TASK

# Clustering II

Visit our website

# Introduction

## WELCOME TO THE SECOND CLUSTERING TASK!

In this task, we describe bottom-up or agglomerative clustering. This is a type of hierarchical clustering, a clustering approach which does not require a commitment to a particular choice of K.

Get in touch

**Connect for support**

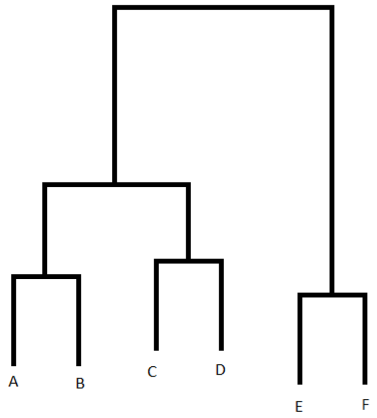
Remember that with our courses, you’re not alone! You can contact an expert code reviewer to get support on any aspect of your course.

The best way to get help is to login to Discord at **https://discord.com/invite/hyperdev** where our specialist team is ready to support you.

Our team is happy to offer you support that is tailored to your individual career or education needs. Do not hesitate to ask a question or for additional support!

## AGGLOMERATIVE HIERARCHICAL CLUSTERING

Hierarchical clustering is an algorithm that builds, as the name suggests, a hierarchy of clusters, which form a tree-like structure called a “dendrogram”, as in the image below.



Initially, each individual data point is assigned to a mini-cluster of its own (at the “leaves” of the dendrogram, A-F) and a measure of dissimilarity (such as Euclidean distance between their features) is defined. Using this dissimilarity metric, the algorithm consists of iteratively merging the most similar two clusters into one larger cluster, starting from the leaves and working upwards until all observations belong to a single cluster. The measure of the dissimilarity determines how far up the dendrogram the merge takes place - greater dissimilarity places the merge higher up the dendrogram. This is called agglomerative clustering because the process is a bottom-up process, and it is called hierarchical because clusters are nested within each other. The dendrogram can then be “cut” at a particular height to obtain the clusters - this is discussed in greater detail later.

## DISSIMILARITY BETWEEN GROUPS OF DATA POINTS

It is relatively simple to apply this algorithm when there is only a single sample in each cluster, but it is also necessary to merge groups of samples, such as the merge of the cluster containing A and B with that containing C and D. The simple measure of dissimilarity (Euclidean distance) discussed earlier is insufficient for this and needs to be expanded as a measure of dissimilarity between clusters, not just single samples. This can be achieved through the introduction of a **linkage criterion**. Similarly to how we used the cluster mean in K-means as the point from which we measured distance, the linkage criterion defines the point from which the distance between clusters should be calculated.

The following are three simple linkage criteria:

1. **Single linkage**
   The dissimilarity metric is calculated between all samples in the one cluster with all samples in the other cluster, and the smallest dissimilarity is chosen. This is called a "pairwise" comparison.
2. **Complete linkage**
   The dissimilarity metric is calculated between all samples in the one cluster with all samples in the other cluster, and the largest dissimilarity is chosen.
3. **Average linkage**
   The dissimilarity metric is calculated between all samples in the one cluster with all samples in the other cluster, and the average dissimilarity is chosen.

Scikit-learn's hierarchical clustering method also comes with an implementation of Ward's method. This more sophisticated measure computes the sum of squares of all possible cluster merges and chooses the optimum merge accordingly. The choice of dissimilarity measure is very important, as it has a strong effect on the resulting dendrogram.  Average, complete, and single linkage are most popular among statisticians. Average and complete linkages are generally preferred over single linkage, as they tend to yield more balanced dendrograms.

