Data Engineering

MG-GY 8441

Market Segmentation

- Agenda
  - Clustering
    - Partitioning methods
    - Hierarchical methods
- References
  - Han, Kamber, Pei, Data Mining: Concepts and Techniques (Chapter 10.1-10.3)
  - Optional
    - Leskovec, Rajaraman, Ullman, Mining of Massive Datasets (Chapter 7.1 - 7.3)

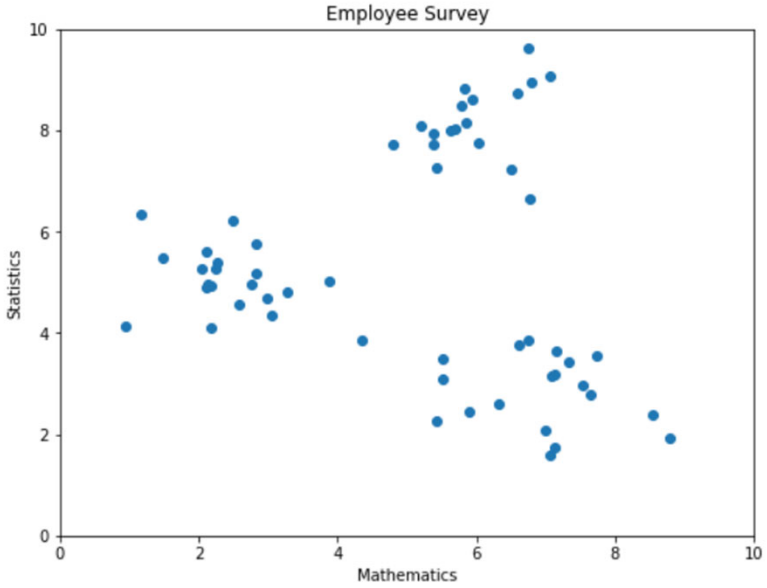
Example

Suppose we are project managers at a company. We perform a job task analysis to determine the staffing requirements for a project.

We have an employee survey from human resources about the backgrounds of the group.

We want to split the group into teams based on skill set.

- similar skills
- diverse skills



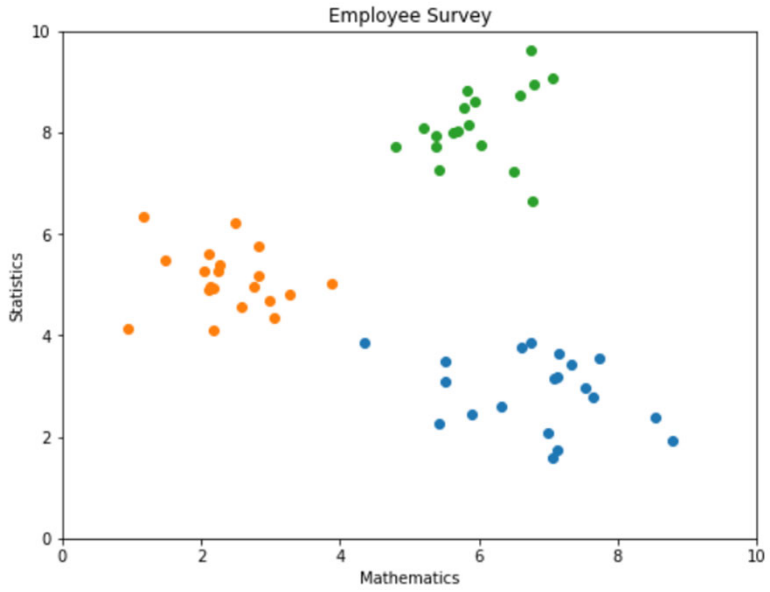
Example

Suppose we are project managers at a company. We perform a job task analysis to determine the staffing requirements for a project.

We have an employee survey from human resources about the backgrounds of the group.

We want to split the group into teams based on skill set.

- similar skills
- diverse skills



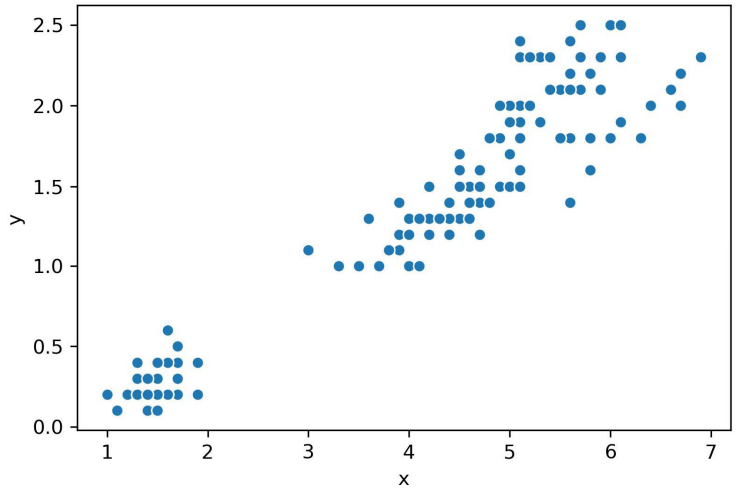
K-Means Clustering Algorithm

K-Means Clustering

Most popular clustering approach: K-Means.

- Pick an arbitrary k, and randomly place k “centers”, each a different color.
- Repeat until convergence:
  - Color points according to the closest center.
  - Move center for each color to center of points with that color.

Raw data →

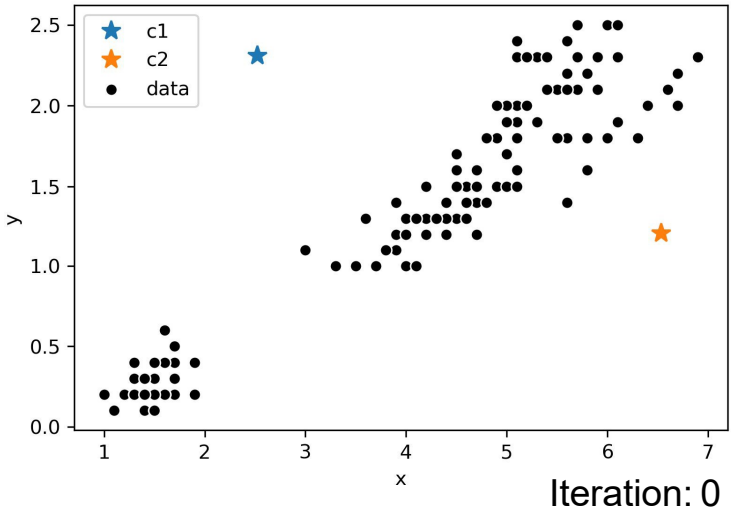


K-Means Clustering

Most popular clustering approach: K-Means.

- Pick an arbitrary k, and **randomly place k "centers",** each a different color.
- Repeat until convergence:
  - Color points according to the closest center.
  - Move center for each color to center of points with that color.

Initial random placement of two centers.

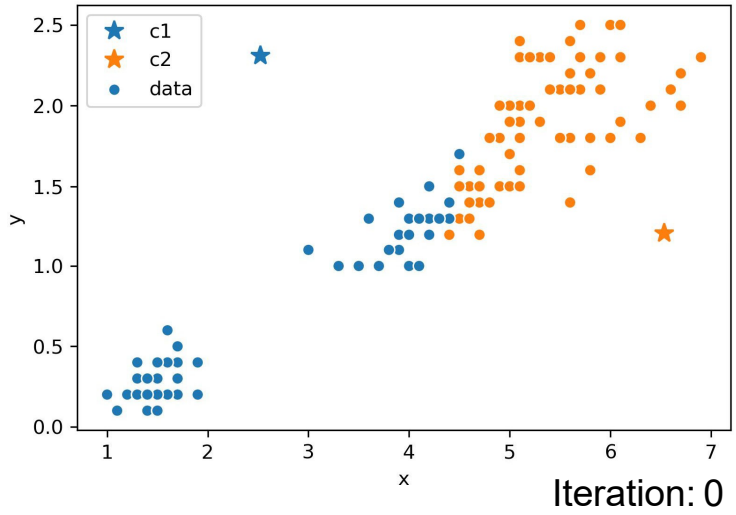


K-Means Clustering

Most popular clustering approach: K-Means.

- Pick an arbitrary k, and randomly place k “centers”, each a different color.
- Repeat until convergence:
  - **Color points according to the closest center.**
  - Move center for each color to center of points with that color.

Data colored by closest center.

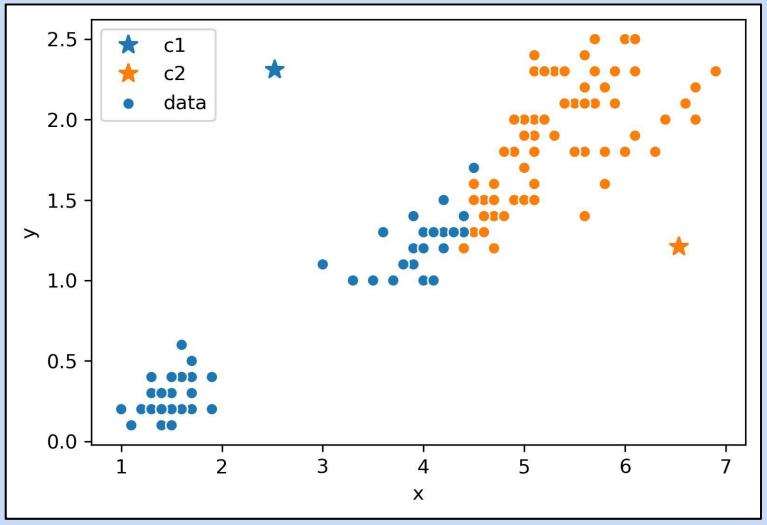


K-Means Clustering

Most popular clustering approach: K-Means.

- Pick an arbitrary k, and randomly place k “centers”, each a different color.
- Repeat until convergence:
  - Color points according to the closest center.
  - **Move center for each color to center of points with that color.**

Where should the centers go next?

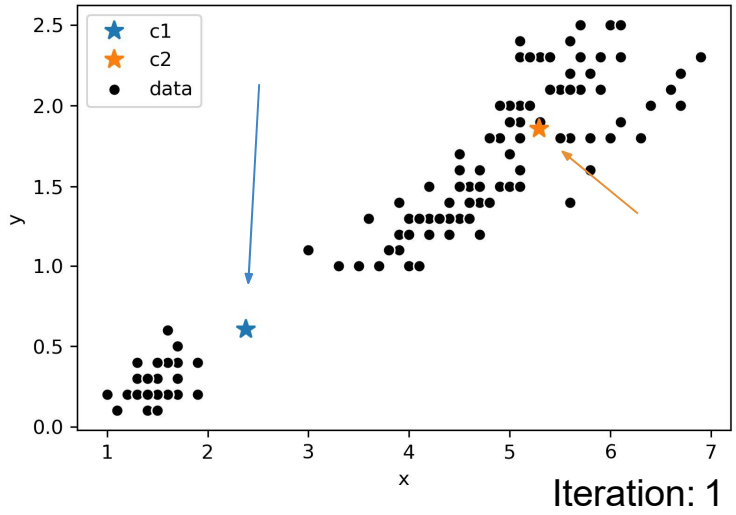


K-Means Clustering

Most popular clustering approach: K-Means.

- Pick an arbitrary k, and randomly place k “centers”, each a different color.
- Repeat until convergence:
  - Color points according to the closest center.
  - **Move center for each color to center of points with that color.**

Centers moved to their new homes.

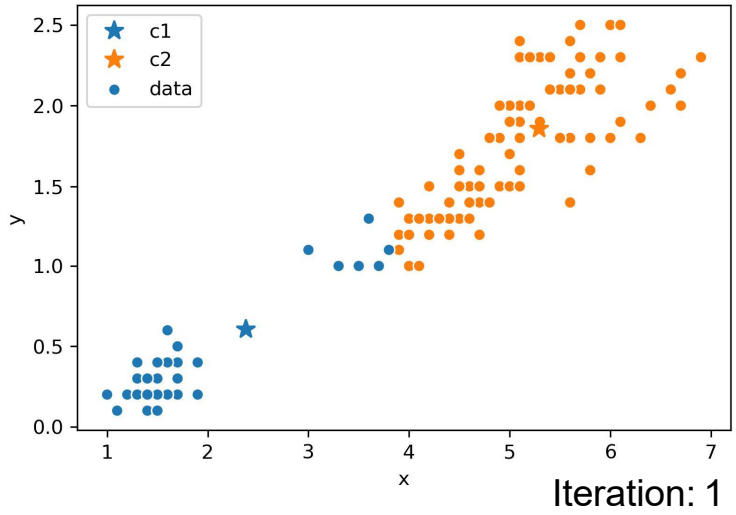


K-Means Clustering

Most popular clustering approach: K-Means.

- Pick an arbitrary k, and randomly place k “centers”, each a different color.
- Repeat until convergence:
  - **Color points according to the closest center.**
  - Move center for each color to center of points with that color.

Data colored by closest center (in new position).

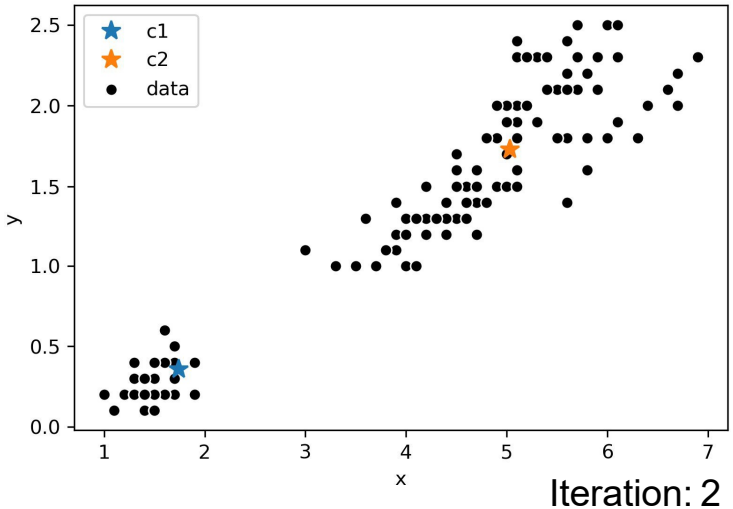


K-Means Clustering

Most popular clustering approach: K-Means.

- Pick an arbitrary k, and randomly place k “centers”, each a different color.
- Repeat until convergence:
  - Color points according to the closest center.
  - **Move center for each color to center of points with that color.**

Centers moved to new position.

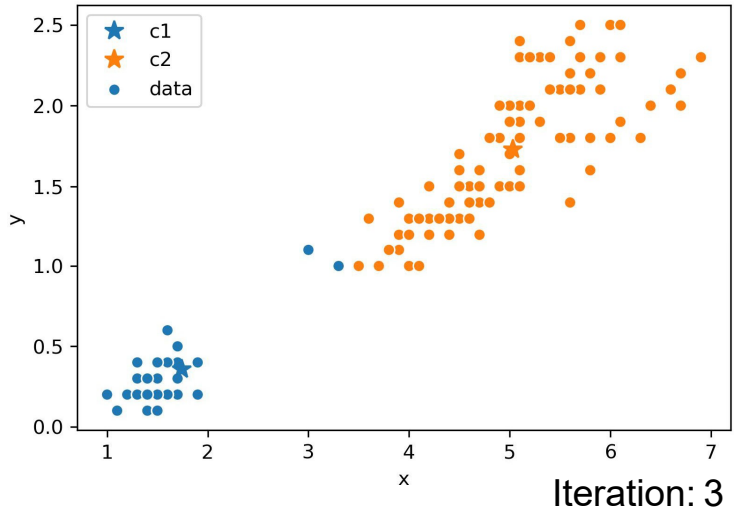


K-Means Clustering

Most popular clustering approach: K-Means.

