Classical Unsupervised Machine Learning

John Armstrong, University of Glasgow

Outline

- What is unsupervised learning?
- How can it be useful in interpreting physical data?
- Different types of classical unsupervised learning:
    1. Clustering Algorithms
    2. Dimensionality Reduction
    3. Random Forests

Different Types of Machine Learning

	Supervised	Unsupervised
Classical	Decision Trees, SVMs, shallow neural networks	Clustering, Dimensionality Reduction
Deep	Multi-layer ANNs/ CNNs, ResNets	GANs, VAEs, INNs

University of Glasgow

Different Types of Machine Learning

	Supervised	Unsupervised
Classical	Decision Trees, SVMs, shallow neural networks	Clustering, Dimensionality Reduction
Deep	Multi-layer ANNs/ CNNs, ResNets	GANs, VAEs, INNs

Classical Unsupervised Learning

- Unsupervised = allowing the computer to identify the important features in your data.
- This is beneficial for identifying patterns or important quantities in your data.
- Prior knowledge of the dataset isn't required!

How does unsupervised learning work?

$$f(\overrightarrow{X}, \{a_1, \ldots, a_n\}) = \overrightarrow{Y}$$

How does unsupervised learning work?

$$f(\overrightarrow{X}, \{a_1, \ldots, a_n\}) = \overrightarrow{Y}$$

Input Data:

- Raw data
- Extracted features
- Distances/correlations

How does unsupervised learning work?

$$f(\overrightarrow{X}, \underbrace{\{a_1, \ldots, a_n\}}) = \overrightarrow{Y}$$

Input Data:

- Raw data
- Extracted features
- Distances/ correlations

Hyperparameters:

- Tuning parameters for the algorithm.
- The supervised bit of unsupervised learning

How does unsupervised learning work?

Internal Choices of the computer:

- How the computer manipulates the input and the hyperparameters

$$f(\overrightarrow{X}, \underbrace{\{a_1, \ldots, a_n\}}) = \overrightarrow{Y}$$

Input Data:

- Raw data
- Extracted features
- Distances/ correlations

Hyperparameters:

- Tuning parameters for the algorithm.
- The supervised bit of unsupervised learning

How does unsupervised learning work?

Internal Choices of the computer:

- How the computer manipulates the input and the hyperparameters

$$f(\overrightarrow{X}, \{a_1, \ldots, a_n\}) = \overrightarrow{Y}$$

Output:

- The output of the algorithm

Input Data:

- Raw data
- Extracted features
- Distances/ correlations

Hyperparameters:

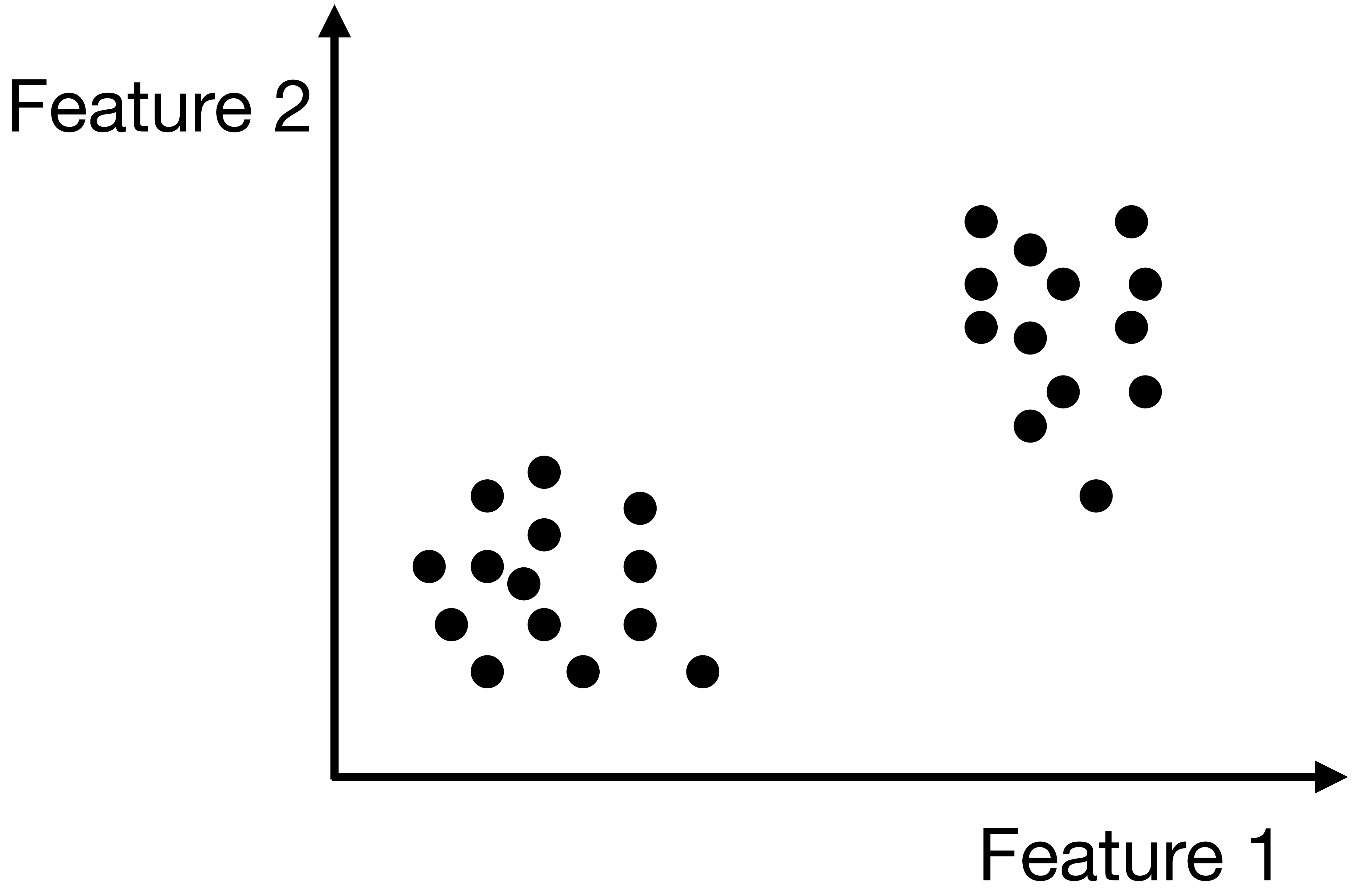
- Tuning parameters for the algorithm.
- The supervised bit of unsupervised learning

Clustering Algorithms

- Groups together data points with similar characteristics
- The methods for doing this typically involve distances in the plane of your observables
- We will look at hierarchical clustering but other methods do exist

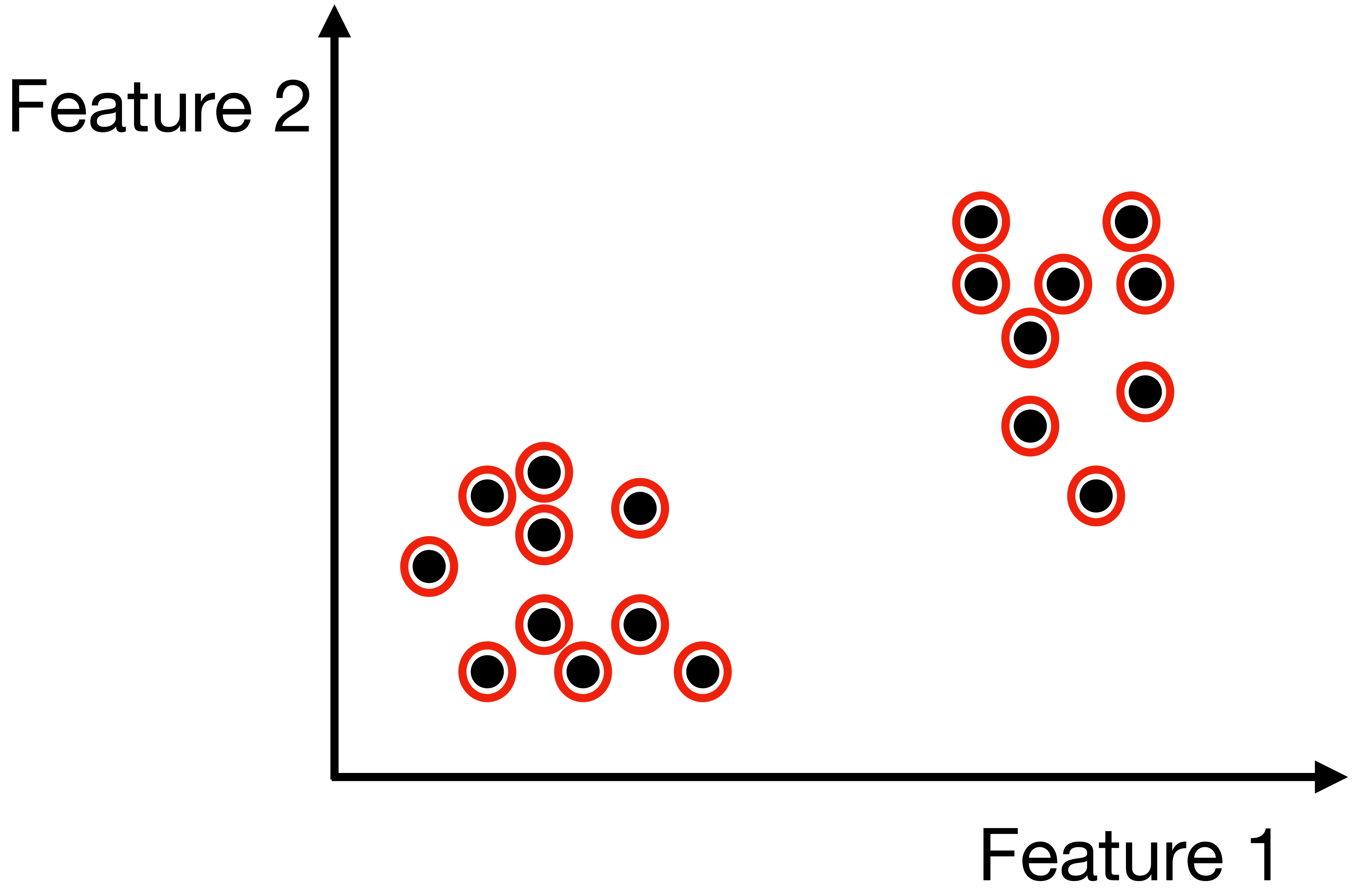
Hierarchical Clustering

The method is simple:



- Each point starts as its own cluster
- The distances between the clusters are calculated
- The two closest clusters are merged into one cluster
- The process is repeated until all the points are in one large cluster

Hierarchical Clustering

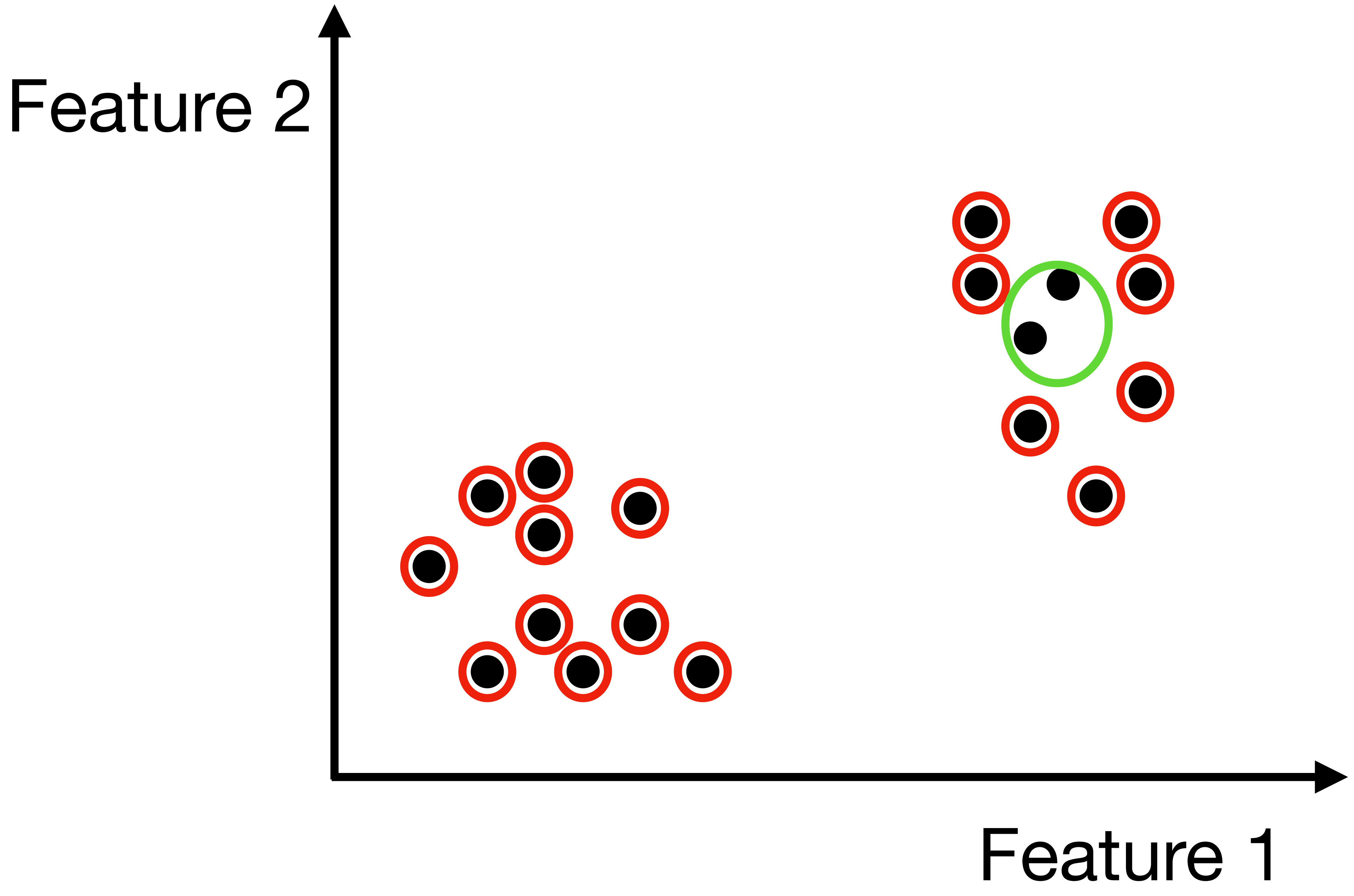


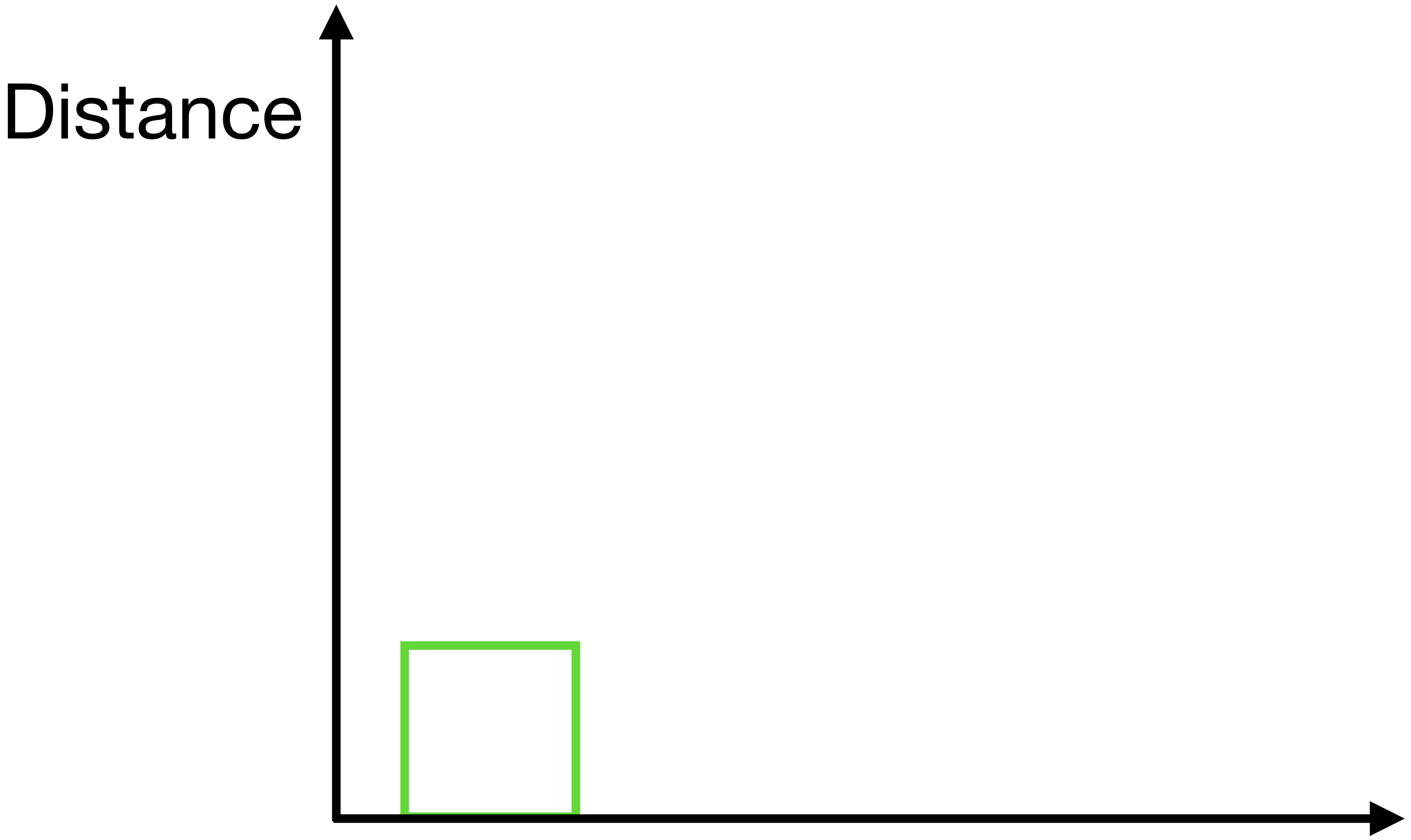
The method is simple:

- Each point starts as its own cluster
- The distances between the clusters are calculated
- The two closest clusters are merged into one cluster
- The process is repeated until all the points are in one large cluster

