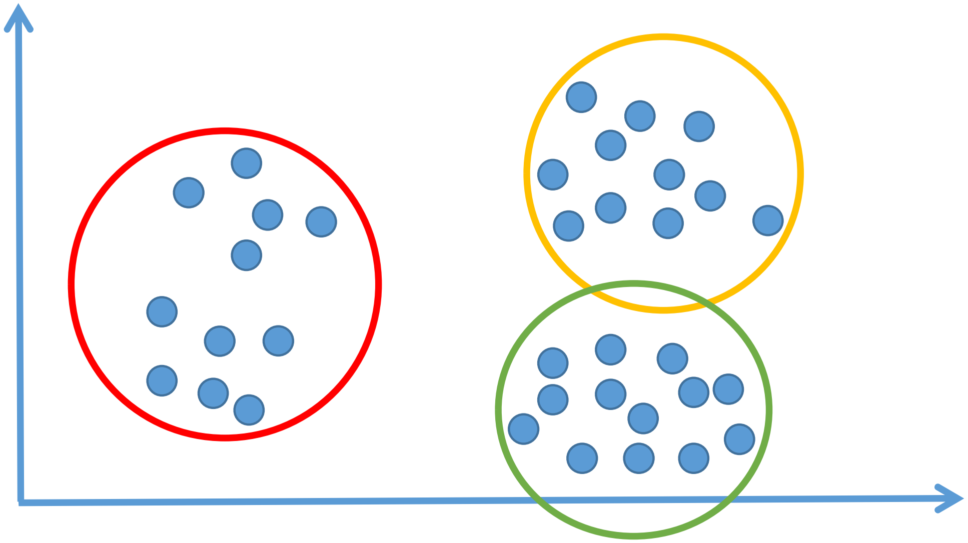
Artificial Intelligence

k - means Clustering

TS. Đào Duy Tuấn

k - Means Clustering

- ✓ Basic idea: group together similar instances
  - High intra-cluster similarity
  - Low inter-cluster similarity



k - Means Clustering

- ✓ Basic idea: group together similar instances
  - ❖ High intra-cluster similarity
  - ❖ Low inter-cluster similarity

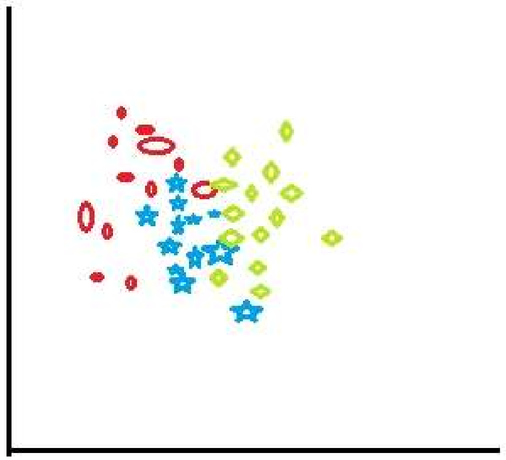


fig 1: before applying
k-means clustering

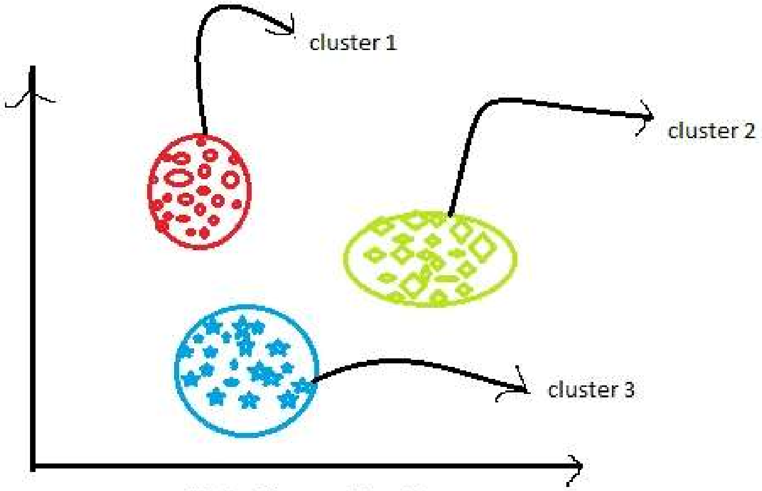


fig 2: After applying K-
means clustering

https://www.analyticsvidhya.com/

k - Means Clustering

- ✓ What is clustering?
  - **Clustering** is the classification of objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often according to some defined distance measure

k - Means Clustering

- Example:
  - Document clustering
    - Web search engine often return thousands of pages --> Difficult for user
    - Clustering can be used to group retrieved documents into categories
  - Customer segmentation
  - Recommendation engines
  - Image compression

k - Means Clustering

✓ Example:

Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs

Land use: Identification of areas of similar land use in an earth observation database

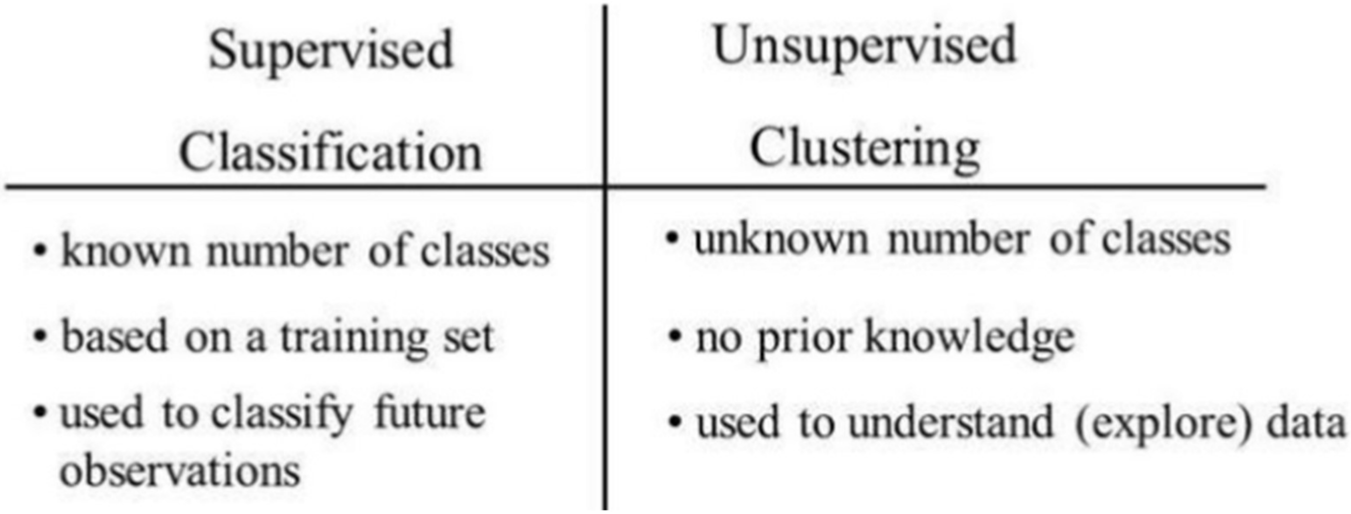
Insurance: Identifying groups of motor insurance policy holders with a high average claim cost

Urban planning: Identifying groups of houses according to their house type, value, and geographical location

Seismology: Observed earth quake epicenters should be clustered along continent faults

k - Means Clustering

- ✓ Supervised or unsupervised?

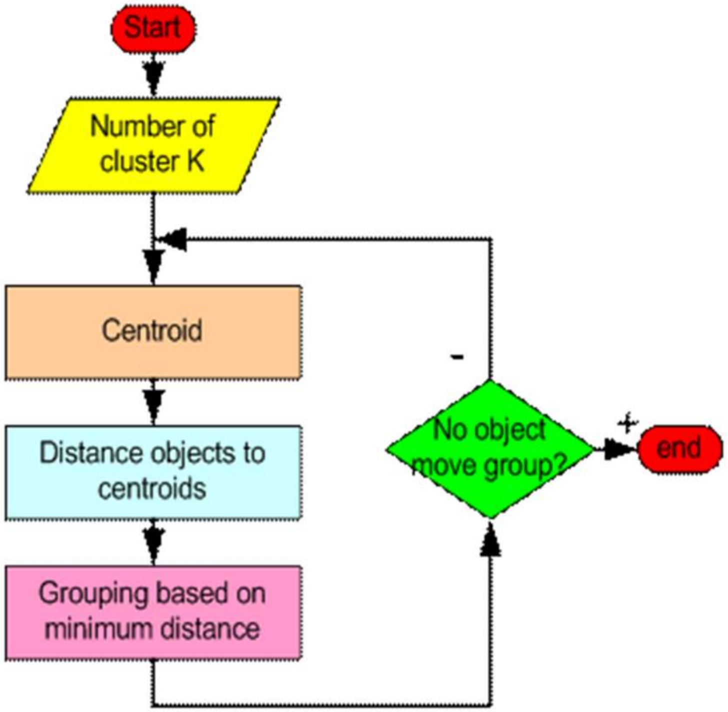


Supervised	Unsupervised
Classification	Clustering
• known number of classes	• unknown number of classes
• based on a training set	• no prior knowledge
• used to classify future observations	• used to understand (explore) data

- ✓ Requires data, but no labels
- ✓ Useful when don’t know what we’re looking for

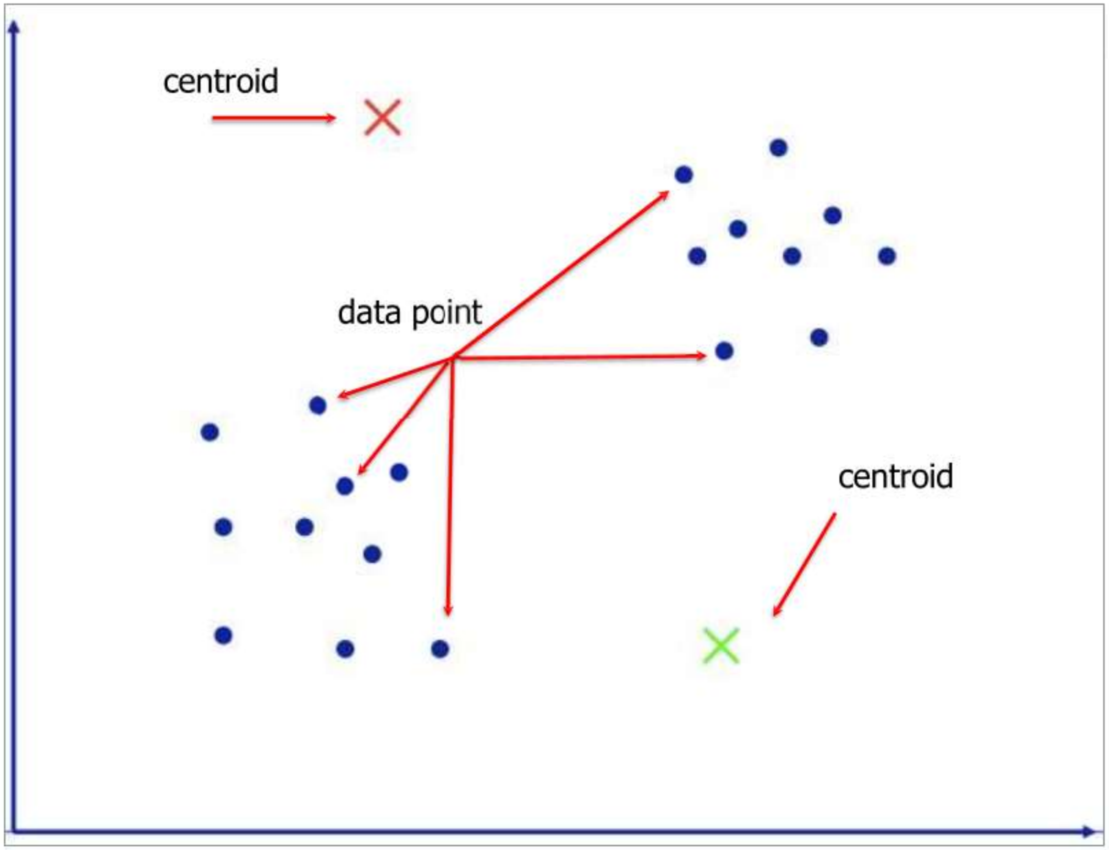
k - Means Clustering

✓ How the K-Mean Clustering algorithm works?


