k-Means Clustering

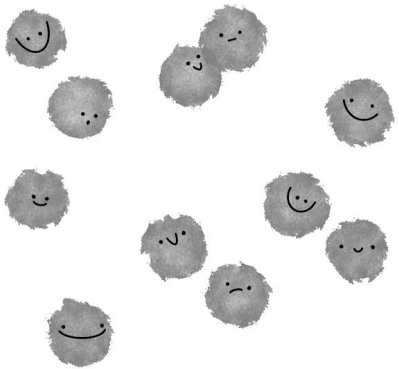
Prof. Dr. Jan Kirenz
HdM Stuttgart

k-Means algorithm example

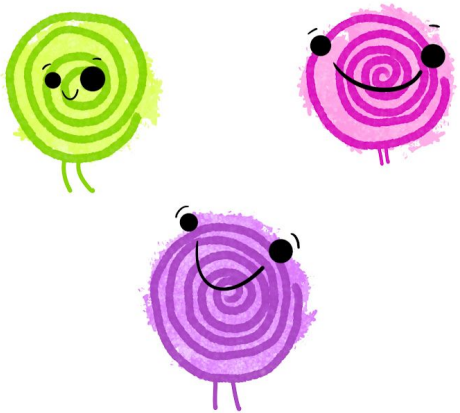
k-means clustering

: assign each observation to one of k clusters based on the nearest cluster centroid.

OBSERVATIONS



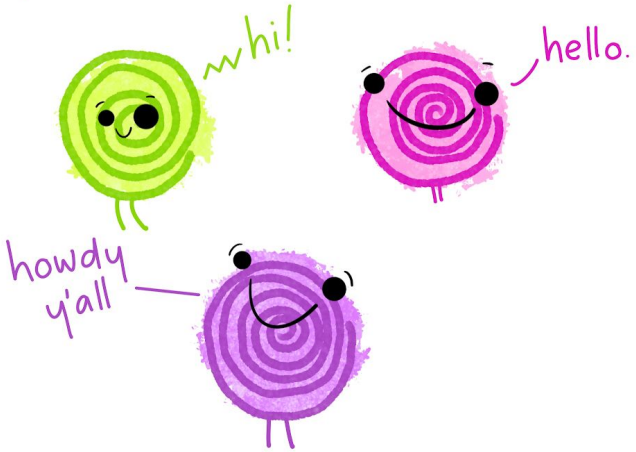
cluster CENTROIDS



Source: Allison Horst https://github.com/allisonhorst/stats-illustrations/tree/master/other-stats-artwork

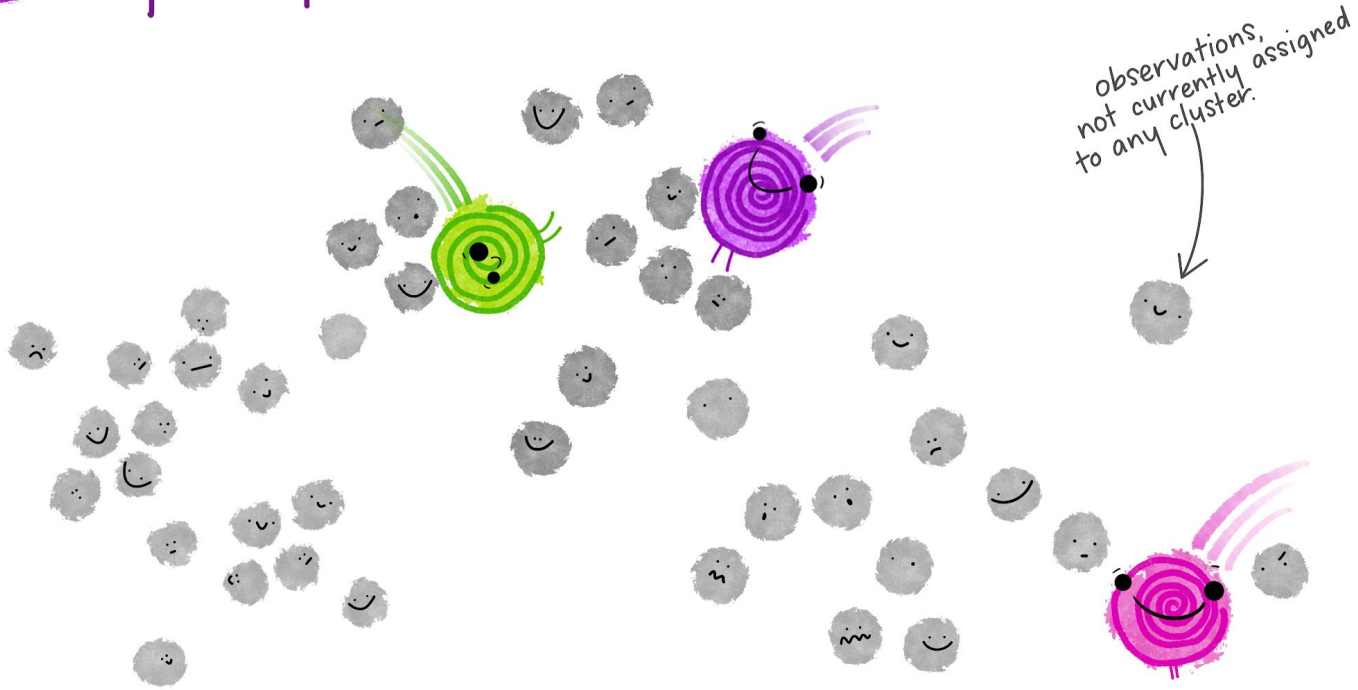
1 Specify the number of clusters (in this example, k = 3).