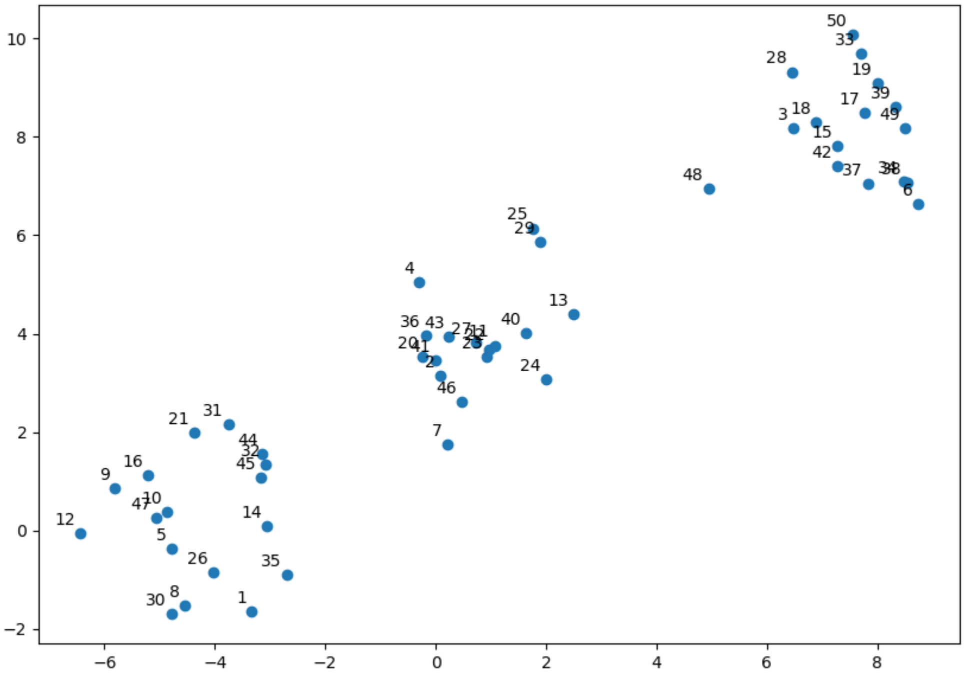
## OBTAINING CLUSTERS FROM A DENDOGRAM

Agglomerative clustering will place all data points in a single final cluster unless the process is stopped early. This means we need to tell the algorithm when to stop, just like with K-means clustering. But unlike with K-means clustering, we have an extra tool at our disposal: a graph that shows us what a range of clusters would look like.

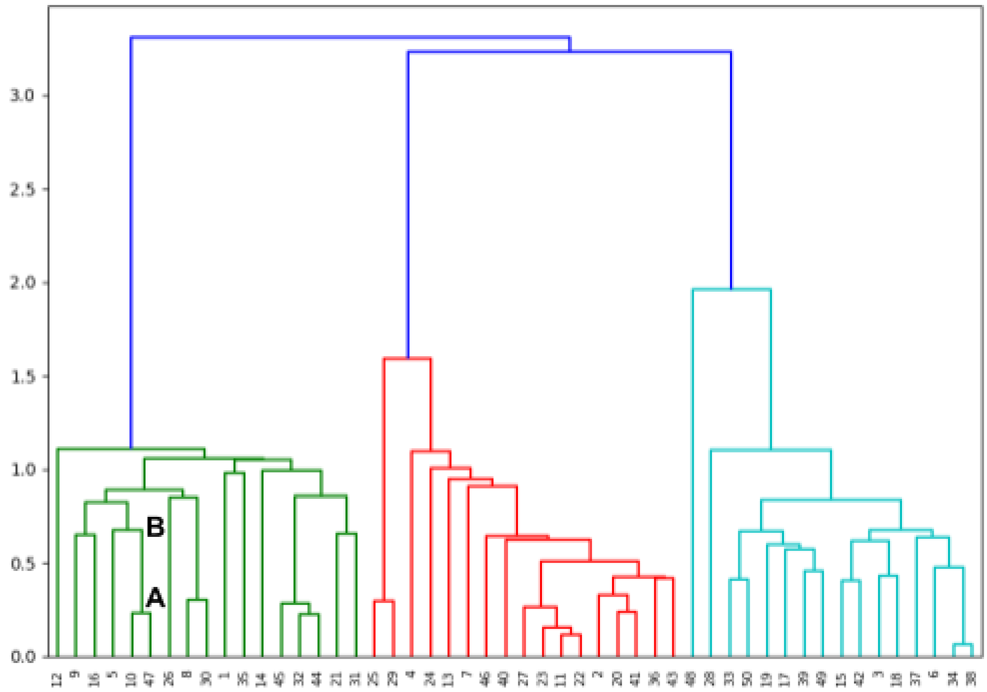
A dendrogram contains two kinds of information. Firstly, it shows each merge that was performed. Secondly, the length of the vertical lines show the distances (Euclidean distance or whatever distance measure was specified) between the merged clusters. An important thing to note: larger distances between clusters means that those clusters are more **dissimilar**.

