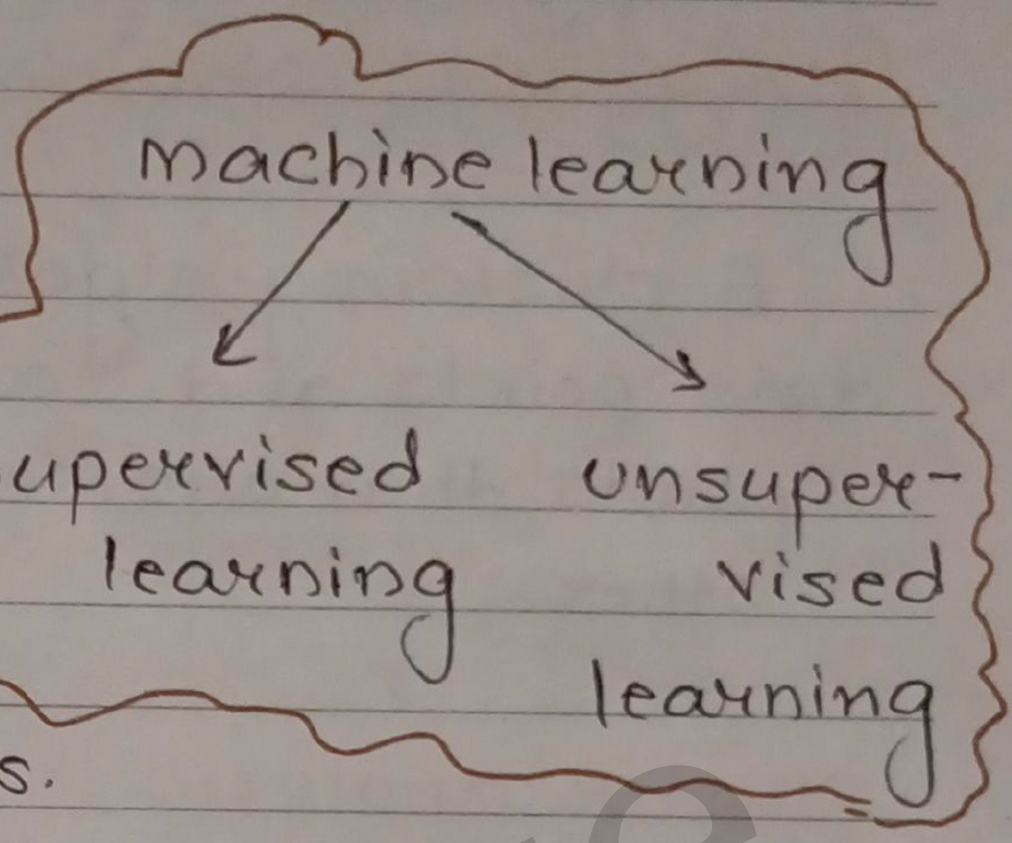
Clustering



Unsupervised machine learning:

unsupervised ML uses ML algorithm to analyze and cluster unlabelled datasets.
These algorithms discovers hidden patterns as data groups without the need of human interaction.

Its ability to discover similarities and differences in information make it the ideal solution for EDA, cross-selling strategies, customer segmentation, and image recognition.

Unsupervised learning models are used for three main tasks

→ clustering
→ association
→ dimensionality reduction

Komal Diwe

What is clustering?

A clustering algorithm looks at a number of data points and automatically finds data points that are related or similar to each other.

cluster analysis is a technique used in data mining and ML to group similar objects into clusters.

