Technische Hochschule Rosenheim

Modul - Unsupervised and Reinforcement Learning (URL)

Bachelor Programme AAI

01 - Clustering

Prof. Dr. Marcel Tilly

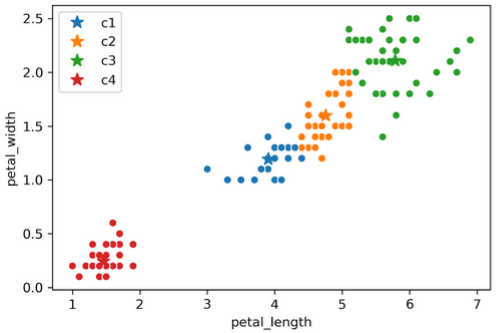
Faculty of Computer Science, Cloud Computing

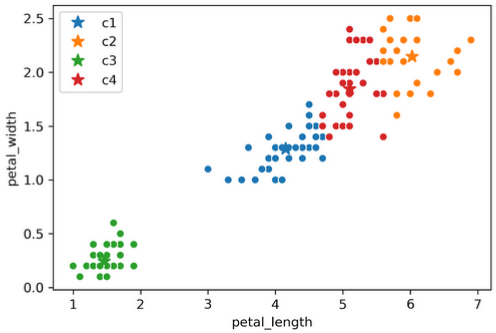
Agenda

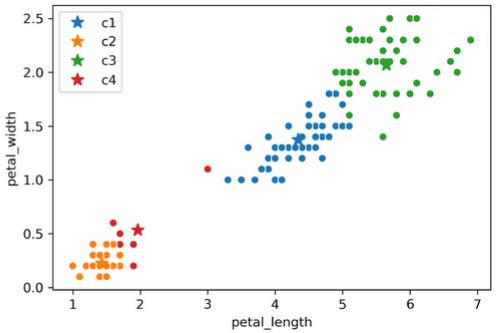
- k-means revisited
- Elbow method and Silhoutte score
- Hierarchical Clustering

K-Means Clustering for K = 4

- Each time you run K-Means, you get a different output.





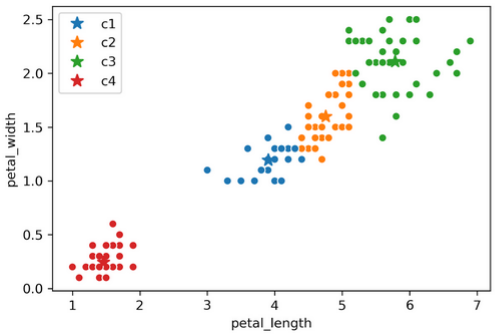


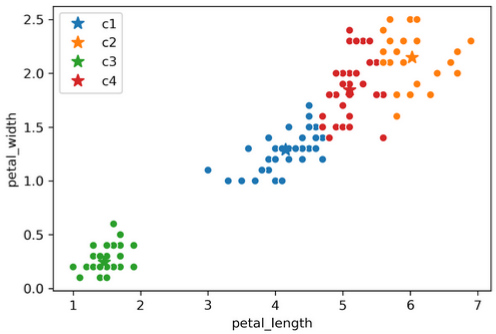
- Which is best?

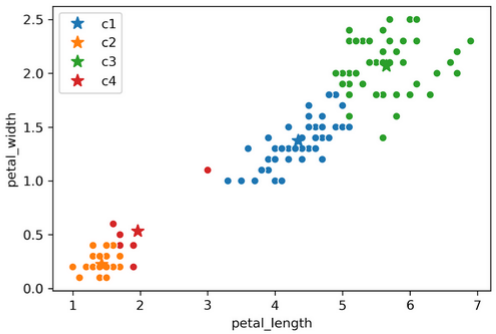
> One approach: Define some sort of loss function.

K-Means Clustering for K = 4

- Each time you run K-Means, you get a different output. Can define a loss to decide which is best.





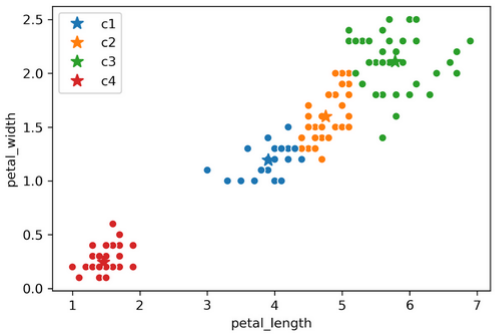


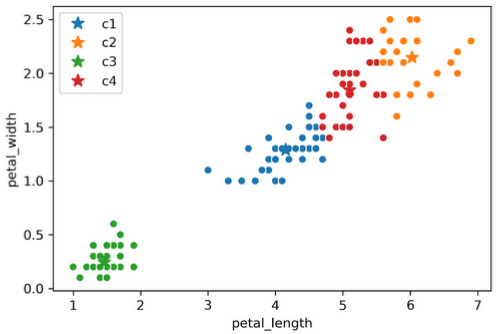
- Come up with a loss function for clustering.

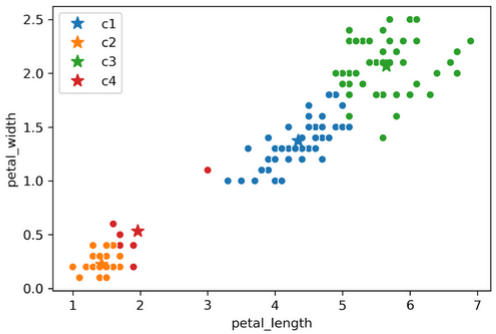
Ideas?