Then imagine k cluster centroids are created.



Source: Allison Horst https://github.com/allisonhorst/stats-illustrations/tree/master/other-stats-artwork

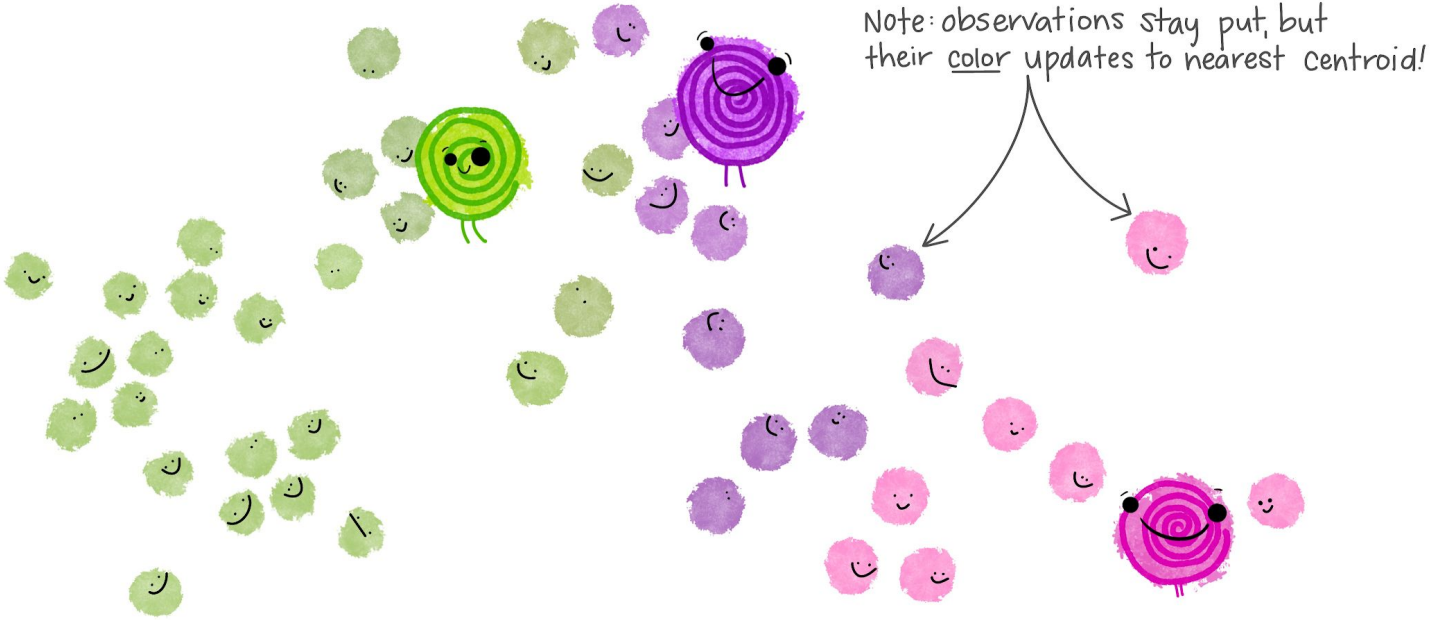
2 Those k centroids get randomly placed in your space.



