IRON HACK

UNSUPERVISED LEARNING

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

Different clustering algorithms use different methods to define what is 'similar'

UNSUPERVISED LEARNING

Typical Usecases

- Topic Identification (when analysing emails, documents articles)

- Image Clustering

- Retail basket analysis

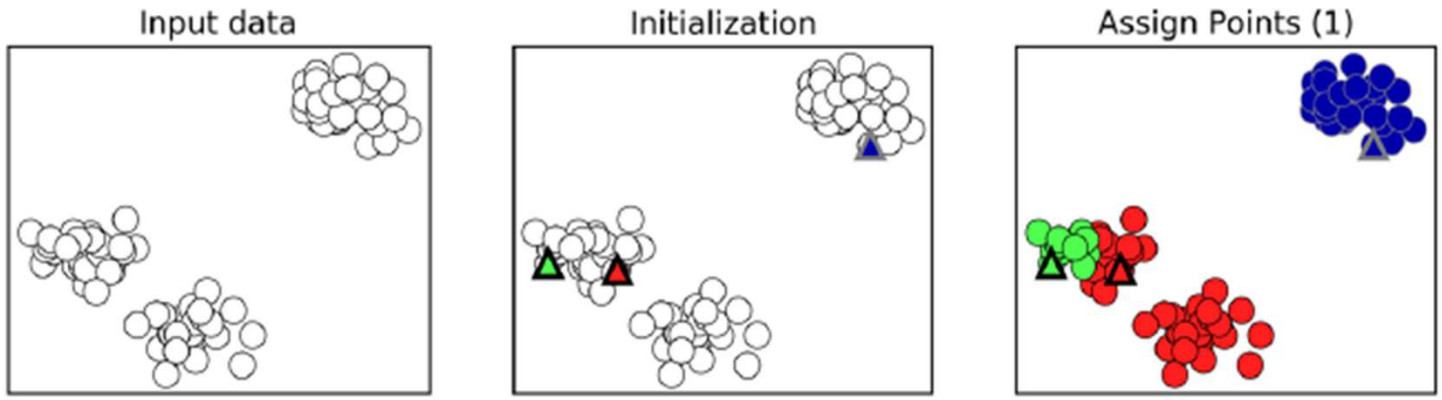
- Dimensionality Reduction

UNSUPERVISED LEARNING

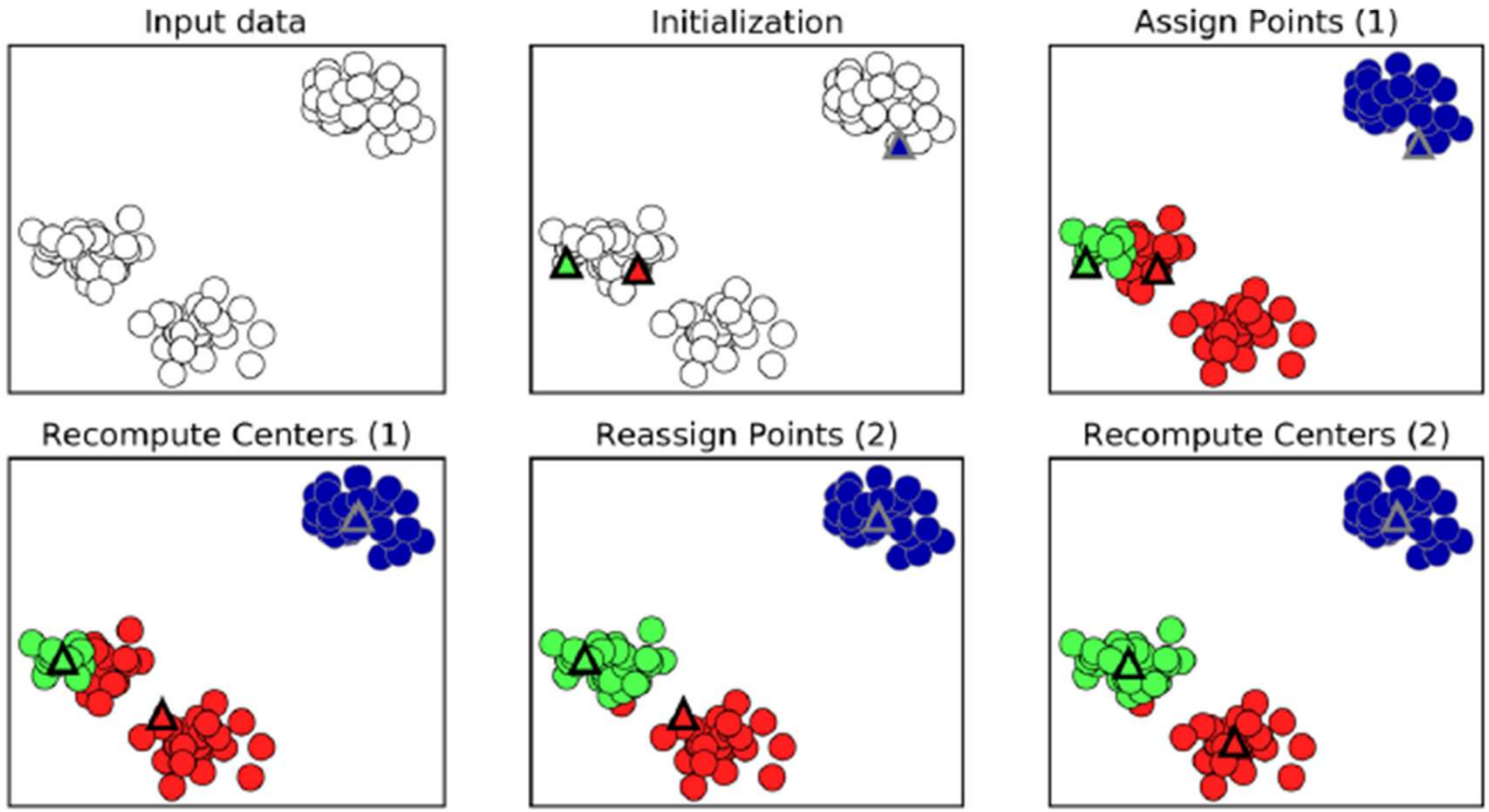
Clustering is the task of partitioning the dataset into groups of similarity called clusters

Clustering algorithms assign a number to each datapoint indicating which cluster it belongs to.