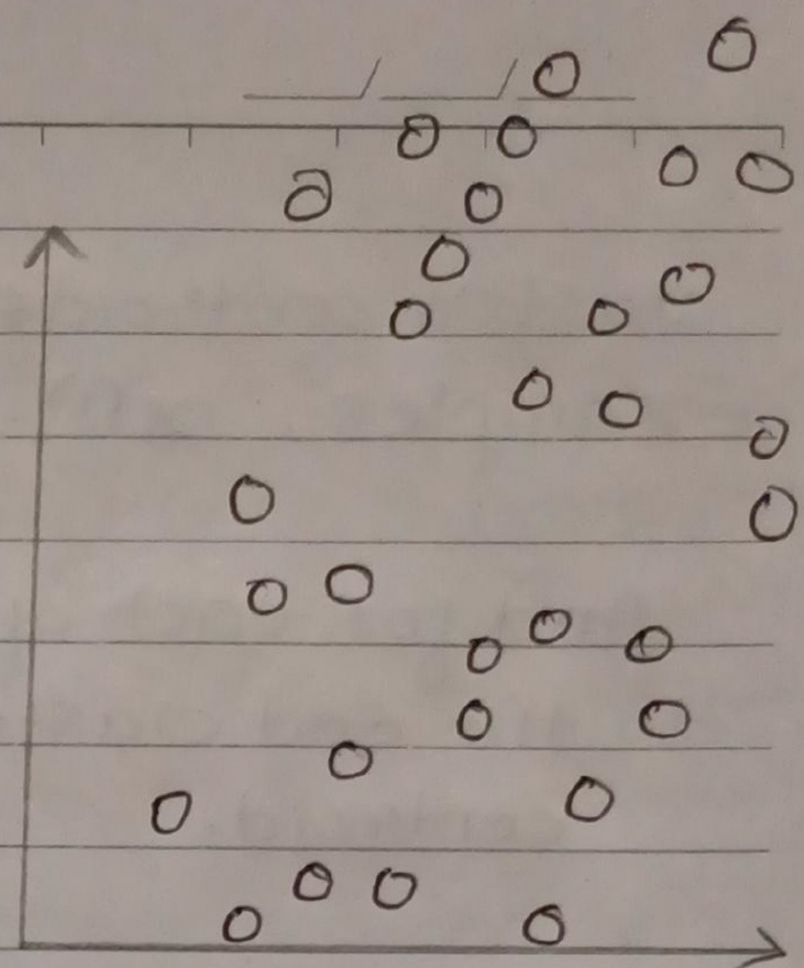
k-means clustering ⟶ the aim is to partition a set of objects into k-clusters in such a way that the sum of the squared distances between the objects ~~into k-clusters in such a way~~ and their assigned cluster mean is minimized.

" Clustering is the process of dividing the entire data into groups (also known as clusters) based on the patterns in the data.

Komal Diwe

K-means Intuition:

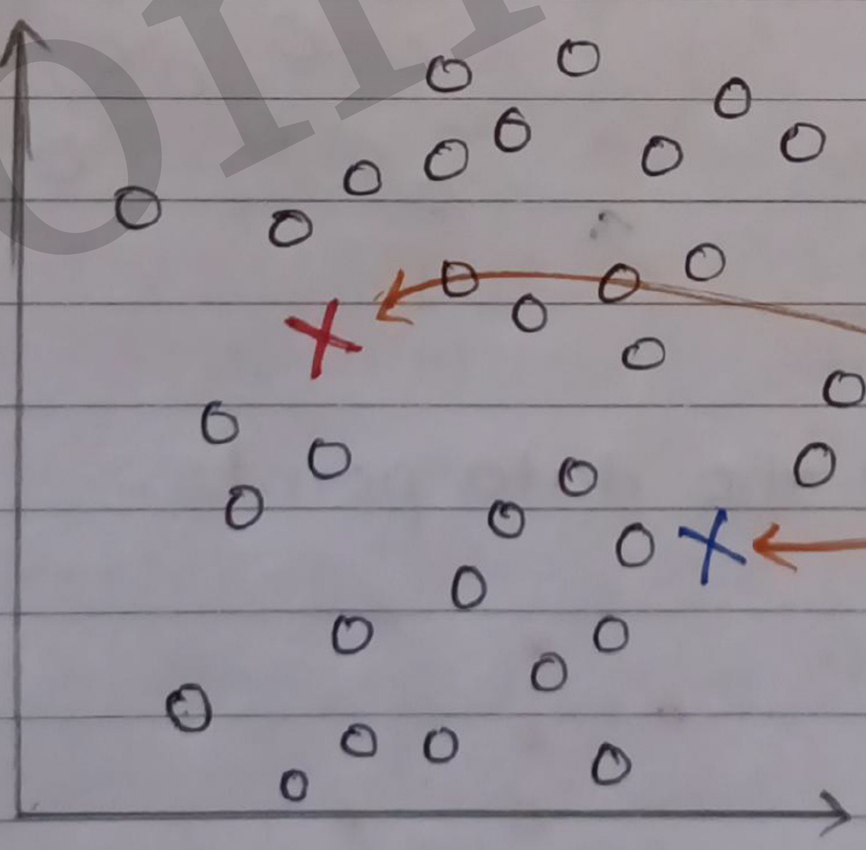
plotted a data set with 30 unlabeled training examples →



K-means will repeatedly do two different things.

1. assign points to cluster centeroids
2. move cluster centroids

① Now, randomly pick two points, at where might be the centers of two different clusters.



