UNSUPERVISED METHODS

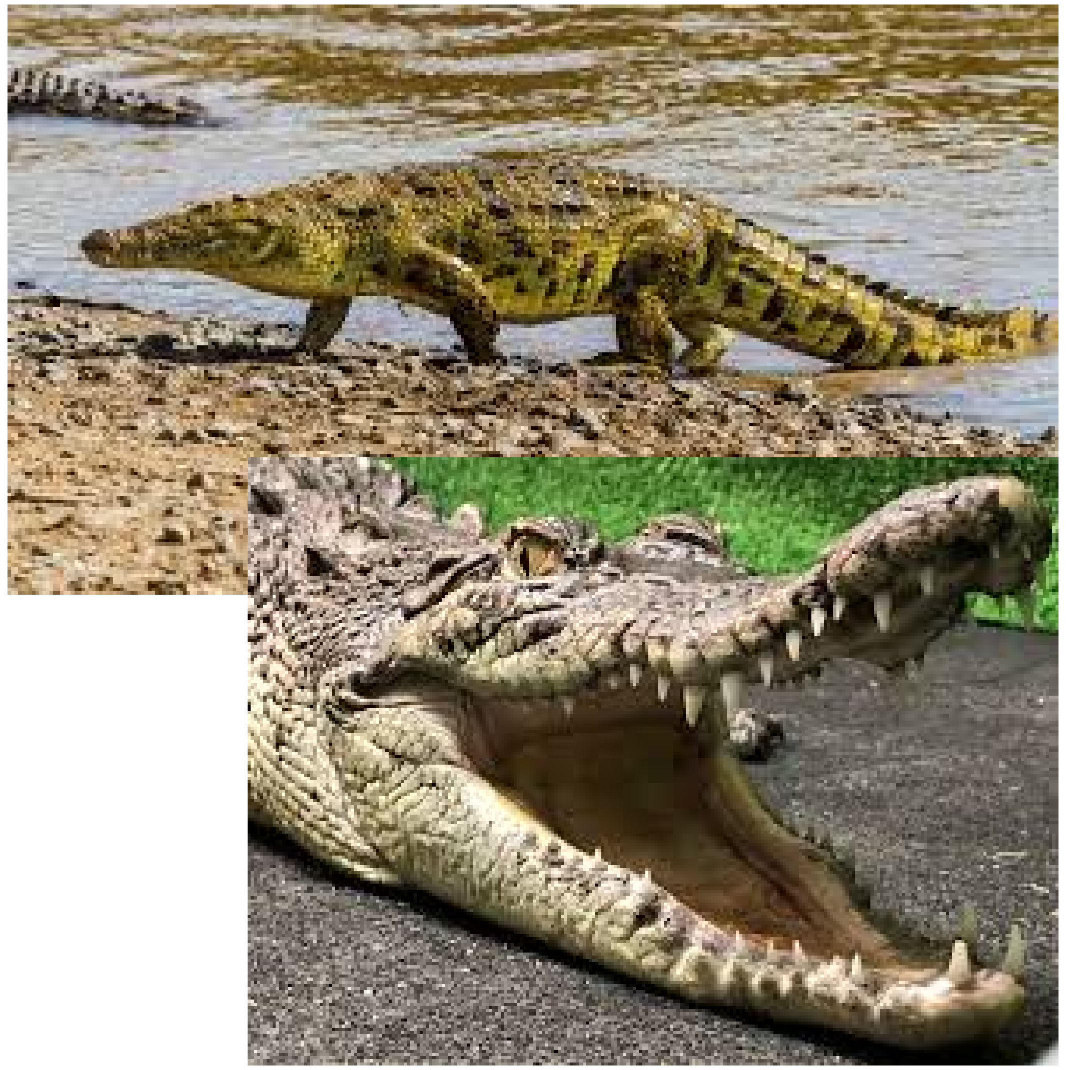
UNSUPERVISED LEARNING

Unsupervised Learning refers to all kinds of machine learning where there is no known output no sort of "labels" that we are using to predict an outcome

Rather, the algorithm is given data and asked to extract knowledge based on any patterns it is able to find

UNSUPERVISED LEARNING

UNSUPERVISED LEARNING



UNSUPERVISED LEARNING

UNSUPERVISED LEARNING

Typical Usecases

- Clustering
    - Topic Identification
    - Image Clustering
    - Customer segmentation
- Dimensionality Reduction
- Outiler Detection
- Generative Modelling

DIMENSIONALITY REDUCTION

Allows us to convert a high dimensional problem into fewer features

Common Dimensionality Reduction Algorithms

- Principal Component Analysis (PCA) - features concentrates variance

- Non-Negative Matrix Factorization (NMF) - features allow reconstruction of original dataset

- SNEs - allows visualizing data as two-dimensional scatter plots

DIMENSIONALITY REDUCTION

Allows us to convert a high dimensional problem into fewer features

Is this the holy grail of feature engineering?

Well... it it is technically useful, reduces noise, improves score and reduces training time
But... it is certainly not interpretable and you often lose any realistic chance of of using domain level knowledge

CLUSTERING TECHNIQUES

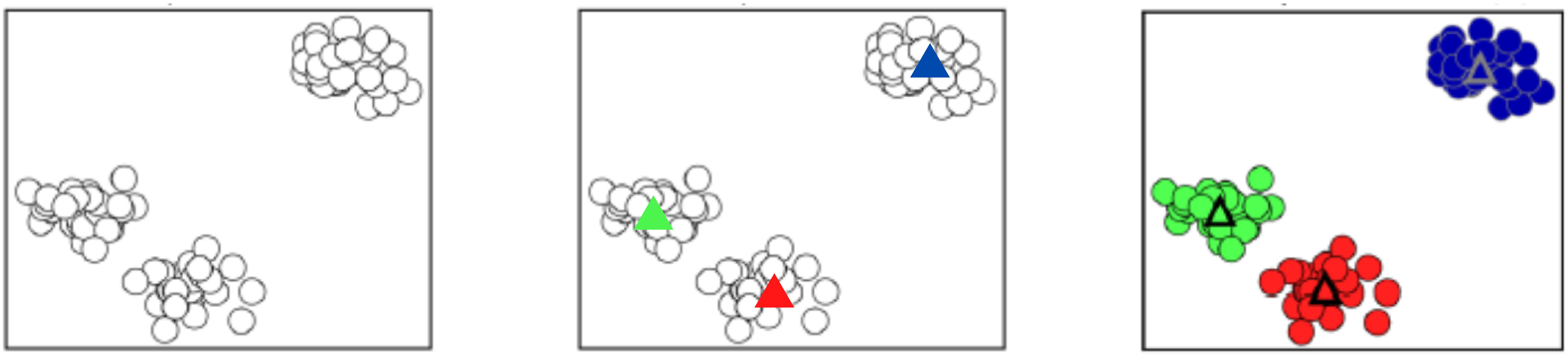
Clustering is the task of partitioning the dataset into groups of similarity called clusters

Clustering algorithms assign a number to each datapoint indicating which cluster it belongs to. It is up to us to interpret this output as being relevant and how.

K-MEANS CLUSTERING

K-means tries to group (cluster) points that are close together.