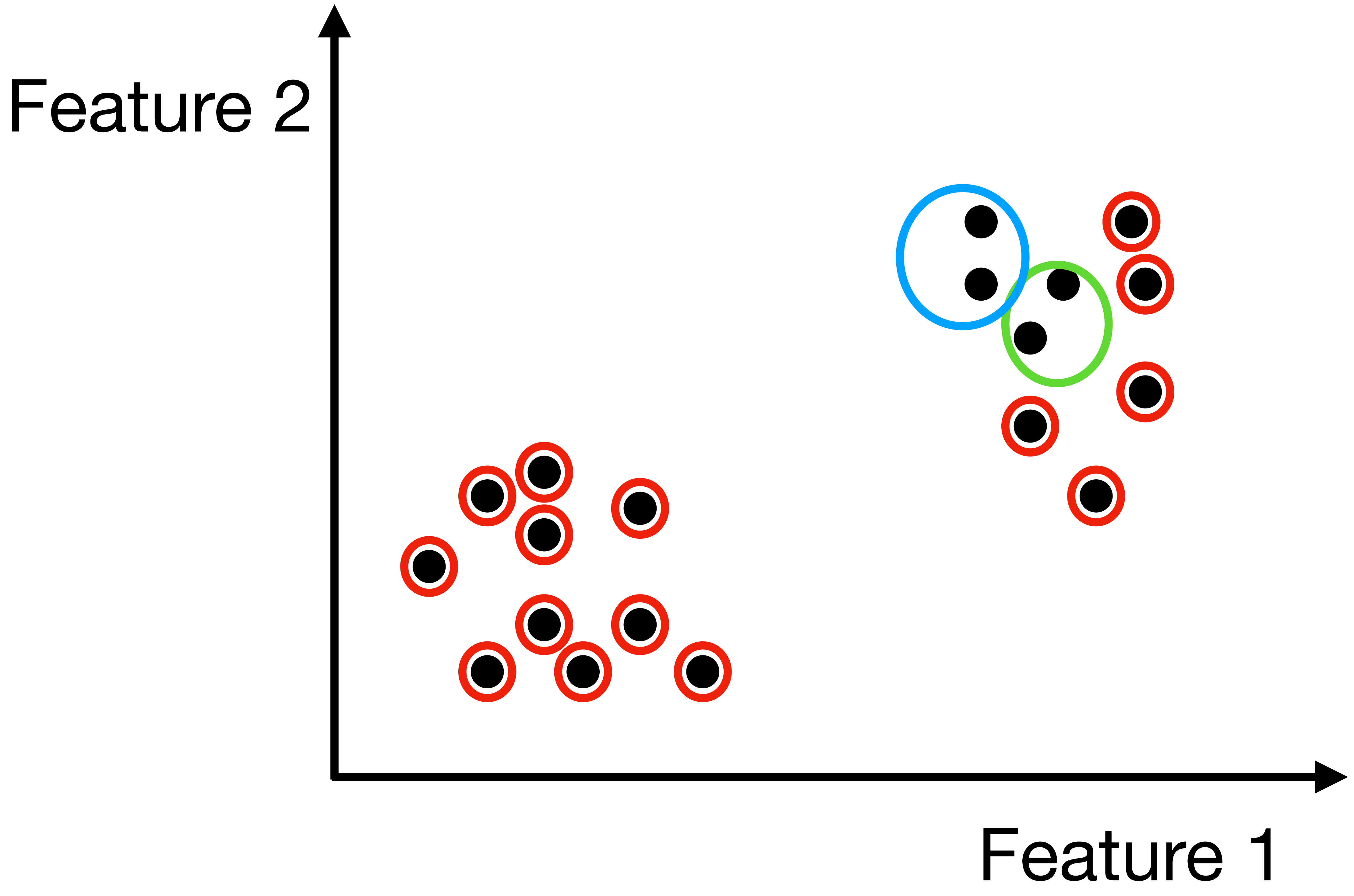
Hierarchical Clustering

Dendrogram

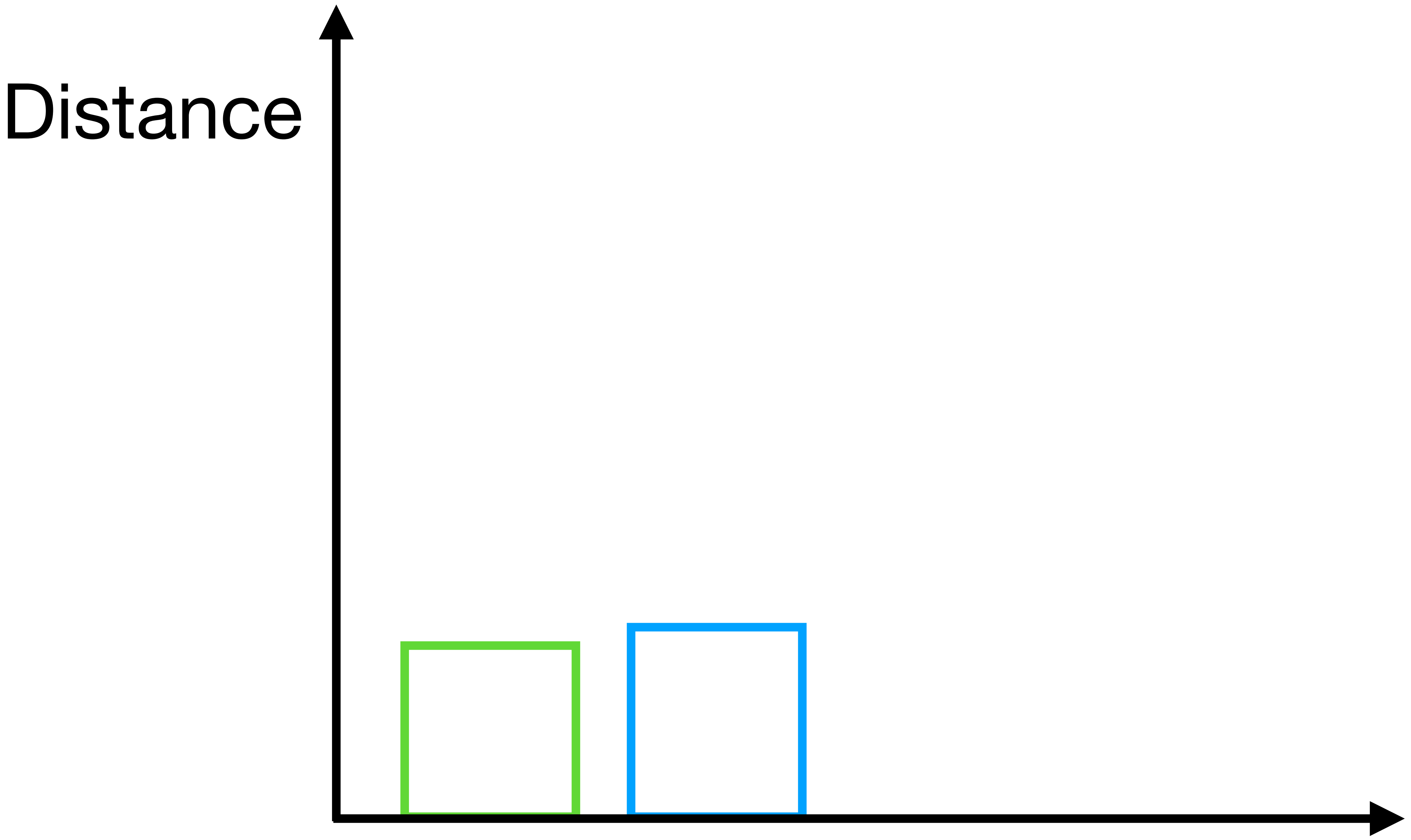




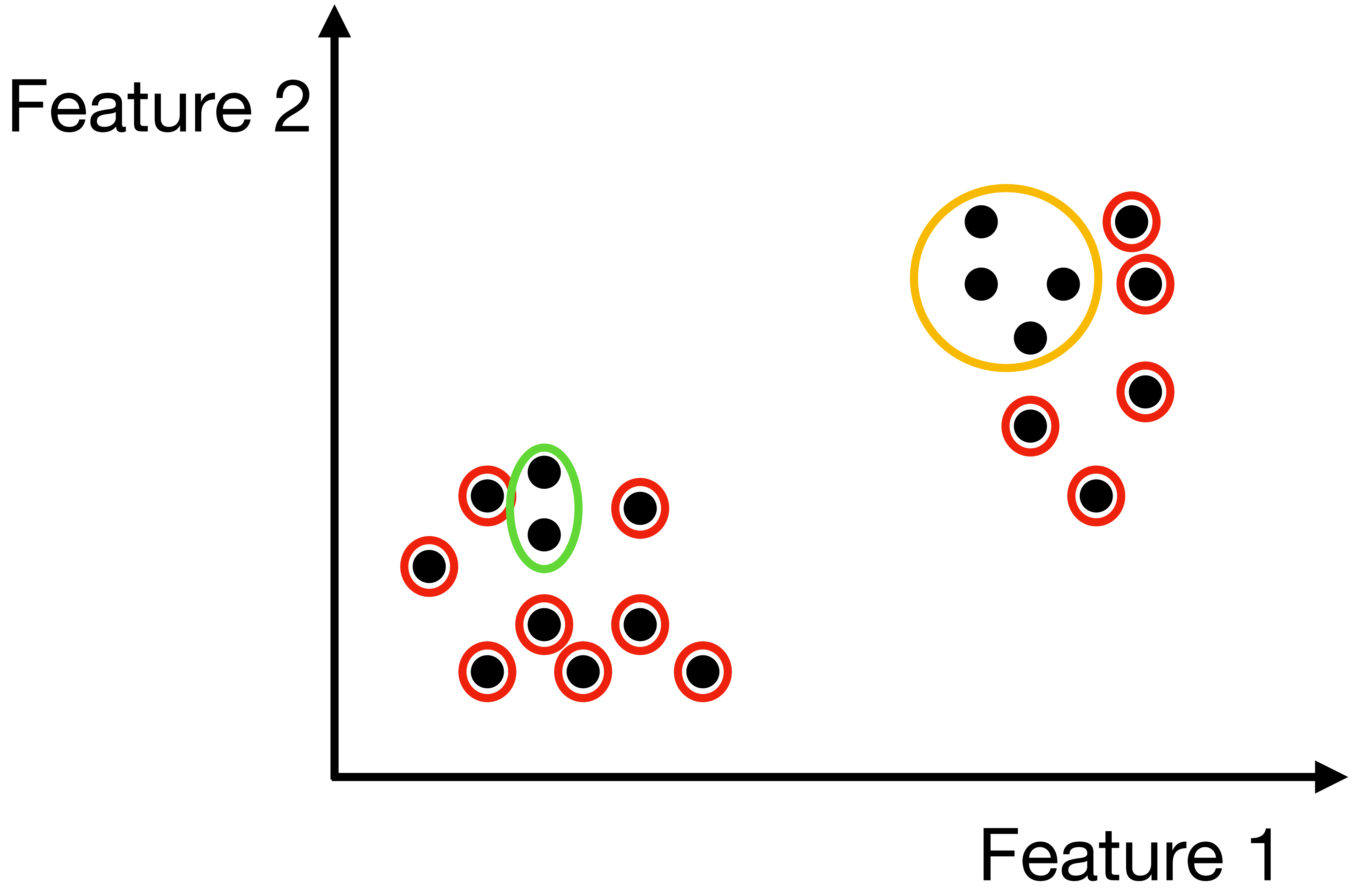
Hierarchical Clustering