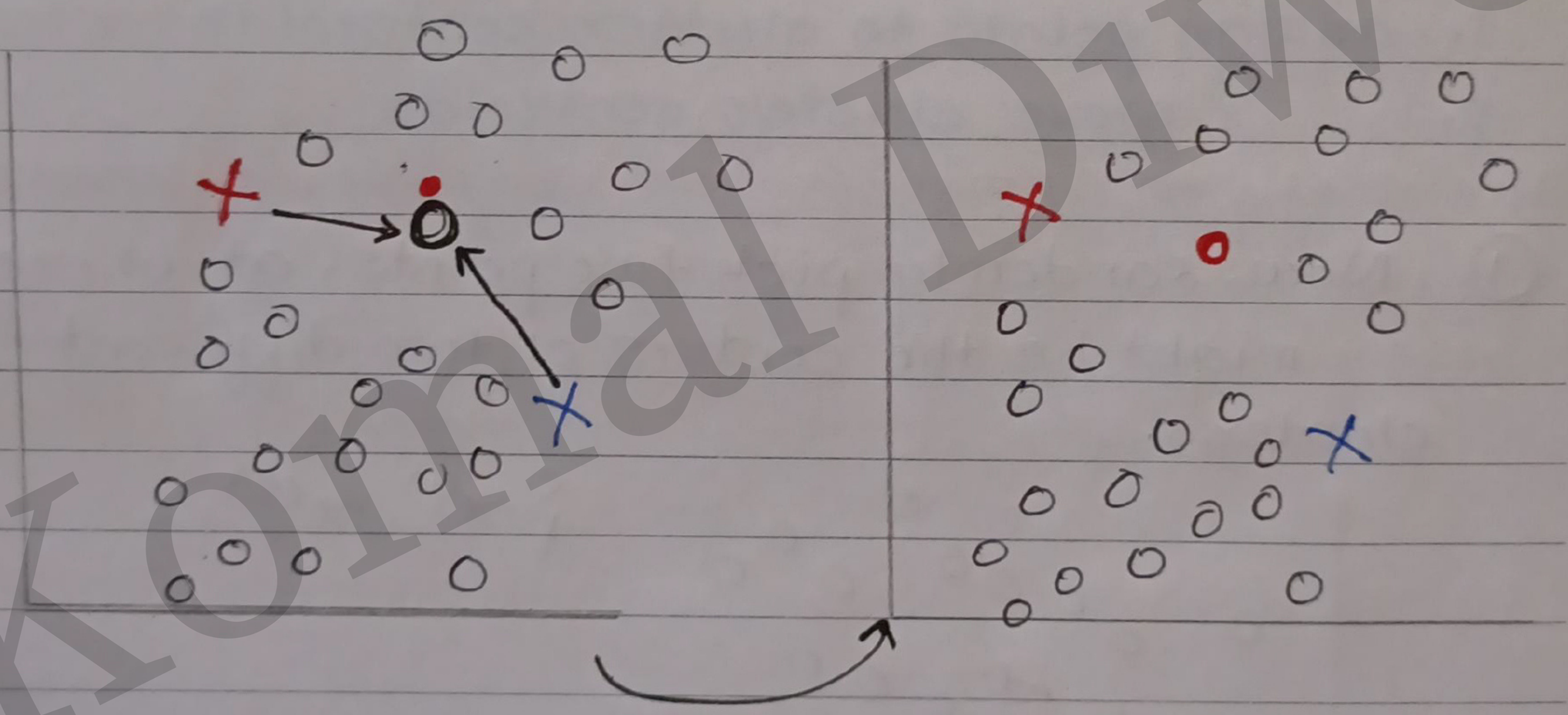
* this is just a random guess.

· After it's made an initial guess at where the

cluster centroids is, it will go through all of these examples, $x^{(1)}$ .... $x^{(30)}$ datapoints.

