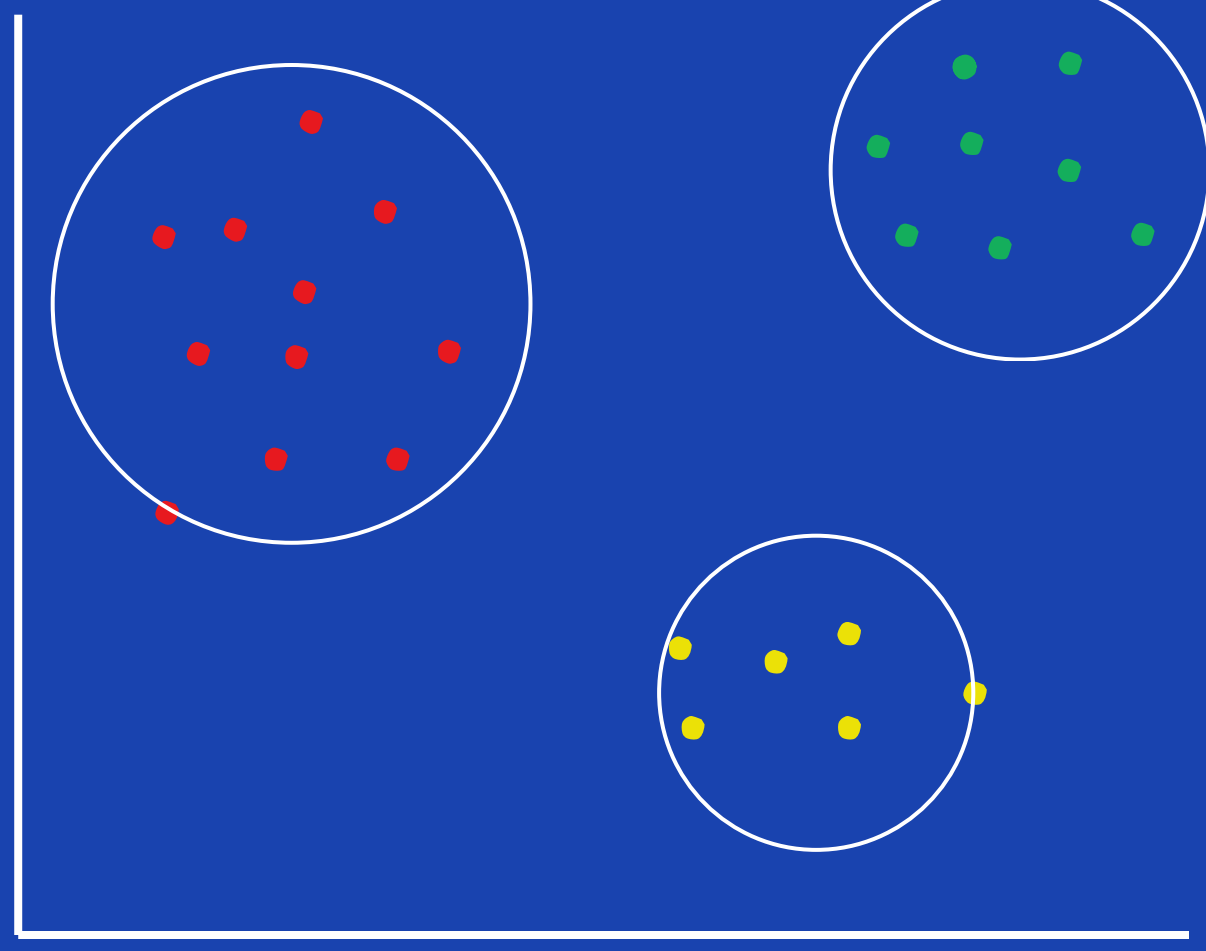
Tech Bytes

ML Refresher - K-Means

by Pranali Bose

What is it?



An unsupervised machine learning algorithm that partitions a dataset into K distinct clusters

