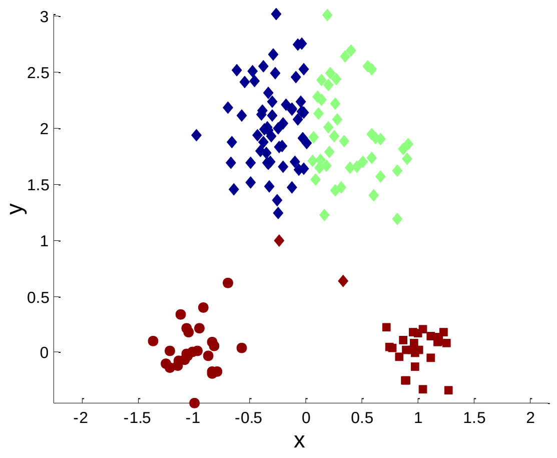
Two different K-means Clusterings



Original Points

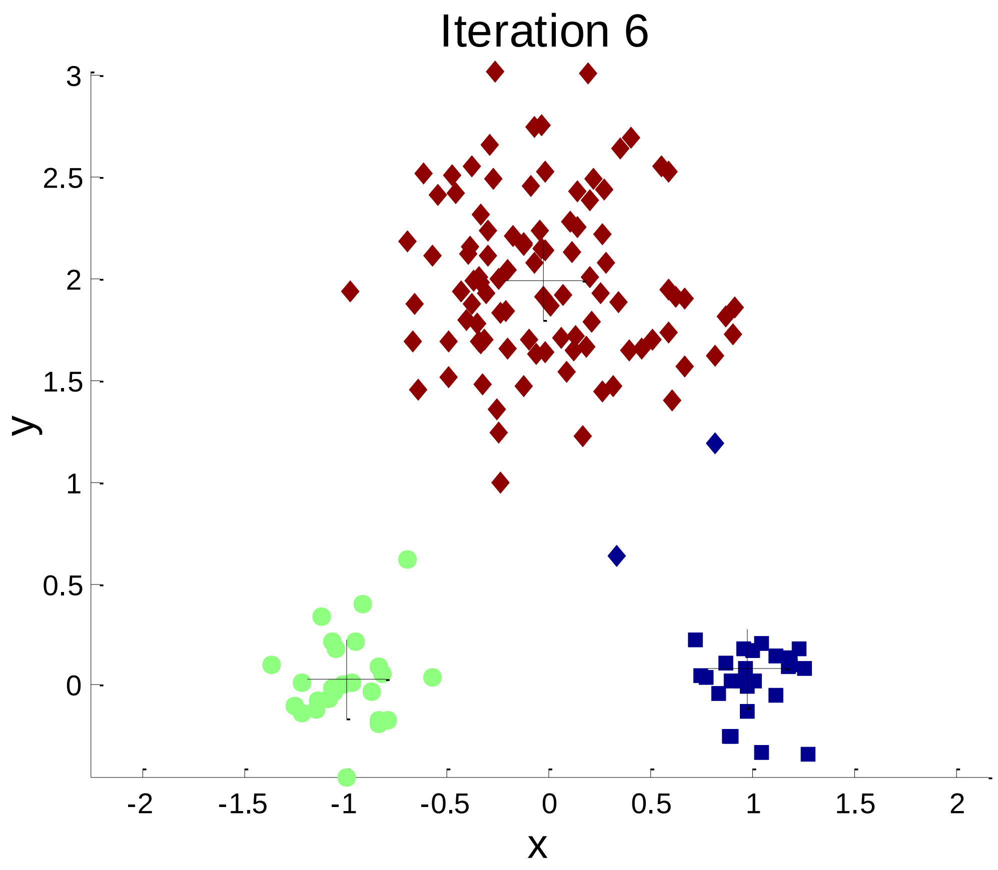


Optimal Clustering

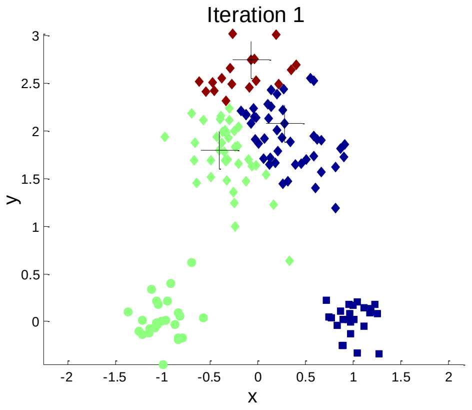


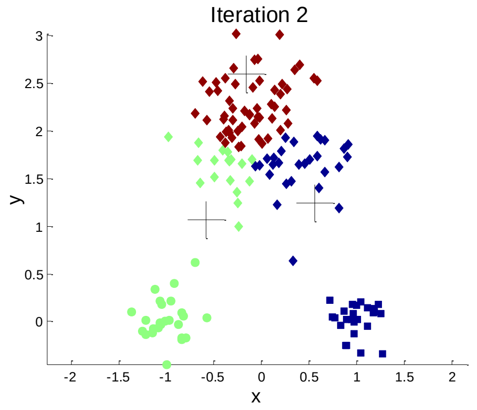
Sub-optimal Clustering

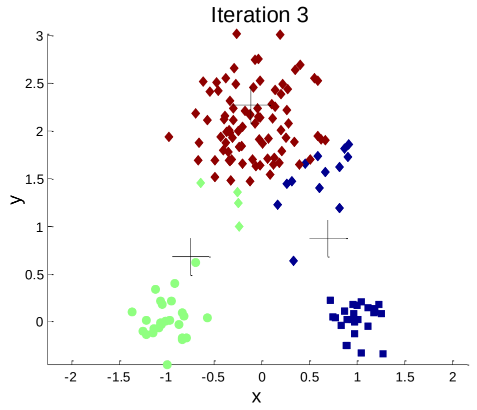
Importance of Choosing Initial Centroids