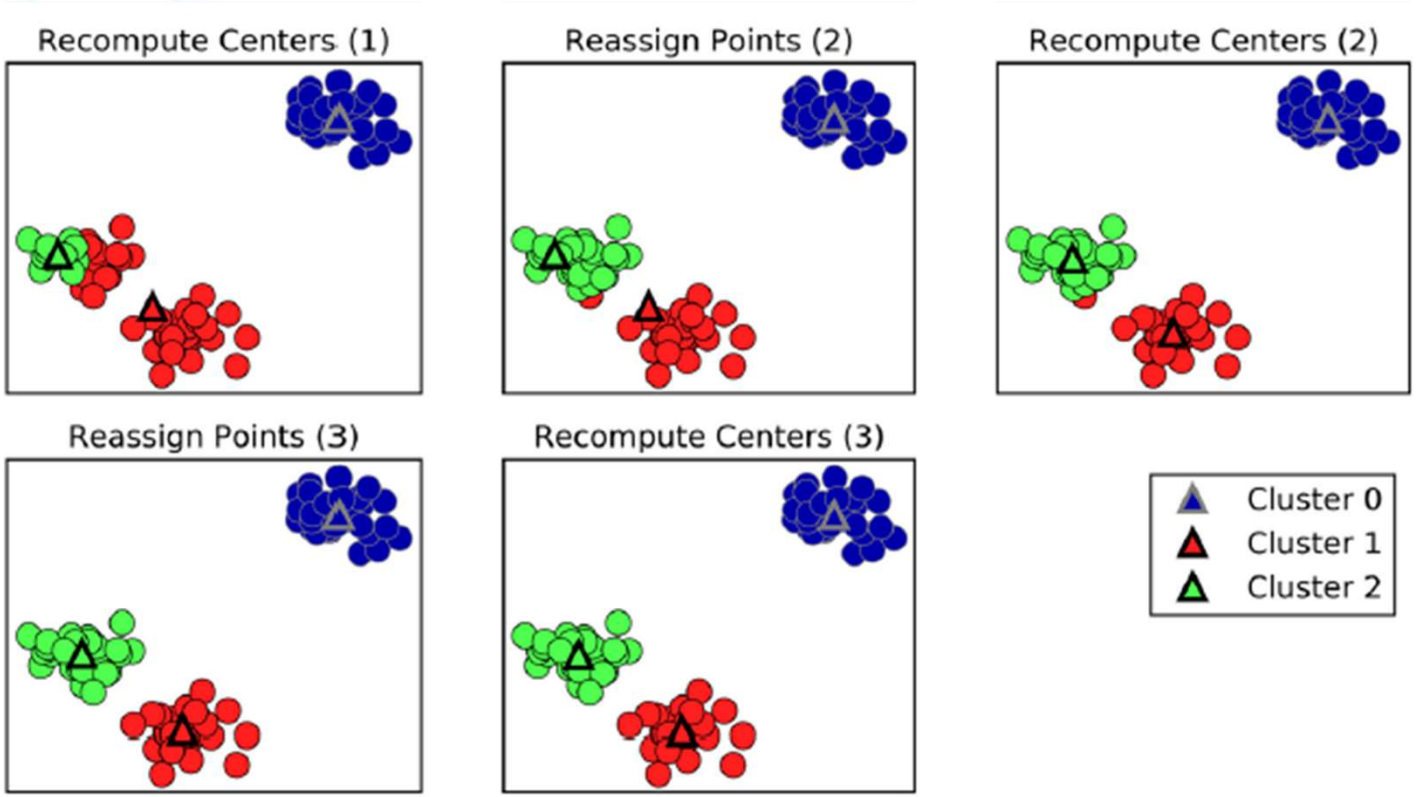
K-MEANS CLUSTERING



K-MEANS CLUSTERING



K-MEANS CLUSTERING

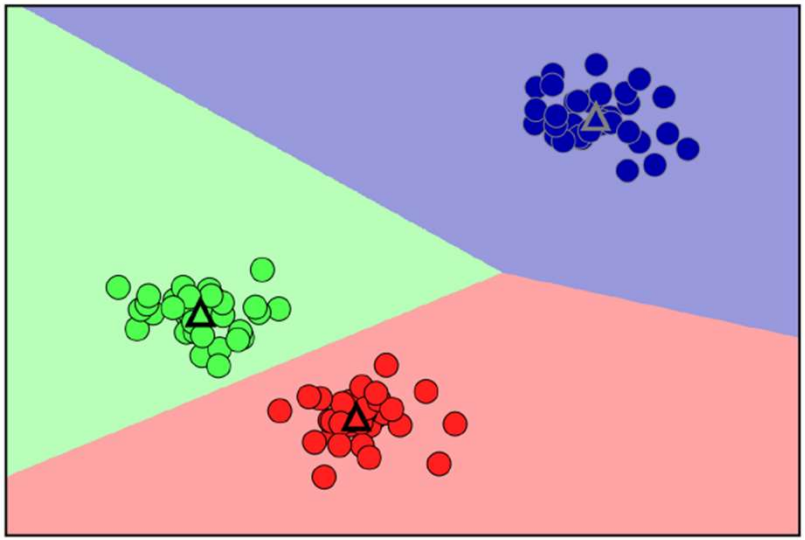


K-MEANS CLUSTERING

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

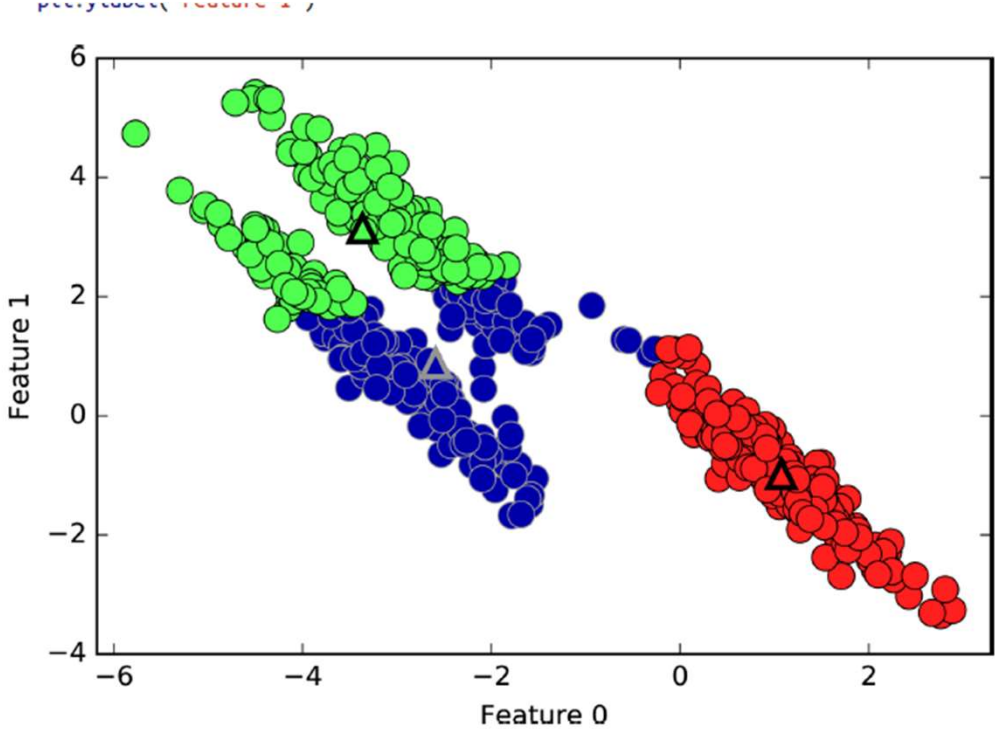
These are known as the Voronoi Regions

Regions identify the closest centroid center to each point

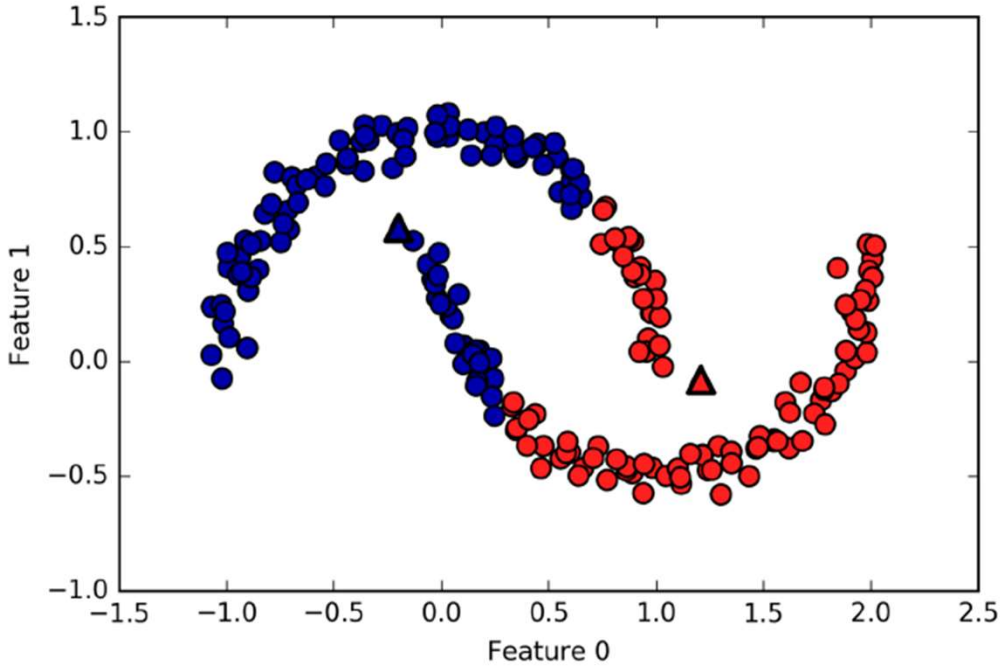


K-MEANS CLUSTERING