Start
Number of cluster K
Centroid
Distance objects to centroids
Grouping based on minimum distance
No object move group?
-
+
end

k - Means Clustering

- ✓ How the K-Mean Clustering algorithm works?



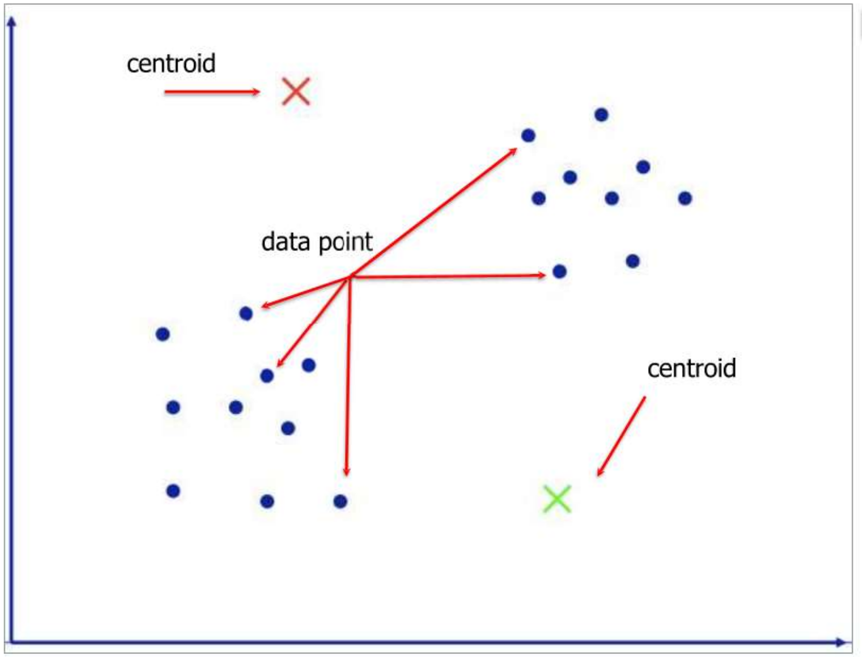
K-means Clustering

- Strengths
  - Simple iterative method
  - User provides “K”
- Weaknesses
  - Often too simple → bad results
  - Difficult to guess the correct “K”

K-means Clustering

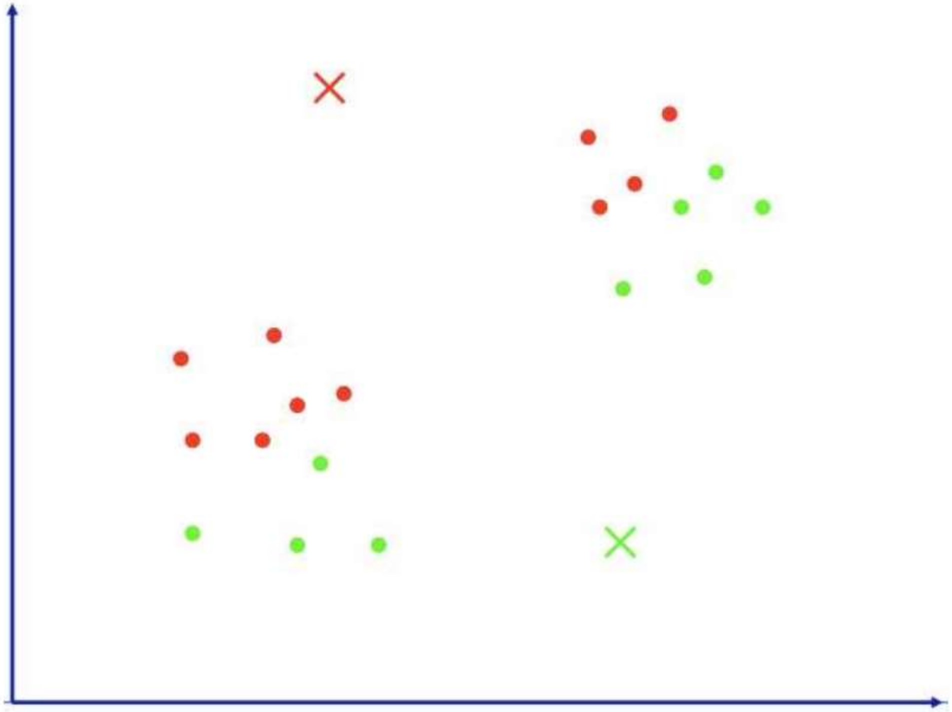
Basic Algorithm:

- Step 0: select K
- Step 1: randomly select initial cluster seeds



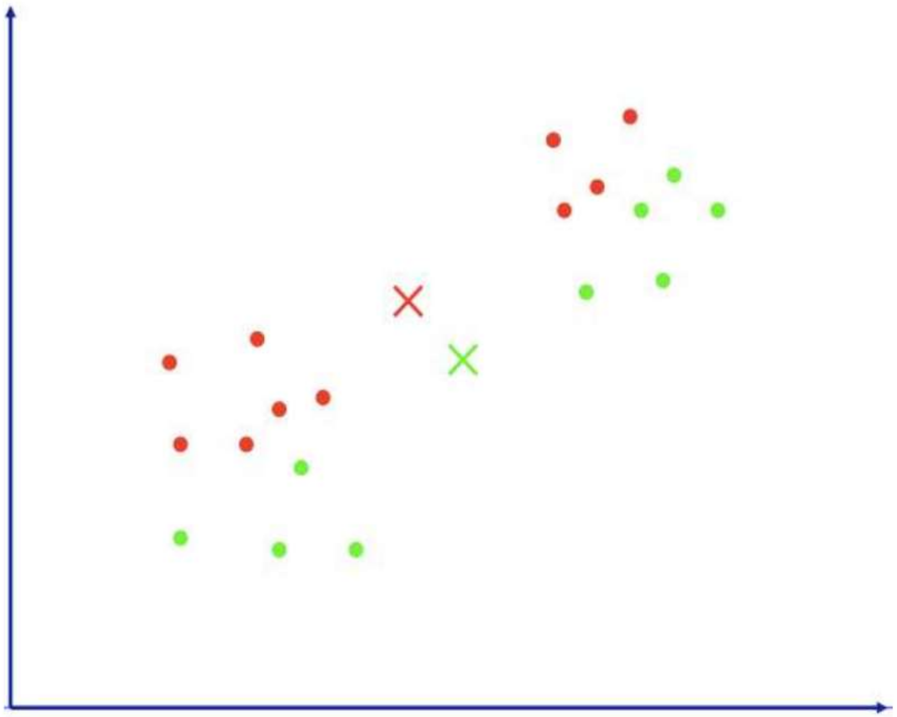
K-means Clustering

- Step 2: Calculate distance from each object to each cluster seed.
- What type of distance should we use?
  - Squared Euclidean distance
- Step 3: Assign each object to the closest cluster.



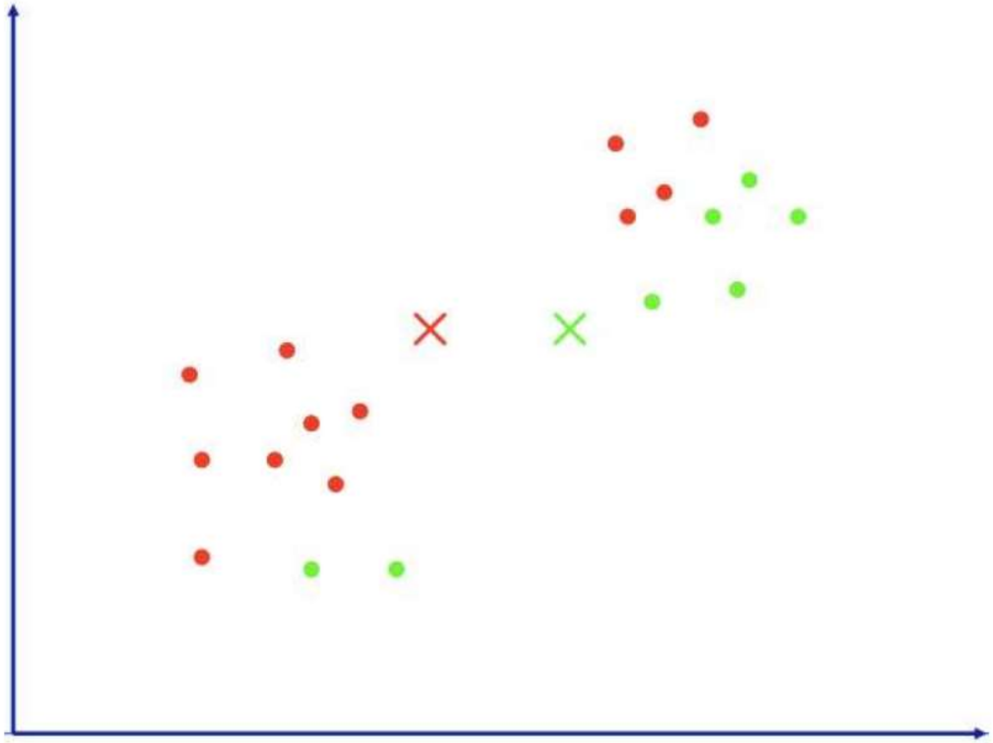
K-means Clustering

- Step 4: Compute the new centroid for each cluster.
- The algorithm repeats until there’s a minimum change of the cluster centers from the last iteration.



