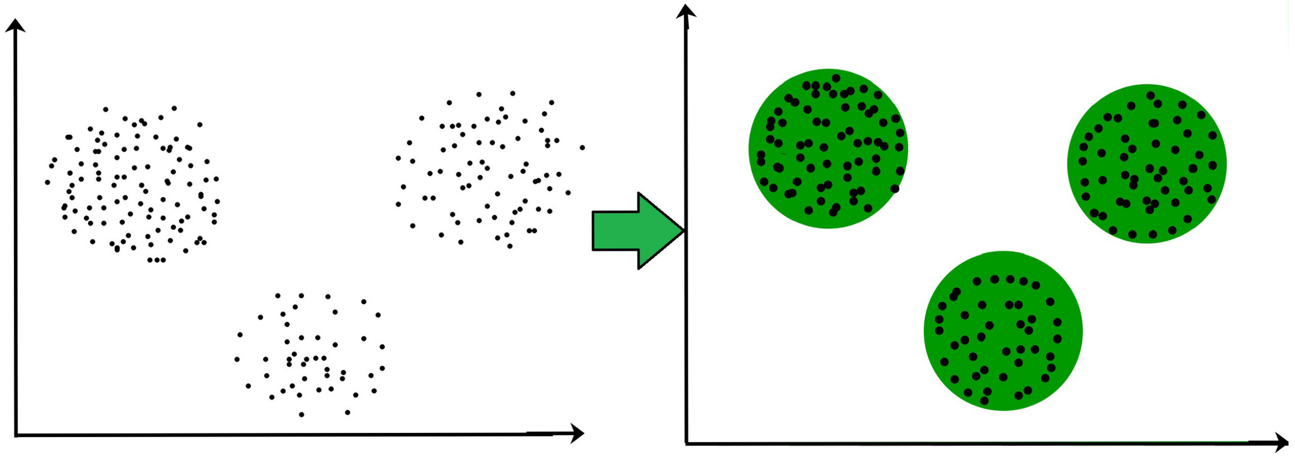
Features
Clustering

PLAKSHA UNIVERSITY

Clustering

Clustering is used to automatically discover “natural” groupings in data.

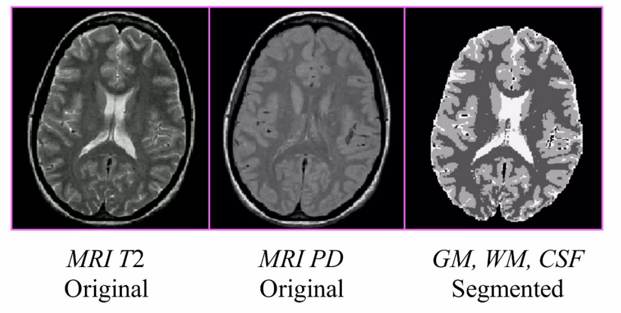
It is used when there are no classes to be predicted but rather the features have to be divided into groups.



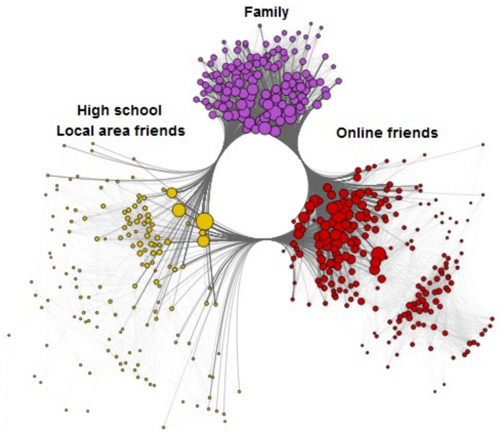
Clustering is an unsupervised method.

When is Clustering helpful?

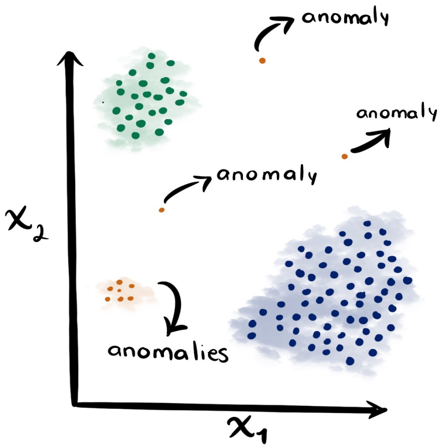
Clustering is quite helpful in problems where patterns need to be discovered. For example:



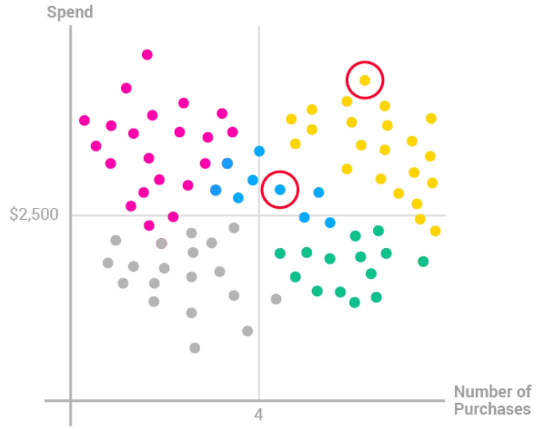
Medical Imaging



Social Media Connections



Anomaly Detection

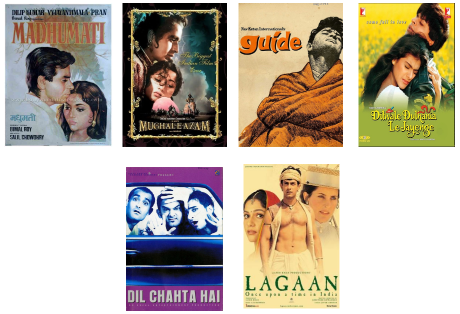


Recommendation Systems

Intro to ML → Features Extraction → Features Clustering → Features Dim Reduction → Features Classification → ML Systems Challenges → ML Real-world Applications → Intro to NNs → Multi-layer NNs → "Deep" NNs → Other ML Methods

Is there something like a natural cluster?

Won 0-6 Filmfares



Won >6 Filmfares

A main lead dies

No main lead dies

No, clustering is quite subjective, and is based on a similarity metric.

How does it work?

