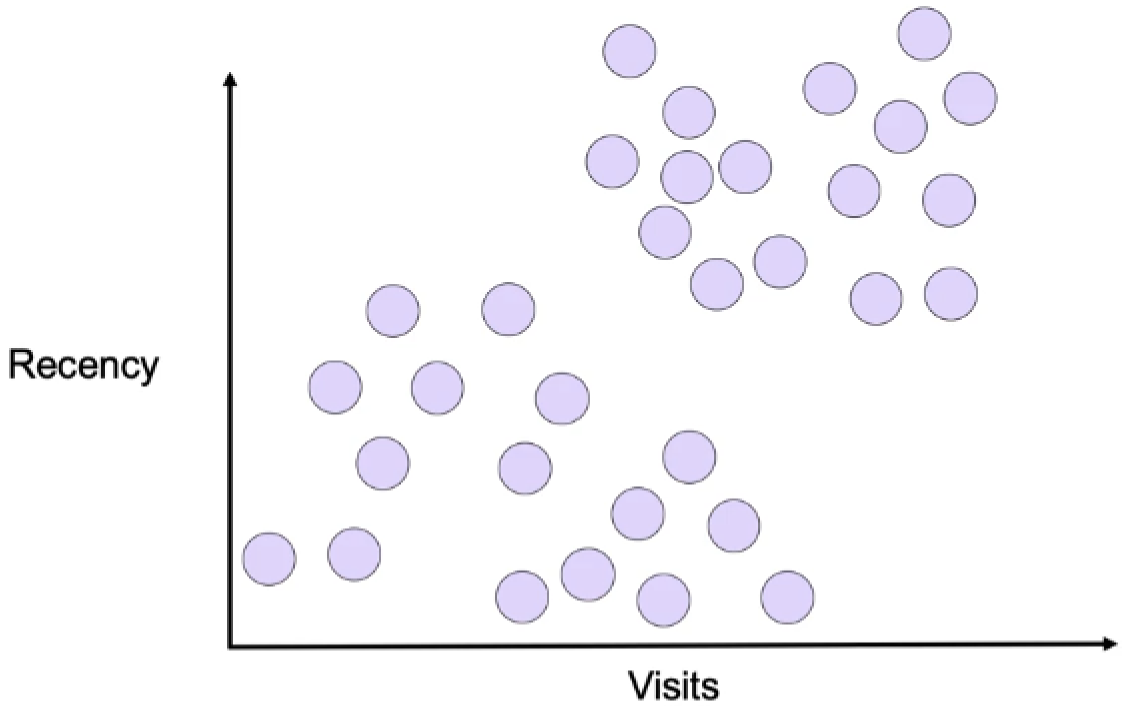
10) HAC

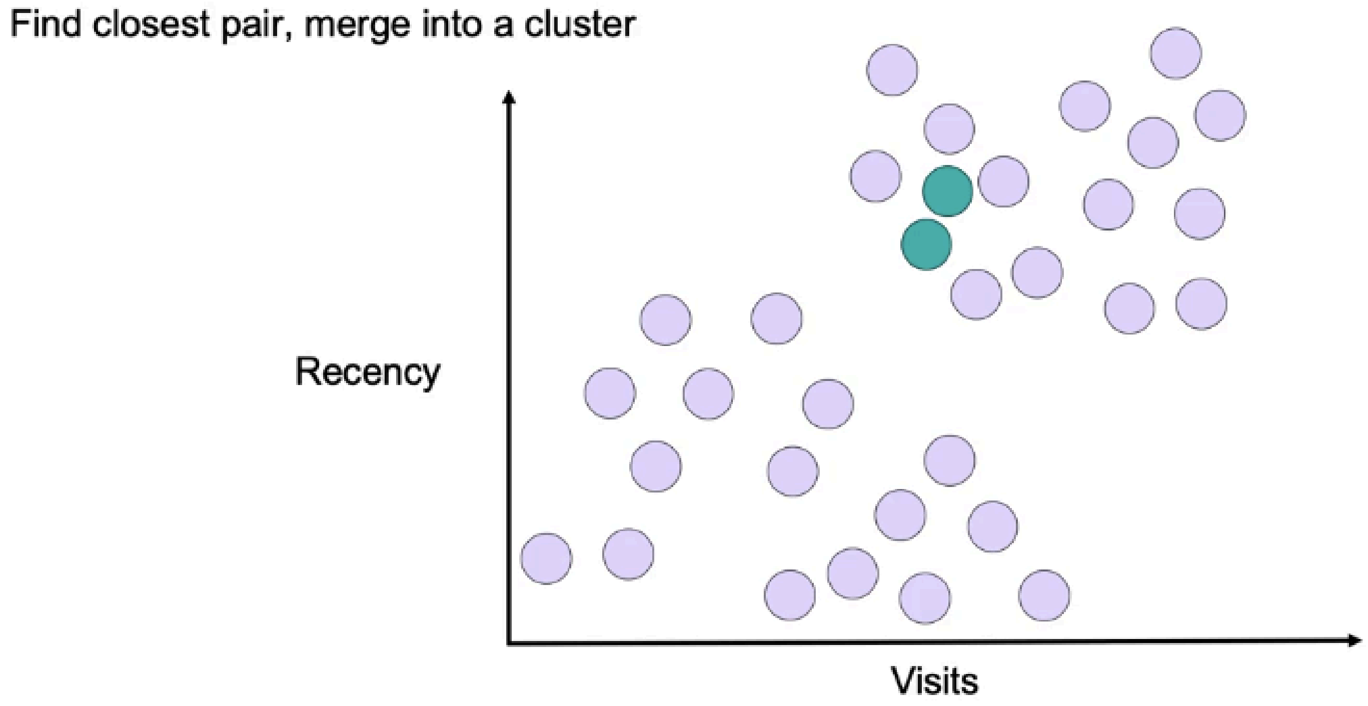
- Stands for Hierarchal Agglomerative Clustering