- Pick an arbitrary k, and randomly place k “centers”, each a different color.
- Repeat until convergence:
  - **Color points according to the closest center.**
  - Move center for each color to center of points with that color.

Data colored by closest center (in new position).

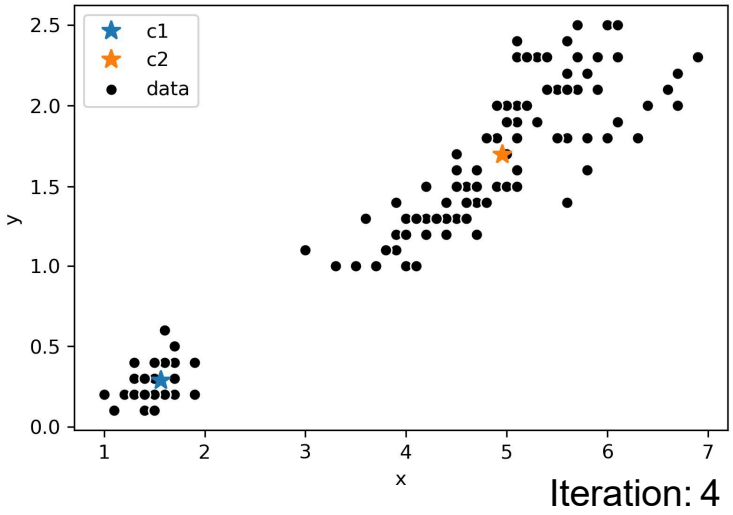


K-Means Clustering

Most popular clustering approach: K-Means.

- Pick an arbitrary k, and randomly place k “centers”, each a different color.
- Repeat until convergence:
  - Color points according to the closest center.
  - **Move center for each color to center of points with that color.**

Centers moved to new position.

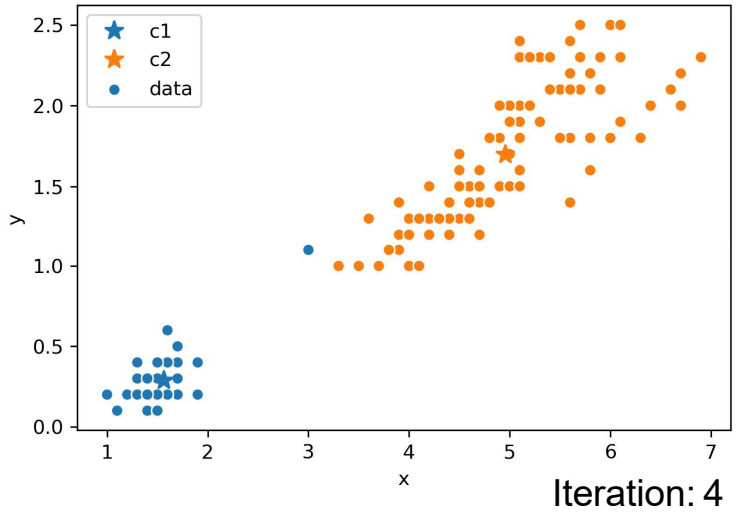


K-Means Clustering

Most popular clustering approach: K-Means.

- Pick an arbitrary k, and randomly place k “centers”, each a different color.
- Repeat until convergence:
  - **Color points according to the closest center.**
  - Move center for each color to center of points with that color.

Data colored by closest center (in new position).

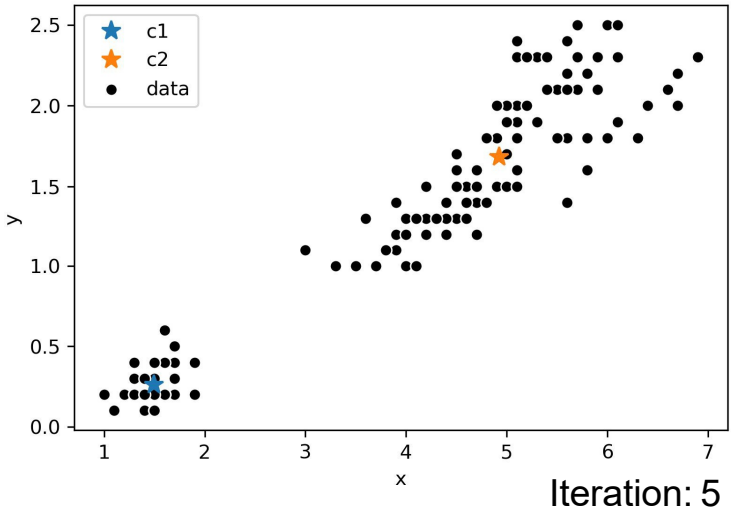


K-Means Clustering

Most popular clustering approach: K-Means.

- Pick an arbitrary k, and randomly place k “centers”, each a different color.
- Repeat until convergence:
  - Color points according to the closest center.
  - **Move center for each color to center of points with that color.**

Centers moved to new position.

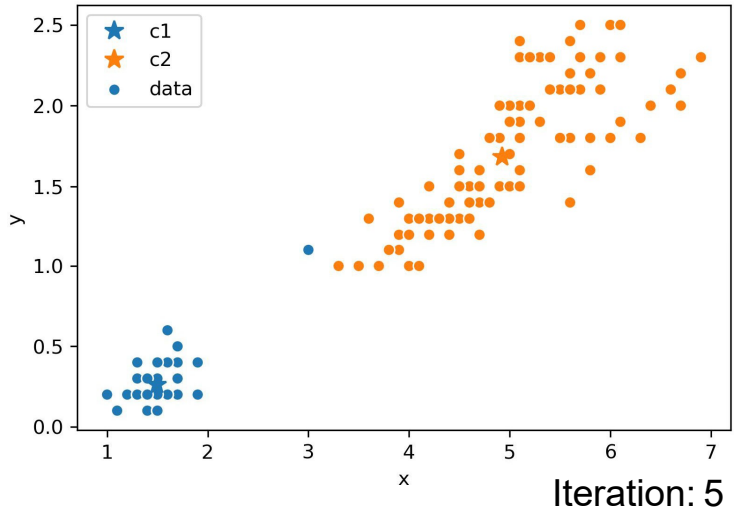


K-Means Clustering

Most popular clustering approach: K-Means.

- Pick an arbitrary k, and randomly place k “centers”, each a different color.
- Repeat until convergence:
  - **Color points according to the closest center.**
  - Move center for each color to center of points with that color.

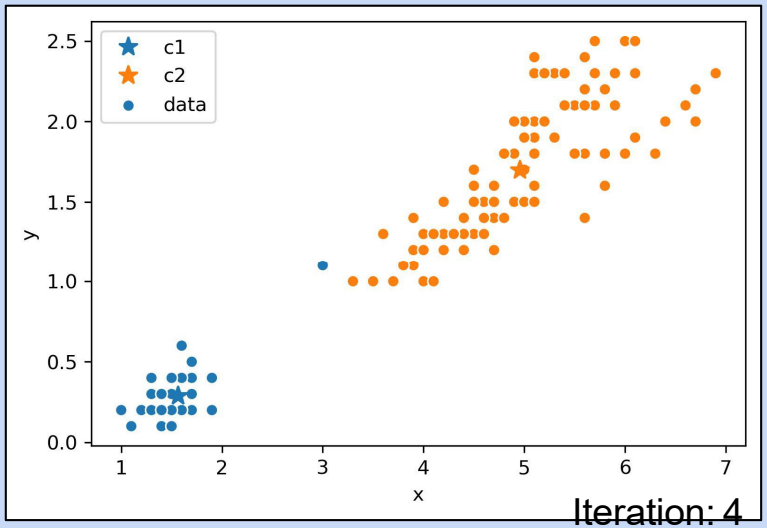
Data colored by closest center (in new position).

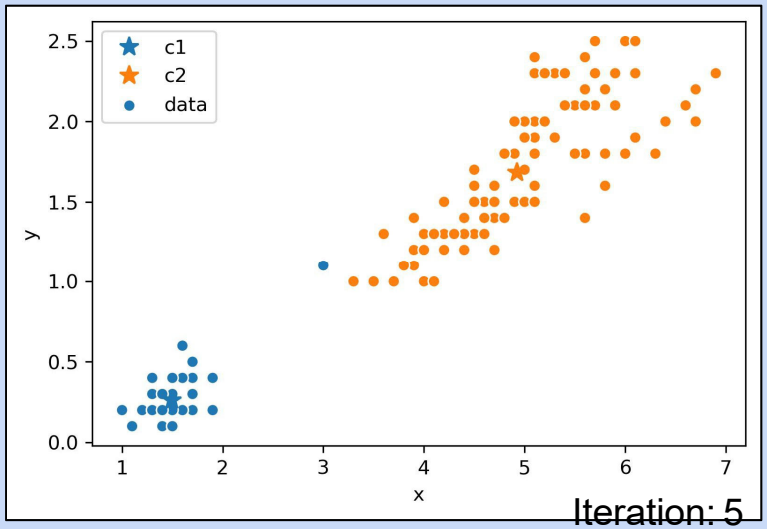


K-Means Clustering

Below we see the results after iteration 4 and 5.

- Centers moved slightly between iteration 4 and 5.
- But no points changed color.
- Are we done?

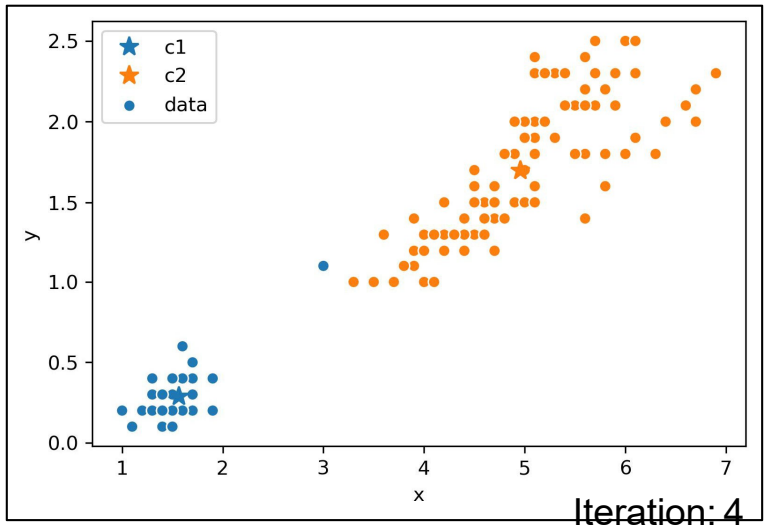


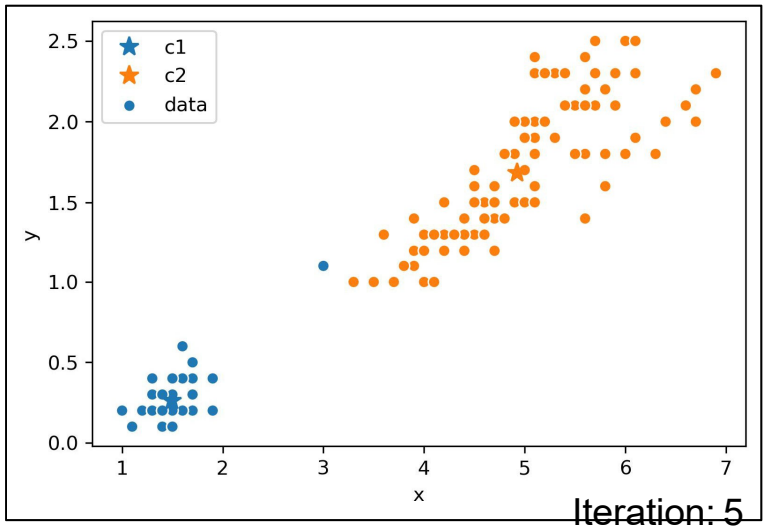


K-Means Clustering

Below we see the results after iteration 4 and 5.

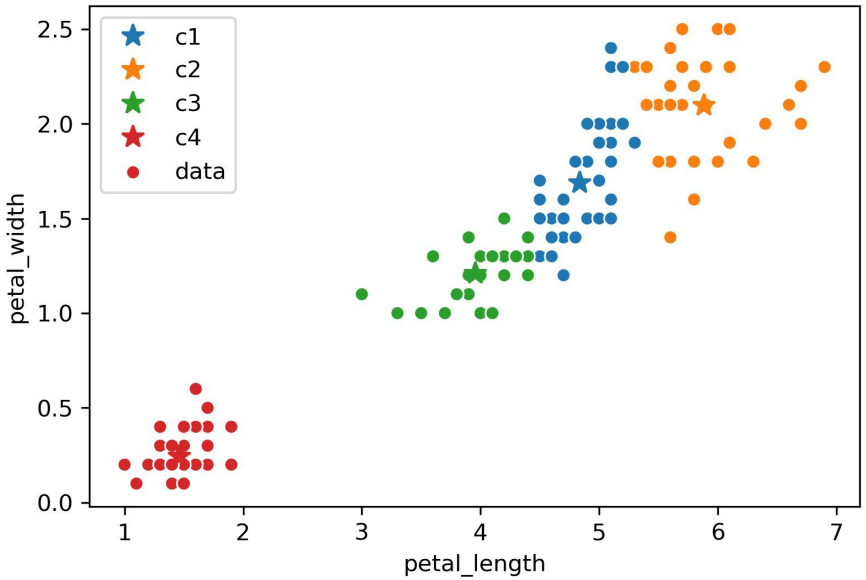
- Centers moved slightly between iteration 4 and 5.
- But no points changed color.
- Are we done?
  - Yes! If we tried iteration 6, we'd see that centers don't move at all.





