Machine Learning to build Intelligent Systems

Manas Dasgupta

Unsupervised Learning with K-Means Clustering

Structure of this Module

K-Means Clustering

TOPICS
Unsupervised Learning
Introduction to K-Means
K-Means Clustering Process Steps
Centroid Optimization / Convergence
Other Considerations
Optimizing Number of Clusters

Unsupervised Learning

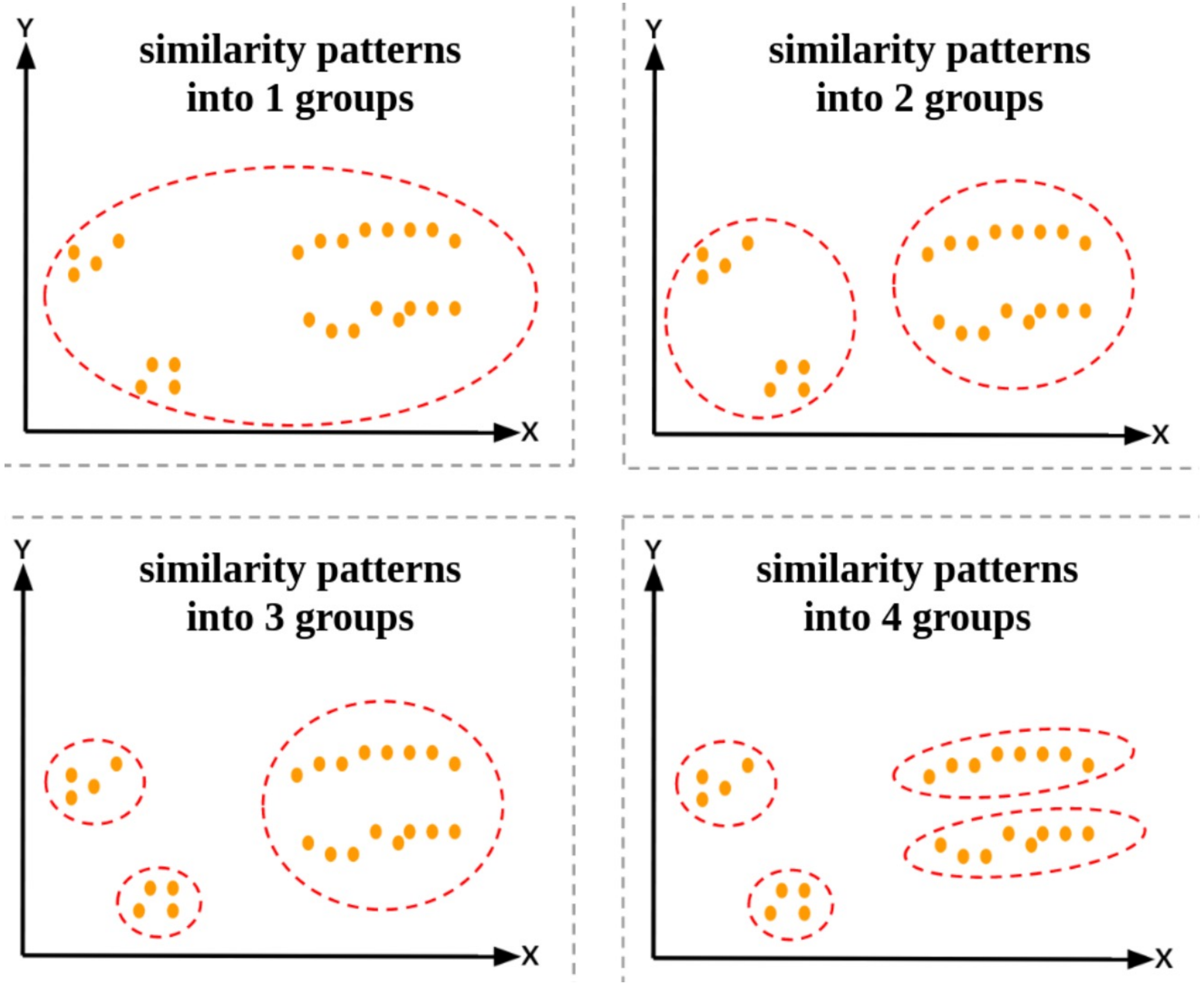
Unsupervised Learning are those Machine Learning techniques in which, ***there are no labels for the training data***. A machine learning algorithm tries to learn the underlying patterns or distributions that govern the data.

