Machine Learning

Lesson 7—Unsupervised Learning with Clustering

Learning Objectives

- Discuss clustering algorithms
- Explain k-means clustering with examples

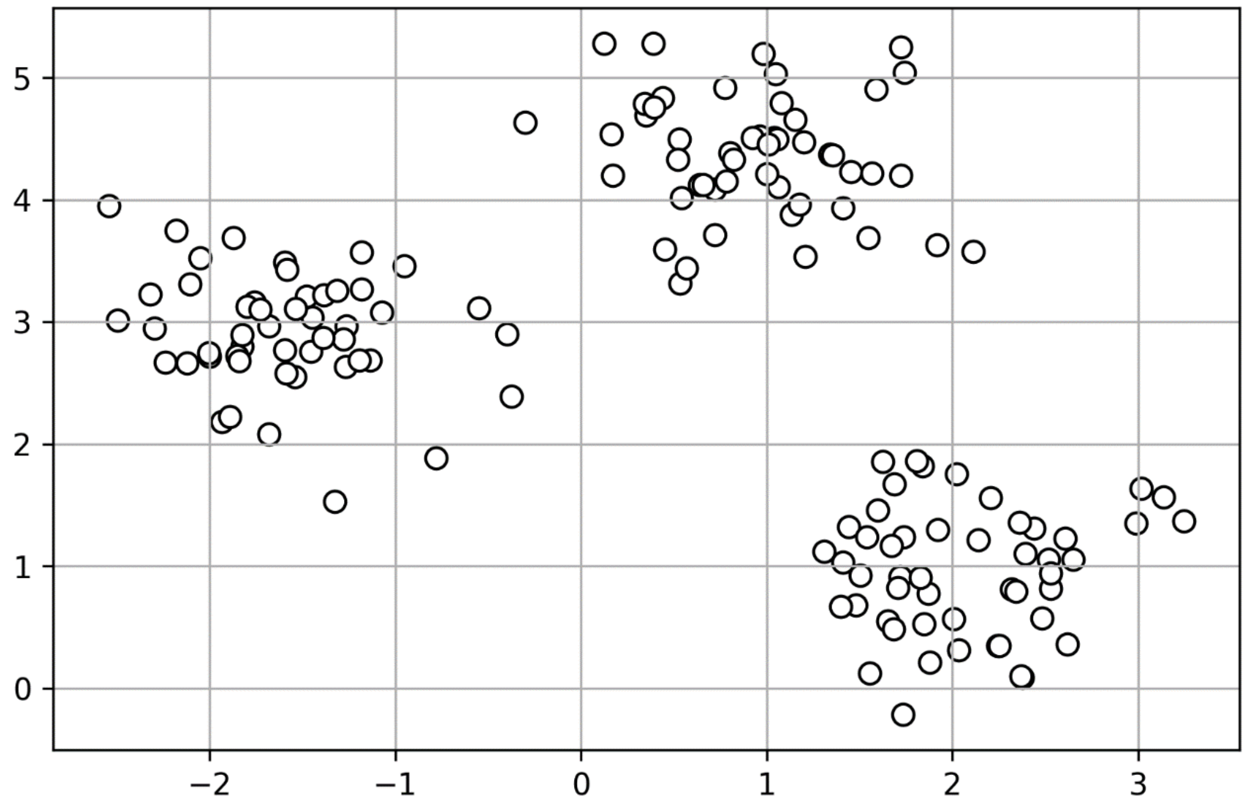
Unsupervised Learning with Clustering

Topic 1—Clustering Algorithms

Recall: Clustering

> Cluster analysis or clustering is the most commonly used technique of unsupervised learning. It is used to find data clusters such that each cluster has the most closely matched data.

Clustering Algorithms



- **Prototype-based Clustering**
- Hierarchical Clustering
- Density-based Clustering (DBSCAN)

This lesson will focus on Prototype-based Clustering.

Source Credit : "Python Machine Learning" by Sebastian Raschka

Prototype-based Clustering

Prototype-based clustering assumes that most data is located near prototypes; example: centroids (average) or medoid (most frequently occurring point)

K-means, a Prototype-based method, is the most popular method for clustering that involves:

- Training data that gets assigned to matching cluster based on similarity
- Iterative process to get data points in the best clusters possible