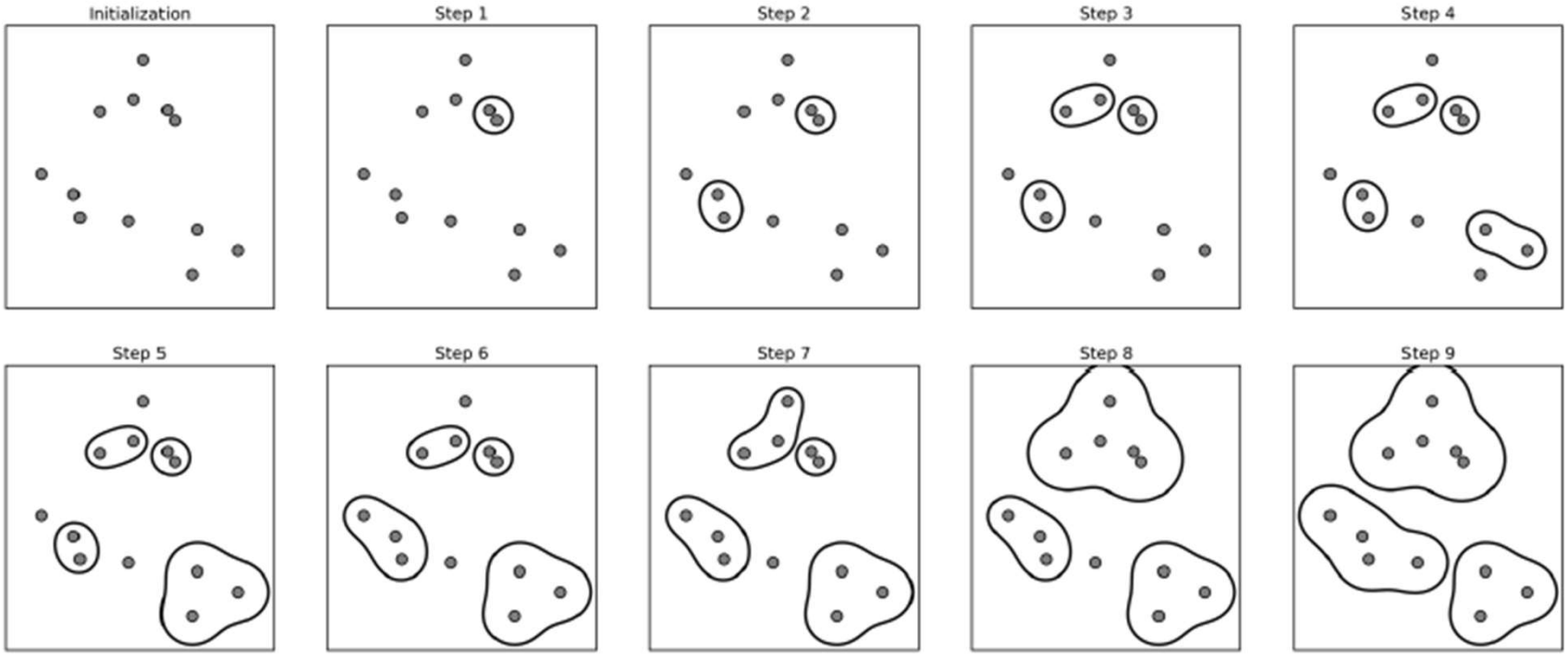
In this coordinate system, the clusters are found in complex shapes which makes it harder for the kmeans to correctly identify them



K-MEANS CLUSTERING - LIMITATIONS



AGGLOMERATIVE CLUSTERING

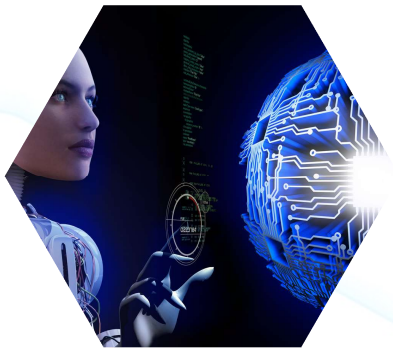


AGGLOMERATIVE CLUSTERING

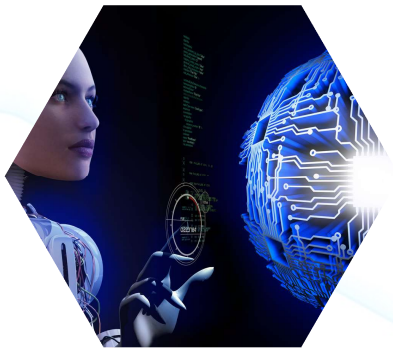
The algorithm starts by declaring each point as its own cluster

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

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

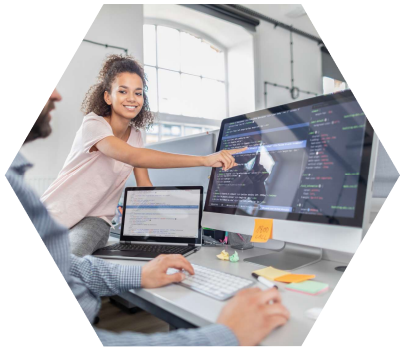


AGGLOMERATIVE CLUSTERING



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.

DBSCAN

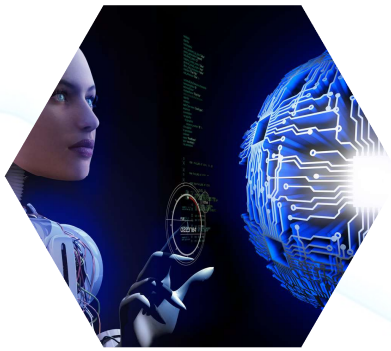
Density Based Spatial Clustering of Applications with Noise

Does not require the user to set the number of clusters "a priori"

