DBSCAN

Density-based Spatial Clustering of Applications with Noise

A presentation by:
Shantan Hastalpuram
Sanketh Sreedharan Muthu
Prem Kumar Bodasingi

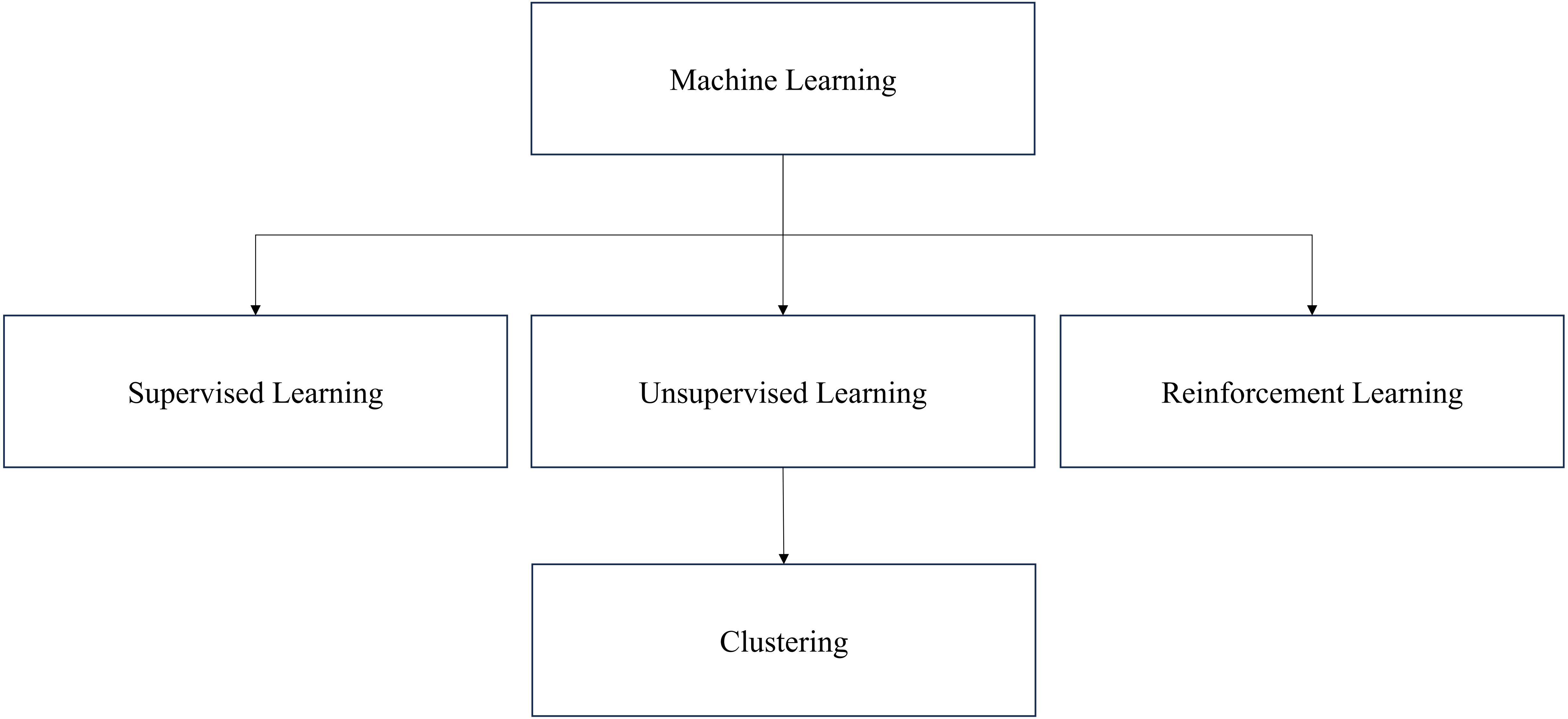
Instructor: Prof. Yasser Alginahi

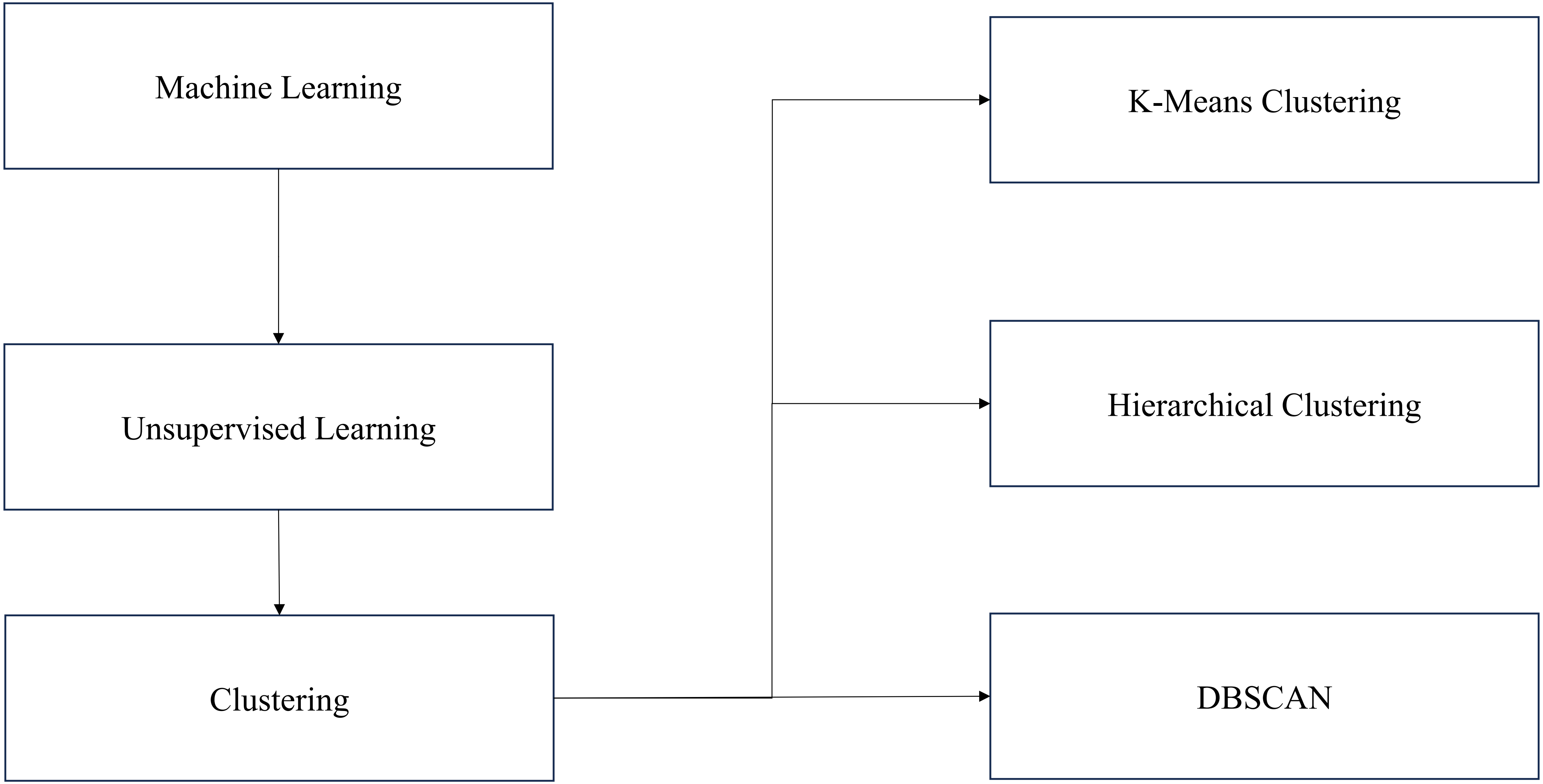
Date: 10-Nov-2023

University of Windsor

Agenda

- Introduction to clustering
- Terms in the abbreviation
- DBSCAN – Introduction and history
- Why do we need DBSCAN
- Further explanation
- Examples and use cases
- References

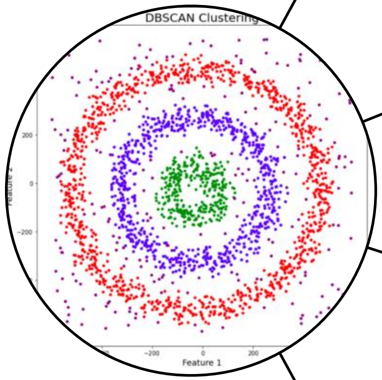

Machine Learning
Supervised Learning
Unsupervised Learning
Reinforcement Learning
Clustering


Machine Learning
Unsupervised Learning
Clustering
K-Means Clustering
Hierarchical Clustering
DBSCAN

Introduction to clustering

- Clustering – grouping unlabelled data into groups/clusters
- Example is identifying calls you receive and grouping them into important, casual, spam, marketing or scam.
- Clustering is done to know more about the data, analyze the data, and recognize patterns if any.