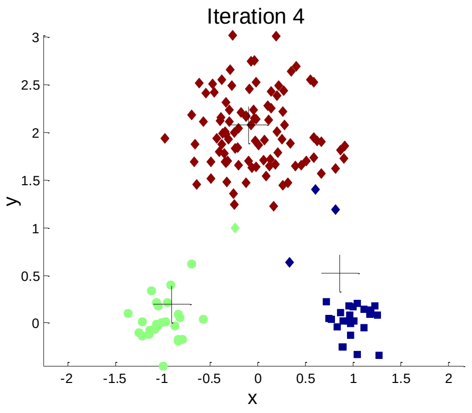


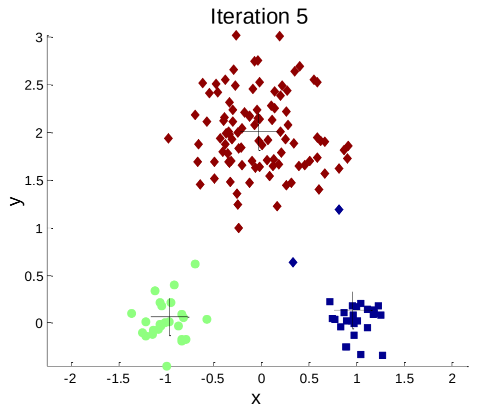
Importance of Choosing Initial Centroids

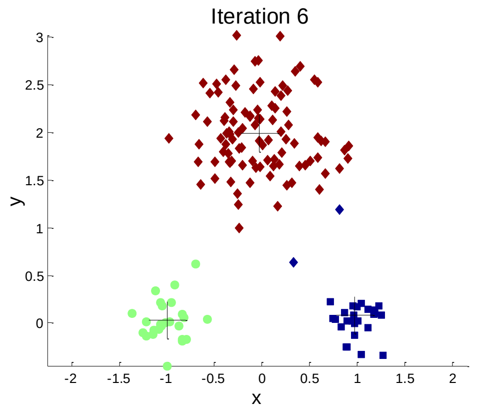












K Means in Python

- K means is supported in Python through sklearn.cluster.Kmeans

class sklearn.cluster.**KMeans**(*n_clusters=8, init='k-means++', n_init=10, max_iter=300, tol=0.0001, precompute_distances='auto', verbose=0, random_state=None, copy_x=True, n_jobs=None, algorithm='auto'*)

- Major parameter to tune is n_clusters and n_jobs (for large datasets)

Kmeans ++

- Python makes use of an enhanced version of Kmeans

1. initialize an empty set S
2. randomly choose the centroid c_1 of the first cluster from the input samples
3. assign c_1 to S
4. for each sample I not in S, find the distance d(I,M) to any of the centroids M in S
5. randomly select the next centroid c_n by the weighted probability:

$$\frac{d(c_n, S)}{\sum_I d(I,S)}$$

6. Repeat steps 2 and 3 until k centroids are chosen
7. Now run classic KMeans

Strengths and Limitations of K Means

The major strength is its very fast speed as it is linear both in terms of number of samples as well as number of data dimensions

One limitation is that the number of clusters (K) needs to be supplied in advance

In practice, I have found that specifying K in advance is not a major limitation provided that the K parameter is tuned (e.g. using GridSearchCV)

A more serious limitation of K means is its inability to deal with non spherical clusters .

Evaluating K-means Clusters

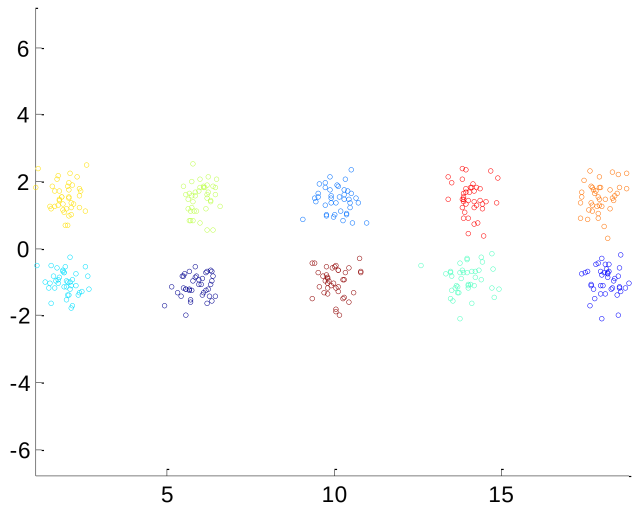
- Most common measure is Sum of Squared Error (SSE)
  - *For each point, the error is the distance to the nearest cluster*
  - *To get SSE, we square these errors and sum them.*

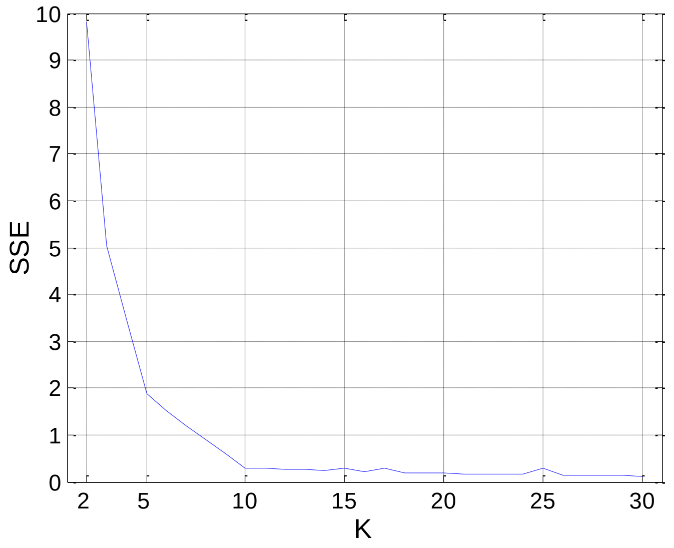
$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

  - *x is a data point in cluster C_i and m_i is the representative point for cluster C_i*
    - can show that m_i corresponds to the center (mean) of the cluster
  - *Given two cluster configurations, we choose the one with the smallest error*
  - *One easy way to reduce SSE is to increase K, the number of clusters*
    - A good clustering with smaller K can have a lower SSE than a poor clustering with higher K