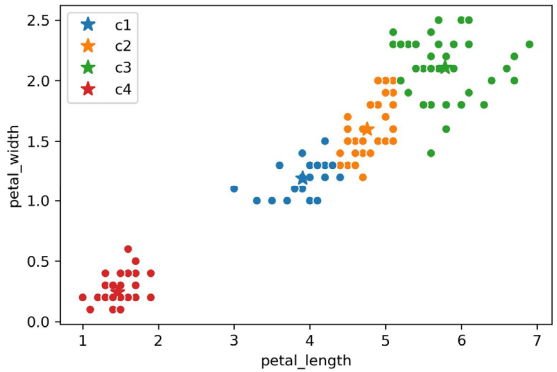
Minimizing Distortion

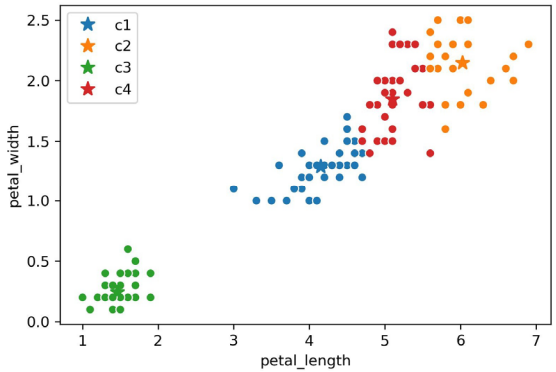
Below is an example of an output for K=4:

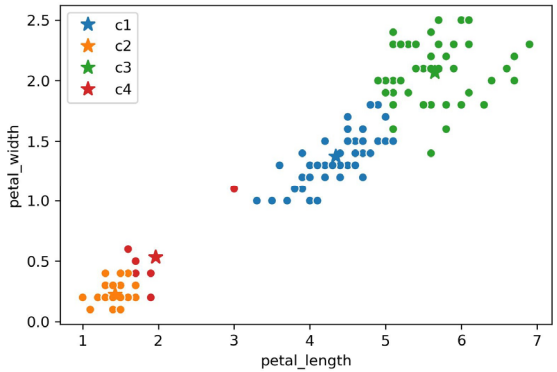


K-Means Clustering for K = 4

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





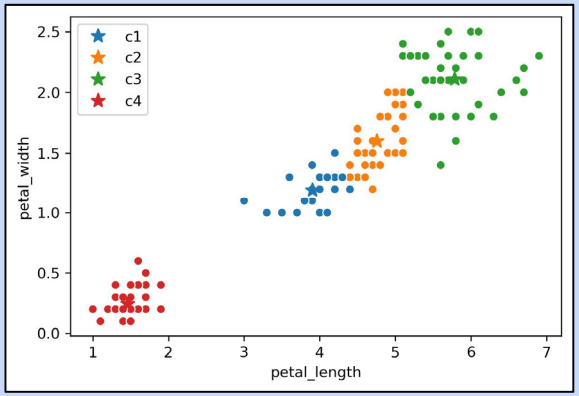


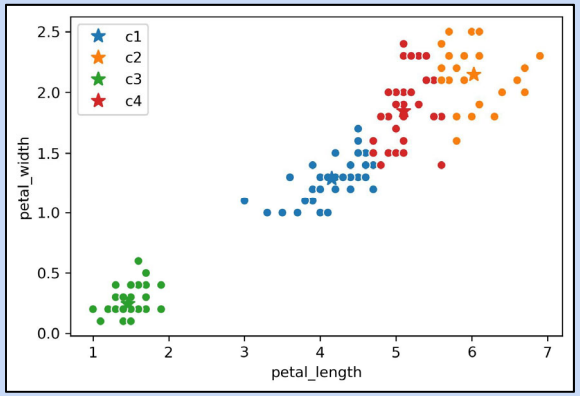
Which is best?

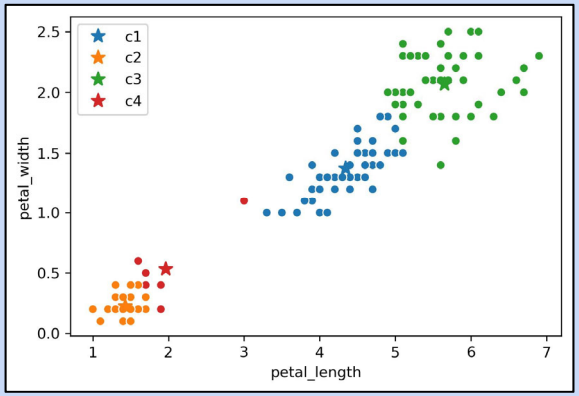
- One approach: Define some sort of loss function.

K-Means Clustering for K = 4

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





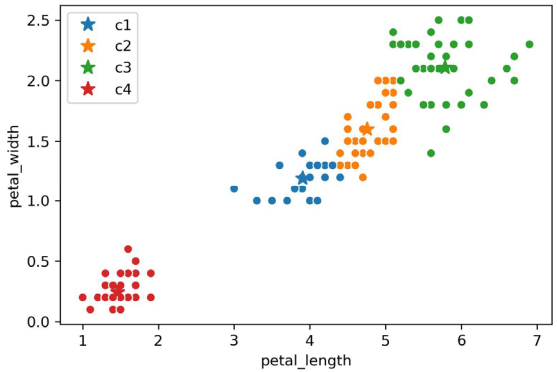


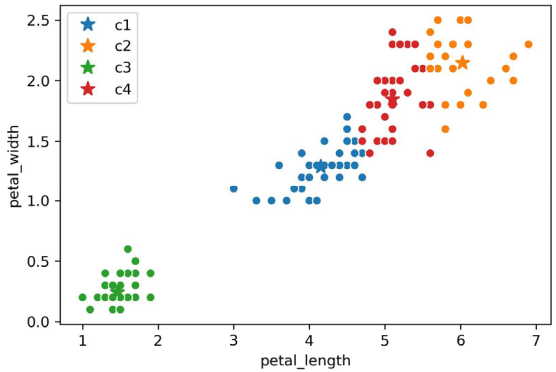
Which is best?

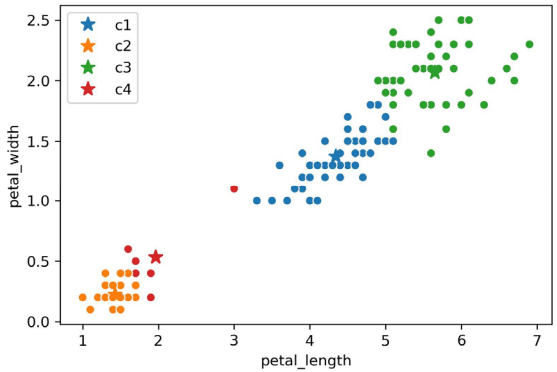
- One approach: Define some sort of loss function.
- Come up with a loss function for clustering.

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. Your 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

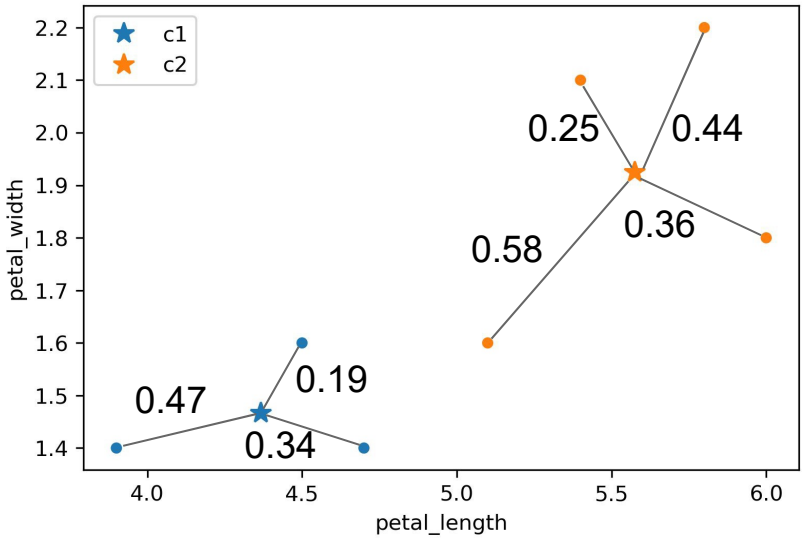
To evaluate different clustering results, we need a loss function.

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.

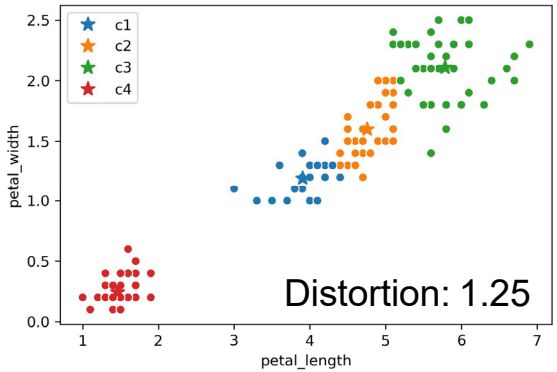
Example:

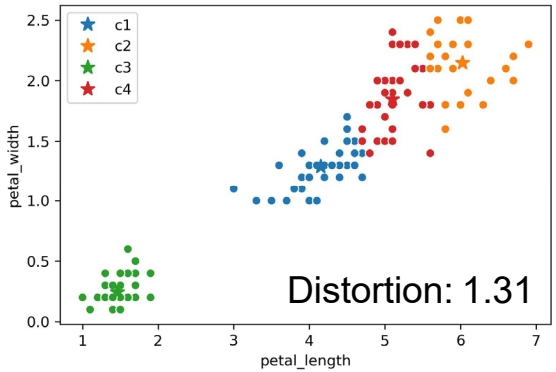
- Inertia: $0.47^2 + 0.19^2 + 0.34^2 + 0.25^2 + 0.58^2 + 0.36^2 + 0.44^2$
- Distortion: $(0.47^2 + 0.19^2 + 0.34^2)/3 + (0.25^2 + 0.58^2 + 0.36^2 + 0.44^2)/4$

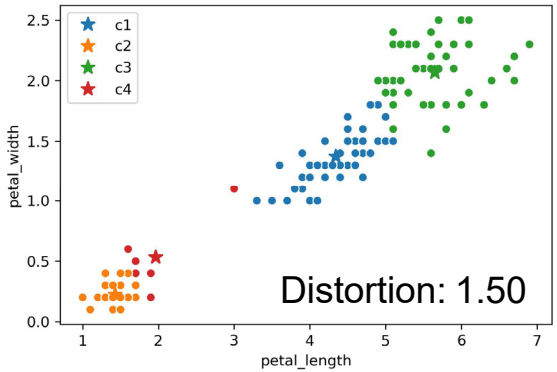


K-Means Clustering for K = 4

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







Our loss function says that the leftmost clustering is best (distortion: 1.25) and rightmost clustering (distortion: 1.5) is worst.

K-Means and Distortion

It turns out that the function K-Means is trying to minimize is distortion.

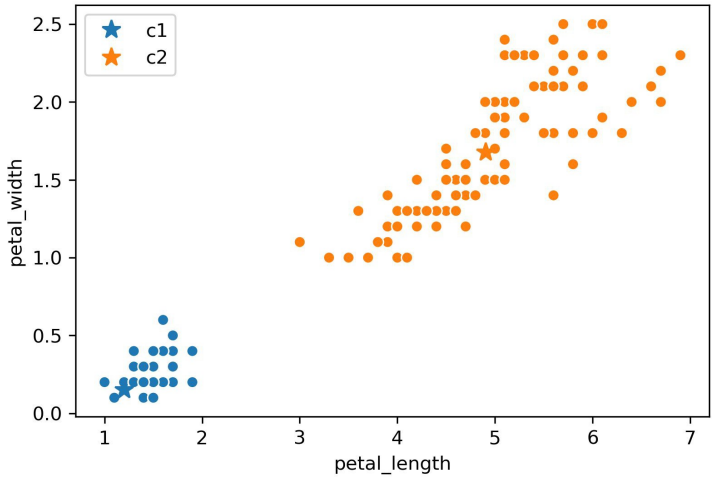
… but often fails to find global optimum. Why? Sketch below.

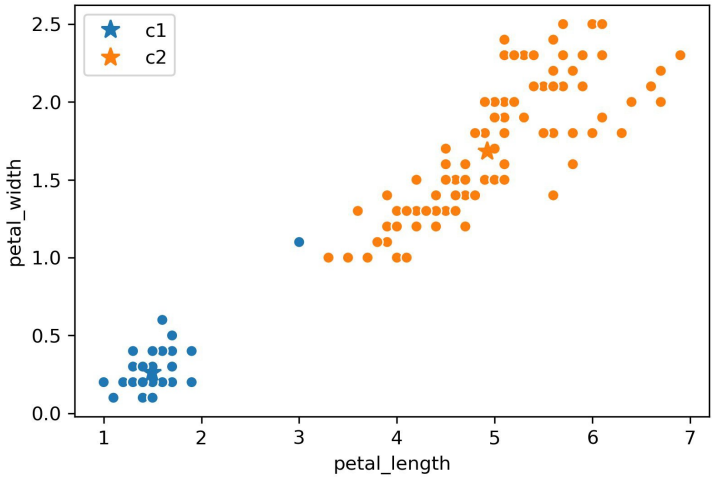
Can think of K-means as a pair of optimizers that take turns.

- First optimizer:
  - Holds center positions constant.
  - Optimizes data colors.
- Second optimizer:
  - Holds data colors constant.
  - Optimizes center positions.
- Neither gets total control.

Agglomerative Clustering

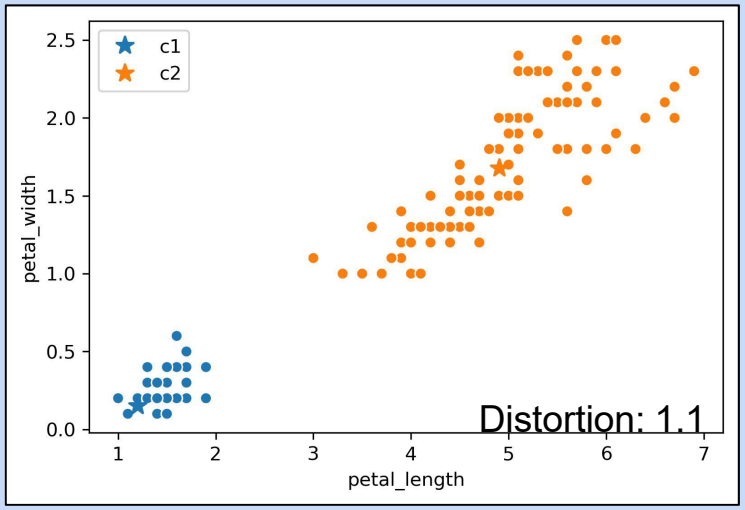
Which clustering result do you like better?

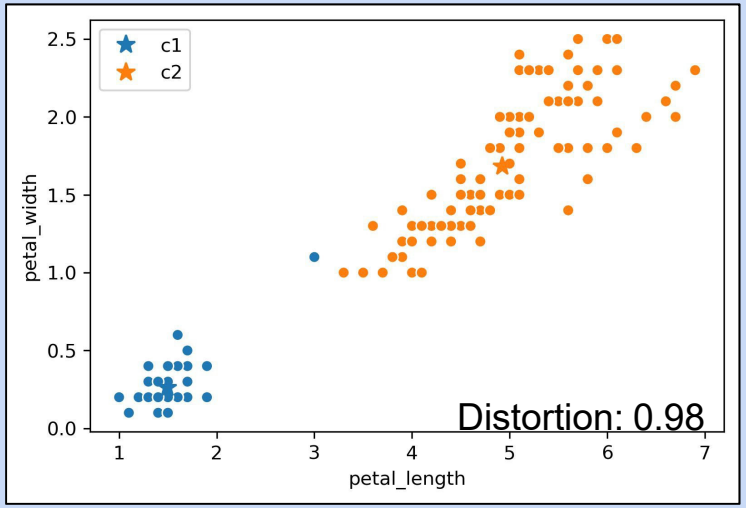




K-Means

Which clustering result do you like better?