Source Credit : "Python Machine Learning" by Sebastian Raschka

Unsupervised Learning with Clustering

Topic 2—K-means Clustering

K-means Clustering: Example

PROBLEM STATEMENT

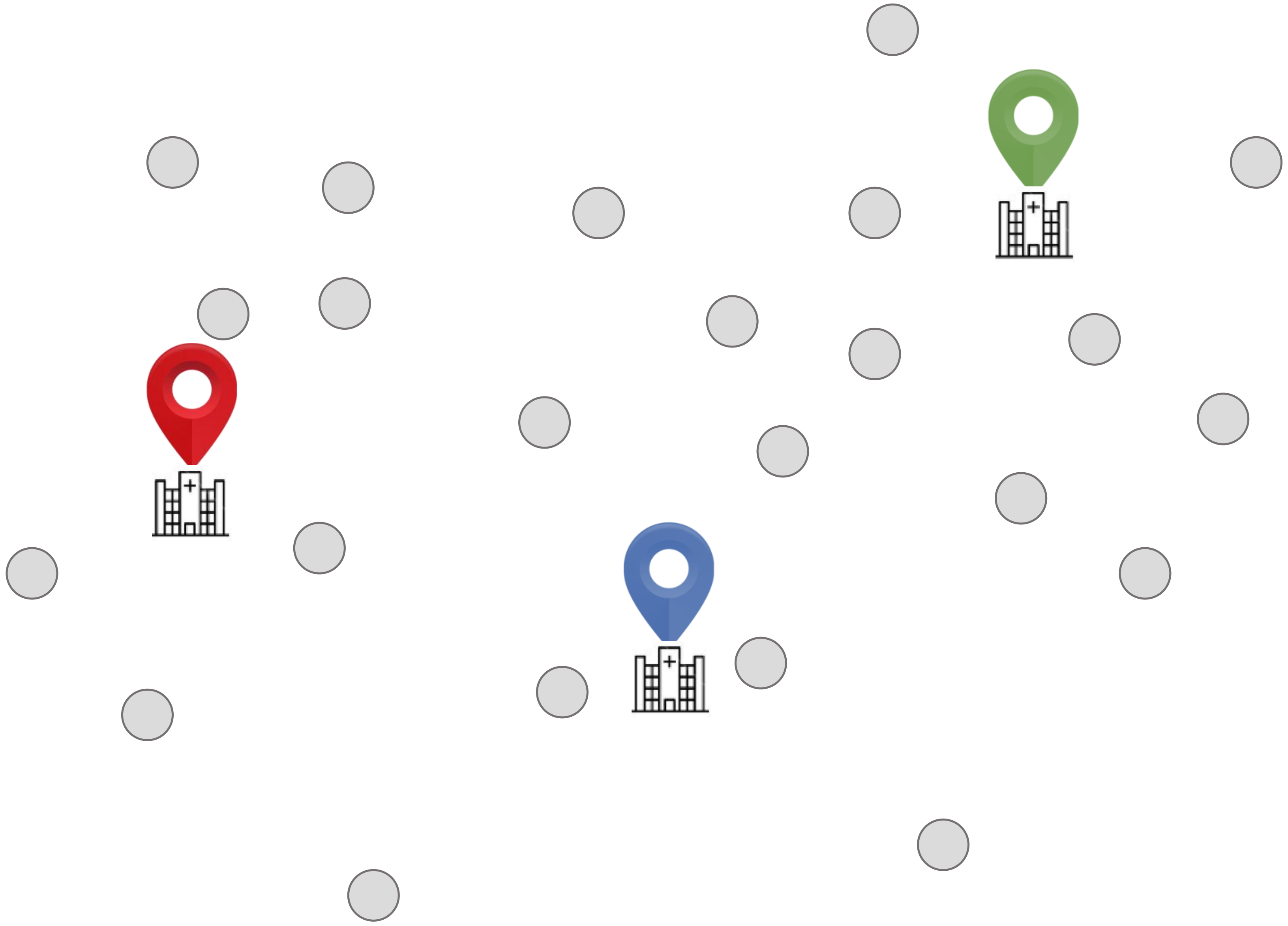
Let’s say, in California the government tries to identify high density clusters to build hospitals (no other ground truth or features are provided apart from the population data). How can the clusters be identified?

K-means Clustering: Example

STEP 1: RANDOMLY PICK K CENTROIDS

Start by picking k centroids. Assume, k = 3

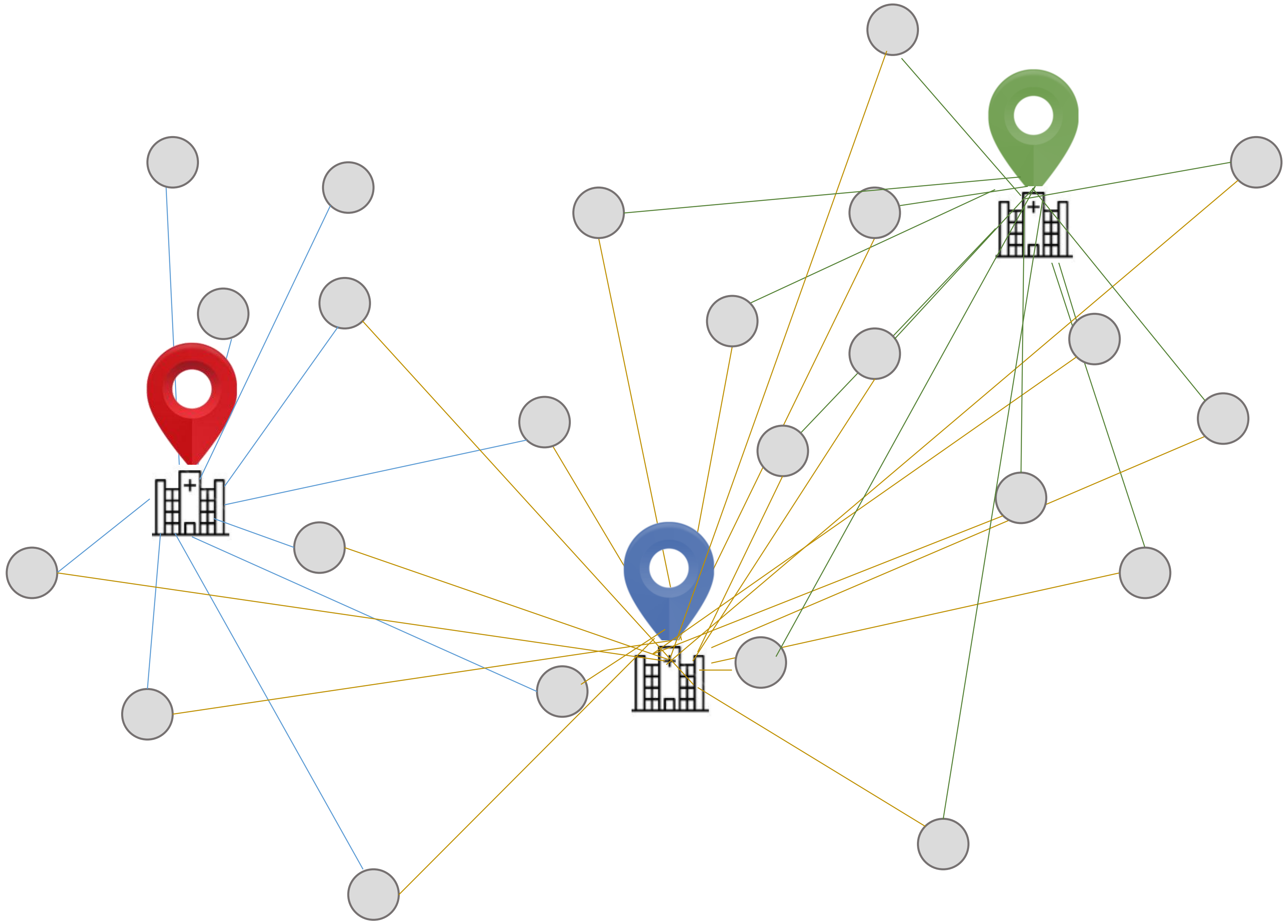
Finding the number of clusters: Use **Elbow Method (to be reviewed later)**



