K- MEANS INTRODUCTION

In this article, we will cover k-means clustering. In general, Clustering is defined as the grouping of data points such that the data points in a group will be similar or related to one another and different from the data points in another group. The goal of clustering is to determine the intrinsic grouping in a set of unlabelled data.

K- means is an unsupervised partitional clustering algorithm that is based on grouping data into k - numbers of clusters by determining centroid using the Euclidean or Manhattan method for distance calculation. It groups the object based on minimum distance.

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Euclidean Distance

ALGORITHM

1. First, initialize the number of clusters, K (Elbow method is generally used in selecting number of clusters)
2. Randomly select the k data points for centroid. A centroid is the imaginary or real location representing the center of the cluster.
3. Categorize each data items to its closest centroid and update the centroid coordinates calculating the average of items coordinates categorized in that group so far
4. Repeat the process for a number of iterations till successive iterations clusters data items into same group

HOW IT WORKS ?

In the beginning, the algorithm chooses *k* centroids in the dataset randomly after shuffling the data. Then it calculates the distance of each point to each centroid using euclidean distance calculation method. Each centroid assigned represents a cluster and the points are assigned to the closest cluster. At the end of the first iteration, the centroid values are recalculated, usually taking the arithmetic mean of all points in the cluster. In every iteration, new centroid values are calculated until successive iterations provide the same centroid value.

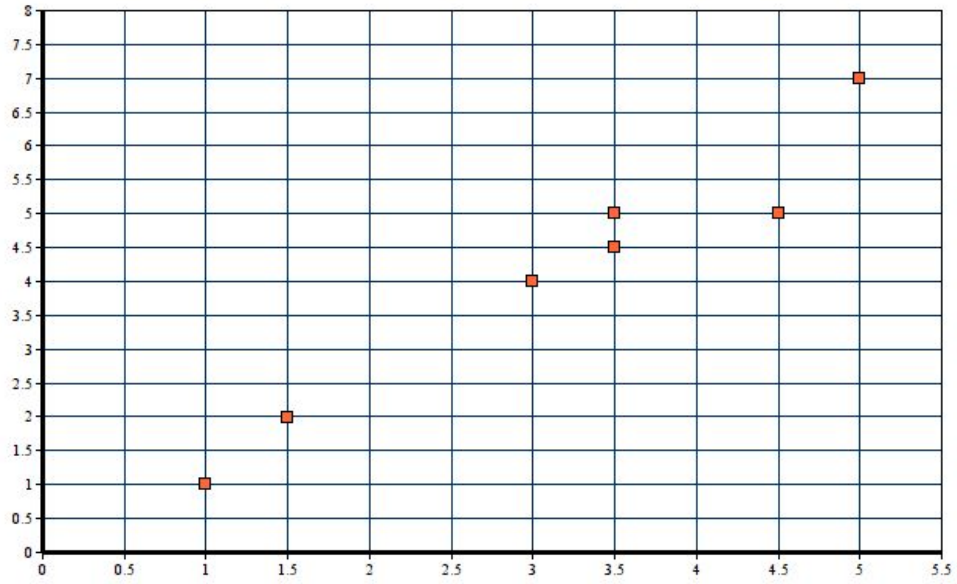
Let's kick things off with a simple example : Suppose we have data points (1,1), (1.5,2), (3,4), (5,7), (3.5,5), (4.5,5), (3.5,4.5). Let us suppose k = 2 i.e. dataset should be grouped in two clusters. Here we are using the Euclidean distance method.