Source: Allison Horst https://github.com/allisonhorst/stats-illustrations/tree/master/other-stats-artwork

Each observation gets temporarily "assigned" to its closest centroid.

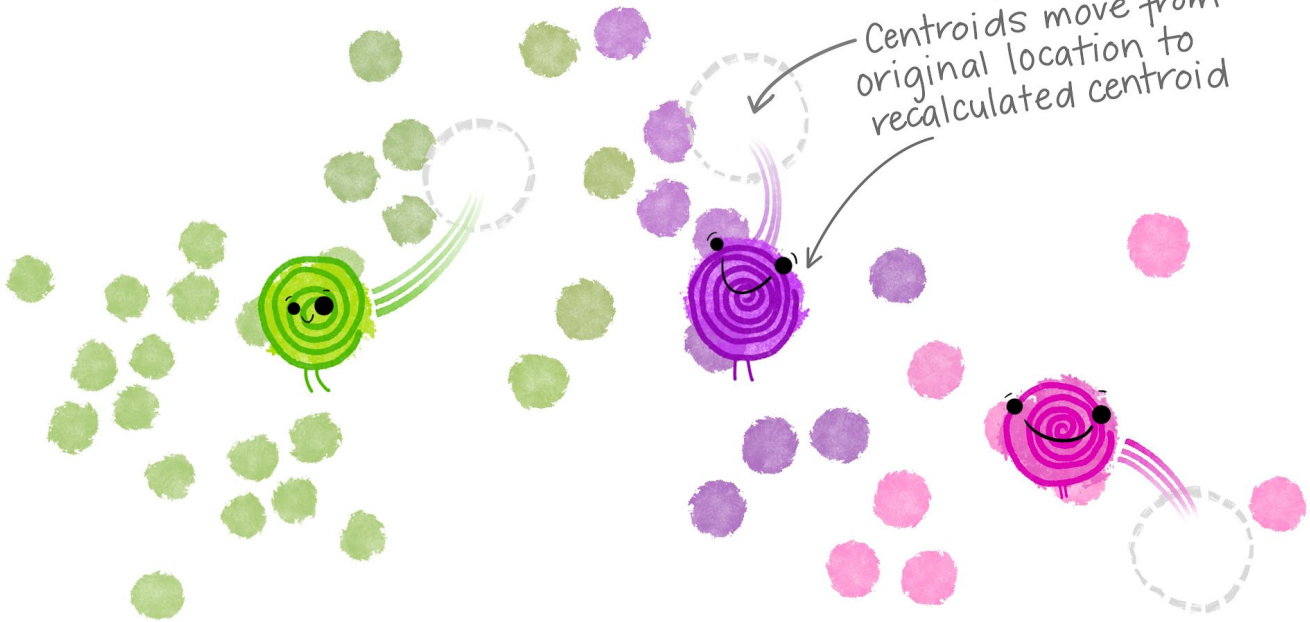
(e.g. by Euclidean distance)



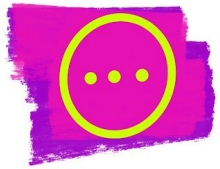
Source: Allison Horst https://github.com/allisonhorst/stats-illustrations/tree/master/other-stats-artwork