How it works

Suppose we have the following data:

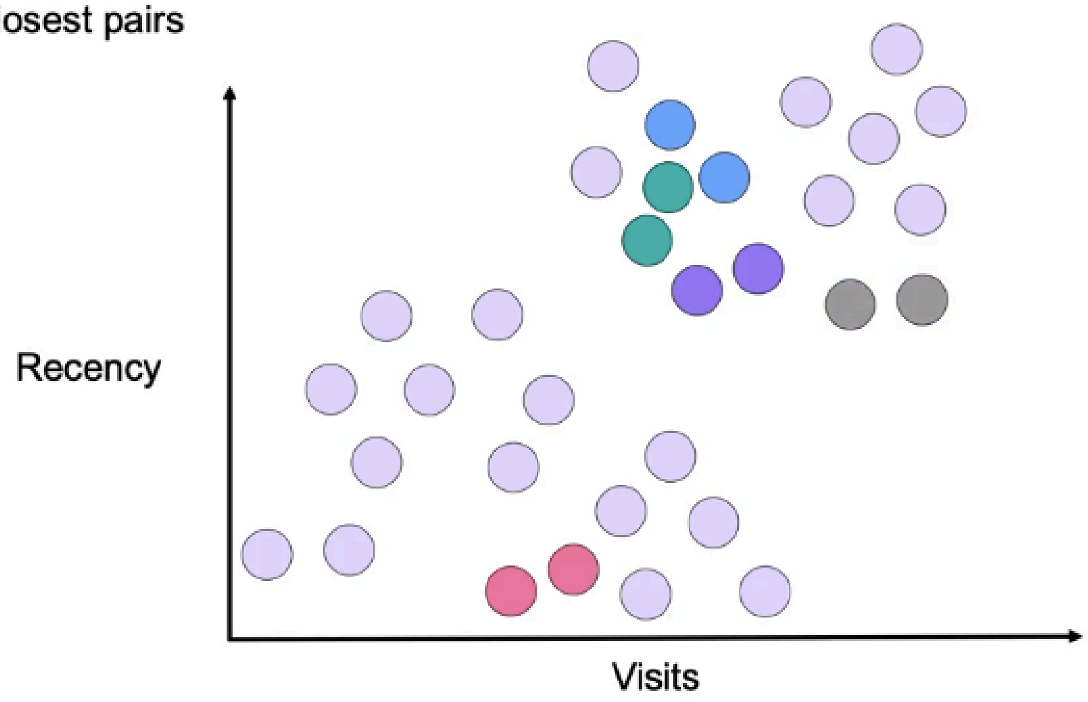


What we do is we merge the closest 2 points in a cluster like so:

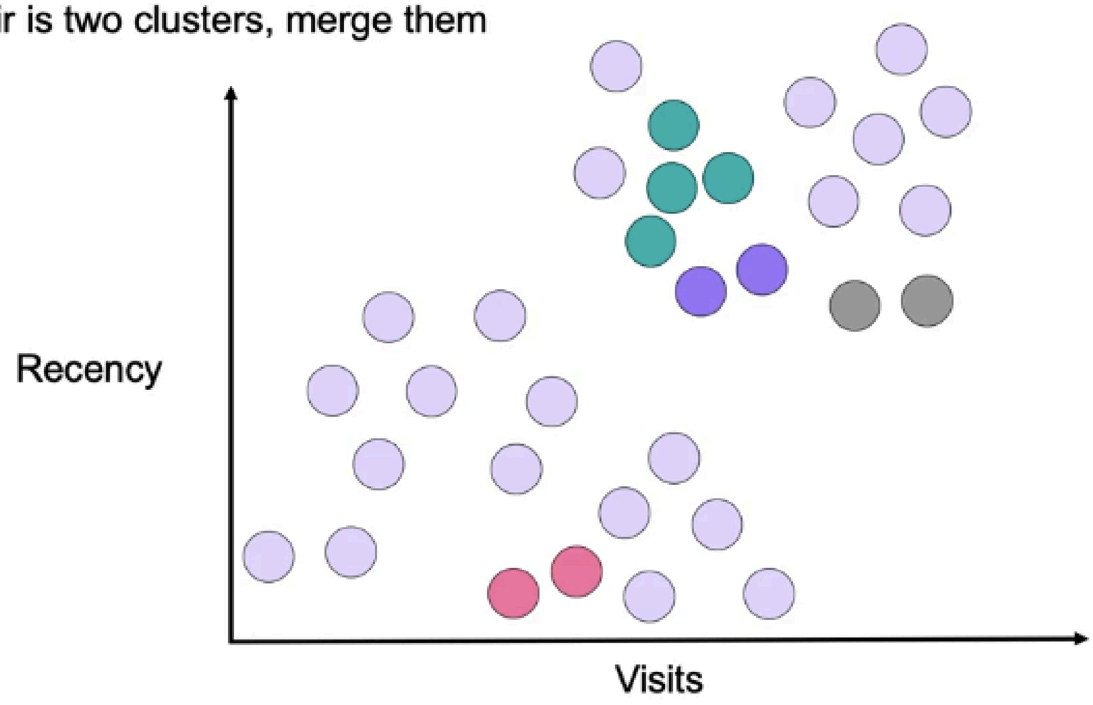


And we keep doing that until we reach the desired number of clusters (or some other criteria). We also have the ability to merge clusters like so:

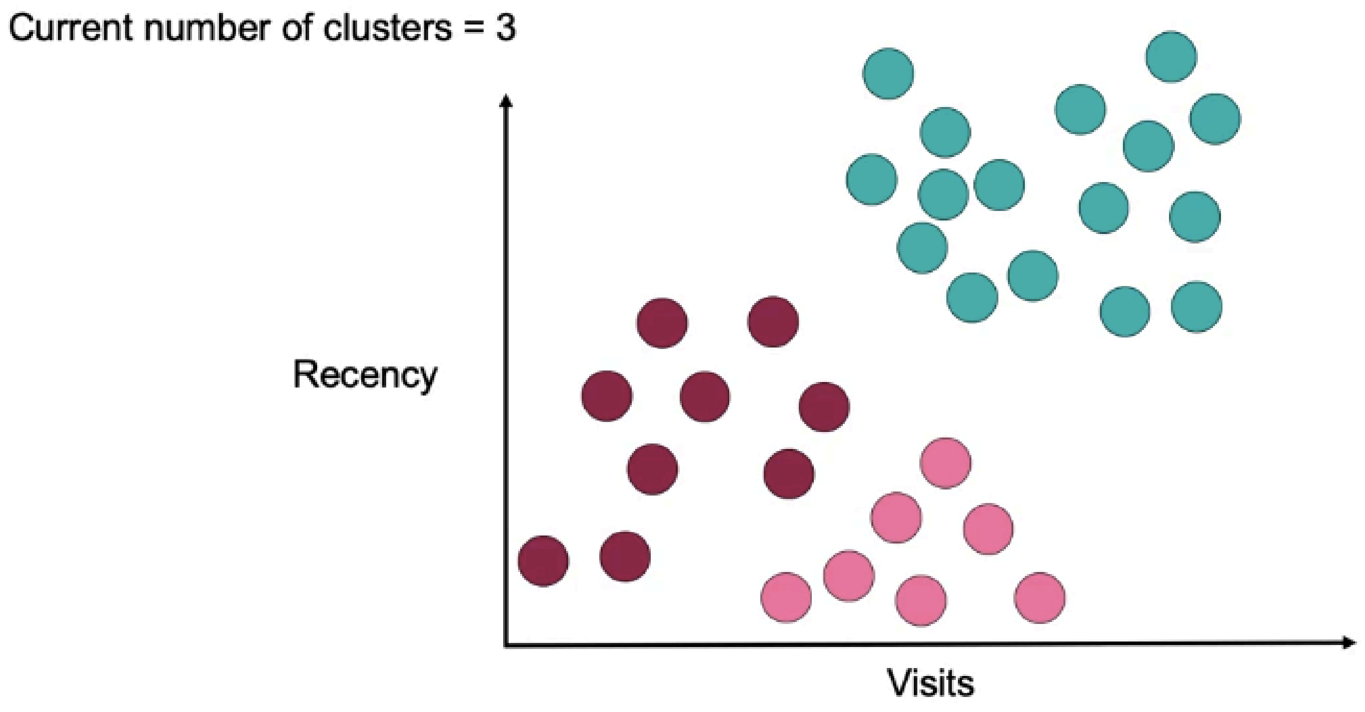
Keep merging closest pairs



If the closest pair is two clusters, merge them



So ultimately, if we had the number of clusters = 3, we will end up with the following clusters:

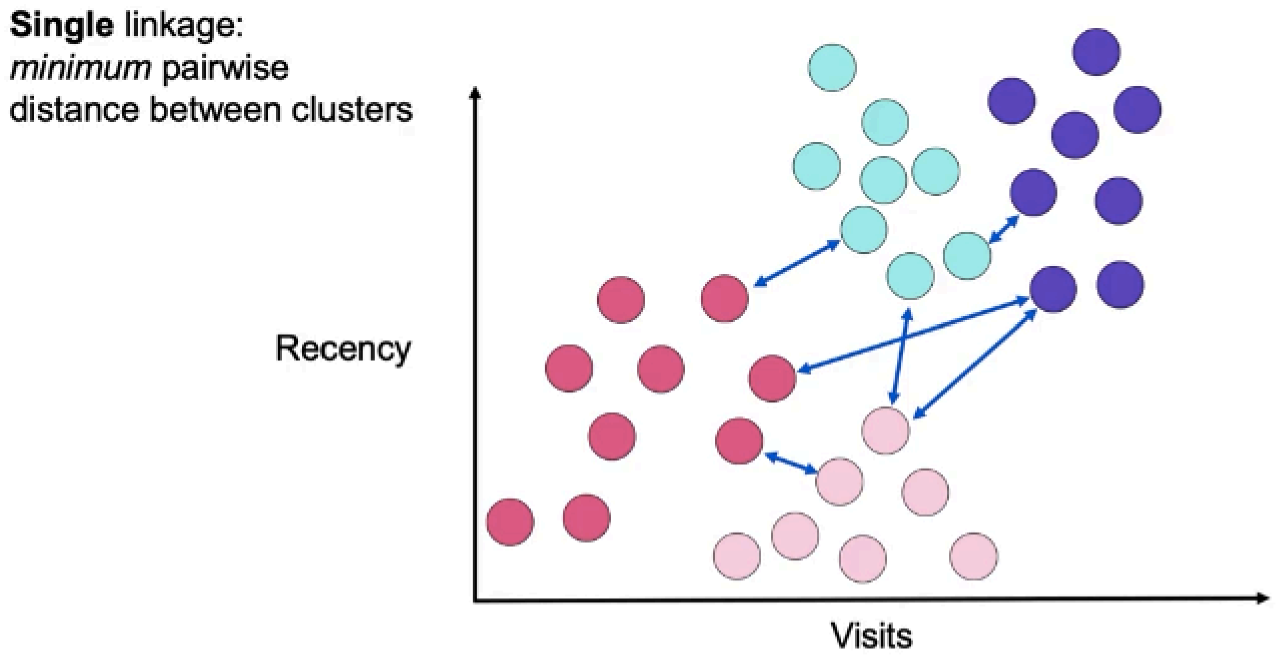


Determining distance b/w clusters

There are various ways to measure distance b/w clusters. These different ways/methods are called the various "linkage types"