Step 1 : It is already defined that k = 2 for this problem

Step 2: Since k = 2, we are randomly selecting two centroid as c1(1,1) and c2(5,7)

Step 3: Now, we calculates the distance of each point to each centroid using euclidean distance calculation method:

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$



ITERATION 01

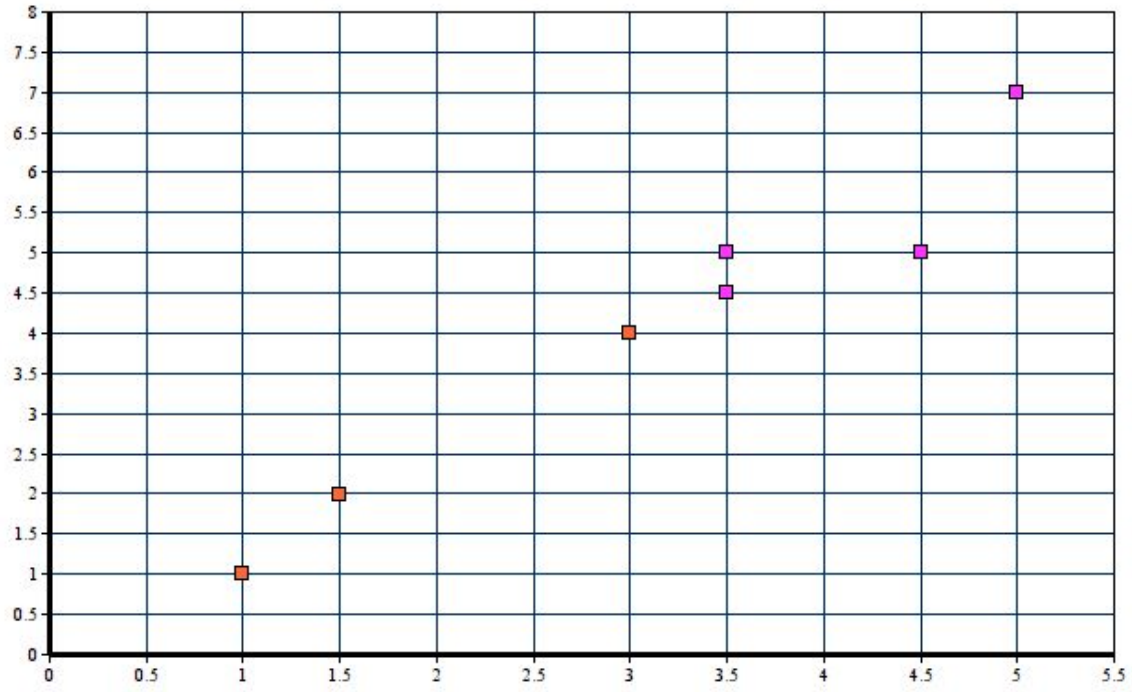
X1	Y1	**X2**	**Y2**	**D1**	X1	Y1	**X2**	**Y2**	**D2**	**Remarks**
1	1	**1**	**1**	**0**	1	1	**5**	**7**	**7.21**	D1<D2 : (1,1) belongs to c1
1.5	2	**1**	**1**	**1.12**	1.5	2	**5**	**7**	**6.1**	D1<D2 : (1.5,2) belongs to c1
3	4	**1**	**1**	**3.61**	3	4	**5**	**7**	**3.61**	D1<D2 : (3,4) belongs to c1
5	7	**1**	**1**	**7.21**	5	7	**5**	**7**	**0**	D1>D2 : (5,7) belongs to c2
3.5	5	**1**	**1**	**4.72**	3.5	5	**5**	**7**	**2.5**	D1>D2 : (3.5,5) belongs to c2

4.5	5	**1**	**1**	**5.32**	4.5	5	**5**	**7**	**2.06**	D1>D2 : (5.5,5) belongs to c2
3.5	4.5	**1**	**1**	**4.3**	3.5	4.5	**5**	**7**	**2.91**	D1>D2 : (3.5,4.5) belongs to c2

Note: D1 & D2 are euclidean distance between centroid (x2,y2) and data points (x1,y1)

In cluster c1 we have (1,1), (1.5,2) and (3,4) whereas centroid c2 contains (5,7), (3.5,5), (4.5,5) & (3.5,4.5). Here, new centroid is the algebraic mean of all the data items in a cluster.