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

Works by identifying regions that are "crowded" or of high density

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



DBSCAN

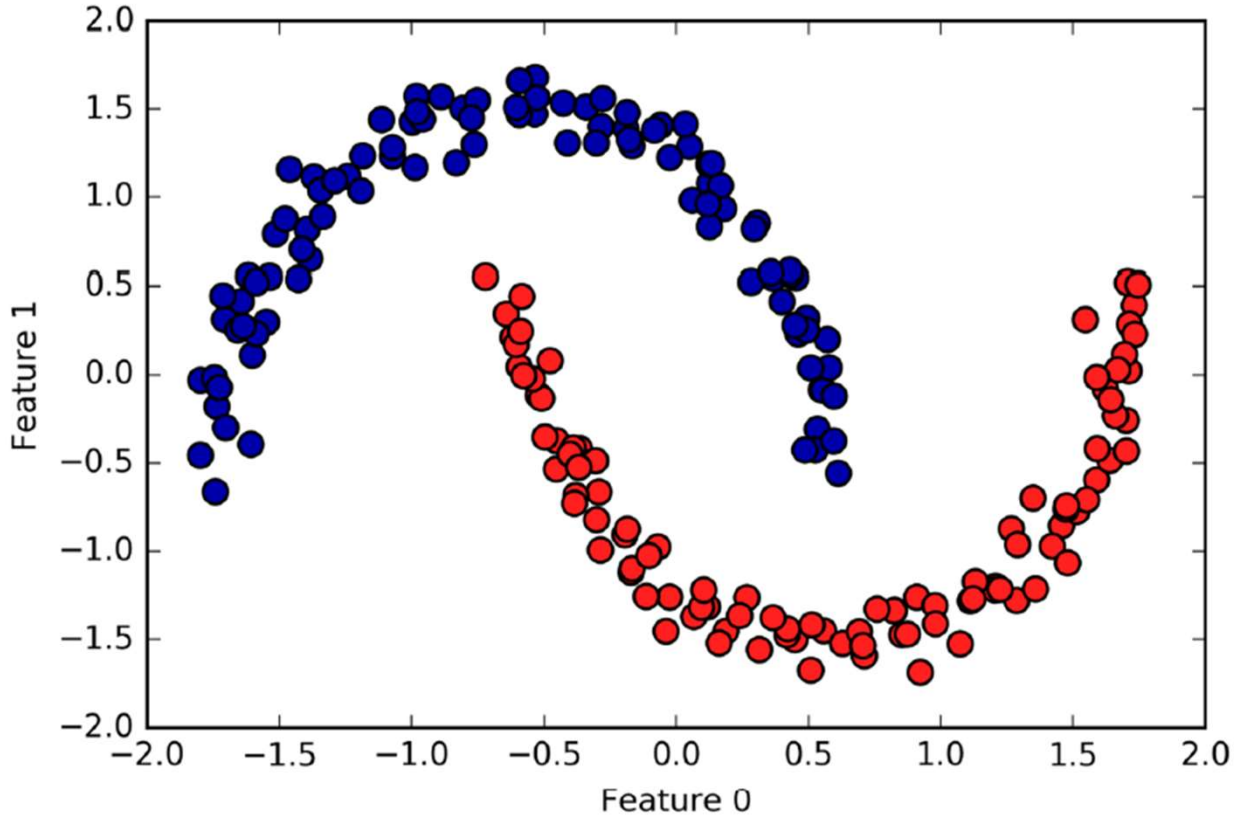
Requires two parameters:

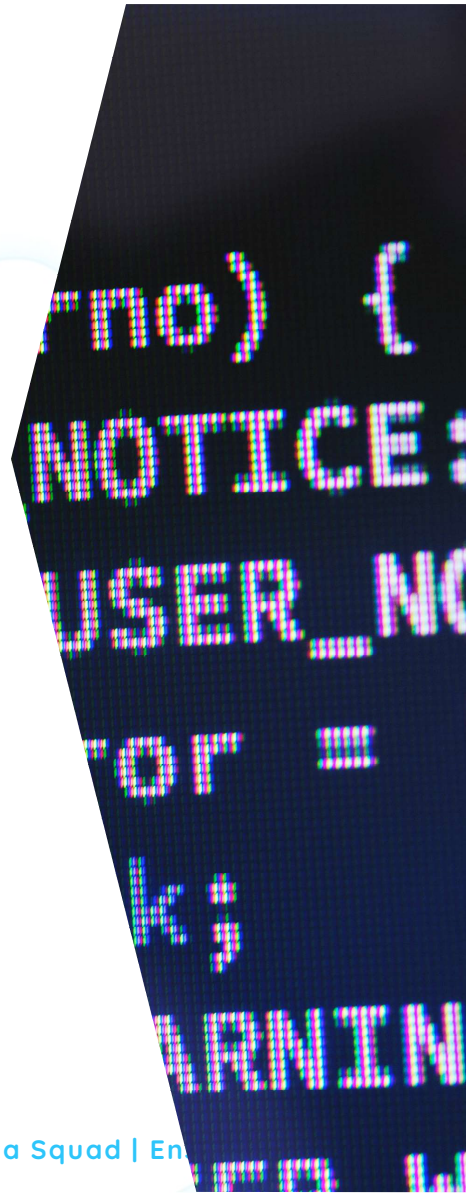
eps: how close points should be to each other to be considered part of a cluster