Clustering works by organizing data into classes such that:

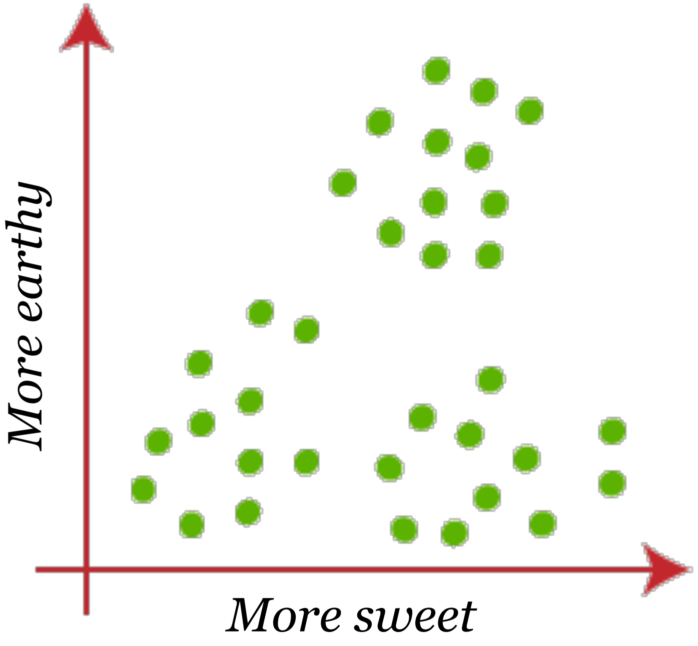
- There is high intra-class similarity.
- There is low inter-class similarity.

For clustering to work, we need:

- A similarity metric (i.e., the features being used to cluster)
- A distance metric (such as Euclidean distance)

Problem Statement

Let's say that the hostel mess at Plaksha University wants to introduce new flavors of seasoning. It has tasked you to collect data from students about how the new flavors should taste like.

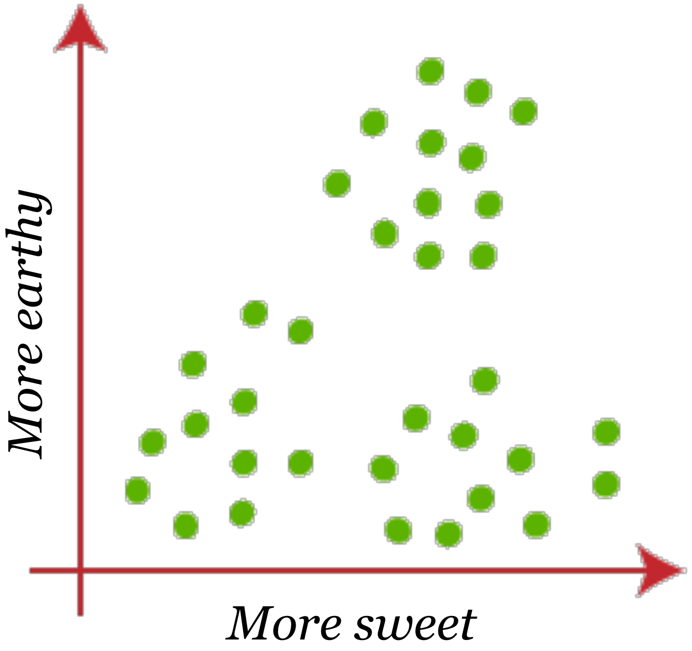


Based on the data you collected from students, "optimally" what kinds of new seasoning should the hostel mess introduce?

K-means Clustering Method

K-means is the easiest method to cluster data and is yet quite efficient. It works in four steps:

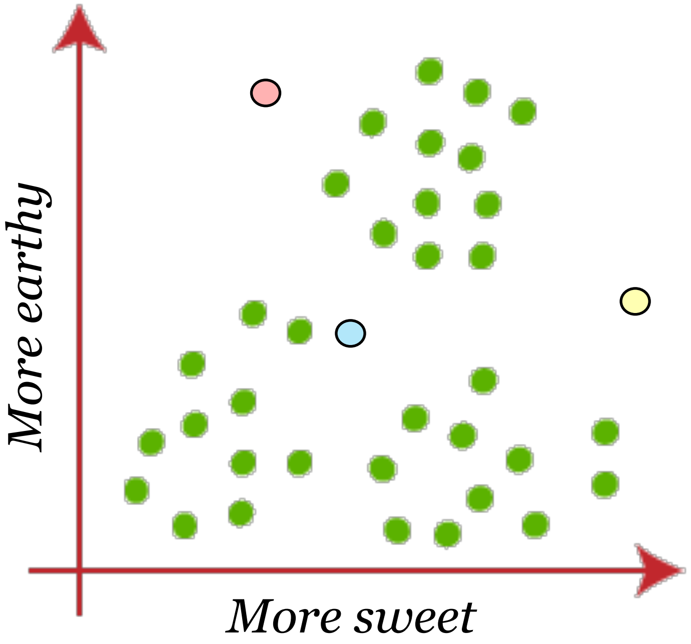
1. Assign K (user-input) initial means randomly in the data. Let's say K = 3.



K-means Clustering Method

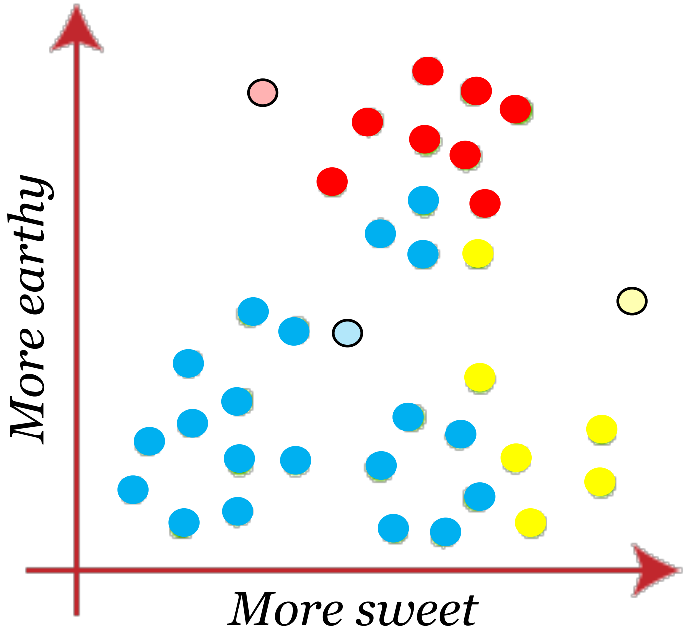
K-means is the easiest method to cluster data and is yet quite efficient. It works in four steps:

1. Assign K (user-input) initial means randomly in the data. Let's say K = 3.



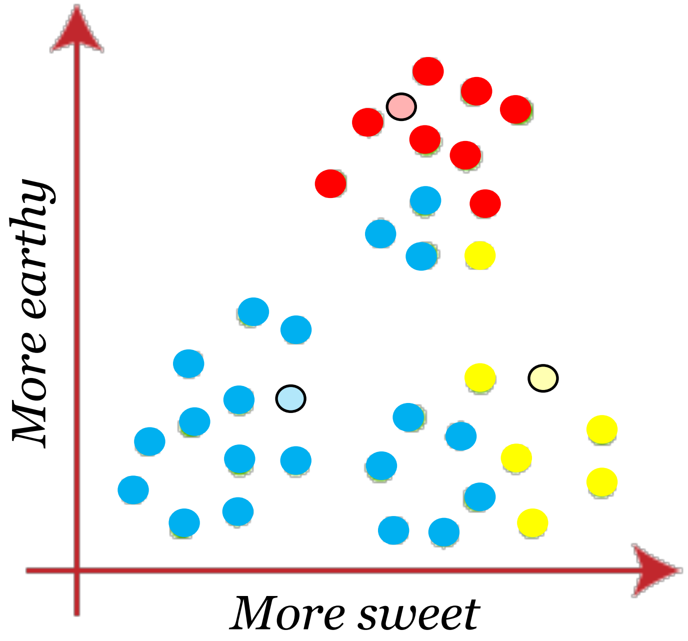
K-means Clustering Method

2. Form K clusters by associating every observation with the nearest mean.



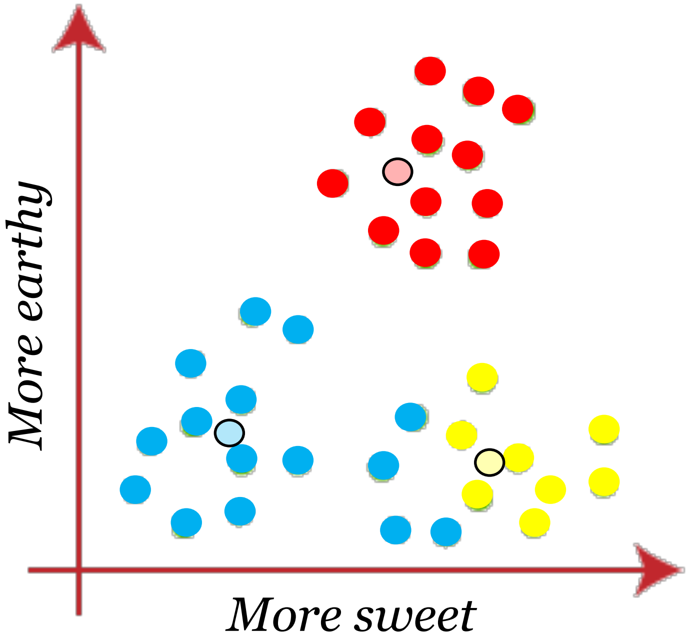
K-means Clustering Method

3. Find the centroids of the new clusters. They become the new means.



K-means Clustering Method

4. Repeat Steps 2 & 3 until convergence has been reached.



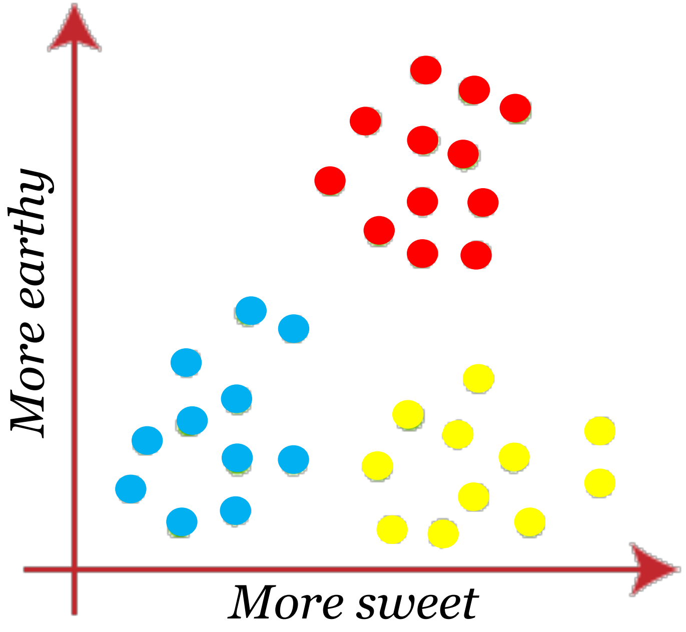
K-means Clustering Method

4. Repeat Steps 2 & 3 until convergence has been reached.

The convergence criterion could be either

a) There is no change in cluster ID of any data point.

b) A pre-specified total number of iterations has been reached.



K-means Advantages

- Easy to implement.
- Scales well to large data sets.
- Guarantees convergence.
- It forms tighter clusters since it is based only on a distance metric.

K-means Questions

- What does K-means try to optimize?
- How can we automatically choose K i.e., the number of clusters?
- Are we sure it will terminate/converge?
- Is K-means “optimal”?

K-means Optimization Criterion

In mathematical terms, K-means clustering works on a set of observations (x_1, x_2, x_3, …, x_n) where each observation is a *d*-dimensional vector, by aiming to partition the *n* observations into *k* sets ($k <= n$) S = {S_1, S_2, S_3, …, S_k) so as to minimize the within-cluster sum of squares (WCSS):

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x_j \in S_i} \left\| x_j - u_i \right\|^2$$

where, u_i is the mean of points in S_i.

$$\boldsymbol{\mu_i} = \frac{1}{|S_i|} \sum_{\mathbf{x} \in S_i} \mathbf{x},$$

Finding the Optimal K

Two methods are predominantly used to find the optimal K

a) Elbow Method

Gives an idea of what K should be based on within cluster Sum of Squares (WCSS) between all data points and their clusters' centroids.

$$\text{WCSS} = \sum_{\text{Pi in Cluster1}} \text{distance}(P_i\ C_1)^2 + \sum_{\text{Pi in Cluster2}} \text{distance}(P_i\ C_2)^2 + \sum_{\text{Pi in CLuster3}} \text{distance}(P_i\ C_3)^2$$

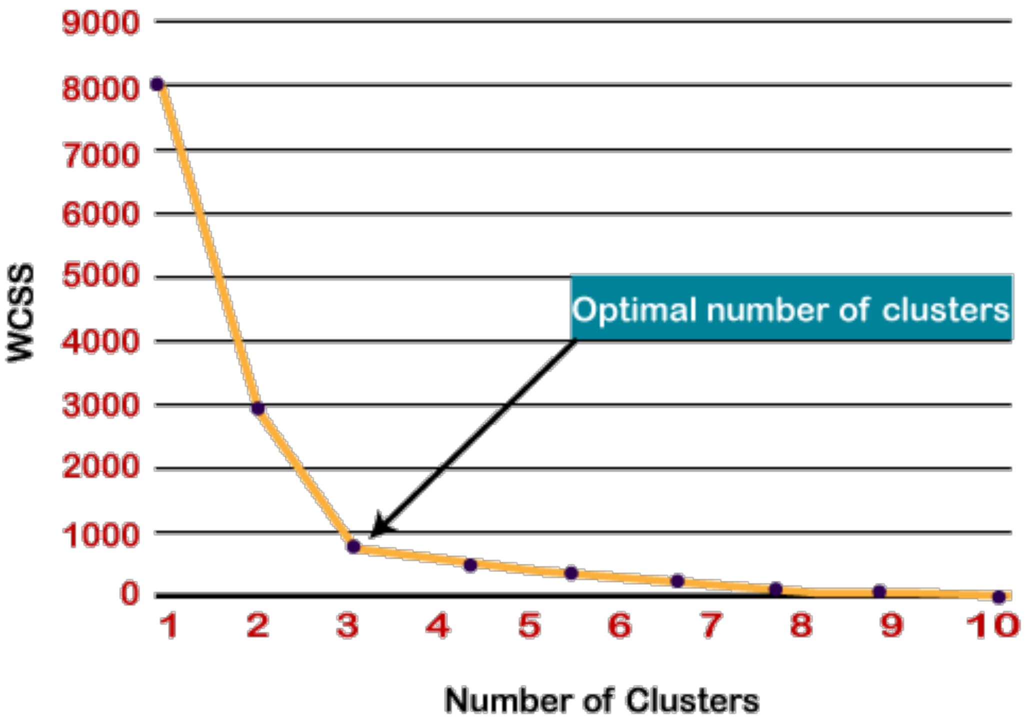
For example, for K = 3 i.e., for three clusters, find the distance of each datapoint in that cluster from the centroid of that cluster.