C1(new) = ((1+1.5+3)/3 , (1+2+4)/3) = **(1.83, 2.33)**
C2(new) = ((5+3.5+4.5+3.5)/4, (7+5+5+4.5)/4) = **(4.125, 5.375)**



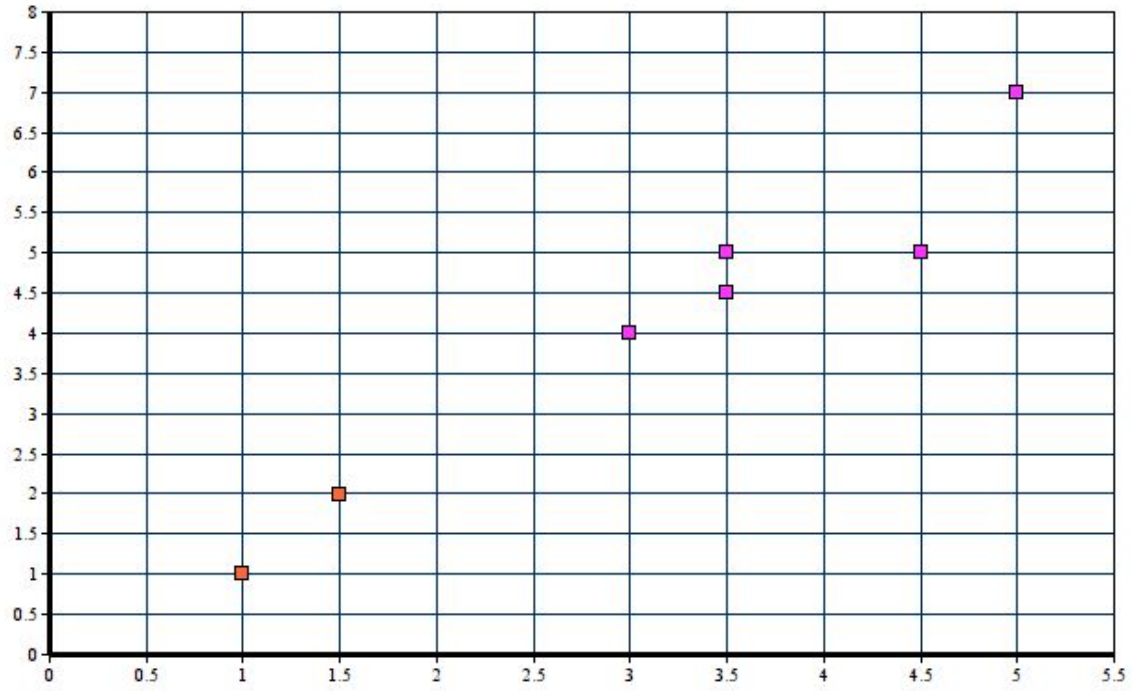
ITERATION 02

X1	Y1	**X2**	**Y2**	**D1**	X1	Y1	**X2**	**Y2**	**D2**	**Remarks**
1	1	**1.83**	**2.33**	**1.56**	1	1	**4.12**	**5.37**	**5.37**	(1,1) belongs to c1
1.5	2	**1.83**	**2.33**	**0.46**	1.5	2	**4.12**	**5.37**	**4.27**	(1.5,2) belongs to c1
3	4	**1.83**	**2.33**	**2.03**	3	4	**4.12**	**5.37**	**1.77**	(3,4) belongs to c2

5	7	**1.83**	**2.33**	**5.64**	5	7	**4.12**	**5.37**	**1.84**	(5,7) belongs to c2
3.5	5	**1.83**	**2.33**	**3.14**	3.5	5	**4.12**	**5.37**	**0.72**	(3.5,5) belongs to c2
4.5	5	**1.83**	**2.33**	**3.77**	4.5	5	**4.12**	**5.37**	**0.53**	(5.5,5) belongs to c2
3.5	4.5	**1.83**	**2.33**	**2.73**	3.5	4.5	**4.12**	**5.37**	**1.07**	(3.5,4.5) belongs to c2

In cluster c1 we have (1,1), (1.5,2)) whereas centroid c2 contains (3,4),(5,7), (3.5,5), (4.5,5) & (3.5,4.5). Here, new centroid is the algebraic mean of all the data items in a cluster.