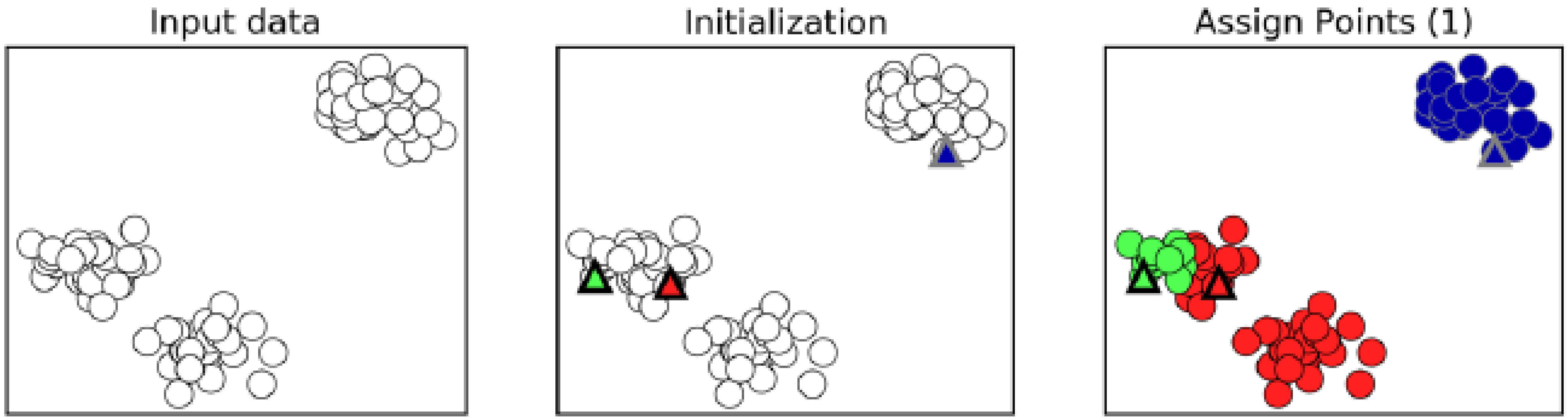
It assumes that each cluster has a center (called centroid). If you are closer to that centroid than any other, you are part of that cluster



But… how to find the centroids?

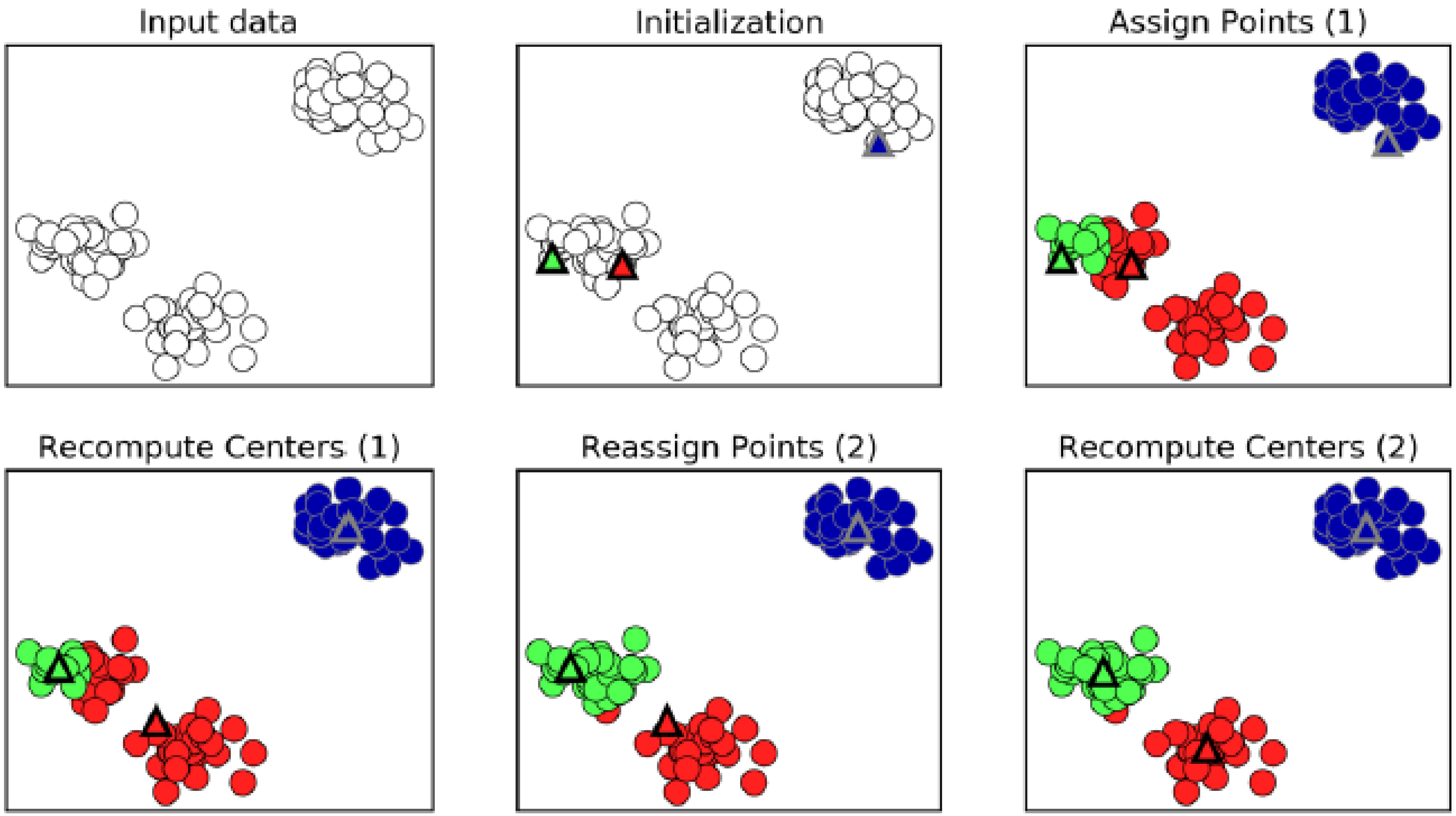
K-MEANS CLUSTERING

Why not random?