Terms in the abbreviation



Density
- The concentration of points in a region

Spatial
- More than 2-dimensional data

Clustering

Noise
- Data which does not fall into any clusters at first glance

DBSCAN – Introduction and history

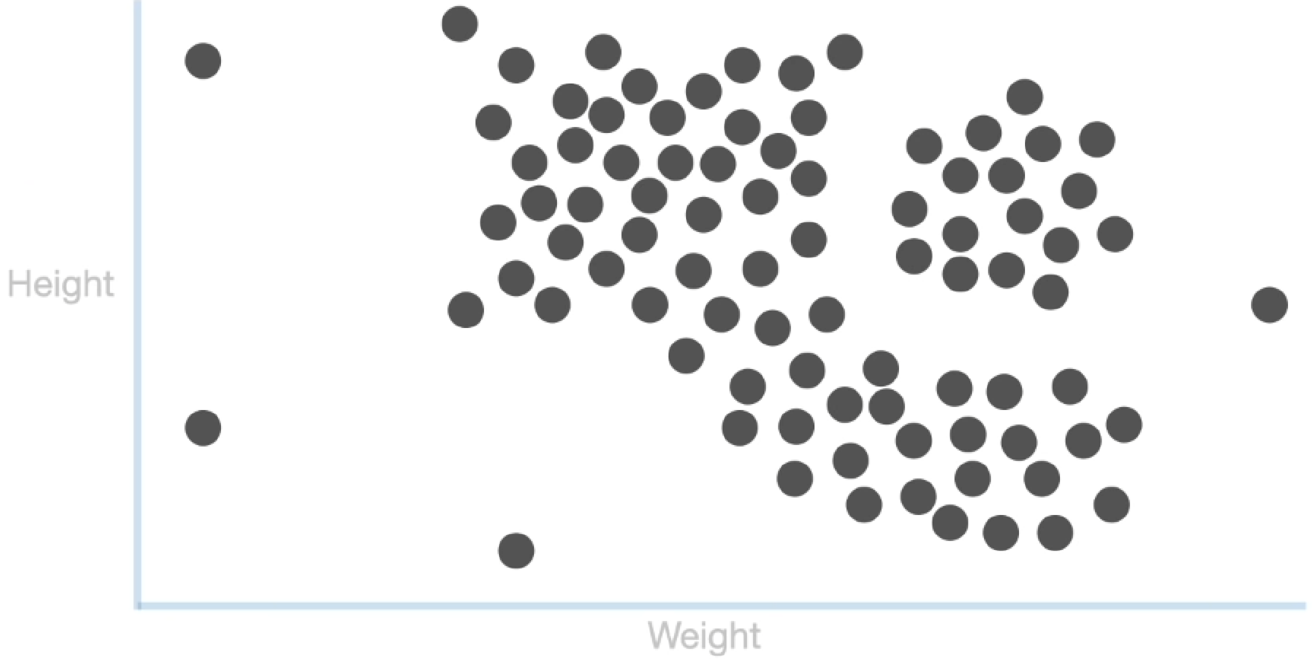



- proposed by Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu in 1996

[Source: https://tinyurl.com/2zs9n6p8.]

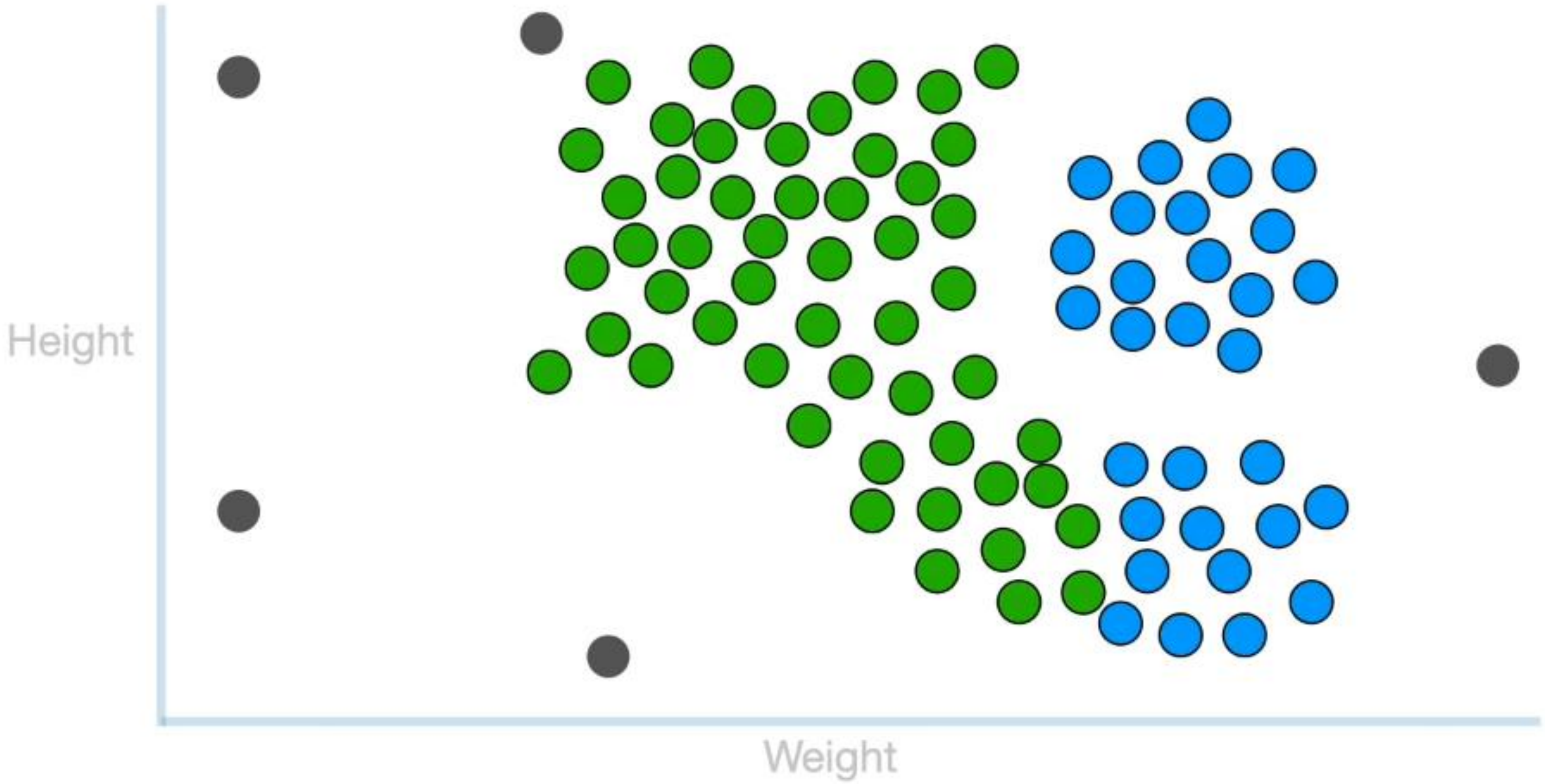
Why do we need DBSCAN

- DBSCAN – clusters just like a person can
- All examples considered for explanation are 2-dimensional



[Source:https://youtu.be/RDZUdRSDOok]