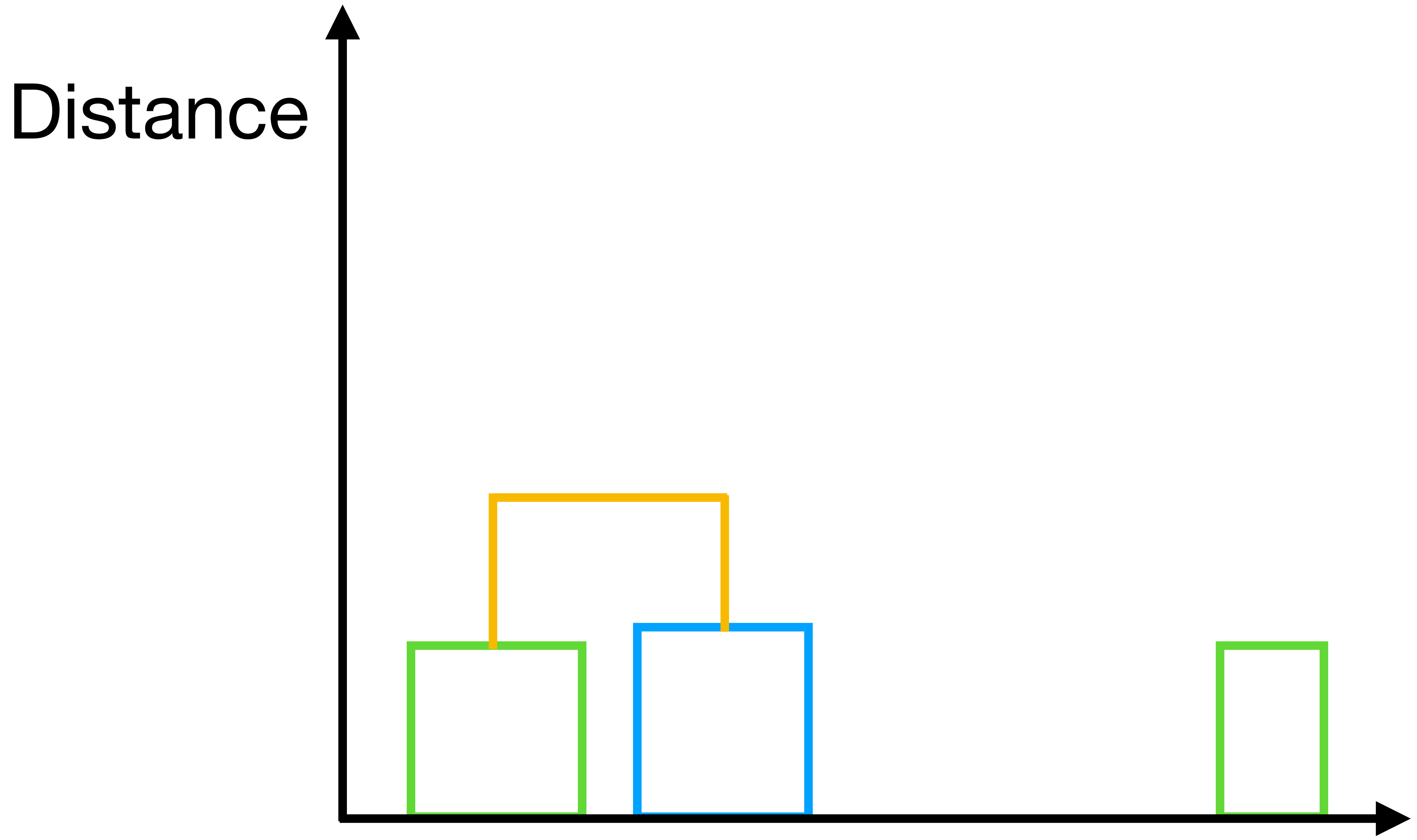
Dendrogram



Hierarchical Clustering

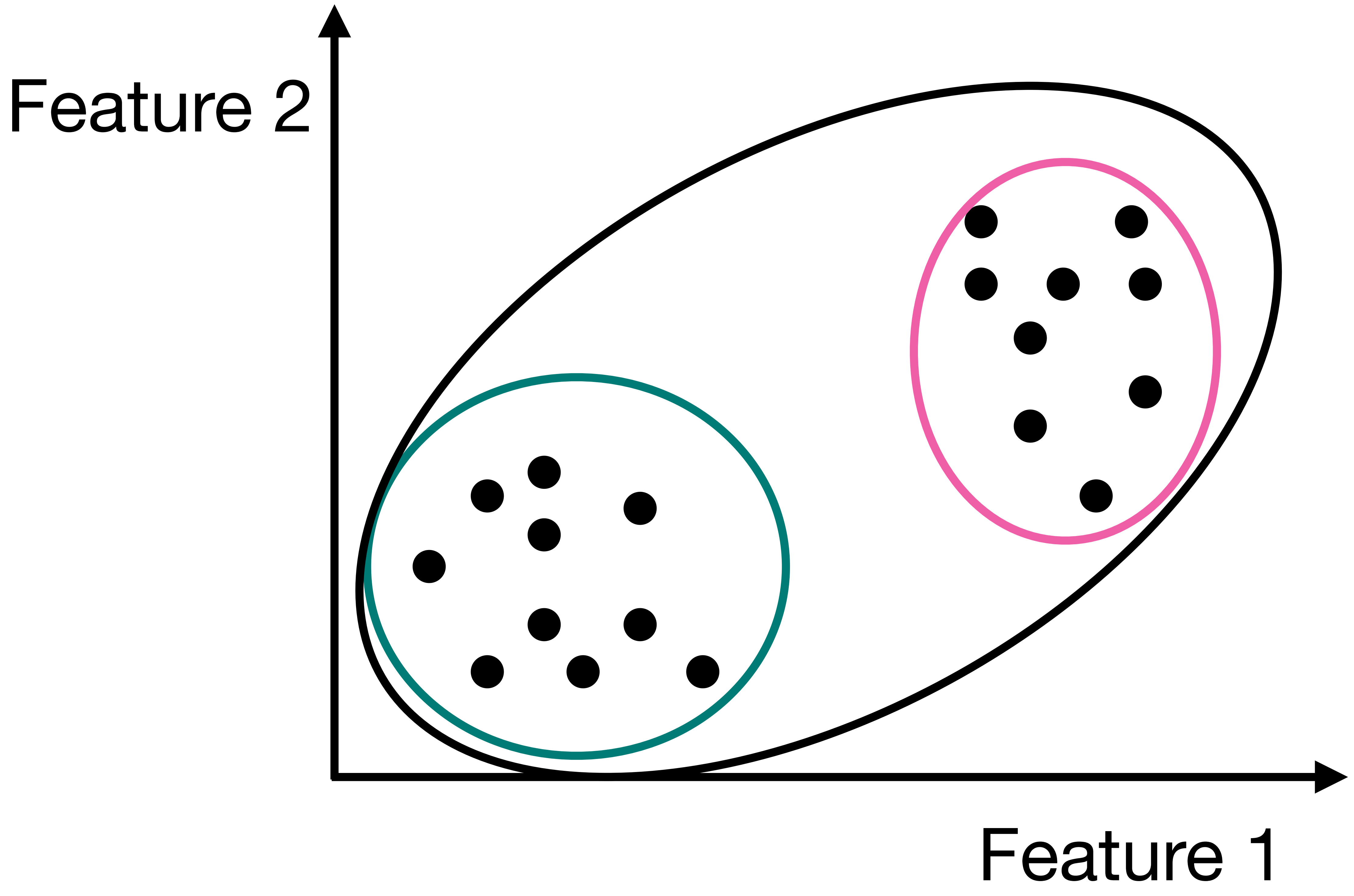


Dendrogram

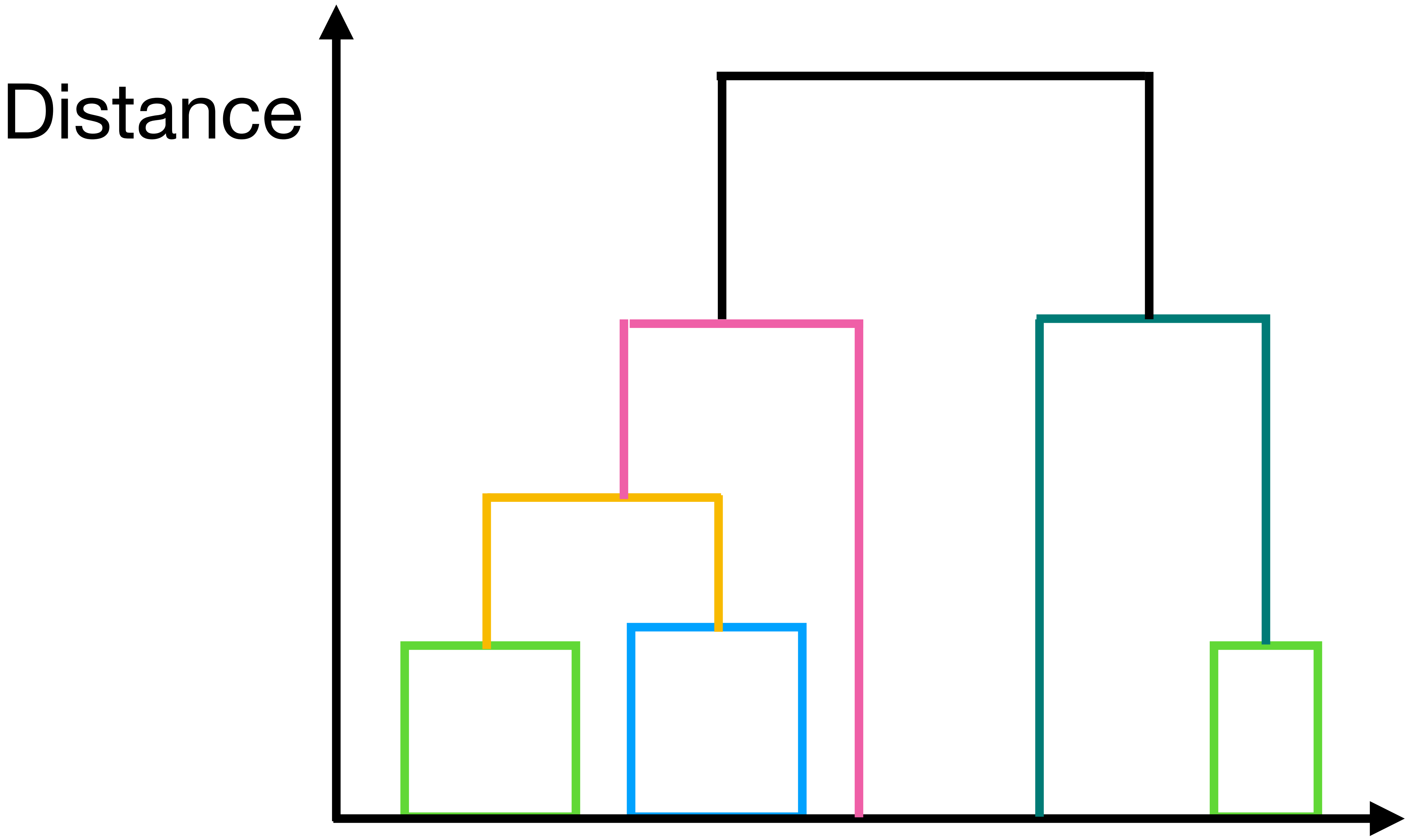


Hierarchical Clustering

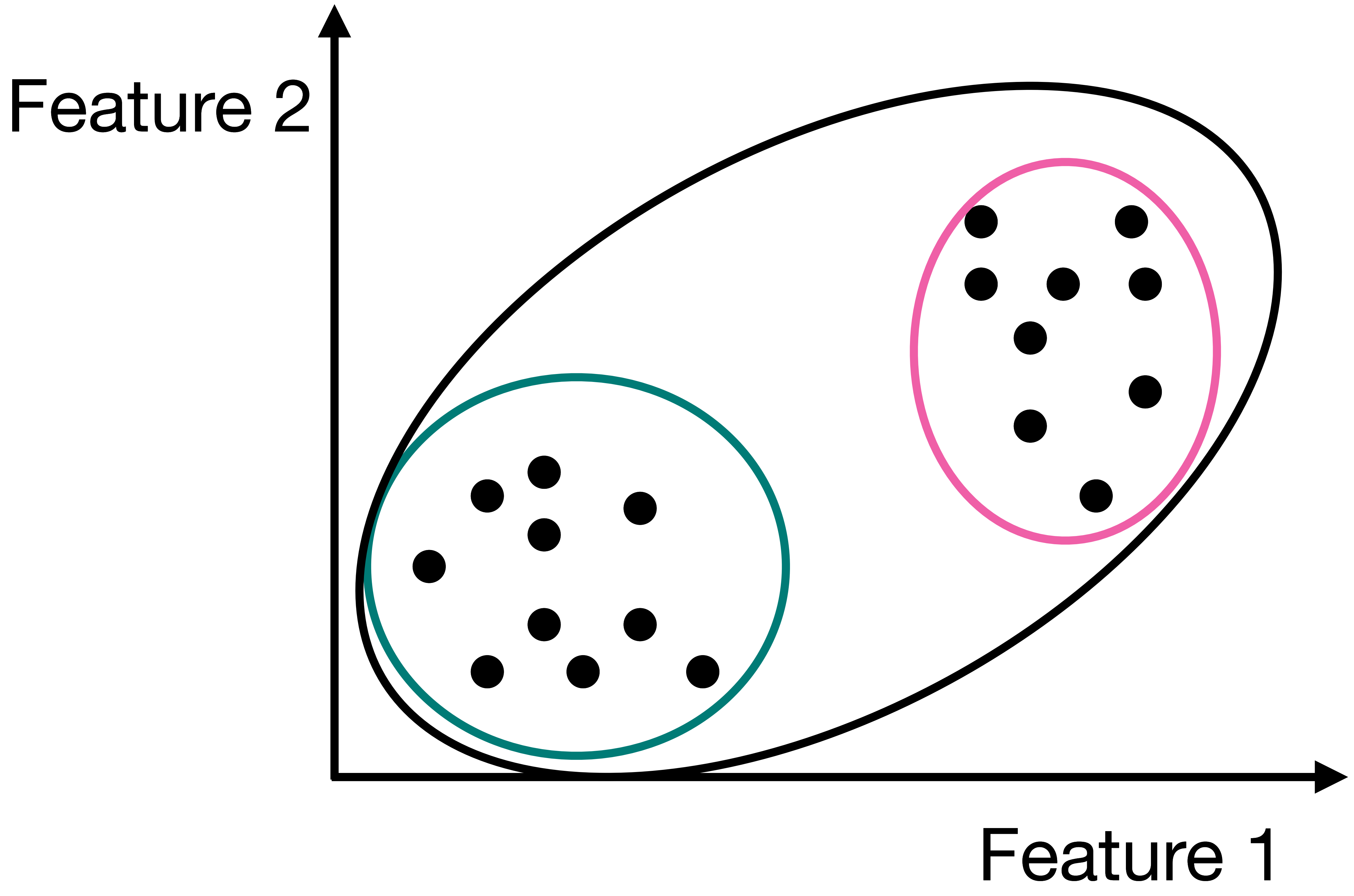