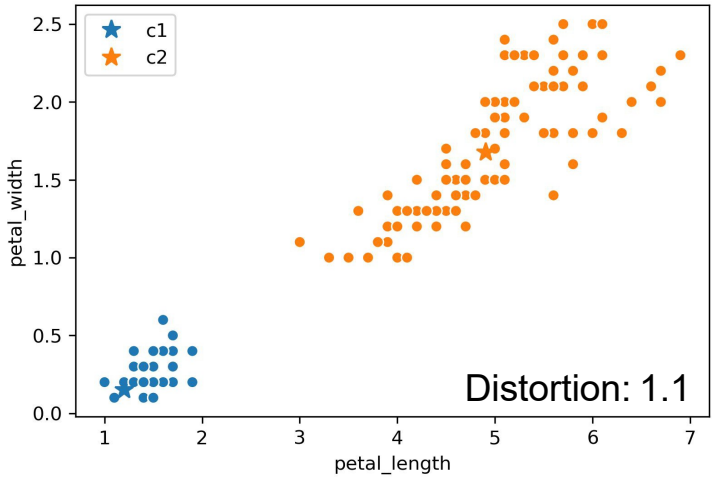


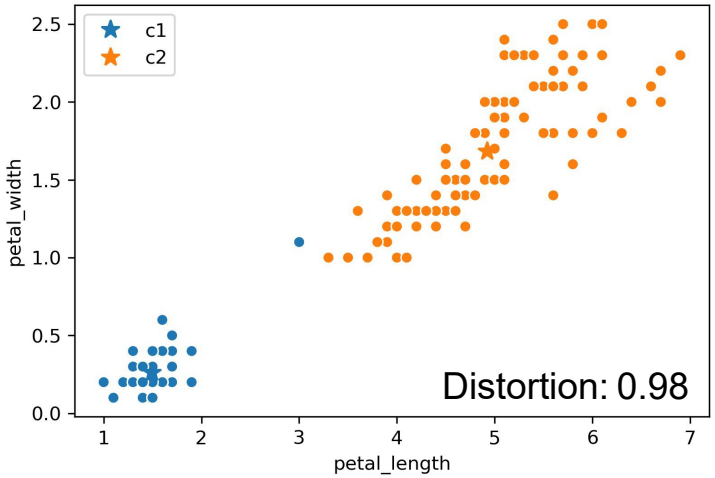
K-Means likes the one on the right better. It has lower distortion.

- Why is the distortion on the right lower?
- Is clustering on the right “wrong”?

K-Means

Which clustering result do you like better?





K-Means likes the one on the right better. It has lower distortion.

- Why is the distortion lower? K-Means optimizes for distance, not “blobbiness”.
- Is clustering on the right “wrong”? Good question!

Agglomerative Clustering

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 "agglomerative clustering".

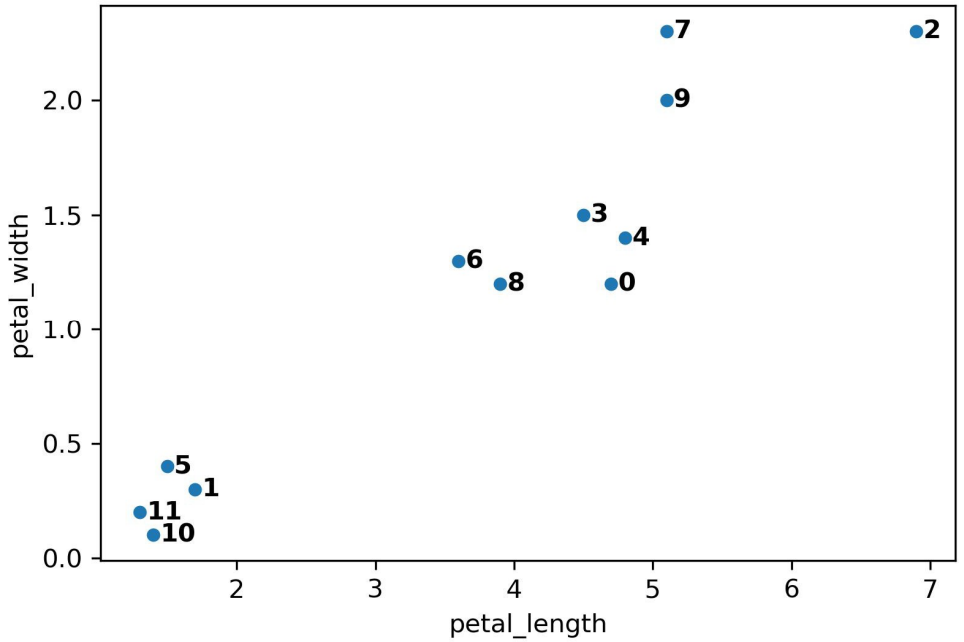
- Basic idea:
  - Every data point starts out as its own cluster.
  - Join clusters with neighbors until we have only K clusters left.

Let's see an example for K = 2.

Agglomerative Clustering Example

When the algorithm starts, every data point is in its own cluster.

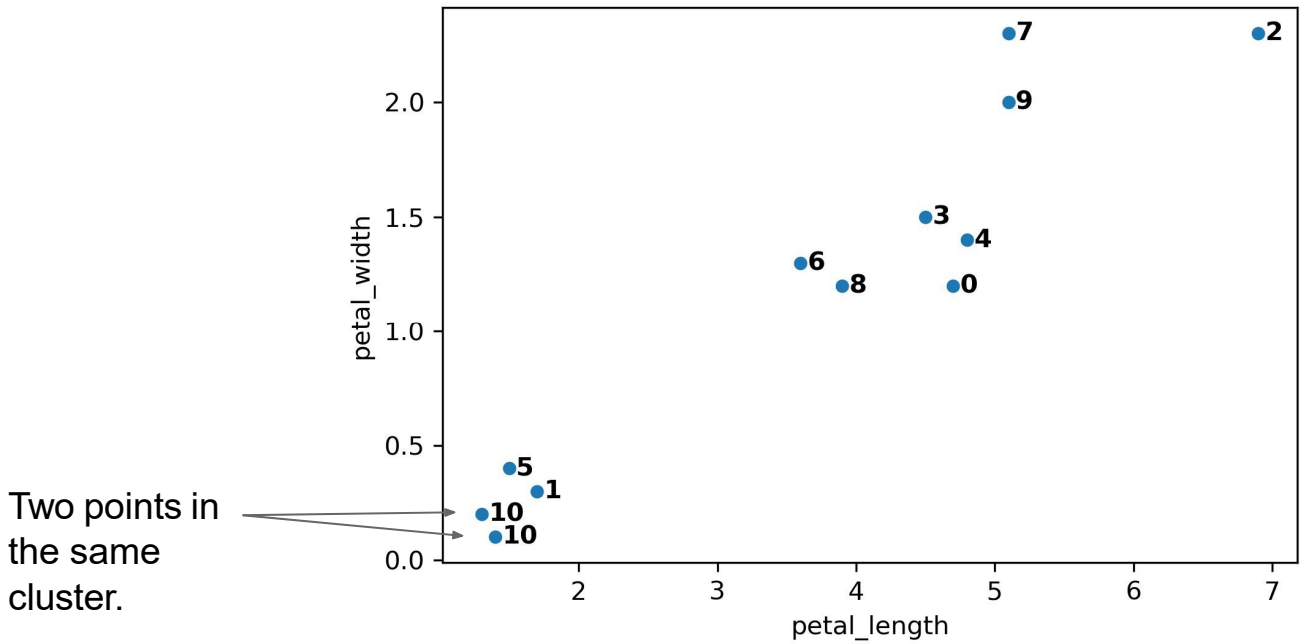
- Below, 12 data points, so 12 clusters.
- Closest clusters are 10 and 11, so merge them.



Agglomerative Clustering Example

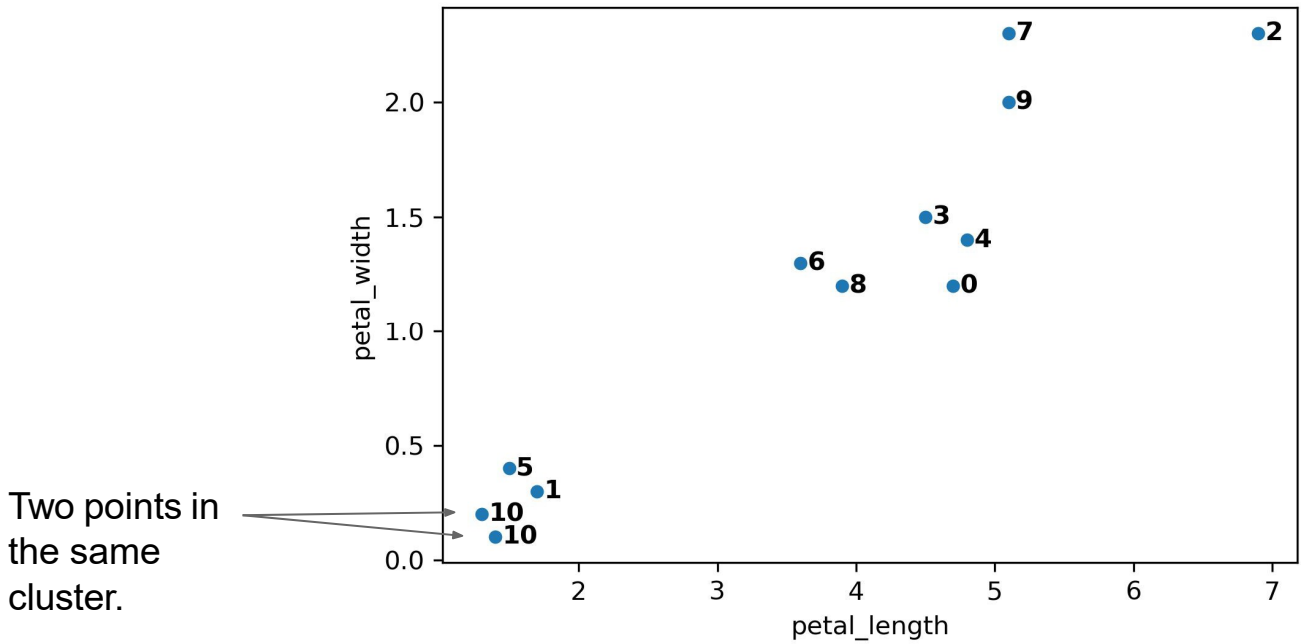
When the algorithm starts, every data point is in its own cluster.

- Below, 12 data points, so 12 clusters.
- Closest clusters are 10 and 11, so merge them.



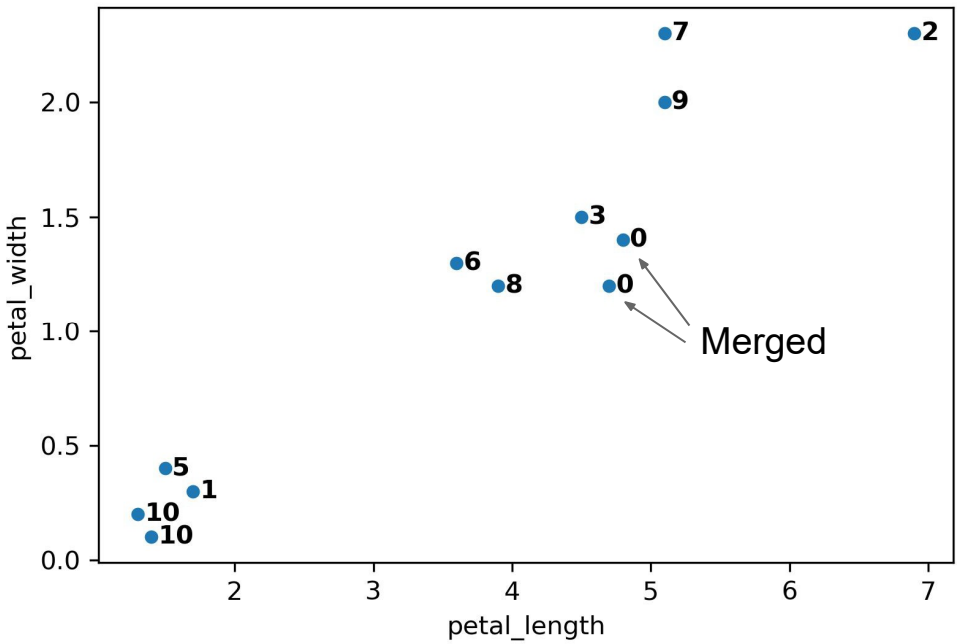
Agglomerative Clustering Example

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



Agglomerative Clustering Example

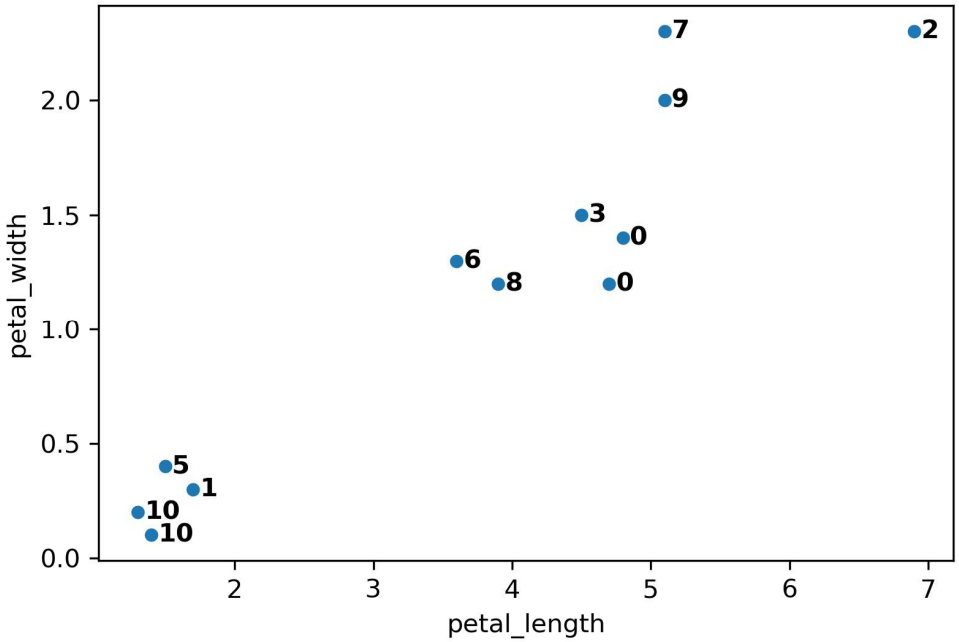
Next two closest are 0 and 4, so merge them.



Agglomerative Clustering Example

At this point we have 10 clusters:

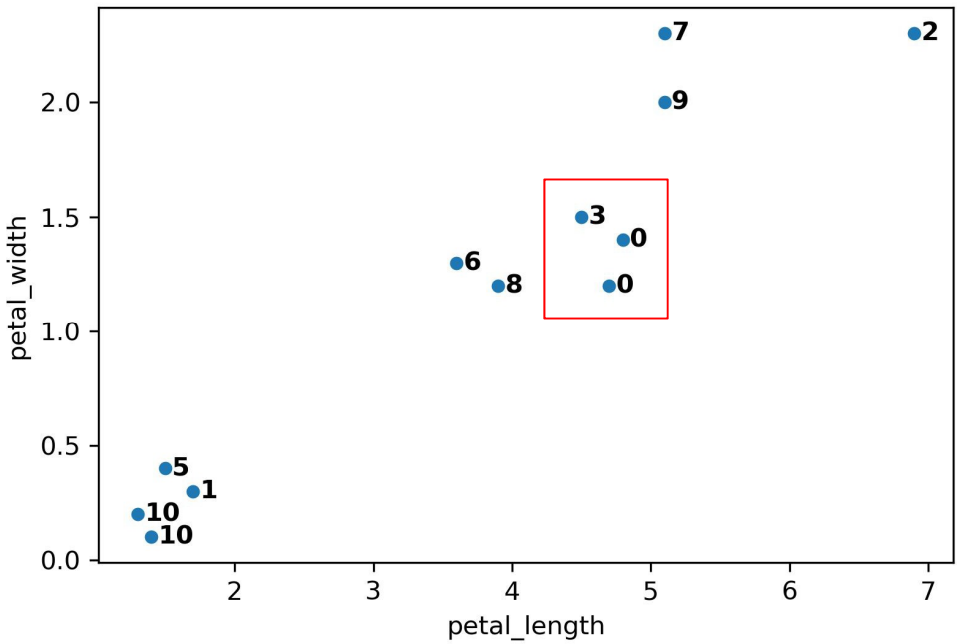
- 8 with a single point {1, 2, 3, 5, 6, 7, 8, 9}.
- 2 with two points {0/0, 10/10}.