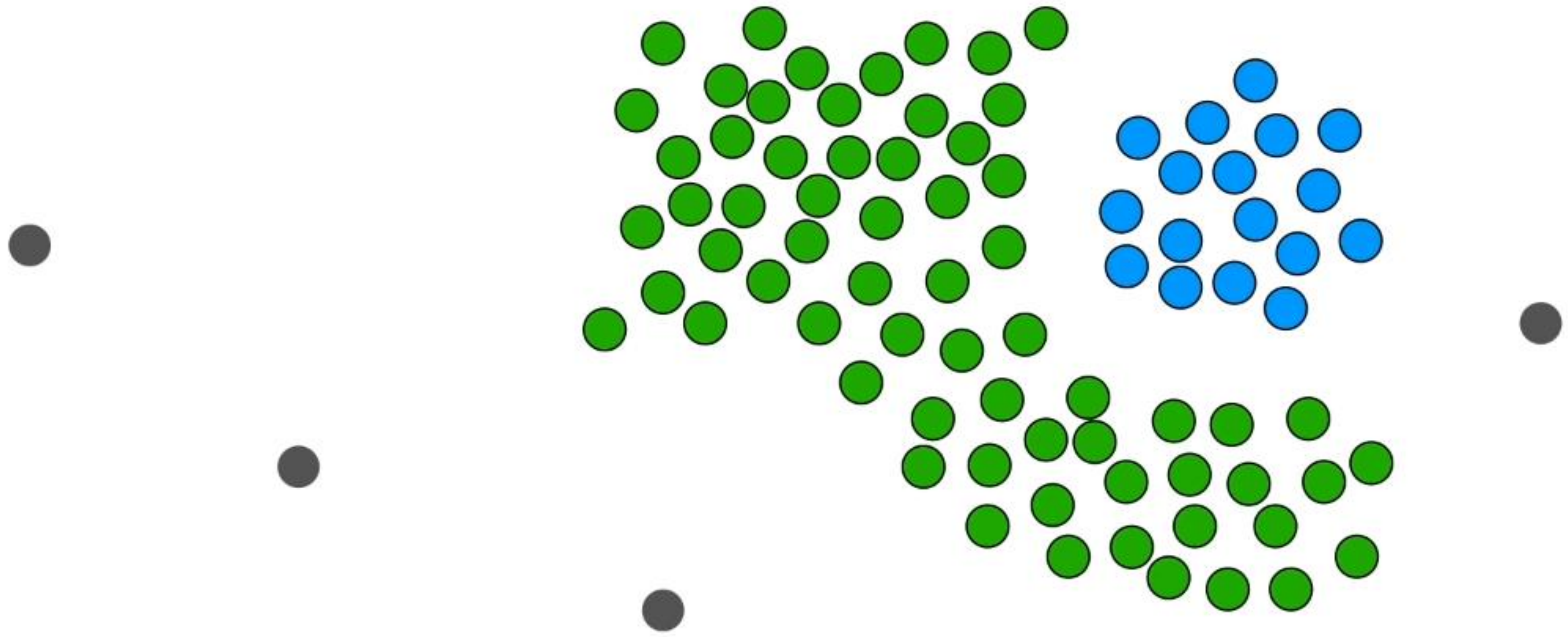
Why do we need DBSCAN



[Source:https://youtu.be/RDZUdRSDOok]

Why do we need DBSCAN

In addition, it is robust to outliers



 [Source:https://youtu.be/RDZUdRSDOok]

Reachability and Connectivity

- These are the two concepts you must comprehend before proceeding.

  Reachability

  Connectivity.

- Two points in DBSCAN can be referred to in terms of reachability and connectivity:

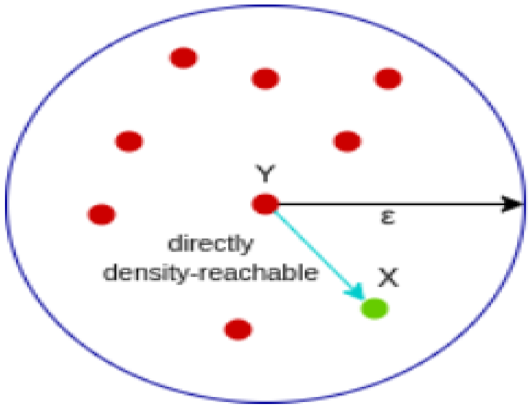
  Directly Density-Reachable

  Density-Reachable

  Density-Connected

Directly Density-Reachable

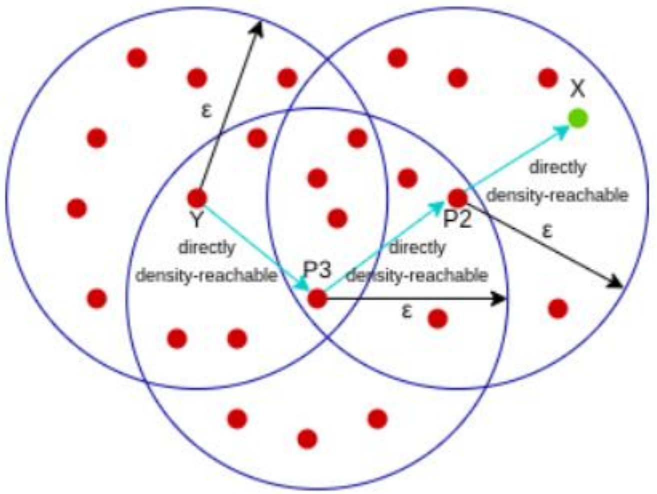
- A point X is directly density-reachable from point Y with repsect to epsilon, minPoints if.
    1. X belongs to the neighborhood of Y, i.e, dist(X, Y) <= epsilon
    2. Y is a core point



[Source:https://www.analyticsvidhya.com/blog/2020/09/how-dbscan-clustering-works/]

Density-Reachable

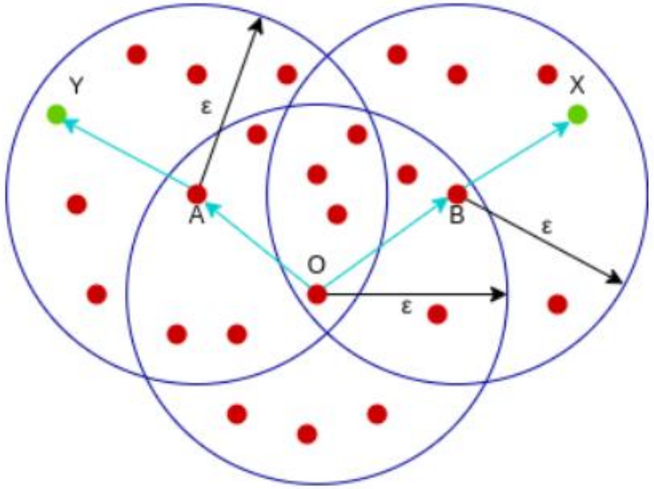
- A point X is density-reachable from point Y w.r.t epsilon, minPoints if there is a chain of points p1, p2, p3, …, pn and p1=X and pn=Y such that pi+1 is directly density-reachable from pi.



[Source:https://www.analyticsvidhya.com/blog/2020/09/how-dbscan-clustering-works/]

Density-Connectivity

- A point X is density-connected from point Y w.r.t epsilon and minPoints if there exists a point O such that both X and Y are density-reachable from O w.r.t to epsilon and minPoints.



[Source:https://www.analyticsvidhya.com/blog/2020/09/how-dbscan-clustering-works/]

Parameter Selection in DBSCAN Clustering

- DBSCAN is very sensitive to the values of epsilon and minPoints.
- The value of minPoints should be at least one greater than the number of dimensions of the dataset, i.e.,

$$\text{minPoints} >= \text{Dimensions} + 1.$$

- The value of epsilon can be decided from the K-distance graph. The point of maximum curvature (elbow) in this graph tells us about the value of epsilon.