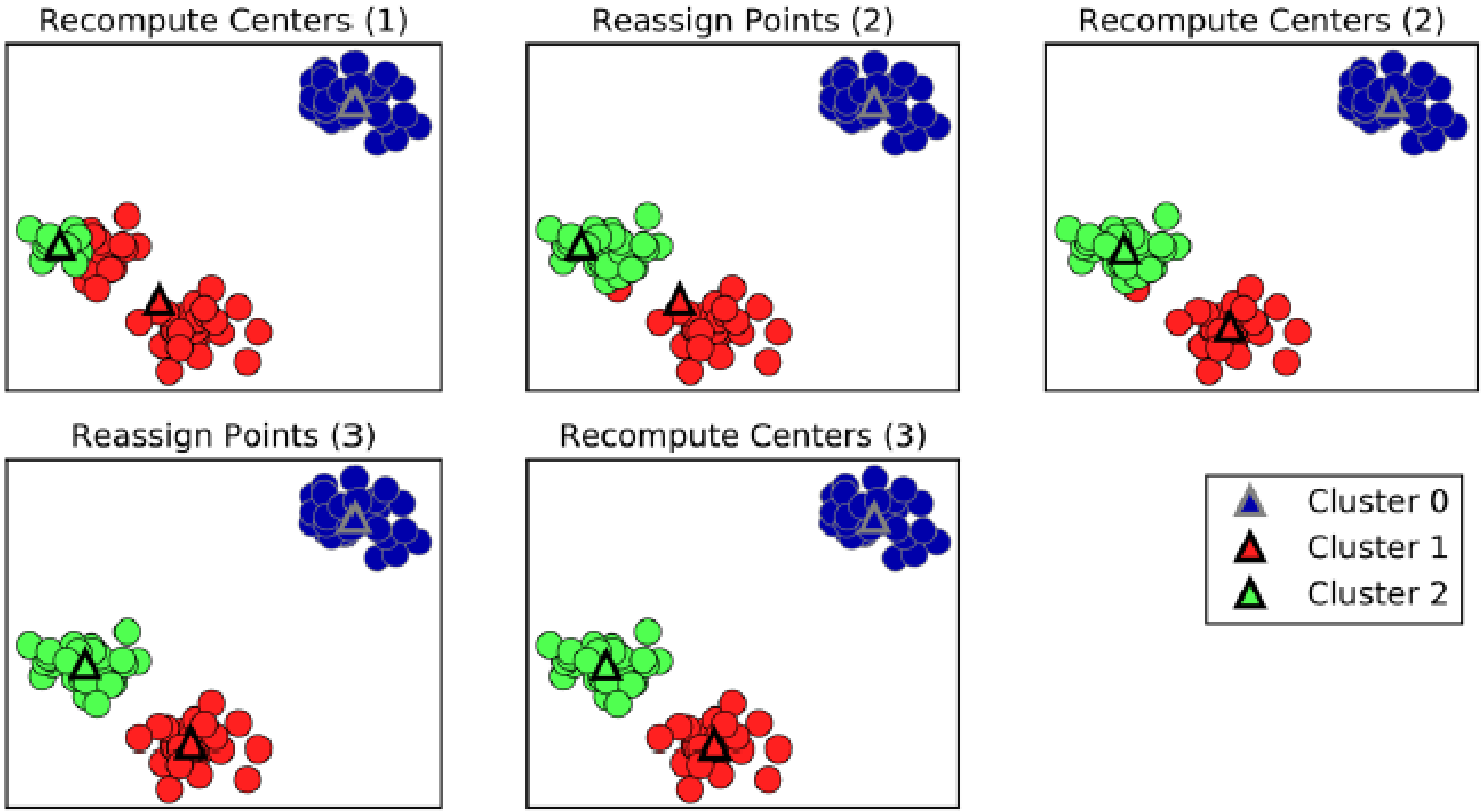


Okay, not great, but now we can make it better

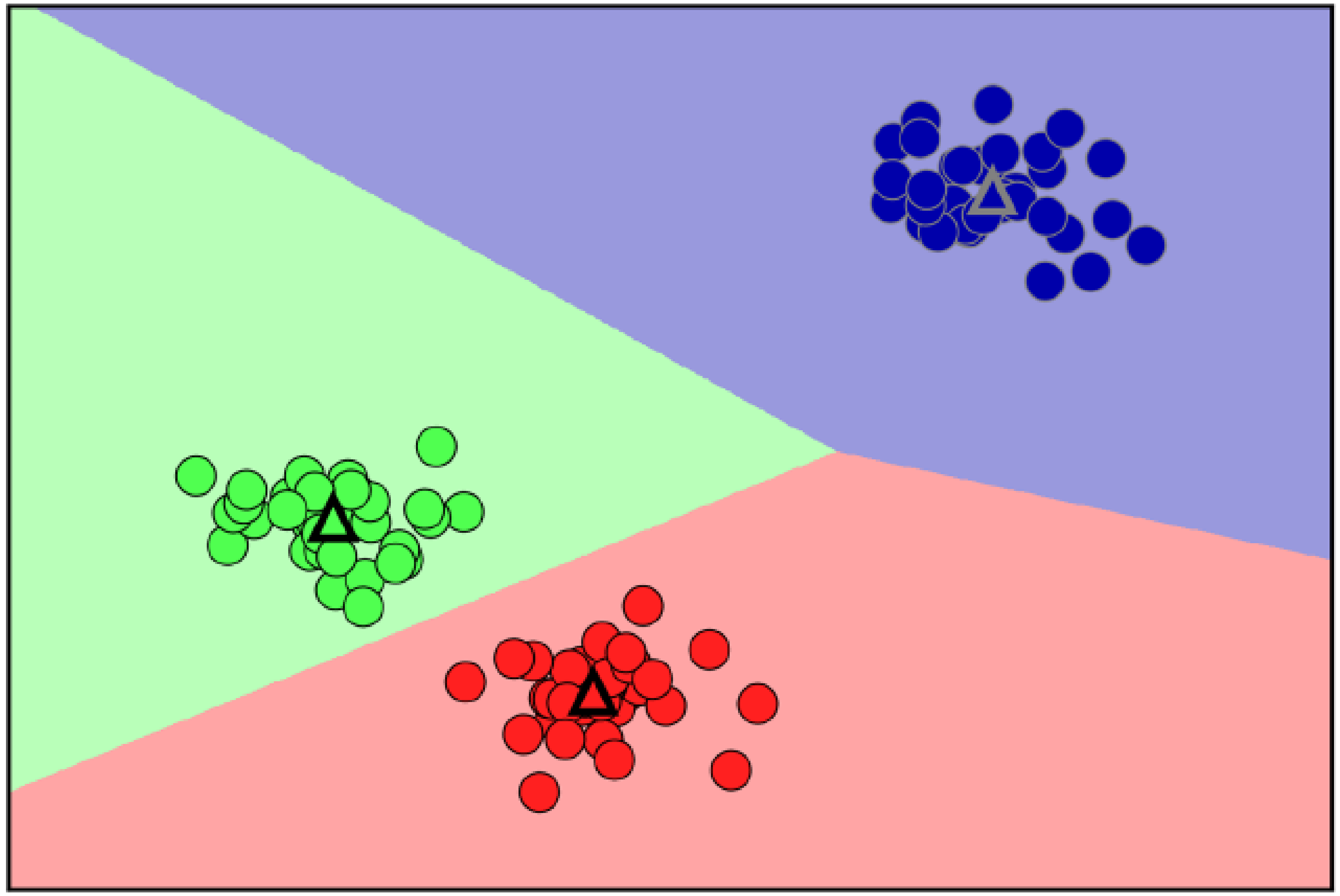
K-MEANS CLUSTERING



K-MEANS CLUSTERING



K-MEANS CLUSTERING



In the end you are left with areas that identify in which a cluster a newly assigned point would be classified.

These are known as the Voronoi Regions

Regions identify the closest centroid center to each point

To the collab...

EVALUATING UNSUPERVISED LEARNING

What are good clusters?

EVALUATING UNSUPERVISED LEARNING

What are good clusters?