Finding the Optimal K

Two methods are predominantly used to find the optimal K

a) Elbow Method

Find WCSS for a range of K



K is taken to be optimal where the graph tends to flatten out and form an elbow.

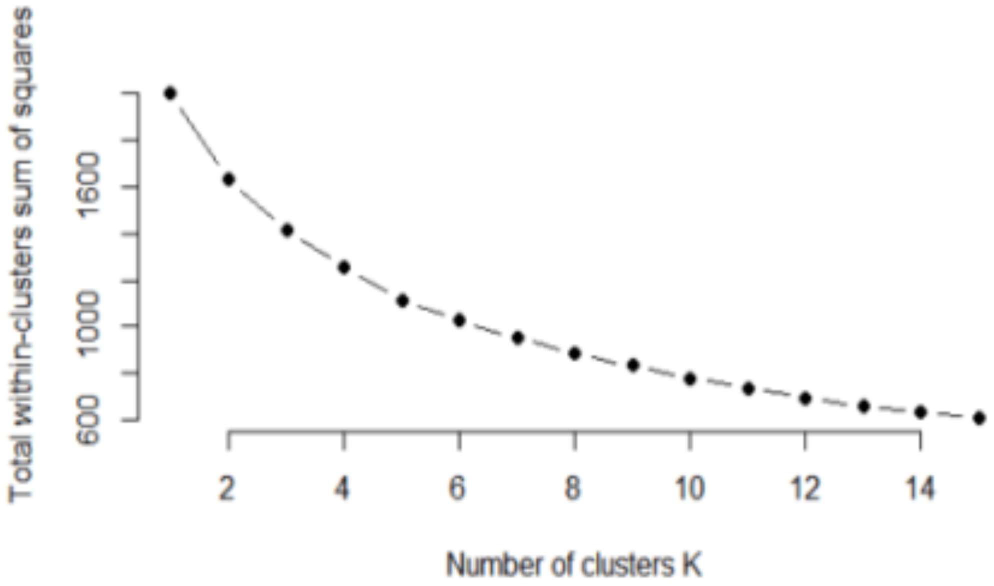
This is because the reduction in WCSS beyond the elbow point is only marginal as we keep increasing K.

Finding the Optimal K

Two methods are predominantly used to find the optimal K

a) Elbow Method

Find WCSS for a range of K



However, sometimes the Elbow point may not be apparent.

In any case, Elbow method only works for K-means since it evaluates the WCSS distance.

What about clustering in general? How do we evaluate K for clustering in general?

Finding the Optimal K

b) Silhouette Method

Silhouette Method works by determining the degree of separation between clusters.

For all data points, in each cluster:

Step 1: Compute the average distance from all data points in the same cluster a^i

Step 2: Compute the average distance from all data points in the closest cluster b^i

Step 3: Compute the coefficient

$$\frac{b^i - a^i}{max(a^i, b^i)}$$

Finding the Optimal K

b) Silhouette Method

$$\frac{b^i - a^i}{max(a^i, b^i)}$$

The coefficient takes values between the interval [-1, 1].

If the value is 0 -> the sample is very close to neighboring clusters.

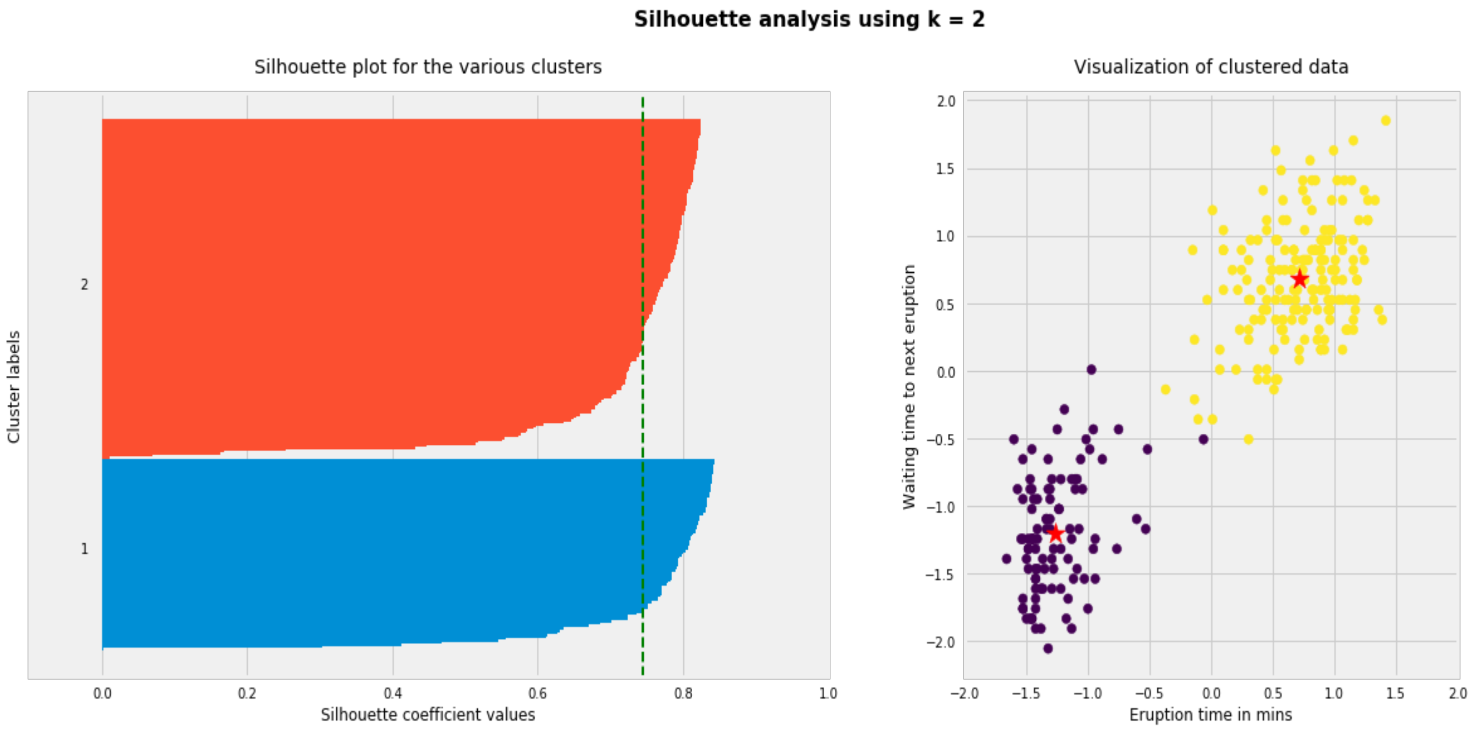
If the value is +1 -> the sample is far away from neighboring clusters (good!).