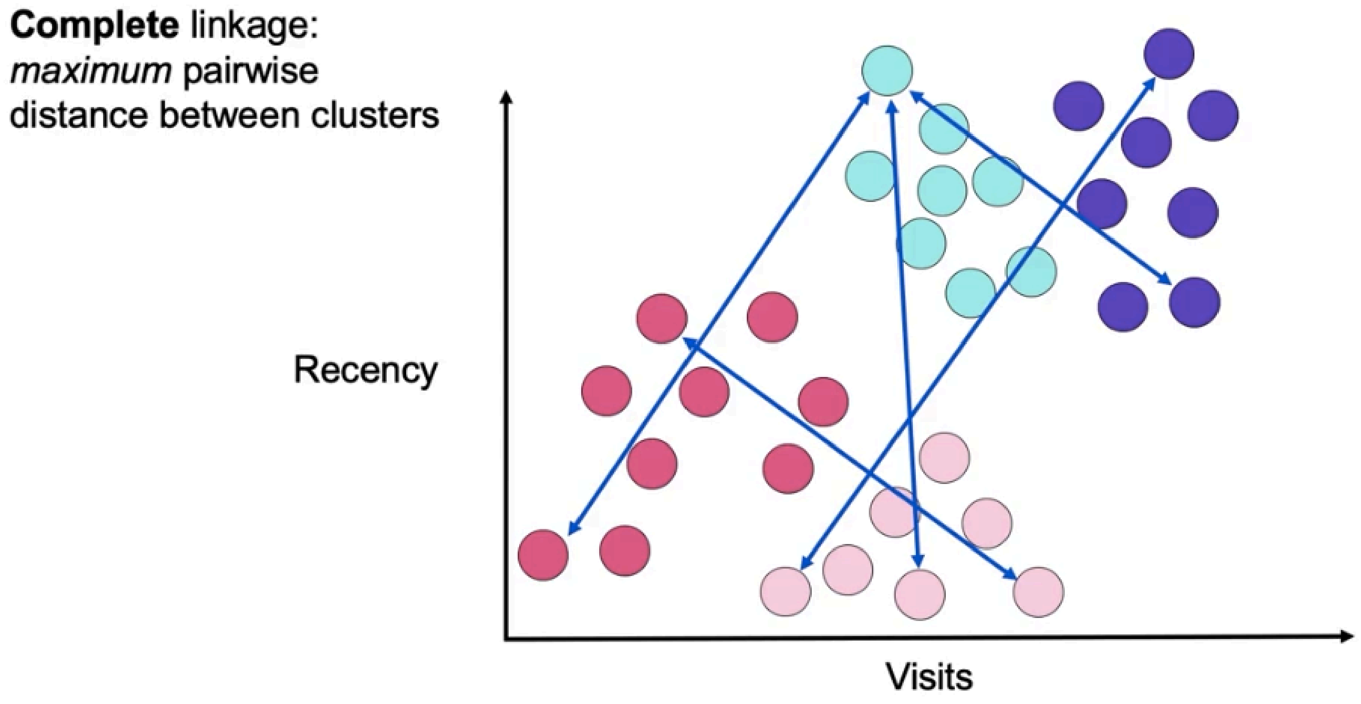
1) Single linkage

This will be the distance between the two closest points of the clusters. A pro is that it can help in ensuring a clear separation of the clusters that have any points within certain distances of one another, so that it can have clear boundaries. A con is that it would not be able to separate out cleanly if theres some noise between the two different clusters, meaning it would be easily skewed by certain outliers falling close to certain clusters