Agglomerative Clustering Example

Tricky question:

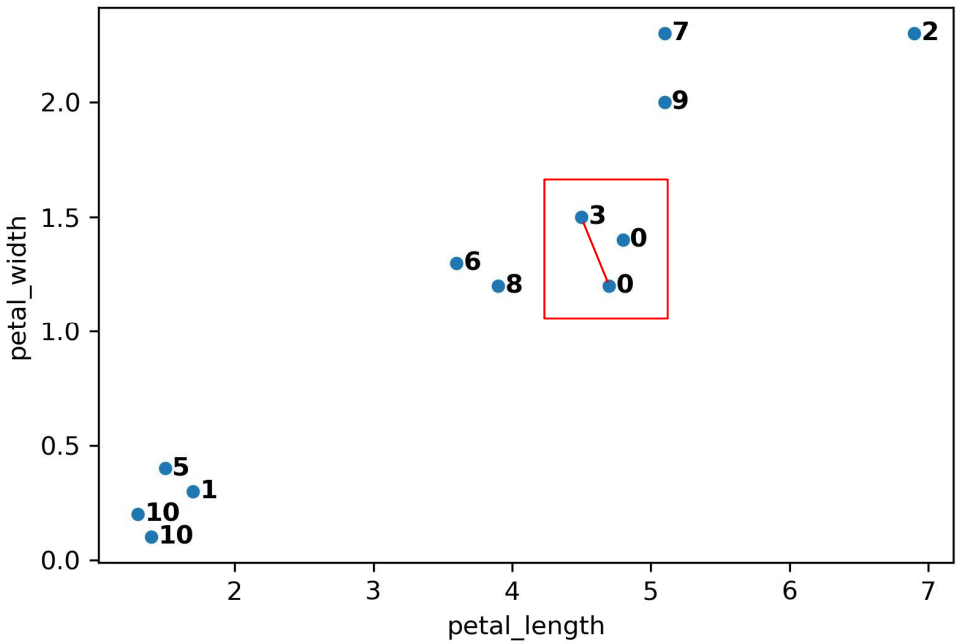
- What is the distance between clusters 0 and 3?



Agglomerative Clustering Example

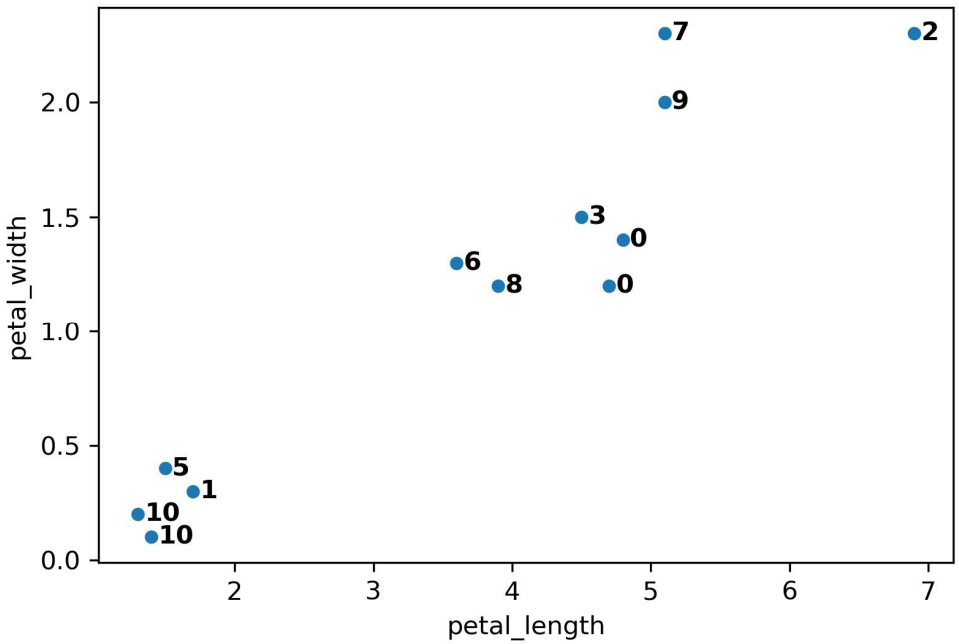
Tricky question:

- What is the distance between clusters 0 and 3?
- There is no right answer. Common choice, use the **max**.



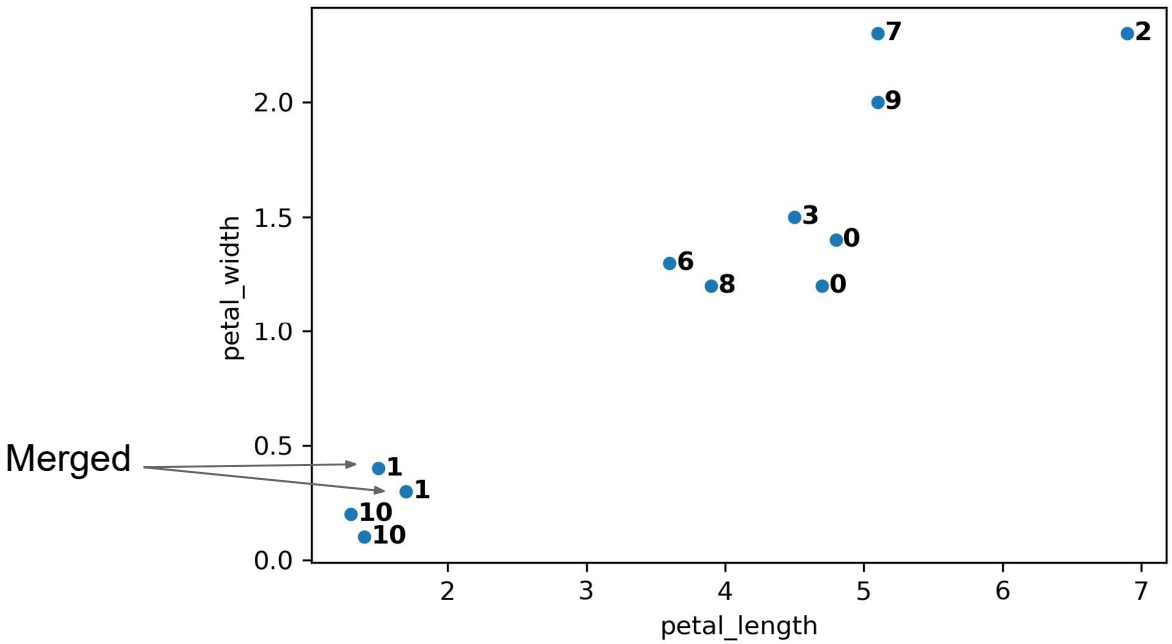
Would be perfectly reasonable to do something else, e.g. average or minimum.

Next two closest clusters are 1 and 5.



Agglomerative Clustering Example

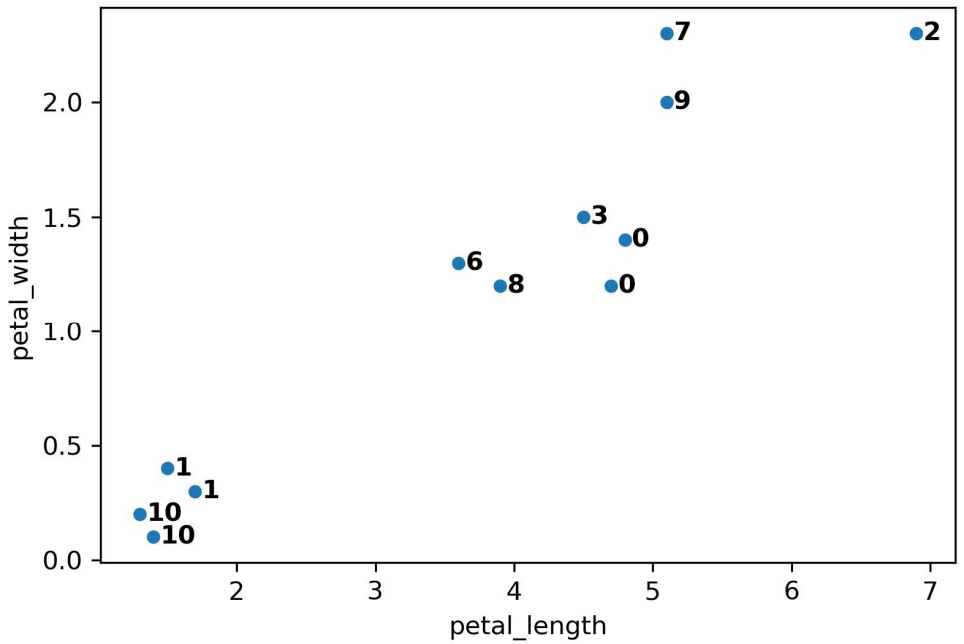
Next two closest clusters are 1 and 5.



Agglomerative Clustering Example

Next two closest clusters are 7 and 9.

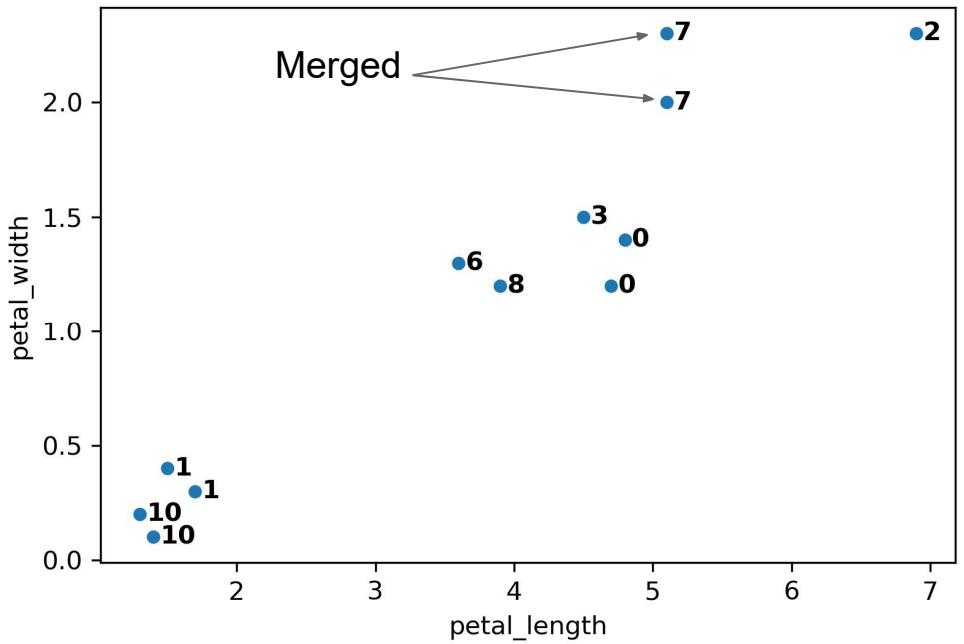
- Note: Might not look that way, but axes are not on the same scale! Y-axis goes only up to 2.5, and x axis goes up to more than 7.



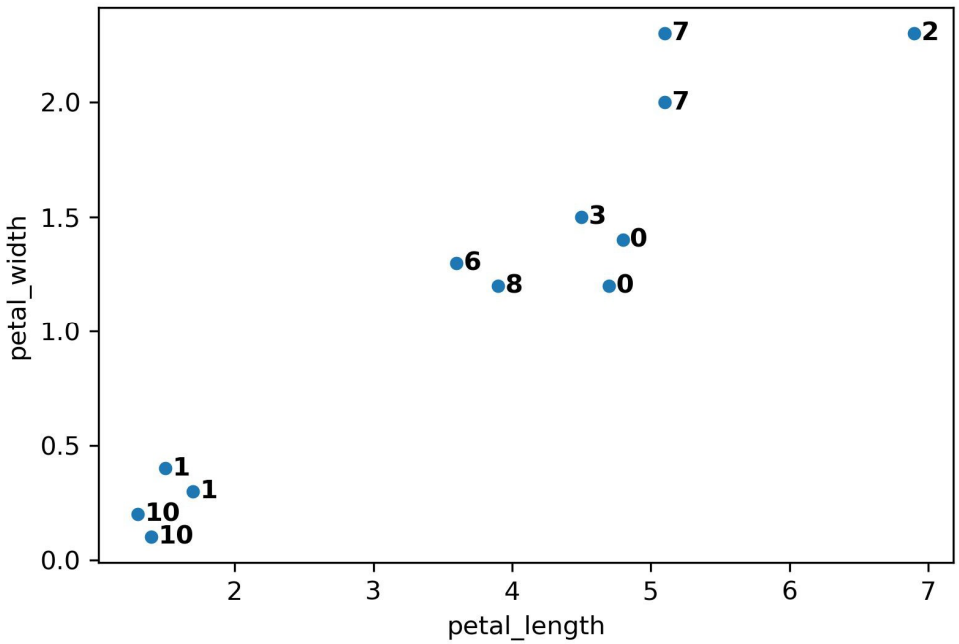
Agglomerative Clustering Example

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.

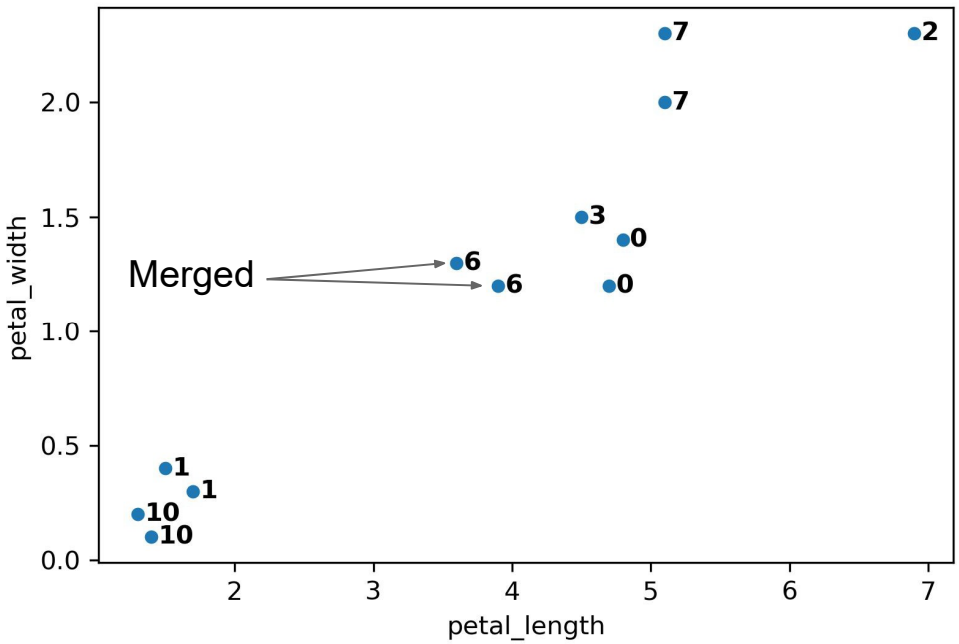


Next closest are 6 and 8.