K-Means Clustering for K = 4

- Each time you run K-Means, you get a different output. Can define a loss to decide which is best.







- Come up with a loss function for clustering.

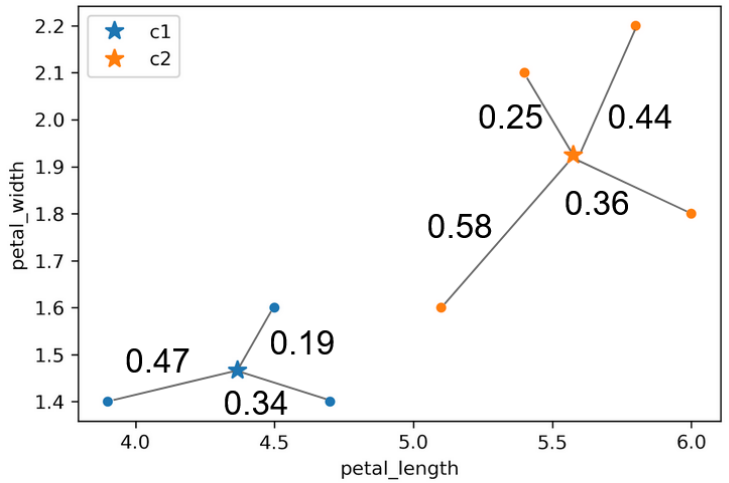
Ideas?

- The sum of distances from each point to its center.
- Could take into account balance of number of points per cluster.

K-Means Clustering for K = 4

- To evaluate different clustering results, we need a loss function.

- Two common loss functions:
  - **Inertia**: Sum of squared distances from each data point to its center.
  - **Distortion**: Weighted sum of squared distances from each data point to its center.



- Example:
  - **Inertia**:
    $0.47^2 + 0.19^2 + 0.34^2 + 0.25^2 + 0.58^2 + 0.36^2 + 0.44^2$
  - **Distortion**:
    $(0.47^2 + 0.19^2 + 0.34^2)/3 + (0.25^2 + 0.58^2 + 0.36^2 + 0.44^2)/4$

Optimizing Distortion

Technische Hochschule Rosenheim

Give an algorithm that optimizes distortion!

- The algorithm should return the EXACT best centers and clusters.
- Don't worry about runtime.

Algorithm

- For all possible k_n clusters:
  - Compute the k centers for that clusters
  - Compute the distortion/inertia for the k centers
  - If current distortion is better than best known, write down the current centers and cluster and call that the new best known.

Code: Distortion_and_Intertia.ipynb

Elbow method

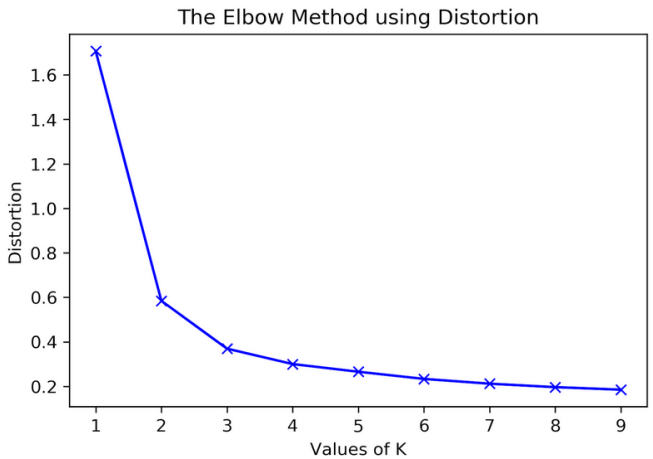
Picking k: Elbow Method

Technische Hochschule Rosenheim

-The **Elbow method** is the oldest method to distinguish the potential optimal cluster number k

- For K-Means, one approach is to plot distortion vs. many different k values.
- Pick the K in the elbow, where we get diminishing returns afterwards.

> Note: Big complicated data often lacks an elbow.



Picking k: Elbow Method

Technische Hochschule Rosenheim

Basic idea:

1. specify k = 2 as the initial optimal cluster number k and then increase k by step 1 to the maximal specified k
2. find centeroids for given k
3. calculate the cost C as the mean distortion
    - sum of the squared Euclidean distances (SSE) divided by N (number of data points) per each cluster
4. increase k and repeat 2.- 4.

- The optimal cluster number k is the point from where on the cost C remains almost unchanged
- This point is called *cost peak value*

Code: Elbow_and_Silhouette.ipynb

Silhouette score

Silhouette Score

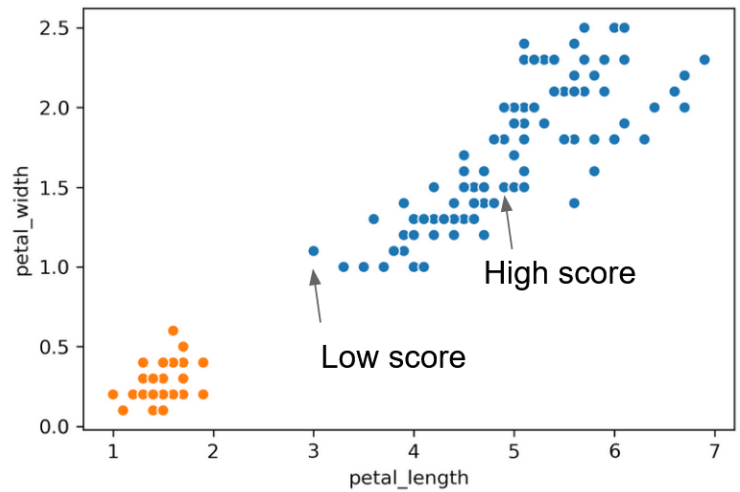
To evaluate how "well clustered" a specific data point is, we can use the "silhouette score", a.k.a. The "silhouette width".