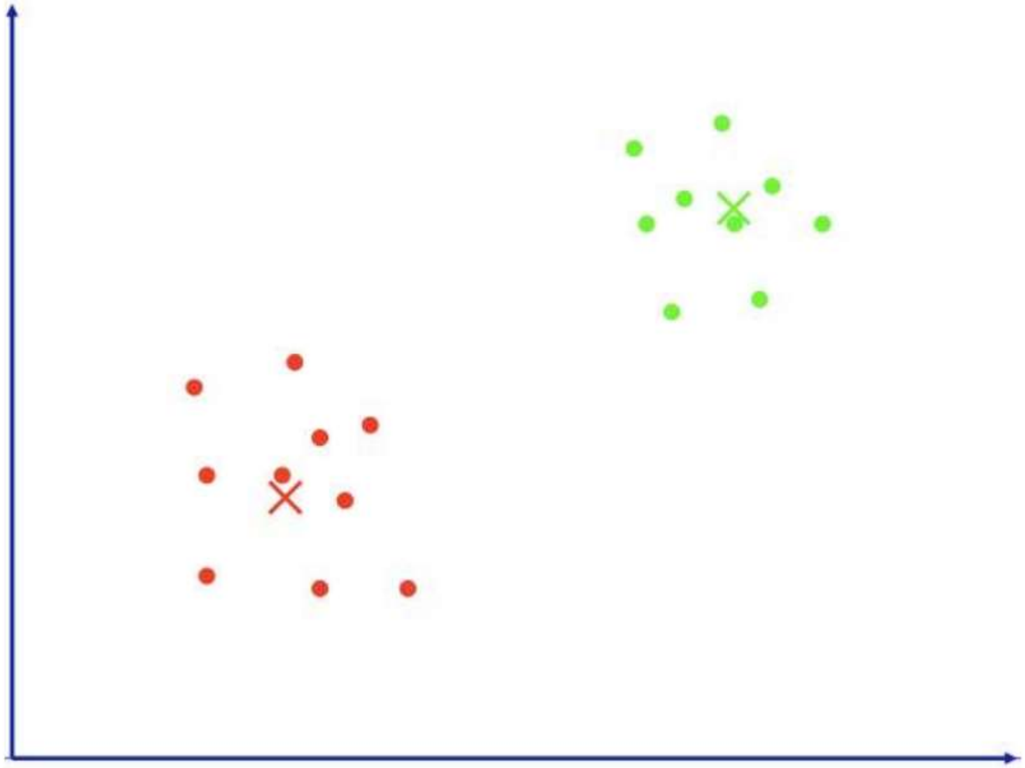
K-means Clustering

- Iterate:
  - Calculate distance from objects to cluste centroids.
  - Assign objects to closest cluster
  - Recalculate new centroids
- Stop based on convergence criteria
  - No change in clusters
  - Max iterations



K-means Clustering

- Iterate:
    - Calculate distance from objects to cluster centroids.
    - Assign objects to closest cluster
    - Recalculate new centroids
- Stop based on convergence criteria
    - No change in clusters
    - Max iterations



K-means Issues

- Distance measure is squared Euclidean
  - Scale should be similar in all dimensions
    - Rescale data?
  - Not good for nominal data. Why?
- Approach tries to minimize the within-cluster sum of squares error (WCSS)
  - Implicit assumption that SSE is similar for each group

WCSS

- The over all WCSS is given by: $\sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$
- The goal is to find the smallest WCSS
- Does this depend on the initial seed values?
- Possibly.

WCSS

- The over all WCSS is given by: $\sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$
- The goal is to find the smallest WCSS
- Does this depend on the initial seed values?
- Possibly.

k - Means Clustering

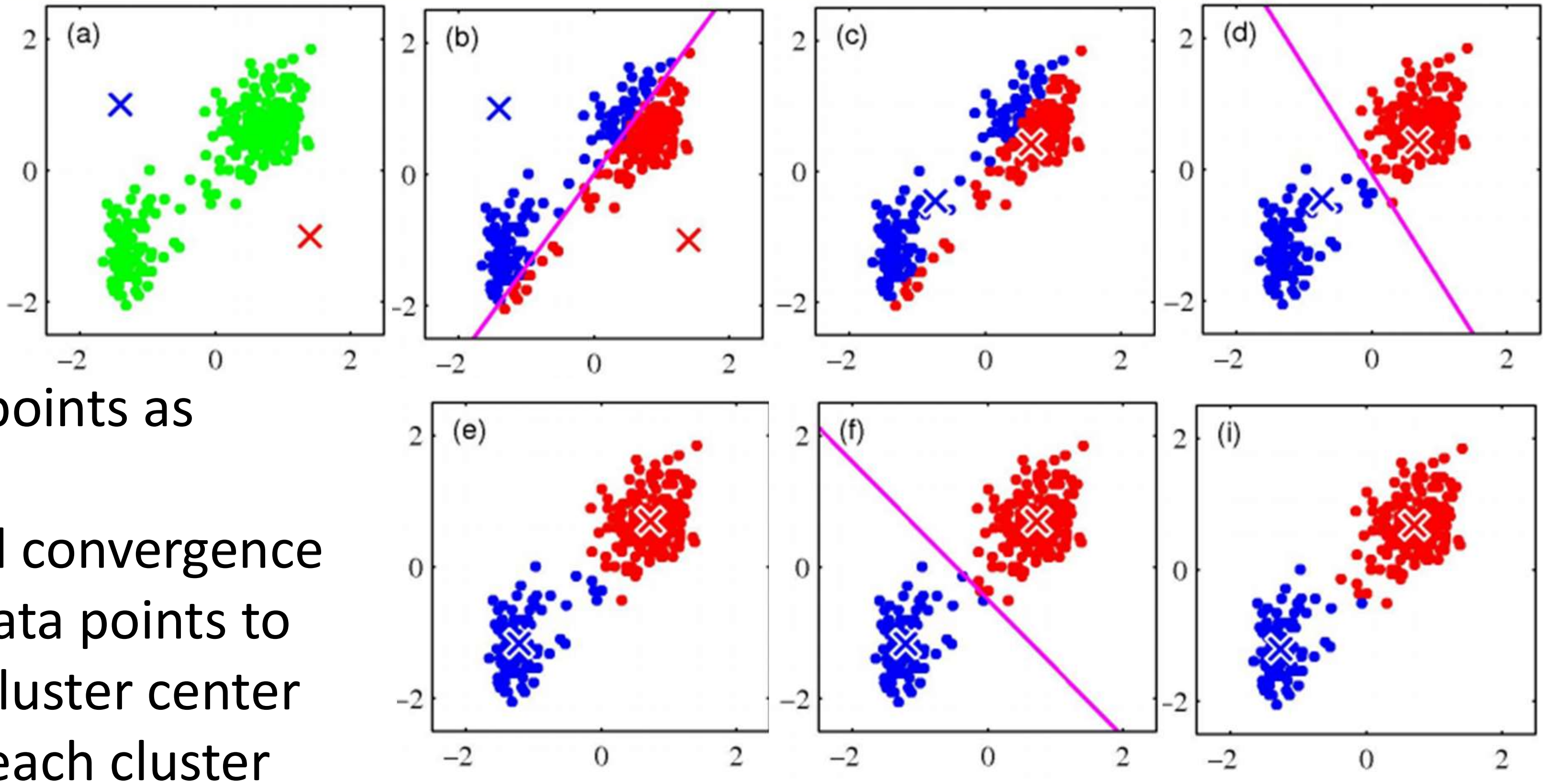
- Requirements
  - An integer k
  - A set of training data (without labels)
  - A metric to measure similarity

- Algorithm
  - Pick k random points as cluster centers
  - Repeat until convergence
    - Assign data points to closest cluster center
    - Update each cluster center to be the mean of its assigned points

Convergence: No points' assignments change

k - Means Clustering

- ✓ Example 1



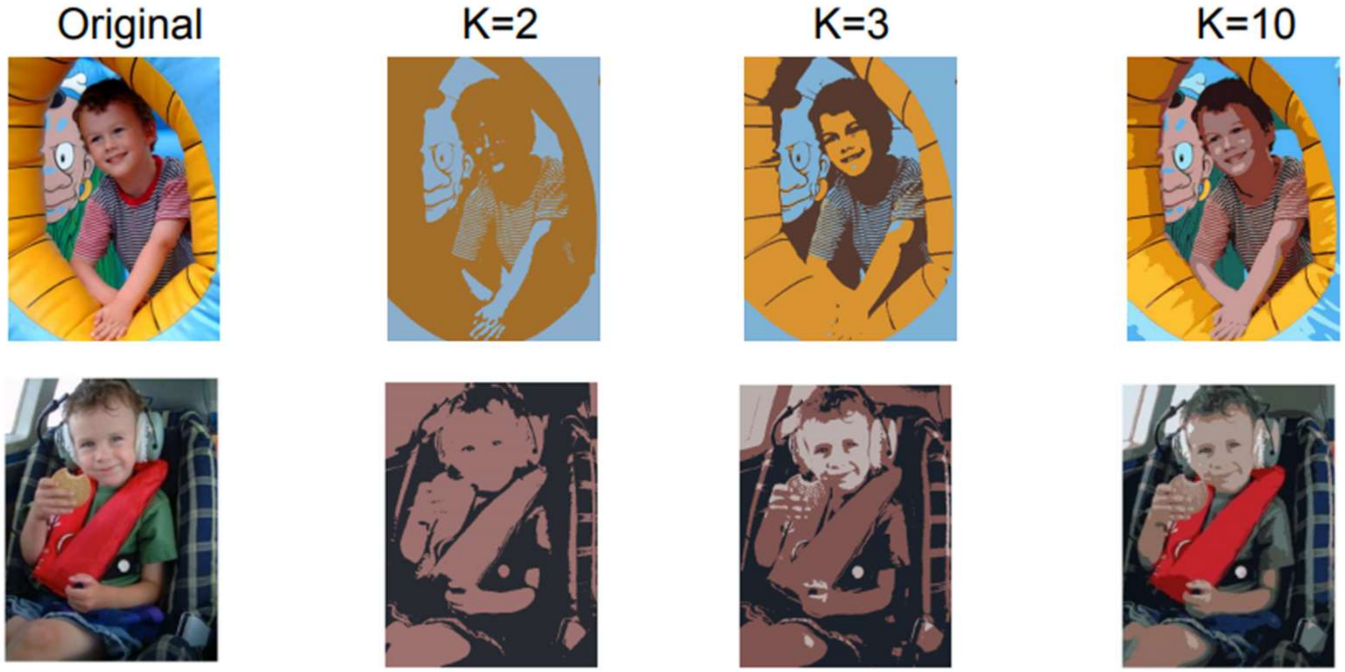
- ✓ Pick k random points as cluster centers
  - ❖ Repeat until convergence
    - ■ Assign data points to closest cluster center
    - ■ Update each cluster center to be the mean of its assigned points