Internal Measures: SSE

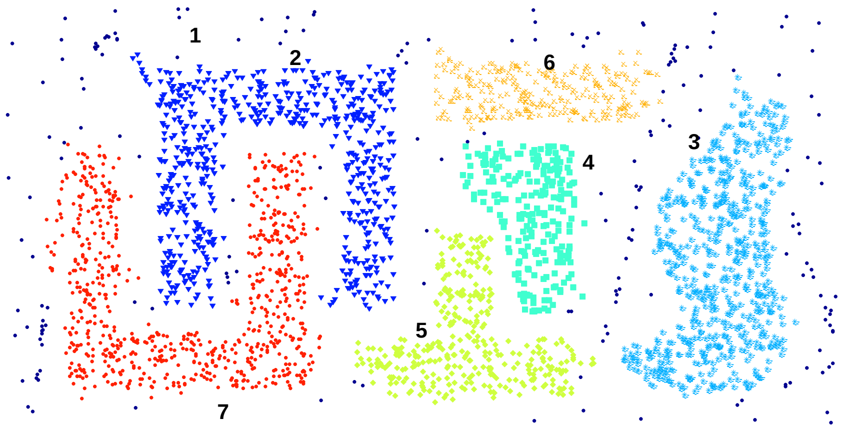
- SSE can be used to estimate the number of clusters
- While this can be effective for data that does not have a complex structure (example below) it does not perform well for datasets that have high a degree of overlap.

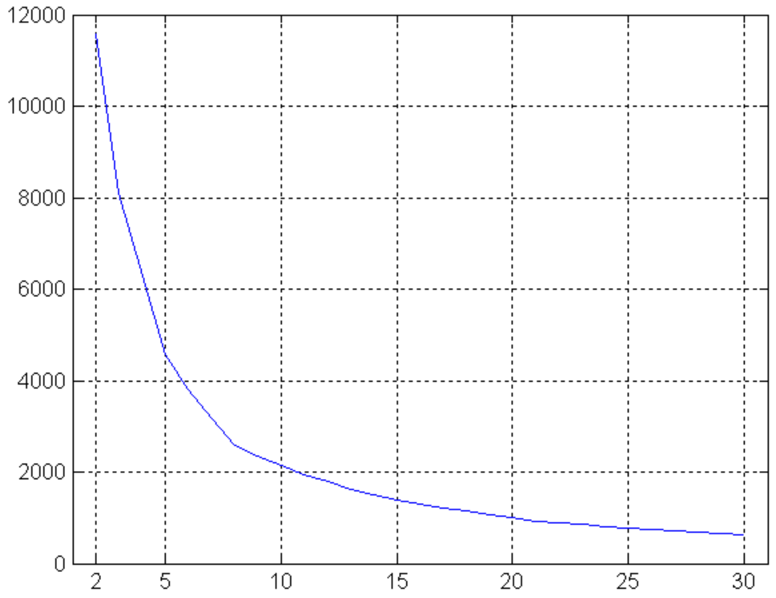




Internal Measures: SSE

- SSE curve for a more complicated data set





SSE of clusters found using K-means

Cluster Silhouette Measure

- The cluster silhouette measure (s) is robust to data overlap between clusters
- S consists of two components

  1. The fit of a data sample i within its own cluster f(i)
  2. The fit of sample i to its neighbouring cluster n(i)

- We first compute f(i), the distance of sample i to its samples within *its* cluster

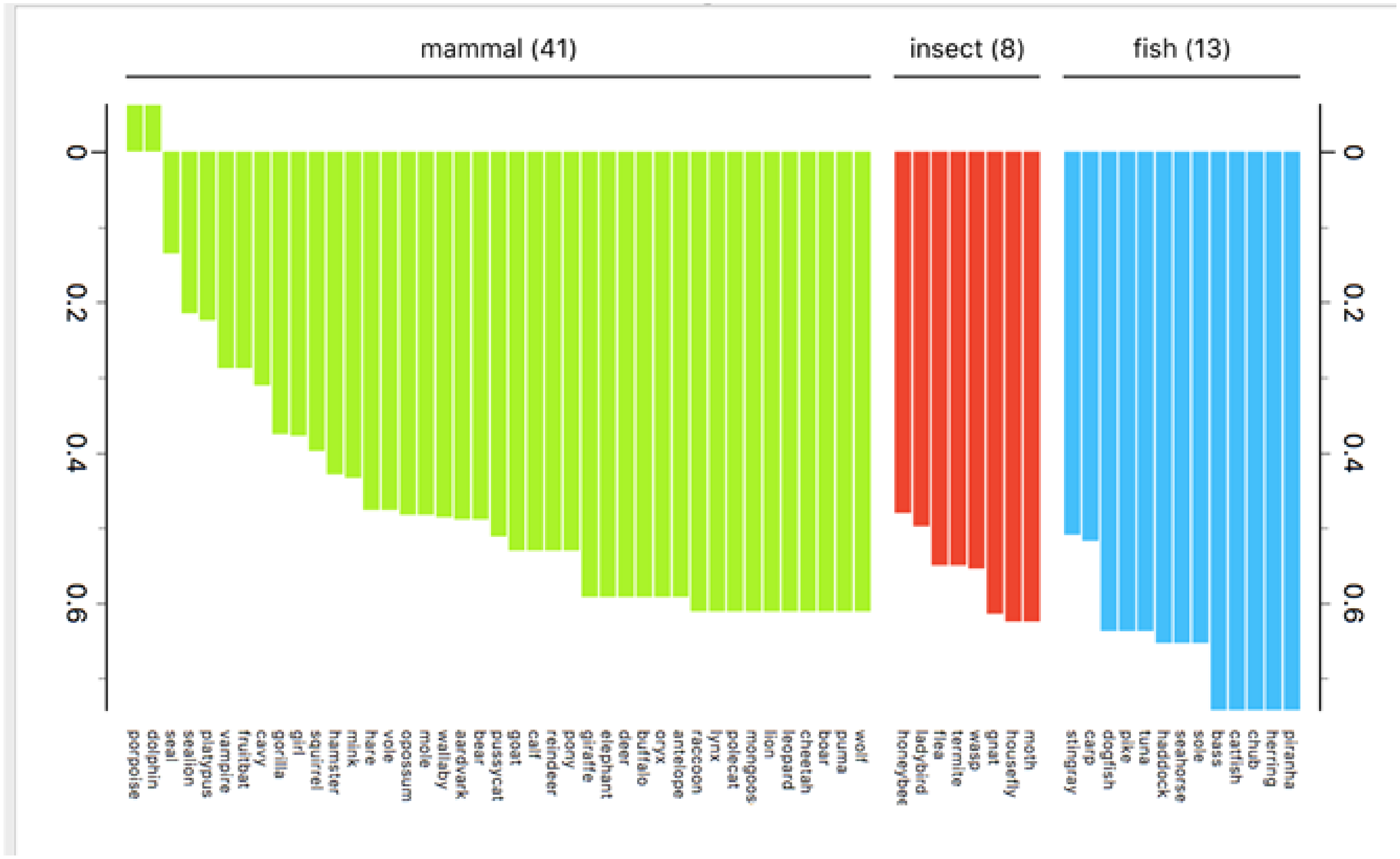
$f(i) = \frac{1}{|C_i|-1} \sum_{j \in C_{i;i \neq j}} d(i,j)$ where $d(i,j)$ is the distance between samples i and j within the cluster