etc….



Dendrogram



Hierarchical Clustering



There are 3 hyperparameters:

1. The number of clusters, k

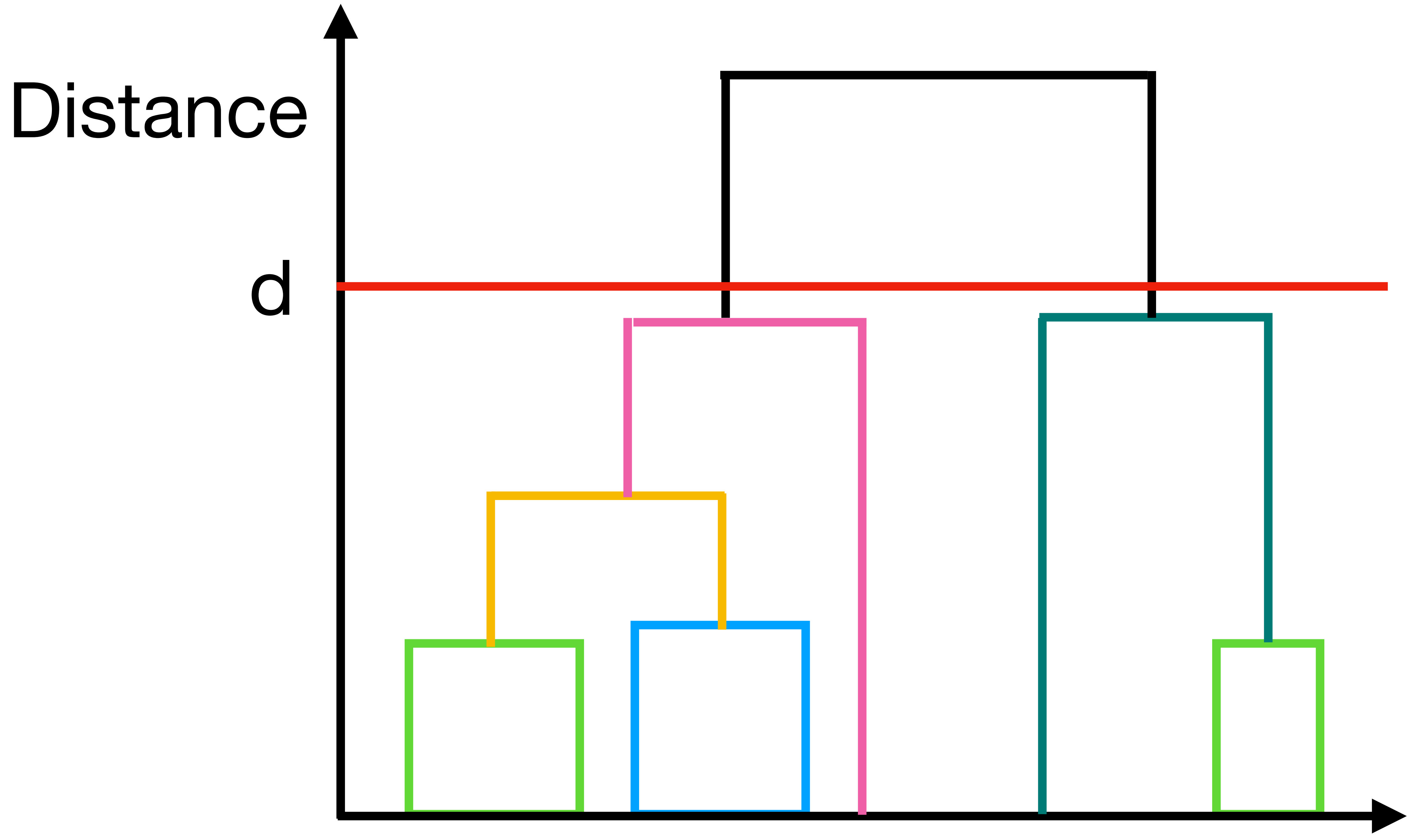
e.g. in our example, it is obvious that k = 2,