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

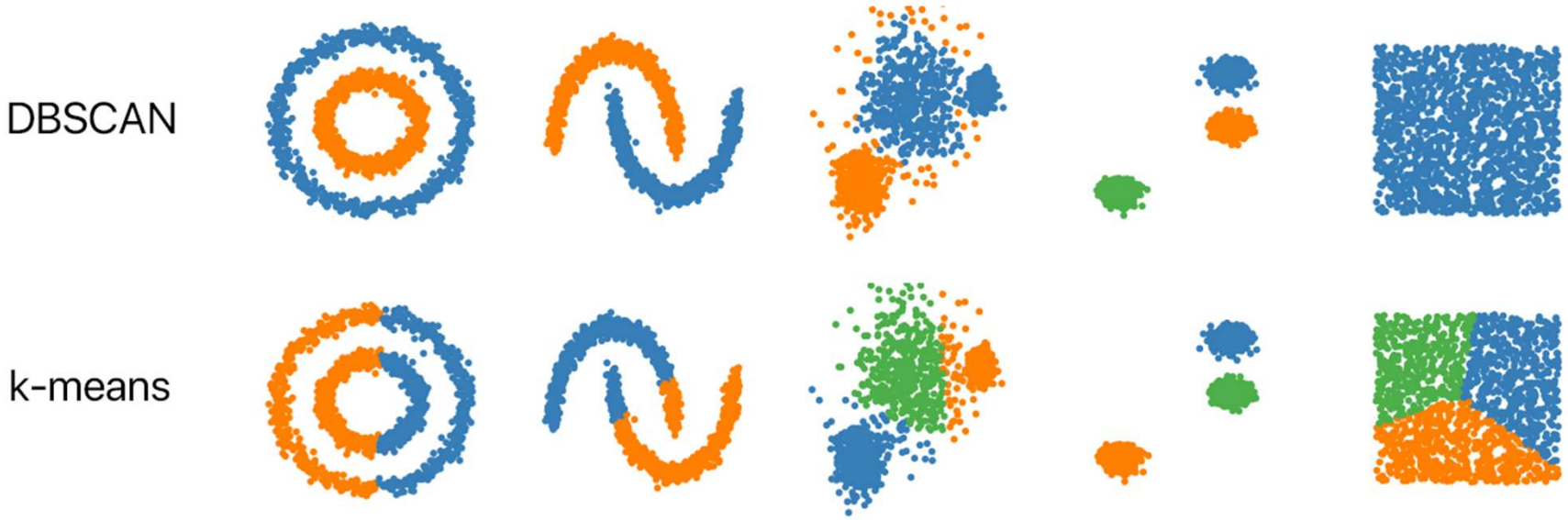
Any values that do not satisfy the density requirements are considered as outliers.

DBSCAN


2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
Feature 1
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
Feature 0



DBSCAN - Demo



EVALUATING UNSUPERVISED LEARNING

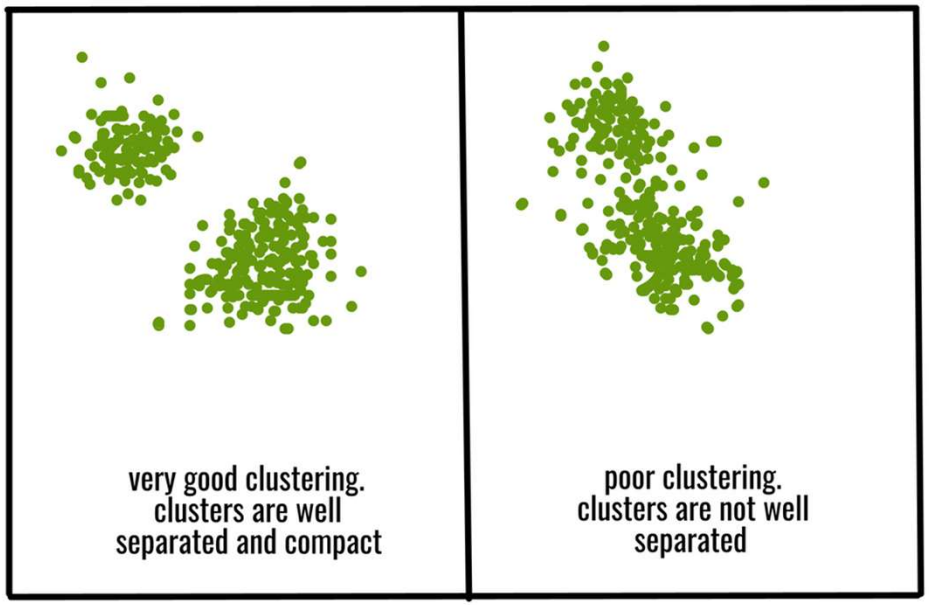
What are good clusters?

EVALUATING UNSUPERVISED LEARNING

What are good clusters?