Real answer: depends on your problem

- **Is separation between clusters important? (think offer differentiation)**
- **Is size of clusters relevant? (think size of market share)**
- **Is tightness of clusters a factor? (think offer design)**
- **...**

EVALUATING UNSUPERVISED LEARNING

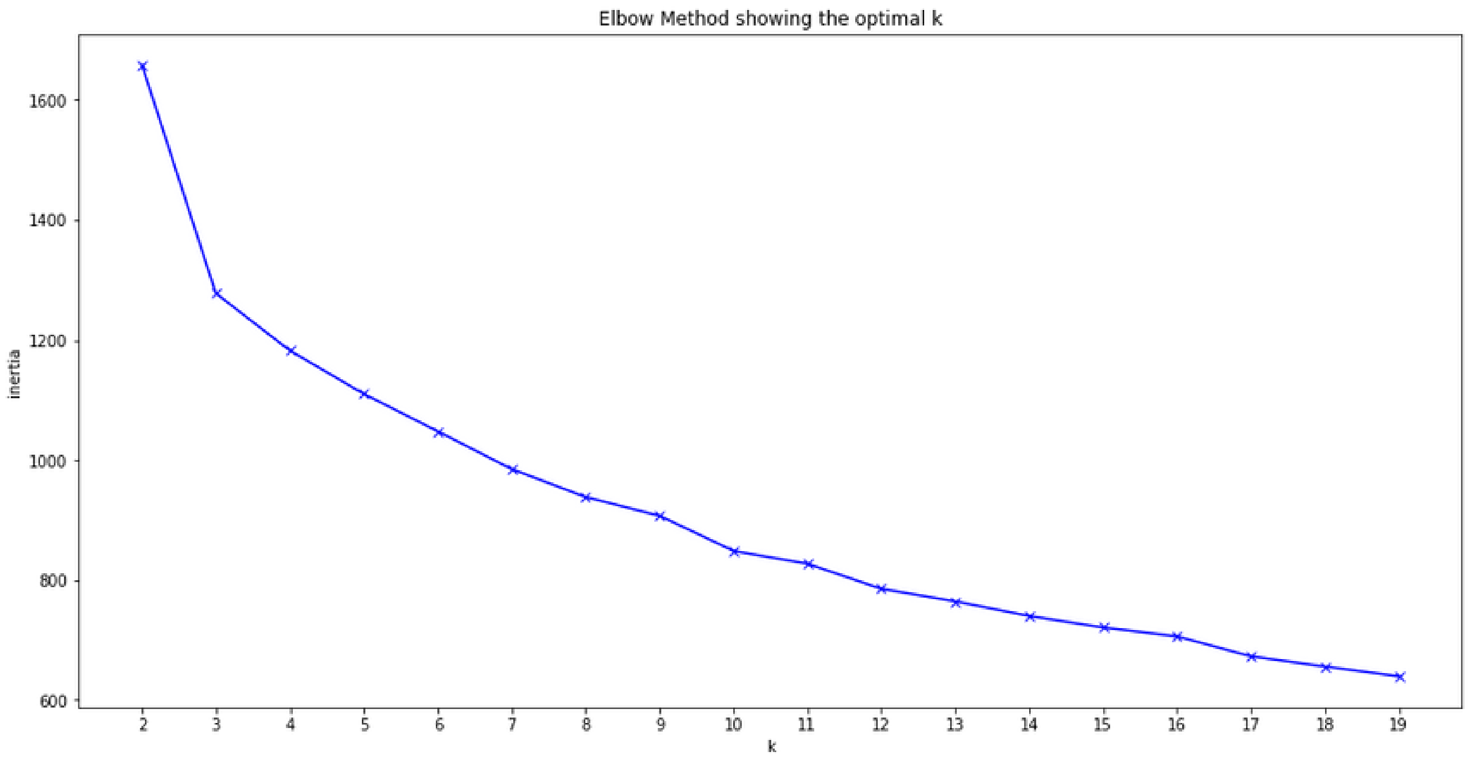
What are good clusters?

Technical answer:

- **Clusters have points tightly packed together**
- **Clusters are far away from each other**