K-means Clustering: Example

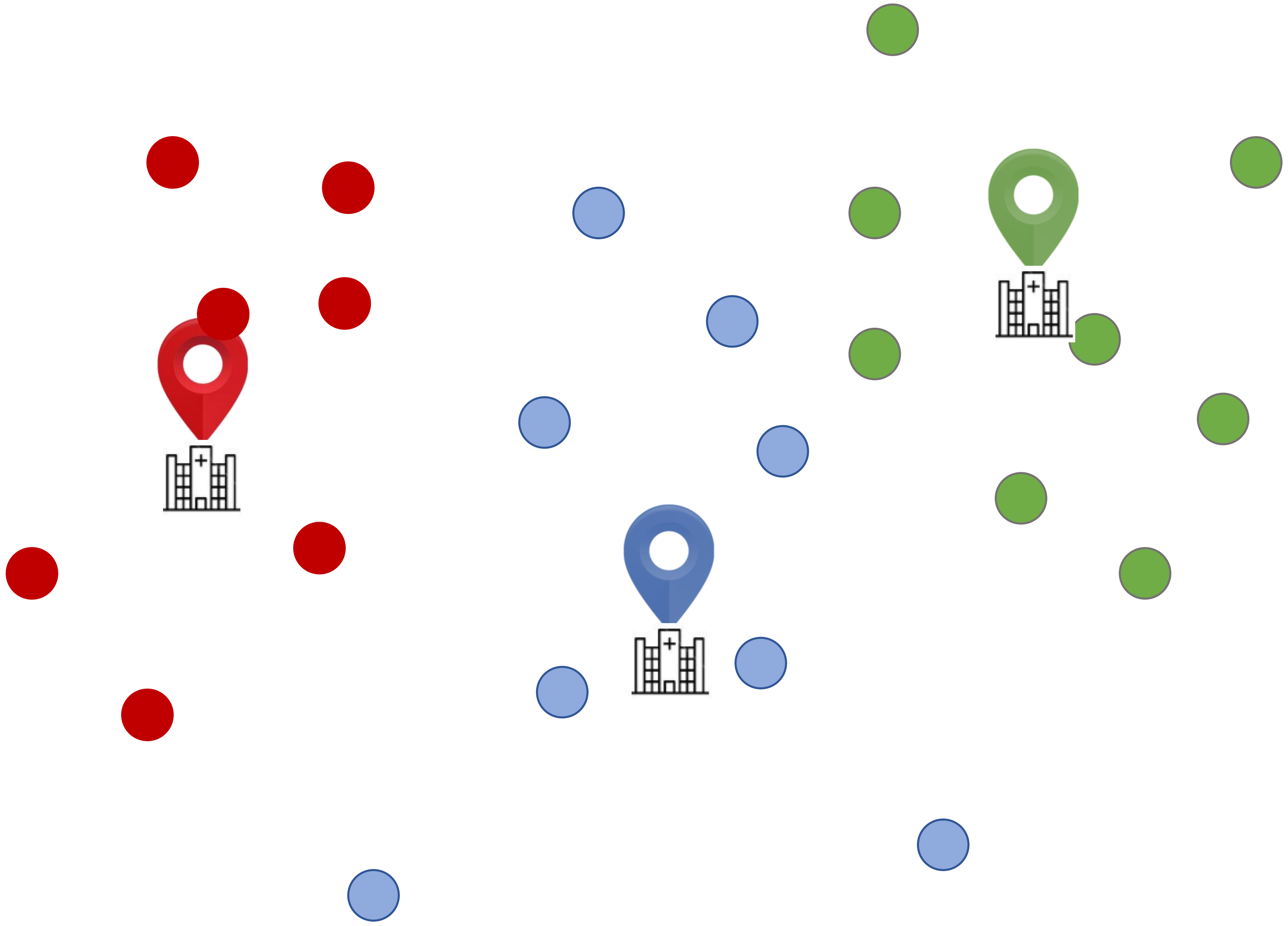
STEP 2: ASSIGN EACH POINT TO THE NEAREST CENTROID μ(j), j ∈ {1,…,k}.

The points are assigned such that the Euclidean distance of each point from the respective centroid is minimized.