- **High score**: Near the other points in its X's cluster.
- **Low score**: Far from the other points in its cluster.

For a data point X the *Silhouette score* S is:

- A = average distance to other points in cluster.
- B = average distance to points in closest cluster.

$$S = \frac{B - A}{max(A, B)}$$



Silhouette Score

Technische Hochschule Rosenheim

For a data point X, Silhouette score S is:

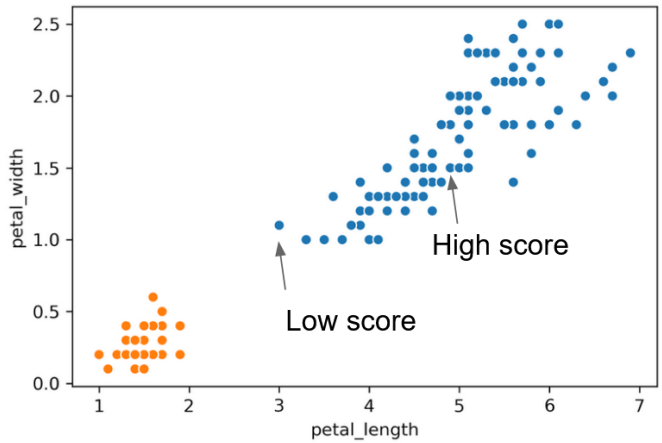
- A = average distance to other points in cluster.
- B = average distance to points in closest cluster.

$$S = \frac{B - A}{max(A, B)}$$

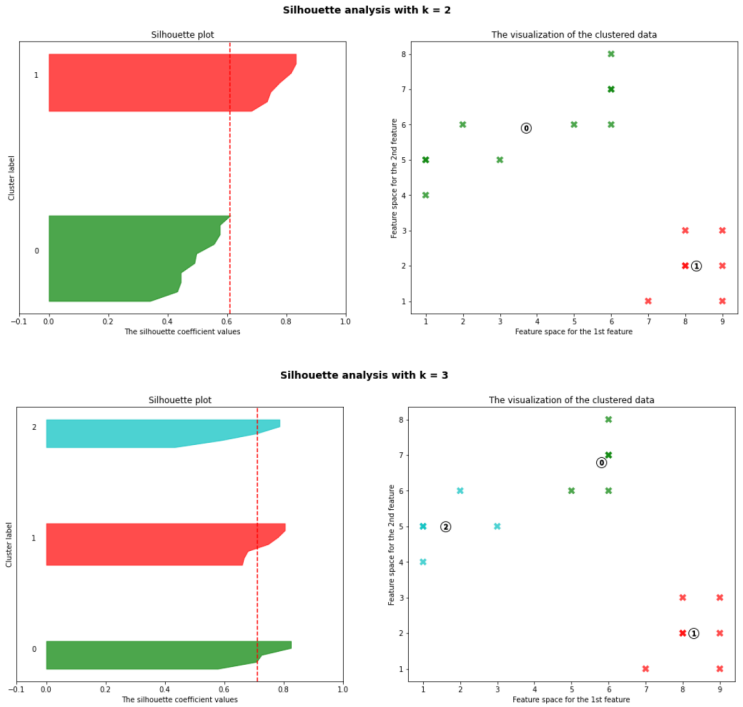
Can S be negative? **Yes**

Average distance to X's clustermates is larger than distance to the closest cluster.

Example: The "low score" point on the right has S = -0.13.



Silhouette Scores, k = 2 vs. k = 3



Code: [Elbow_and_Silhouette.ipynb]