INERTIA SCORE

mean squared distance between each instance and its closest centroid

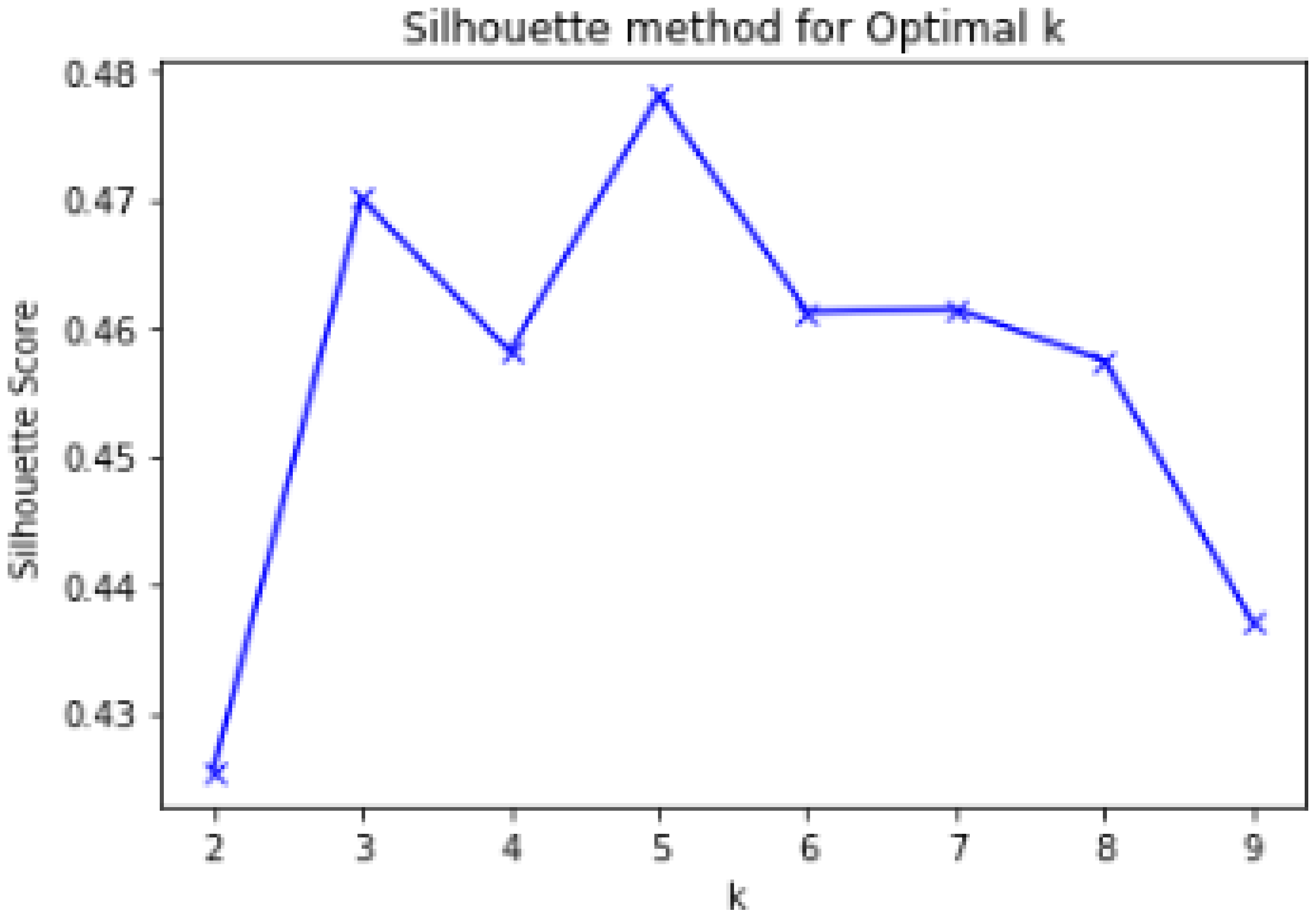


SILHOUETTE SCORE

$$s = \frac{b - a}{max(a, b)}$$

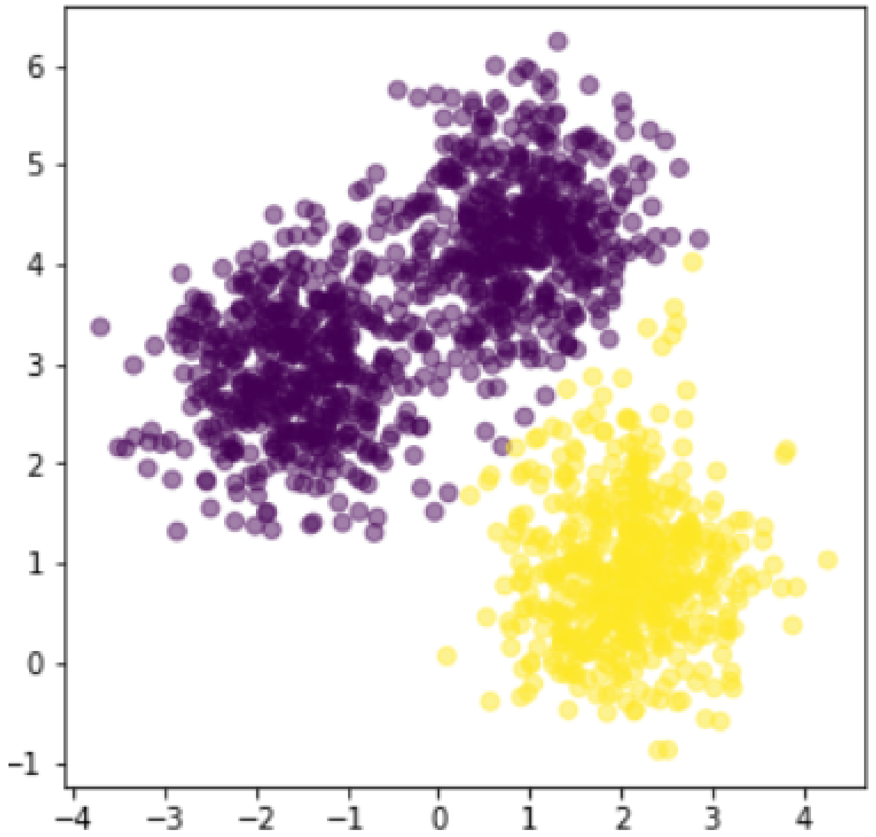
a: mean distance between a sample point and all other points in the same cluster