K-means Clustering: Example

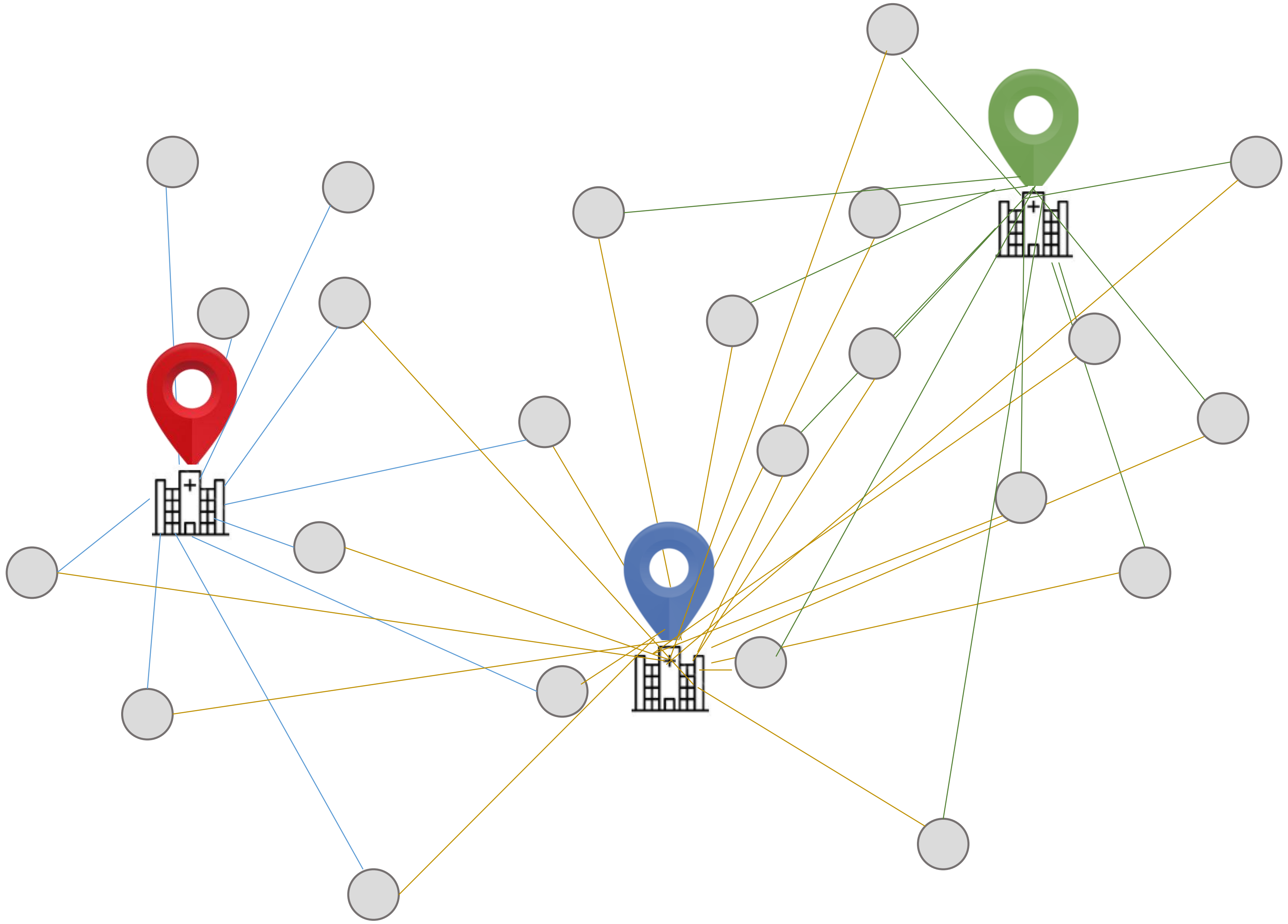
STEP 3: MOVE EACH CENTROID TO THE CENTRE OF THE RESPECTIVE CLUSTER