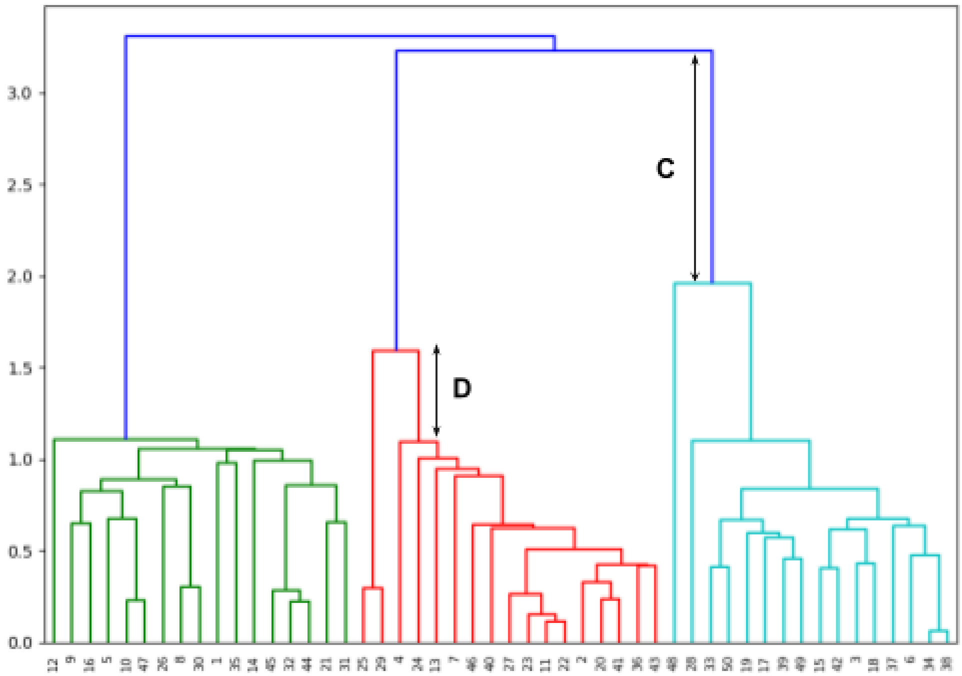
In the following example, datapoints 10 and 47 are merged and have very short vertical lines (A), which means they are similar. Pairs (9, 16) and (5, (10, 47)) are fairly dissimilar from each other internally, but their averages are fairly similar, so their merge has fairly short lines (C).

We can use the dendrogram to identify a clustering in which all clusters have low within-cluster variance, by looking at which merges would merge highly dissimilar clusters. Merges between dissimilar clusters (clusters that are further away from each other) look like lines which stretch out high and then merge at the top. To cut the process off before such clusters can be merged, we specify a minimum distance between clusters. A good minimum distance can be determined by finding the section of the y-axis that corresponds to the largest section of the graph with no horizontal line in it. For illustration, see sections C and D, of which C is clearly the largest.



In the graph, you can see three well-defined clusters. Logically speaking, we know that we're looking for three clusters in the dendrogram. With this in mind, let's generate some dendrograms.





Recall that we are looking for the most *dissimilar* clusters. The larger the distance between clusters, the more dissimilarity there is. The y axis on the dendrogram represents distance. It follows that the blue lines represent our desired clusters. Because there are three blue lines, it makes sense that there are three clusters.

# Instructions

Read the Jupyter notebook in this task's folder before trying the compulsory task.

# Compulsory Task

- Load the Iris data set. Select two features from the data to use in this exercise and scale the data.
- Using single and complete linkages, and Euclidean and Cityblock distance metrics, print dendrograms for the different combinations of these. You should have 4 dendrograms. Different distance metrics are listed **here**.
- Pick one dendrogram to go forward with.
    - Choose a fixed number of clusters based on the dendrogram of your choice. These should be the most dissimilar clusters.
    - Run agglomerative hierarchical clustering with that number of clusters (and the linkage method and distance metric used for that dendrogram).
    - Verify the clusters you obtained by using the Silhouette score and comment on your confidence in your clustering solution.

If you are having any difficulties, please feel free to contact our specialist team **on Discord** for support.

## Things to look out for:

1. Make sure that you have installed and set up all programs correctly. You have set up **Dropbox** correctly if you are reading this, but **VS Code** or **Anaconda** may not be installed correctly.
2. If you are not using Windows, please ask a reviewer for alternative instructions.

## Completed the task(s)?

Ask an expert to review your work!

**Review work**

Rate us

## Share your thoughts

HyperionDev strives to provide internationally-excellent course content that helps you achieve your learning outcomes.

Think that the content of this task, or this course as a whole, can be improved, or think we’ve done a good job?

**Click here** to share your thoughts anonymously.