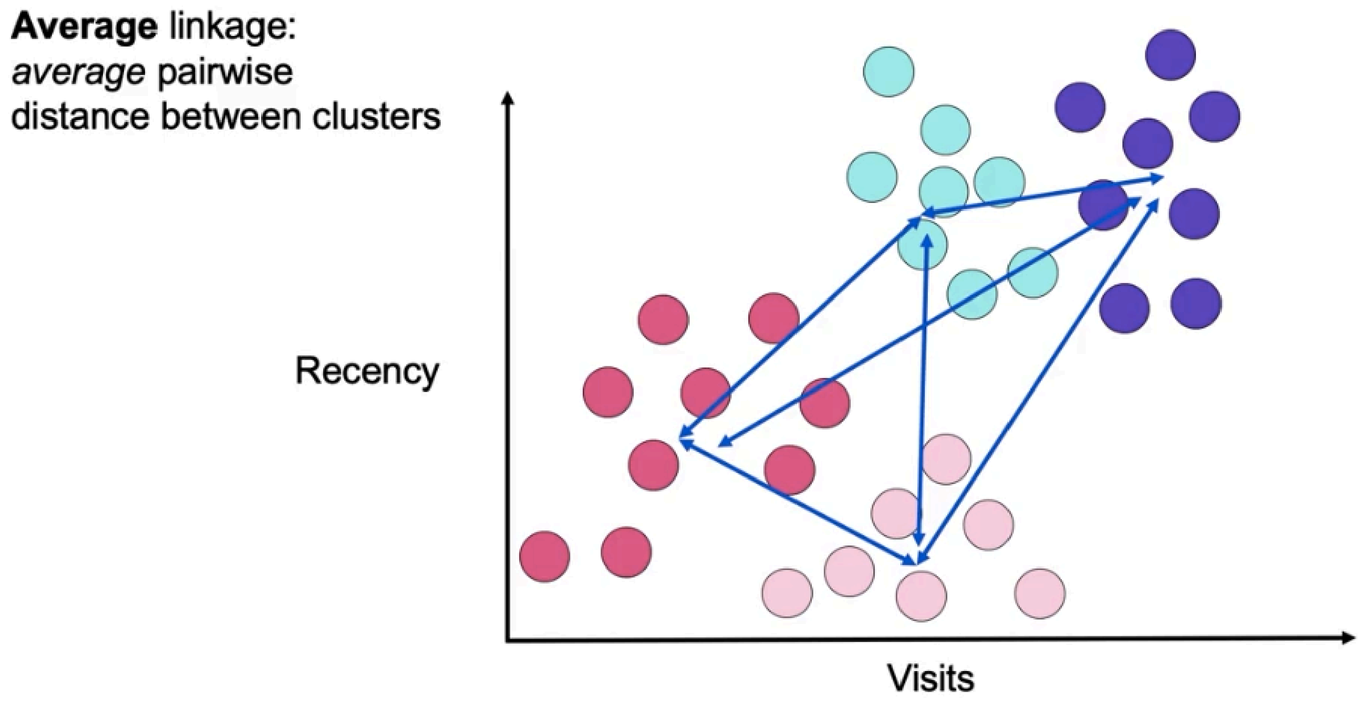
2) Complete linkage

Similar to single linkage, but this will take the maximum distance instead of the minimum distance, ie the distance between the two farthest points. A pro of this method is that it will do a much better job of separating out the clusters if theres a bit of noise or overlapping points of the two clusters, unlike single linkage. But a con of this method is that it can tend to break apart larger existing clusters dependent on where that maximum distance of the different points may end up lying



3) Average linkage

Or we can just take the averages of the clusters (sort of like centroids) and measure the distance between them. It is sort of a neutral method when compared to single and complete linkage, meaning it has both their pros and cons but to a less extend



