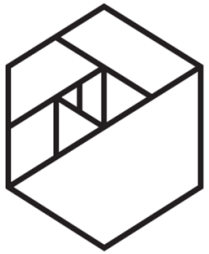


Clustering Methods

INTRODUCTION TO DATA SCIENCE – FALL 2018

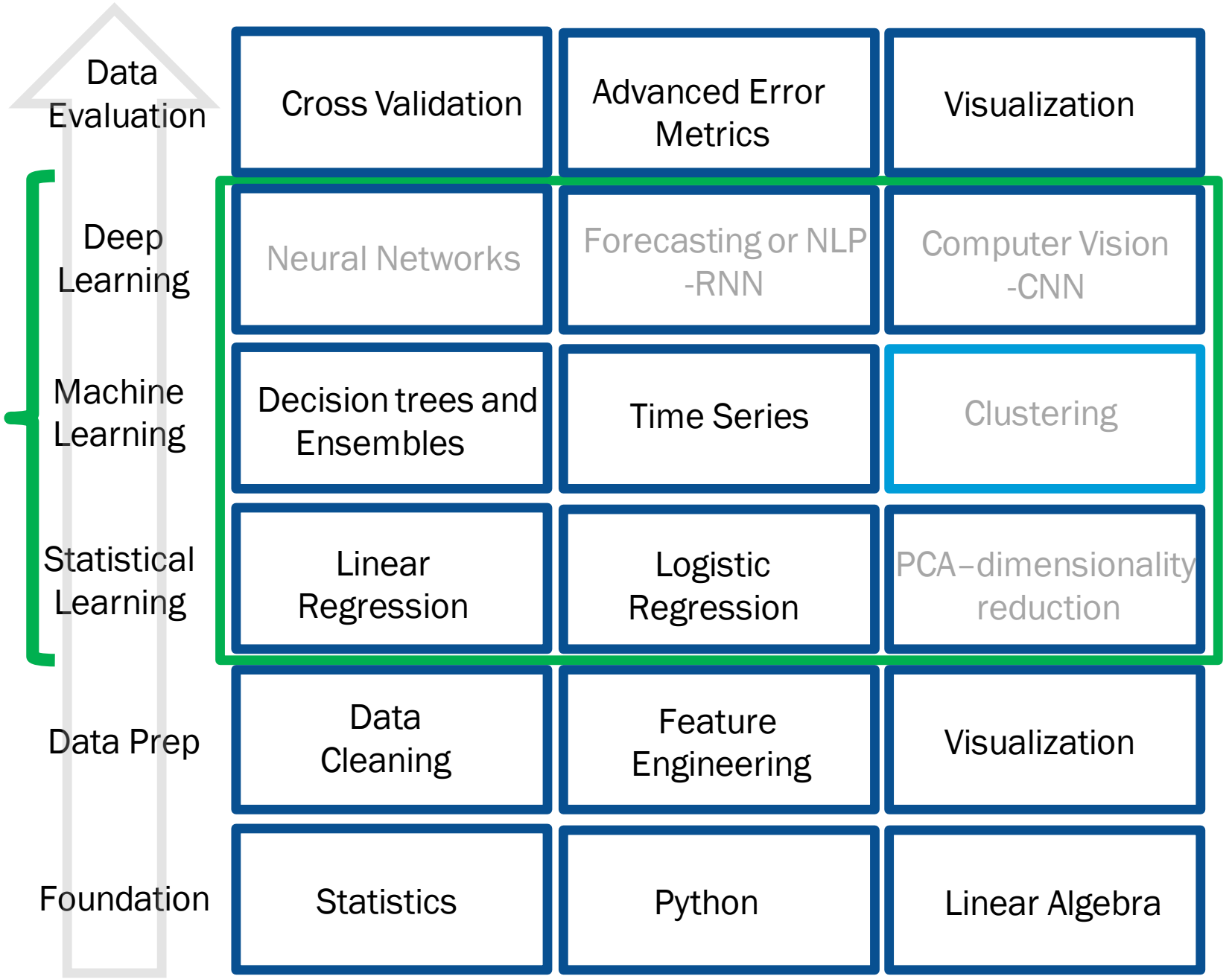
SESSION 9

AGENDA

1. Unsupervised learning
2. K-means clustering
3. DBSCAN
4. Hierarchical clustering

Introduction to Data Science

- Learning the steps in the Data Science Process
- Learning multiple model methodologies