k - Means Clustering

- ✓ Example 2
- ✓ Example 3

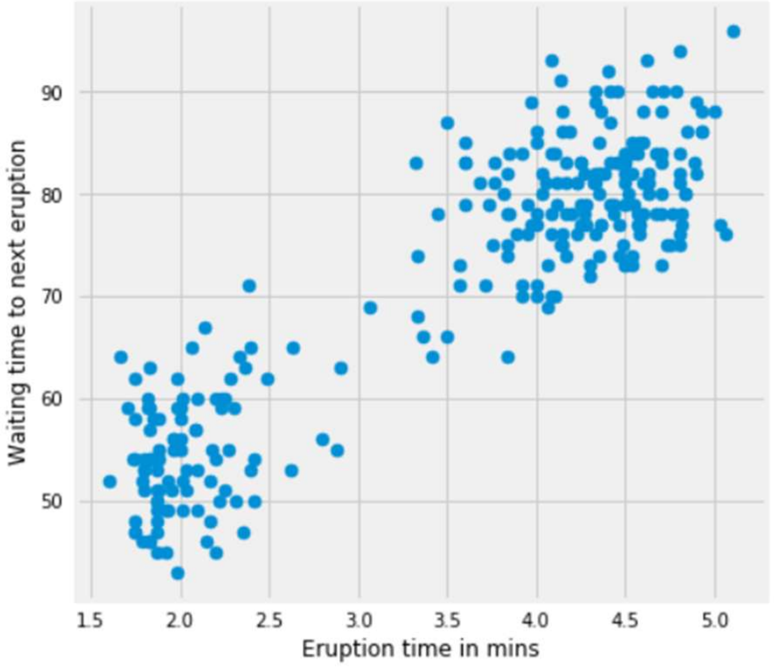
k - Means Clustering

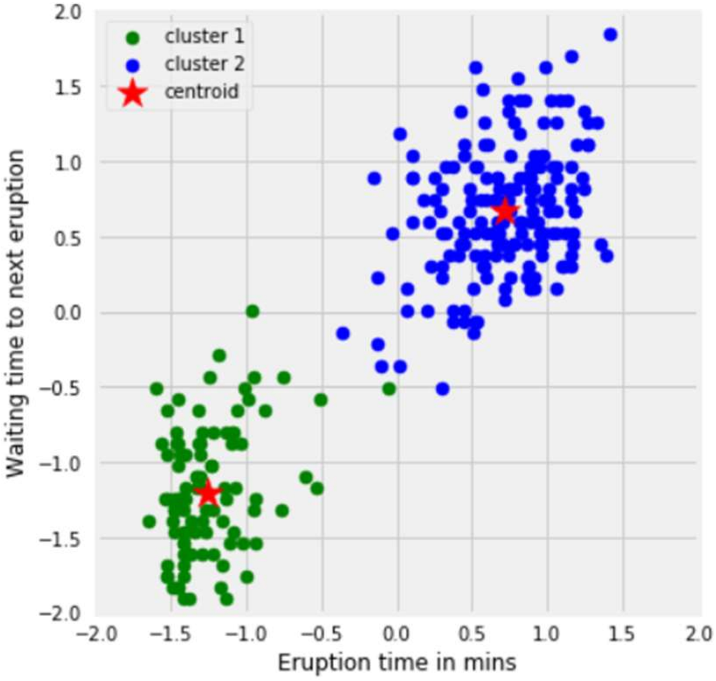
- ✓ Example: Image segmentation
    - ❖ Segmentation: partition an image into regions each of which has reasonably homogenous visual appearance



k - Means Clustering

- ✓ Example: Geyser eruptions
  - ❖ Eruption time (mins)
  - ❖ Waiting time to next eruption (mins)





k - Means Clustering

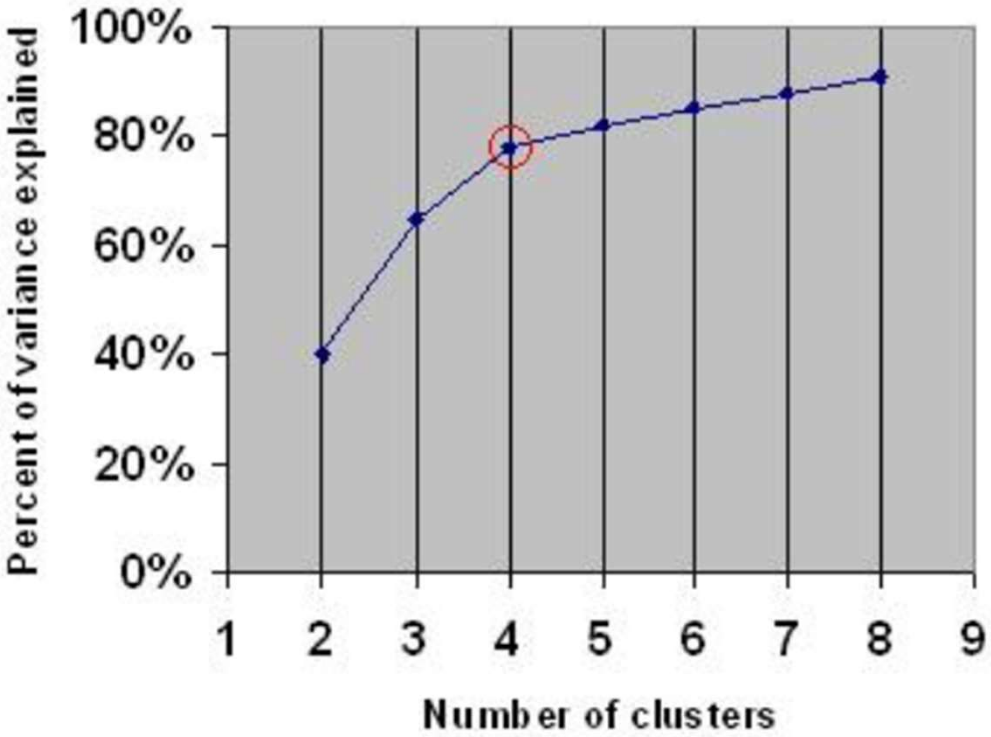
- Properties
  - Guaranteed to converge in a finite number of iterations
  - Running time per iteration
    - Assign data points to closest cluster center
      O(kN)
    - Update the cluster center to be the mean of its assigned points
      O(N)

k - Means Clustering

- How to measure similarity?
  - Similarity is subjective
  - Depends on data, cases, users, etc.
  - Not always straightforward which metrics work well
  - “Trial and error” can be used
  - Examples of similarity measures: Euclidean, Mahattan, cosine distance

k - Means Clustering

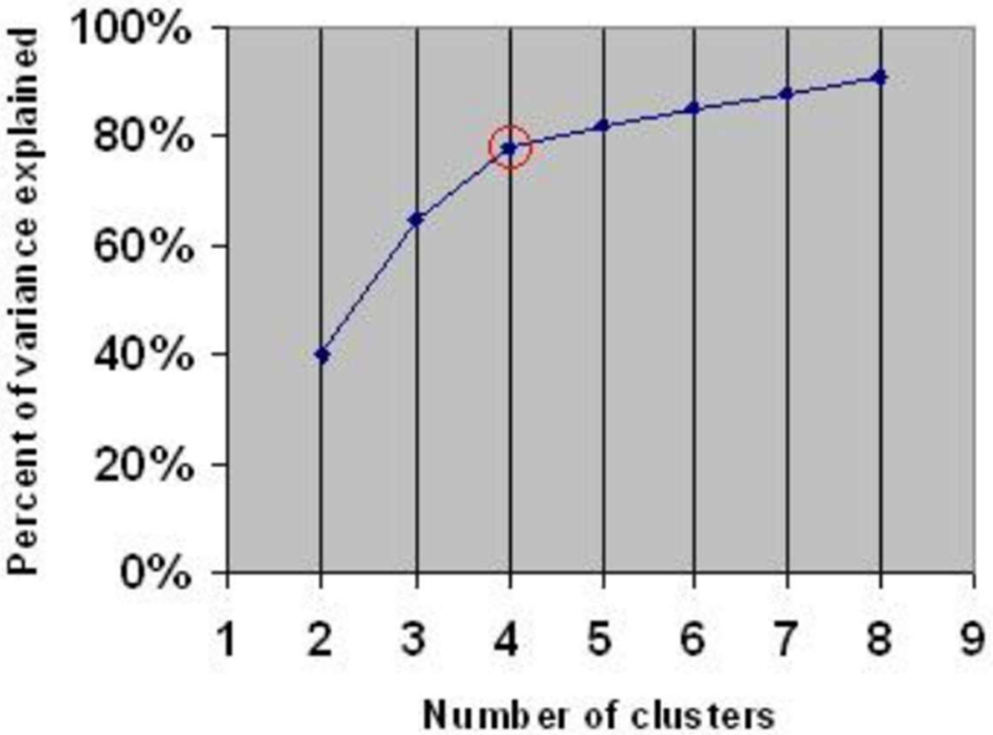
- ✓ How to choose k?
  - ❖ Elbow method



Percentage of variance explained is the ratio of the between-group variance to the total variance

k - Means Clustering

- ✓ How to choose k?
  - ❖ Elbow method

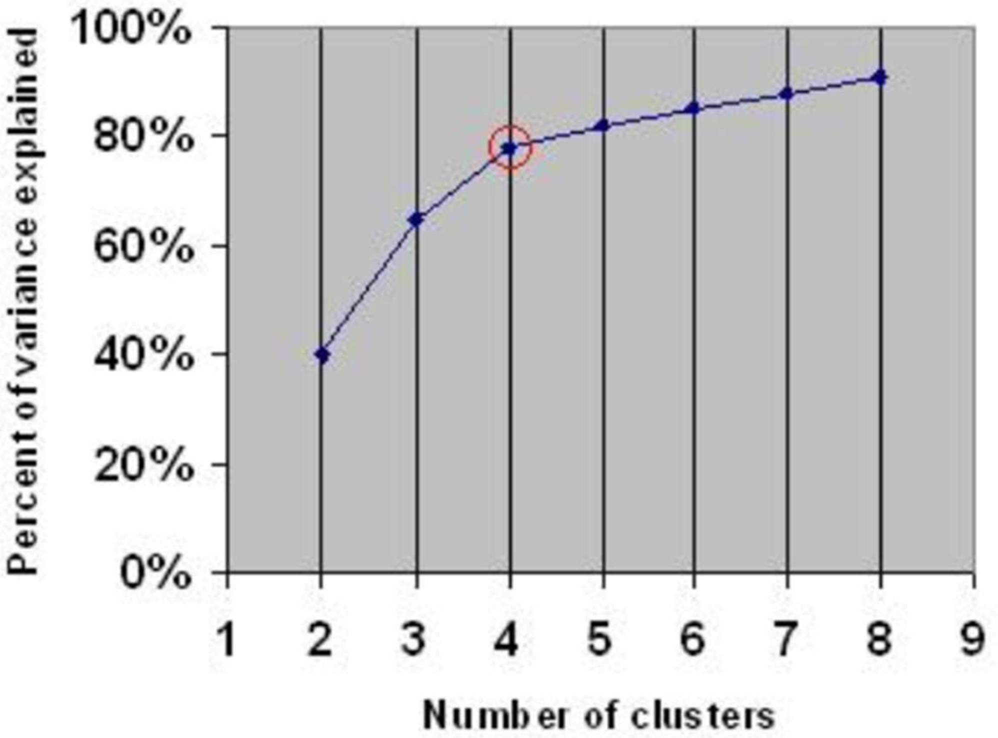


Percentage of variance explained is the ratio of the between-group variance to the total variance

k - Means Clustering

✓ How to choose k?