Clusters have points tightly packed together

Clusters are far away from each other

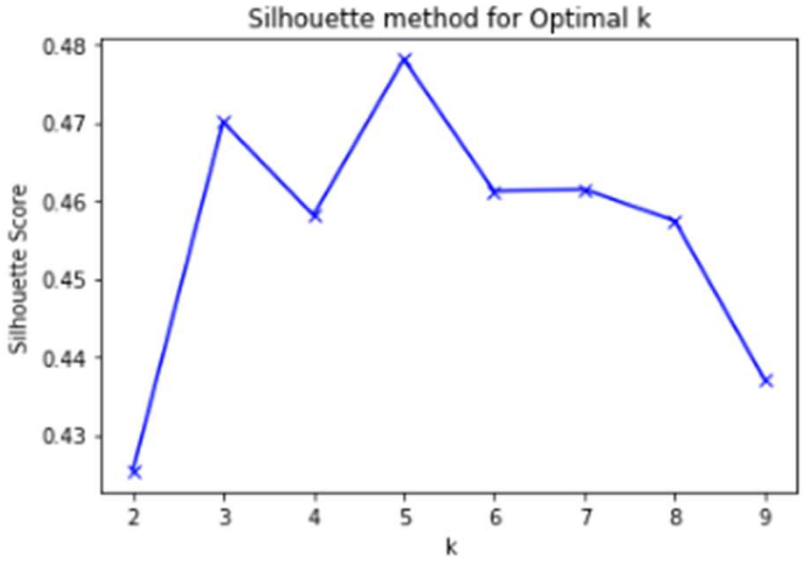


SILHOUETTE SCORE

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

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



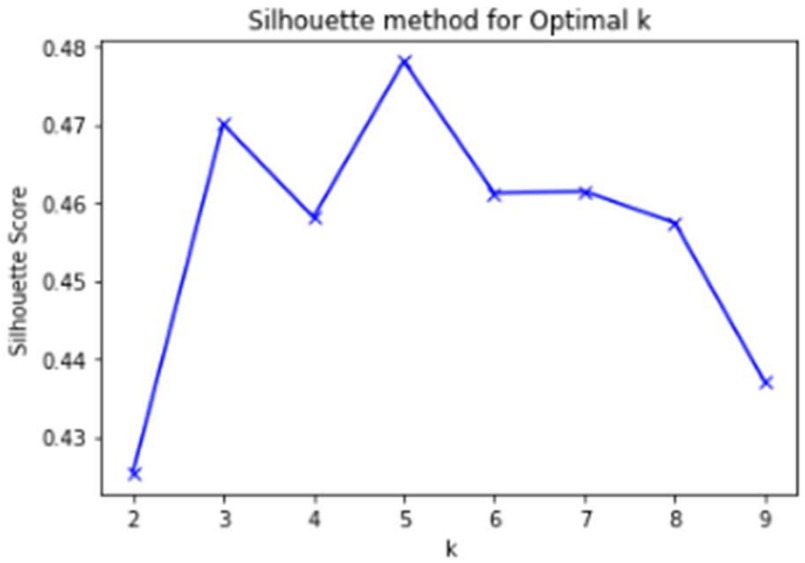
SILHOUETTE SCORE

separation density

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

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

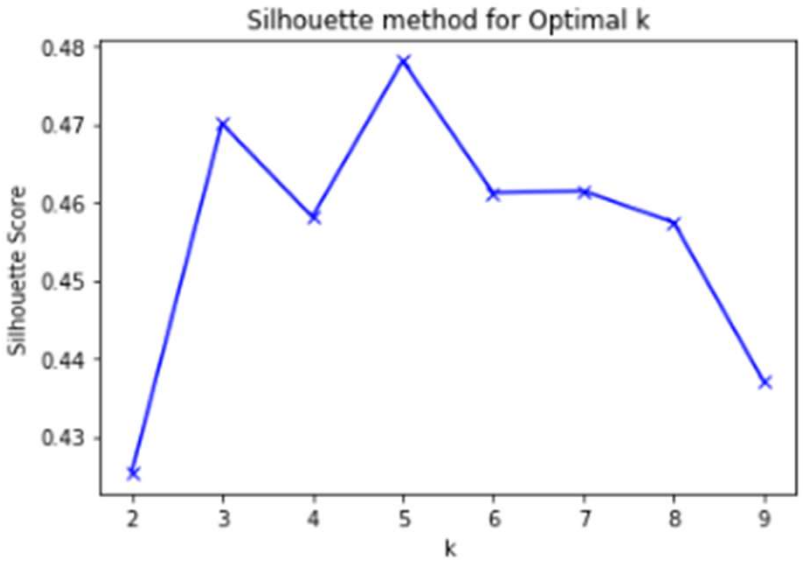
b: mean distance between the sample and all other points on the nearest cluster



SILHOUETTE SCORE

- **The Silhouette score does not say anything about the usefulness of clusters in a particular case. All it says is how well clusters behave in the definitional sense of clusters - e.g. how dense they are and how well separated.**

- **The score is very similar to the metric that algorithms like KMeans seek to optimize, so it can lead to overfitting (we are judging a model based on how well it does the exact thing it was trained to do). Having a hold-out set of data can help with this issue.**



DIMENSIONALITY REDUCTION

Jupyter (4)

DIMENSIONALITY REDUCTION

Allows us to convert a high dimensional problem into fewer features

Common Dimensionality Reduction Algorithms

(this we will see)

- **Principal Component Analysis (PCA) - features concentrates variance**

(these we will not)

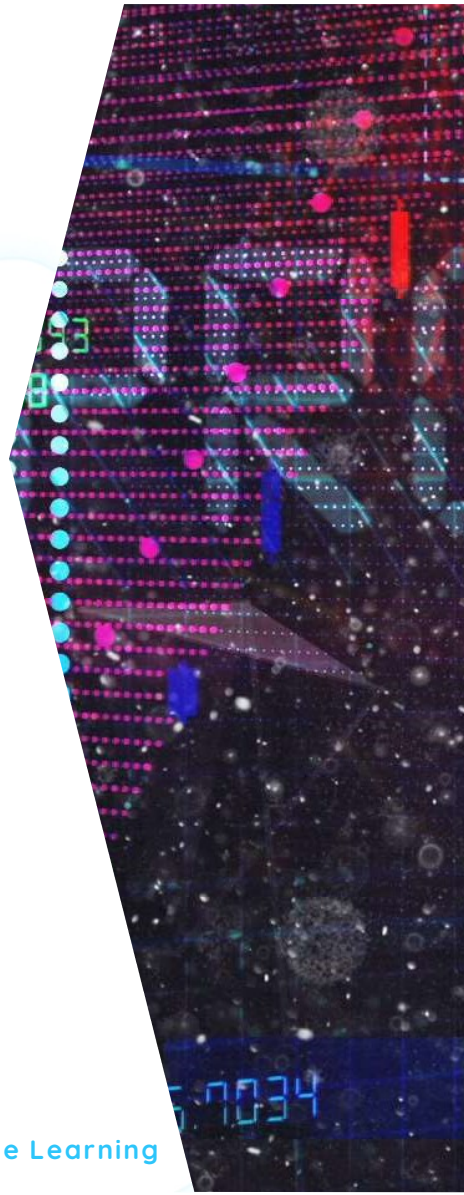
- **non-Negative Matrix Factorization (NMF) (features allow reconstruction of original dataset), SNEs (allows visualizing data as two-dimensional scatter plots), uMaping, etc**

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



PRINCIPAL COMPONENT ANALYSIS

If you could pick a single feature for learning, which one would you pick?

- the one with the most variance, since that is the most likely to carry discriminative information (e.g. salary vs free time/day)

What if you could pick a second one?

