* DBScan Clustering

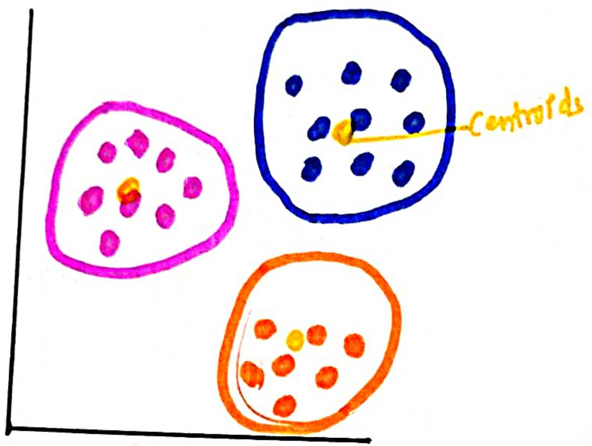
↓

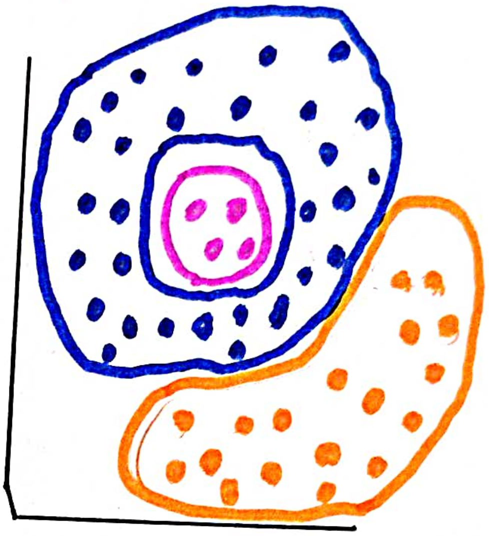
Density Based Spatial Clustering of application with noise.

Density-Based clustering

* Spherical-shape clusters

* Arbitary shape clusters





- K-means assigns all Points To a cluster Even if they do not belong in any

- Density based clustering Locates regions of "high density", and seperates Outliers.

*** Based on Given density of values it is identifying the clusters

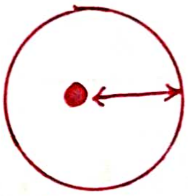
* What is DBSCAN?

* DBSCAN:

- it is one of the most common clustering algorithms
- Works based on density of objects.

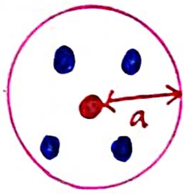
* R (radius of neighborhood)

- Radius (R) that if includes enough number of points with in, we call it a dense area.

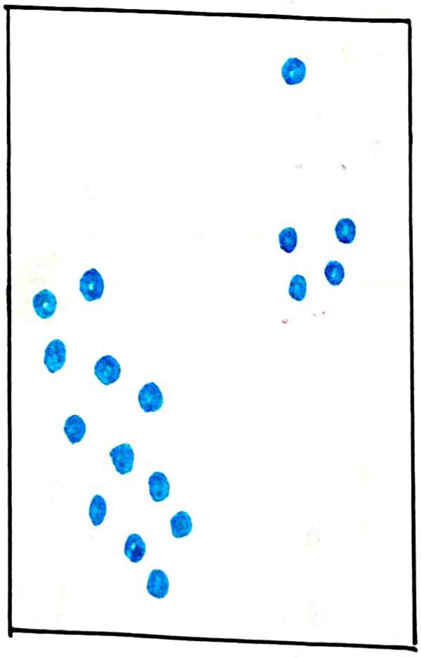


* M (min number of neighbors)

- The minimum number of data points we want in a neighborhood to define a cluster.