K-means Clustering: Example

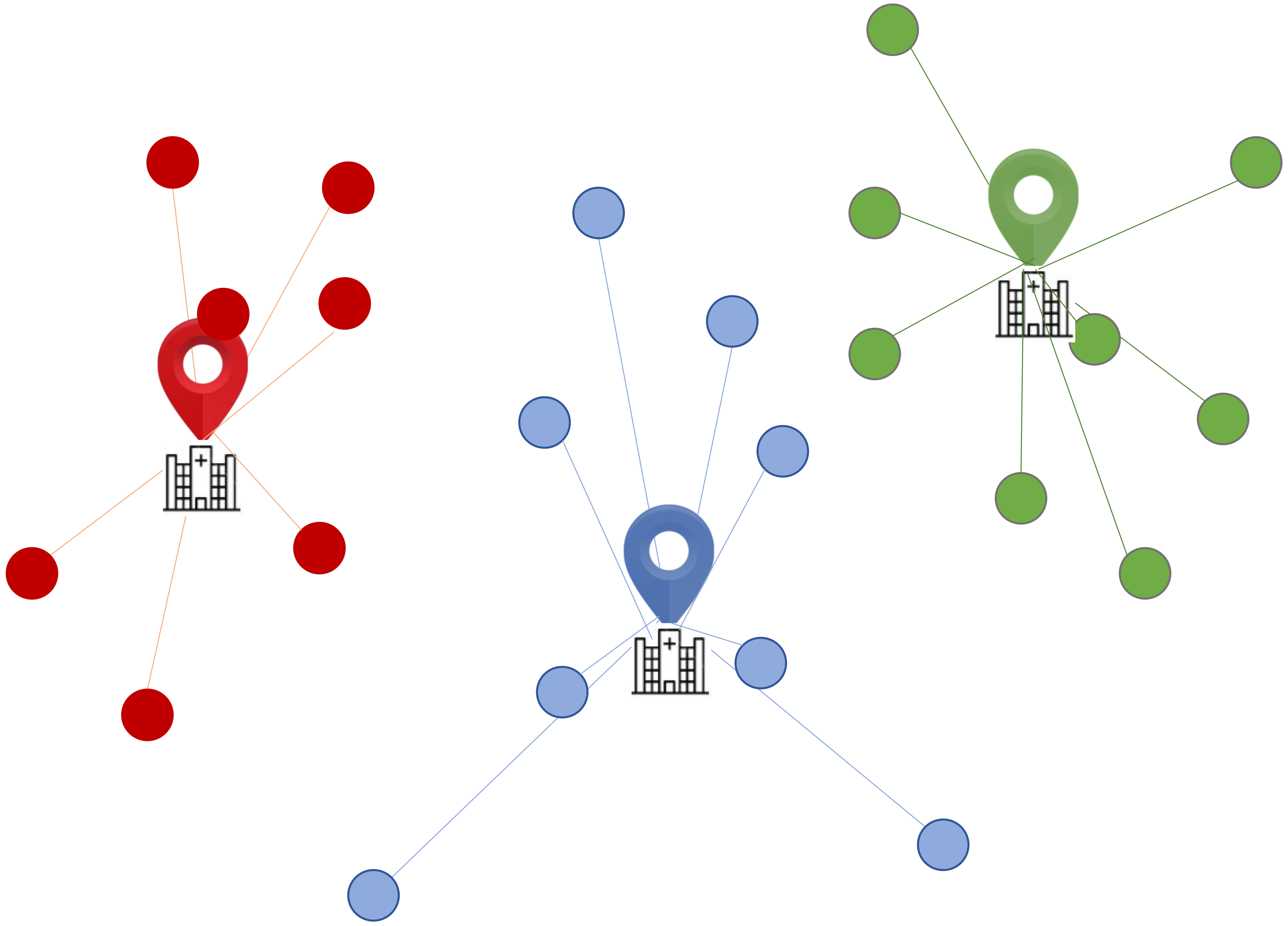
STEP 4: CALCULATE DISTANCE OF THE CENTROIDS FROM EACH POINT AGAIN

Calculate the **Euclidean distance between each point and its centroid.**



K-means Clustering: Example

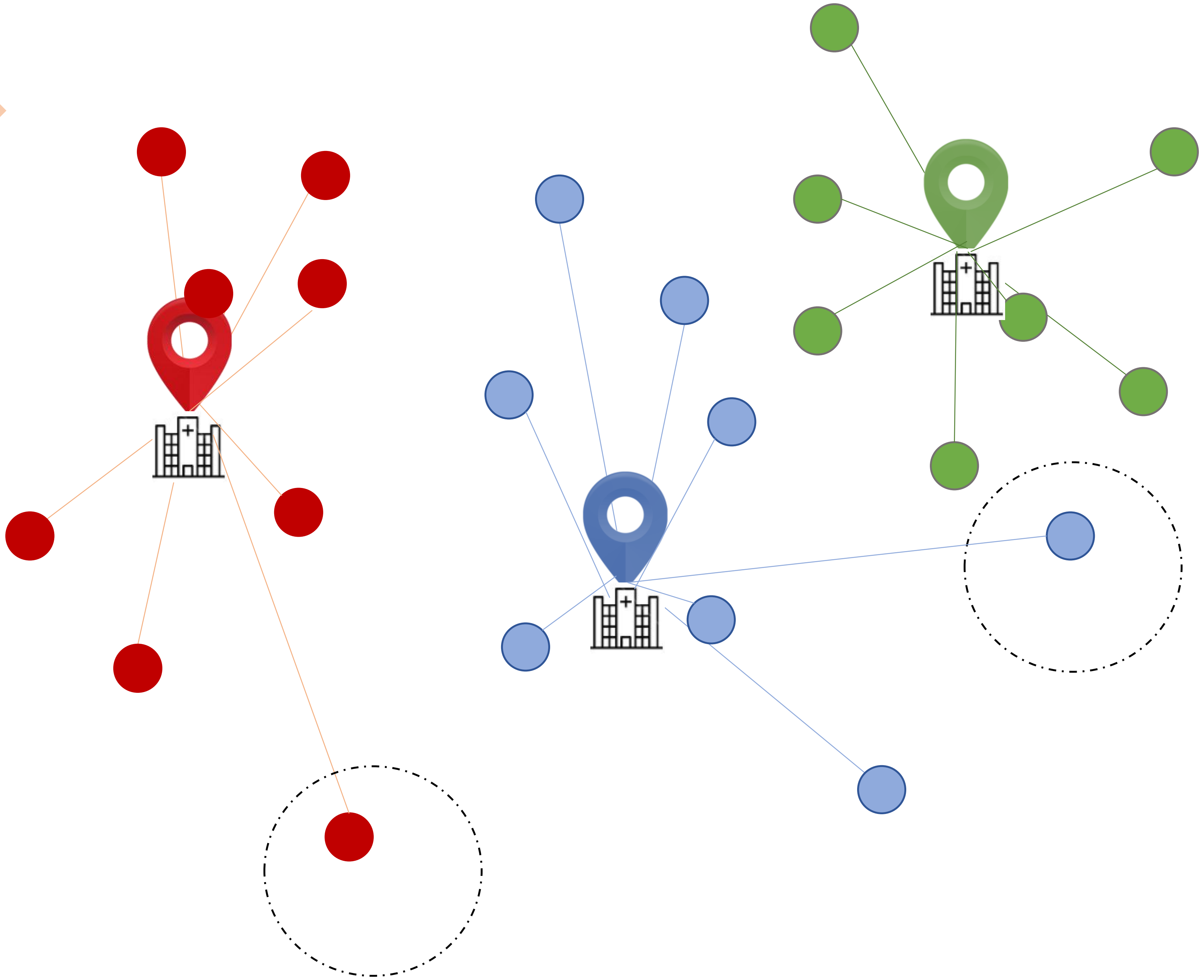
STEP 5: MOVE POINTS ACROSS CLUSTERS AND RE-CALCULATE THE DISTANCE FROM THE CENTROID



K-means Clustering: Example

STEP 6: KEEP MOVING THE POINTS ACROSS CLUSTERS UNTIL THE EUCLIDEAN DISTANCE IS MINIMIZED

Repeat the steps until within-cluster Euclidean distance is minimized for each cluster (or a user-defined limit on number of iterations is reached)



Giving It a Mathematical Angle

The analysis was based on a lot of calculations.