Some cool use cases of Unsupervised Learning

Use Case	Why Unsupervised
Customer Segmentation	A large ecommerce company wants to run specific / targeted promo campaigns for different groups of customers based on their spend patterns. However, from millions of transactions, the company is not able to find the patterns easily.
Text Clustering	A Law Consulting firm deals with tens of thousands of documents in its digital repository accumulated over the years. However, there is no classification or grouping of such documents and it wants to group them into clusters so that search and retrieval for references are easy.
Logistics Optimization	A Delivery Logistics company wants to analyze the daily fleet movement data on goods and locations and find patterns of clusters. It wants to optimize the fleet size and movement routes based on clusters hidden in the data.

K-Means Clustering

K-Means Clustering is an Unsupervised Learning Algorithm, which groups the unlabelled dataset into 'K' different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.



K-Means Clustering

- The objective of K-means is to group similar data points together and discover underlying patterns. To achieve this objective, K-means looks for a fixed number (*k*) of clusters in a dataset.

- A ***cluster*** refers to a collection of data points aggregated together because of certain similarities.

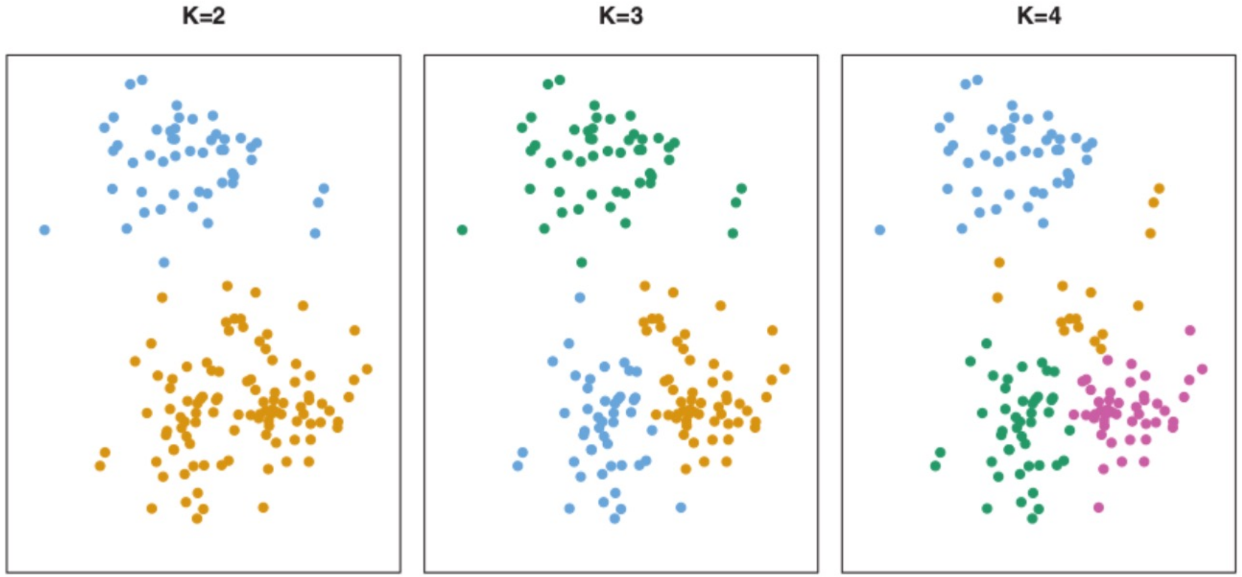
- We'll define a target number '*k*', which refers to the number of ***centroids*** you need in the dataset. A centroid is the location (data point) representing the center of the cluster.