Ex:-



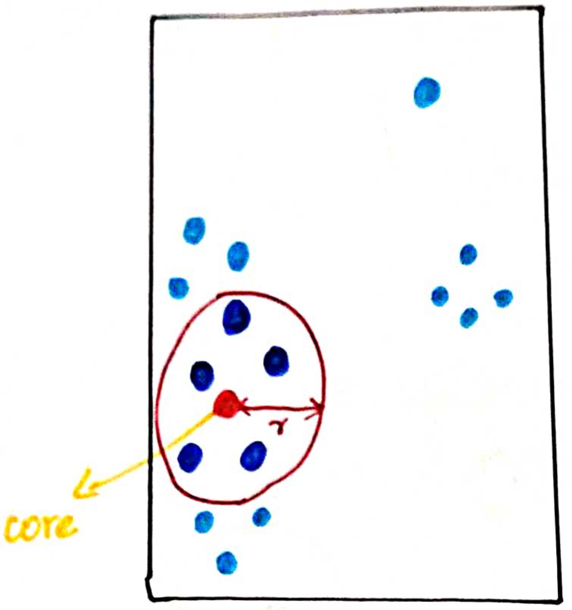
Each point is either :

- core Point
- border Point
- Outlier point

∴ R = 2 unit, M = 6

* DBSCAN algorithm :- Core point

↓ which is satisfy minimum no. of neighbours.