C1(new) = ((1+1.5)/2 , (1+2)/2) = **(1.25,1.5)**
C2(new) = ((3+5+3.5+4.5+3.5)/5, (4+7+5+5+4.5)/5) = **(3.9, 5.1)**



ITERATION 03

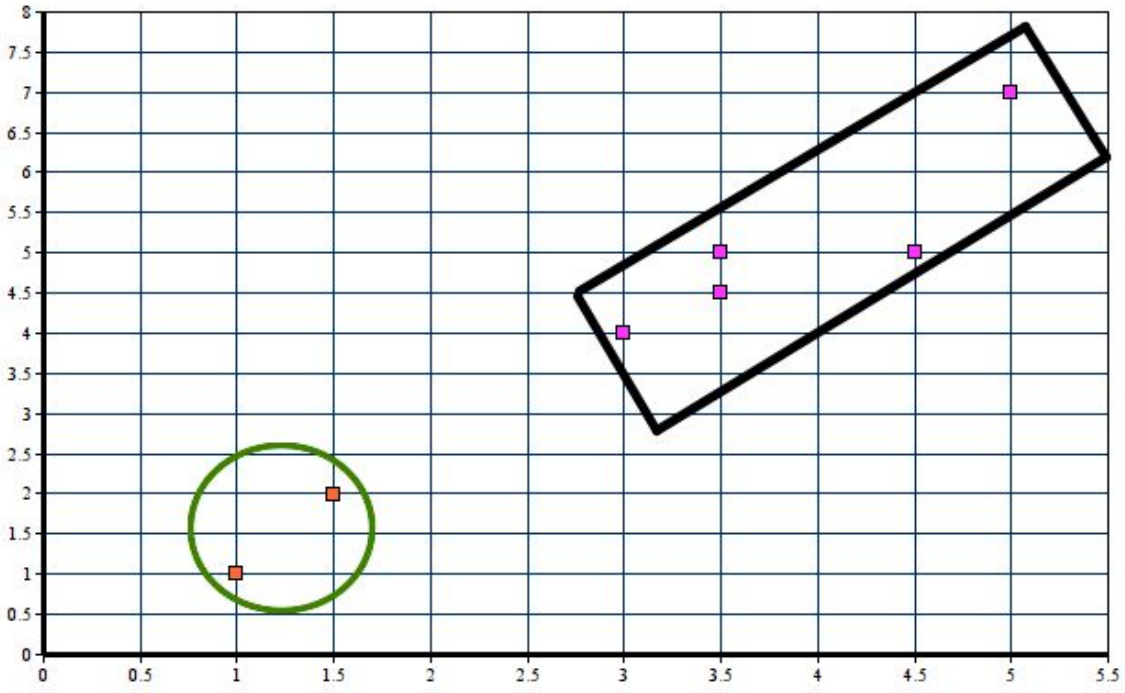
X1	Y1	**X2**	**Y2**	**D1**	X1	Y1	**X2**	**Y2**	**D2**	**Remarks**
1	1	**1.25**	**1.5**	**0.56**	1	1	**3.9**	**5.1**	**5.02**	(1,1) belongs to c1
1.5	2	**1.25**	**1.5**	**0.56**	1.5	2	**3.9**	**5.1**	**3.92**	(1.5,2) belongs to c1

3	4	**1.25**	**1.5**	**3.05**	3	4	**3.9**	**5.1**	**1.42**	(3,4) belongs to c2
5	7	**1.25**	**1.5**	**6.66**	5	7	**3.9**	**5.1**	**2.19**	(5,7) belongs to c2
3.5	5	**1.25**	**1.5**	**4.16**	3.5	5	**3.9**	**5.1**	**0.41**	(3.5,5) belongs to c2
4.5	5	**1.25**	**1.5**	**4.77**	4.5	5	**3.9**	**5.1**	**0.60**	(5.5,5) belongs to c2
3.5	4.5	**1.25**	**1.5**	**3.75**	3.5	4.5	**3.9**	**5.1**	**0.72**	(3.5,4.5) belongs to c2

In cluster c1 we have (1,1), (1.5,2)) whereas centroid c2 contains (3,4),(5,7), (3.5,5), (4.5,5) & (3.5,4.5). Here, new centroid is the algebraic mean of all the data items in a cluster.