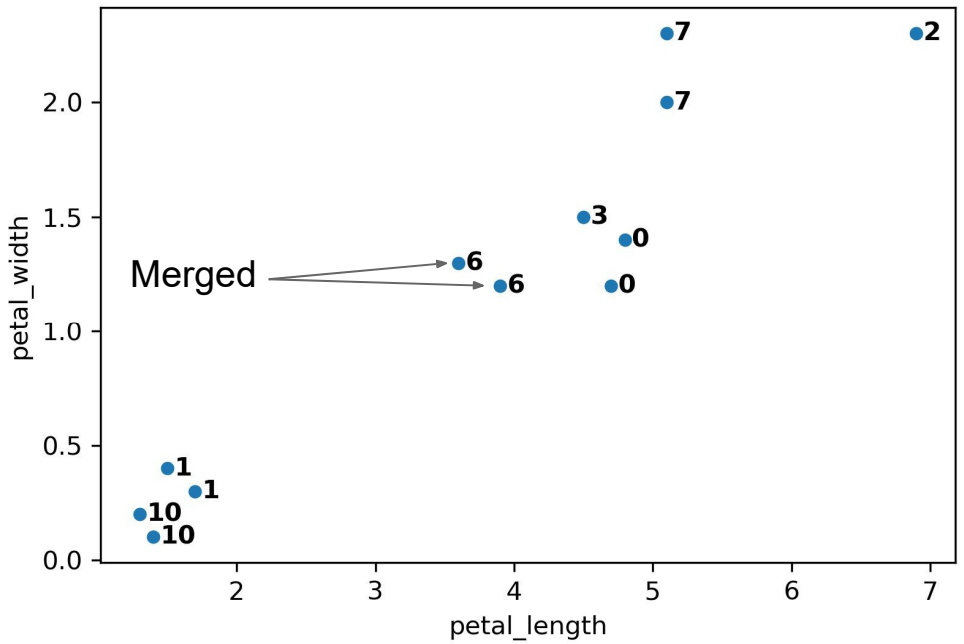
Agglomerative Clustering Example

Next closest are 6 and 8.



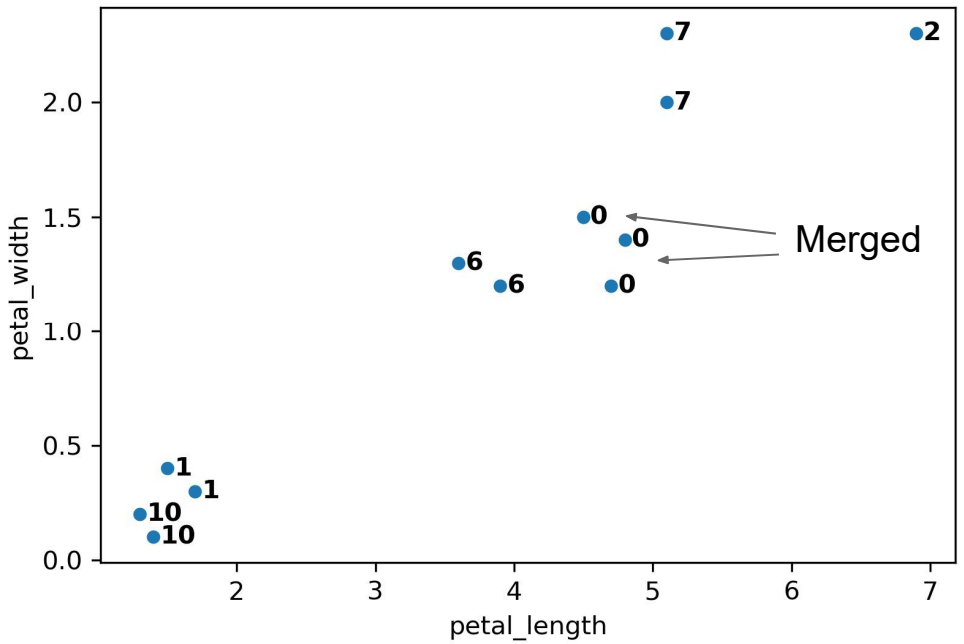
Agglomerative Clustering Example

Now 0 and 3 are closest. Merge them next.



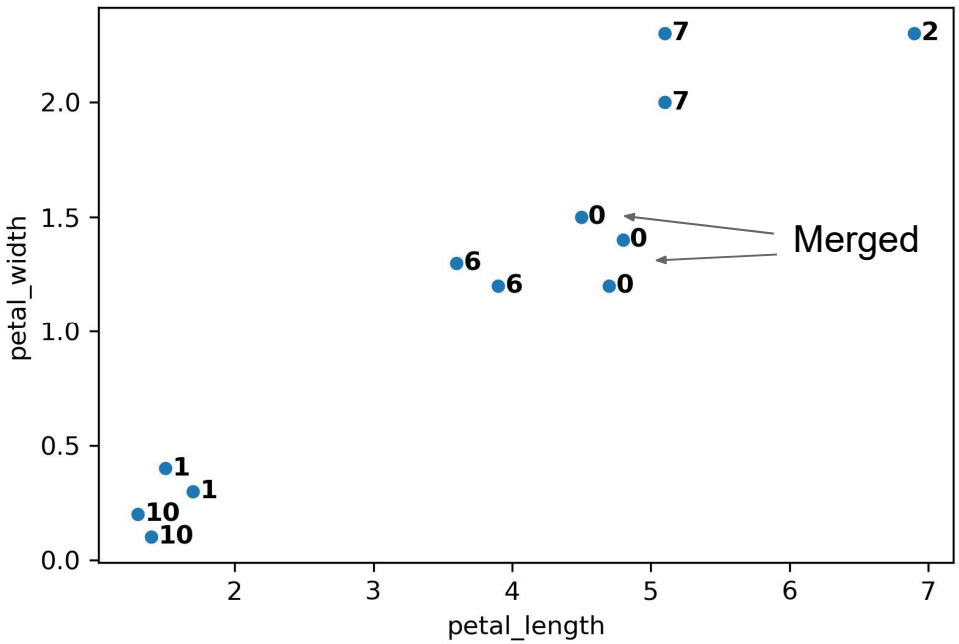
Agglomerative Clustering Example

Now 0 and 3 are closest. Merge them next.



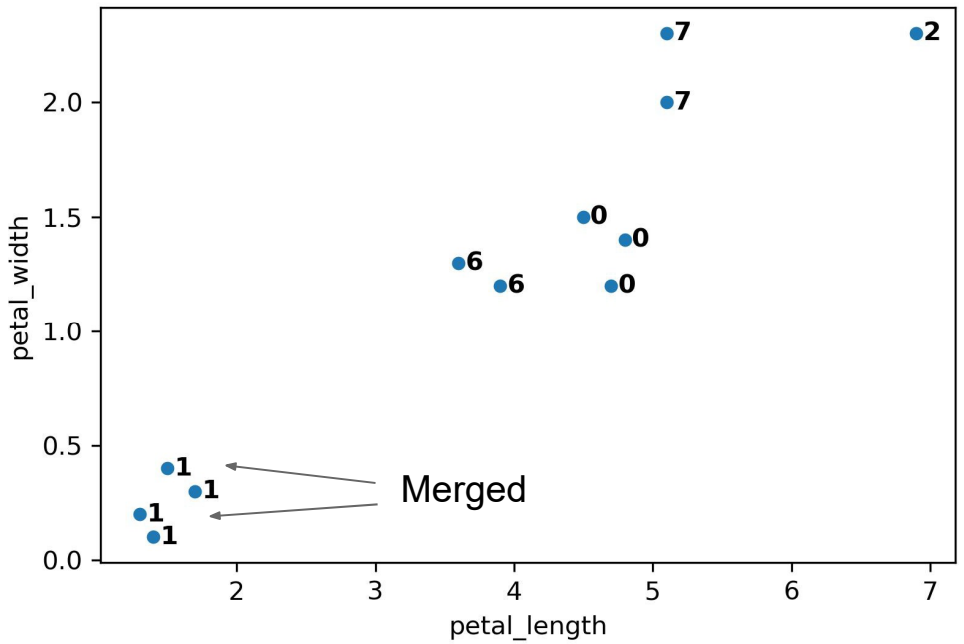
Agglomerative Clustering Example

Next up are 1 and 10.



Agglomerative Clustering Example

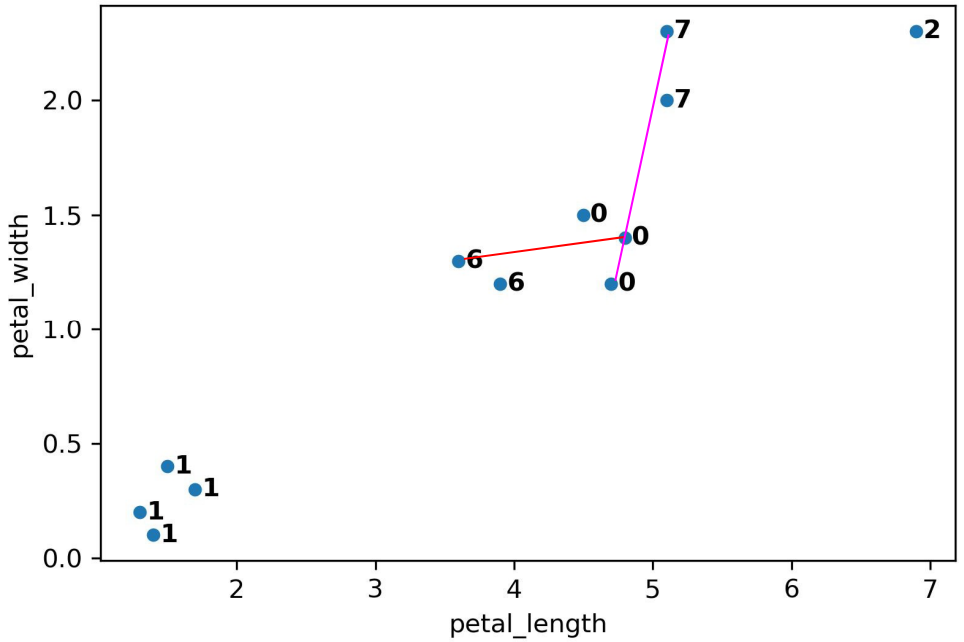
Next up are 1 and 10.



Agglomerative Clustering Example

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.



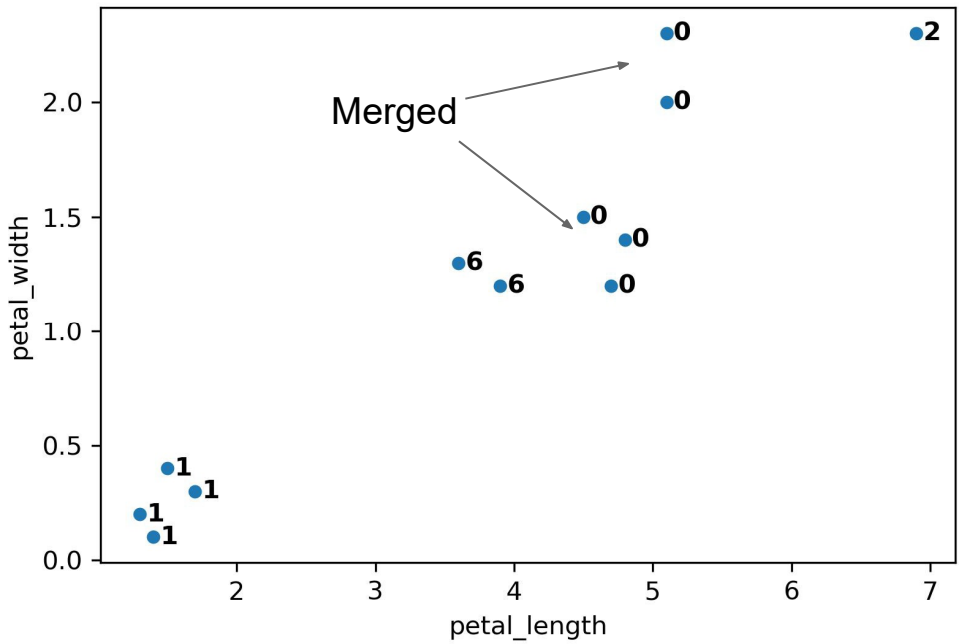
Purple line is shorter than red line.

Note: Doesn't look visually shorter due to different scales for x and y axes.

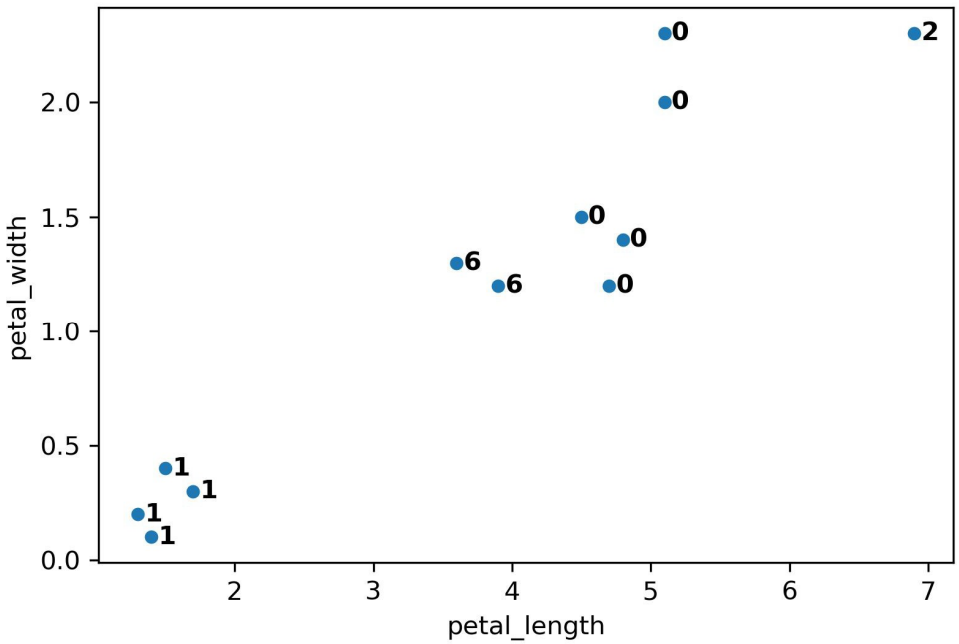
Agglomerative Clustering Example

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.