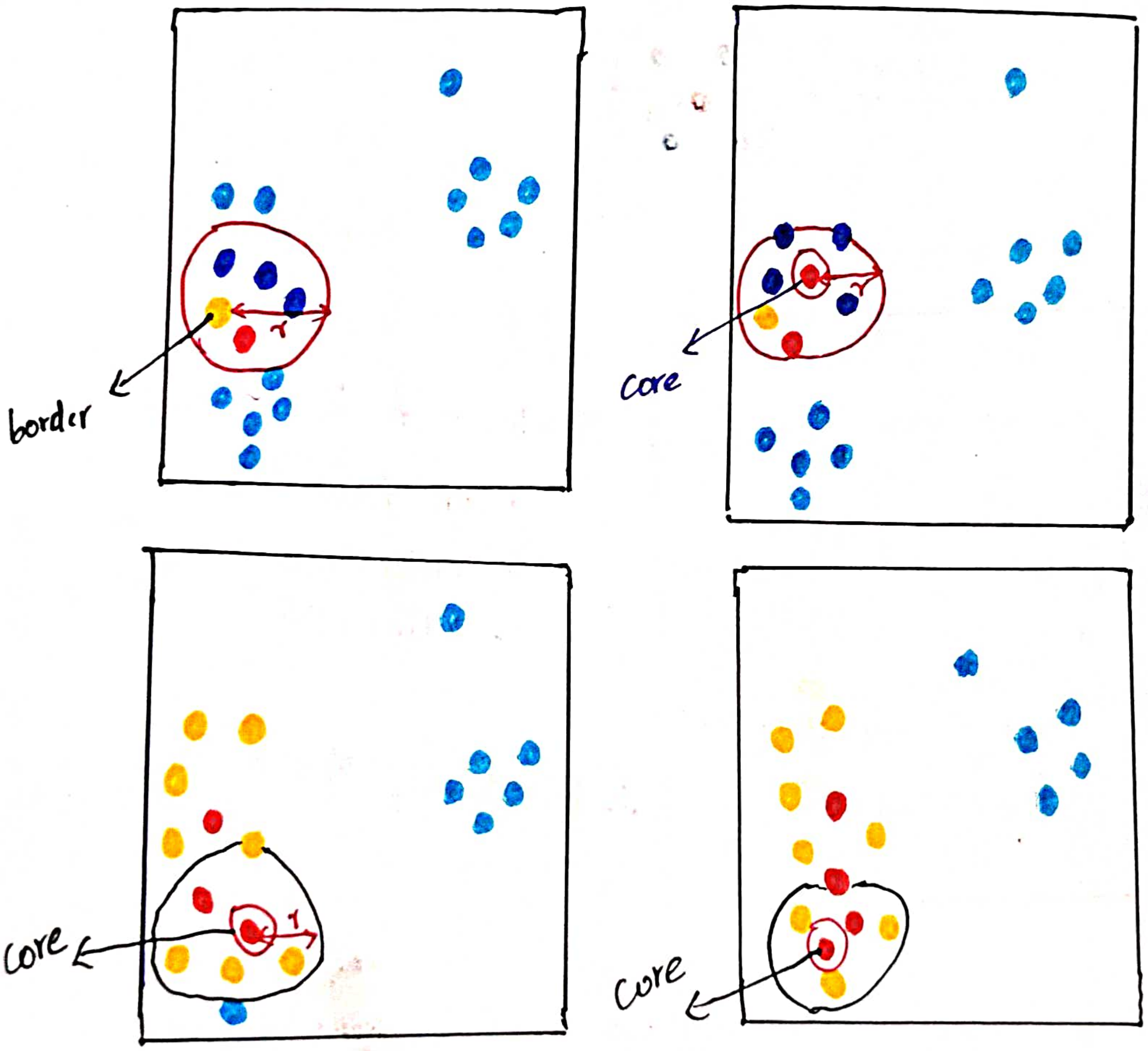


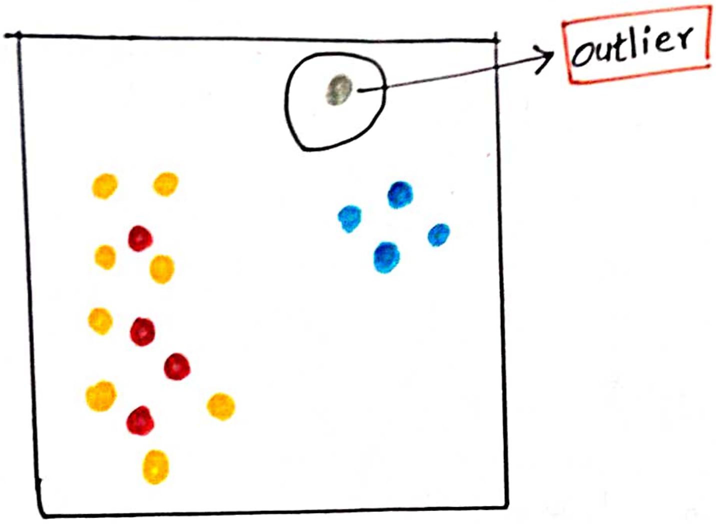
R = 2 unit, M = 6

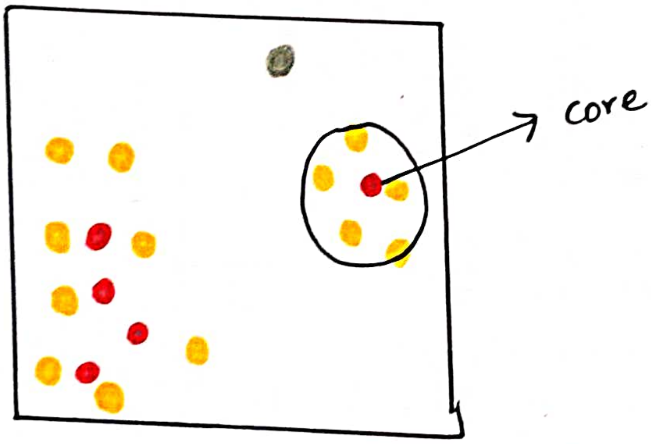
↓ Minimum.