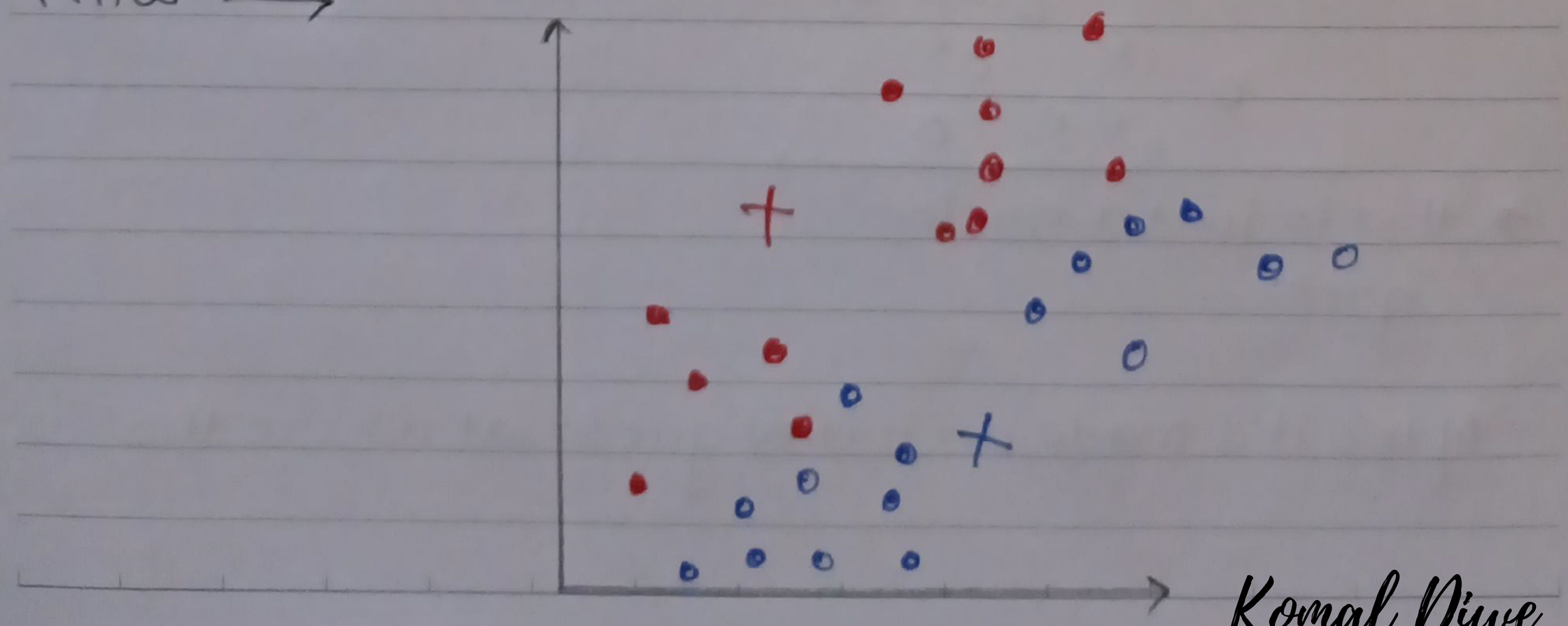
And for each of them it will check if it is closer to the red cluster centroid or blue cluster centroid.

Step 01: assign each point to its closest centroid.



Thus, it will do with all the data points.