b: mean distance between the sample and all other points on the nearest cluster

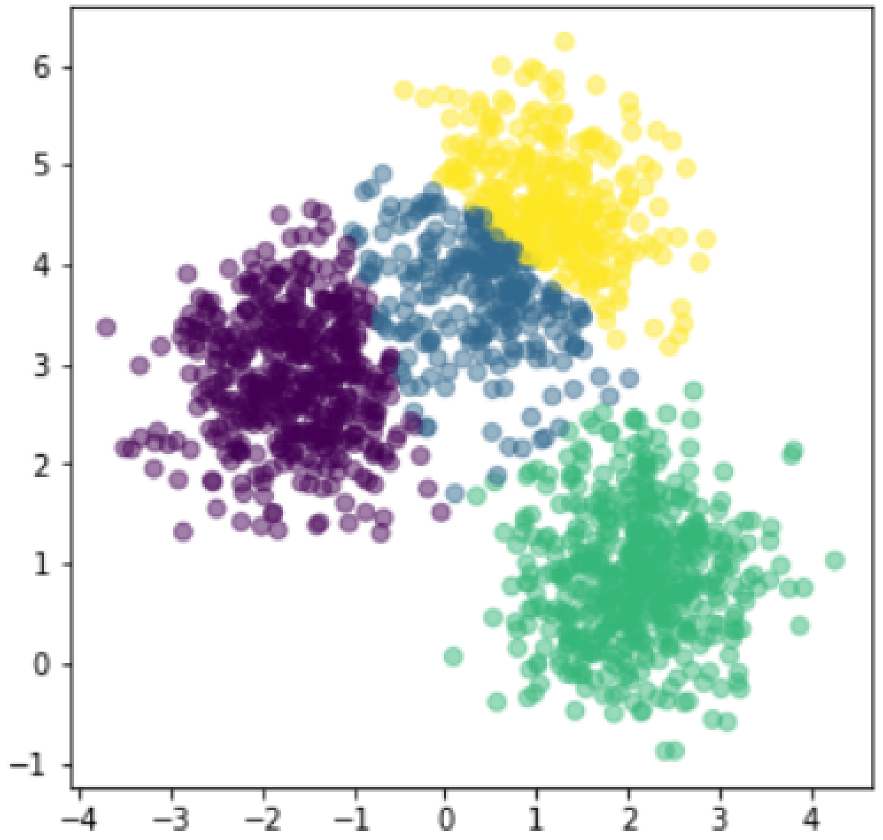


OPTIMIZING NUMBER OF CLUSTERS

Getting the right number of clusters is quite important



Too few clusters and we do not catch meaningful separations in the data

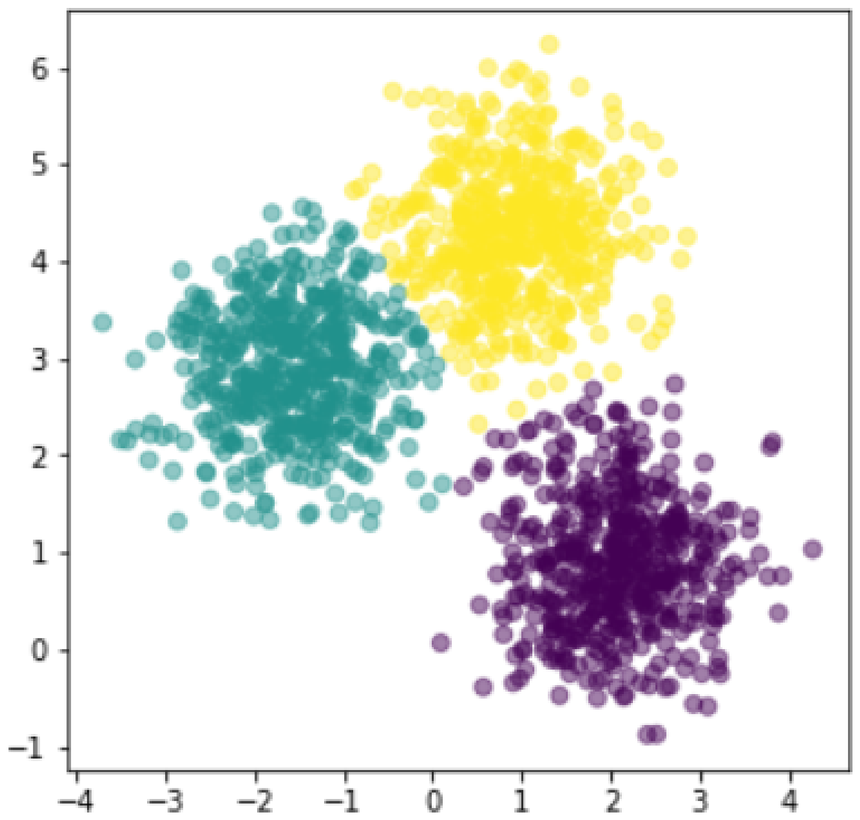


Too many clusters and we separate data that should be together

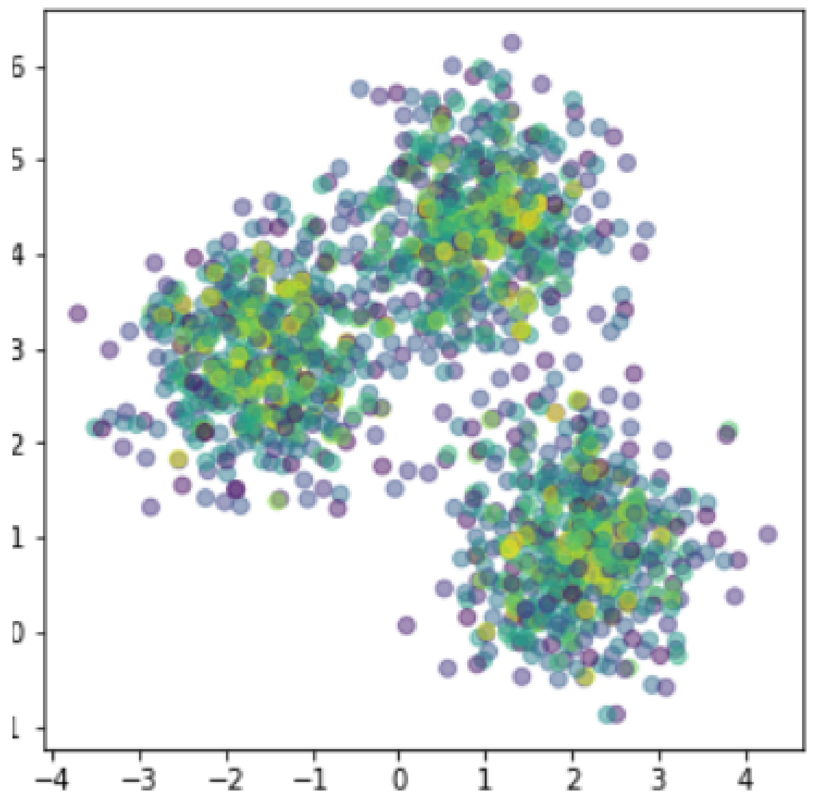
OPTIMIZING NUMBER OF CLUSTERS

Often we plot some metric of error versus the number of clusters...

... but lowest error comes from non-helpful numbers of clusters



Error = 5.56
K = 3



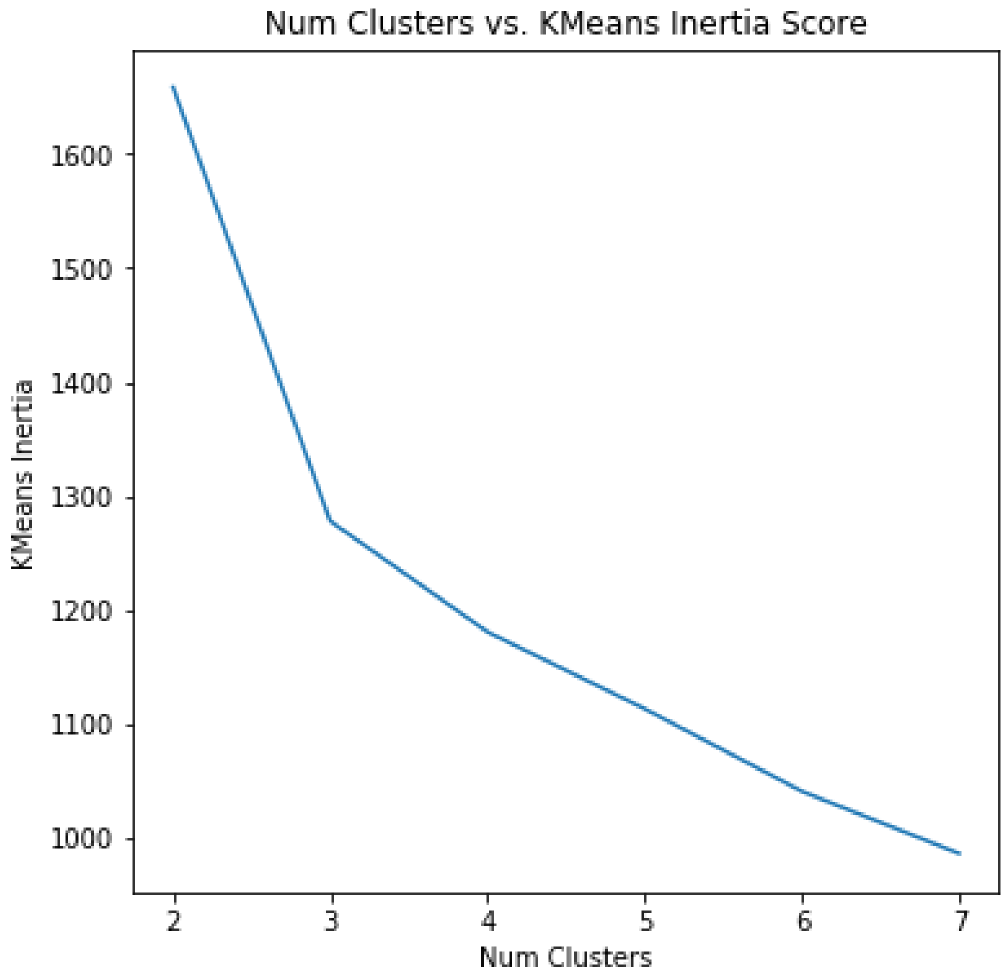
Error = 0
K = 1500

Error = average sum of square distance

OPTIMIZING NUMBER OF CLUSTERS

So what we need is to find a good tradeoff between number of clusters and error metrics

We plot the error versus the number of clusters...