- In other words, the K-means algorithm ***identifies k number of centroids***, and then ***allocates every data point to the nearest cluster.***

- The '***means***' in the K-means refers to averaging of the data, that is, finding the centroid.

K-Means Clustering – What is 'K'

K-means clustering is a simple and elegant approach for partitioning a data set into K distinct, non-overlapping clusters. To perform K-means clustering, we must first specify the desired number of clusters K; then the K-means algorithm will assign each observation to exactly one of the K clusters.



K-Means Clustering – Basis and Steps

The K-means algorithm clusters data by trying to separate samples in 'n' groups of equal variance, ***minimizing a criterion known as the inertia or within-cluster sum-of-squares***. The K-means algorithm aims to choose centroid that minimise the **inertia**, or **within-cluster sum-of-squares criterion.**

$$\sum_{i=0}^{n} \min_{\mu_j \in C}(||x_i - \mu_j||^2)$$

1. **Determine the value "k", the value "k" represents the number of clusters.**
2. **Randomly select 'k' distinct centroid (new data points as cluster initialization)**
3. **Measure the distance (euclidean distance) between each point and the centroid**
4. **Assign the each point to the nearest cluster**
5. **Calculate the mean of each cluster as new centroid**
6. **Repeat step 3–5 with the new center of cluster till Sum of Square Distances between all Data Points stabilize**

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

p, q = two points in Euclidean n-space

q_i, p_i = Euclidean vectors, starting from the origin of the space (initial point)

n = n-space

p	**(3,4)**
q	**(7,8)**
d(p,q)	**sqrt((7-3)^2 + (8-4)^2)**
	5.66

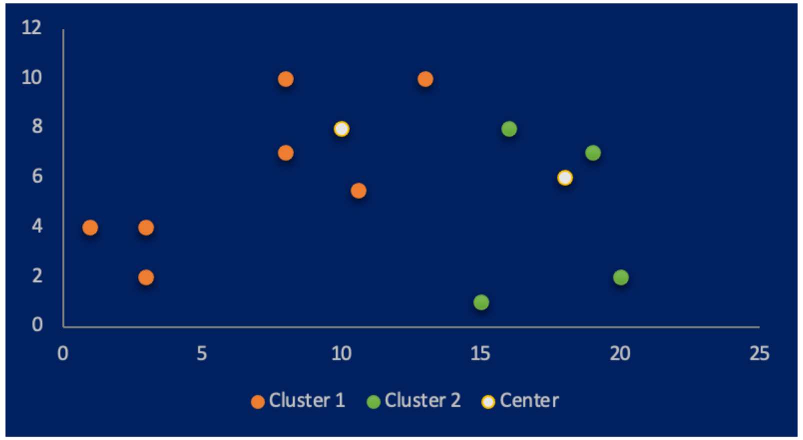
K-Means Clustering – Calculations-1

Iteration 1

X	Y	Cluster
8	10	1
20	2	2
16	8	2
8	7	1
1	4	1
13	10	1
15	1	2
19	7	2
3	4	1
3	2	1
11	6	1