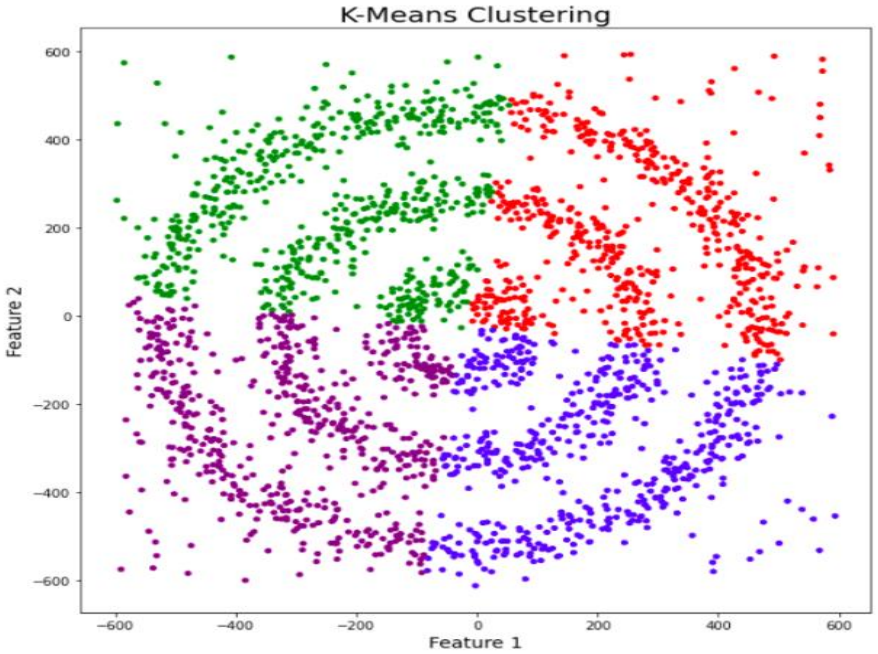
Clustering methods

- There are three types of clustering methods such as

  K Means Clustering
  Hierarchical Clustering
  DBSCAN Clustering

K Means Clustering Method

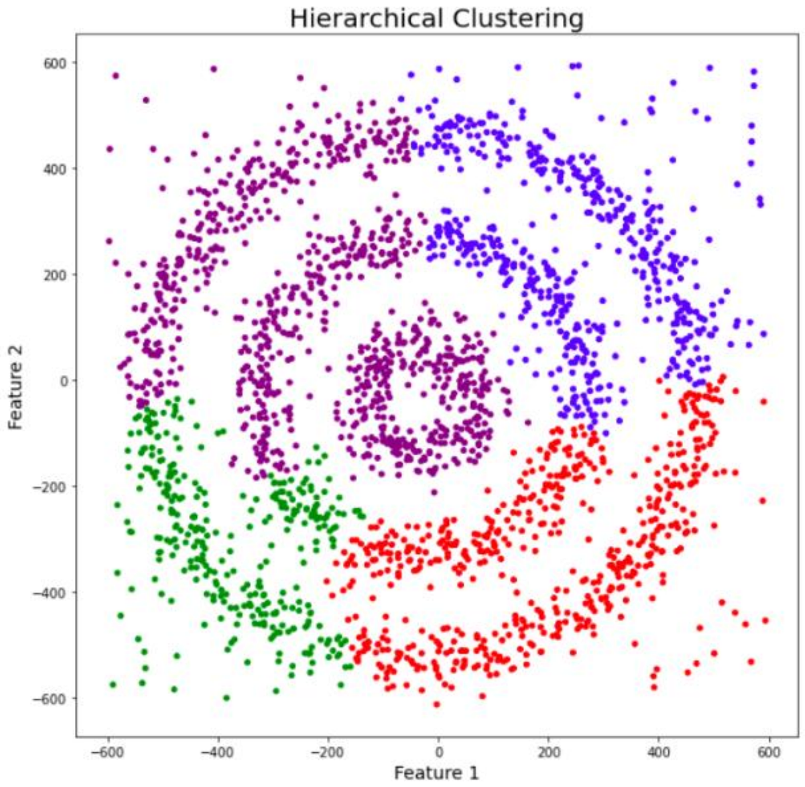
- K-means is a partitioning method that divides a dataset into K clusters.
- It minimizes the variance within each cluster, forming clusters around centroids.



[Source:https://www.analyticsvidhya.com/blog/2020/09/how-dbscan-clustering-works/]

Hierarchical Clustering Method

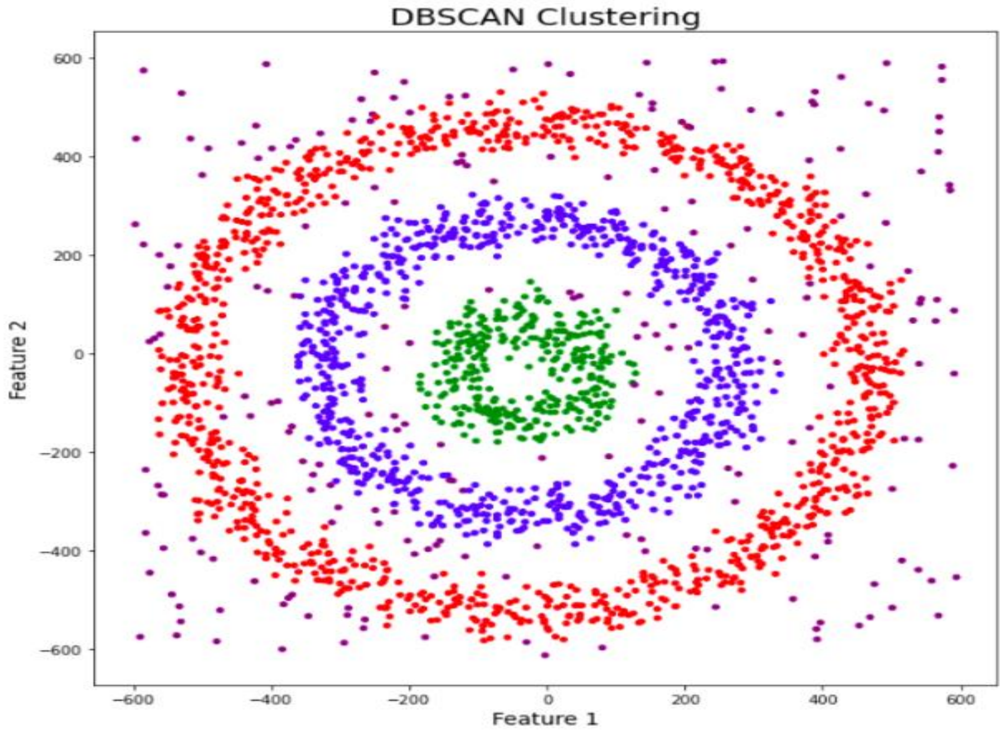
- Hierarchical clustering creates a tree of clusters (dendrogram), representing the relationships between data points.



[Source:https://www.analyticsvidhya.com/blog/2020/09/how-dbscan-clustering-works/]

DBSCAN Clustering Method

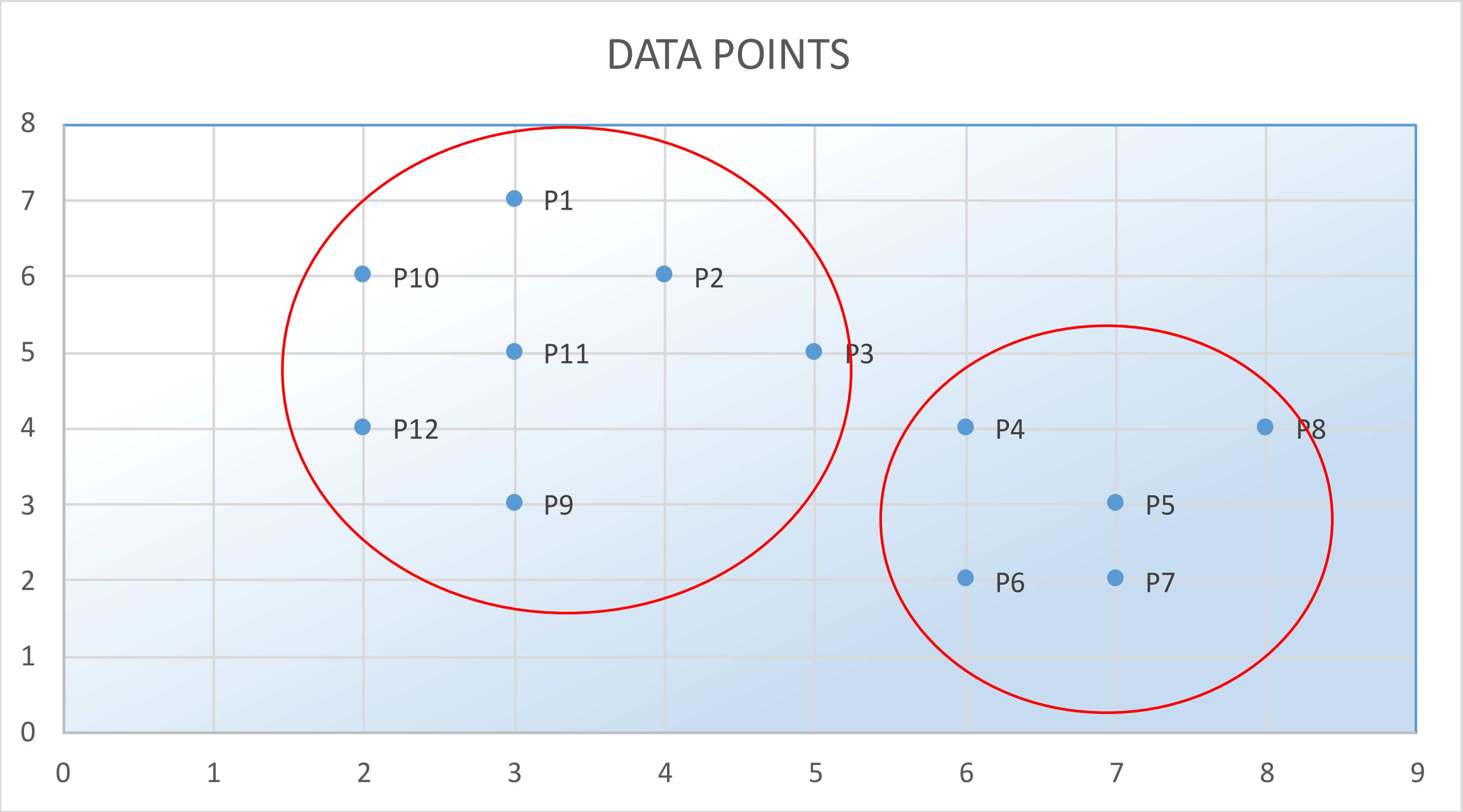
- DBSCAN identifies clusters based on the density of data points.
- Effective in discovering clusters of arbitrary shapes.
- Robust to noise and outliers.