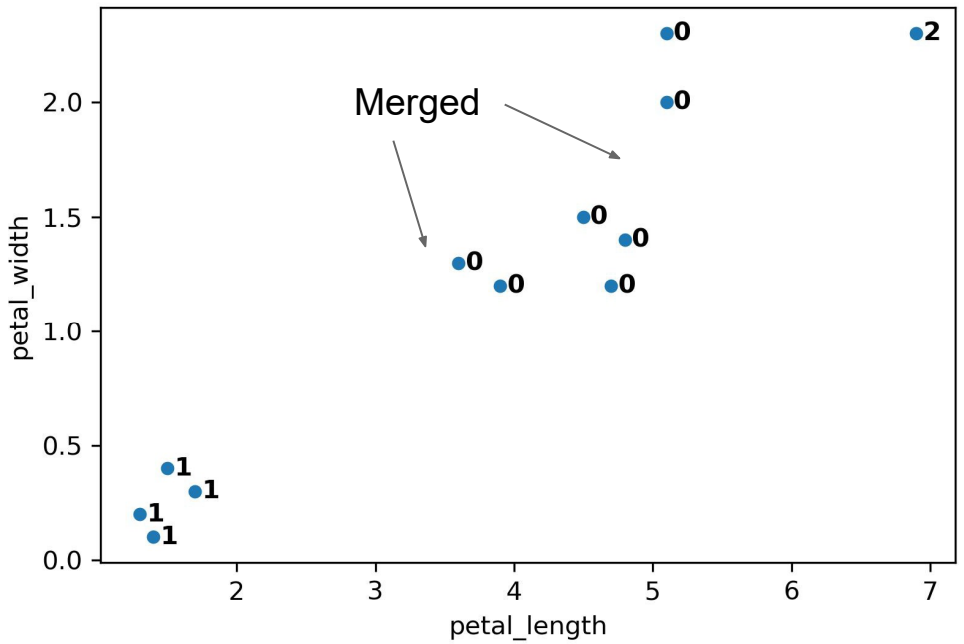


Next up are 0 and 6.

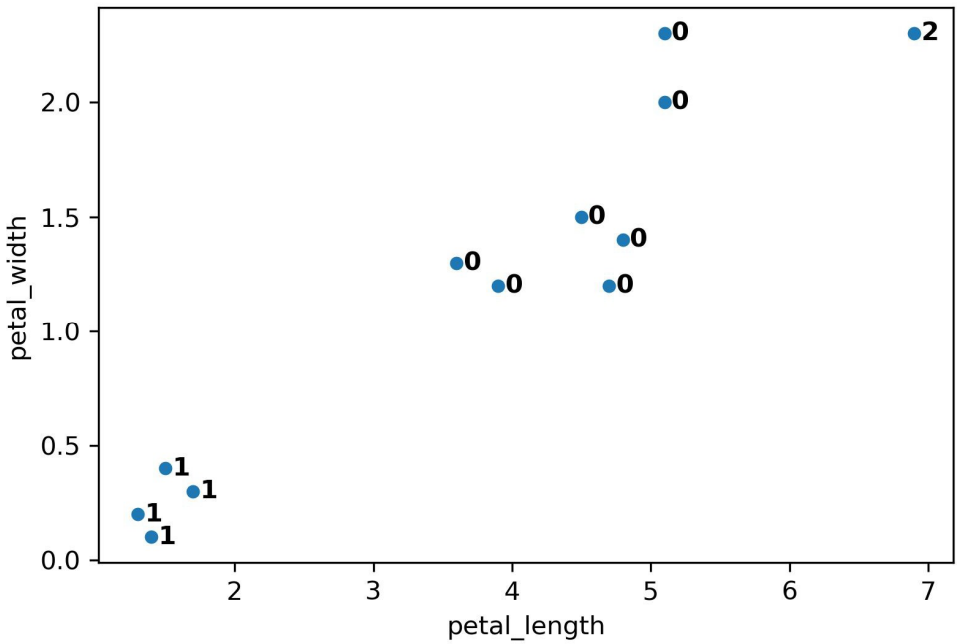


Agglomerative Clustering Example

Next up are 0 and 6.



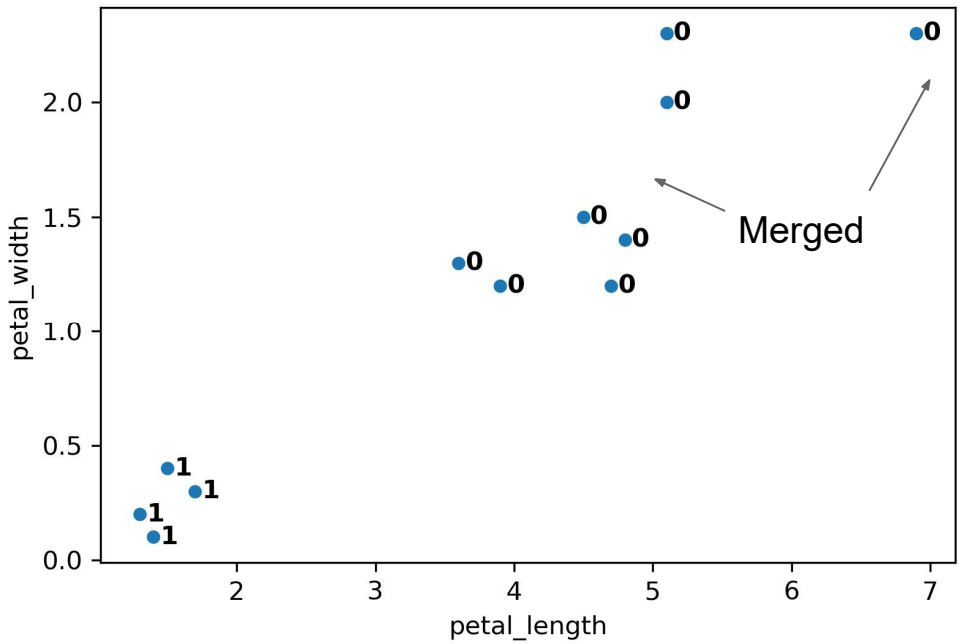
Next up are 0 and 2.



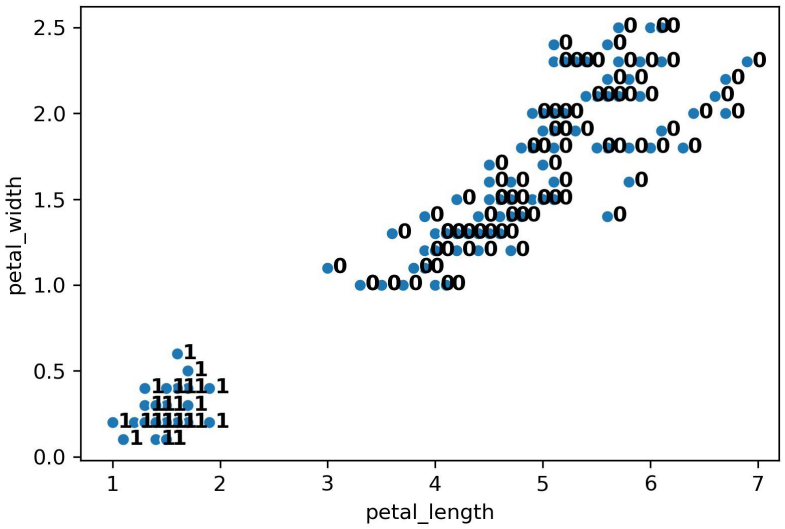
Agglomerative Clustering Example

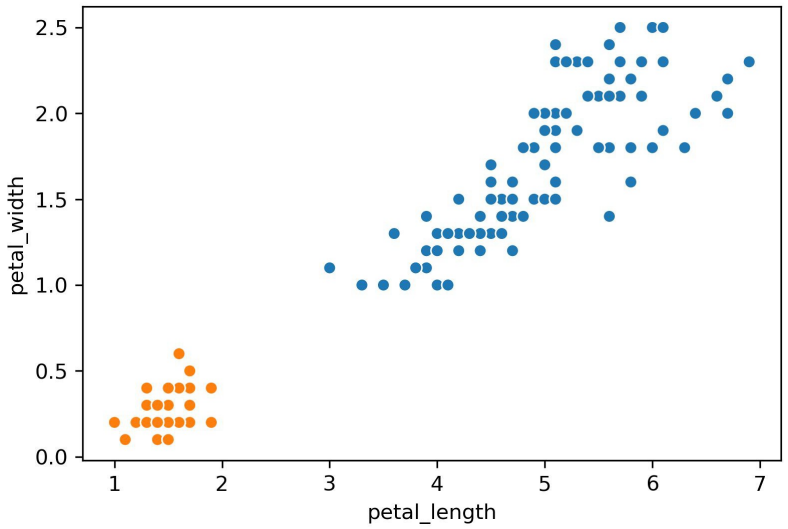
Next up are 0 and 2.

- At this point, we are done, because we only have two clusters left.



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

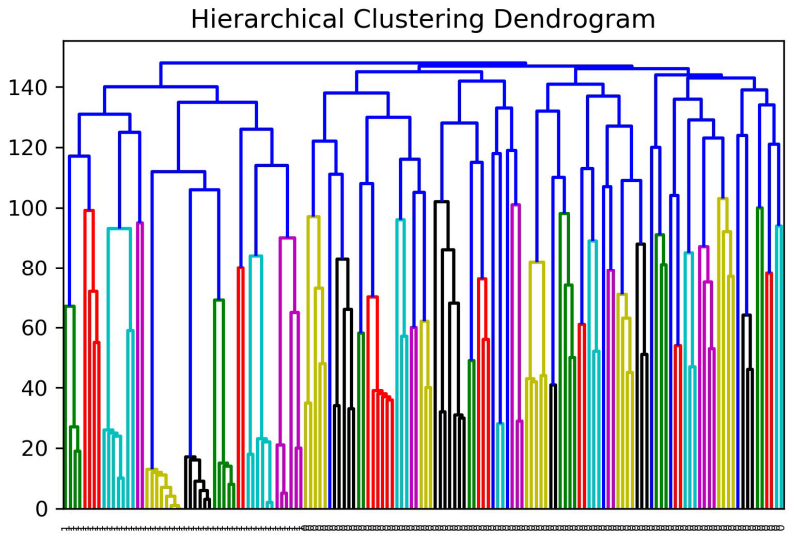




Clustering and Dendrograms

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”.
  - Won’t discuss any further, but you might see these in the wild.



Picking K

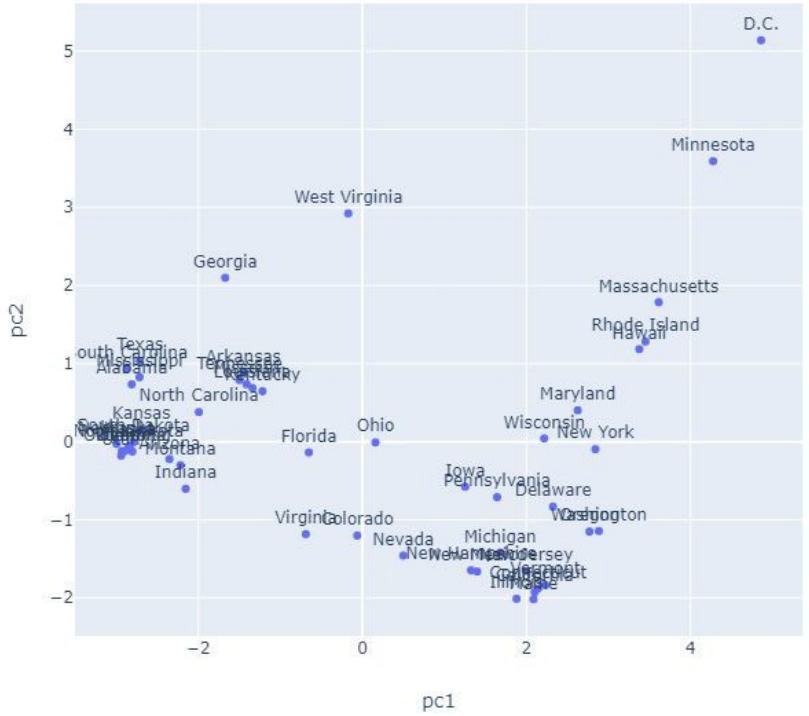
Picking K

The algorithms we've discussed today require us to pick a K before we start.

- But how do we pick K?

Often, best K is subjective.

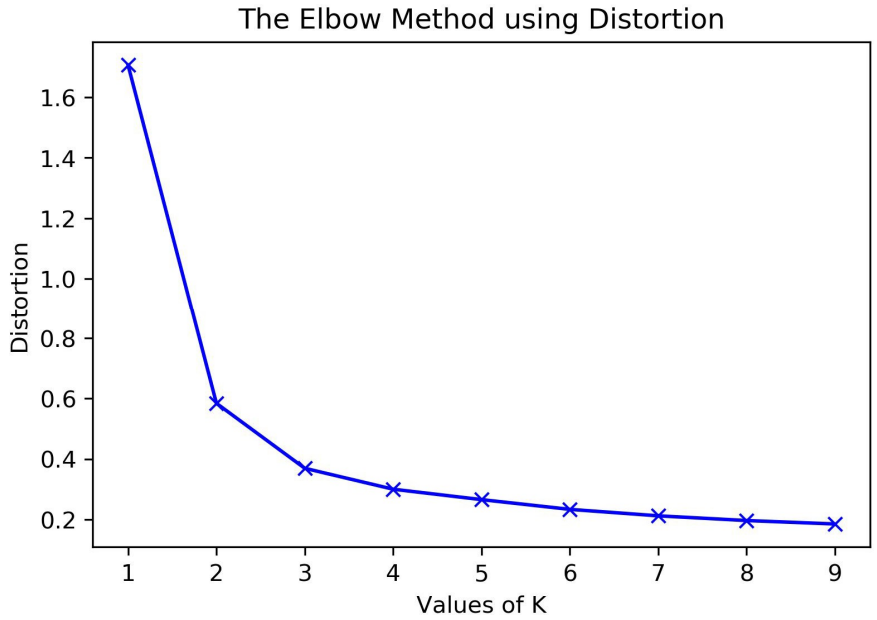
- How many clusters are there here?



Picking K: Elbow Method

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.