@allison_horst

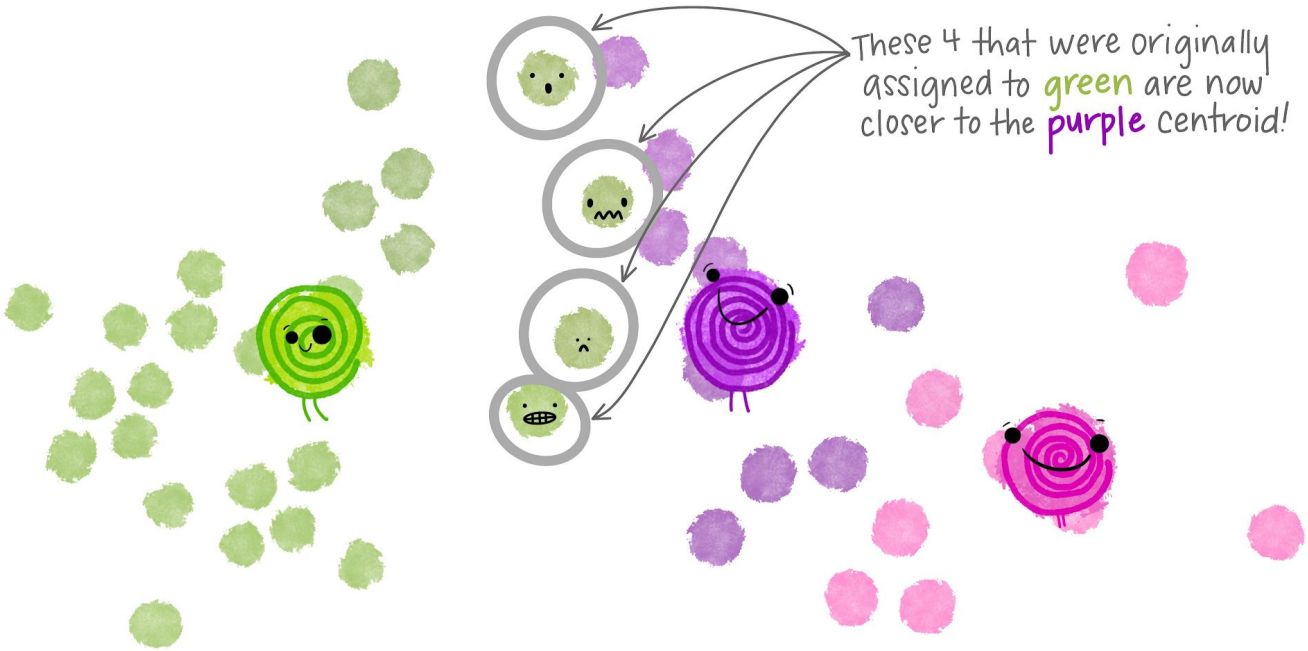
Then the centroid of each cluster is calculated based on all observations assigned to that cluster...



Source: Allison Horst https://github.com/allisonhorst/stats-illustrations/tree/master/other-stats-artwork



UH OH. Now that the cluster centroids have moved, some of the observations are now closer to a different centroid!



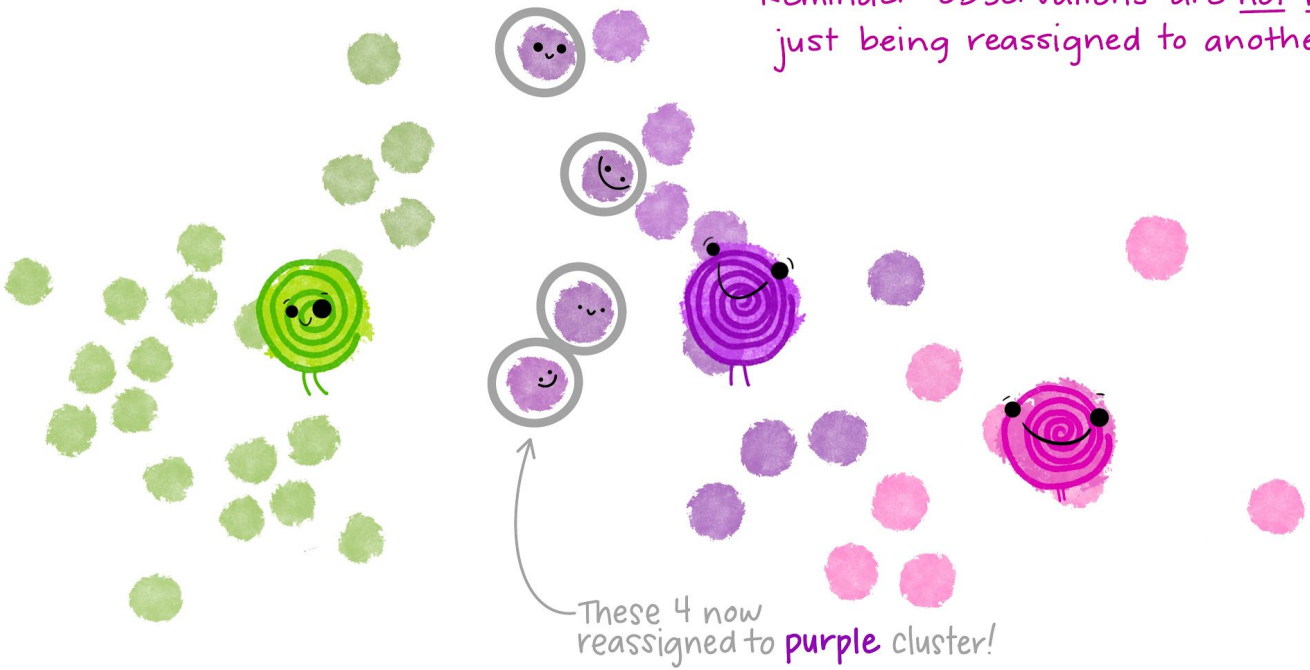
Source: Allison Horst https://github.com/allisonhorst/stats-illustrations/tree/master/other-stats-artwork