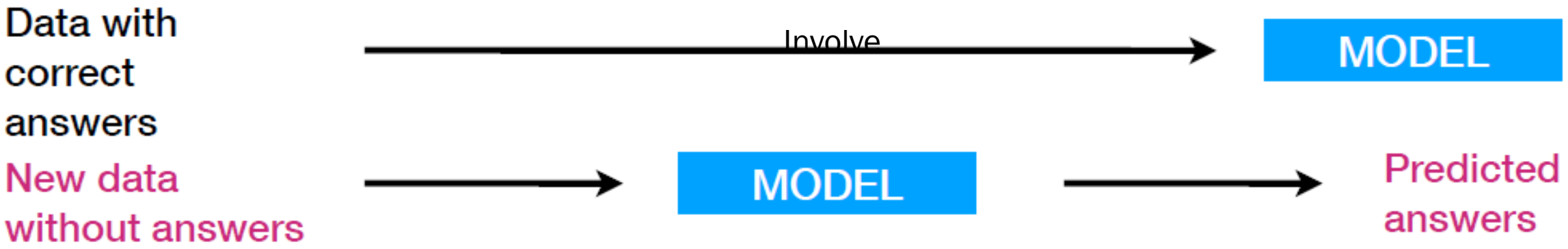


Unsupervised learning

DATA WITHOUT LABELS

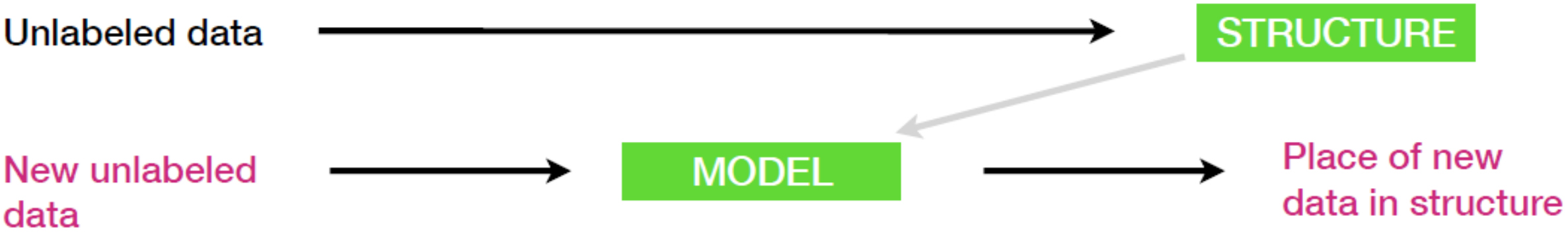
Supervised learning problems

- Involve constructing an accurate model that can predict some kind of an outcome when past data has labels for those outcomes



Unsupervised learning problems

- Involve constructing models where labels on historical data are unavailable



Unsupervised business problems

- Similarity matching – How can we identify similar individuals based on data we know about them
- Example: IBM is interested in finding customers similar to their best business customers
- K-nearest neighbors, hierarchical clustering

Unsupervised business problems

- Clustering– Not driven by any specific purpose
- Example: Do our customers form natural groups or segments?
- Preliminary exploration, may lead to questions like:
  - What products should we offer?
  - How should our customer care team be developed?
- K-means clustering, DBSCAN

Unsupervised business problems

- Dimensionality reduction
- Reducing the number of predictors you have and determining which are the most important
- Example: Can we determine what are the most important variables that influence gallons of gasoline purchased?
- PCA, SVD