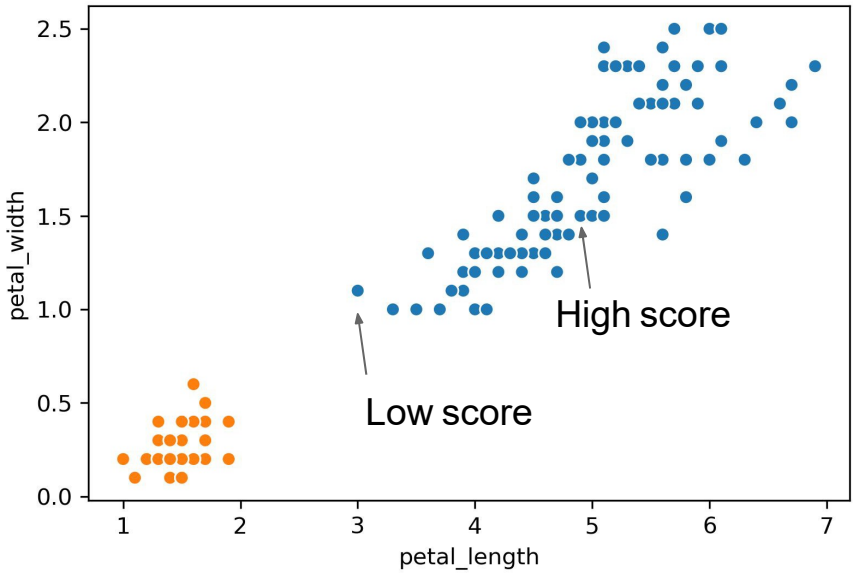
Silhouette Scores

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, score S is:

- A = average distance to other points in cluster.
- B = average distance to points in closest cluster.
- S = (B - A) / max(A, B)



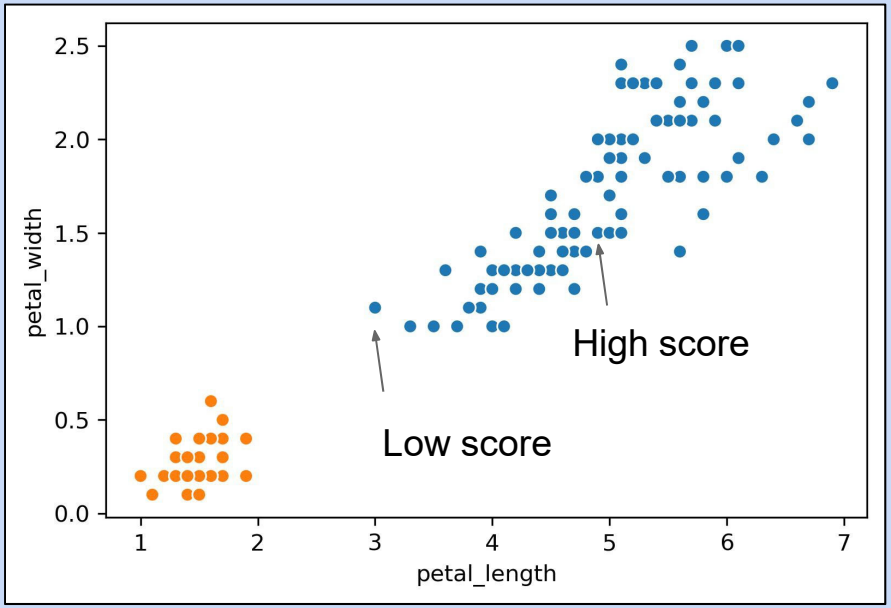
Silhouette Scores

For a data point X:

- A = average distance to other points in X's cluster.
- B = average distance to points in closest cluster.
- S = (B - A) / max(A, B)

What is the highest possible S?

- How can this happen?



Silhouette Scores

For a data point X:

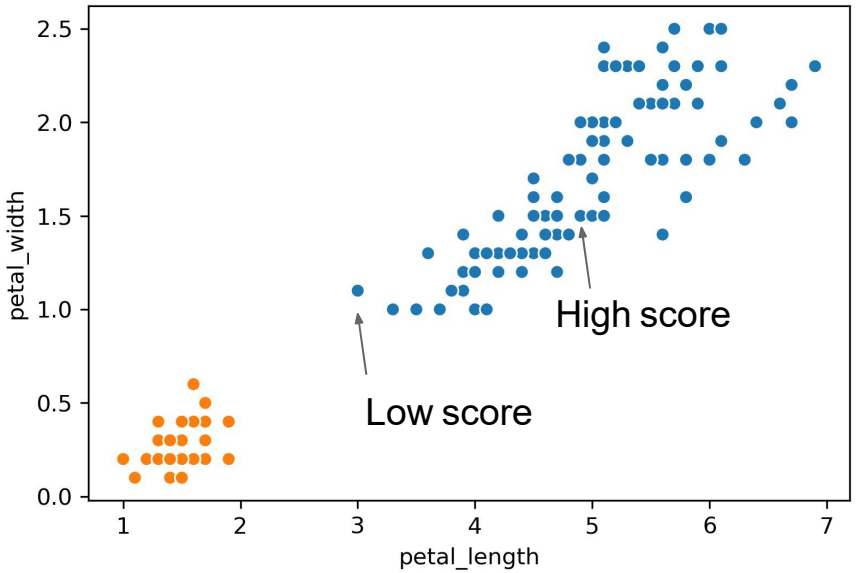
- A = average distance to other points in X's cluster.
- B = average distance to points in closest cluster.
- S = (B - A) / max(A, B)

What is the highest possible S?

- How can this happen?

Highest possible S is 1.

- This happens if every point in X's cluster is right on top of X.

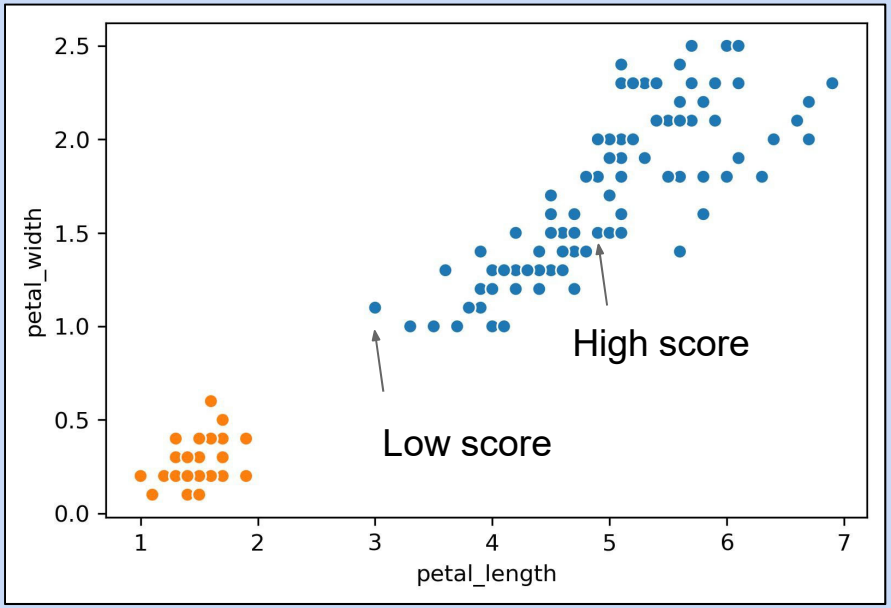


Silhouette Scores

For a data point X:

- A = average distance to other points in X's cluster.
- B = average distance to points in closest cluster.
- S = (B - A) / max(A, B)

Can S be negative?



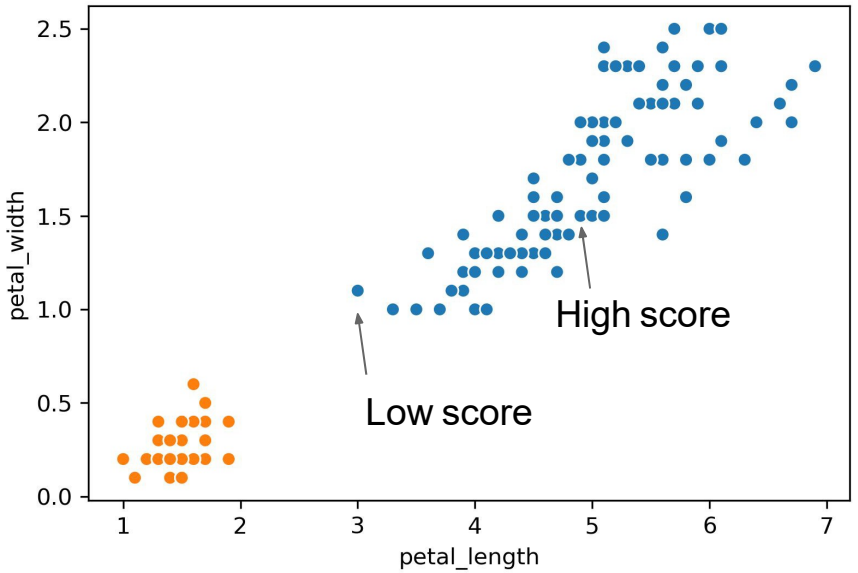
Silhouette Scores

For a data point X:

- A = average distance to other points in X's cluster.
- B = average distance to points in closest cluster.
- S = (B - A) / max(A, B)

Can S be negative?

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



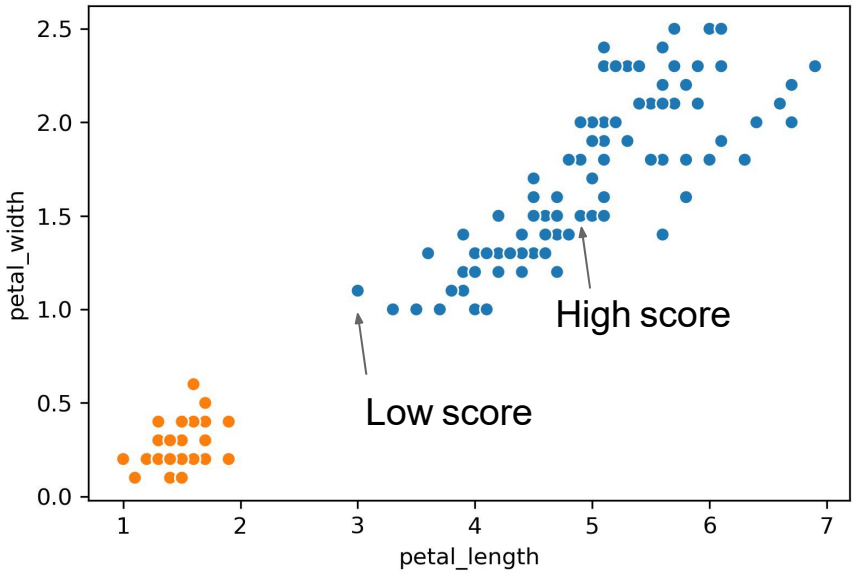
Silhouette Scores

For a data point X:

- A = average distance to other points in X's cluster.
- B = average distance to points in closest cluster.
- S = (B - A) / max(A, B)

Can S be negative?

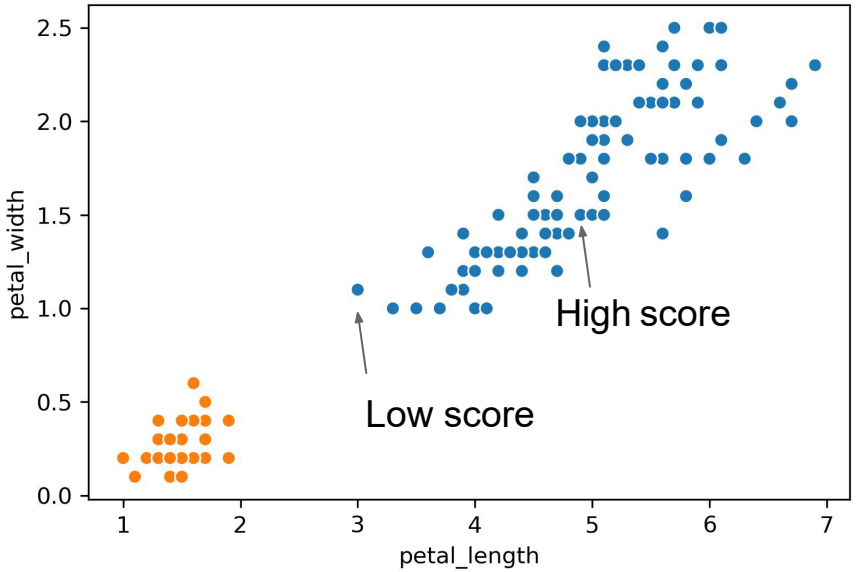
- Yes. Average distance to X's clustermates is larger than distance to the closest cluster.
- Example: The "low score" point on the right has S = -0.13.



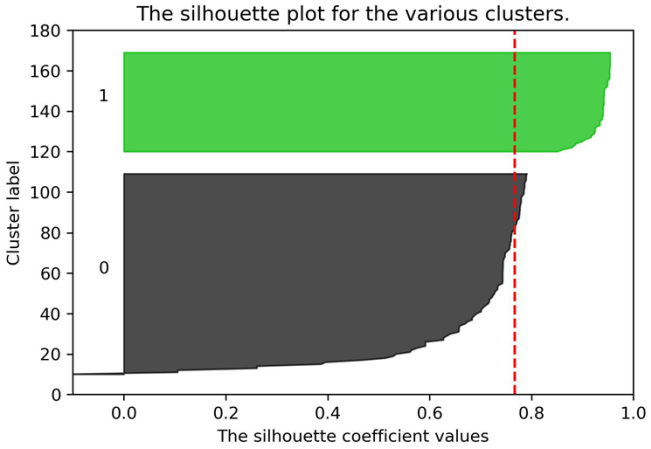
Silhouette Plot

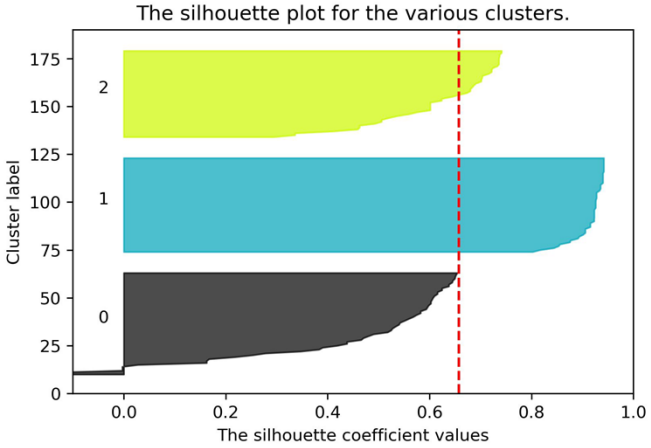
We can plot the Silhouette Scores for all of our data points.

- Points with large silhouette widths are deeply embedded in their cluster.
- Red dotted line shows the average.

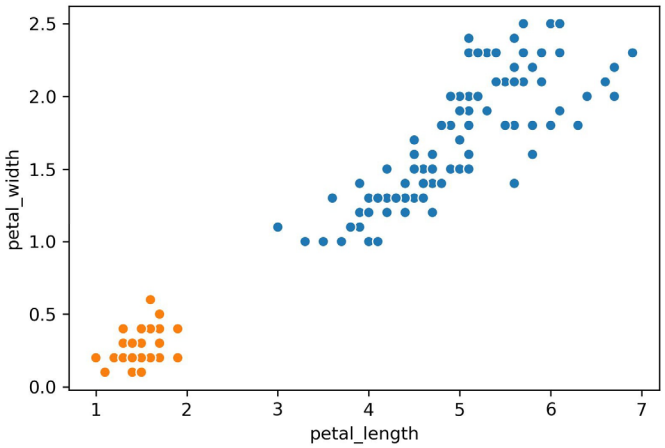


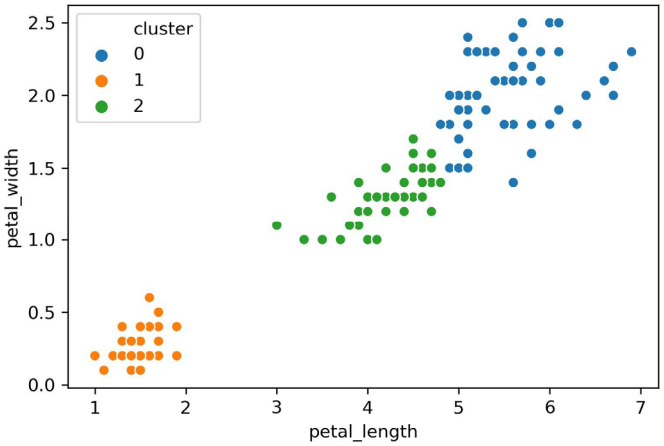
Silhouette Scores, K = 2 vs. K = 3





Average silhouette score is lower.





Picking K: Real World Metrics

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.