Now let’s understand the mathematical aspect.

Created by Bamdewanto - Freepik.com

K-means Clustering: Example

MATHEMATICAL REPRESENTATION

Finding the number of clusters: Use **Elbow Method**

Calculate the **Euclidian distance**

- A key challenge in Clustering is that you have to pre-set the number of clusters. This influences the quality of clustering.
- Unlike Supervised Learning, here one does not have ground truth labels. Hence, to check quality of clustering, one has to use intrinsic methods, such as the within-cluster SSE, also called Distortion.
- In the scikit-learn ML library, this value is available via the *inertia_* attribute after fitting a K-means model.

SSE is sum of squared errors. It represents the sum of distances of points within a cluster from its centroid.

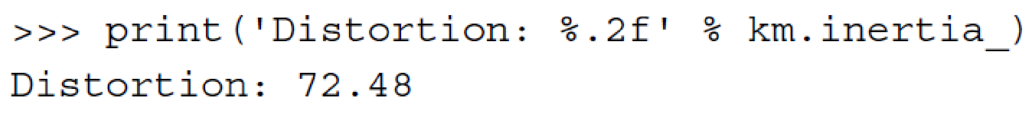
K-means Clustering: Example

MATHEMATICAL REPRESENTATION

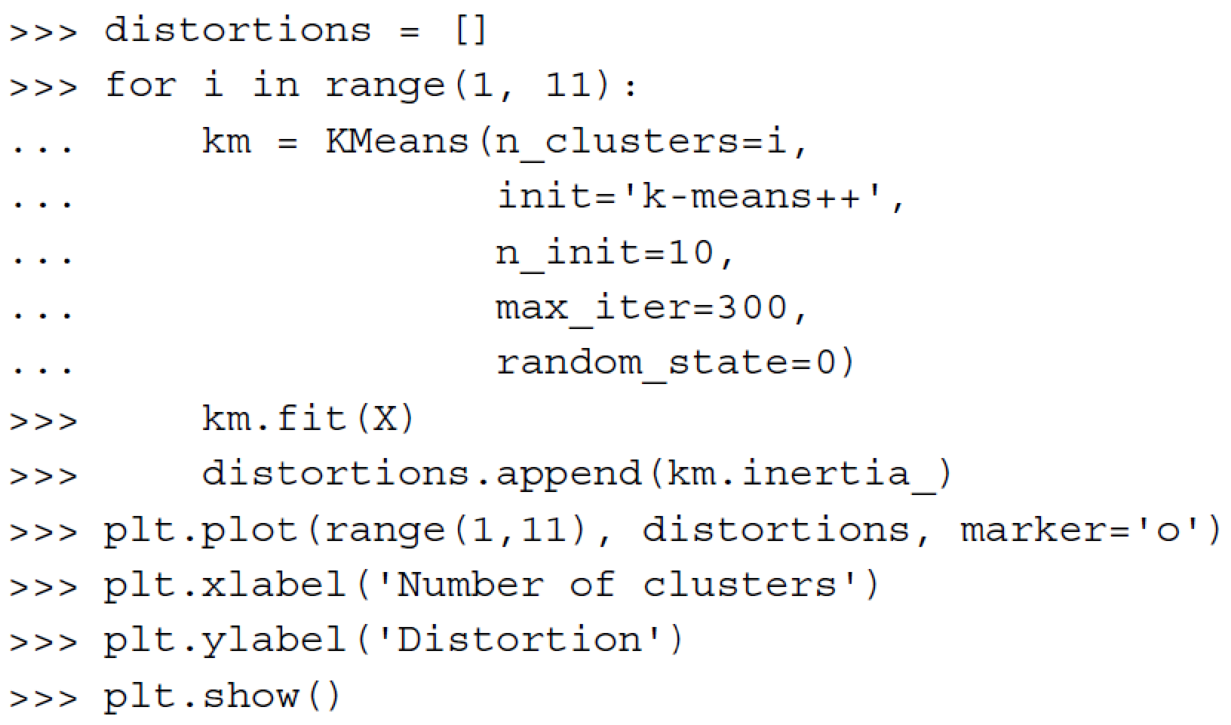
Finding the number of clusters: Use **Elbow Method**

Calculate the **Euclidian distance**

- One could plot the Distortion against the number of clusters *k*. Intuitively, if *k* increases, distortion should decrease. This is because the samples will be close to their assigned centroids.
- This plot is called the Elbow method. It indicates the optimum number of clusters at the position of the elbow, the point where distortion begins to increase most rapidly.



```
>>> print('Distortion: %.2f' % km.inertia_)
Distortion: 72.48
```



```
>>> distortions = []
>>> for i in range(1, 11):
...     km = KMeans(n_clusters=i,
...                 init='k-means++',
...                 n_init=10,
...                 max_iter=300,
...                 random_state=0)
>>>     km.fit(X)
>>>     distortions.append(km.inertia_)
>>> plt.plot(range(1,11), distortions, marker='o')
>>> plt.xlabel('Number of clusters')
>>> plt.ylabel('Distortion')
>>> plt.show()
```