K-means clustering

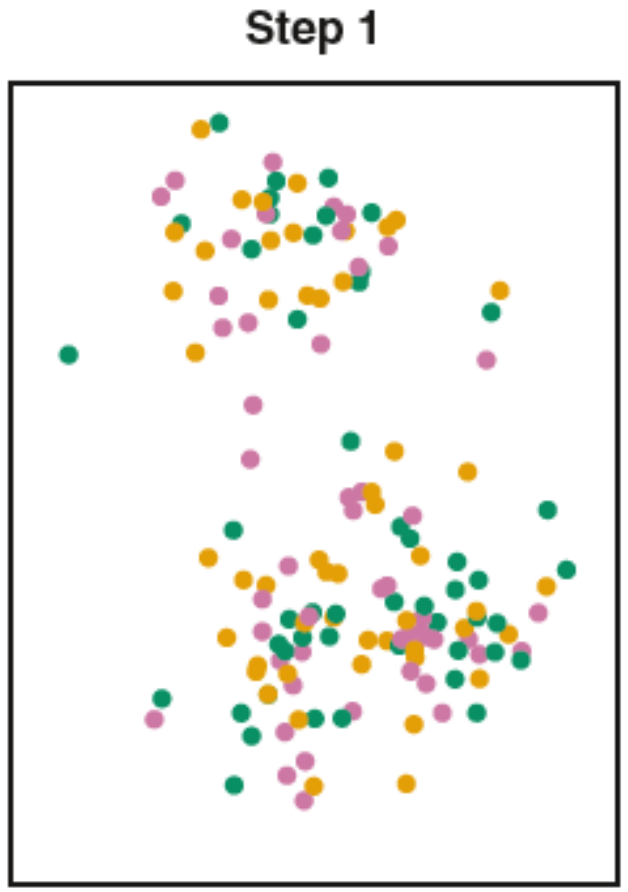
CLUSTERING METHOD 1

K-means algorithm

- Input is a single parameter (k), which is the number of clusters you want the underlying data to fall into, and attempts to find those clusters automatically
- In this example we choose k = 3

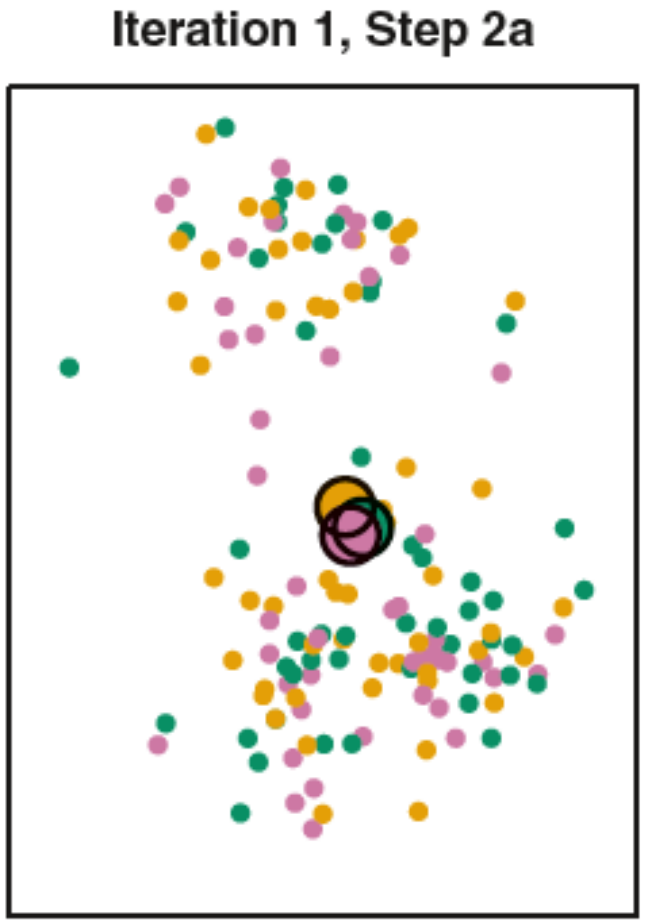
K-means algorithm

- Each observation is assigned a random cluster (pink, yellow, or green)