If the value is -1 -> the sample is assigned to the wrong cluster (bad!).

Thus, the coefficient scores for all data points should be as closer to 1 as possible.

Finding the Optimal K

b) Silhouette Method

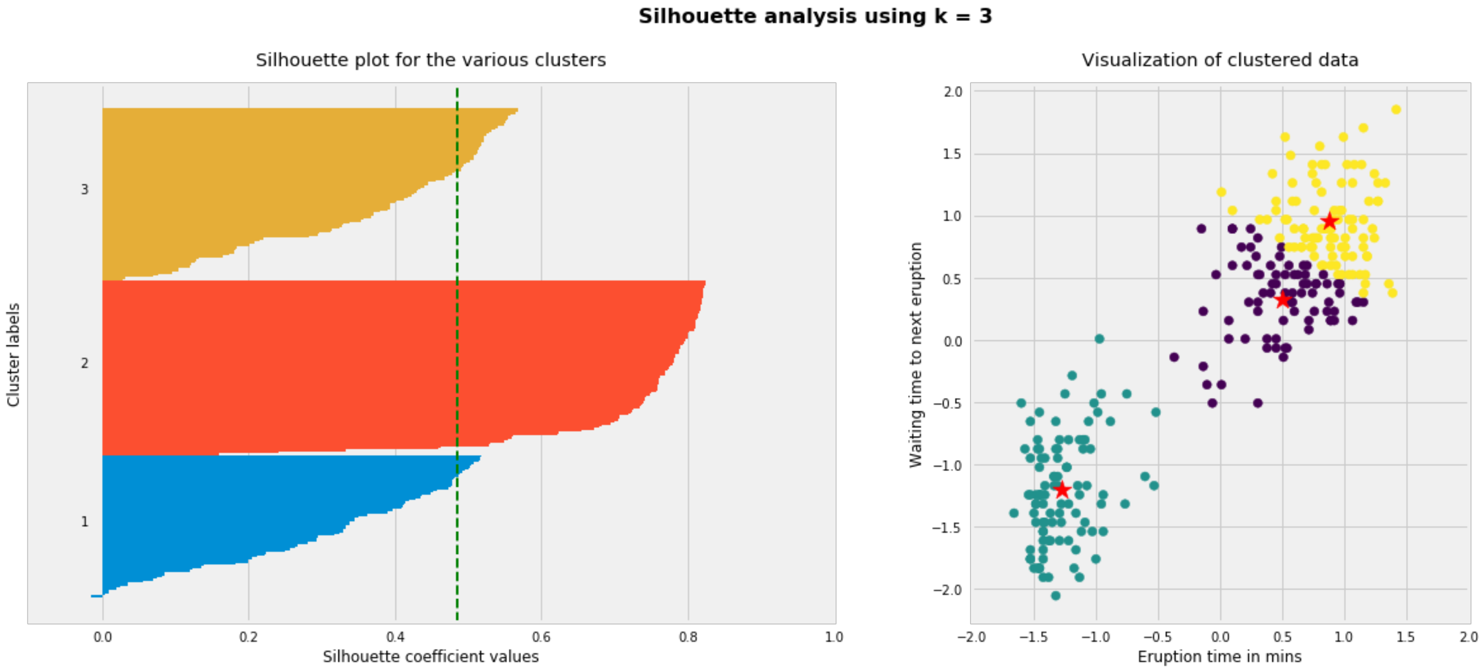


The width of a cluster label is proportional to the number of data points in that cluster.