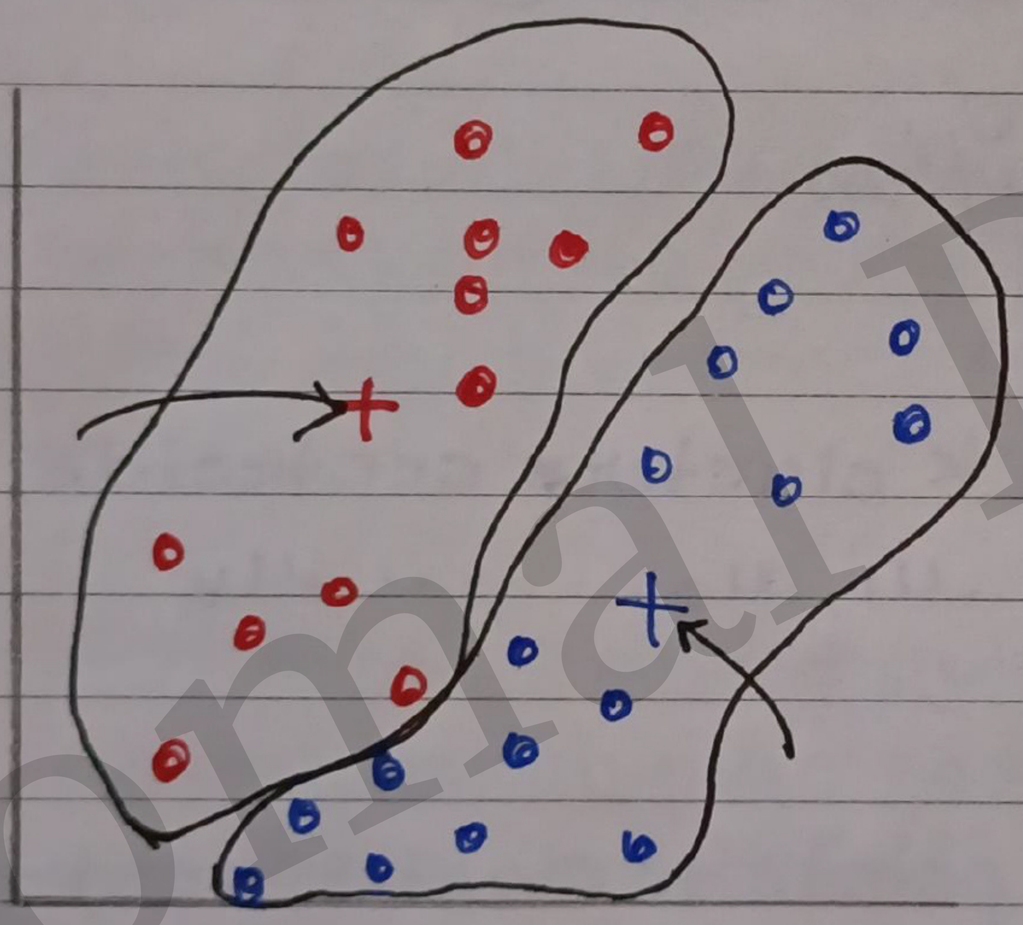
Final ⟶



Step 02: Recompute the centroids

It'll look at all the red points and take an average of them. and it will move the red cross to whatever is the average location of the red dots.

same thing for blue points./ dots.



Slightly improved guesses for the locations of the cluster centroids.

Now keep repeating the above two steps till there is no change in the cluster centroids location.

If we keep on doing those two steps, we find that there are no more changes to the colors of the points or to the locations of the clusters centroids.

And so this means that at this point the K-means clustering algorithm has converged.

Because applying those two steps over and over, results in no further changes to either the assignment to the point to the centroids or the location of the cluster centroids.

K-means Algorithm

Randomly initialize K cluster centroids

$\mu_1, \mu_2, \cdots\cdots, \mu_K$

Repeat {

assign points to cluster centroids

for i = 1 to m

$c^{(i)}$:= index (from 1 to k) of cluster centroid closest to $x^{(i)}$

$\min_k \|x^{(i)} - \mu_k\|$

move cluster centroids

for k = 1 to k

μ_k := average (mean) of points assigned to cluster k

}

1. Randomly initialize k cluster centroids, $\mu_1, \mu_2, \ldots \mu_k$, where k is the number of clusters.

2. Assign points to clusters by computing the distance between each point and each centroid and assigning the point to the closest centroid.

3. Update the cluster centroids by computing the mean of all the points assigned to each cluster, which becomes the new centroid location.