K-means algorithm

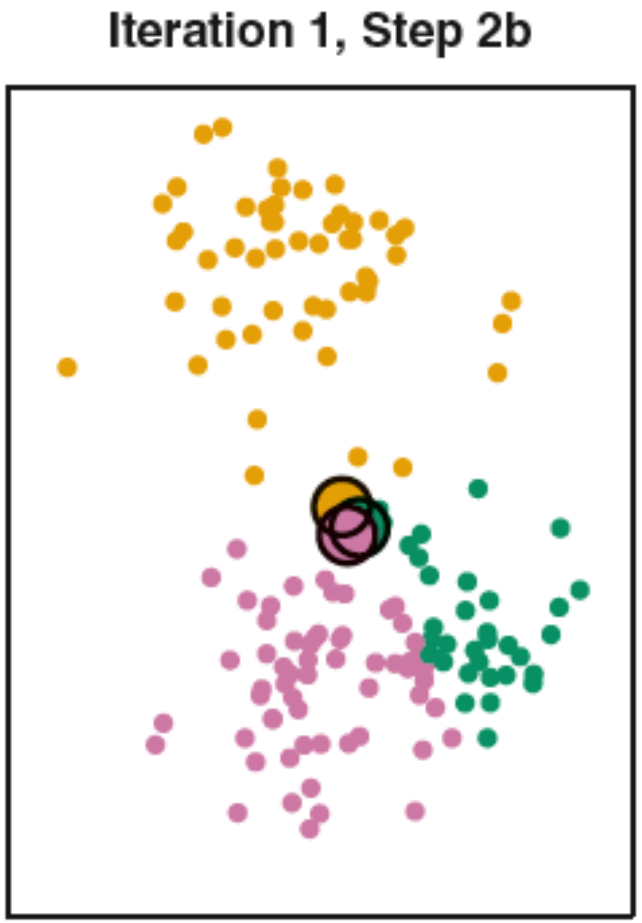
- Calculate the cluster centroids, these are shown by large colored disks
- Initially the centers are overlapping because the initial cluster assignments were chosen at random



K-means algorithm

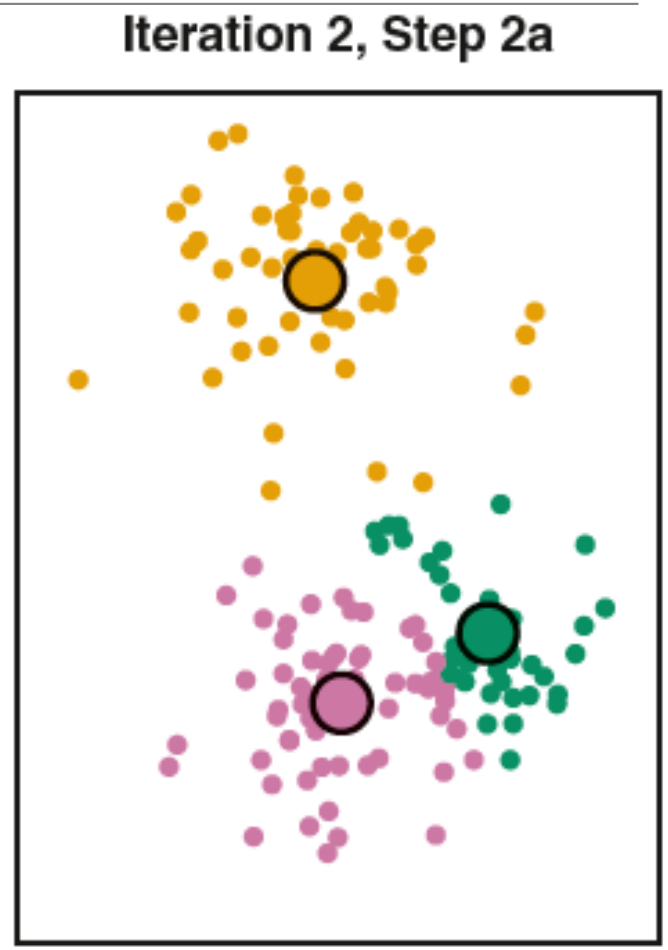
- Each observation is then assigned to the nearest centroid
- Use Euclidean distance to measure, where two points (x, y) have k number of features