[Source:https://www.analyticsvidhya.com/blog/2020/09/how-dbscan-clustering-works/]

Random unlabeled data

POINTS	X	Y
P1	**3**	**7**
P2	**4**	**6**
P3	**5**	**5**
P4	**6**	**4**
P5	**7**	**3**
P6	**6**	**2**
P7	**7**	**2**
P8	**8**	**4**
P9	**3**	**3**
P10	**2**	**6**
P11	**3**	**5**
P12	**2**	**4**



1. Select random data, shown above

2. Plotting in 2-d Cartesian plan

3. visualising the possible no. of clusters,

[Source:https://www.youtube.com/watch?v=-p354tQsKrs&ab_channel=MaheshHuddar]

Analysis of the random data

Min_Pts = **4**, Epsilon (radius)= **1.9**

Euclidean distance= $\sqrt{(x1-x2)^2+(y1-y2)^2}$

P1 : (3,7)
P2 : (4,6)
P3 : (5,5)
P4 : (6,4)
P5 : (7,3)
P6 : (6,2)
P7 : (7,2)
P8 : (8,4)
P9 : (3,3)
P10 : (2,6)
P11 : (3,5)
P12 : (2,4)

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
P1	0.00											
P2	1.41	0.00										
P3	2.83	1.41	0.00									
P4	4.24	2.83	1.41	0.00								
P5	5.66	4.24	2.83	1.41	0.00							
P6	5.83	4.47	3.16	2.00	1.41	0.00						
P7	6.40	5.00	3.61	2.24	1.00	1.00	0.00					
P8	5.83	4.47	3.16	2.00	1.41	2.83	2.24	0.00				
P9	4.00	3.16	2.83	3.16	4.00	3.16	4.12	5.10	0.00			
P10	1.41	2.00	3.16	4.47	5.83	5.66	6.40	6.32	3.16	0.00		
P11	2.00	1.41	2.00	3.16	4.47	4.24	5.00	5.10	2.00	1.41	0.00	
P12	3.16	2.83	3.16	4.00	5.10	4.47	5.39	6.00	1.41	2.00	1.41	0.00

Distance Matrix

P1	P2, P10
P2	P1,P3,P11
P3	P2,P4
P4	P3,P5
P5	P4,P6,P7,P8
P6	P5,P7
P7	P5,P6
P8	P5
P9	P12
P10	P1,P11
P11	P2,P10,P12
P12	P9,P11

4.Finding distance between each individual data point to remaining all data points, using distance formula, plotting in distance matrix

5. Taking each data point as core point finding the number of remaining data points fall in the core region of radius i.e., epsilon= 1.9

Note : To find the remaining points , Eg : consider core point say (**P5**) using distance matrix

1. From P5 check **horizontally** how many points are under the eplison value (>=1.9) i.e (**P4**)
2. From P5 check **vertically**, how many are falling under the eplison value (>=1.9) i.e (**P6,P7,P8**)

P5 as core point, P4,P6,P7,P8 are in core region

[Source:https://www.youtube.com/watch?v=-p354tQsKrs&ab_channel=MaheshHuddar]

Results of the analysis

Min_Pts = **4**, Epsilon (radius)= **1.9**

P1	P2, P10
P2	P1,P3,P11
P3	P2,P4
P4	P3,P5
P5	P4,P6,P7,P8
P6	P5,P7
P7	P5,P6
P8	P5
P9	P12
P10	P1,P11
P11	P2,P10,P12
P12	P9,P11

POINTS	STATUS	
P1	NOISE	BORDER
P2	**CORE**	
P3	NOISE	BORDER
P4	NOISE	BORDER
P5	**CORE**	
P6	NOISE	BORDER
P7	NOISE	BORDER
P8	NOISE	BORDER
P9	NOISE	
P10	NOISE	BORDER
P11	**CORE**	
P12	NOISE	BORDER

6. The region with minimum 4 data points is consider as core region , since assumed minimum points to be 4 point, P2, P5,P11 are forming core region ,the rest other considered as noise

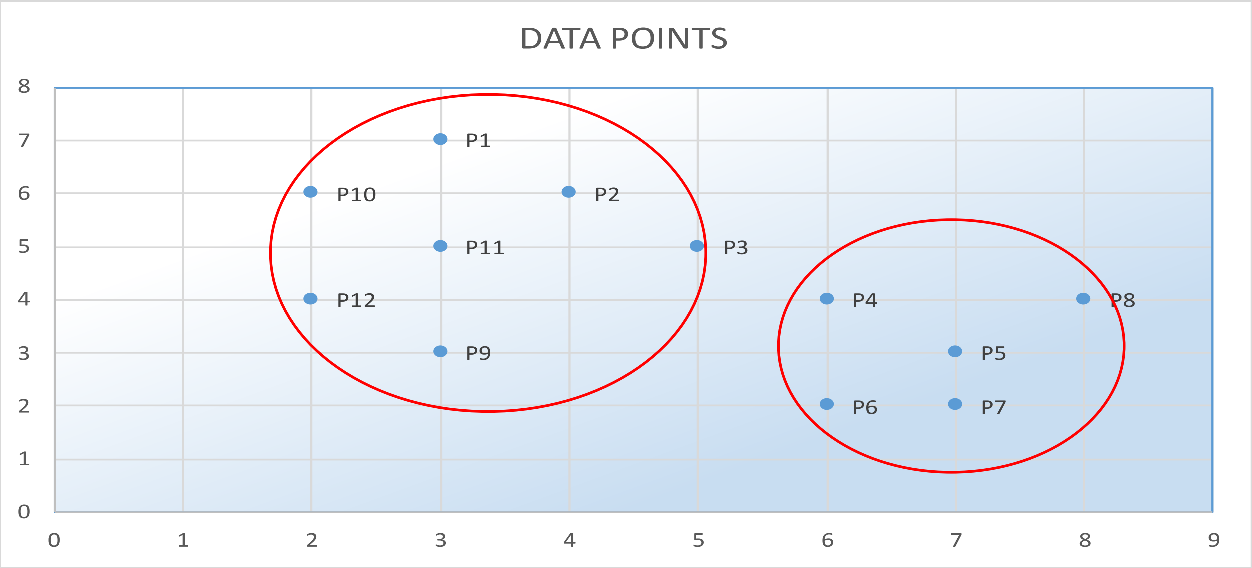
7. The noise can some time be border to core region , checking all the Nosie data points whether they are falling in any core region, P1 is falling in P2 core region , hence P1 is border Point, which can be used to form cluster, Similarly, P3, P4,P6,P7,P8,P10,P1 are border points

8. The noise points which are not falling in any core region, then it is consider as noise, P9 is noise.

[Source:https://www.youtube.com/watch?v=-p354tQsKrs&ab_channel=MaheshHuddar]