Picking K: Real World Metrics

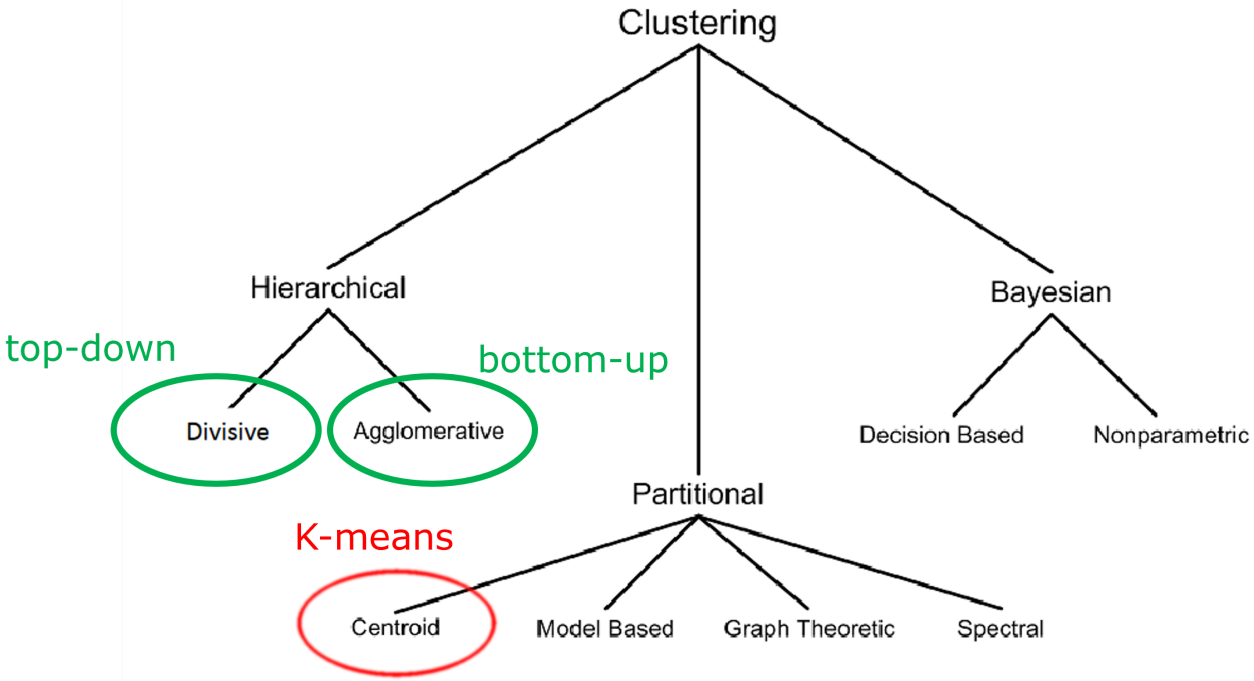
Technische Hochschule Rosenheim

- Sometimes you can rely on real world metrics to guide your choice of K.
- Perform 2 clusterings:
  - Cluster heights and weights of customers with K = 3 to design Small, Medium, and Large shirts.
  - Cluster heights and weights of customers with K = 5 to design XS, S, M, L, and XL shirts.
- To pick K:
  - Consider projected costs and sales for the 2 different Ks.
  - Pick the one that maximizes profit.

Hierarchical clustering

Terminology

Data clustering algorithms can be hierarchical or partitional.



Hierarchical Clustering

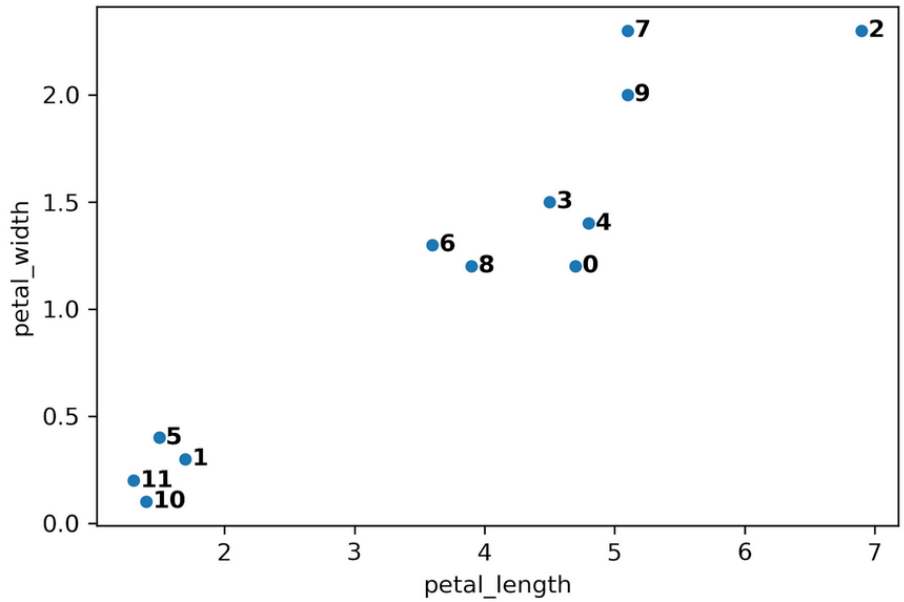
Technische Hochschule Rosenheim

- As with *regression* and *classification*, there are many ways to do clustering.
- So far we've seen K-Means, which attempts to minimize distortion.
  - Results not guaranteed to optimize distortion.
  - Even global optimum may not match our intuition of the best result.
- Let's discuss an alternate idea known as *hierarchical clustering*.
- Basic idea:
  - Every data point starts out as its own cluster.
  - Join clusters with neighbors until we have only k clusters left.

Let's see an example for k = 2.