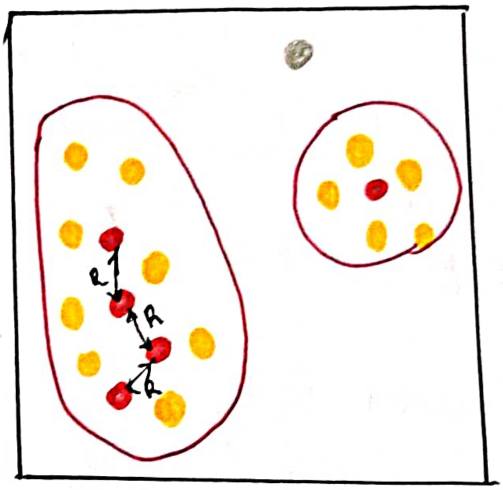
* DBSCAN algorithm : border point

R= 2unit , M=6









R= 2unit , M=6 .

CODE :

Same data & Same steps till modelling

MODEL: DBSCAN

```
from sklearn.cluster import DBSCAN
# dbs = DBSCAN (eps=5 , min_samples = 5)
```

↳ Radius.

Predict

```
# y_dbs = dbs.fit_Predict (x)
# y-dbs

Out : array([-1, 0, -1, 0, -1, 0, -1, -1, 0, 0, -1
                  - - - - - - - - - - -
          - - - - - - - - - - - - ])
            - - - - - - - - - -
```

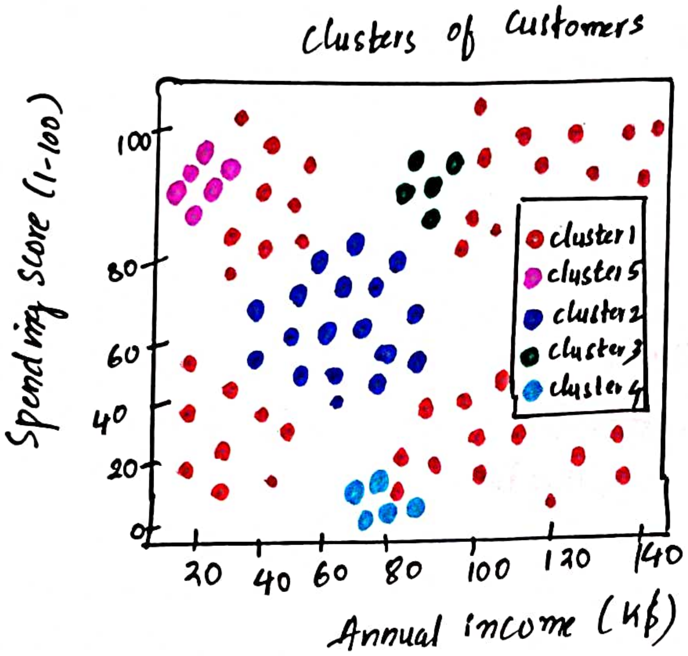
```
# np. unique (y-dps)

Out: array ([-1, 0, 1, 2, 3, 4]).
```

Visualising the clusters

```
   # cluster 1
# plt. scatter (x [y-dbs == -1, 0], x [y-dbs == -1, 1],
                    s=100, c = "red", label= "cluster 1")

   # cluster 5
# plt. scatter (x [y-dbs == 0, 0], x [y-dbs == 0, 1],
                    s = 100, c = "magenta", label= "cluster 5")

   # cluster 2
# plt. scatter (x [y- dbs == 1, 0], x [y-dbs == 1, 1],
                    s= 100, c = "blue", label= "cluster 2")

   # cluster 3
# plt. scatter (x [y-dbs == 2, 0], x [y-dbs == 2, 1],
                    s=100, c = "green", label = "cluster 3")

   # cluster 4
# plt. scatter (x [y-dbs == 3, 0], x [y-dbs == 3, 1],
                    s=100, c = "cyan", label = "cluster 4")
```

```
#plt.title("clusters of customers")
# plt.xlabel ("Annual income (K$ ")")
# plt.ylabel ("spending score (1-100)")
#plt.legend()
# plt.show()
```

Out]:



04/05/22
5:00pm.