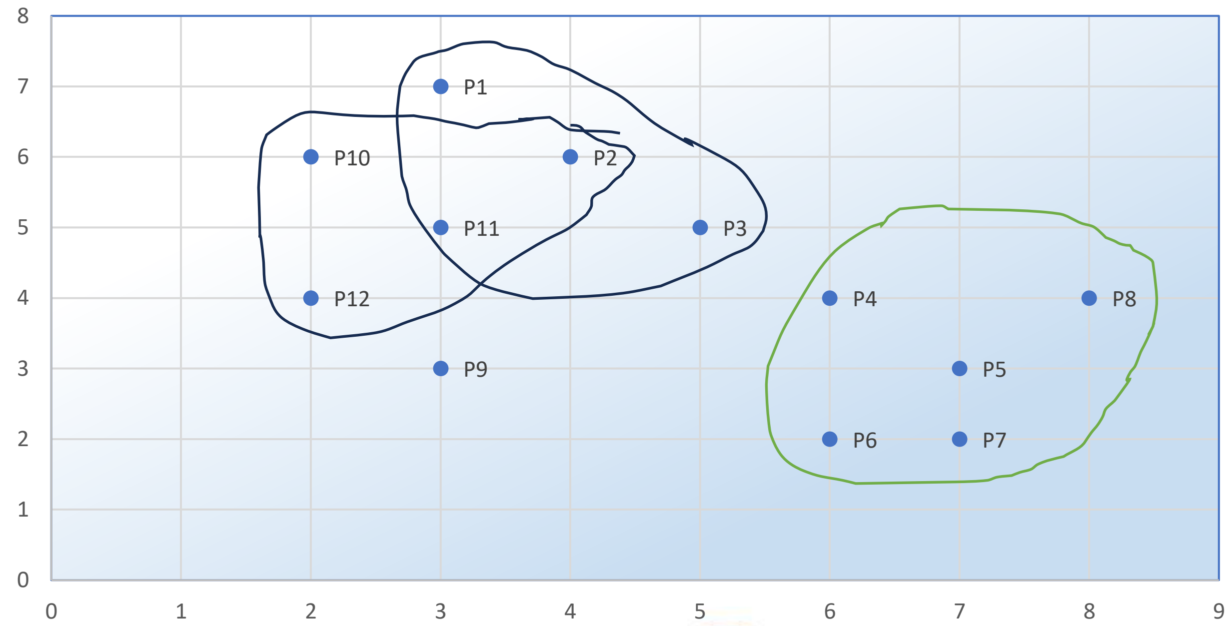
Results of the analysis

WITHOUT DBSCAN CLUSTERING



Points	Neighbours	Status	
P1	P2, P10	Noise	Border
P2	P1,P3,P11	Core	
P3	P2,P4	Noise	Border
P4	P3,P5	Noise	Border
P5	P4,P6,P7,P8	Core	
P6	P5,P7	Noise	Border
P7	P5,P6	Noise	Border
P8	P5	Noise	Border
P9	P12	Noise	
P10	P1,P11	Noise	Border
P11	P2,P10,P12	Core	
P12	P9,P11	Noise	Border

WITH DBSCAN CLUSTERING



[Source:https://www.youtube.com/watch?v=-p354tQsKrs&ab_channel=MaheshHuddar]

Algorithm : Comparing DCSCAN clustering with K-Means clustering and hierarchal clustering

```
# importing libraries
import numpy as np
import pandas as pd
import math
import matplotlib.pyplot as plt
import matplotlib
```

```
# Function for creating datapoints in the form of a circle
def PointsInCircum(r,n=100):
    return [(math.cos(2*math.pi/n*x)*r+np.random.normal(-30,30),math.sin(2*math.pi/n*x)*r+np.random.normal(-30,30)) for x in range(1,n+1)]
```

```
# Creating data points in the form of a circle
df=pd.DataFrame(PointsInCircum(500,1000))
df=df.append(PointsInCircum(300,700))
df=df.append(PointsInCircum(100,300))
```

```
# Adding noise to the dataset
df=df.append([(np.random.randint(-600,600),np.random.randint(-600,600)) for i in range(300)])
```

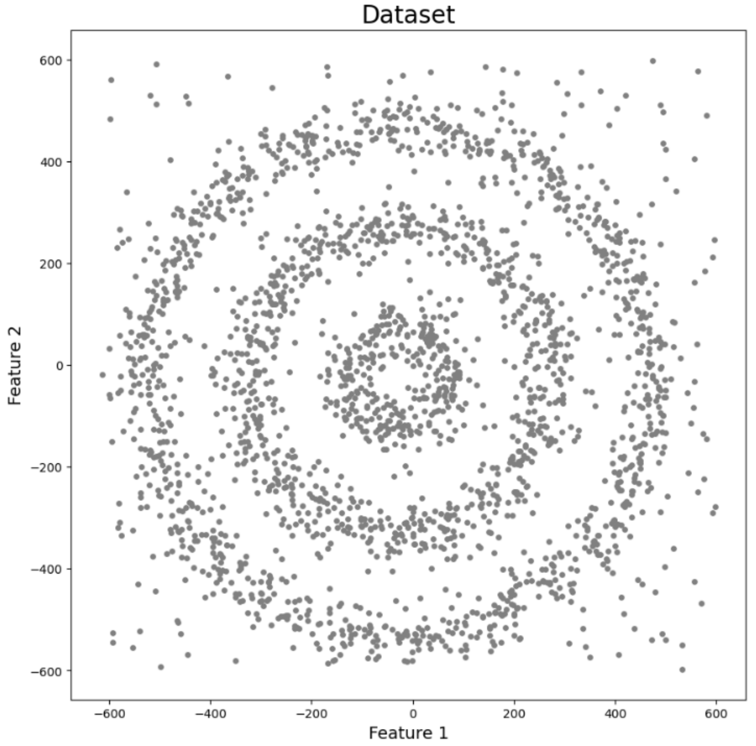

[Source:https://www.analyticsvidhya.com/blog/2020/09/how-dbscan-clustering-works/]

visualising dataset
df

```
df
```

	0	1
0	445.663733	-28.953897
1	486.210213	8.180645
2	460.869550	-21.729519
3	457.361959	-20.573038
4	510.353345	0.274173
...	...	...
295	-524.000000	-310.000000
296	408.000000	323.000000
297	-396.000000	324.000000
298	64.000000	437.000000

```
# plotting the data points
plt.figure(figsize=(10,10))
plt.scatter(df[0],df[1],s=15,color='grey')
plt.title('Dataset',fontsize=20)
plt.xlabel('Feature 1',fontsize=14)
plt.ylabel('Feature 2',fontsize=14)
plt.show()
```



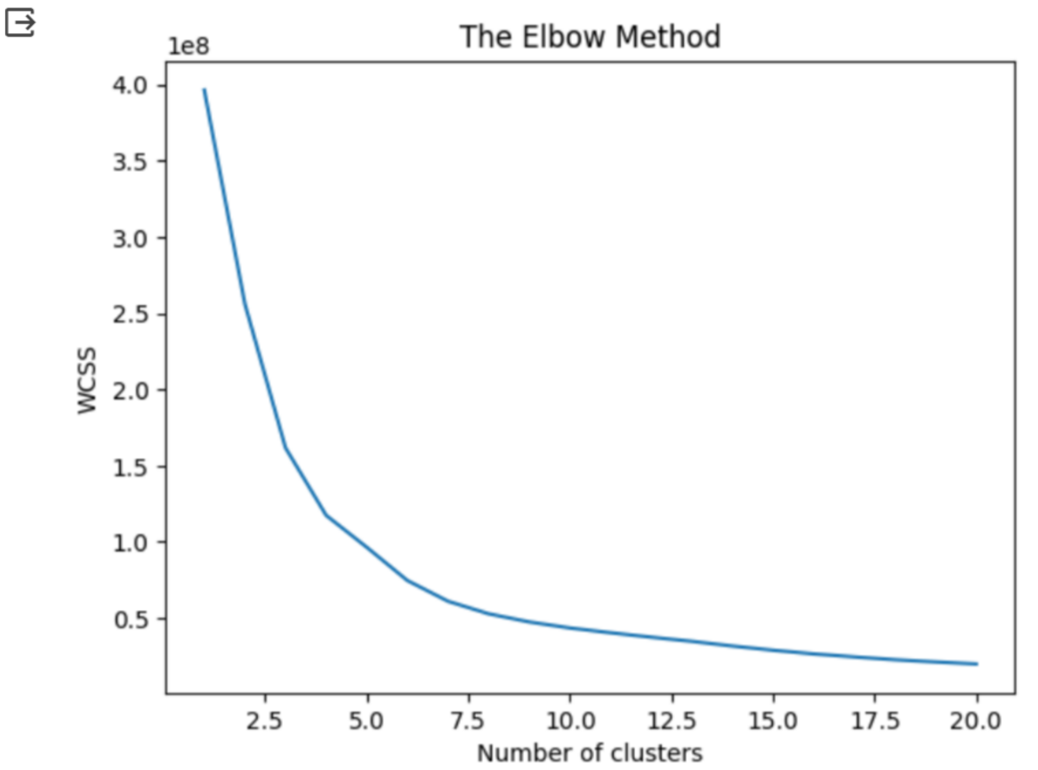
[Source:https://www.analyticsvidhya.com/blog/2020/09/how-dbscan-clustering-works/]