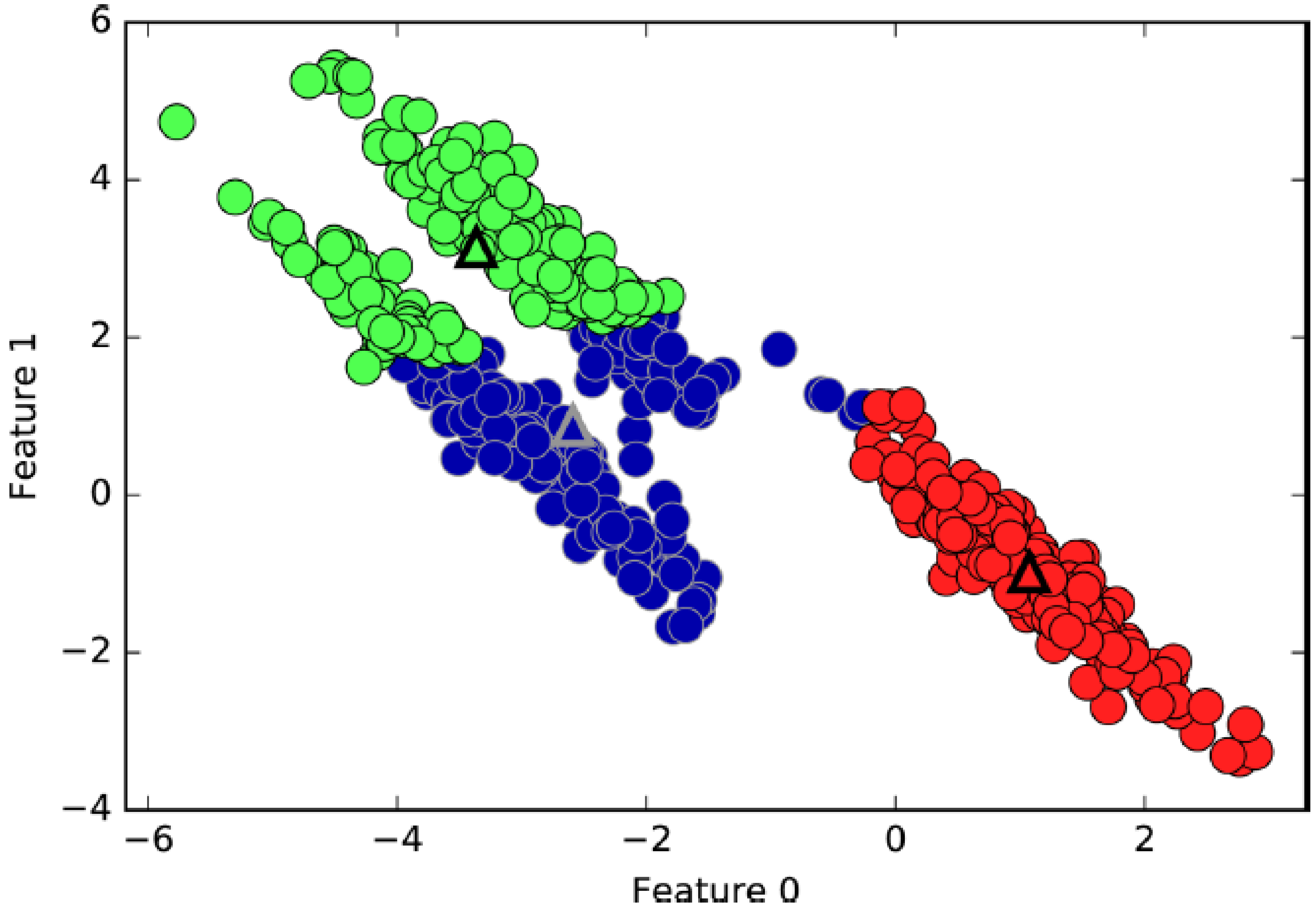


... and try to find where we start to get diminishing returns

visually, we look for where the "elbow" of the curve bends

To the collab...

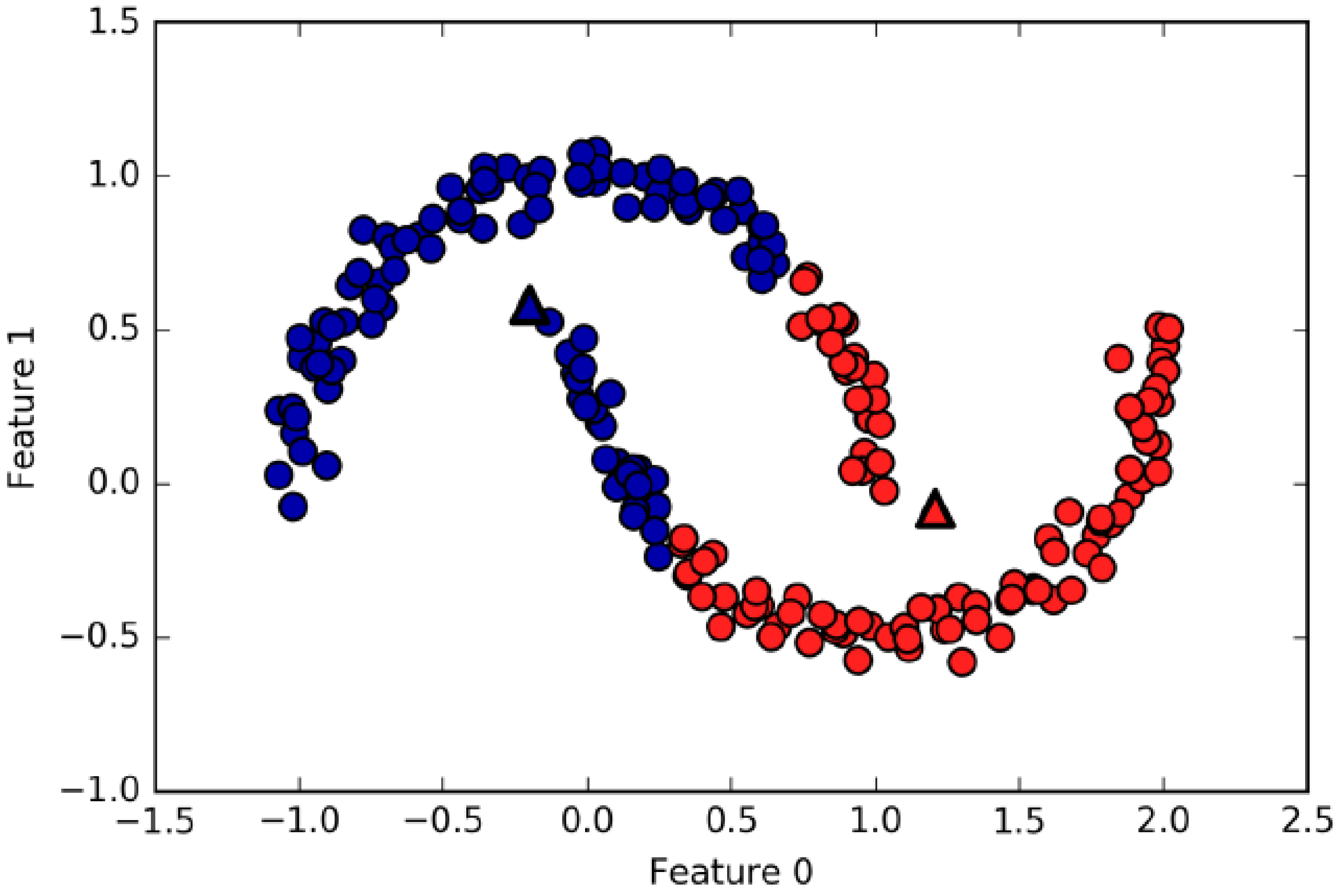
K-MEANS CLUSTERING - LIMITATIONS



In this image, the diagonal direction is privileged over the others, but since "distance" does not care about direction, k-means fails.

... this would be a good case to use a PCA, by the way.