- the one with the second most variance.... but that may have an issue if it is too correlated with the first one (e.g. salary vs wealth)

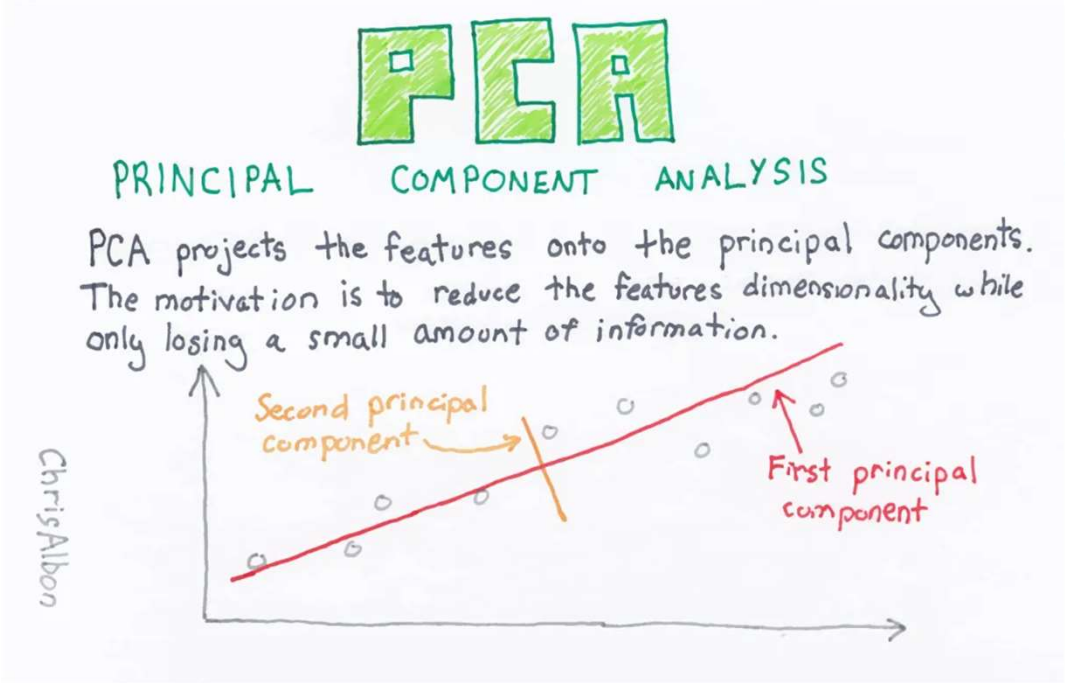
So instead you pick the feature that captures the most variance except for the variance already captured in the first feature

PCA does this systematically

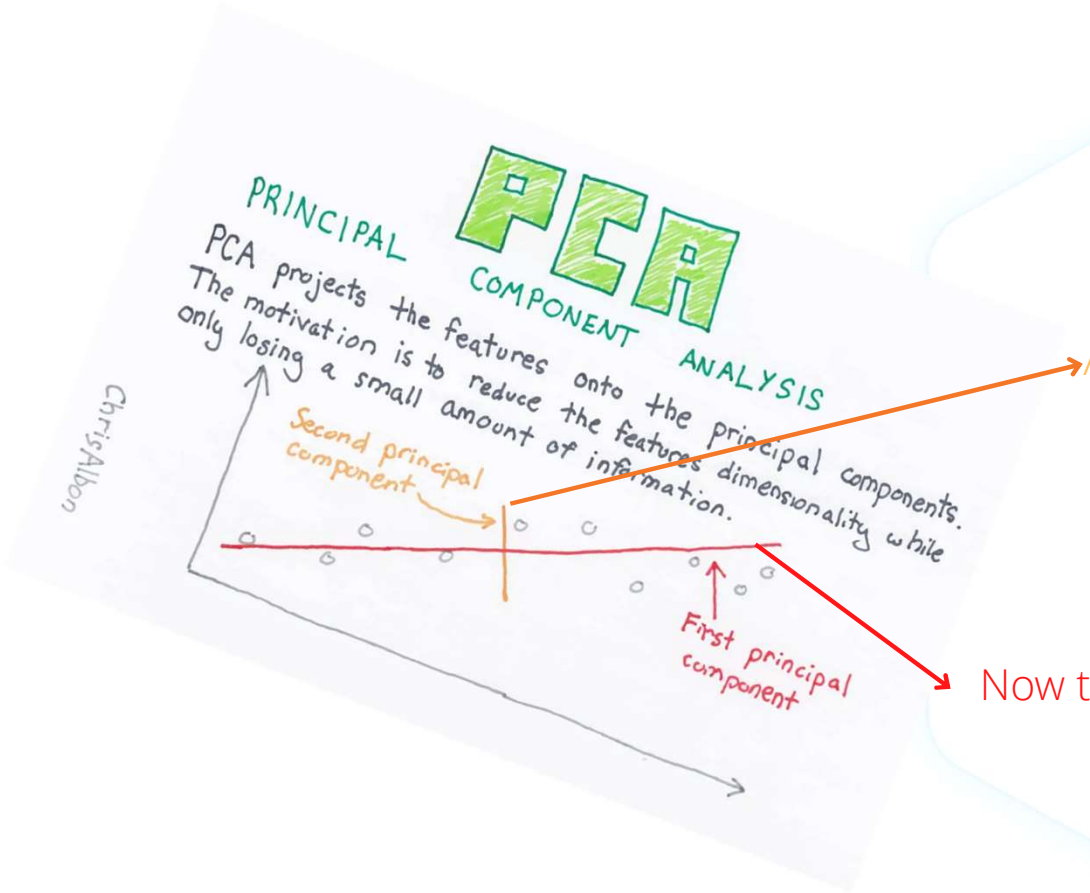
PRINCIPAL COMPONENT ANALYSIS

The PCA finds and combines the features which carry the most information into new "features"

The objective is to make the new features statistically uncorrelated so that each carries as much information as possible



PRINCIPAL COMPONENT ANALYSIS



And this is feature 2

Now this is feature1

ANY QUESTIONS ?