Average Silhouette Coefficient score for two clusters: 0.75

Finding the Optimal K

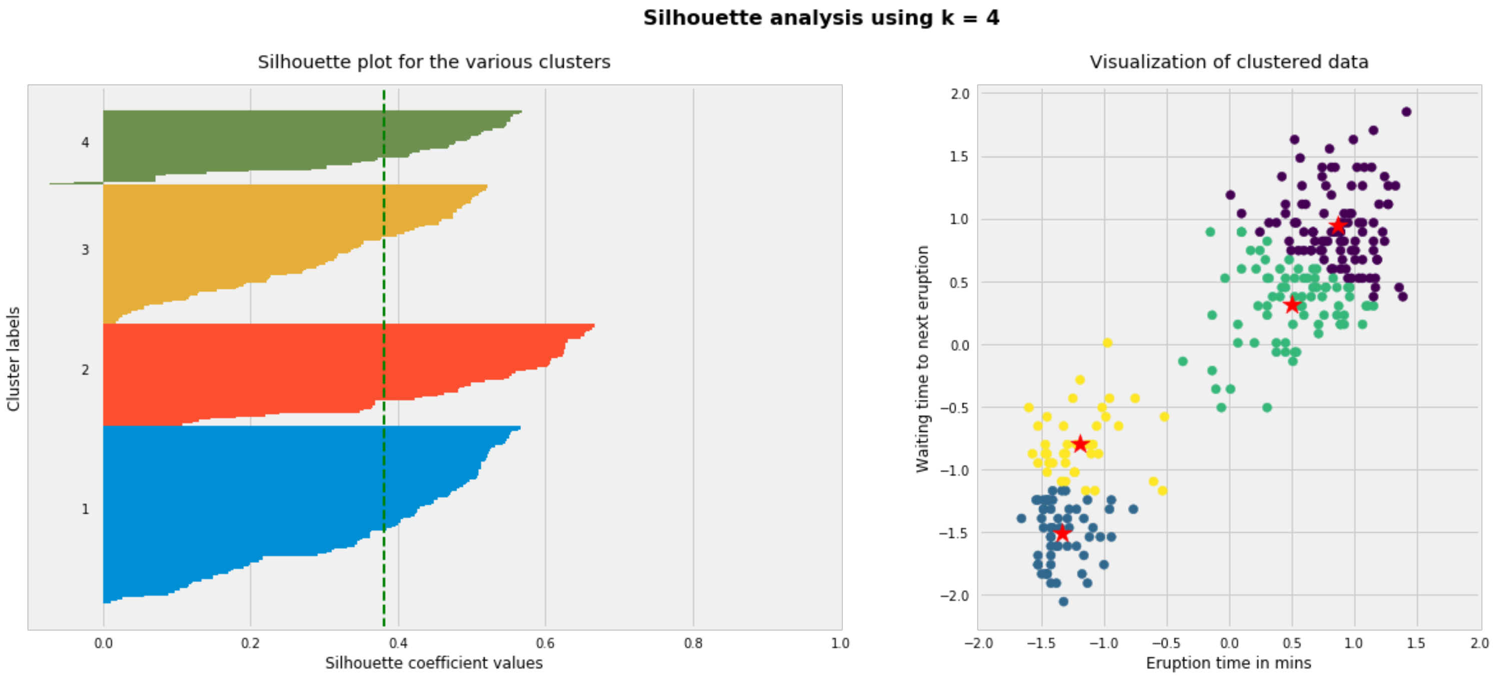
b) Silhouette Method



Average Silhouette Coefficient score for three clusters: 0.48

Finding the Optimal K

b) Silhouette Method



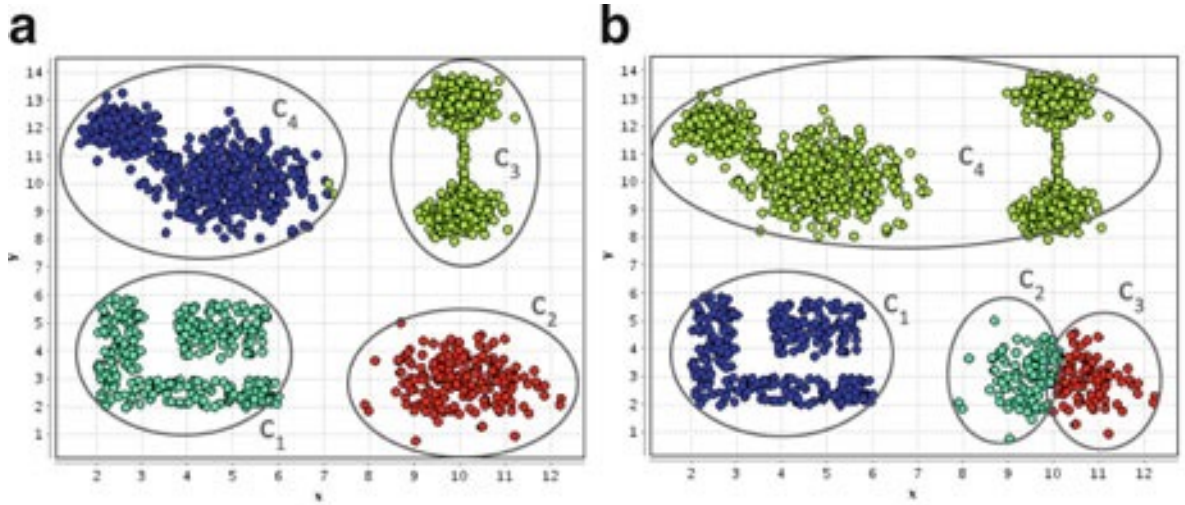
Average Silhouette Coefficient score for four clusters: 0.38

Thus, two clusters are optimal for this dataset since they have the highest Silhouette Coefficient score.

Does K-Means find “optimal” clusters?

- Not necessarily!

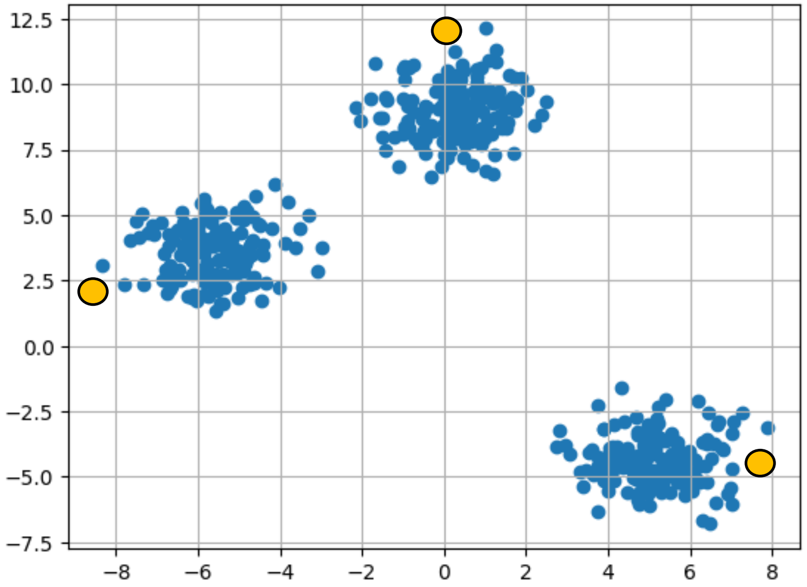
- K-means is impacted by the initial random locations of means. There could be configurations in which K-means has converged but is unable to find the “optimal” configuration.



k=4 k=4

How to find a “good” optima in K-means?

- Be careful about the starting points. If needed, give a pseudo-random start rather than a random start (also known as K-means++ algorithm).



One useful trick is to place first mean on top of a randomly chosen data point in a corner.

Put the second mean on a datapoint that is as far away as possible from the first mean.

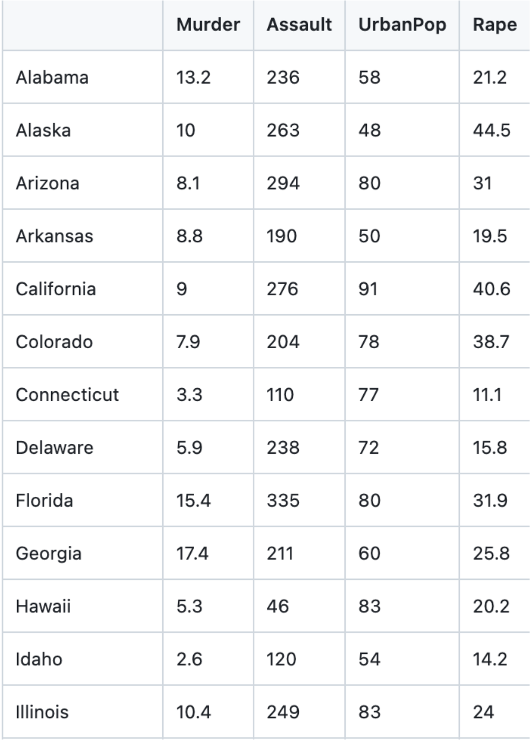
Put the third mean on a datapoint that is as far away as the first two means…and so on!

This helps in putting initial means around all the corners formed by the datapoints.

- Do many runs of K-means and validate how each configuration is evolving.

Let’s look at a real-life example

Let’s say you are a policy enthusiast who wants to recommend the government about policy-making against crime in different cities. You have demographic and crime data for 51 different cities.