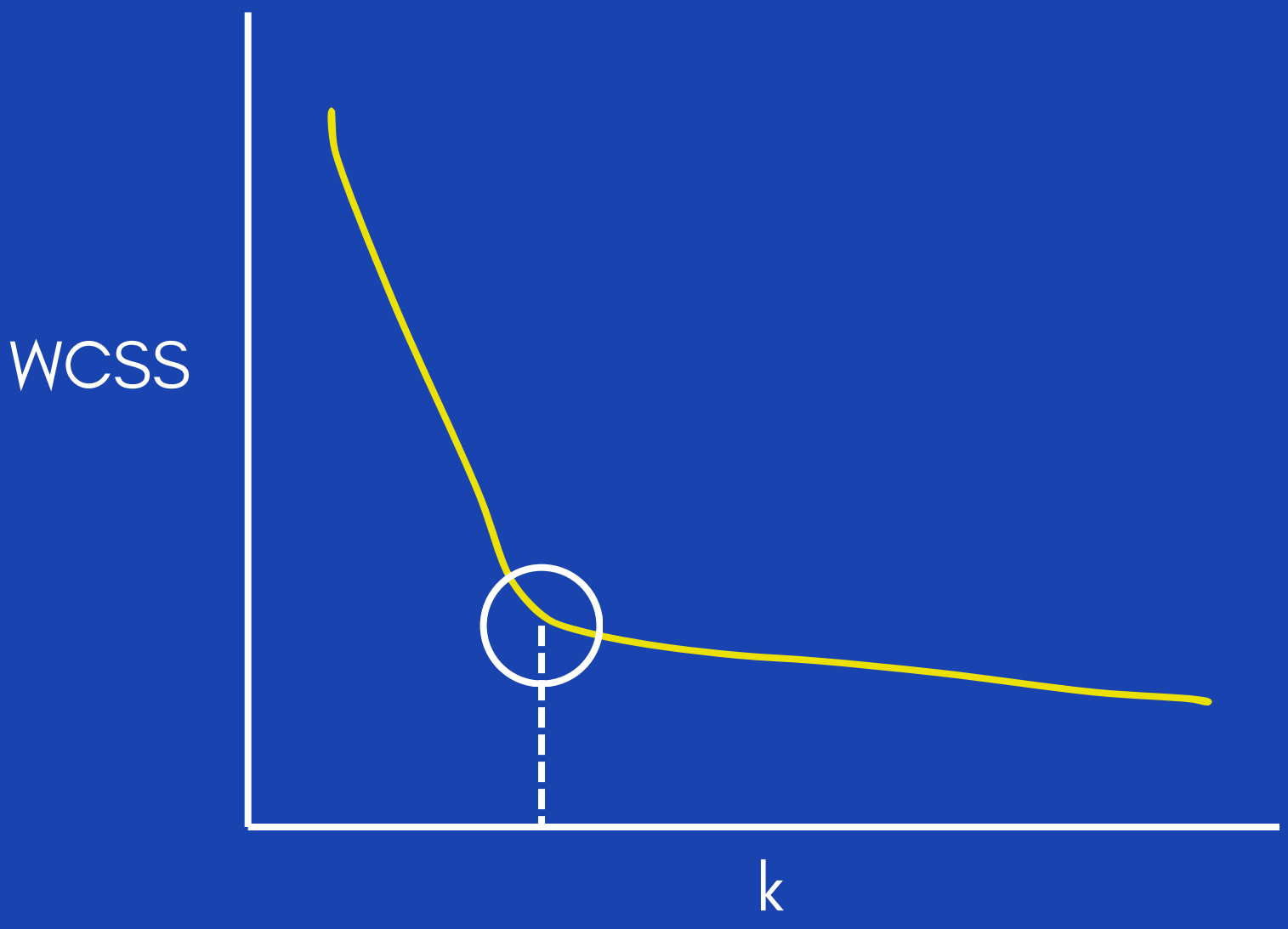
Key Concepts

- **Clusters**: The data gets partitioned into K groups called clusters.
- **Centroid**: The average position of the data points within a cluster
- **Distance Metric** – Typically uses Euclidean distance to measure how close data points are to centroids.

How does it work?

- **Step 1**: Choose the Number of Clusters (K)
- **Step 2**: Initialize K Centroids Randomly
- **Step 3**: Assign Data Points to the Nearest Centroid
- **Step 4**: Update Centroids
- **Step 5**: Repeat Steps 3 & 4 Until Convergence

Choosing k



Elbow Method

- Compute WCSS for each K.
- Plot WCSS vs. K.
- Look for the "elbow point" (where WCSS stops decreasing sharply).

What is WCSS?

- **W**ithin-**c**luster **S**um of **S**quares measures how well data points are clustered around their centroids.
- It is the sum of squared distances between each data point and its assigned centroid:

$$\sum_{i=1}^{k} \sum_{x \in Ci} (x - \mu i)^2$$

where:

- K = number of clusters
- Ci = points in cluster i
- μi = centroid of cluster i
- x = data points in cluster i

Ponder Upon!

- Why does K-Means use Euclidean distance and not Manhattan or Cosine distance? What would change if it did?
- How would you modify K-Means to handle outliers better?
- What happens if you set K to a very large number — say, equal to the number of data points?
- Can you think of a real-world scenario where K-Means would fail to cluster properly? Why?

Tech Bytes

Find this useful?

Let me know in the comments which topic would you like to see next

Follow for more...