Cluster Silhouette Measure

- Next, we determine the distance n(i) of the sample to its nearest cluster.
- $n(i) = min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j)$
- Now $s(i) = \frac{n(i) - f(i)}{\max(f(i), n(i))}$
- It is easy to see that $-1 \leq s(i) \leq 1$
- The overall cluster silhouette is then the average of $s(i)$ over all samples
- The higher the value of s the better is the quality of the cluster configuration.

Evaluation using Silhouette Measure

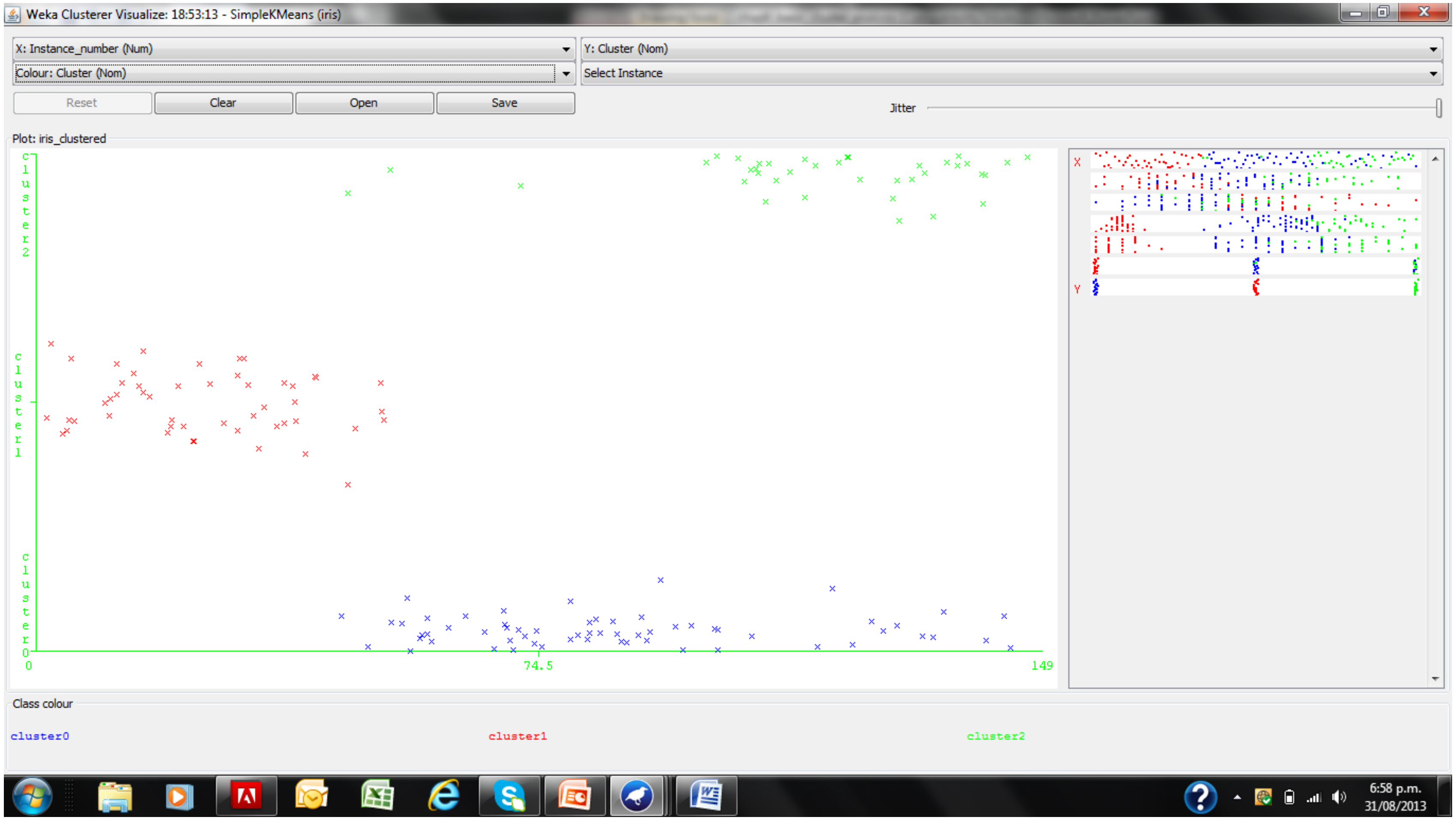
- The Zoo dataset is clustered into three broad categories of animals, namely *mammals*, *fish* and *insects*.



