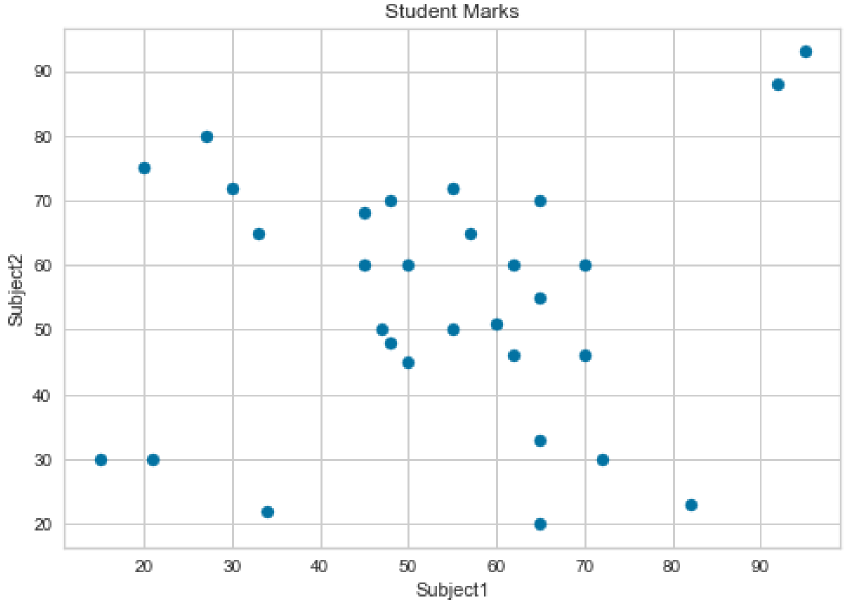
Student Marks (Out of 100)

Roll#	Subject1	Subject2	Roll#	Subject1	Subject2	Roll#	Subject1	Subject2
1	48	48	11	57	65	21	65	20
2	92	88	12	33	65	22	72	30
3	34	22	13	82	23	23	55	72
4	45	68	14	70	46	24	95	93
5	50	45	15	21	30	25	27	80
6	20	75	16	65	33	26	15	30
7	48	70	17	65	55	27	45	60
8	60	51	18	30	72	28	50	60
9	55	50	19	62	46	29	70	60
10	62	60	20	47	50	30	65	70



The above table contains the marks of 30 students that they have scored for 2 subjects. The scatter plot above shows the distribution of these data points.

Using this dataset, let's look at some of the clustering algorithms and analyse their outputs.

K-Means Clustering

A flat/exclusive algorithm which forms a specified number of clusters based on the cluster centers and data points that are at the shortest distance from the cluster centers.

Cluster centers are randomly chosen for each cluster in the beginning and then adjusted so as to form tight clusters with maximum inter cluster distance.

Uses euclidean method for distance calculation.

Number of clusters is the most important hyper-parameter for this algorithm.

DBScan - Density Based Spatial Clustering of applications with noise

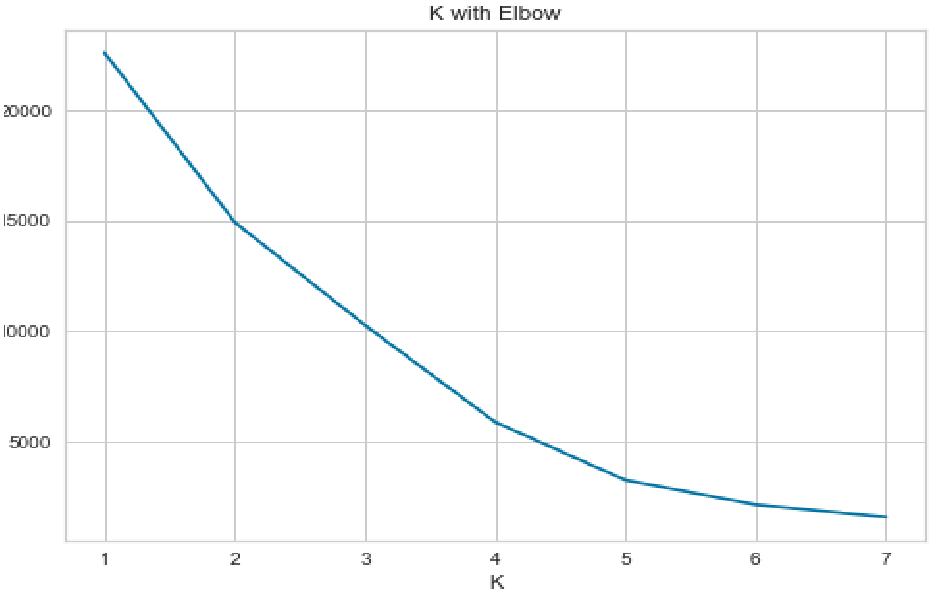
A density based clustering algorithm which forms clusters based on a specified distance (epsilon) of each data point from a specified number of data points (Min Samples).