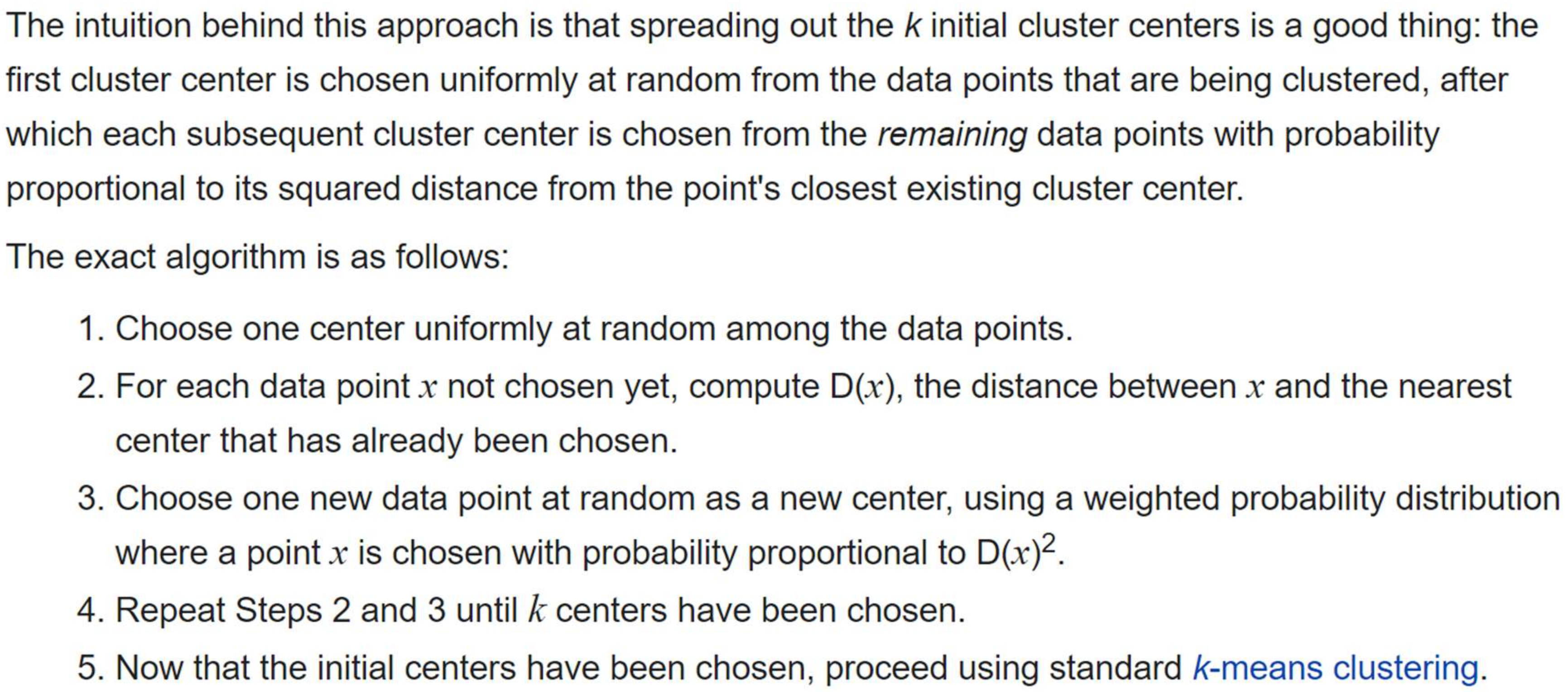
- Elbow method



Percentage of variance explained is the ratio of the between-group variance to the total variance

k - Means Clustering

- How to initialize centroids?
  - K-means ++



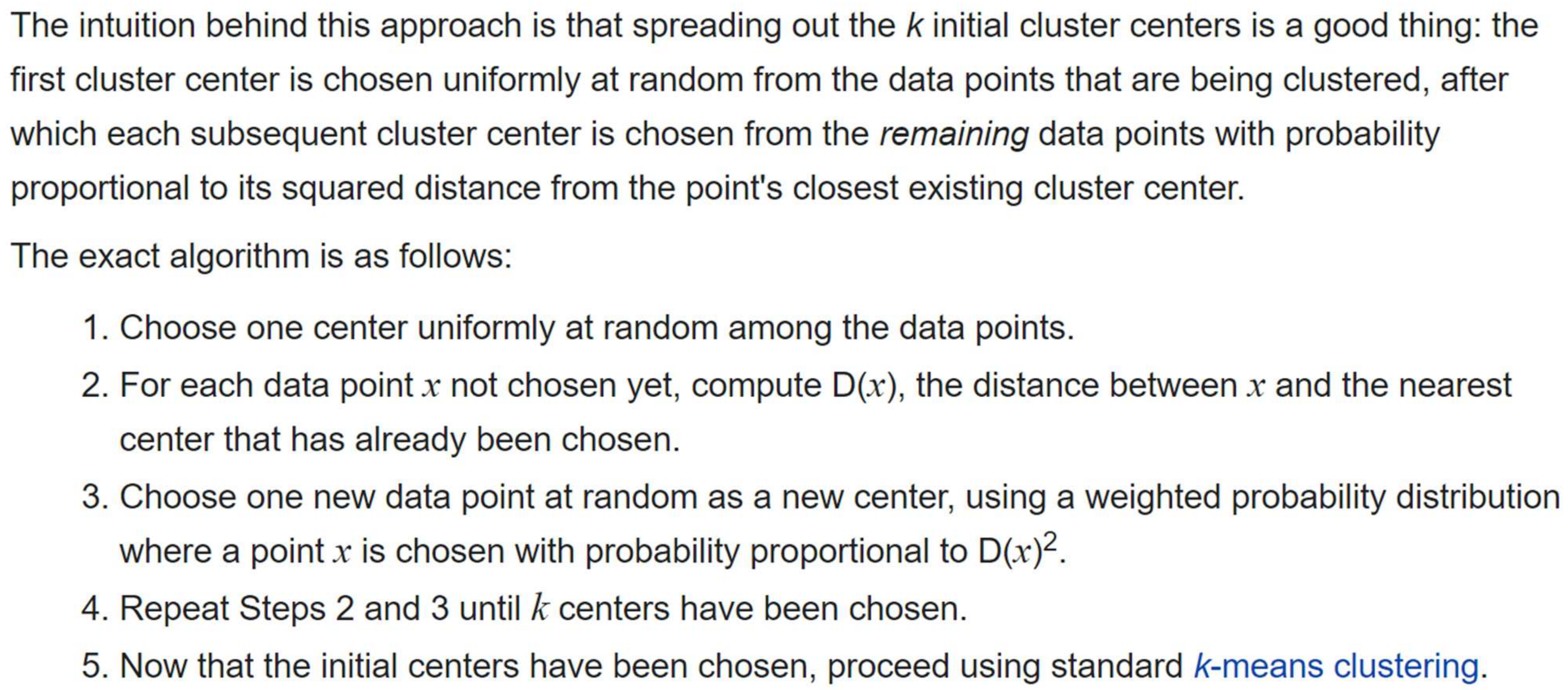
The intuition behind this approach is that spreading out the *k* initial cluster centers is a good thing: the first cluster center is chosen uniformly at random from the data points that are being clustered, after which each subsequent cluster center is chosen from the *remaining* data points with probability proportional to its squared distance from the point's closest existing cluster center.

The exact algorithm is as follows:

1. Choose one center uniformly at random among the data points.
2. For each data point x not chosen yet, compute D(x), the distance between x and the nearest center that has already been chosen.
3. Choose one new data point at random as a new center, using a weighted probability distribution where a point x is chosen with probability proportional to D(x)^2.
4. Repeat Steps 2 and 3 until k centers have been chosen.
5. Now that the initial centers have been chosen, proceed using standard *k*-means clustering.

k - Means Clustering

- How to initialize centroids?
  - K-means ++



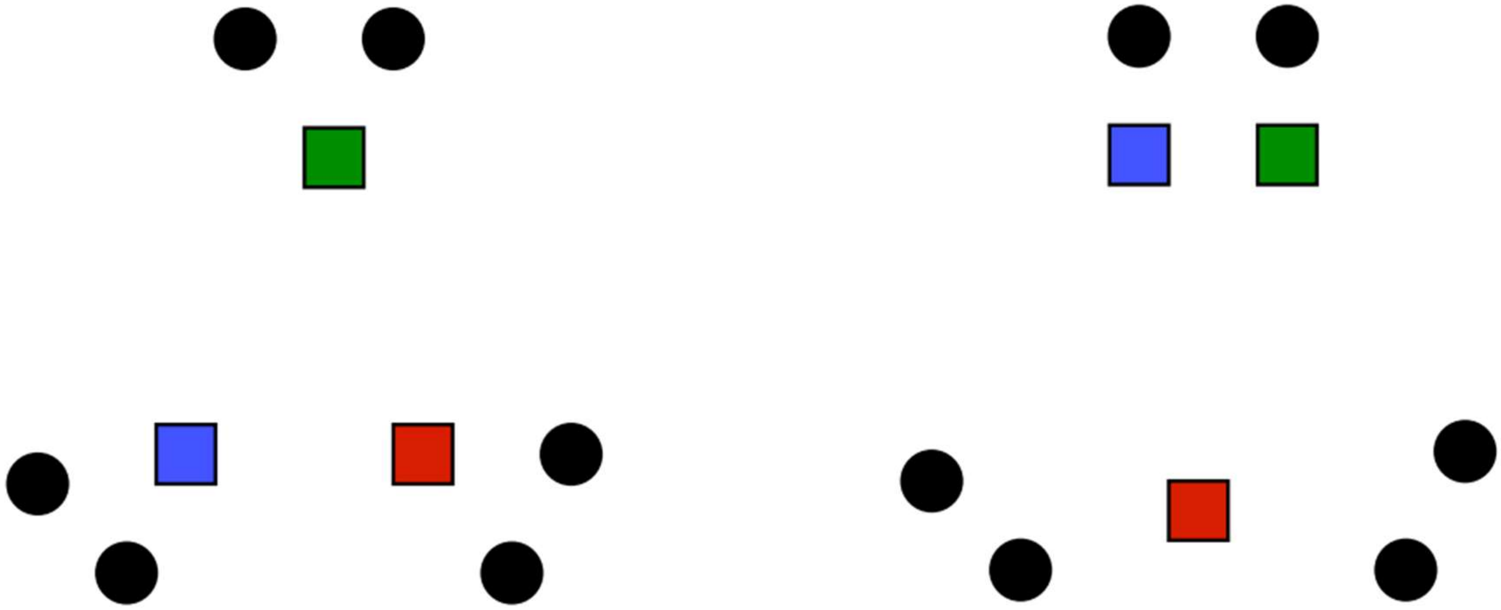
The intuition behind this approach is that spreading out the *k* initial cluster centers is a good thing: the first cluster center is chosen uniformly at random from the data points that are being clustered, after which each subsequent cluster center is chosen from the *remaining* data points with probability proportional to its squared distance from the point's closest existing cluster center.

The exact algorithm is as follows:

1. Choose one center uniformly at random among the data points.
2. For each data point x not chosen yet, compute D(x), the distance between x and the nearest center that has already been chosen.
3. Choose one new data point at random as a new center, using a weighted probability distribution where a point x is chosen with probability proportional to D(x)^2.
4. Repeat Steps 2 and 3 until k centers have been chosen.
5. Now that the initial centers have been chosen, proceed using standard *k*-means clustering.

k - Means Clustering

- k-means clustering: heuristic
  - Requires initial means
  - Does matter what you pick



k - Means Clustering

✓ Drawbacks

1. When the numbers of data are not so many, initial grouping will determine the cluster significantly.
2. The number of cluster, K, must be determined before hand. Its disadvantage is that it does not yield the same result with each run, since the resulting clusters depend on the initial random assignments.
3. We never know the real cluster, using the same data, because if it is inputted in a different order it may produce different cluster if the number of data is few.
4. It is sensitive to initial condition. Different initial condition may produce different result of cluster. The algorithm may be trapped in the *local optimum*.

k - Means Clustering

- Silhouette refers to a method of interpretation and validation of consistency within clusters of data. The technique provides a succinct graphical representation of how well each object has been classified. It was proposed by Belgian statistician Peter Rousseeuw in 1987.
- The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from −1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighbouring clusters.
- If most objects have a high value, then the clustering configuration is appropriate. If many points have a low or negative value, then the clustering configuration may have too many or too few clusters.
- The silhouette can be calculated with any distance metric, such as the Euclidean distance or the Manhattan distance.

Silhouette Coefficient:

- Silhouette Coefficient or silhouette score is a metric used to calculate the goodness of a clustering technique. Its value ranges from -1 to 1.
- 1: Means clusters are well apart from each other and clearly distinguished.
- 0: Means clusters are indifferent, or we can say that the distance between clusters is not significant.
- -1: Means clusters are assigned in the wrong way.