University of Windsor

Plotting **K-Means** clusters

Finding the number of optimal clustering for K-Mean using Elbow Method

```
from sklearn.cluster import KMeans
wcss = []
for i in range(1,21):
   kmeans = KMeans(n_clusters= i, init = 'k-means++', max_iter= 300, n_init= 10)
   kmeans.fit(df)
   wcss.append(kmeans.inertia_)
plt.plot(range(1,21), wcss)
plt.title("The Elbow Method")
plt.xlabel("Number of clusters")
plt.ylabel("WCSS")
plt.show()
```



From the elbow curve, the optimal clusters are 4

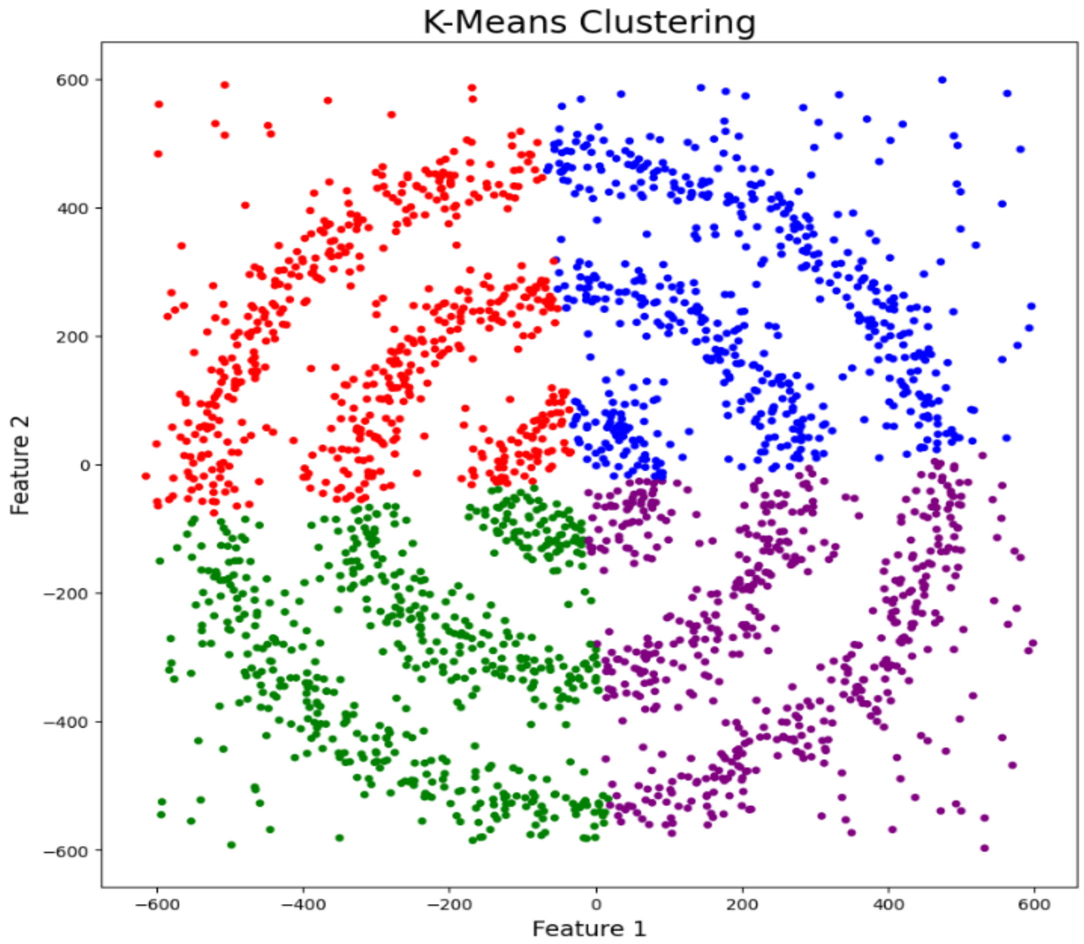
Importing KMeans from sklearn and Fitting the model to kmeans

```
from sklearn.cluster import KMeans
k_means=KMeans(n_clusters=4,random_state=42, n_init=10)
k_means.fit(df[[0,1]])

df['KMeans_labels']=k_means.labels_
```

[Source:https://www.analyticsvidhya.com/blog/2020/09/how-dbscan-clustering-works/]

```python
# Plotting resulting clusters
colors=['purple','red','blue','green']
plt.figure(figsize=(10,10))
plt.scatter(df[0],df[1],c=df['KMeans_labels'],cmap=matplotlib.colors.ListedColormap(colors),s=15)
plt.title('K-Means Clustering',fontsize=20)
plt.xlabel('Feature 1',fontsize=14)
plt.ylabel('Feature 2',fontsize=14)
plt.show()
```



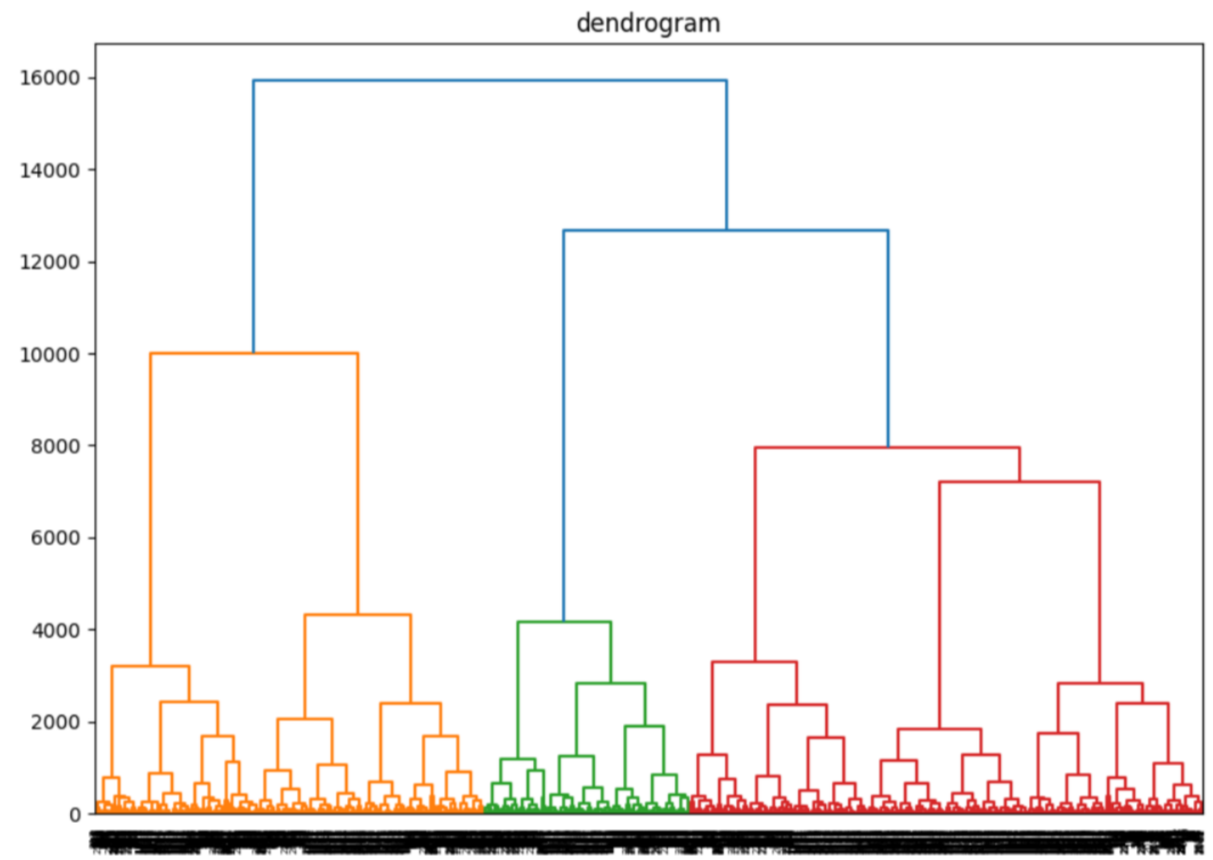
[Source:https://www.analyticsvidhya.com/blog/2020/09/how-dbscan-clustering-works/]

Plotting **Hierarchical** clusters

Finding the number of optimal clustering for hierarchical using Dendrogram

```
import scipy.cluster.hierarchy as sch
plt.figure(figsize=(10,7))

dendrogram =sch.dendrogram(sch.linkage(df,method="ward"))
plt.title("dendrogram")
plt.show()
```