$$d(x, y) = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2}$$



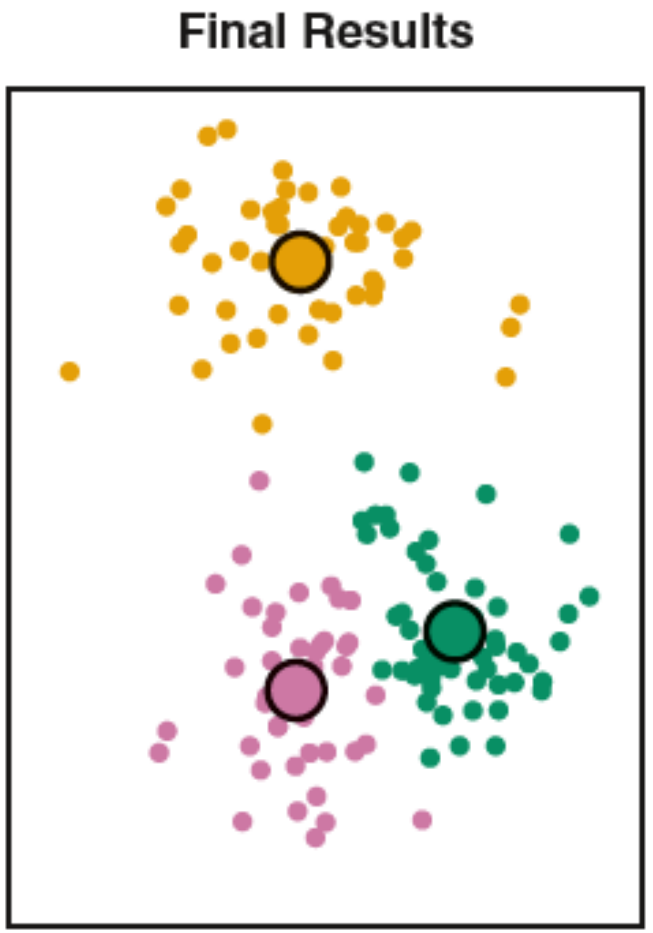
K-means algorithm

- In the second iteration, we repeat the process, of calculating the centroids
- We see that the colored disks have shifted from the center of the feature space to the center of their clusters