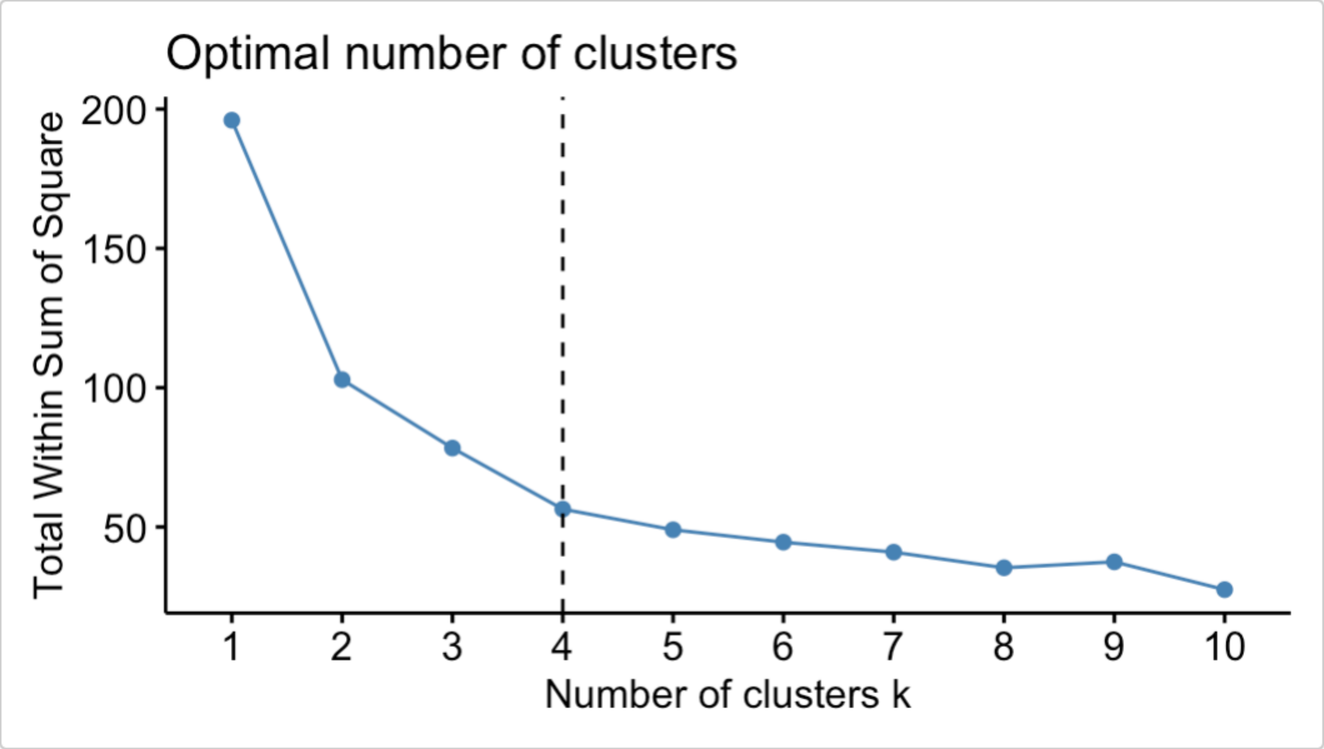
	Murder	Assault	UrbanPop	Rape
Alabama	13.2	236	58	21.2
Alaska	10	263	48	44.5
Arizona	8.1	294	80	31
Arkansas	8.8	190	50	19.5
California	9	276	91	40.6
Colorado	7.9	204	78	38.7
Connecticut	3.3	110	77	11.1
Delaware	5.9	238	72	15.8
Florida	15.4	335	80	31.9
Georgia	17.4	211	60	25.8
Hawaii	5.3	46	83	20.2
Idaho	2.6	120	54	14.2
Illinois	10.4	249	83	24

One wasteful way would be to generate 51 policy documents i.e. separate document for each city!

Or, we can find patterns in the data by clustering to see how many “optimal” clusters are formed and accordingly prepare documents for a group of cities that have similar population, crime rate, etc.

Let’s look at a real-life example

First, let’s generate an Elbow plot for different number of clusters.



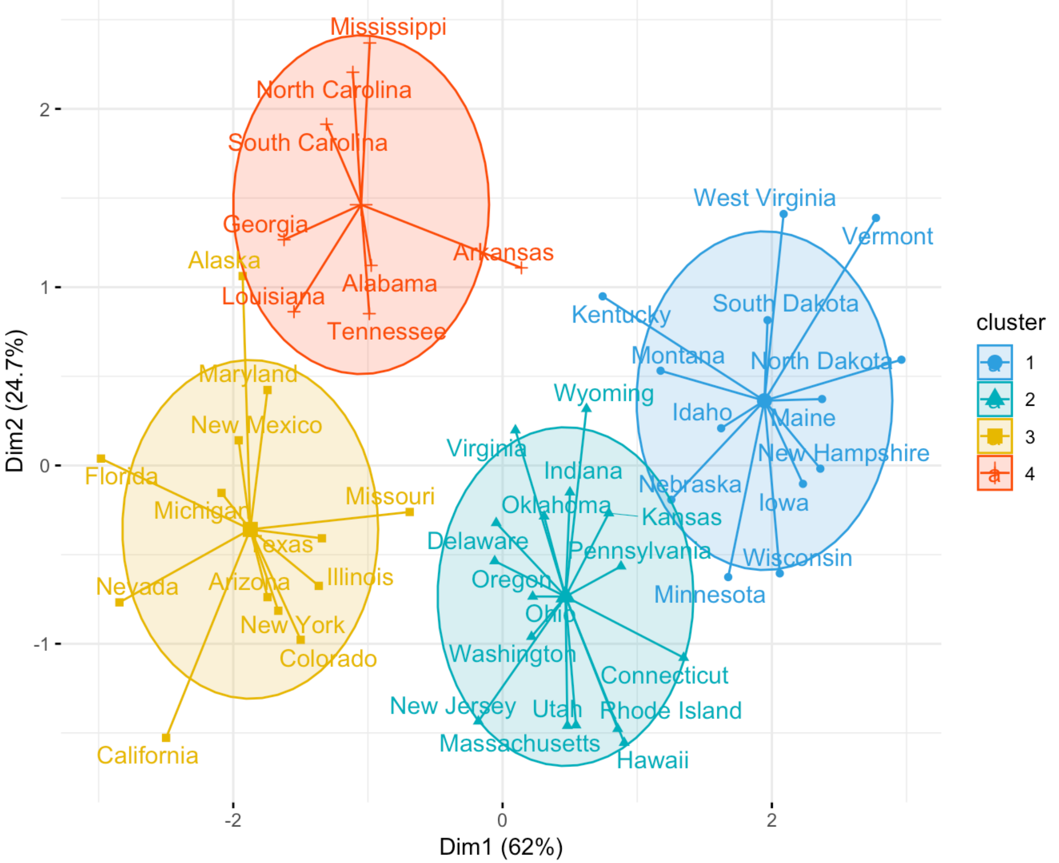
So, it seems like there are 4 clusters of cities.

We can generate 4 documents in total and they should be able to separately map each city’s requirements.

But, which cities are there in each cluster?

Let’s look at a real-life example

We use k-Means clustering to visualize which cities form a cluster.