Visualizing Clusters

- We use the Iris dataset and cluster using k means with 3 values of k: 3,4,5
- First run k means with k=3. We instruct k means to ignore the class attribute and start the clustering process

Visualizing k means with k=3


Weka Clusterer Visualize: 18:53:13 - SimpleKMeans (iris)

X: Instance_number (Num)

Y: Cluster (Nom)

Colour: Cluster (Nom)

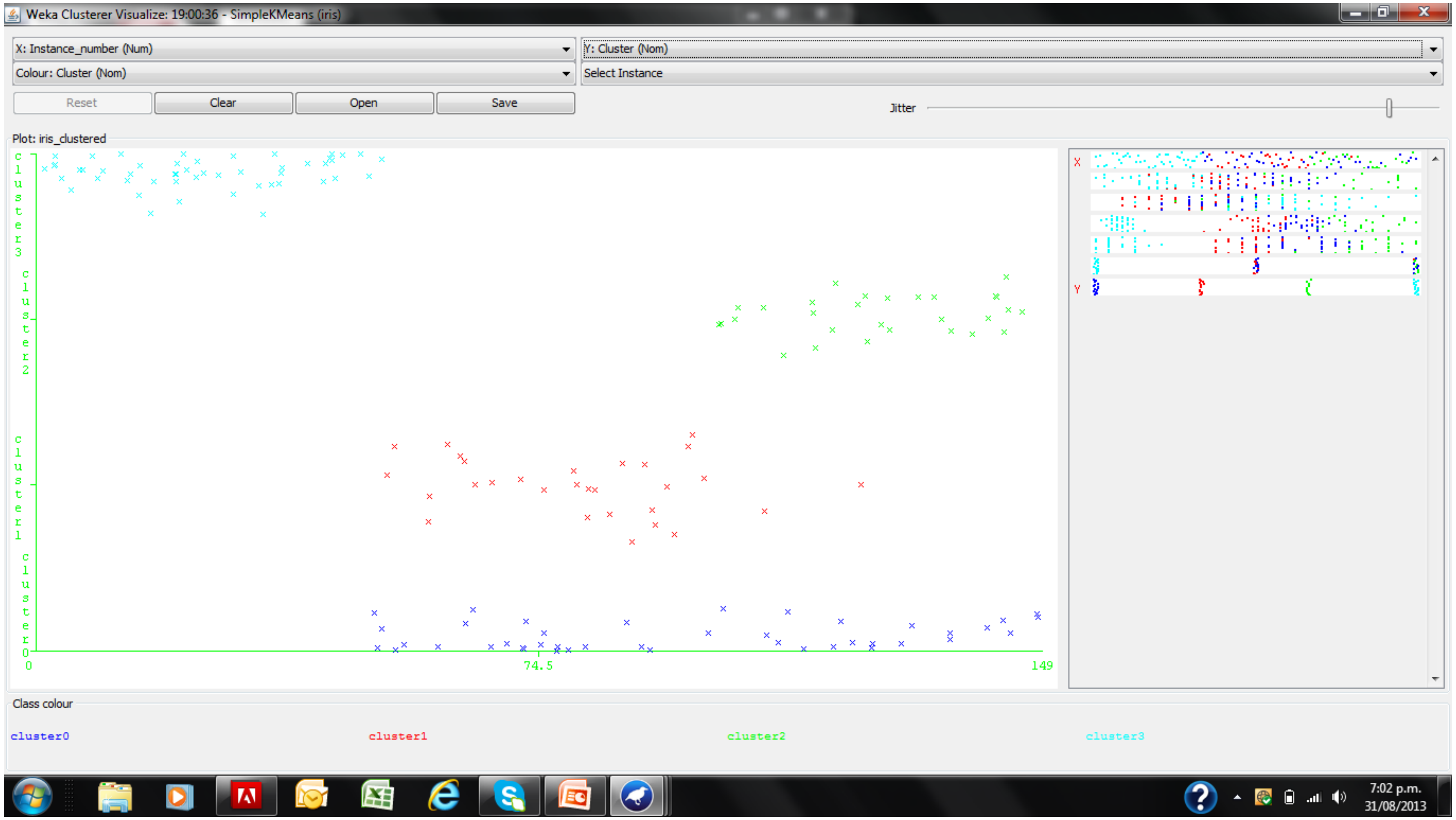
Select Instance

Plot: iris_clustered

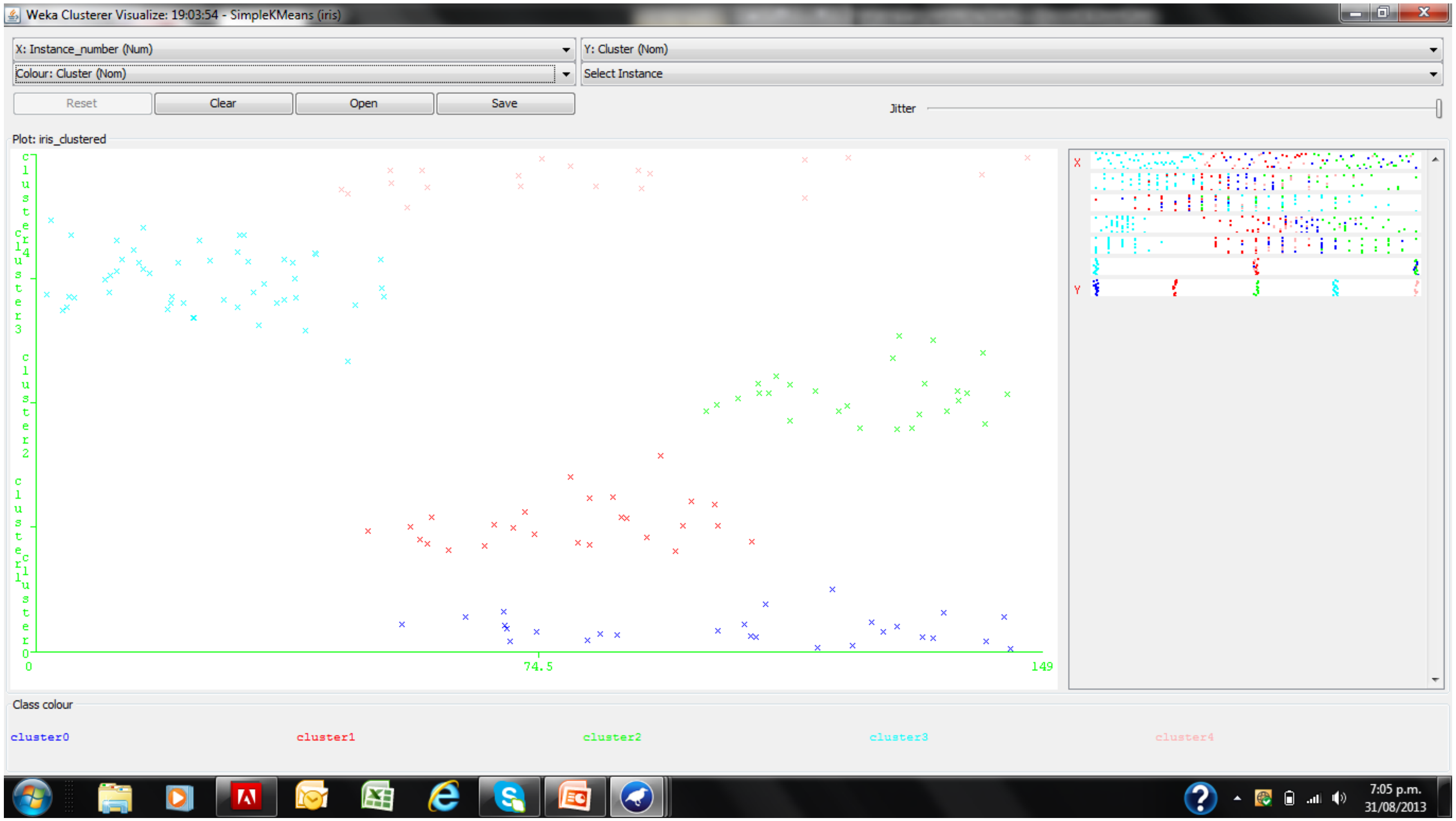
Class colour

cluster0 cluster1 cluster2

Visualizing k means with k=4



Visualizing k means with k=5



Weka Clusterer Visualize: 19:03:54 - SimpleKMeans (iris)

X: Instance_number (Num)

Y: Cluster (Nom)

Colour: Cluster (Nom)