5

NO PROBLEM!

Observations get reassigned* to a different cluster based on the recalculated centroid.

*Reminder: observations are not moving, just being reassigned to another cluster.



@allison_horst

Source: Allison Horst https://github.com/allisonhorst/stats-illustrations/tree/master/other-stats-artwork

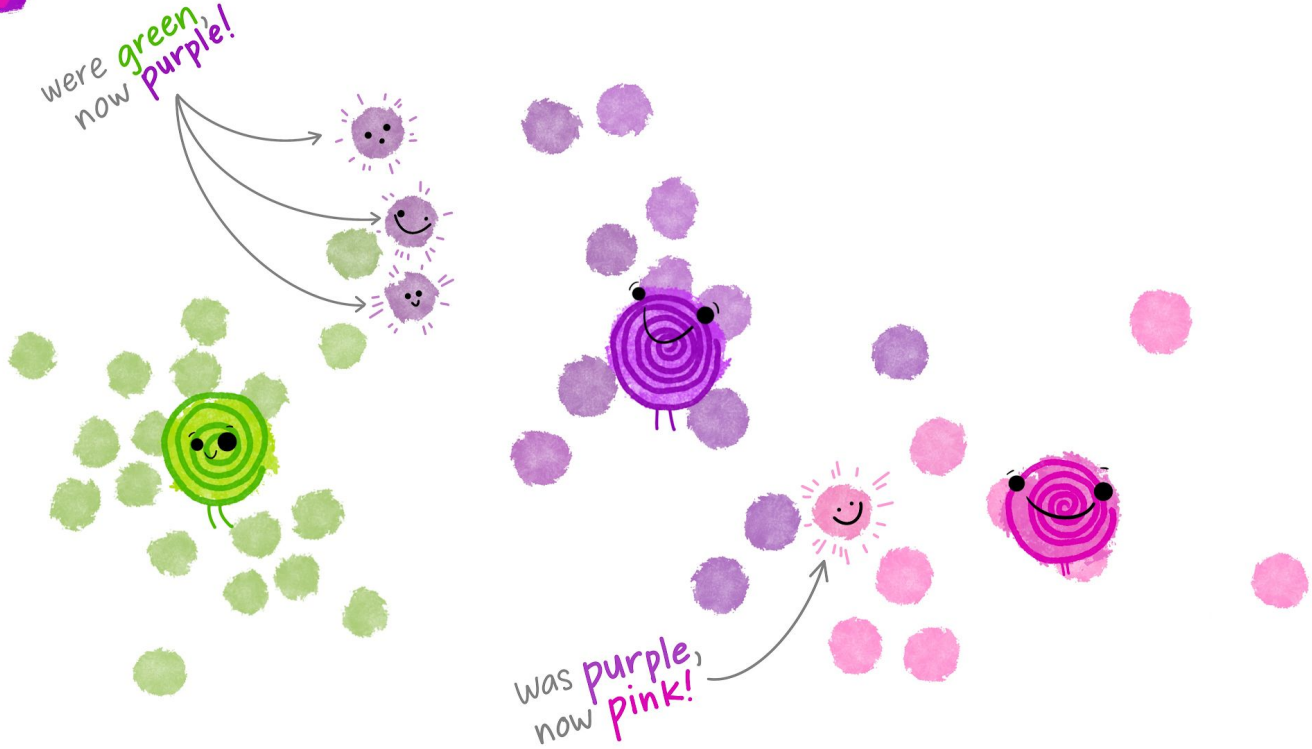
But now that observations have been reassigned, the centroids need to move again [recalculate centroids from updated clusters]