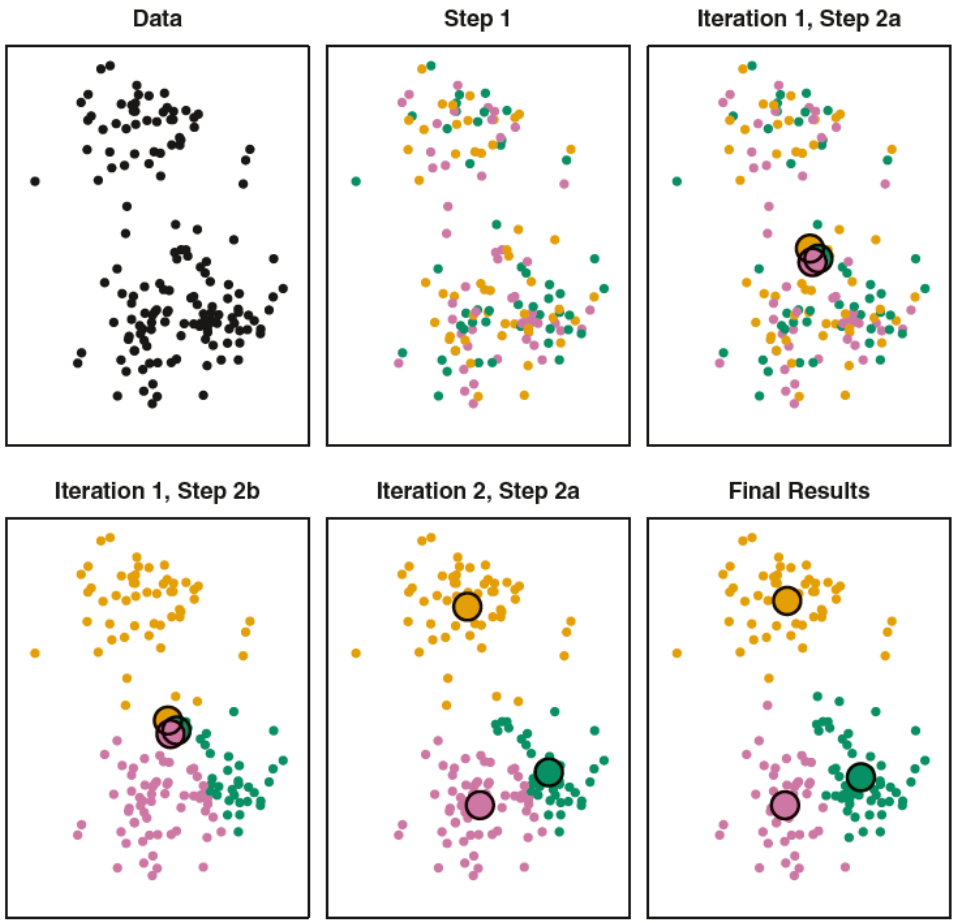
K-means algorithm

- The results obtained after ten iterations
- Stopping criteria: the cluster composition stops changing



K-means algorithm

1. Randomly assign a number to each observation, initial cluster assignments
2. Iterate until cluster assignments stop changing
   a. For each of the K clusters, compute the centroid.
   b. Assign each observation to the cluster whose centroid is closest using Euclidean distance