Silhouette Index

- Silhouette analysis refers to a method of interpretation and validation of consistency within clusters of data.
- The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation).
- It can be used to study the separation distance between the resulting clusters. The silhouette plot displays a measure of how close each point in one cluster is to points in the neighbouring clusters and thus provides a way to assess parameters like number of clusters visually.

UNDERSTANDING Silhouette:

- Assume the data have been clustered via any technique, such as k-medoids or k-means, into k clusters.
- For data point i € CI (data point i in the cluster CI), let

$$a(i) = \frac{1}{|C_I| - 1} \sum_{j \in C_I, i \neq j} d(i, j)$$

- be the mean distance between i and all other data points in the same cluster, where | C | is the number of points belonging to cluster i, and d(i,j) is the distance between data points i and j in the cluster CI (we divide by | CI | -1 because we do not include the distance d(i,i) in the sum). We can interpret a(i) as a measure of how well i is assigned to its cluster (the smaller the value, the better the assignment).

- We then define the mean dissimilarity of point i to some cluster CJ as the mean of the distance from i to all points in CJ (where CJ ≠CI).
- For each data point i ∈ CI, we now define

$$b(i) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{j \in C_J} d(i,j)$$

- to be the smallest (hence the min operator in the formula) mean distance of i to all points in any other cluster (i.e., in any cluster of which i is not a member). The cluster with this smallest mean dissimilarity is said to be the "neighbouring cluster" of i because it is the next best fit cluster for point i.
- We now data point I

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \text{ if } |C_I| > 1$$

- And $s(i) = 0, \text{ if } |C_I| = 1$

- Which can be also written as

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases}$$

- From the above definition it is clear that

$$-1 \leq s(i) \leq 1$$

- Note that a(i) is not clearly defined for clusters with size = 1, in which case we set . This choice is arbitrary, but neutral in the sense that it is at the midpoint of the bounds, -1 and 1. For s(i) to be close to 1 we require a(i) << b(i). As a(i) is a measure of how dissimilar i is to its own cluster, a small value means it is well matched. Furthermore, a large b(i) implies that i is badly matched to its neighbouring cluster. Thus an s(i) close to 1 means that the data is appropriately clustered. If s(i) is close to -1, then by the same logic we see that i would be more appropriate if it was clustered in its neighbouring cluster. An s(i) near zero means that the datum is on the border of two natural clusters.

Properties:

Silhouette Values

- A silhouette value is a combination of two scores: cohesion and separation.

Cohesion

- Cohesion measures the similarity of the points in the same cluster. So, we can call it an intra-cluster metric.
- Let C be a cluster and xi, xj∈ C two points in it. Then, we can interpret the distance between them as a measure of their similarity. From there, we define the cohesion of point xi in its cluster xj as the mean distance $a_i = \mathrm{mean}_{x_j \in C}(distance(x_i, x_j))$

Separation

- On the other hand, separation refers to the degree to which the clusters don’t overlap. So, it’s an inter-cluster metric.
- Intuitively, the distance between the clusters speaks about the “goodness of their separation”. So, we define the separation of xi ∈ C1 as the minimum mean distance between xi and other clusters:

$$b_i = \min_{C_2 \neq C_1} (mean_{x_j \in C_2}(distance(x_i, x_j)))$$

Combining Cohesion and Separation into a Silhouette Value

- Then, the silhouette value of a point x is:

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

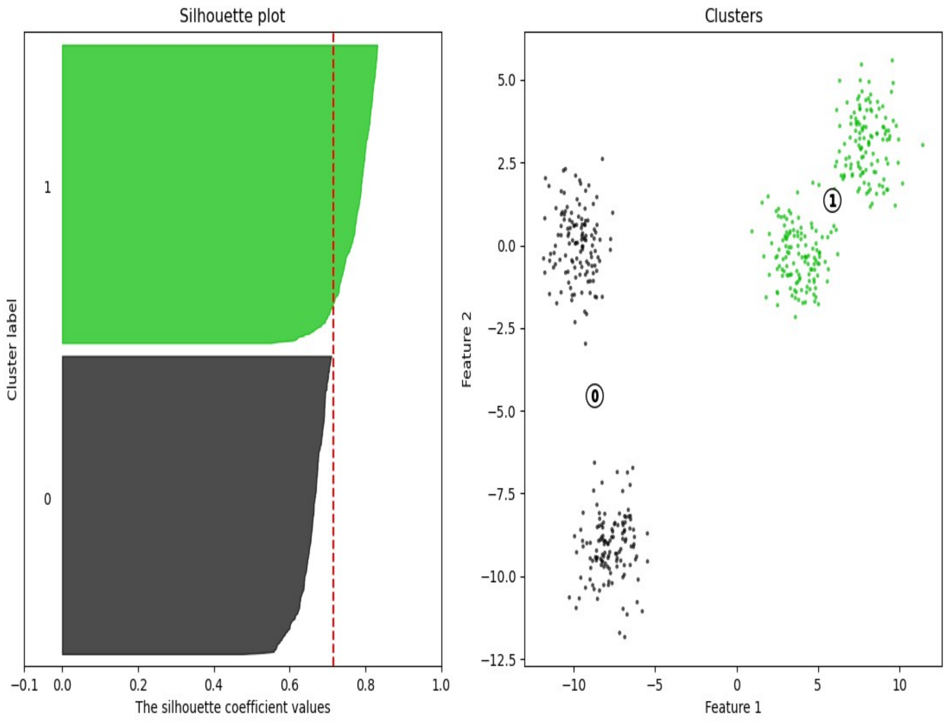
- Its range is [-1 ,1]. The higher the silhouette value, the more certain we can be that its label is correct. So, a high mean silhouette value of all the points indicates a good clustering.

Calculation of Silhouette Value

- If the Silhouette index value is high, the object is well-matched to its own cluster and poorly matched to neighbouring clusters. The Silhouette Coefficient is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample. The Silhouette Coefficient is defined as –
- S(i) = $\frac{(b(i) - a(i))}{max \{ (a(i), b(i)) \}}$
- Where,
- a(i) is the average dissimilarity of ith object to all other objects in the same cluster
- b(i) is the average dissimilarity of ith object with all objects in the closest cluster.

Silhouette Plots

- The silhouette of a cluster visualizes the silhouette values s_i of all the points in it in the decreasing order. A silhouette plot shows the silhouettes of all the clusters in random order. Additionally, it inserts blank spaces between consecutive clusters and can color them differently.
- For example, here’s a plot for four clusters we got with the K-Means clustering algorithm on an ad-hoc two-dimensional dataset:

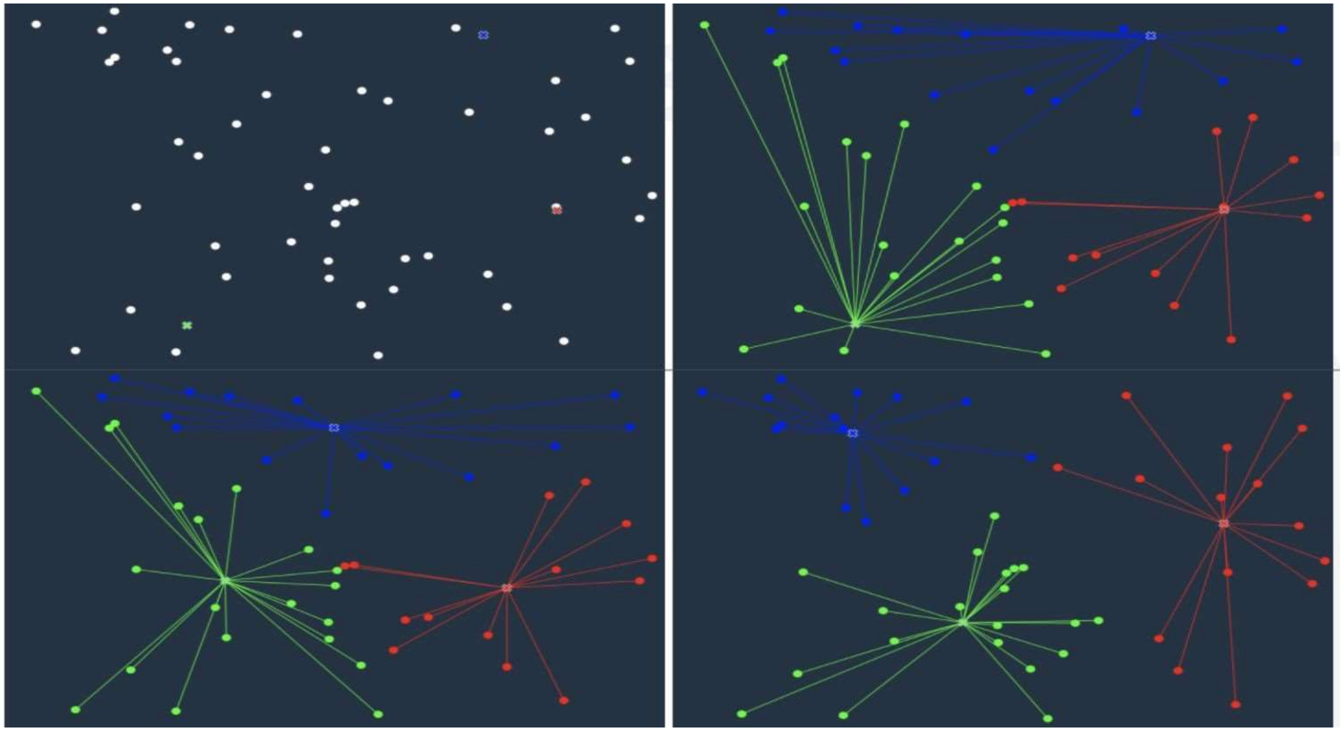


k - Means Exercise

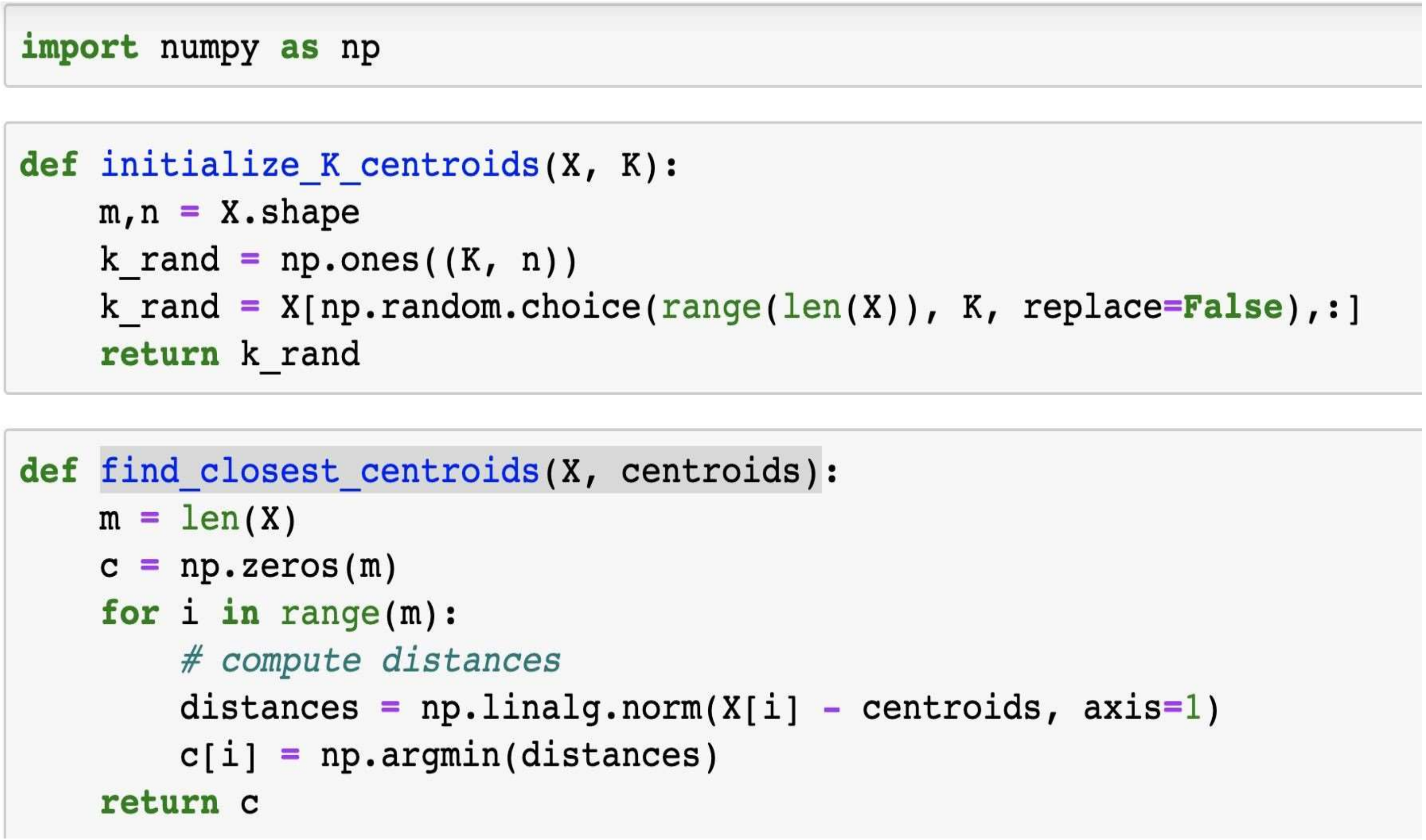
- **Given the centroids of 2 clusters** of 2D data as follows:
  - **Centroid of cluster 1**: (1, 5)
  - **Centroid of cluster 2**: (4, 1)
- **Assume there are 3 data samples A, B, and C with the feature vectors** as follows:
  - A: (1.1, 1.2)
  - B: (2.0, 3.0)
  - C: (6.3, 1.5)
- **Which cluster do these data samples belong to?**

K means Demo

- Demo http://tech.nitoyon.com/ja/blog/2013/11/07/k-means/
-



Demo 1

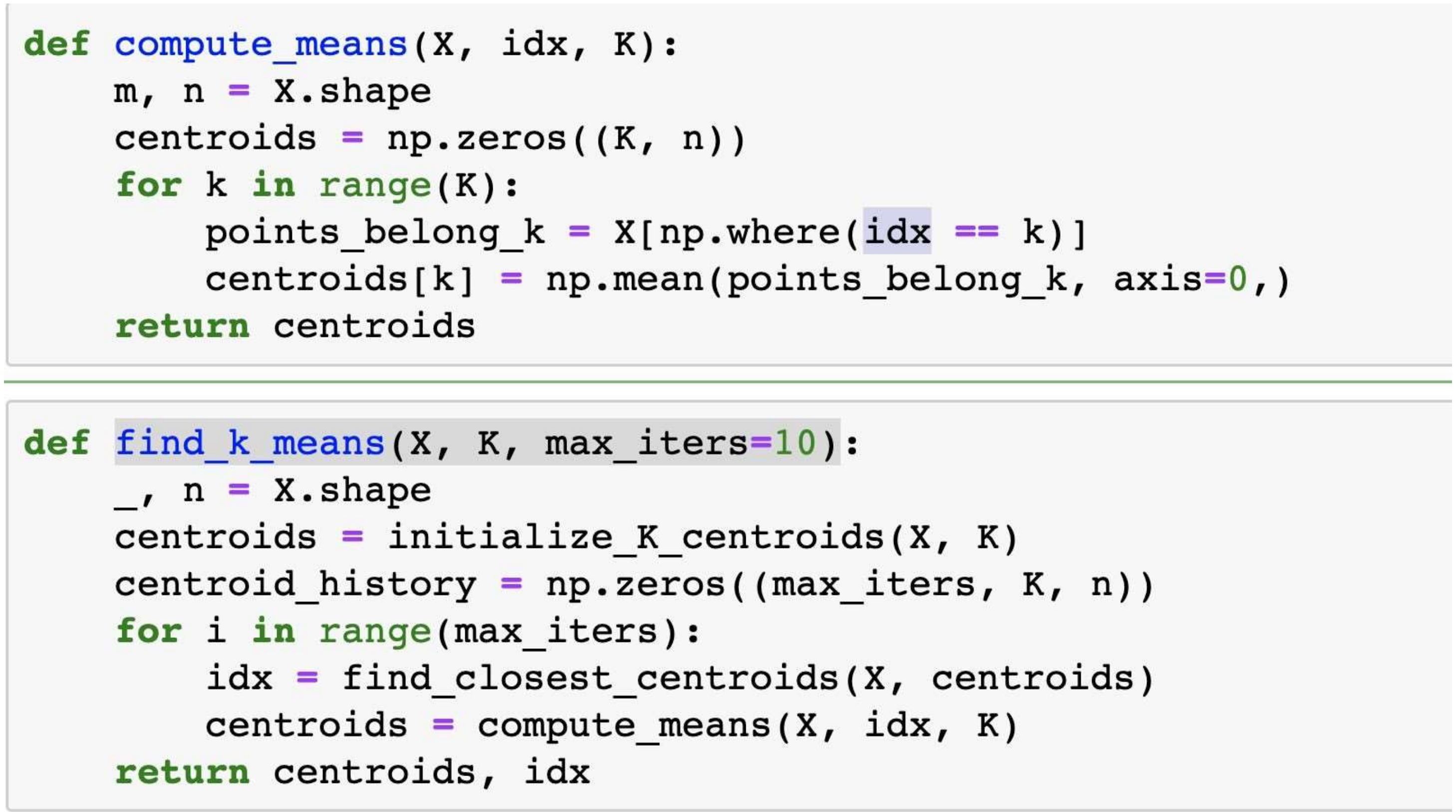


```python
import numpy as np
```

```python
def initialize_K_centroids(X, K):
    m,n = X.shape
    k_rand = np.ones((K, n))
    k_rand = X[np.random.choice(range(len(X)), K, replace=False),:]
    return k_rand
```

```python
def find_closest_centroids(X, centroids):
    m = len(X)
    c = np.zeros(m)
    for i in range(m):
        # compute distances
        distances = np.linalg.norm(X[i] - centroids, axis=1)
        c[i] = np.argmin(distances)
    return c
```

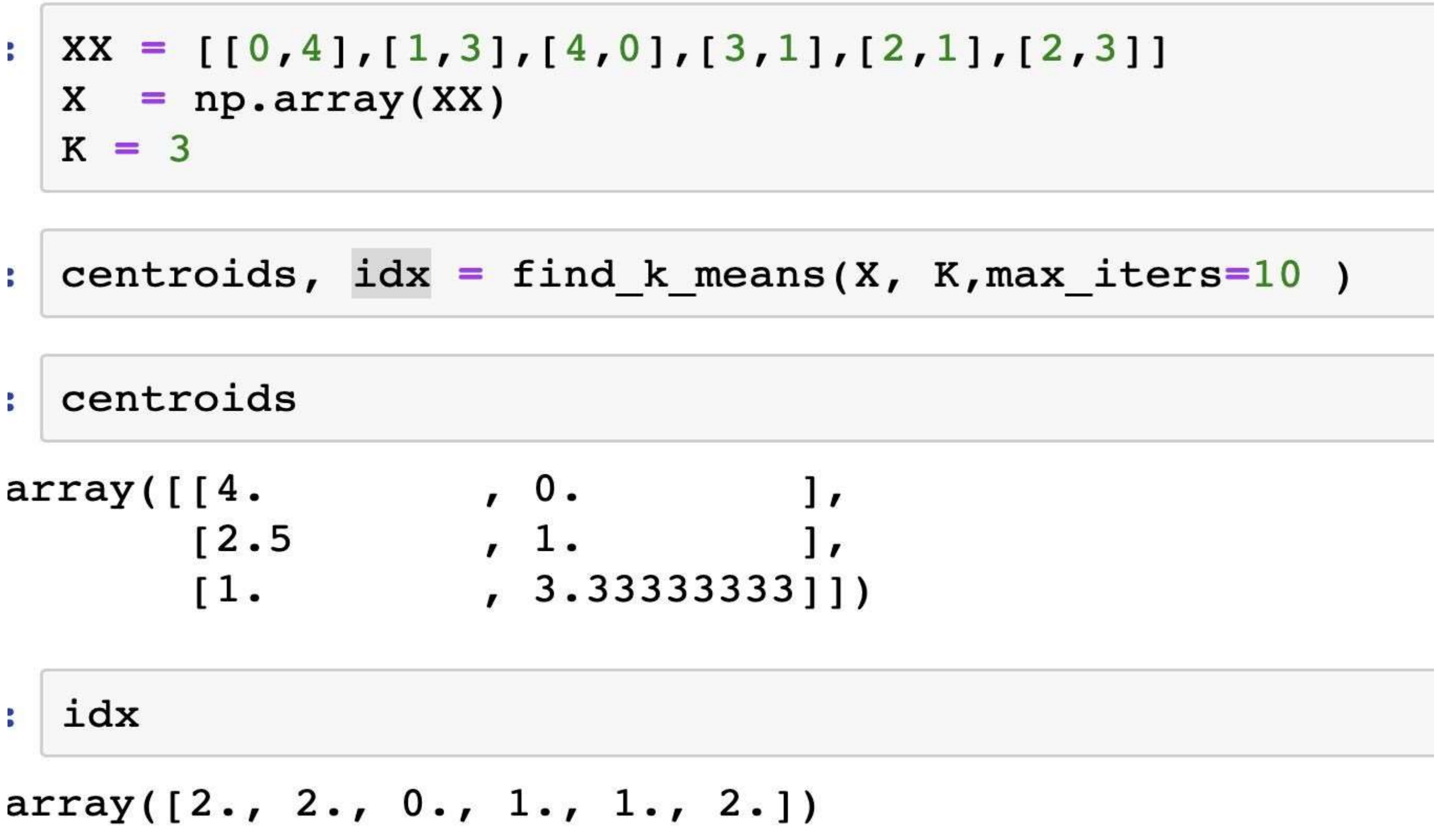
Demo 1 (t.)



```python
def compute_means(X, idx, K):
    m, n = X.shape
    centroids = np.zeros((K, n))
    for k in range(K):
        points_belong_k = X[np.where(idx == k)]
        centroids[k] = np.mean(points_belong_k, axis=0,)
    return centroids
```

```python
def find_k_means(X, K, max_iters=10):
    _, n = X.shape
    centroids = initialize_K_centroids(X, K)
    centroid_history = np.zeros((max_iters, K, n))
    for i in range(max_iters):
        idx = find_closest_centroids(X, centroids)
        centroids = compute_means(X, idx, K)
    return centroids, idx
```