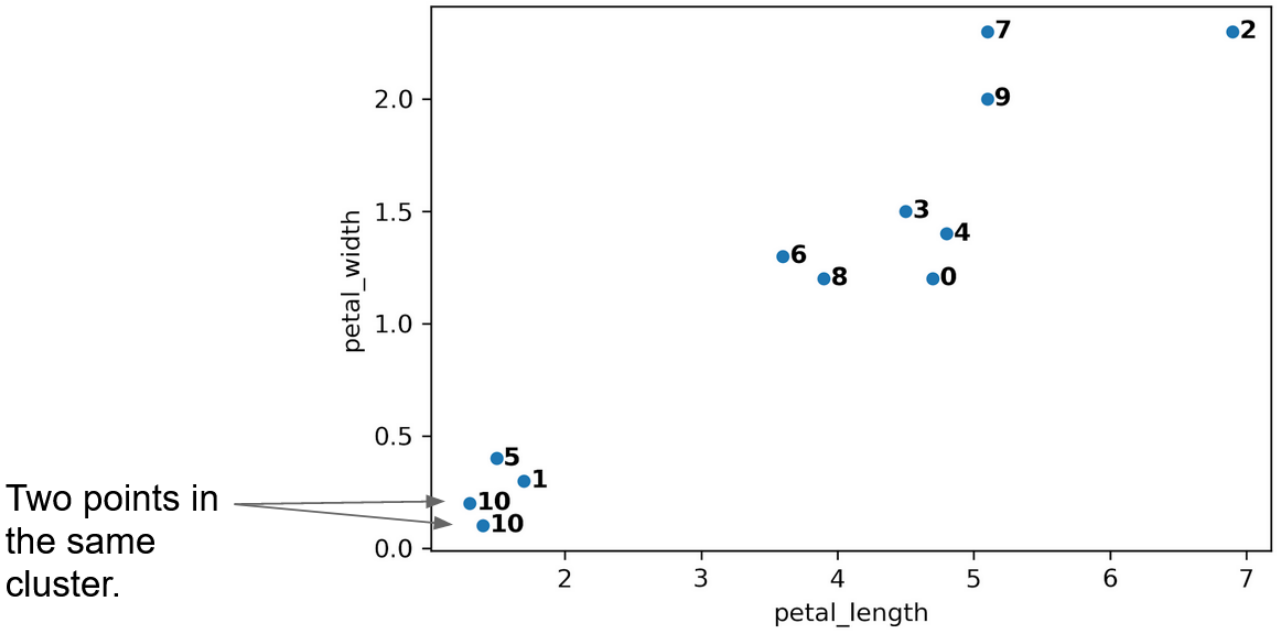
Hierarchical Clustering

- When the algorithm starts, every data point is in its own cluster.
  - Below, 12 data points, so 12 clusters.
  - Closest clusters are 10 and 11, so merge them.



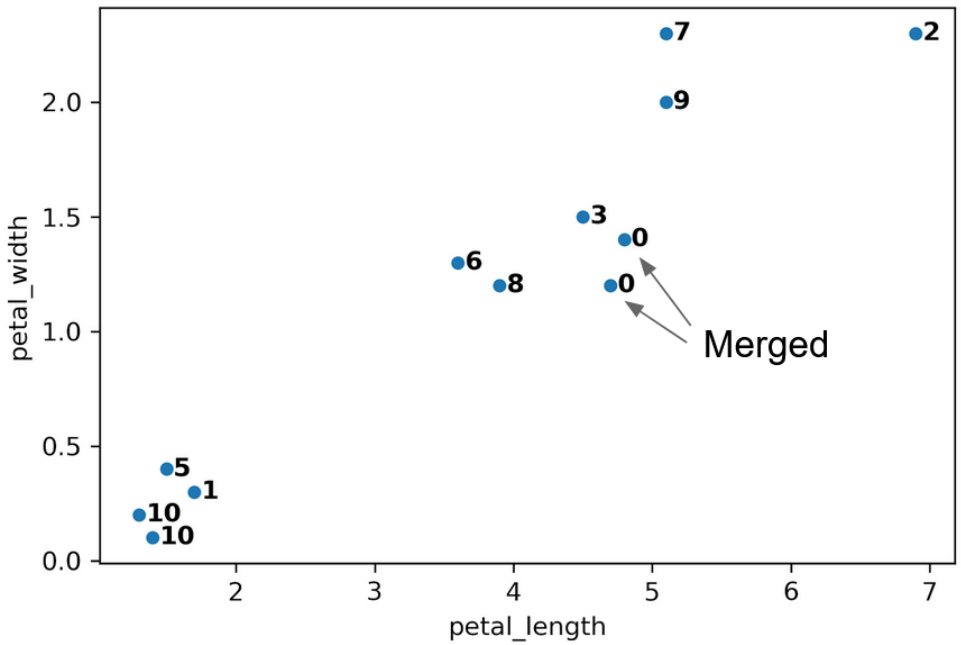
Hierarchical Clustering

- When the algorithm starts, every data point is in its own cluster.
  - Below, 12 data points, so 12 clusters.
  - Closest clusters are 10 and 11, so merge them.



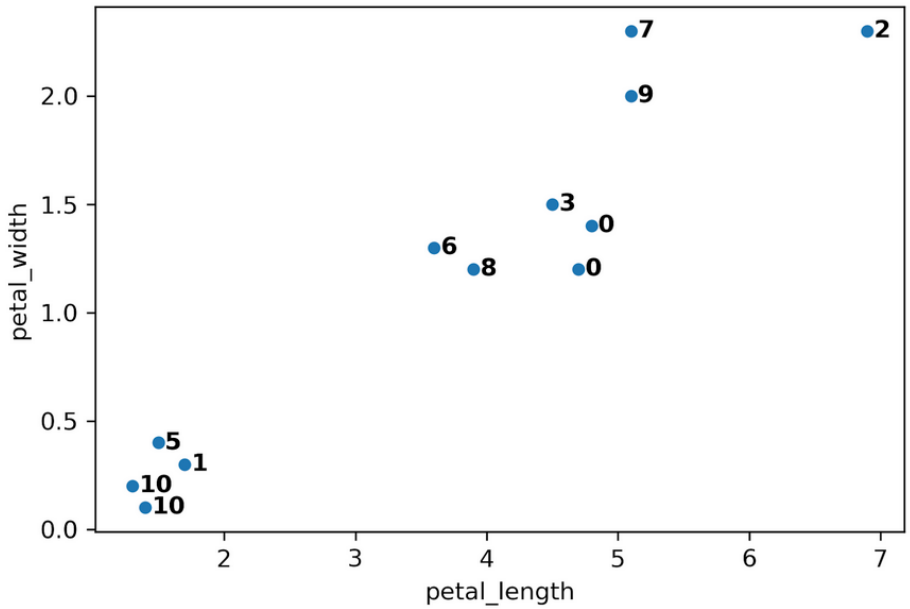
Hierarchical Clustering

- Next two closest are 0 and 4, so merge them.