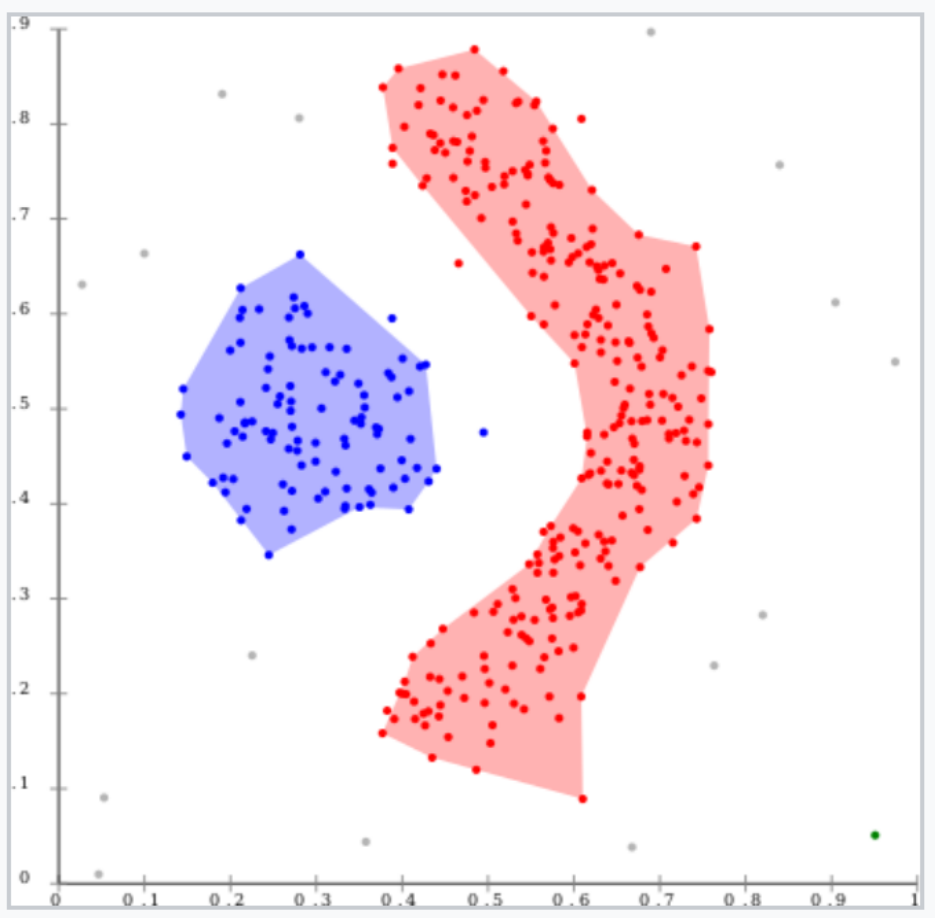


DBSCAN

CLUSTERING METHOD 2

DBSCAN

- Density-Based Spatial Clustering of Applications with Noise
- K-means does not care about the density of data, but DBSCAN does.
- Assumes that regions of high density in your data should be treated as clusters



DBSCAN

1.Choose an arbitrary starting point in your dataset that has not been seen.

2.Retrieve this point's **Є**-neighborhood (all points that are within a distance **Є** from it), and if it contains at least ***min_samples**, a cluster is started.

3.Otherwise, the point is labeled as an outlier (-1). **Note: This point might later be found in a sufficiently sized ε-environment of a different point and hence be made part of a cluster.**

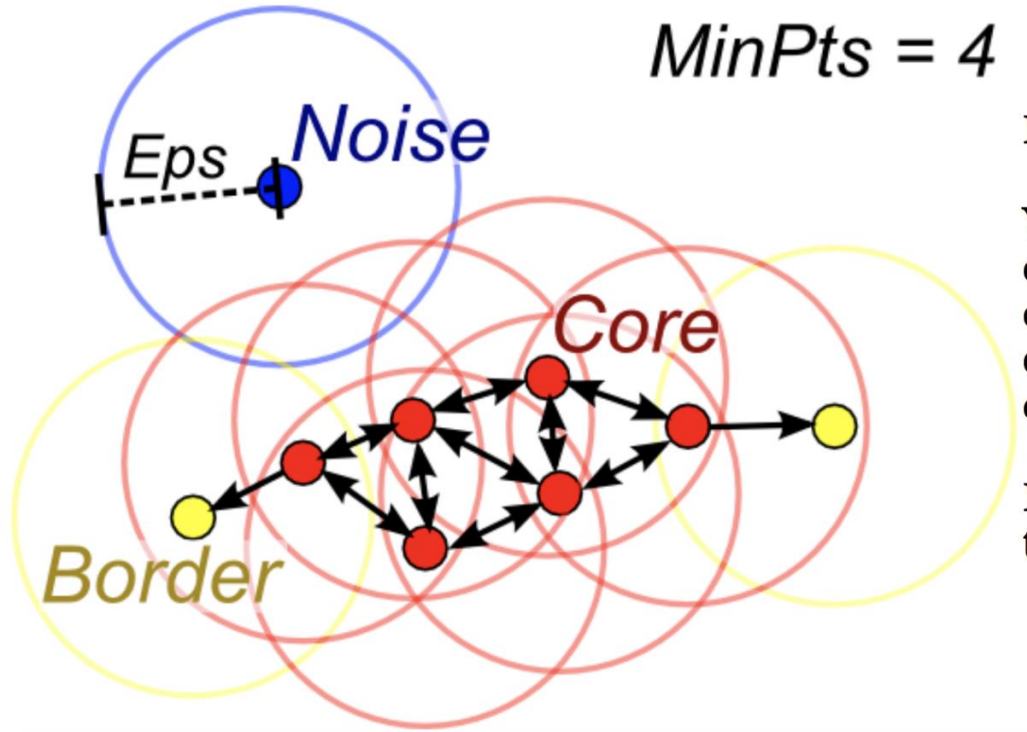
4.If a point is found to be a dense part of a cluster, its **Є**-neighborhood is also part of that cluster. All points that are found within the **Є**-neighborhood are added, as is their own **Є**-neighborhood when they are also dense.