Source Credit : "Python Machine Learning" by Sebastian Raschka

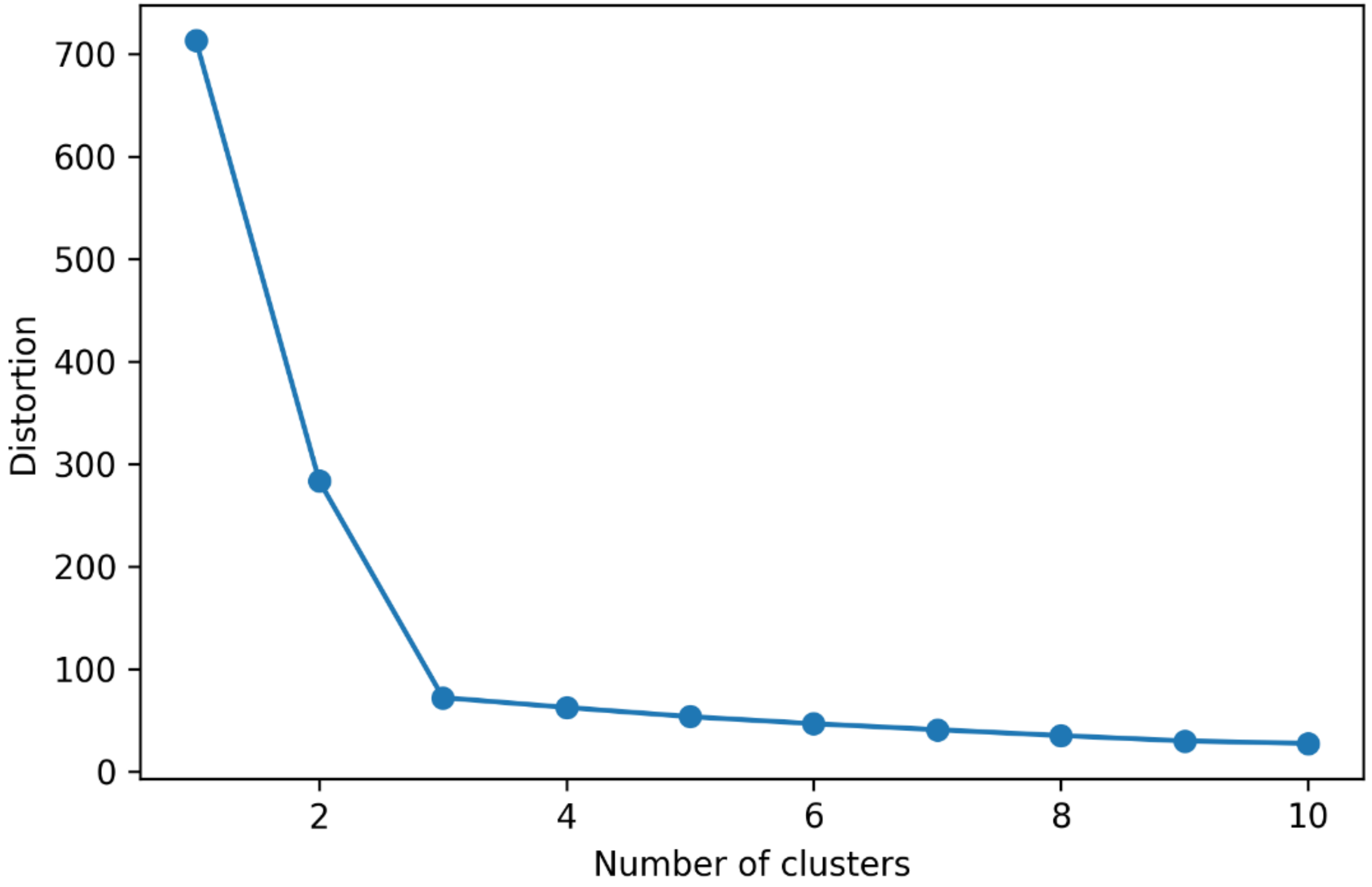
K-means Clustering: Example

MATHEMATICAL REPRESENTATION

The adjoining Elbow method suggests that k = 3 is the most optimum number of clusters.

Finding the number of clusters: Use **Elbow Method**

Calculate the **Euclidian distance**



Source Credit : "Python Machine Learning" by Sebastian Raschka

K-means Clustering: Example

MATHEMATICAL REPRESENTATION

Finding the number of clusters: Use Elbow Method

Calculate the **Euclidean distance**

- K-means is based on finding points close to cluster centroids. The distance between two points *x* and *y* can be measured by the squared Euclidean distance between them in an m-dimensional space.
  - Here, *j* refers to *j*-th dimension (or *j*-th feature) of the data point.

$$d(\boldsymbol{x}, \boldsymbol{y})^2 = \sum_{j=1}^{m} \left(x_j - y_j\right)^2 = \left\| \boldsymbol{x} - \boldsymbol{y} \right\|_2^2$$

Source Credit : "Python Machine Learning" by Sebastian Raschka

K-means Clustering: Example

MATHEMATICAL REPRESENTATION

Finding the number of clusters: Use Elbow Method

Calculate the **Euclidean distance**

- Based on this, the optimization problem is to minimize the within-cluster sum of squared errors (SSE), which is sometimes also called the **cluster inertia**.
  - Here, *j* refers to *j*-th cluster. $\mu^{(j)}$ is the centroid of that cluster.
  - $w^{(i,j)} = 1$ if the sample $x^{(i)}$ is in cluster *j*, and 0 otherwise.