Hierarchical Clustering

- At this point we have 10 clusters:
  - 8 with a single point {1, 2, 3, 5, 6, 7, 8, 9}.
  - 2 with two points {0/0, 10/10}.

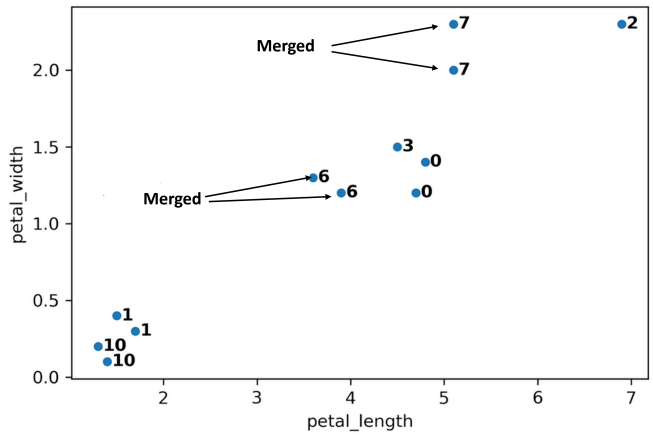


Hierarchical Clustering

- Next two closest clusters are 7 and 9.

  > Note: Might not look that way, but axes are not on the same scale! Y-axis goes only up to 2.5, and x axis goes up to more than 7.

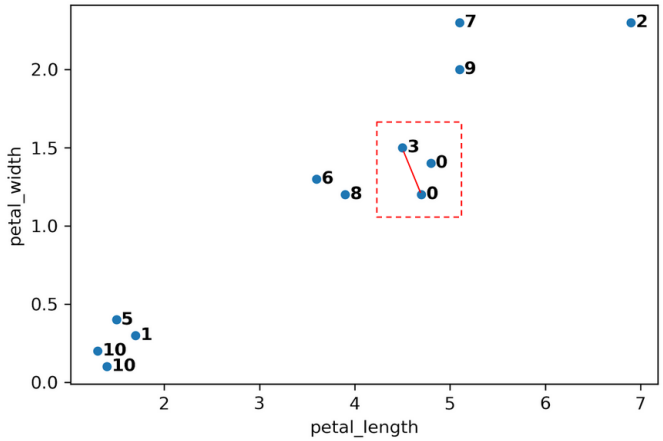
- Next closest are 6 and 8.



Hierarchical Clustering

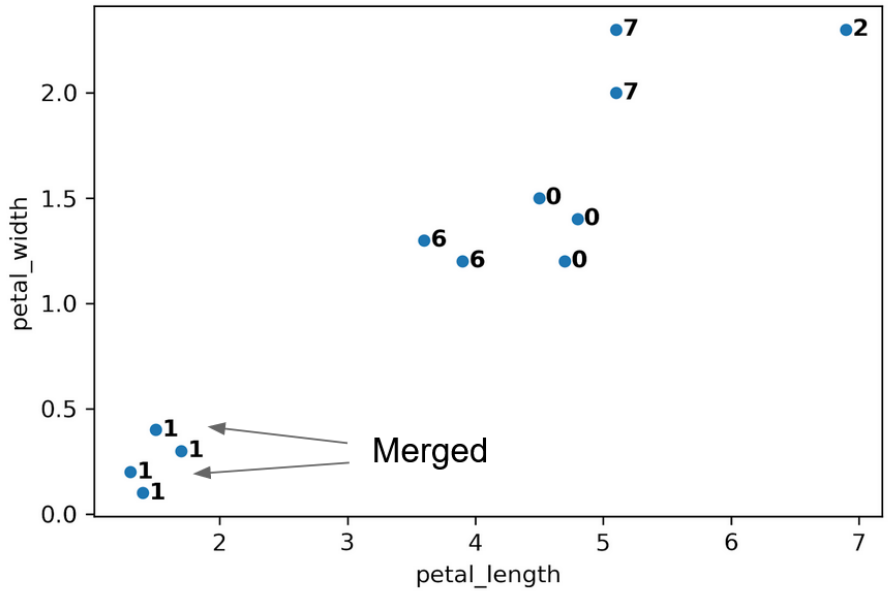
Technische Hochschule Rosenheim

- Tricky question:
  - What is the distance between clusters 0 and 3?
  - There is no right answer. Common choice, use the max, min or average. (Linkage)
  - Merge them next



