# Classification: Clustering

## Unsupervised Algorithm:

# K-Means Algorithm for Clustering

K-Means is an unsupervised machine learning Algorithm used for clustering.

It partitions a dataset into K clusters based on similarity.

Clusters are formed by minimizing the sum of squared distances between data points and the centroid of their assigned cluster.

## Example of Implementation of K-Means:

| Individual | Variable 1 | Variable 2 |
|---|---|---|
| 1 | 1.0 | 1.0 |
| 2 | 1.5 | 2.0 |
| 3 | 3.0 | 4.0 |
| 4 | 5.0 | 7.0 |
| 5 | 3.5 | 5.0 |
| 6 | 4.5 | 5.0 |
| 7 | 3.5 | 4.5 |

**Step 1:** Choose K initial centroids (points representing cluster centers)

Randomly we choose following two centroids (K=2) for two clusters.

In this case the 2 centroids are:

$m1 = (1.0, 1.0)$ and $m2 = (5.0, 7.0)$

**Step 2:** Assign each data point to the nearest centroid, forming K clusters. For this we have to find the distance of each data point by using formulas like Euclidean or Manhattan.

$$\text{Euclidean Distance} = \sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$$

$$\text{Manhattan Distance} = |x_1-x_2| + |y_1-y_2|$$

| Individual | Variable 1 | Variable 2 | Centroid 1 | Centroid 2 |
|---|---|---|---|---|
| 1 | 1.0 | 1.0 | 0 | 7.21 |
| 2 | 1.5 | 2.0 | 1.12 | 6.10 |
| 3 | 3.0 | 4.0 | 3.61 | 3.61 |
| 4 | 5.0 | 7.0 | 7.21 | 0 |
| 5 | 3.5 | 5.0 | 4.72 | 2.5 |
| 6 | 4.5 | 5.0 | 5.31 | 2.06 |
| 7 | 3.5 | 4.5 | 4.30 | 2.92 |

Thus, we obtain two clusters containing:

$\{1,2,3\}$ and $\{4,5,6,7\}$

**Step 3:** Recalculate the centroids as the mean of all points in the cluster.

$$m_1 = \left(\frac{1}{3}(1.0+1.5+3.0), \frac{1}{3}(1.0+2.0+4.0)\right)$$
$$= (1.83, 2.33)$$

$$m_2 = \left(\frac{1}{4}(5.0+3.5+4.5+3.5), \frac{1}{4}(7.0+5.0+5.0+4.5)\right)$$
$$= (4.12, 5.38)$$

**Step 4:** Repeat until there is no change in the cluster.

| Individual | Variable 1 | Variable 2 | Centroid 1 | Centroid 2 |
|---|---|---|---|---|
| 1 | 1.0 | 1.0 | 1.57 | 7.21 |
| 2 | 1.5 | 2.0 | 0.47 | 6.10 |
| 3 | 3.0 | 4.0 | 2.04 | 1.78 |
| 4 | 5.0 | 7.0 | 5.64 | 1.84 |
| 5 | 3.5 | 5.0 | 3.15 | 0.73 |
| 6 | 4.5 | 5.0 | 3.78 | 0.54 |
| 7 | 3.5 | 4.5 | 2.74 | 1.08 |

New Clusters: $\{1,2\}$ and $\{3,4,5,6,7\}$

Next Centroids: $m1 = (1.25, 1.5)$ & $m2 = (3.9, 5.1)$

| Individual | Variable 1 | Variable 2 | Centroid 1 | Centroid 2 |
|---|---|---|---|---|
| 1 | 1.0 | 1.0 | 0.58 | 5.02 |
| 2 | 1.5 | 2.0 | 0.58 | 3.92 |
| 3 | 3.0 | 4.0 | 3.05 | 1.42 |
| 4 | 5.0 | 7.0 | 6.66 | 2.20 |
| 5 | 3.5 | 5.0 | 4.18 | 0.41 |
| 6 | 4.5 | 5.0 | 4.78 | 0.61 |
| 7 | 3.5 | 4.5 | 3.75 | 0.72 |

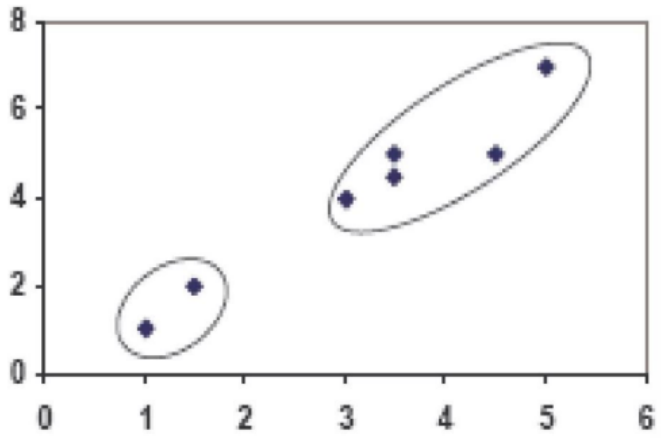
Hence, there is no change in clusters.

The final 2 clusters are:

$\{1,2\}$ and $\{3,4,5,6,7\}$

**Note:** In case of K-medians Algorithm, we just take median instead of mean for finding new centroids.

## Plot




## Supervised Algorithm:

# K_Nearest Neighbors Algorithm (K-NN):

Example: Predicting Movie Genre

| MDb Rating | Duration | Genre |
|---|---|---|
| 8.0 (Mission Impossible) | 160 | Action |
| 6.2 (Gadar 2) | 170 | Action |
| 7.2 (Rocky & Rani) | 160 | Comedy |
| 8.2 (OMG 2) | 155 | Comedy |

Now predict the genre of "Barbie" movie with IMDb rating 7.4 and duration 144.

Step 1: Calculate Distances:

Calculate the Euclidean distance between the new movie and each movie in data set.

Distance to $(8.0, 160) = \sqrt{(7.4-8.0)^2 + (114-160)^2} = 46.0$

Distance to $(6.2, 160) = 56.01$

Distance to $(7.2, 168) = 54.00$

Distance to $(8.2, 155) = 41.00$

Step 2: Select Nearest Neighbors:

Let, $K = 3$ $\quad K = \{41.00, 46.00, 54.00\}$

Step 3: Majority Voting (Classification):

{Action, Comedy, Comedy} So the genre is Comedy of "Barbie" movie.