4. Repeat steps 2 and 3 until the cluster assignments no longer change. or a maximum number of iterations is reached.

Komal Diwe

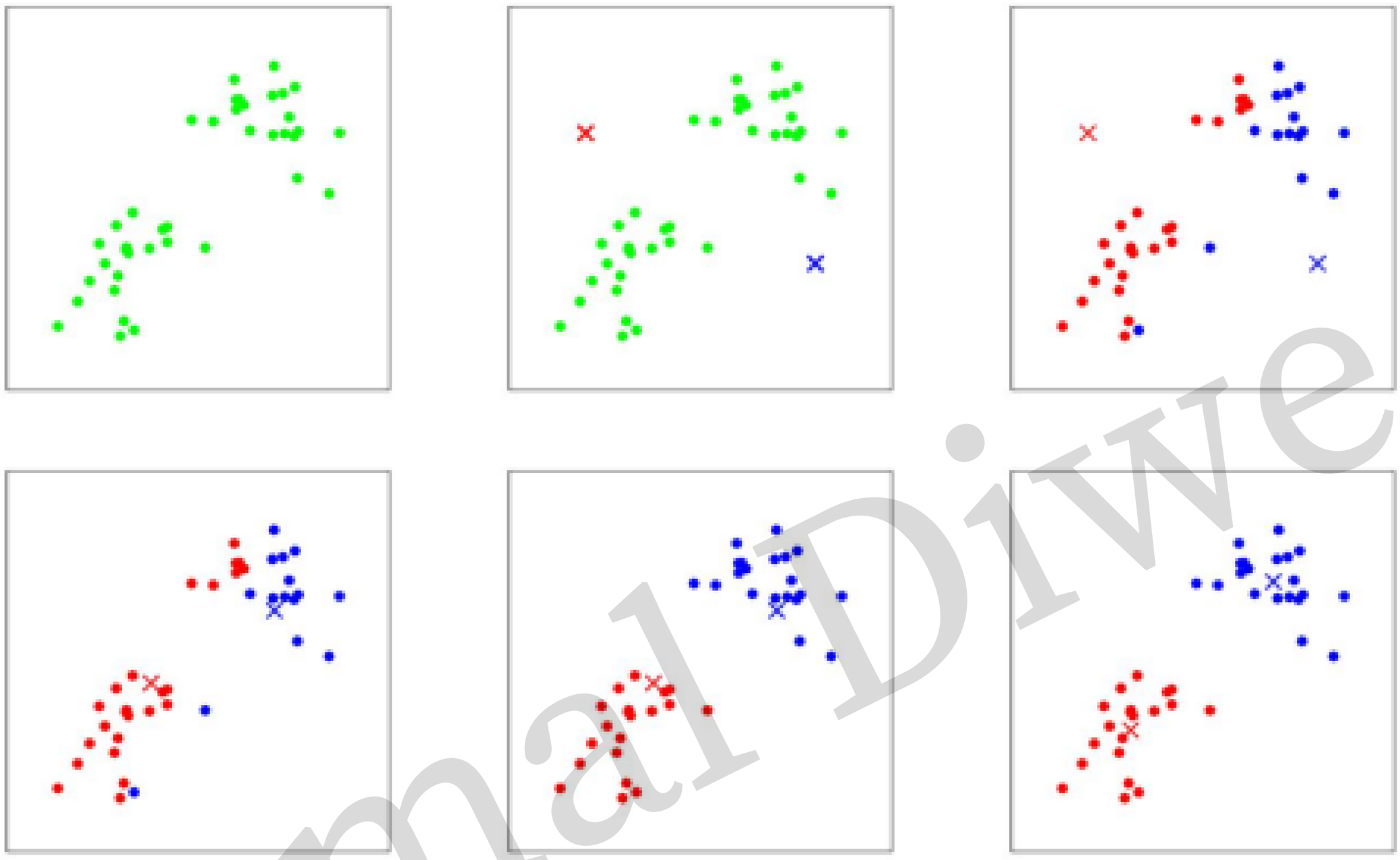


Figure 10.1: K-means algorithm. Training examples are shown as dots, and cluster centroids are shown as crosses. (a) Original dataset. (b) Random initial cluster centroids (in this instance, not chosen to be equal to two training examples). (c-f) Illustration of running two iterations of k-means. In each iteration, we assign each training example to the closest cluster centroid (shown by "painting" the training examples the same color as the cluster centroid to which is assigned); then we move each cluster centroid to the mean of the points assigned to it. (Best viewed in color.) Images courtesy Michael Jordan.

K-means Optimization objective:

$c^{(i)}$ = index of cluster $(1, 2, \ldots, k)$ to which example $x^{(i)}$ is currently assigned

μ_k = cluster centroid k

$\mu_{c^{(i)}}$ = cluster centroid of cluster to which example $x^{(i)}$ has been assigned

cost function

$$J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_k) = \frac{1}{m} \sum_{i=1}^{m} \| x^{(i)} - \mu_{c^{(i)}} \|^2$$

$$\min_{c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_k} J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots \mu_k)$$

This function is also known as distortion function.

J measures the sum of square distances between each training example $x^{(i)}$ and the cluster centroid $\mu_{c^{(i)}}$ to which it has been assigned.