Classifies the data points as core points, border points and noise for cluster formation. Direct or indirect connectivity between data points is calculated for forming clusters.

Enables Outlier detection

Distance calculation method used is Euclidean by default.

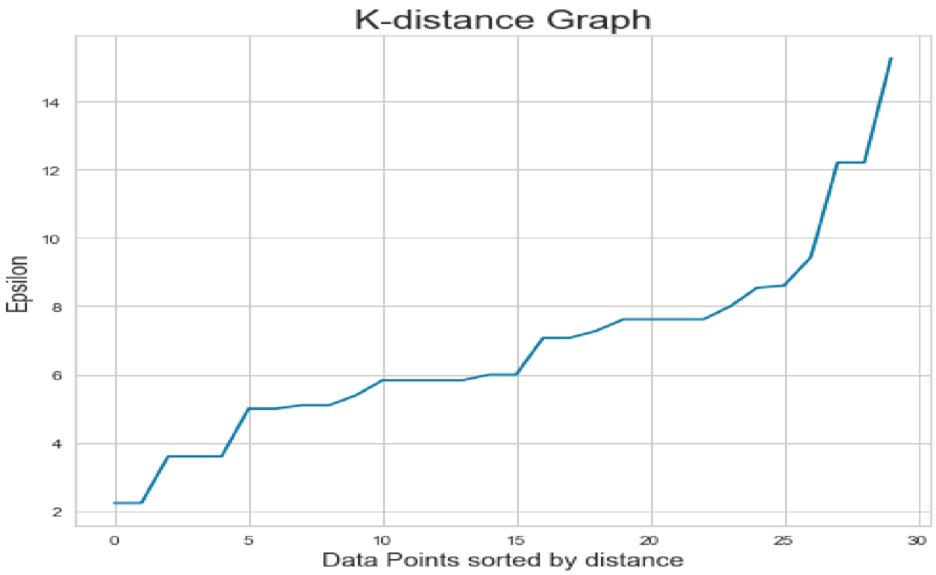
Elbow Analysis for determining the n_clusters parameter



From the above diagram, the Elbow starts tapering off maximum at K = 4 and keeps reducing till K=6. So the ideal number #of clusters seems to be in this range. Let’s run the K Means clustering algorithm for these 3 K values and analyse the outputs.

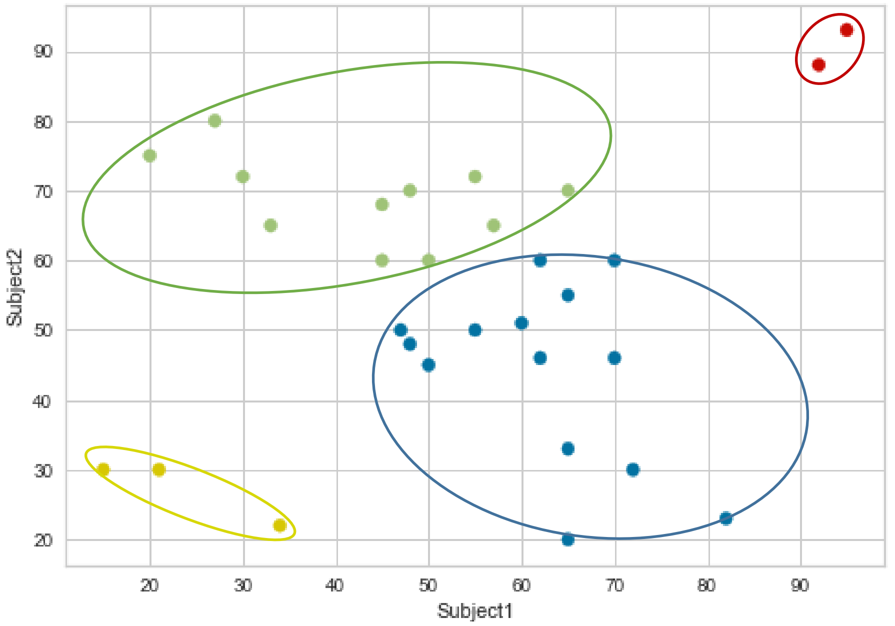
K-Means Output with hyper-parameter n_clusters = 4, 5 and 6