however in reality, it may be more difficult and

you may need to experiment with k

Tuning k can be done by changing the cutoff

distance d in your dendrogram

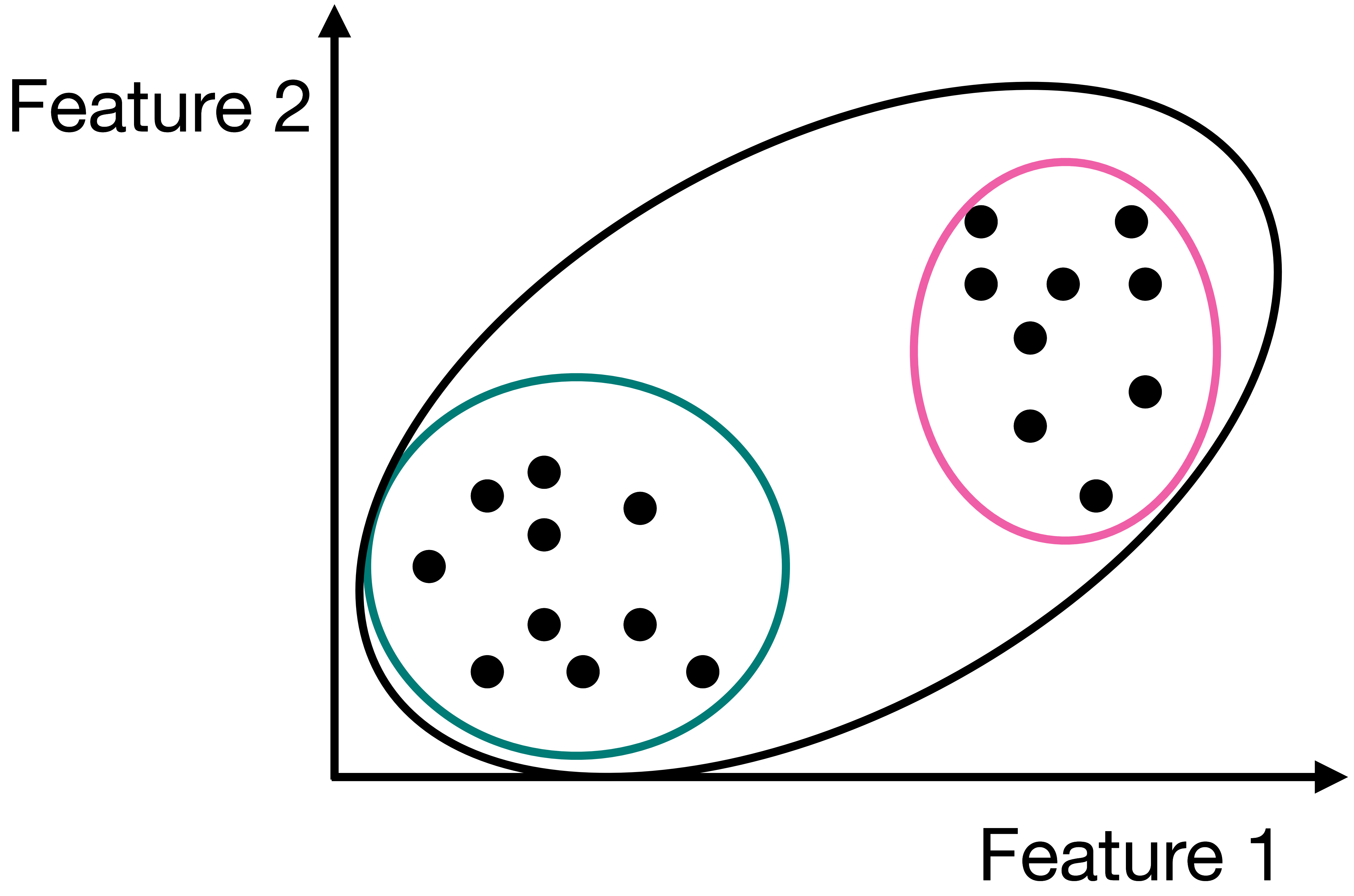


There are 3 hyperparameters:

2. Cutoff distance, d

This is the distance at which you decide there is too large a distance between clusters for them to be part of one larger cluster

Hierarchical Clustering

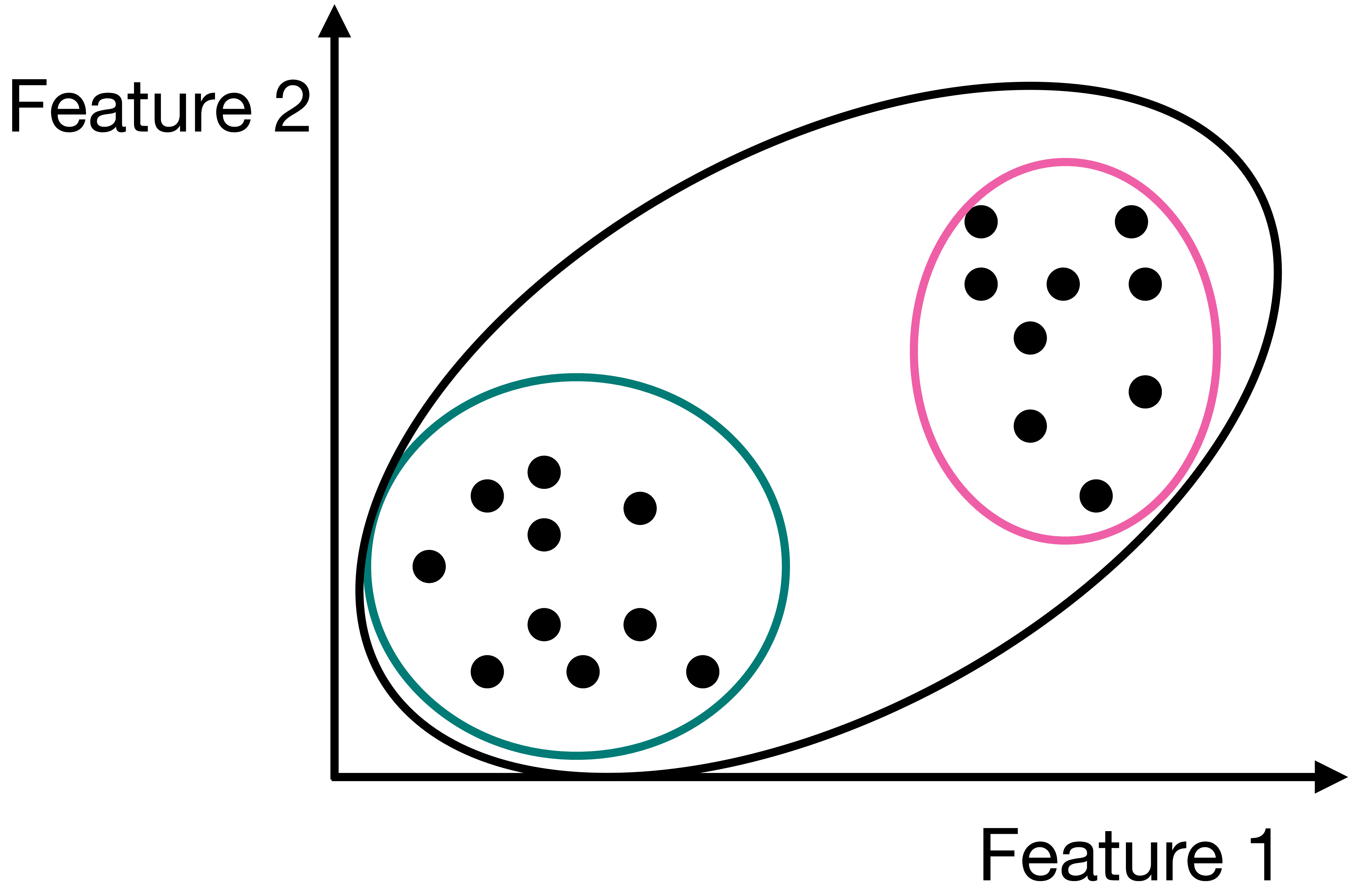


There are 3 hyperparameters:

3. The distance metric to calculate the distance between clusters e.g. Euclidean

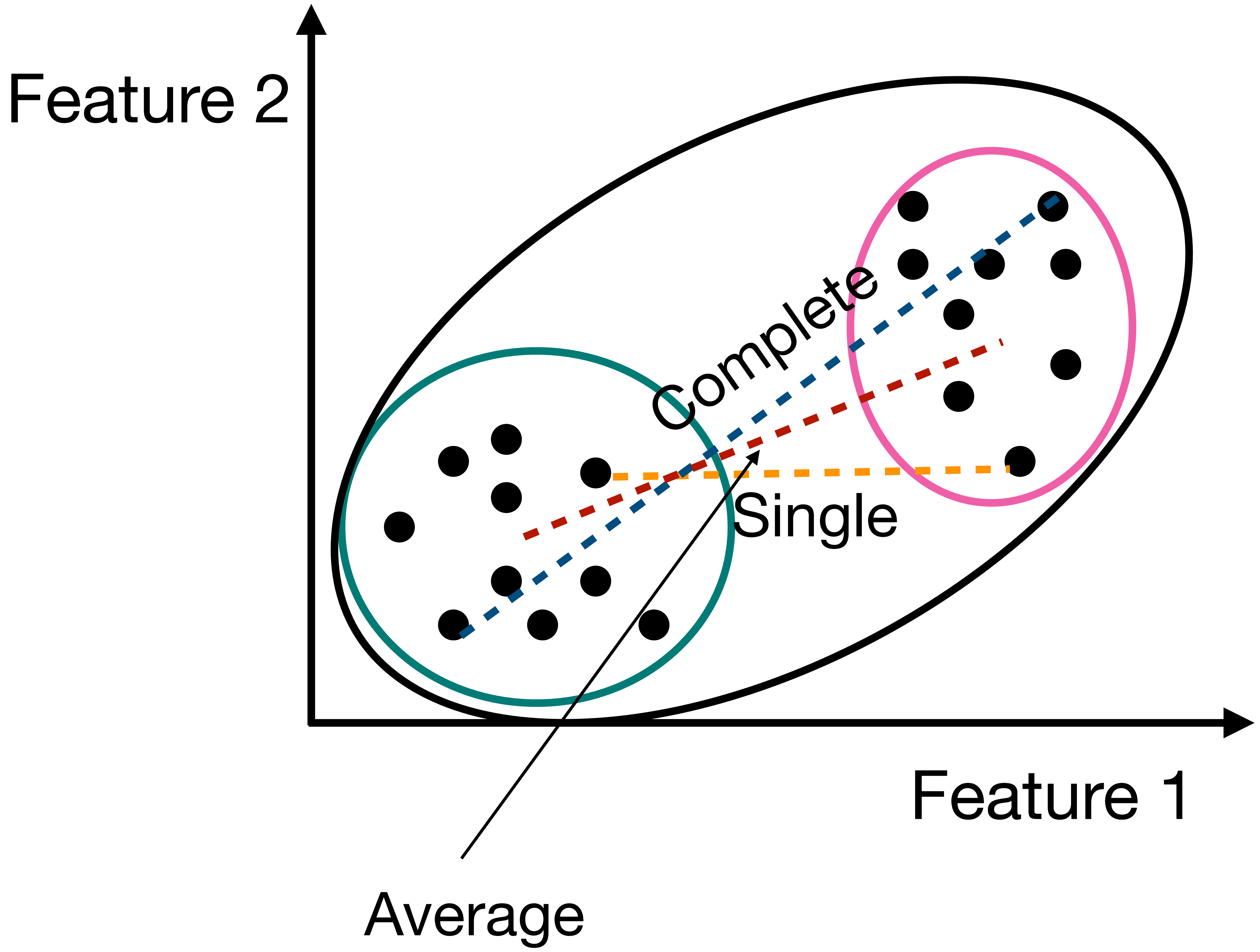
$$d(\overrightarrow{x}_1, \overrightarrow{x}_2) = ||x_2 - x_1|| = \sqrt{\sum_{i=1}^{n} \left(x_1^i - x_2^i\right)^2}$$

Hierarchical Clustering



Linkage method:

- Finding distances between clusters is something also to be tuned (find which works best for you)
- When clusters have multiple points, defining the “distance” between two clusters is non-trivial
- How we do this is known as our algorithm’s **linkage method**



Examples of linkage methods:

- Single — distance between two closest points
- Average — distance between average cluster position
- Complete — distance between two furthest points
- Ward — keeps growth in sum of squares as small as possible (requires Euclidean metric)

In summary:

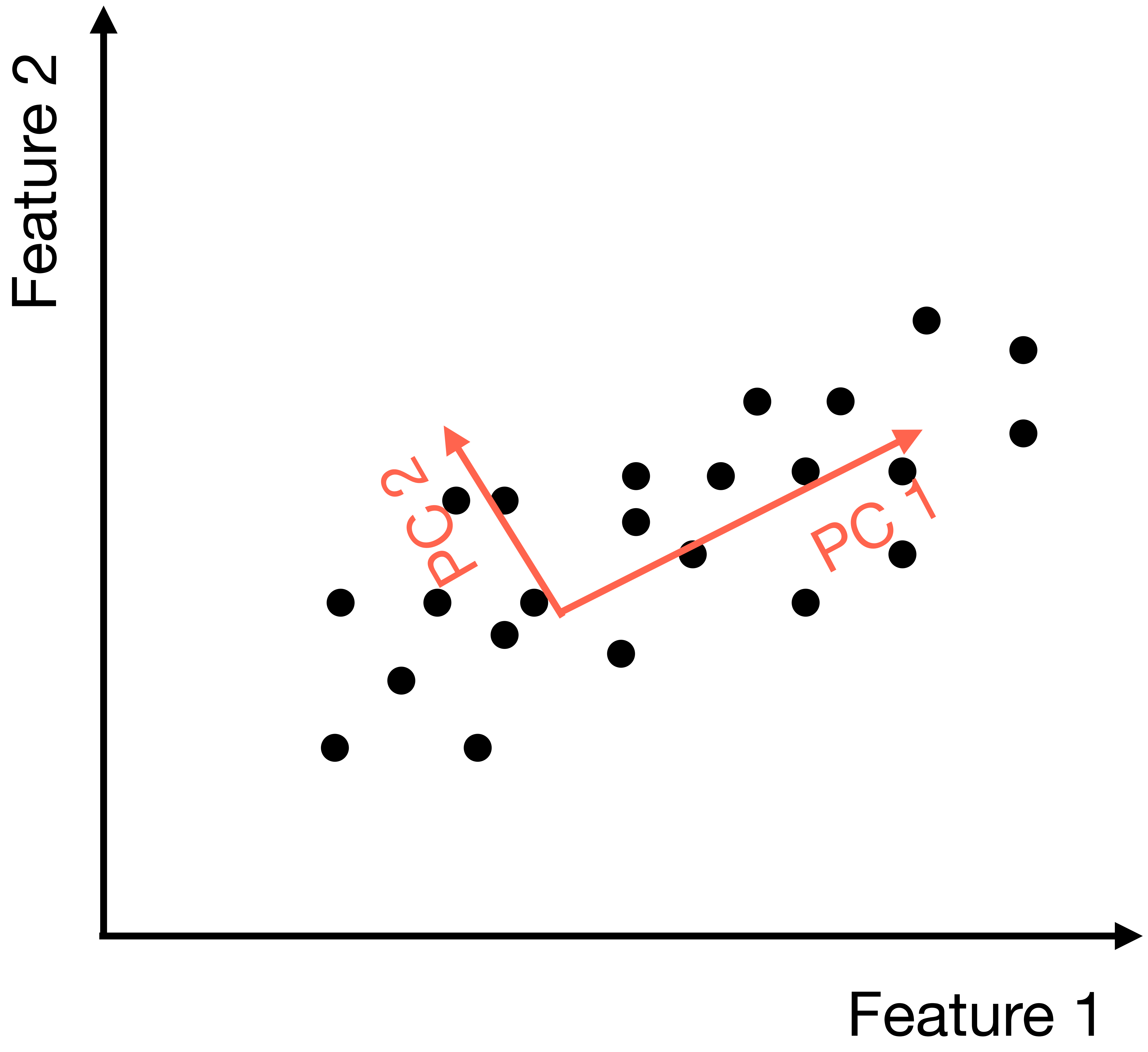
- Hierarchical clustering is very simple but has the potential to be very powerful
- We need to consider 4 different choices when using it:
  1. The number of clusters, k
  2. The cutoff distance, d
  3. The distance metric
  4. The linkage method

Dimensionality Reduction

- Decomposition of data into important features and embedding a high-dimensional dataset into a lower-dimensional space.
- Important as certain features of your data may be redundant
- Improves performance of supervised learning
- Data compression/uncover complex trends

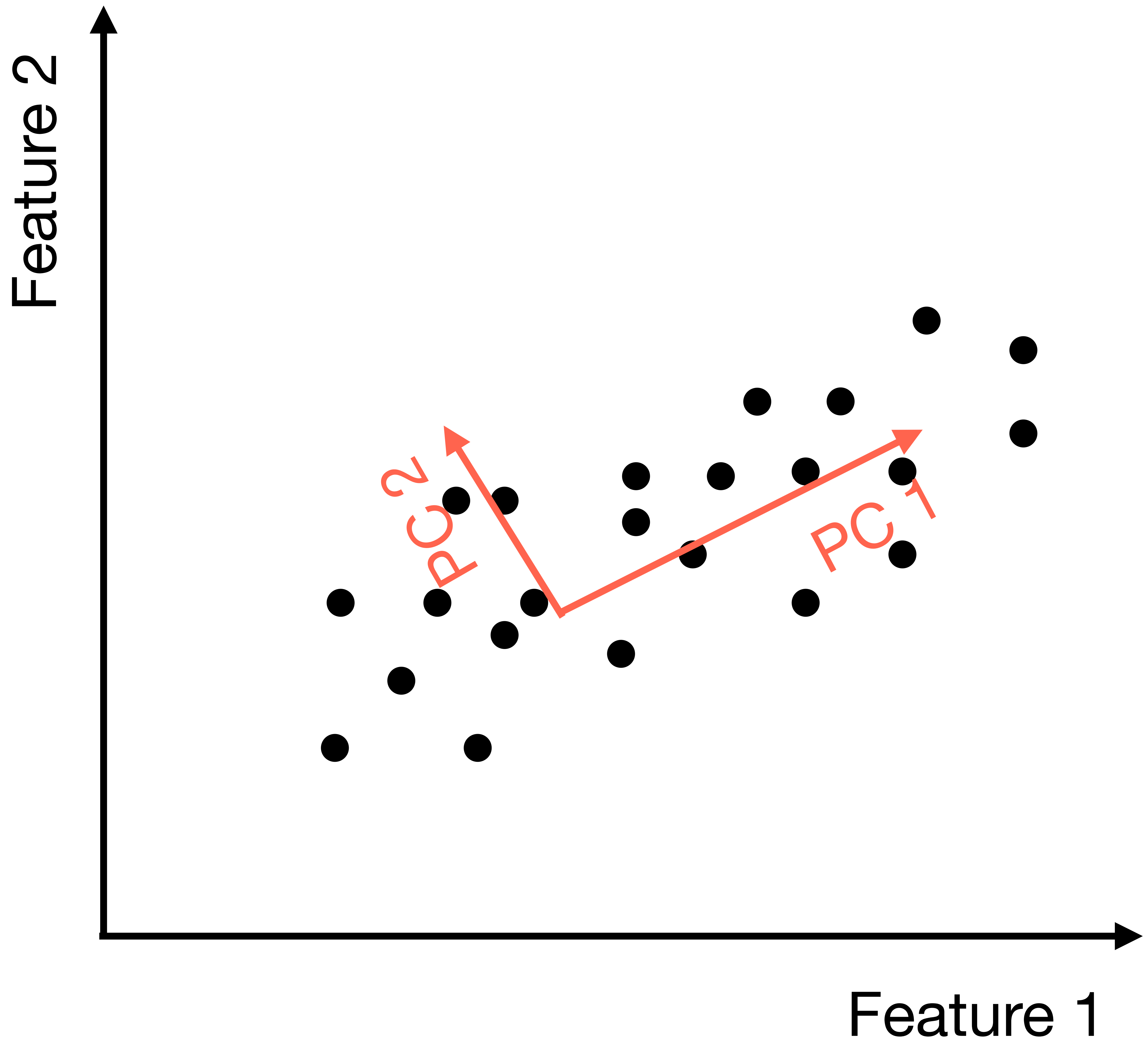
Principal Component Analysis (PCA)