Source: Allison Horst https://github.com/allisonhorst/stats-illustrations/tree/master/other-stats-artwork

@allison_horst

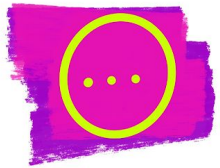
7

Again, now observations are reassigned as needed to the closest centroid.



Source: Allison Horst https://github.com/allisonhorst/stats-illustrations/tree/master/other-stats-artwork

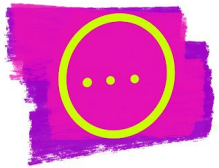
@allison_horst



Then the centroid for each cluster is recalculated...

...which means observations will be reassigned...

Source: Allison Horst https://github.com/allisonhorst/stats-illustrations/tree/master/other-stats-artwork



That iterative process of

Recalculate cluster centroids
↳ Reassign observations to nearest centroid
↳ Recalculate cluster centroids
↳ Reassign observations to nearest centroid
↳ Recalculate cluster centroids
↳ Reassign observations to nearest centroid

↓

Continues until nothing is moving or being reassigned anymore!

Source: Allison Horst https://github.com/allisonhorst/stats-illustrations/tree/master/other-stats-artwork

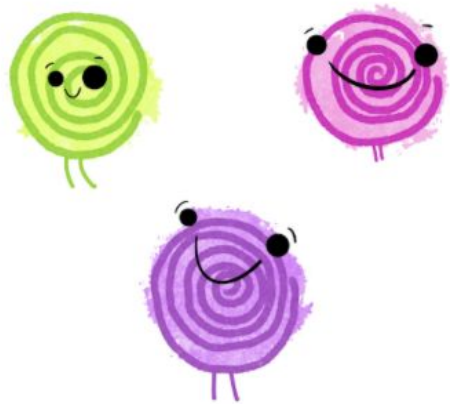
Which means the iteration is done and each observation is assigned to its final cluster.

Source: Allison Horst https://github.com/allisonhorst/stats-illustrations/tree/master/other-stats-artwork

@allison_horst

Advantages of k-means

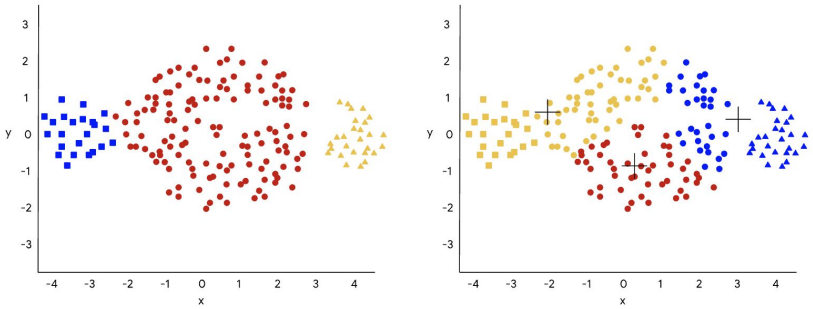
- Relatively **simple** to implement.
- Easily **adapts** to new examples.
- **Generalizes** to clusters of different shapes and sizes, such as elliptical clusters.