Demo 1(t.)



```
XX = [[0,4],[1,3],[4,0],[3,1],[2,1],[2,3]]
X  = np.array(XX)
K = 3
```

```
centroids, idx = find_k_means(X, K,max_iters=10 )
```

```
centroids
```

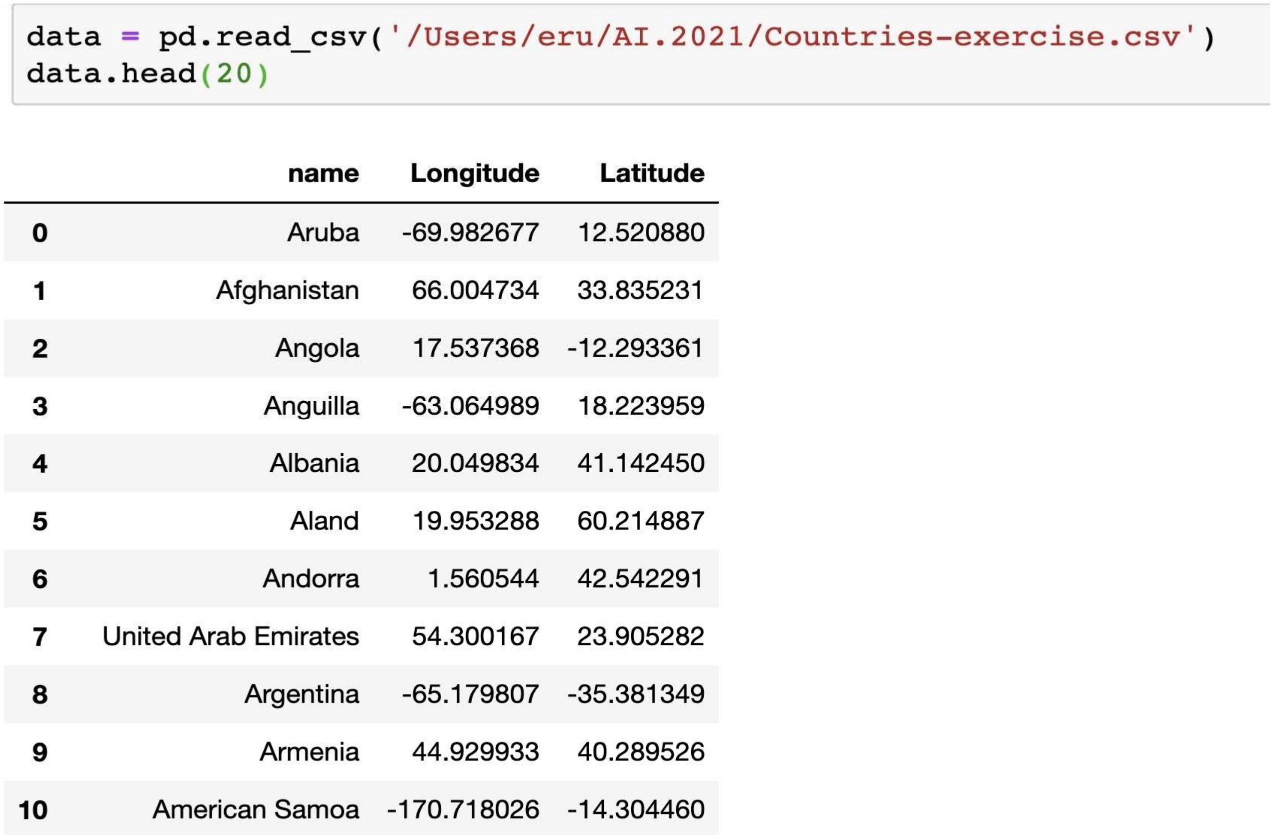
```
array([[4.        , 0.        ],
       [2.5       , 1.        ],
       [1.        , 3.33333333]])
```

```
idx
```

```
array([2., 2., 0., 1., 1., 2.])
```

k_rand

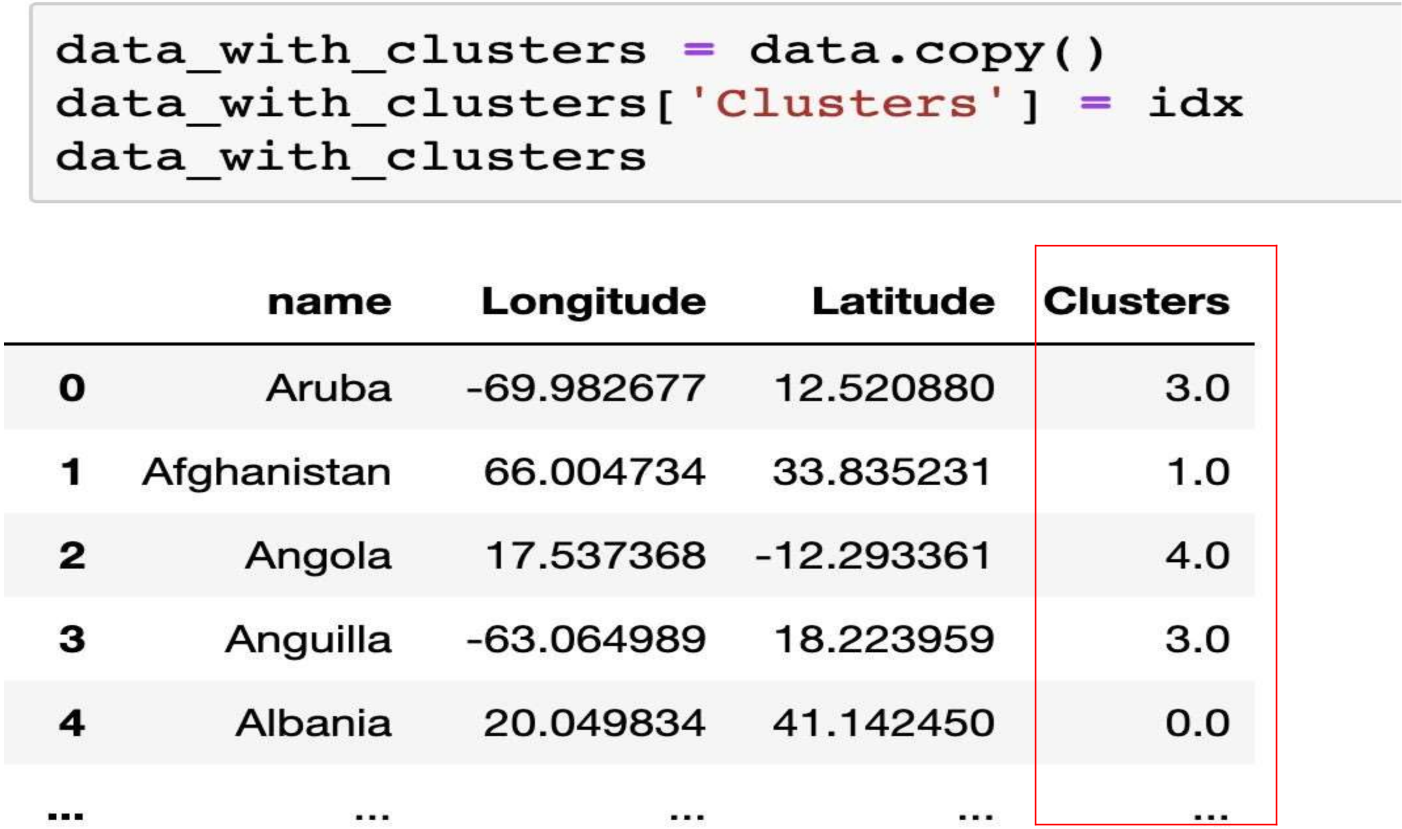
Demo 2



```
data = pd.read_csv('/Users/eru/AI.2021/Countries-exercise.csv')
data.head(20)
```

	name	Longitude	Latitude
0	Aruba	-69.982677	12.520880
1	Afghanistan	66.004734	33.835231
2	Angola	17.537368	-12.293361
3	Anguilla	-63.064989	18.223959
4	Albania	20.049834	41.142450
5	Aland	19.953288	60.214887
6	Andorra	1.560544	42.542291
7	United Arab Emirates	54.300167	23.905282
8	Argentina	-65.179807	-35.381349
9	Armenia	44.929933	40.289526
10	American Samoa	-170.718026	-14.304460

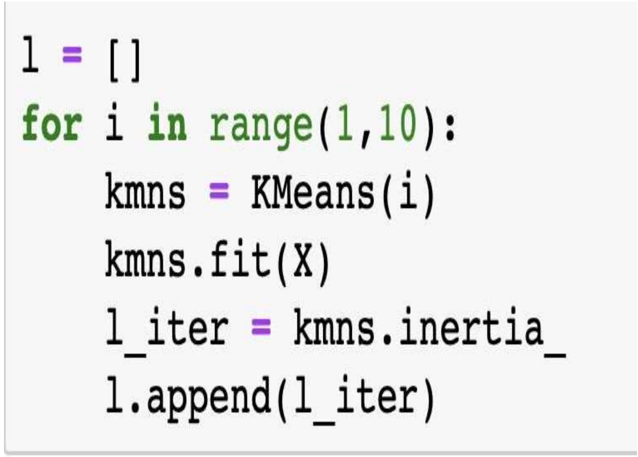
Demo 2



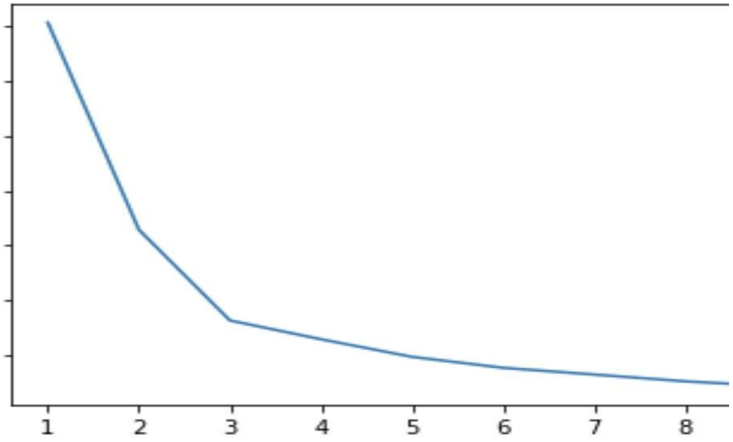
```python
data_with_clusters = data.copy()
data_with_clusters['Clusters'] = idx
data_with_clusters
```

	name	Longitude	Latitude	Clusters
0	Aruba	-69.982677	12.520880	3.0
1	Afghanistan	66.004734	33.835231	1.0
2	Angola	17.537368	-12.293361	4.0
3	Anguilla	-63.064989	18.223959	3.0
4	Albania	20.049834	41.142450	0.0
...	...	...	...	...

Demo 3



```
l = []
for i in range(1,10):
    kmns = KMeans(i)
    kmns.fit(X)
    l_iter = kmns.inertia_
    l.append(l_iter)
```



k - Means Clustering

- ✓ Sources:
  - ❖ http://people.csail.mit.edu/dsontag/courses/ml12/slides/lecture14.pdf
  - ❖ https://www.slideshare.net/annafensel/kmeans-clustering-122651195
  - ❖ https://en.wikipedia.org/wiki/Elbow_method_(clustering)
  - ❖ https://www2.stat.duke.edu/courses/Fall02/sta290/datasets/geyser
  - ❖ Pham Viet Cuong Dept. Control Engineering & Automation, FEEE Ho Chi Minh City University of Technology.