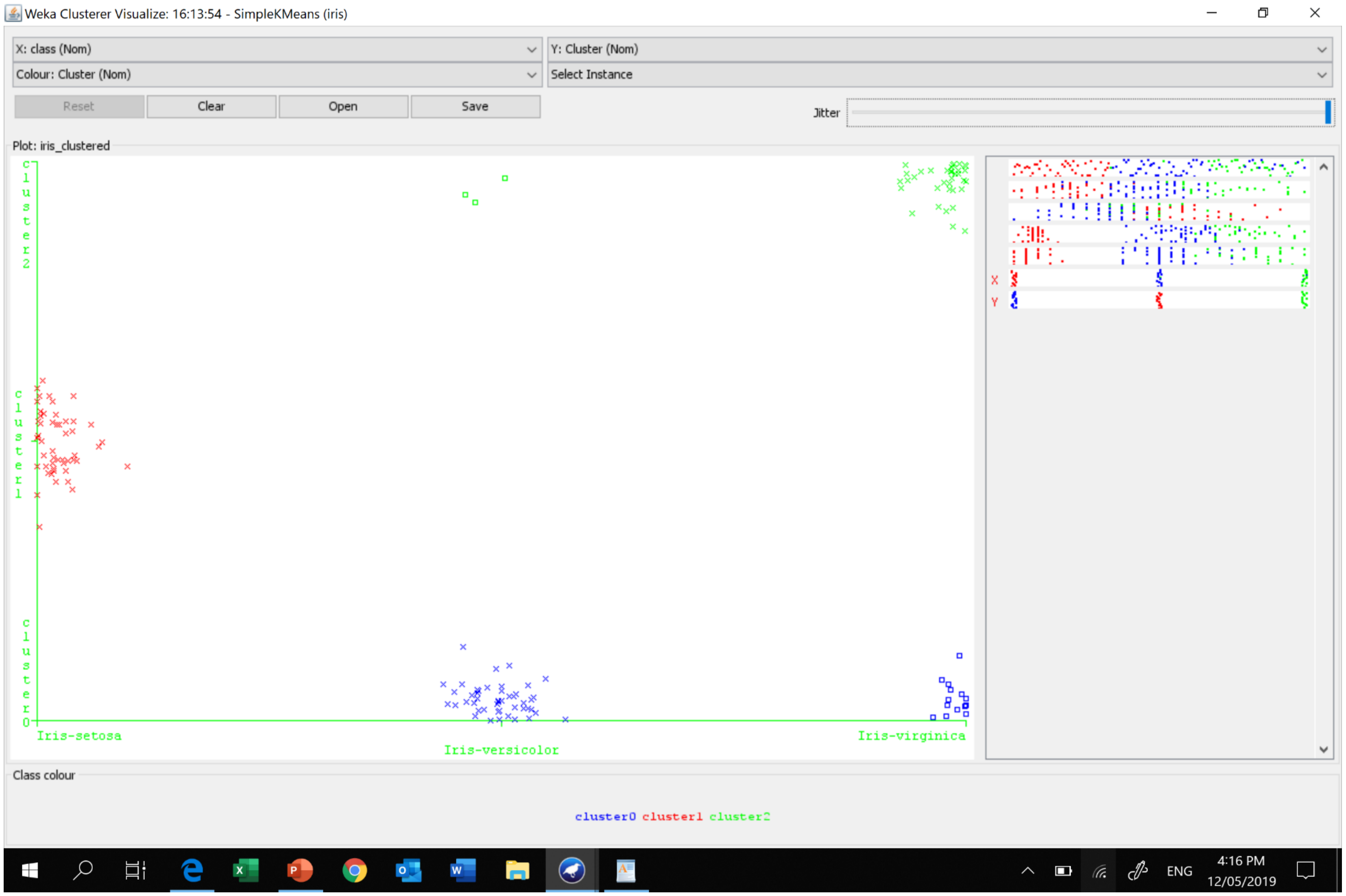
Select Instance

Plot: iris_clustered

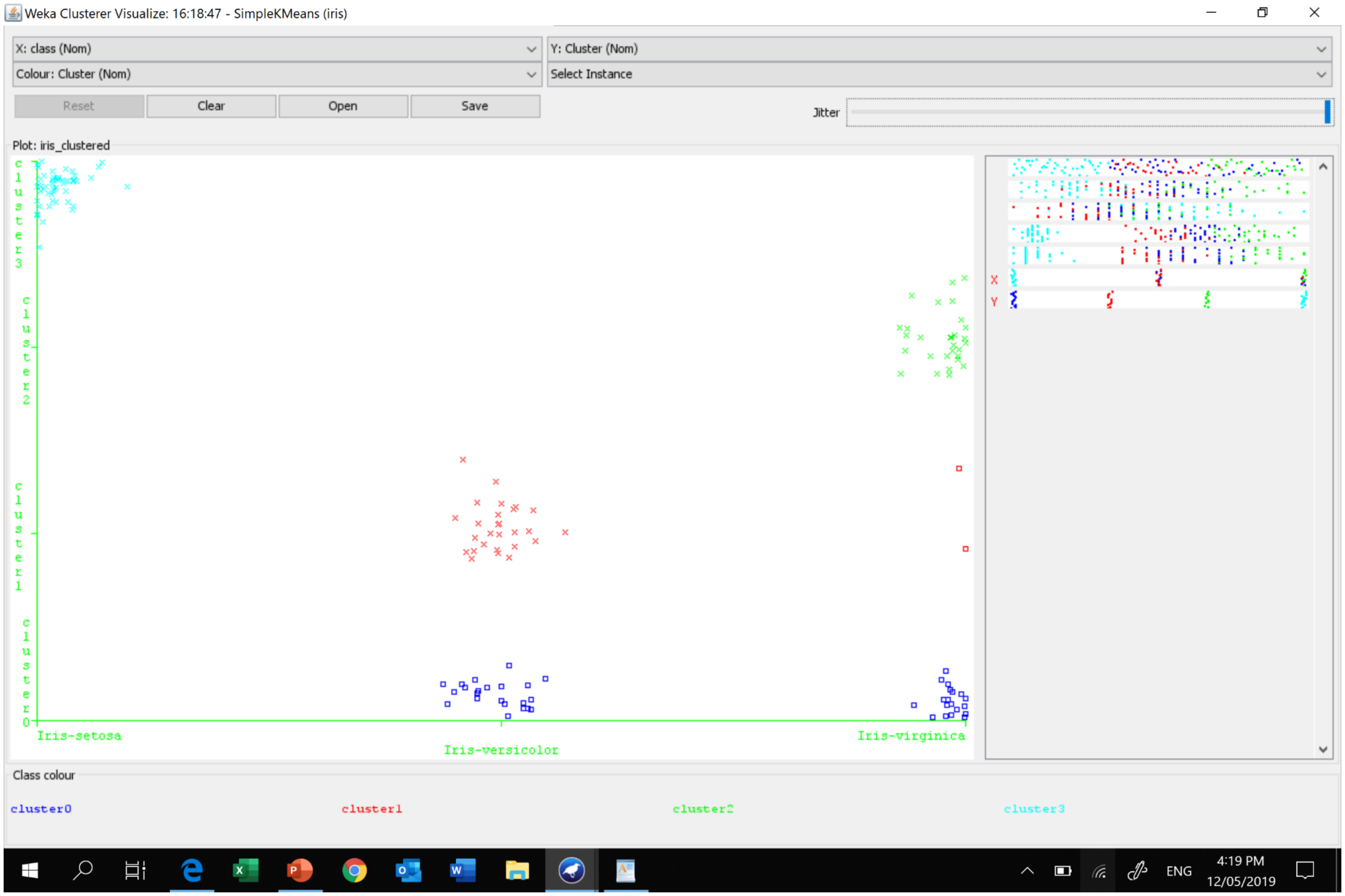
Cluster to Class Evaluation (k=3)

- When class labels are available a cluster to class evaluation can be done
- Weka supports this as ‘cluster mode option” in the opening screen for k-means