5.Continue until the density-connected cluster is completely found.

6.Find a new unvisited point to process, rinse and repeat.

DBSCAN

- Parameters are
  - Epsilon
  - MinPts



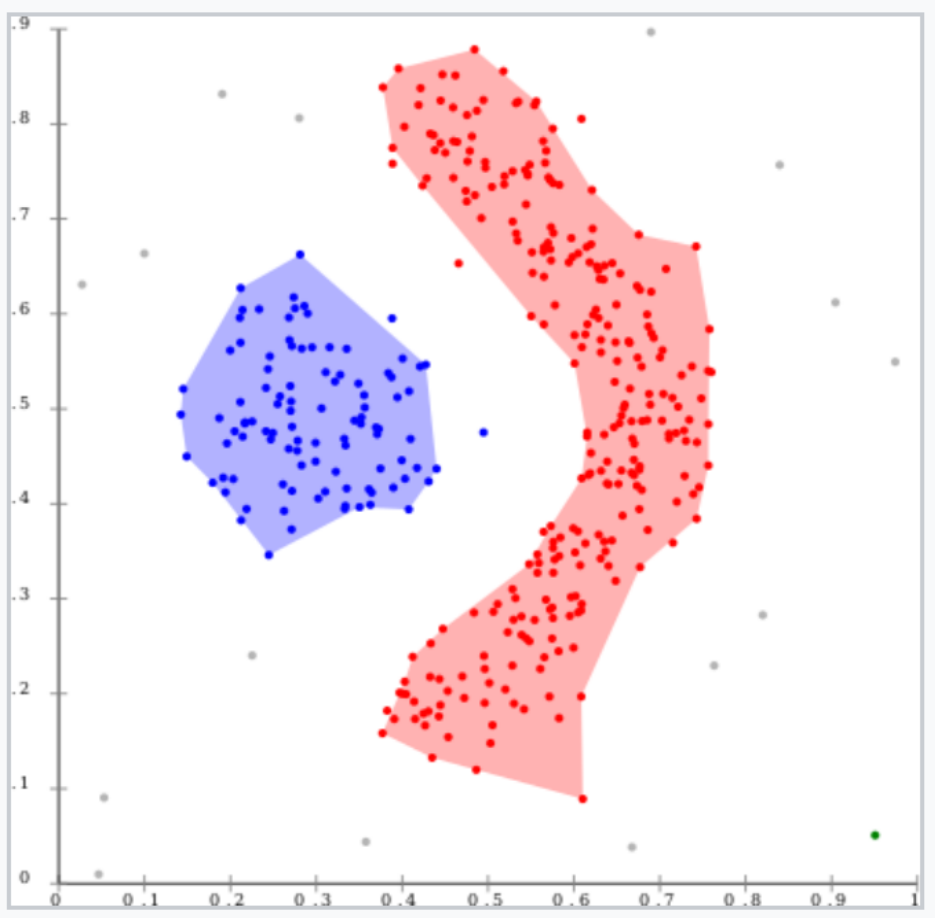
Red: Core Points

Yellow: Border points. Still part of the cluster because it's within epsilon of a core point, but not does not meet the min_points criteria

Blue: Noise point. Not assigned to a cluster

Differences with k-means clustering

- DBSCAN determines the number of clusters automatically, whereas k-means requires the number of clusters as a parameter
- DBSCAN takes any distance metric you want, whereas k-means only works with Euclidean distance



Appendix

Acknowledgements

Material for these slides are taken from but not limited to the following sources:

James, Gareth An Introduction to Statistical Learning

Provost, Foster Data Science for Business

https://medium.com/@elutins/dbscan-what-is-it-when-to-use-it-how-to-use-it-8bd506293818