Center	X	Y
1	10.0	8.0
2	18.0	6.0

SSE	343.6



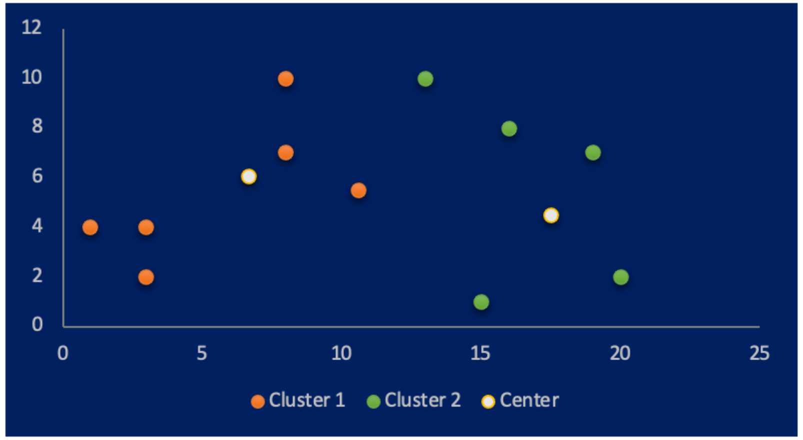
Min Squared Distance	Dist_C1	Dist_C2
8.0	2.8	10.8
20.0	11.7	4.5
8.0	6.0	2.8
5.0	2.2	10.0
97.0	9.8	17.1
13.0	3.6	6.4
34.0	8.6	5.8
2.0	9.1	1.4
65.0	8.1	15.1
85.0	9.2	15.5
6.6	2.6	7.4

Iteration 2

X	Y	Cluster
8	10	1
20	2	2
16	8	2
8	7	1
1	4	1
13	10	2
15	1	2
19	7	2
3	4	1
3	2	1
11	6	1

Center	X	Y
1	6.7	6.1
2	17.5	4.5

SSE	224.2



Min Squared Distance	Dist_C1	Dist_C2
17.2	4.2	11.0
12.5	14.0	3.5
14.5	9.5	3.8
2.7	1.6	9.8
36.3	6.0	16.5
50.5	7.5	7.1
18.5	9.8	4.3
8.5	12.4	2.9
17.7	4.2	14.5
30.0	5.5	14.7
15.9	4.0	7.0

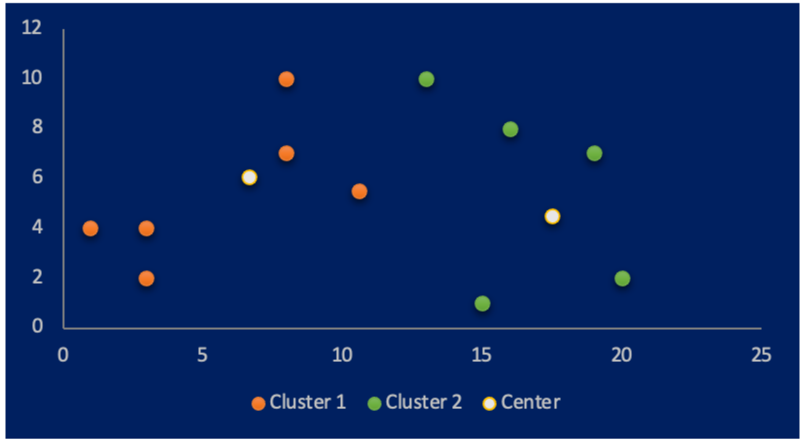
K-Means Clustering – Calculations-2

Iteration 3

X	Y	Cluster
8	10	1
20	2	2
16	8	2
8	7	1
1	4	1
13	10	2
15	1	2
19	7	2
3	4	1
3	2	1
11	6	1

Center	X	Y
1	5.6	5.4
2	16.6	5.6

SSE	204.8



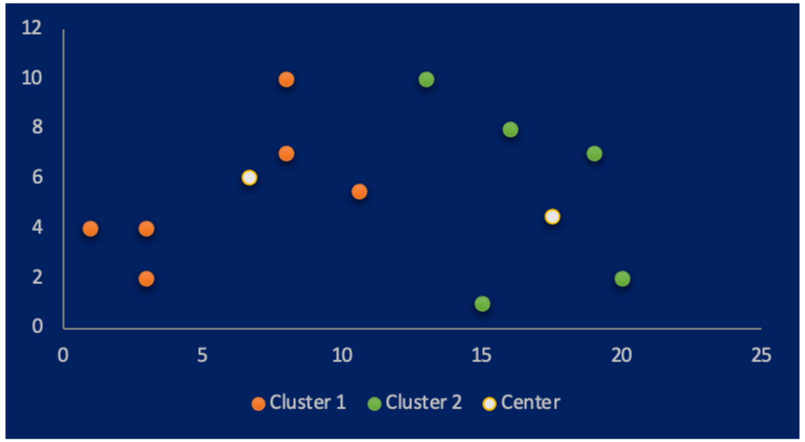
Min Squared Distance	Dist_C1	Dist_C2
26.8	5.2	9.7
24.5	14.8	5.0
6.1	10.7	2.5
8.3	2.9	8.7
23.2	4.8	15.7
32.3	8.7	5.7
23.7	10.4	4.9
7.7	13.5	2.8
8.8	3.0	13.7
18.4	4.3	14.1
25.0	5.0	6.0