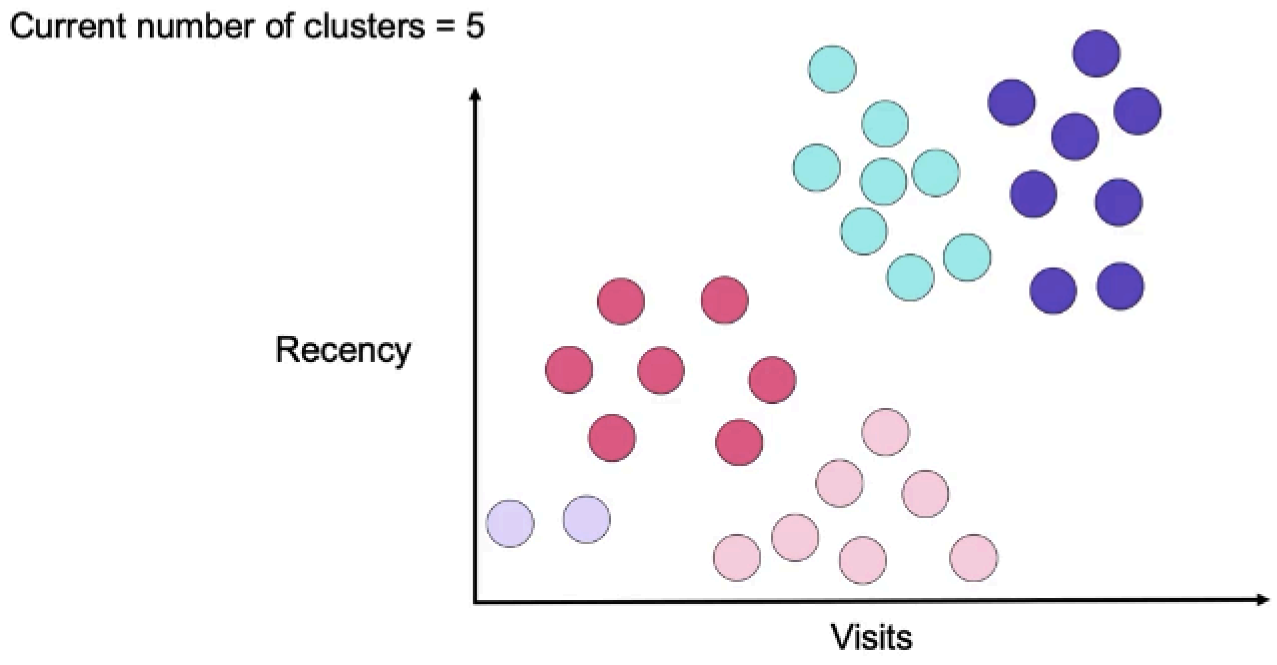
4) Ward linkage

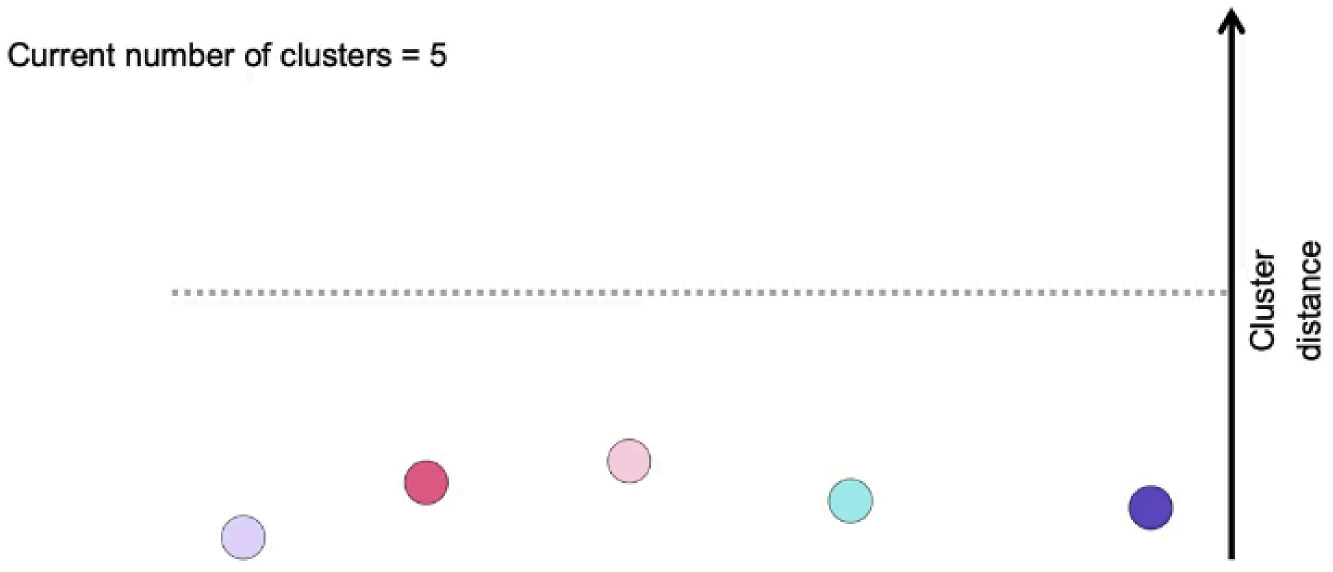
Marge based on inertia. So when merging, it will merge the clusters which would minimise that inertia value. (Remember, intertia is the sum of squared distances from each point (x_i) to its cluster's centroid (C_k))