Source: Google Developers https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages

Disadvantages of k-means

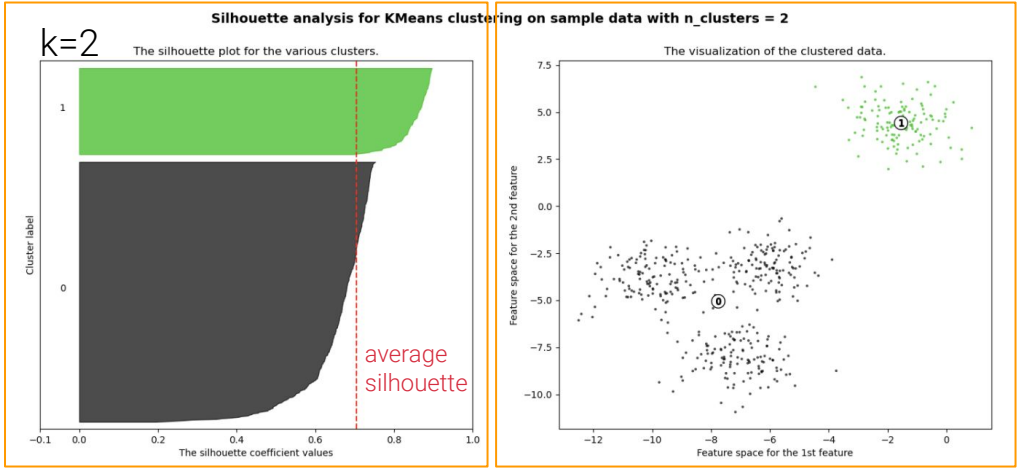
- **Choosing k** manually.
- Being dependent on **initial values**
- Clustering data of **varying sizes** and **density**.
- Clustering **outliers**.
- Scaling with number of **dimensions**.



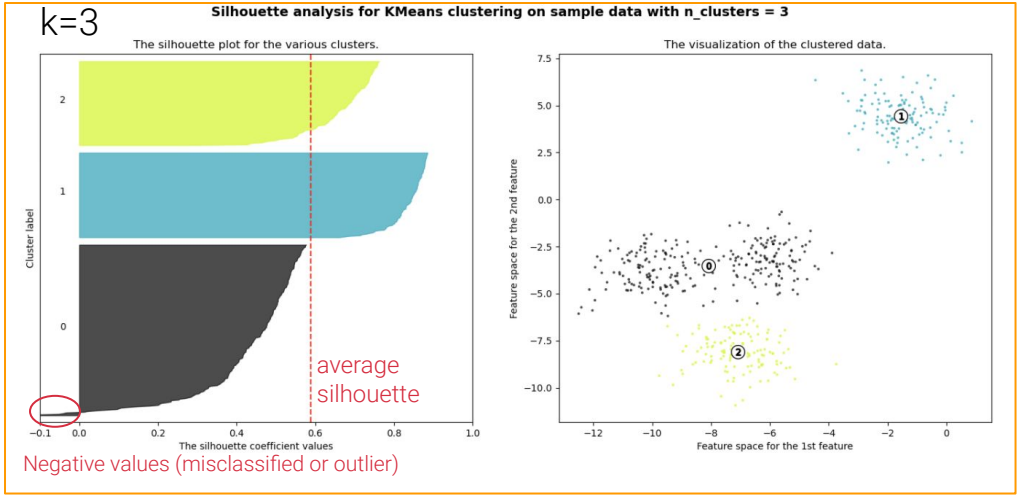
Source: Google Developers https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages

Silhouette analysis

k=2



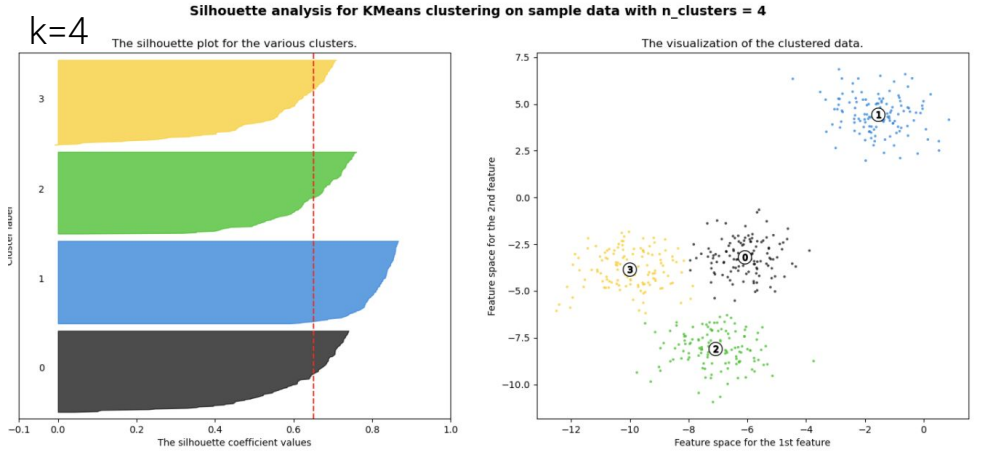
k=3



- Silhouette coefficient is between -1 and +1
- Higher is better
- Pick **k** where average silhouette is closest to 1