$$SSE = \sum_{i=1}^{n} \sum_{j=1}^{k} w^{(i,j)} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}^{(j)} \right\|_2^2$$

Source Credit : "Python Machine Learning" by Sebastian Raschka

K-means Clustering: Example

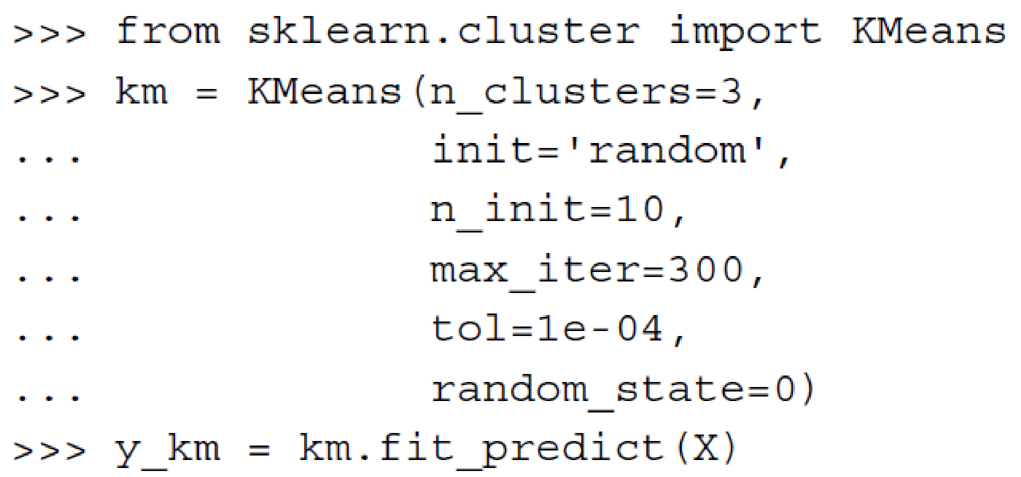
MATHEMATICAL REPRESENTATION

Finding the number of clusters: Use Elbow Method

Calculate the **Euclidean distance**

Scikit-learn cluster module has the K-means function. In the code shown,

- k = 3
- n_init = 10 : which means that you run the clustering logic 10 times, each time with random cluster centroids. Finally, the model with the lowest SSE among the 10 schemes gets chosen.
- max_iter = 300 : which means within each of the 10 runs, iterate 300 times to find ideal clusters.
- If convergence happens before 300 iterations, it will stop early.
- A large max_iter is computationally intensive (if convergence does not happen early).
- tol is another parameter which governs tolerance with regard to changes in the within-cluster SSE to declare convergence. A larger tol means it will declare convergence sooner.



```
>>> from sklearn.cluster import KMeans
>>> km = KMeans(n_clusters=3,
...             init='random',
...             n_init=10,
...             max_iter=300,
...             tol=1e-04,
...             random_state=0)
>>> y_km = km.fit_predict(X)
```

Source Credit : "Python Machine Learning" by Sebastian Raschka

Other Examples of K-means Clustering

- Grouping articles (example: Google news)
- Grouping customers who share similar interests, example: analyzing customers who like contemporary fashion vs. those who prefer traditional clothing
- Classifying high risk and low risk patients from a patient pool
- Segregating criminals from normal crowd in a security control process

Demo

Clustering

Perform Clustering algorithm and the Elbow method on a random dataset. You can perform this in the lab environment with the dataset available in the LMS.

Key Takeaways

✓ The most common form of Unsupervised Learning is Clustering, which involves segregating data based on similarity between data instances.

✓ K-means is a popular technique for Clustering. It involves an iterative process to find cluster centers called centroids and assigning data points to one of the centroids.

✓ K-means finds clusters by minimizing the within-cluster distance of data points from respective centroids.

✓ The Elbow method is used to determine the most optimum number of clusters.

Quiz

QUIZ 1

K-Means Clustering is a type of _________________ learning.

a. Supervised

b. Unsupervised

c. Reinforcement

d. Semi-supervised

QUIZ 1

K-Means Clustering is a type of ________________ learning.

a. Supervised

b. Unsupervised

c. Reinforcement

d. Semi-supervised

The correct answer is **b. Unsupervised Learning**

K-Means Clustering is a type of unsupervised learning.

QUIZ

2

In the Elbow method, the ideal number of clusters is found at:

a. the point where distortions begin to increase most rapidly

b. the point where the distortions begin to decrease most rapidly

c. the point where the number of clusters is maximum

d. the point where the number of clusters is minimum

QUIZ 2

In the Elbow method, the ideal number of clusters is found at:

a. the point where distortions begin to increase most rapidly

b. the point where the distortions begin to decrease most rapidly

c. the point where the number of clusters is maximum

d. the point where the number of clusters is minimum

The correct answer is **a. The point where distortions begin to increase most rapidly**

In the Elbow method, the ideal number of clusters is found at the point where distortions begin to increase most rapidly.

Hands-on Assignments

Demo File	Assignment	What it demonstrates?
ClusteringKmeans.py	Modify the number of clusters k to 2 and note the observations.	Demonstrate Clustering algorithm and the Elbow method on a random dataset.
	Modify the n_samples from 150 to 15000 and number of centers to 4 with n_clusters as 3. Check the output and note your observations.	Modify the code and check the output.
	Modify the code to change the n_samples from 150 to 15000 and number of centers to 4, keeping n_clusters at 4. Check the output.	Modify the code and check the output.
	Modify the number of clusters k to 6 and note the observations.	Modify the code and check the output.

This concludes “Unsupervised Learning with Clustering.”

The next lesson is “Introduction to Deep Learning.”