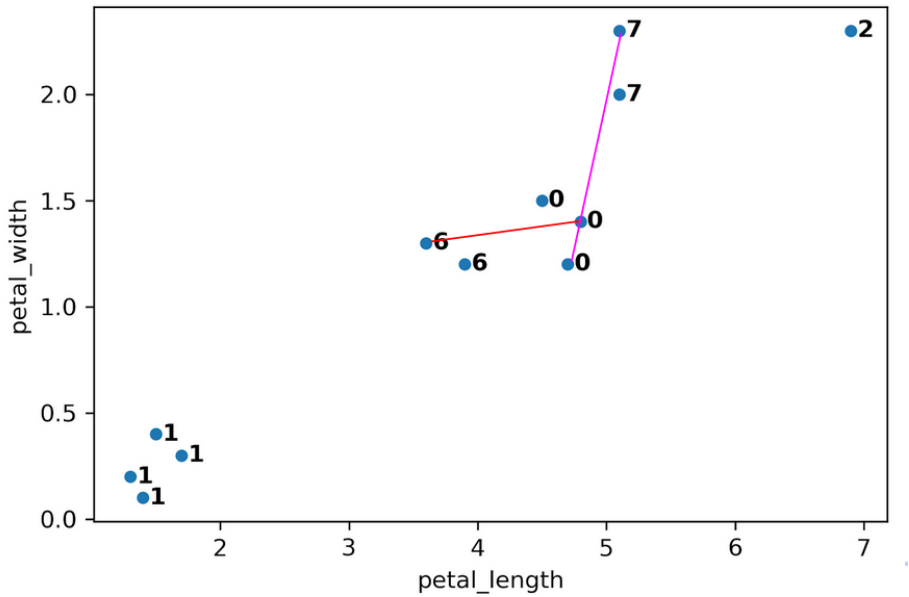
Hierarchical Clustering

- Same applies to 10 and 1
- Merge them



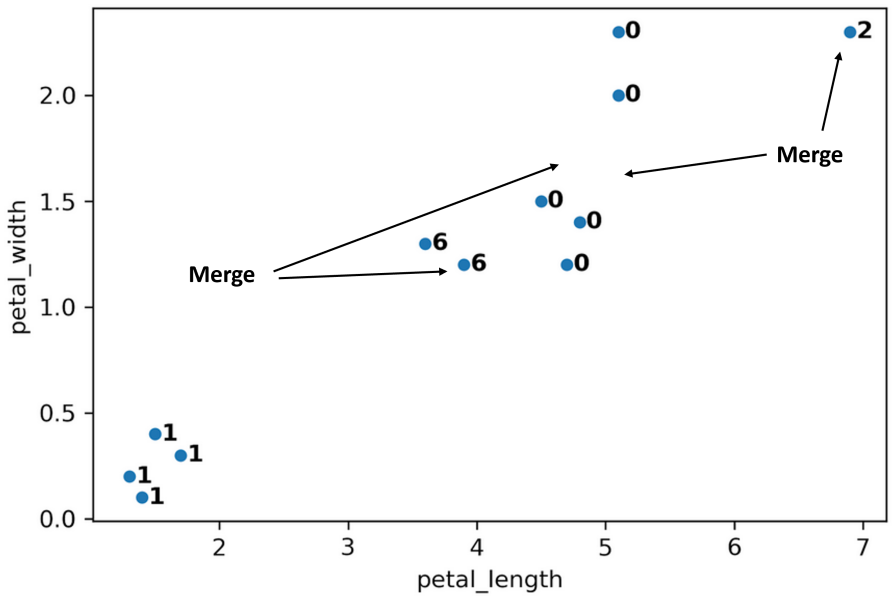
Hierarchical Clustering

- Next up are 0 and 7. Why?
- Max line between any member of 0 and 6 is longer than max line between any member of 0 and 7.



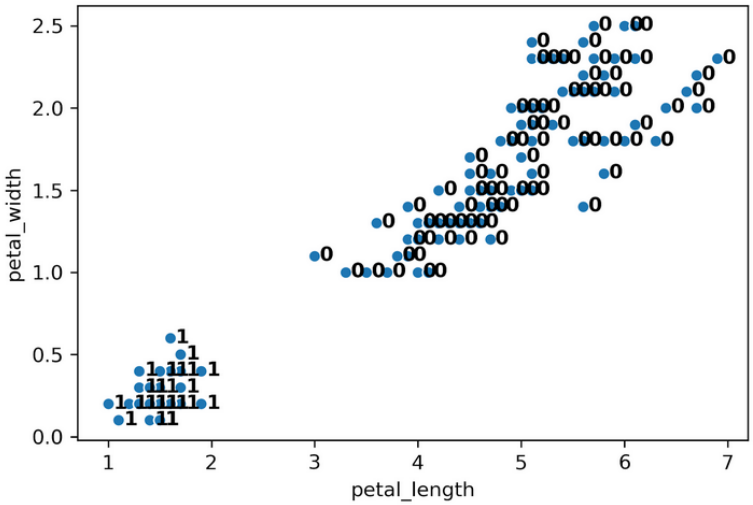
Hierarchical Clustering

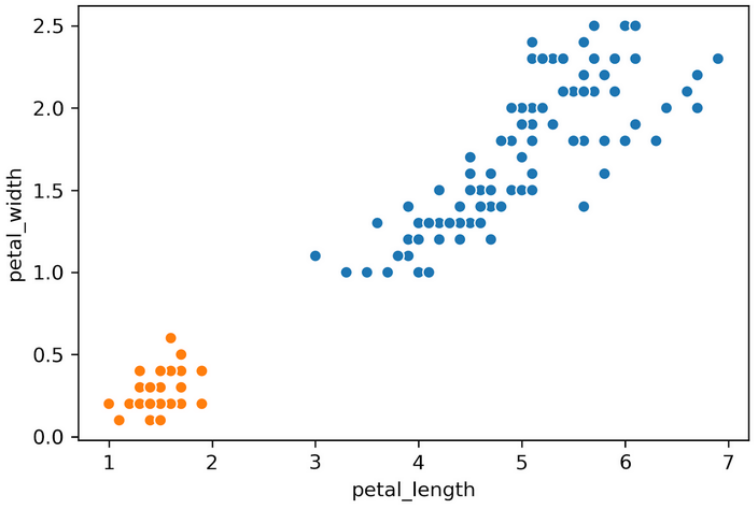
- Next up are 0 and 6.
- Then, 0 and 1.
- Merge them.