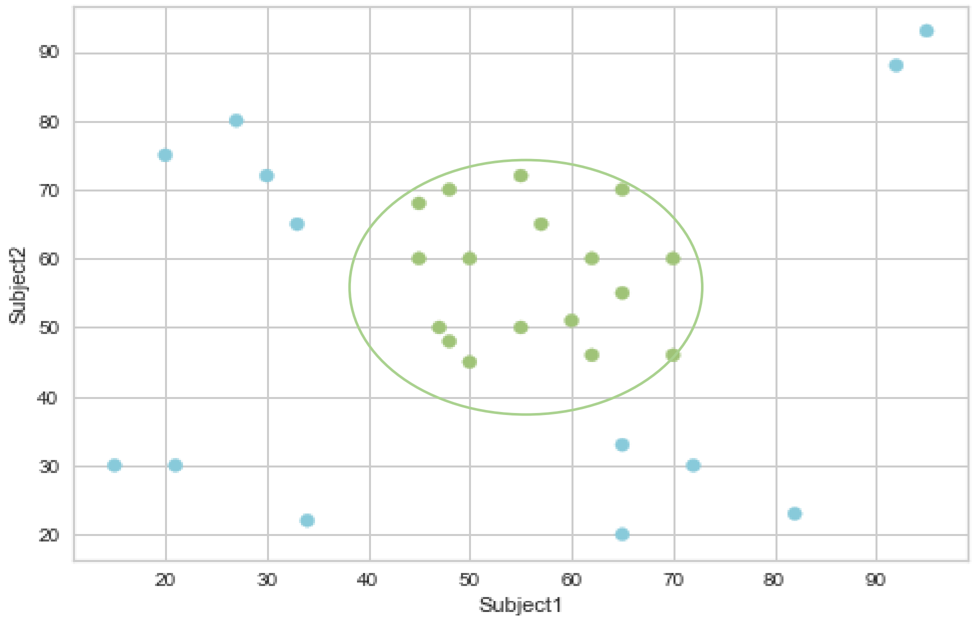
K Distance Graph for determining the epsilon parameter