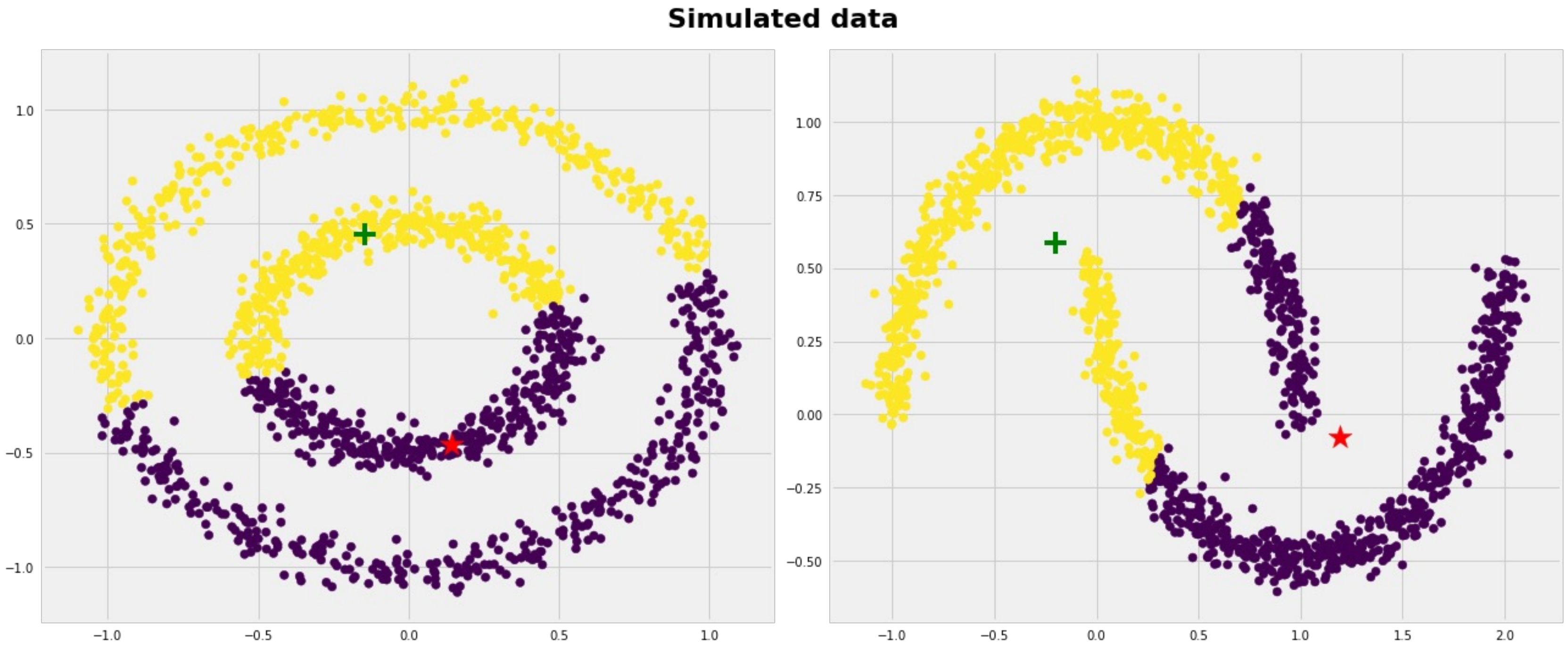


Based on this plot, we can also make additional recommendations by bucketing cities as (high-population, high-crime), (low-population, high-crime), etc.

Furthermore, we can track over the years if our recommendations are working or not. For example, if a city has moved from (high-population, high-crime) cluster to a (high-population, low-crime) cluster.

K-means Disadvantages

- It does not do a good job for complicated geometric shapes.

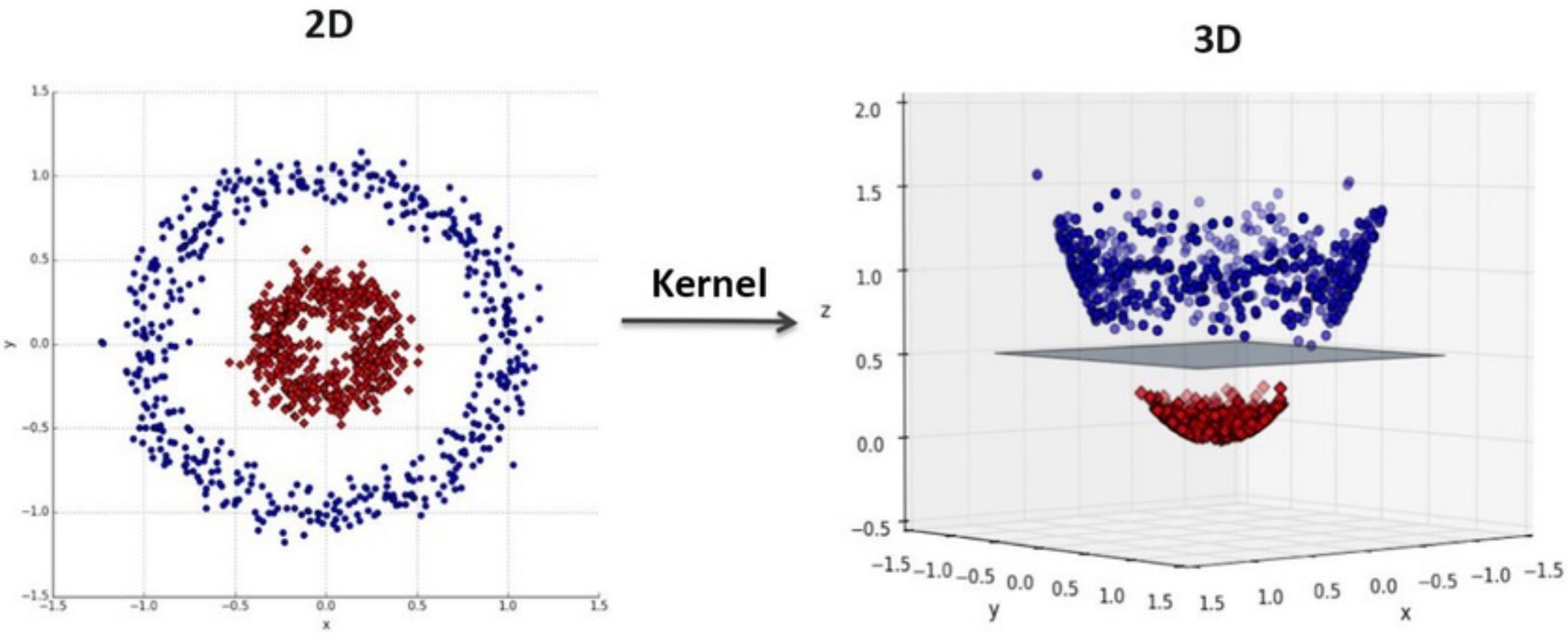


Simulated data

- The data needs to be standardized first (mean = 0, std = 1) before applying K-means.

K-means “Kernel Trick”

- Lower-dimensional data can be transformed to a higher dimensionality by using a kernel function. Once done, K-means may work well in the higher dimensional data. We will study the Kernel Trick in detail in SVM-based Feature Classification.



Summary

This lecture

- Clustering of data using k-Means, its advantages and disadvantages.
- How to find the “optimal” value of K in the K-means algorithm.

Next lecture

- GMM clustering technique that will require two parameters to be optimized and will be more robust.
- DBScan clustering technique that will not require the user to provide a value of K and will be able to detect noise/outliers in the data.

Questions?