Finally: Hierarchical Clustering

- On the full dataset, our agglomerative clustering algorithm gets the “right” output.

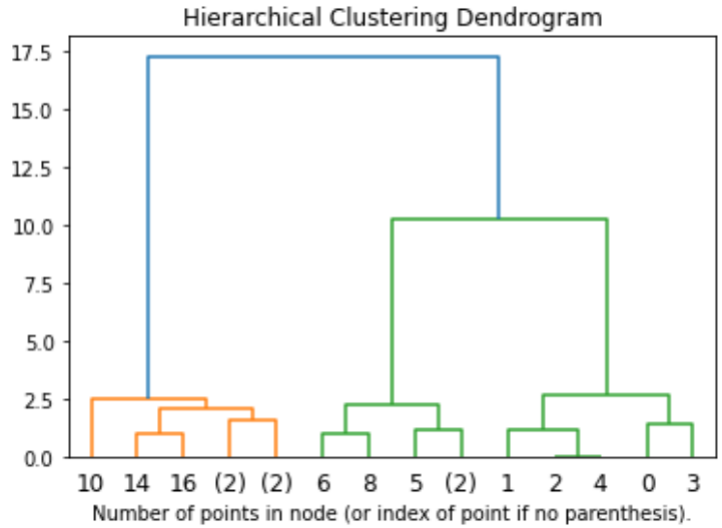




Clustering and Dendograms

Technische Hochschule Rosenheim

- Agglomerative clustering is one form of “hierarchical clustering”
- Can keep track of when two clusters got merged
- Each cluster is a tree
- Can visualize merging hierarchy, resulting in a *dendrogram*



> Looking at the dendrogram, the highest vertical distance that doesn’t intersect with any clusters hints to k.
>
> Code: ./HClustering.ipynb

Linkage

Technische Hochschule Rosenheim

- *Linkage* is used to calculate your centroids in hierarchical clustering
- *Linkage* is the concept of determining how you can calculate the distances within clusters

In SciPy package there is:

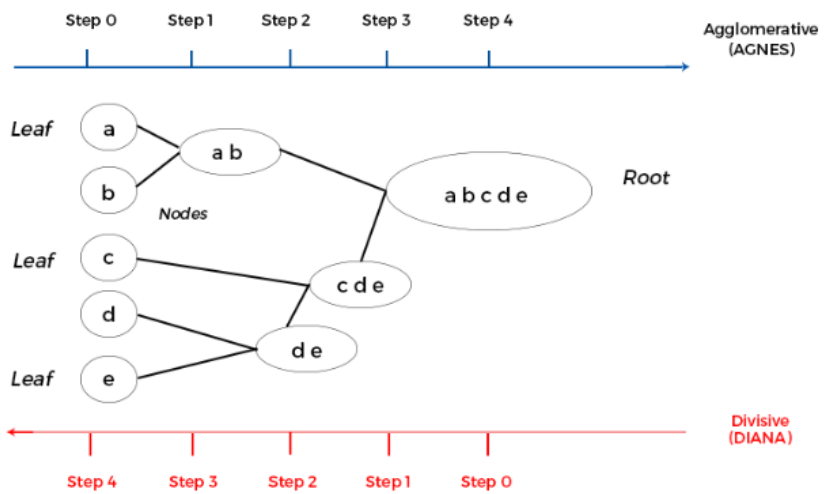
- **single**: `dist(a,b) = min(dist(a[i],b[j]))`
- **complete**: `dist(a,b) = min(dist(a,b))`
- more: **centroid**, **average**, **weighted** and **ward**

see documentation for linkage:
https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.linkage.html?highlight=linkage

> a[i] is i^th point in first cluster and b[j] is j^th point in second cluster

Divisive Clustering

- So far, *agglomerative* clustering which means bottom_up (most common!)
- The opposite, top-down, is called **divisive hierarchical clustering**



Agglomorative vs. Divisive

Technische Hochschule Rosenheim

- **Agglomerative clustering**: Commonly referred to as AGNES (AGglomerative NESting) works in a bottom-up manner.
- **Divisive hierarchical clustering**: Commonly referred to as DIANA (DIvise ANAlysis) works in a top-down manner.
- Divisive clustering is more complex as compared to agglomerative clustering, as in case of divisive clustering we need a flat clustering method as “subroutine” to split each cluster until we have each data having its own singleton cluster.
- Divisive clustering is more efficient if we do not generate a complete hierarchy all the way down to individual data leaves.
- Divisive algorithm is also more accurate. Agglomerative clustering makes decisions by considering the local patterns or neighbor points without initially taking into account the global distribution of data. These early decisions cannot be undone.

k-means vs H-Clustering

Technische Hochschule Rosenheim

- in hierarchical clustering you can start clustering without knowing k a priori
  - Start algorithm and cluster then decide which one makes sense
- k-means is super simplistic and easy to explain
- Hierarchical clustering has more parameters to tweak, thus it deals better with *abnormally* shaped data which results in better clusters (outlier)
- k-means might take some time to converge depending on centroids a picked at start

Summary

Technische Hochschule Rosenheim

- Today we discussed *Clustering*
- Two solutions:
  - K-Means: Tries to optimize a loss function called distortion. No known algorithm to do this optimally.
- Agglomerative/ hierarchical clustering
- Our version of these algorithms required a hyperparameter k
- 4 ways to pick K
  - Intuitively
  - Elbow method
  - Silhouette scores
  - harnessing real world metrics.

Exercise

- Work on Python!
- Work on Elbow method and Silhouette scores for a given dataset
- Try hierarchical clustering