- Orthogonal linear transformation to basis of so-called “principal components”
- These are the directions in the data with the most variance



Principal Component Analysis (PCA)

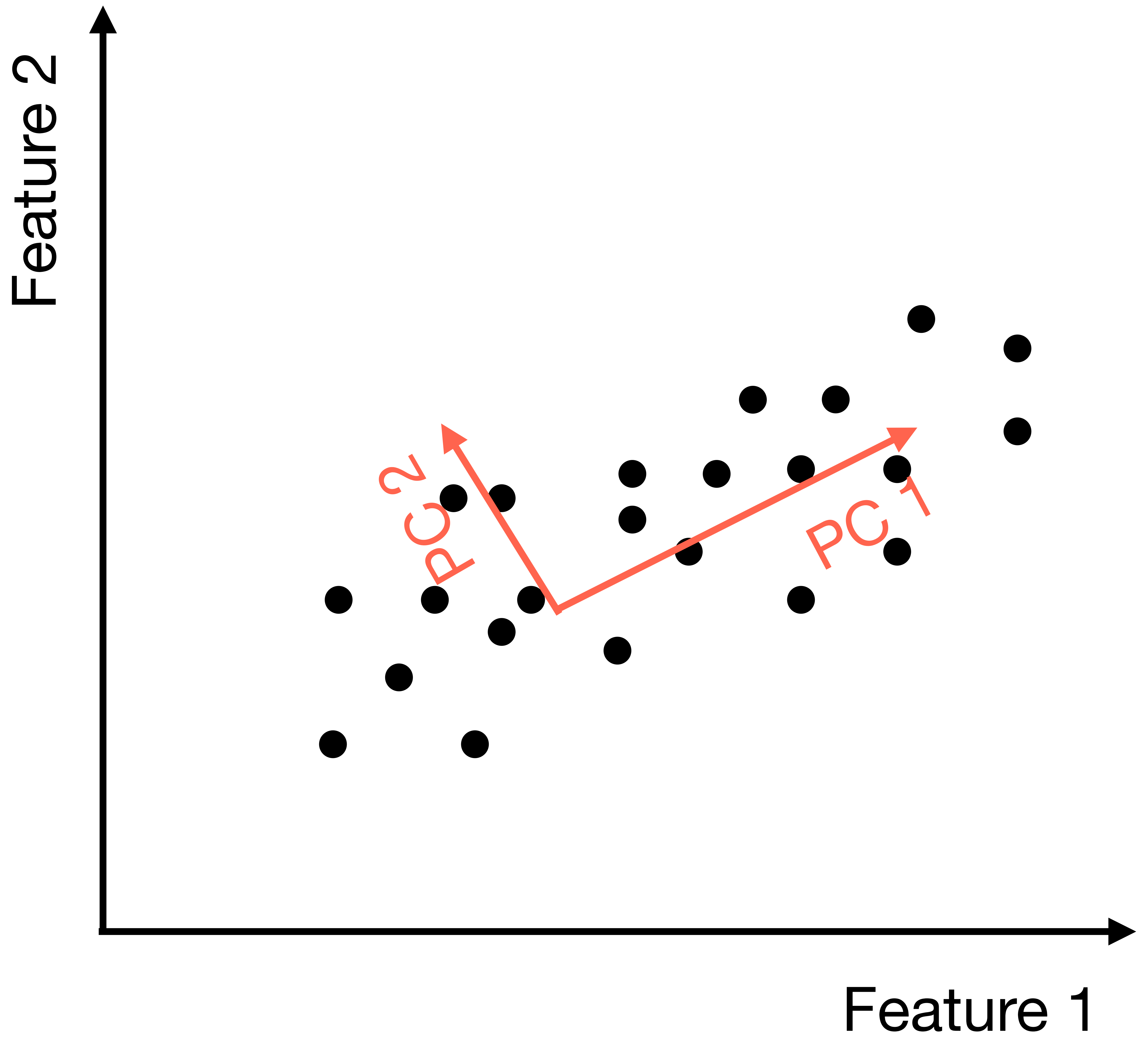
- The first principal component has the largest variance in the data
- Then each proceeding component has the largest variance of the remaining directions



Principal Component Analysis (PCA)

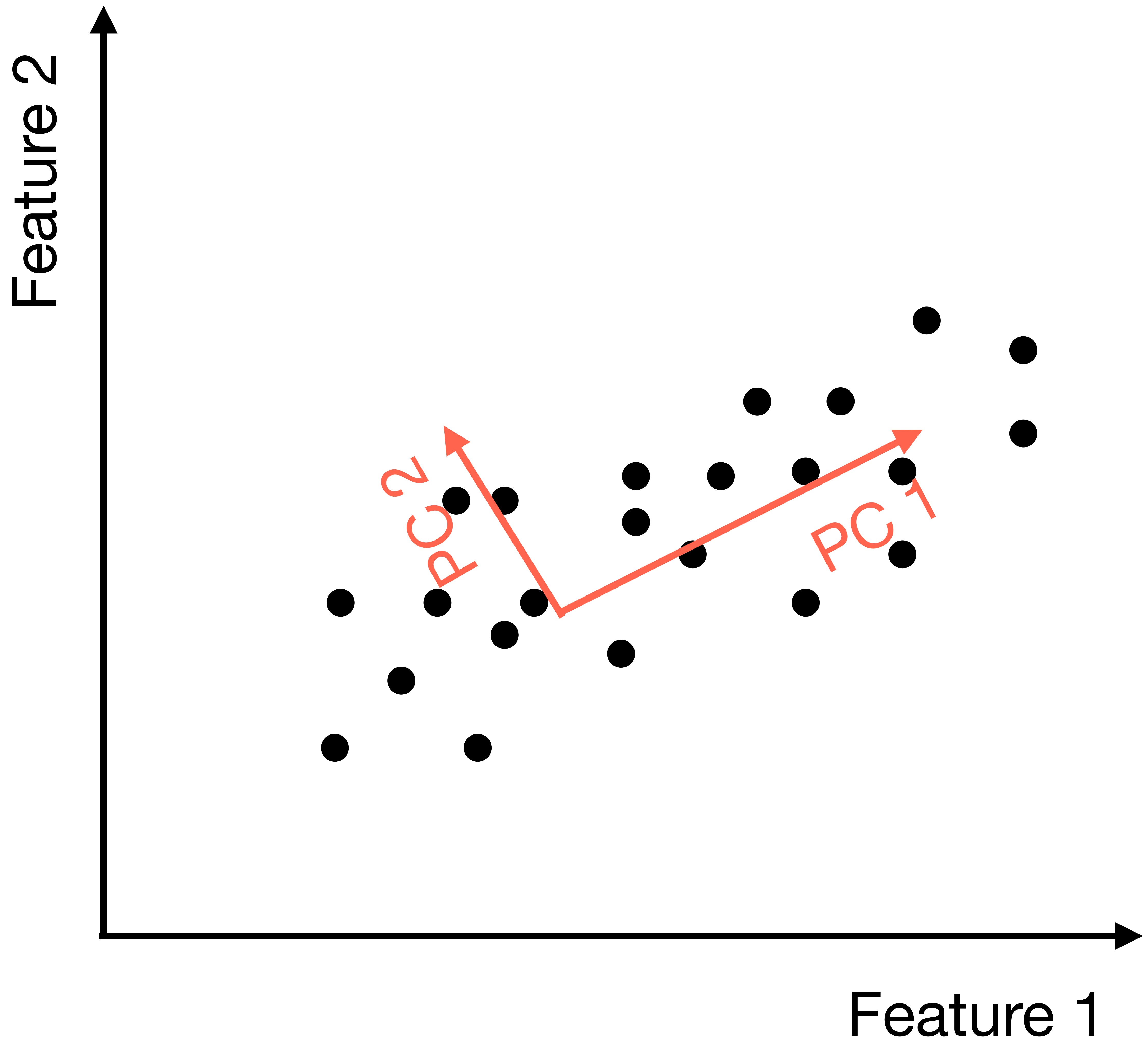
- PCA allows you to represent the data as a linear combination of the principal components

$$x = \sum_{i=1}^{N} A_i \times PC_i$$



Principal Component Analysis (PCA)

- Data can then be compressed by expressing it in terms of the most important principal components
- This also gives a low-dimensional representation



There are many other methods

- I have only explained two of a variety of techniques that can be used
- There are many other clustering and dimensionality reduction techniques to be used that may be more useful to your research
- Classical unsupervised learning can also be used for outlier detection and classification which we have not covered.
- https://scikit-learn.org/ has good examples for many methods and an easy-to-use API
- sklearn also has a great cheatsheet: http://blog.kaggle.com/2015/04/15/scikit-learn-video-2-setting-up-python-for-machine-learning/ on deciding what algorithm to use

University of Glasgow

Exercises

- The exercises can be found on my GitHub: https://bit.ly/2DtXei2 and are (hopefully) self-explanatory
- Feel free to email me with any questions about machine learning and/or the CDT (whether it be structure or courses or events): j.armstrong.2@research.gla.ac.uk
- Honest feedback is appreciated!