- Ensure that the class labels are ignored in the clustering process – the labels are only used for evaluation, and does not play a role in the clustering process

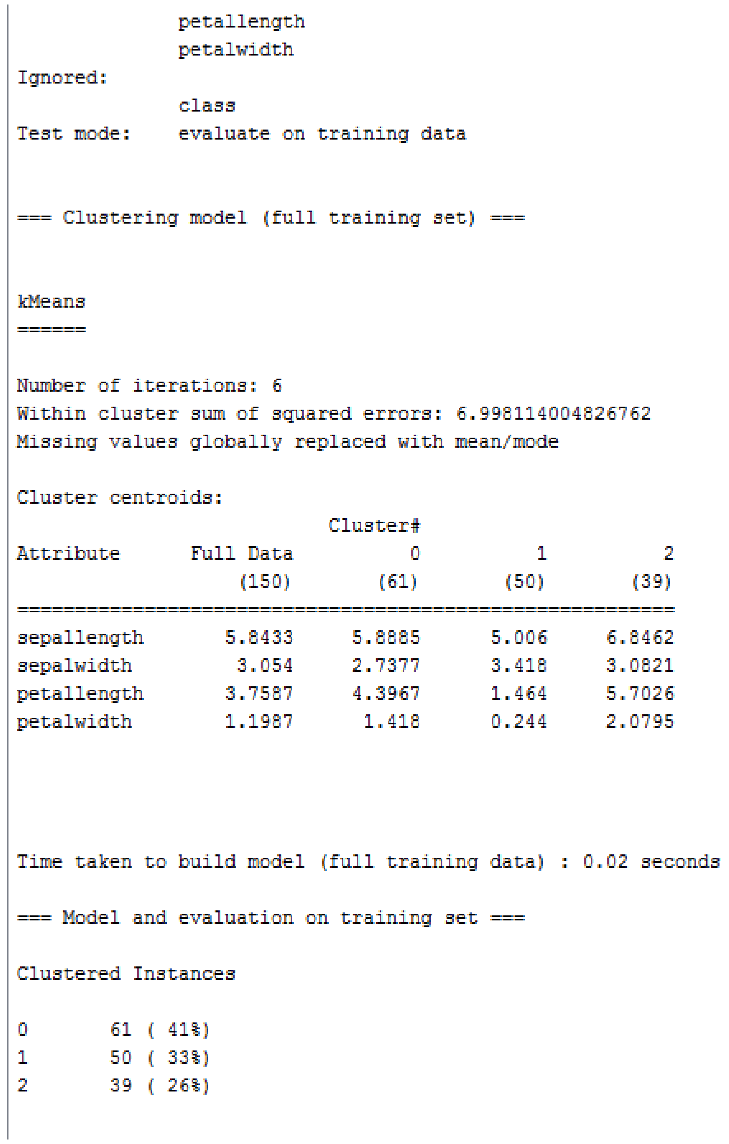
Cluster to Class Evaluation (k=3)