Iteration 4

X	Y	Cluster
8	10	1
20	2	2
16	8	2
8	7	1
1	4	1
13	10	2
15	1	2
19	7	2
3	4	1
3	2	1
11	6	1

Center	X	Y
1	5.6	5.4
2	16.6	5.6

SSE	204.8



Min Squared Distance	Dist_C1	Dist_C2
26.8	5.2	9.7
24.5	14.8	5.0
6.1	10.7	2.5
8.3	2.9	8.7
23.2	4.8	15.7
32.3	8.7	5.7
23.7	10.4	4.9
7.7	13.5	2.8
8.8	3.0	13.7
18.4	4.3	14.1
25.0	5.0	6.0

K-Means Clustering - Points

- The number of clusters that you want to divide your data points into, i.e. **the value of K has to be pre-determined.**
- The K-means algorithm is **non-deterministic**. This means that the final outcome of clustering can be different each time the algorithm is run even on the same data set. This is because, as you saw, the final cluster that you get can vary by the choice of the initial cluster centres.
- **Outliers have an impact on the clusters** and hence it is suggested to bring attributes into the same scale using **Standardisation**.
- Since the distance metric used in the clustering process is the Euclidean distance, you need to bring all your attributes on the same scale. This can be achieved through **Feature Standardisation**.

Optimizing ‘K’

Till now, we have made an assumption that the number of clusters, i.e., the value of ‘k’ to be predetermined by the user. In this section, we will see how we can find the optimum ‘k’. We will discuss the following two methods.

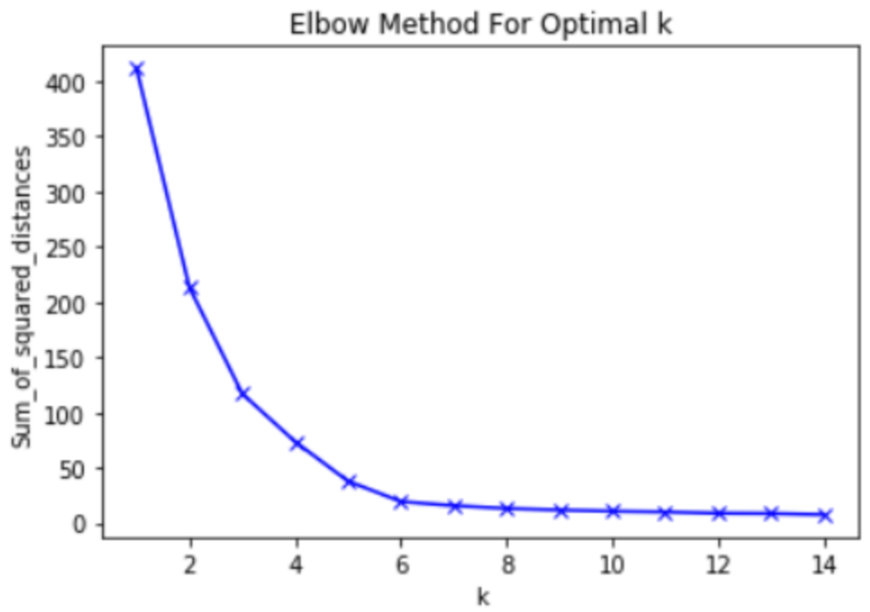
The Elbow Method

The Silhouette Method

The Elbow Method

Calculate the **Within-Cluster-Sum of Squared** Errors (WSS) for **different values of k**, and choose the k for which WSS starts to move towards the lowest. In the plot of WSS-versus-k, this is visible as an **elbow.**

In the plot here, the elbow is at k=6 indicating the optimal k for this dataset is 6.