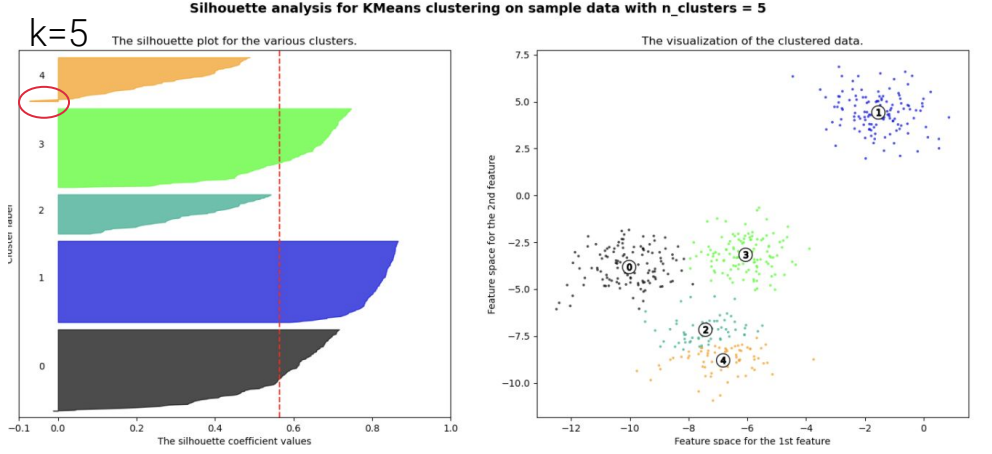
https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

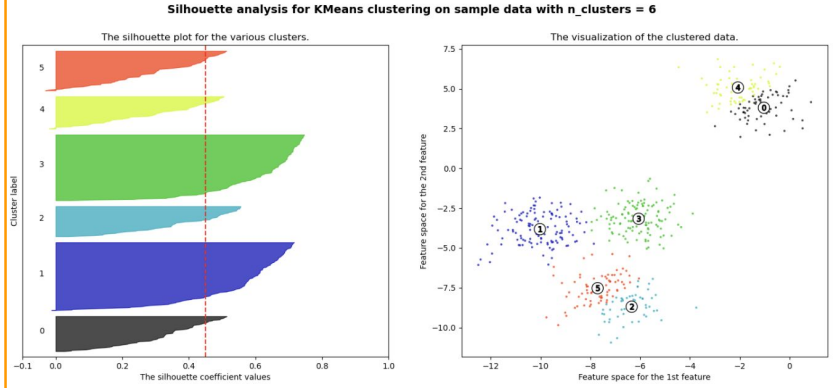
k=4




```
For n_clusters = 2 The average silhouette_score is : 0.7049787496083262
For n_clusters = 3 The average silhouette_score is : 0.5882004012129721
For n_clusters = 4 The average silhouette_score is : 0.6505186632729437
For n_clusters = 5 The average silhouette_score is : 0.56376469026194
For n_clusters = 6 The average silhouette_score is : 0.4504666294372765
```

k=5





https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

Selecting the number of clusters with **silhouette analysis** on K-Means clustering

- Can be used to study the **separation distance** between the resulting clusters.
- Measure of how **close** each point in one cluster is to points in the neighboring clusters
- Assess parameters like number of clusters visually.
- Measure has a range of [**-1, 1**].

Silhouette coefficients

- Near **+1** indicate that the sample is far away from the neighboring clusters.
- **0** indicates that the sample is on or very close to the decision boundary between two neighboring clusters
- **Negative values** indicate that those samples might have been assigned to the wrong cluster or are outliers

Source: Scikit-learn documentation https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html