Using cluster distance to stop

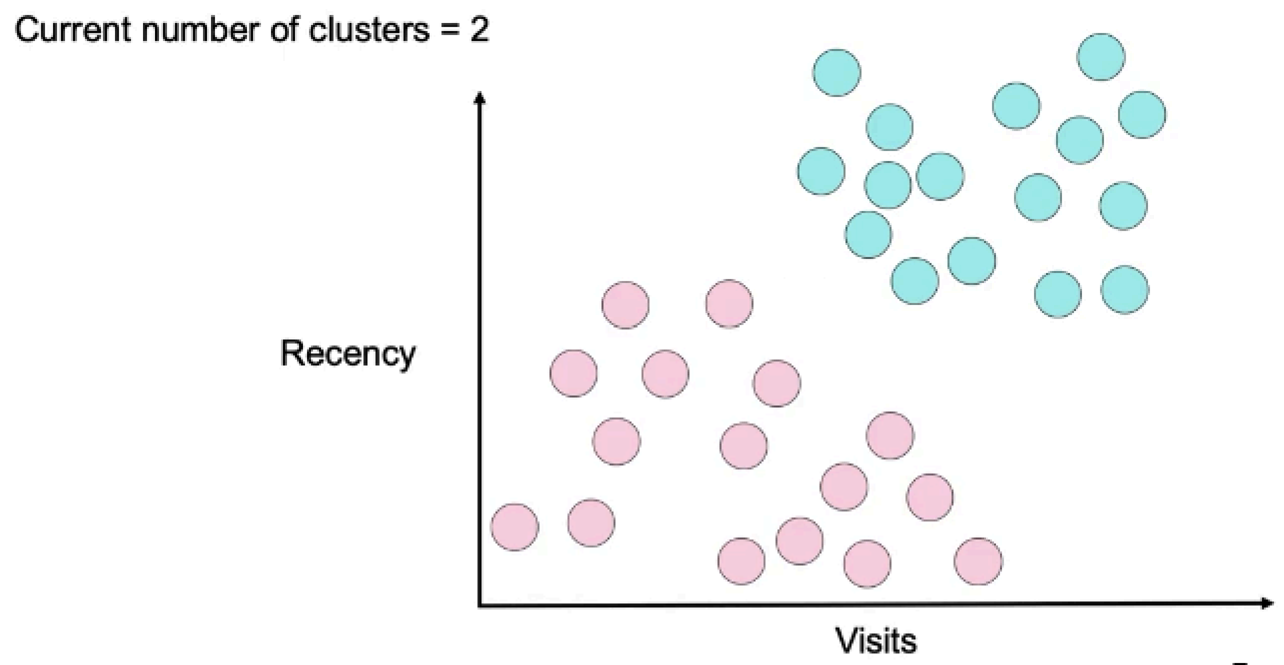
Rather than using number of clusters as a criteria to stop, we can use the cluster distance as a criteria to stop. We can set a threshold and keep pairing points/clusters until we can make sure that the distance b/w all the clusters is atleast that amount

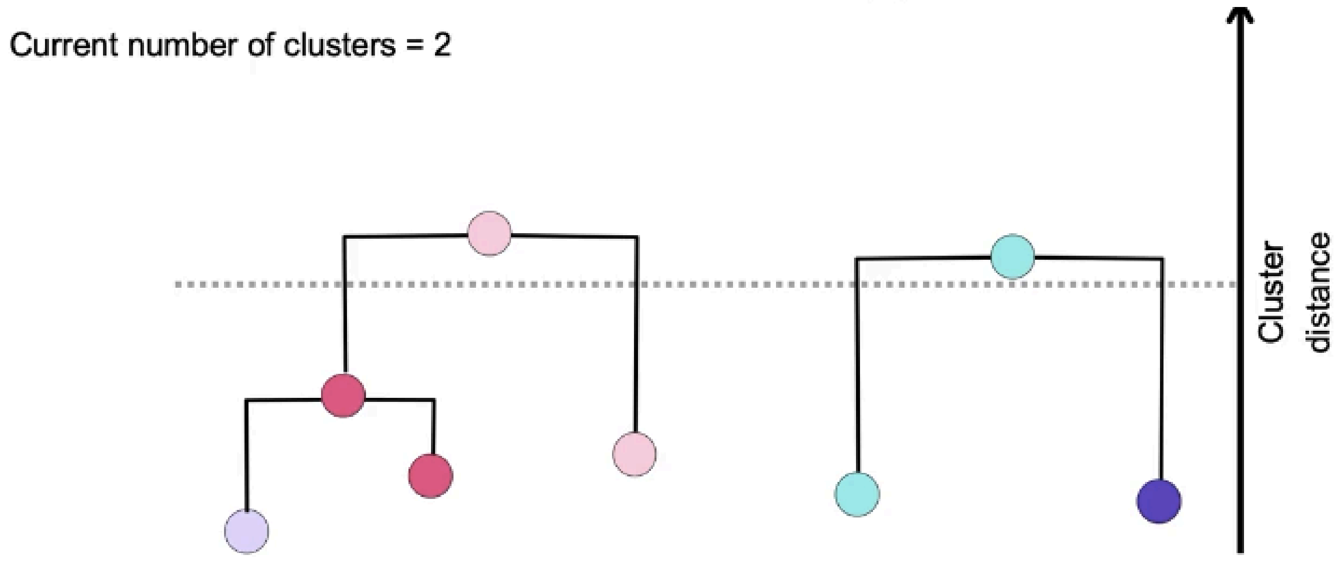
So suppose we are at this point:





We will keep clustering until we ensure that the distance of all the clusters exceed that threshold, like so:





Code

```python
# https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html
from sklearn.cluster import AgglomerativeClustering

model = AgglomerativeClustering(n_clusters=3, metric='euclidean', linkage='ward')
model.fit(X1)
y_predict = model.predict(X2)
```