The Silhouette Method

The Silhouette value measures **how similar a point is to its own cluster compared to other clusters.**

The range of the Silhouette value is between +1 and -1. A high value is desirable and indicates that the point is placed in the correct cluster. If many points have a negative Silhouette value, it may indicate that we have created too many or too few clusters.

It is apparent that the Silhouette Coefficient is calculated for all data points (observations).

Computing Silhouette Coefficient:

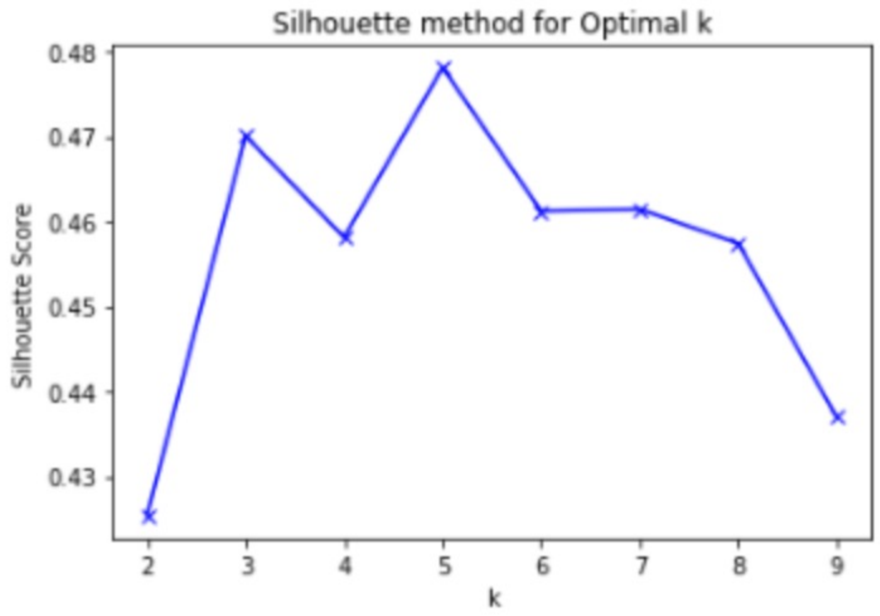
1. Compute a(i): The average distance of that point with all other points in the same cluster.
2. Compute b(i): The average distance of that point with all the points in the closest cluster to its cluster.
3. Compute s(i) — silhouette coefficient of ‘i’th observation using below mentioned formula.

After computing the silhouette coefficient for each point, average it out to get the ***silhouette score***.

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

The Silhouette Method

- As per this method k=3 was a local optima, whereas k=5 should be chosen for the number of clusters.
- This method is better as it makes the decision regarding the optimal number of clusters more meaningful and clear.
- But this metric is computation expensive as the coefficient is calculated for every instance.
- Therefore, decision regarding the optimal metric to be chosen for the number of cluster decision is to be made according to the needs of the product.