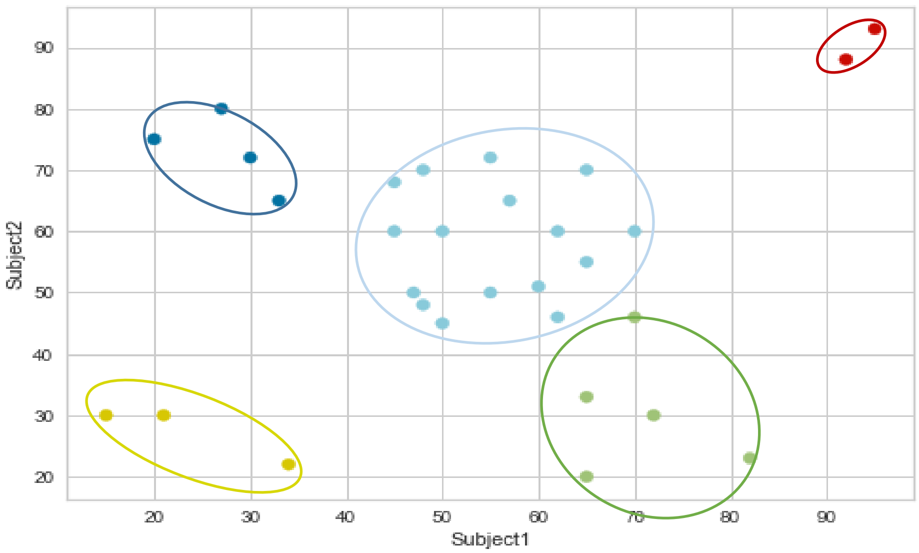


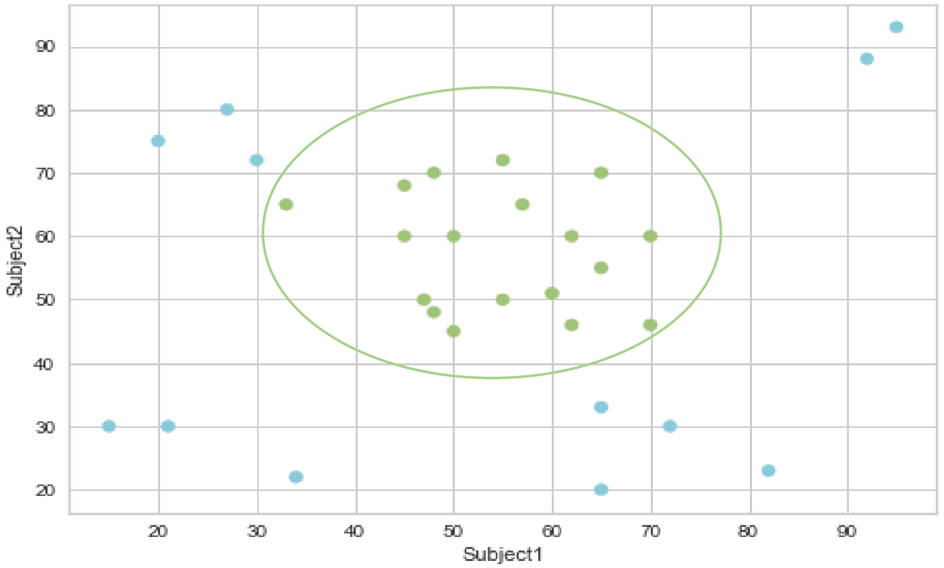
From the K Distance graph, the epsilon value seems to lie in the range between 12 and 14. We will run the DBScan algorithm for these 3 epsilon values and analyse the outputs.

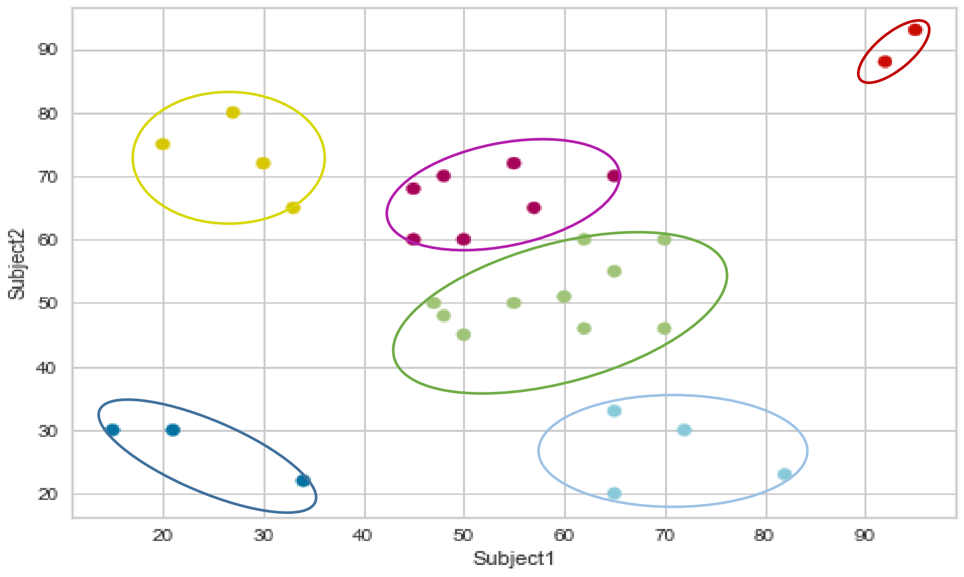
DBScan Output with hyper-parameter eps = 12, 13 and 14