Cluster to Class Evaluation (k=4)

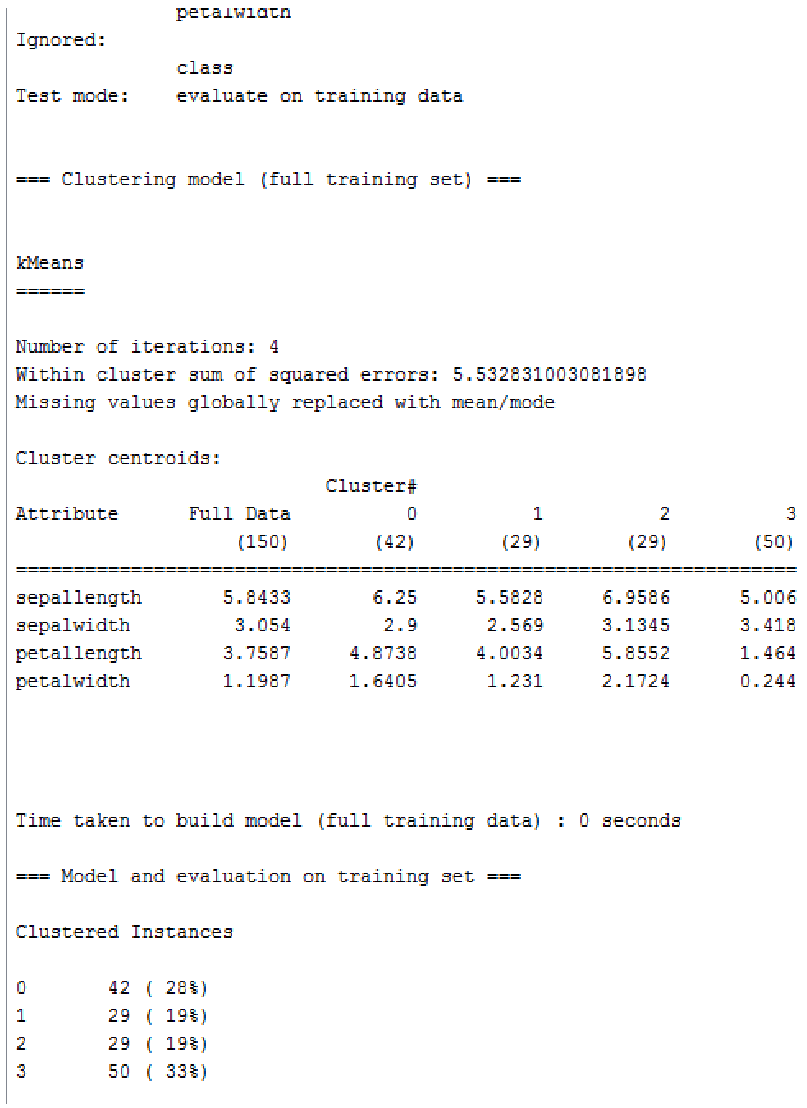


Evaluating K means through SSE (k=3)



```
              petallength
              petalwidth
Ignored:
              class
Test mode:    evaluate on training data


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


kMeans
======

Number of iterations: 6
Within cluster sum of squared errors: 6.998114004826762
Missing values globally replaced with mean/mode

Cluster centroids:
                           Cluster#
Attribute      Full Data          0          1          2
                   (150)       (61)       (50)       (39)
=========================================================
sepallength       5.8433     5.8885      5.006     6.8462
sepalwidth         3.054     2.7377      3.418     3.0821
petallength       3.7587     4.3967      1.464     5.7026
petalwidth        1.1987      1.418      0.244     2.0795




Time taken to build model (full training data) : 0.02 seconds

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

Clustered Instances

0       61 ( 41%)
1       50 ( 33%)
2       39 ( 26%)
```

Evaluating K means through SSE (k=4)



```
                 petalwidth
Ignored:
                 class
Test mode:       evaluate on training data


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


kMeans
======

Number of iterations: 4
Within cluster sum of squared errors: 5.532831003081898
Missing values globally replaced with mean/mode

Cluster centroids:
                           Cluster#
Attribute      Full Data          0          1          2          3
                   (150)       (42)       (29)       (29)       (50)
====================================================================
sepallength       5.8433       6.25     5.5828     6.9586      5.006
sepalwidth         3.054        2.9      2.569     3.1345      3.418
petallength       3.7587     4.8738     4.0034     5.8552      1.464
petalwidth        1.1987     1.6405      1.231     2.1724      0.244




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

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

Clustered Instances

0       42 ( 28%)
1       29 ( 19%)
2       29 ( 19%)
3       50 ( 33%)
```