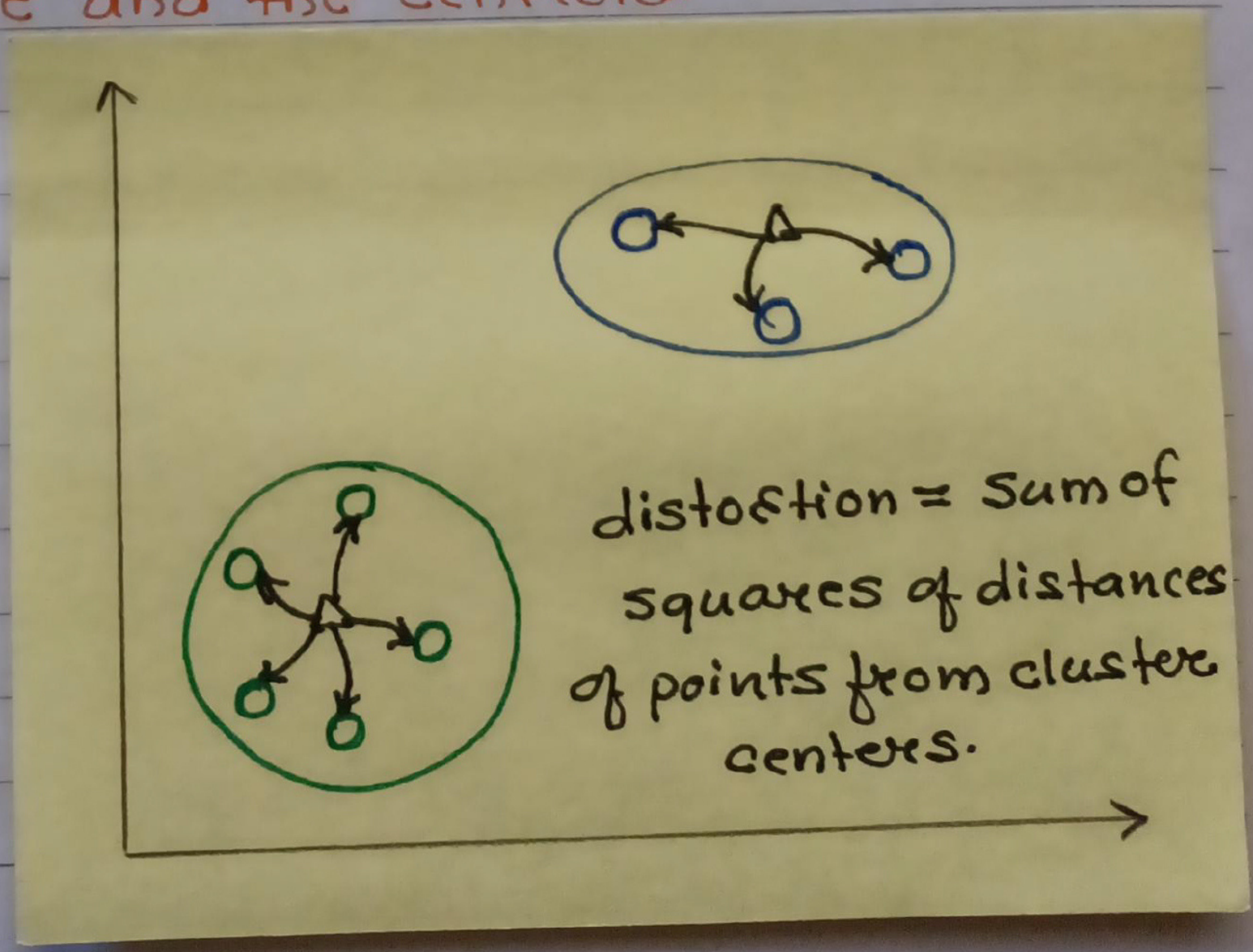
__/__/__

The k-means algorithm is optimizing a specific cost function called the distortion function.

This function measures the average square distance between every training example and the location of the cluster centroid to which it has been assigned.

The algorithm first updates the cluster assignments to minimize the cost function while holding the centroid locations constant, and then updates the centroid locations to minimize the cost function while holding the cluster assignments constant.

During the first step, the algorithm assigns a training example to the closest centroid to minimize the distance of square distance betn the example and the centroid.



Initializing K-means

Random initialization:

choose $K < m$ → no. of cluster centers.
↳ training examples

For $i = 1$ to 100 {

Randomly initialize k-means.
Run k-means. Get $c^{(1)}, \ldots c^{(m)}, \mu_1, \ldots \mu_K$
compute cost function (distortion)

$J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$

}

Pick set of clusters that gave lowest cost J.

- The first step in k-means clustering is to choose random locations as initial guess.

To avoid poor results, K-means can be run multiple times with different random initializations.

After running k-means multiple times, the set of cluster with the lowest distortion or cast function can be selected.

How to find the right k ?

- No absolute method to find right number of clusters (k) in k-means clustering.