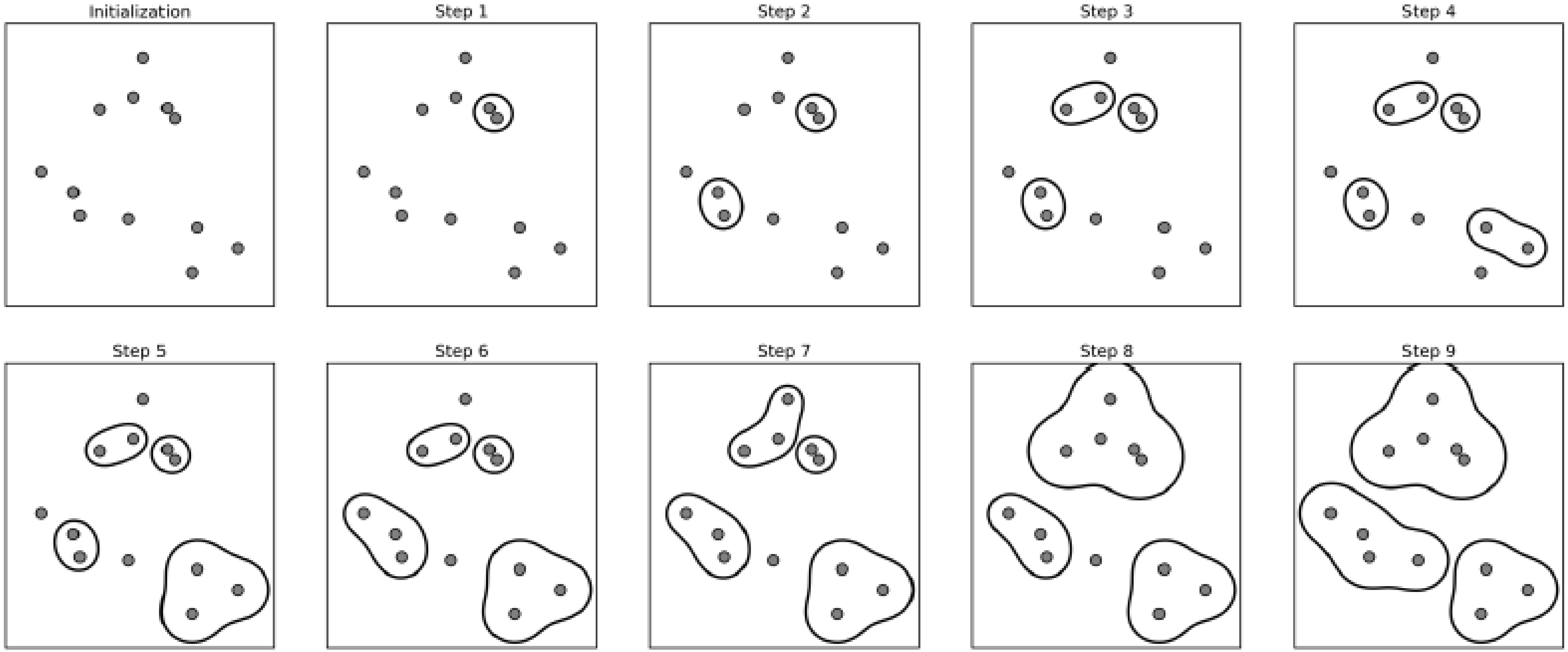
K-MEANS CLUSTERING - LIMITATIONS



In this coordinate system, the clusters are found in complex shapes which makes it harder for the kmeans to correctly identify them

Remember, k-means assumes that each cluster has a center

AGGLOMERATIVE CLUSTERING



AGGLOMERATIVE CLUSTERING

The algorithm starts by declaring each point as its own cluster

Then a criteria of merging between clusters is iteratively applied until the desired number of clusters is reached

There are several criteria of linkage that specify what are the two most similar clusters to merge

AGGLOMERATIVE CLUSTERING

There are several criteria of linkage that specify what are the two most similar clusters to merge

`ward`

The default choice, `ward` picks the two clusters to merge such that the variance within all clusters increases the least. This often leads to clusters that are relatively equally sized.

`average`

`average` linkage merges the two clusters that have the smallest average distance between all their points.

`complete`

`complete` linkage (also known as maximum linkage) merges the two clusters that have the smallest maximum distance between their points.

To the collab...

DBSCAN

Until now all methods we studied assumed a certain number of clusters/centroids defined à priori

DBSCAN - Density Based Spatial Clustering of Applications with Noise - does not require the user to set the number of clusters.

It works by identifying regions that are "crowded" or of high density

Is able to identify clusters of complex shapes and points that are not part of any cluster

As a consequence it looks for dense regions followed by relatively empty regions

https://cdn-images-1.medium.com/max/1600/1*tc8UF-h0nQqUfLC8-0ulnQ.gif

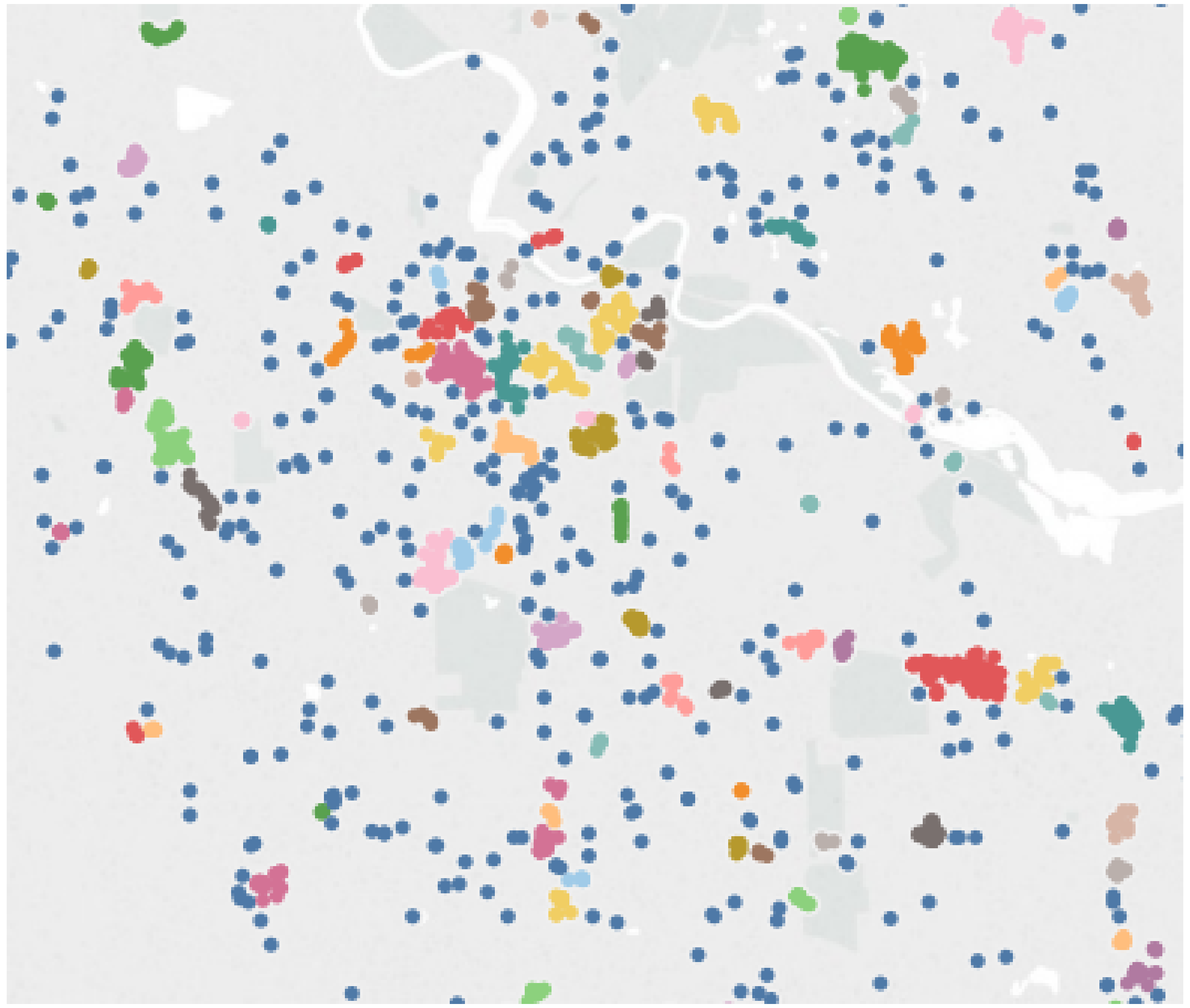
DBSCAN

Requires two parameters:

eps: how close points should be to each other to be considered part of a cluster

minPoints: the minumum number of points to form a dense region. E.g. if equals to 5 then we need at least 5 points within a eps distance to be considered a cluster

Any values that do not satisfy the density requirements are considered as non-clustered



ANY QUESTIONS ?