C1(new) = ((1+1.5)/2 , (1+2)/2) = **(1.25,1.5)**
C2(new) = ((3+5+3.5+4.5+3.5)/5, (4+7+5+5+4.5)/5) = **(3.9, 5.1)**

Step 4 : In the 2nd and 3rd iteration, we obtained the same centroid points. Hence clusters of above data point is :



CODE FROM SCRATCH

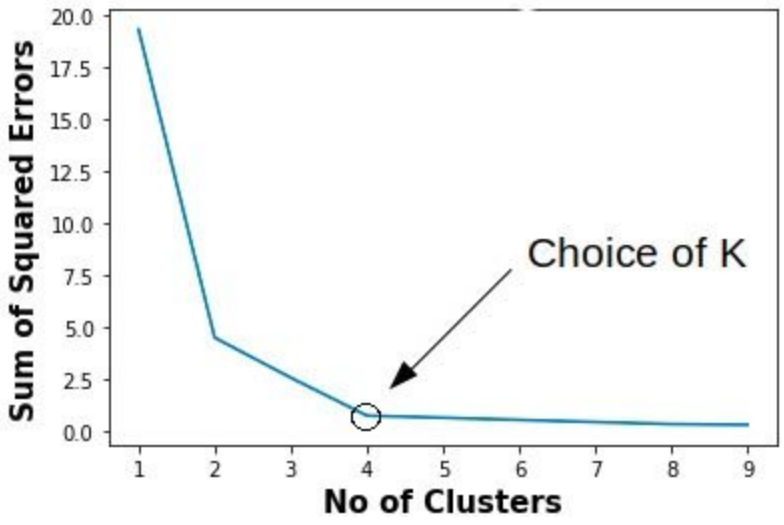
https://github.com/AI-HUB-COURSE/Kmeans-by-DS

CHOOSING VALUE OF K

While working with the k-means model, one thing we must keep in mind is the number of clusters ‘k’ . We should make sure that we are choosing the optimum number of clusters for the given data set. But, here arises a question, how to choose the optimum value of k ?? We use the elbow method which is generally used in analysing the optimum value of k .
Elbow method is based on the principle that “**Sum of squares of distances of every data point from its corresponding cluster centroid should be as minimum as possible”.**

STEPS OF CHOOSING BEST K VALUE

1. Run k-means clustering model on various values of k
2. For each value of K, calculate Sum of squares of distances of every data point from its corresponding cluster centroid which is called WCSS (Within-Cluster Sums of Squares)
3. Plot the value of WCSS with respect to various values of K
4. To select the value of k , we choose the value where there is bend (knee) on the plot i.e. WCSS isn’t increasing rapidly.



PROS OF K MEANS

- Relatively simple to learn and understand as the algorithm solely depends on the euclidean method of distance calculation.
- K means works on minimising Sum of squares of distances, hence it guarantees convergence
- Computational cost is O(K*n*d) , hence K means is fast and efficient

CONS OF K MEANS

- Difficulty in choosing optimum number of clusters K
- K means has problem when clusters are of different size, densities, and non-globular shapes
- K means has problems when data contains outliers
- As the number of dimensions increases, the difficulty in getting the algorithm to converge increases due to the curse of dimensionality
- If there is **overlapping between clusters**, k-means doesn't have an intrinsic measure for uncertainty

REFERENCES

1. Understanding the Mathematics behind K-Means Clustering
2. **Implementing K-means Clustering from Scratch - in Python**
3. **Machine-Learning-Algorithms-from-Scratch**
4. Mathematics behind K-Mean Clustering algorithm