Gradient Descent

Python Demo

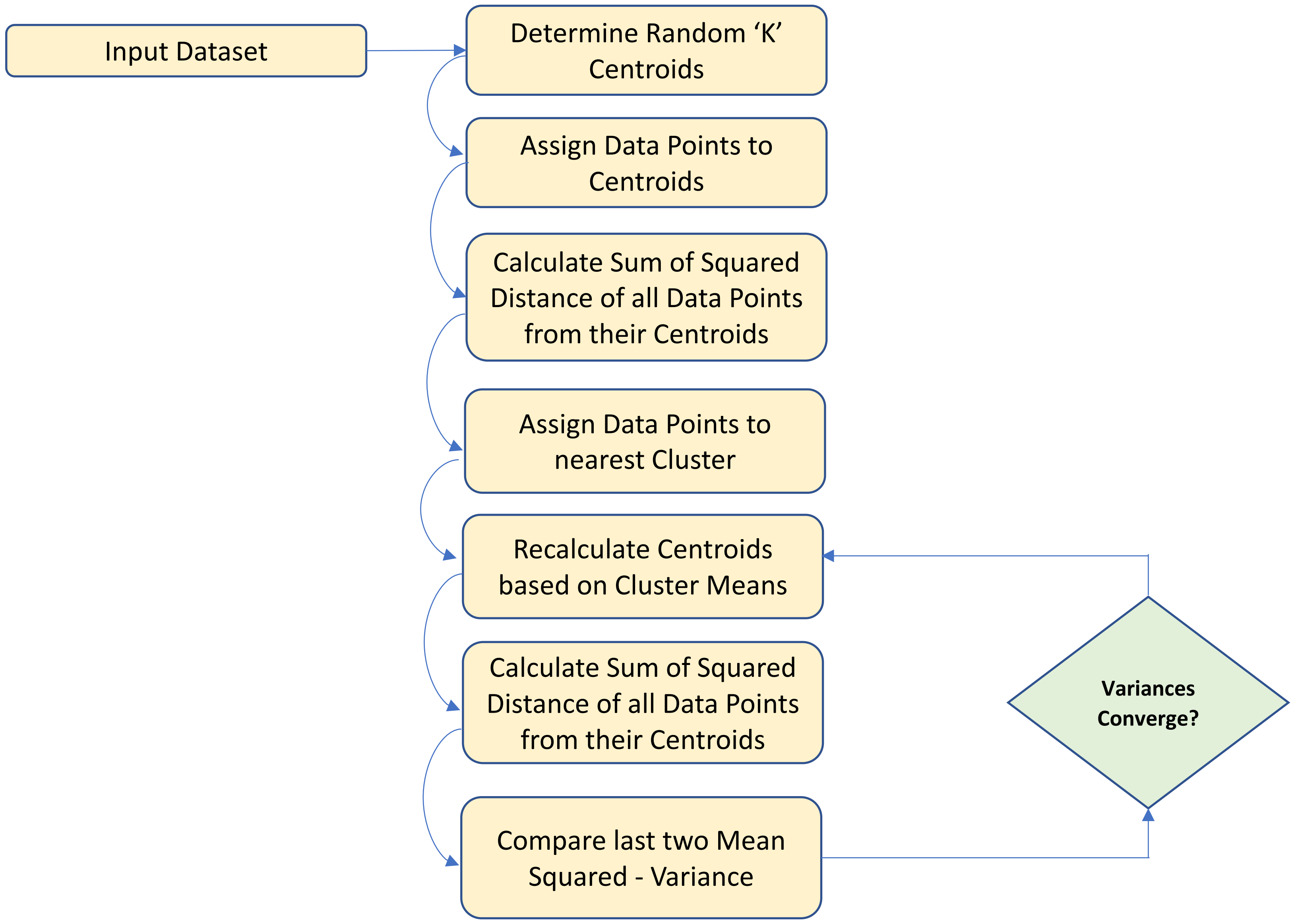
- **Customer Segmentation using K-Means**

Hope you have liked this Video.
Please help us by providing your Ratings and Comments for this Course!

Thank You!!
Manas Dasgupta

Happy Learning!!