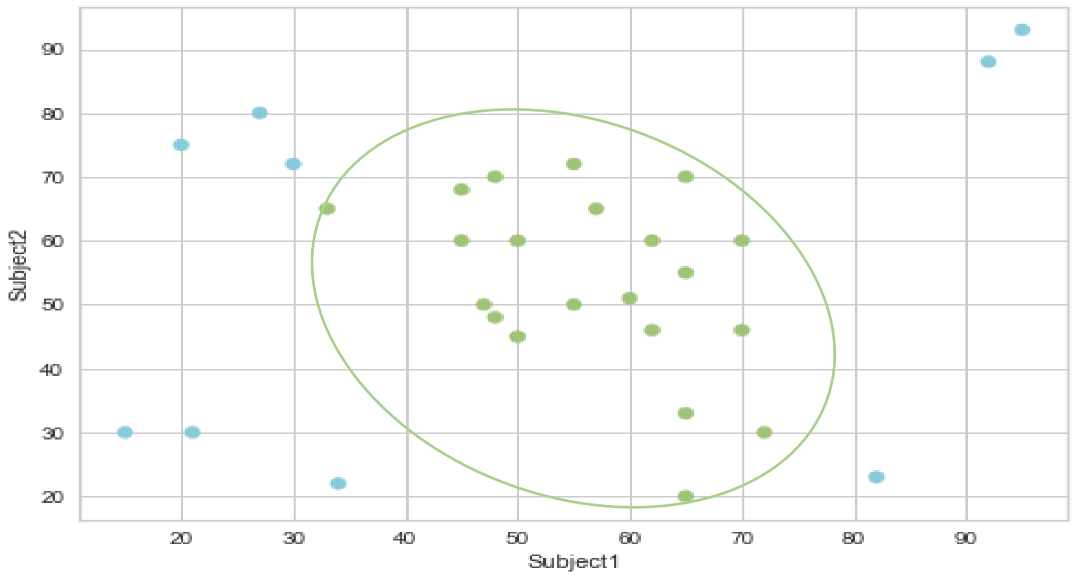

Subject2
90
80
70
60
50
40
30
20
20
30
40
50
60
70
80
90
Subject1


Subject2
90
80
70
60
50
40
30
20
20
30
40
50
60
70
80
90
Subject1


Subject2
90
80
70
60
50
40
30
20
20
30
40
50
60
70
80
90
Subject1


Subject2
90
80
70
60
50
40
30
20
20
30
40
50
60
70
80
90
Subject1





K Means Output Analysis

With 4 clusters, it looks like the algorithm has identified 4 groups of students, red cluster has students with high score in both subjects, yellow has students with low score in both subjects, green cluster has student with >60 marks in Subject2 and blue cluster has students with <=60 marks in Subject2.

With 5 clusters, the algorithm has identified 5 groups of students, red cluster has students with high score in both subjects, yellow has students with low score in both subjects, blue cluster are the students with average marks in both subjects, dark blue cluster has students with high marks in Subject2 but score poorly in Subject1 and green cluster has students high marks in Subject1 but score poorly in Subject2. 0

And with 6 clusters, within the average students group, it has formed 2 clusters, one with score >= 60 for Subject 2 and the other for the rest of the average students.

DBScan Output Analysis

With all the 3 epsilon values, the output of DBScan shows one cluster, in which most of the data points are closely located. These are the students with an average score of 45-75 in both subjects. Rest of the data points are identified as noise or outliers.

The cluster with epsilon 12 has the most accurate cluster formation for students with average score. With epsilon 13 and 14 few students with poor score in both subjects have been included in the cluster.

This Students marks data may not be the best use case for DBScan. But if we re-name this data set to 'House Location' and the attributes represent the location of houses in a locality, this clustering helps identify the most populated part of that locality, etc.

DBScan can be used to identify clusters where closeness of not just the immediate data points but even the data points that are close to the immediate data points need to be part of the cluster, thereby forming clusters which may have any arbitrary shape.