- Elbow method

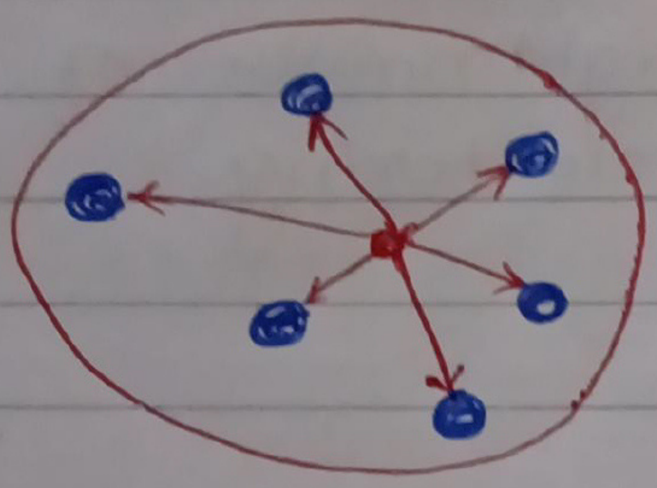
The elbow method, which involves plotting the distortion function J as a function of k, is one technique used to determine k.

However, for many applications, the appropriate number of clusters is ambiguous, so the down-stream purpose should be considered when evaluating k-means.

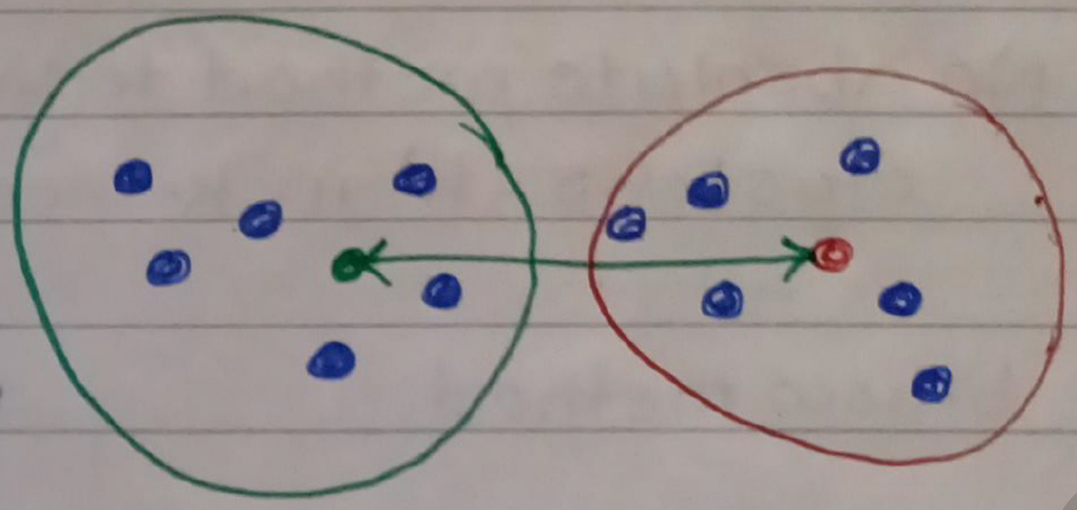
How to validate cluster ?

1. Dumm Index (Dunn Index)
2. Silhouette score

Dunn Index



Intra cluster distance



Inter cluster distance

$$\text{Dunn Index} = \frac{\min(\text{Inter cluster distance})}{\max(\text{Intra cluster distance})}$$

Dunn Index is the ratio of the minimum of inter-cluster distances and maximum of intra-cluster distances.

We want to maximize the dunn Index. The more the value of the Dunn Index, the better the clusters will be.

~~let's~~ Intuition behind the dunn index :

$$\text{Dunn Index} = \frac{\min(\text{Inter cluster distance})}{\max(\text{Intra cluster distance})}$$

clusters are far aparet

In order to maximize the value of the Dunn Index, the numerator should be maximum.

Here, we are taking the minimum of the inter cluster distances. So, the distance between even the closest clusters should be more. which will eventually make sure that the clusters are far away from each other.

$$\text{Dunn Index} = \frac{\min(\text{Inter cluster distance})}{\max(\text{intra cluster distance})}$$

clusters are compact

also, the denominator should be minimum to maximize the dunn Index.

Here, we are taking the maximum of all intracluster distances.

Again, the intuition is same here. The maximum distance between the cluster centroids and the points should be minimum, eventually ensuring that the clusters are compact.

Silhouette score

The silhouette score measures the similarity of each point to its own cluster compared to other clusters,
and the silhouette plot visualizes these scores for each sample.

$$\text{silhouette score} = \frac{bi - ai}{\max(a, b)}$$

where, $a_i \rightarrow$ Intra-cluster distance
$b_i \rightarrow$ Nearest-cluster distance

Silhouette score ranges from -1 to 1.

A high silhouette score indicates that the clusters are well separated, and each sample is more similar to the samples in its own cluster than to sample in other clusters.

A silhouette score close to 0 suggests overlapping clusters, and a -ve score suggests poor clustering solutions.

Komal Diwe