K-Means Clustering – Process Steps



Fully Editable Icon Sets: A

You can Resize without losing quality

You can Change Fill Color & Line Color

FREE
PPT TEMPLATES

www.allppt.com

Fully Editable Icon Sets: B

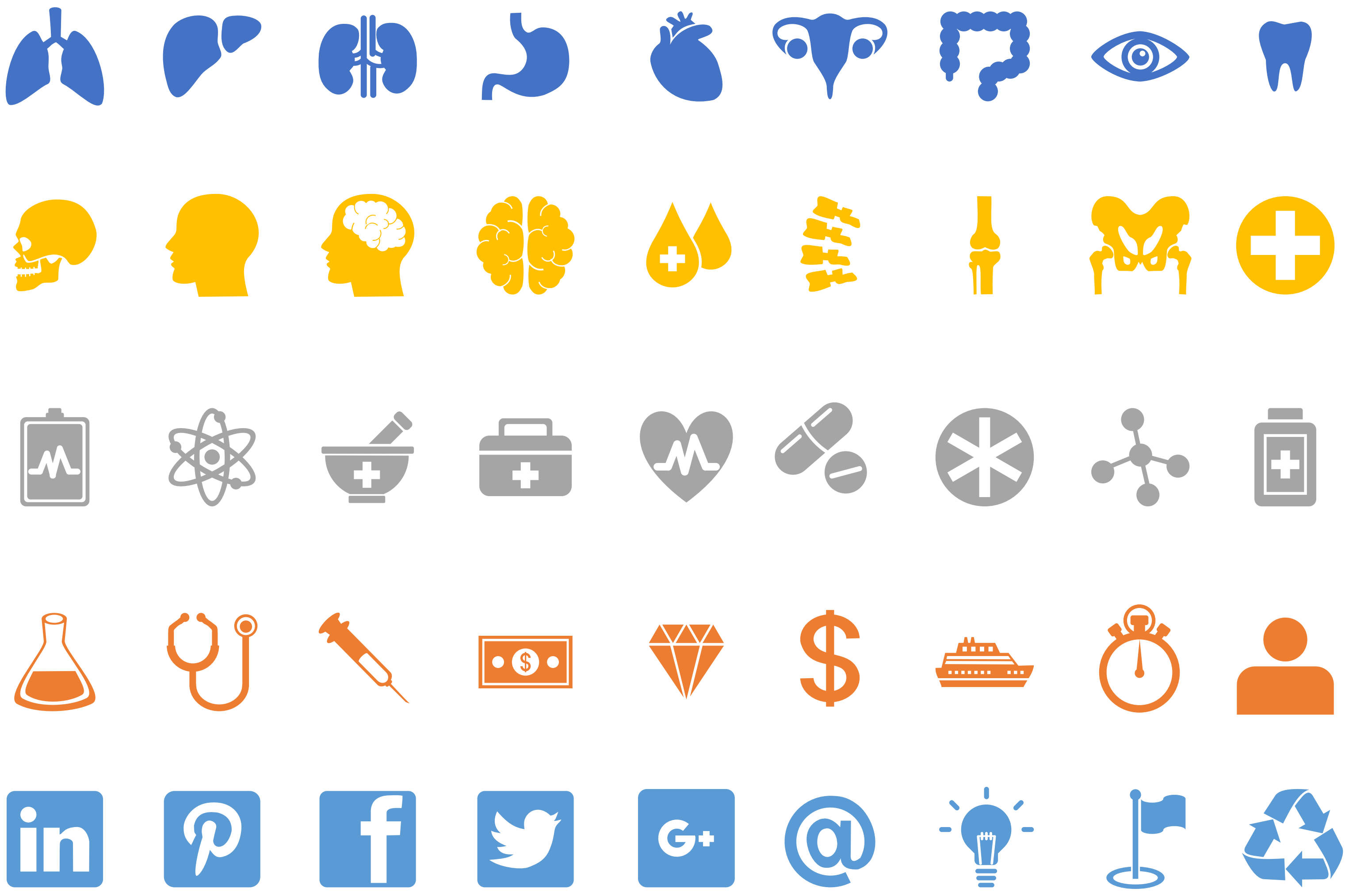
You can Resize without losing quality

You can Change Fill Color & Line Color

FREE

PPT TEMPLATES

www.allppt.com



Fully Editable Icon Sets: C

You can Resize without losing quality

You can Change Fill Color & Line Color

FREE

PPT TEMPLATES