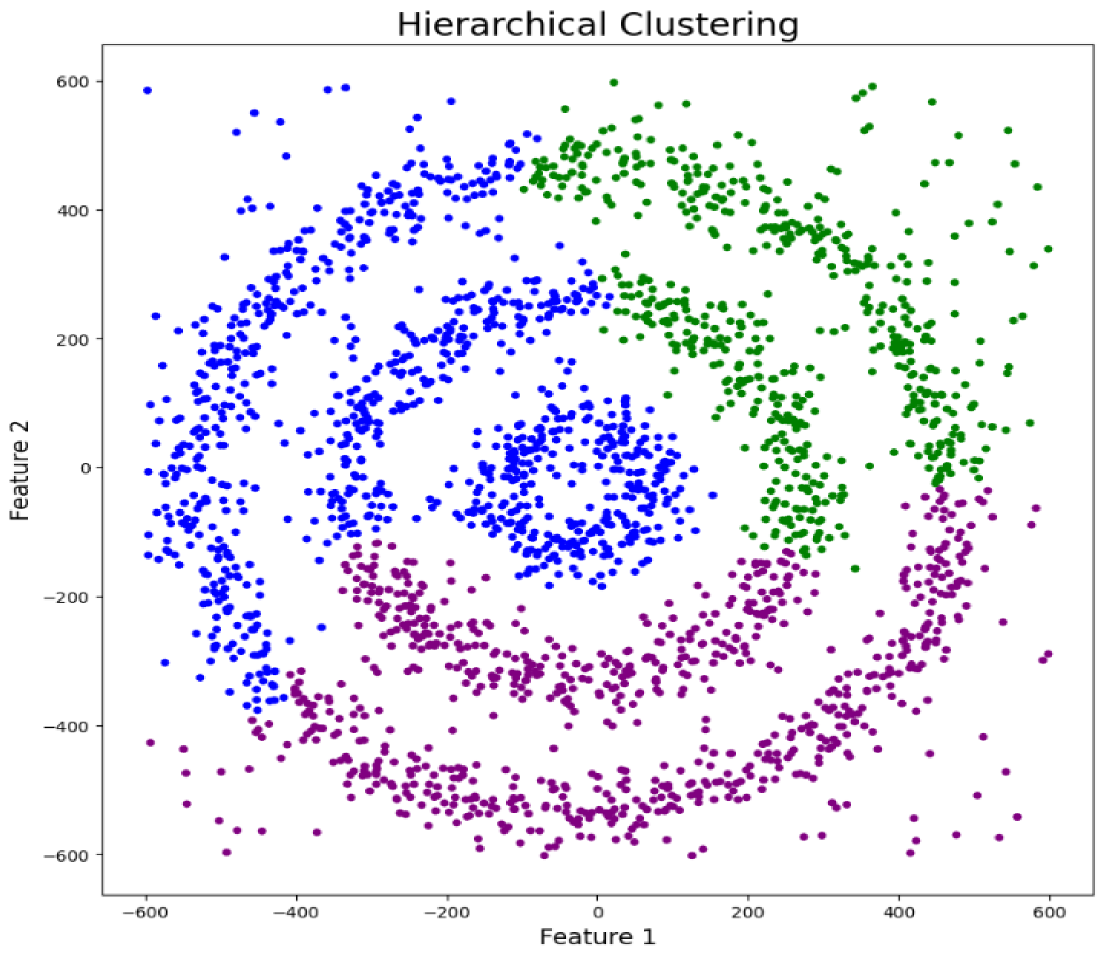
From the dendrogram, the optimal clusters are 3

Importing Agglomerative clustering from sklearn and Fitting the model

```
from sklearn.cluster import AgglomerativeClustering
model = AgglomerativeClustering(n_clusters=3, affinity='euclidean')
model.fit(df[[0,1]])
df['HR_labels']=model.labels_
```

[Source:https://www.analyticsvidhya.com/blog/2020/09/how-dbscan-clustering-works/]

```python
# Plotting resulting clusters
plt.figure(figsize=(10,10))
plt.scatter(df[0],df[1],c=df['HR_labels'],cmap=matplotlib.colors.ListedColormap(colors),s=15)
plt.title('Hierarchical Clustering',fontsize=20)
plt.xlabel('Feature 1',fontsize=14)
plt.ylabel('Feature 2',fontsize=14)
plt.show()
```



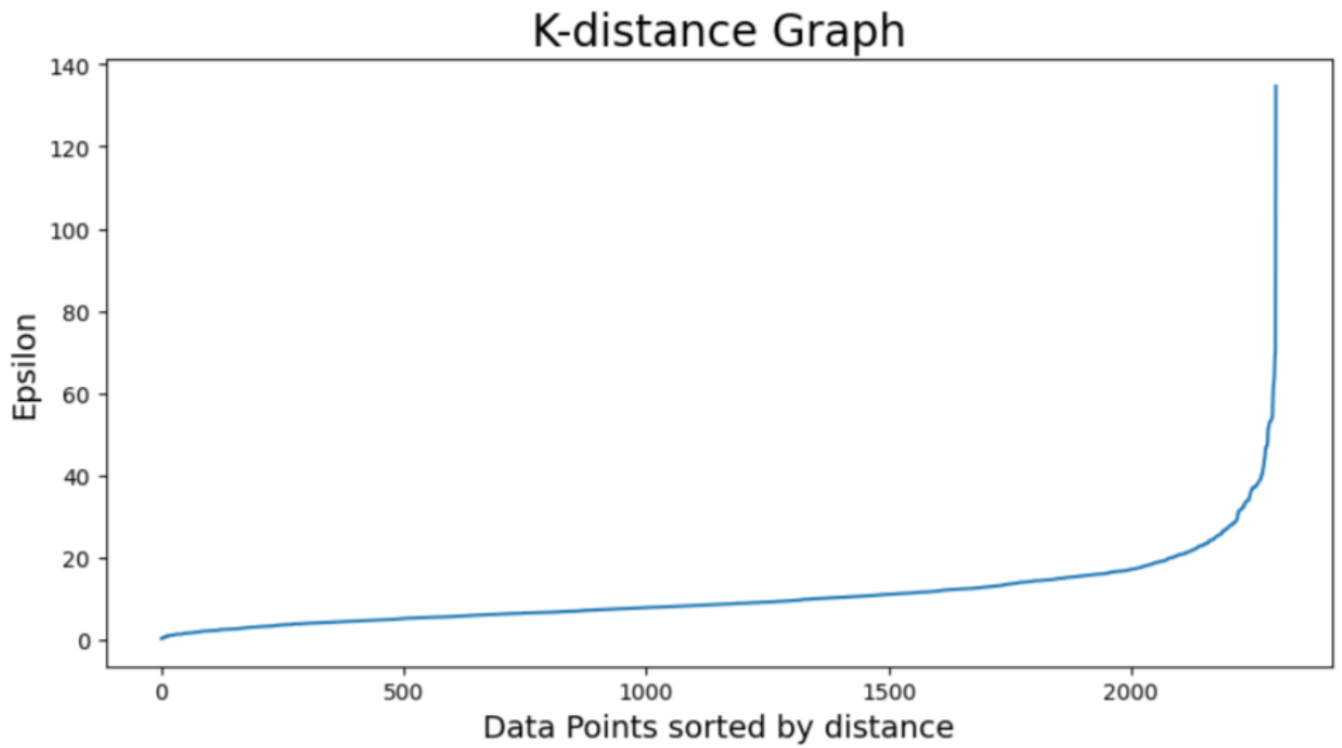
[Source:https://www.analyticsvidhya.com/blog/2020/09/how-dbscan-clustering-works/]

Plotting **DBSCAN** clusters

Finding the value of EPSILON using K-distance graph

```
# Plotting K-distance Graph
from sklearn.neighbors import NearestNeighbors
neigh = NearestNeighbors(n_neighbors=2)
nbrs = neigh.fit(df[[0,1]])
distances, indices = nbrs.kneighbors(df[[0,1]])

distances = np.sort(distances, axis=0)
distances = distances[:,1]
plt.figure(figsize=(10,5))
plt.plot(distances)
plt.title('K-distance Graph',fontsize=20)
plt.xlabel('Data Points sorted by distance',fontsize=14)
plt.ylabel('Epsilon',fontsize=14)
plt.show()
```



Importing DBSCAN from SKlearn And Fitting the model

```
from sklearn.cluster import DBSCAN
dbscan_opt=DBSCAN(eps=35,min_samples=5)
dbscan_opt.fit(df[[0,1]])
```

The Optimal EPSILON value from the graph is 35

And Considering minimum points/samples =5

[Source:https://www.analyticsvidhya.com/blog/2020/09/how-dbscan-clustering-works/]

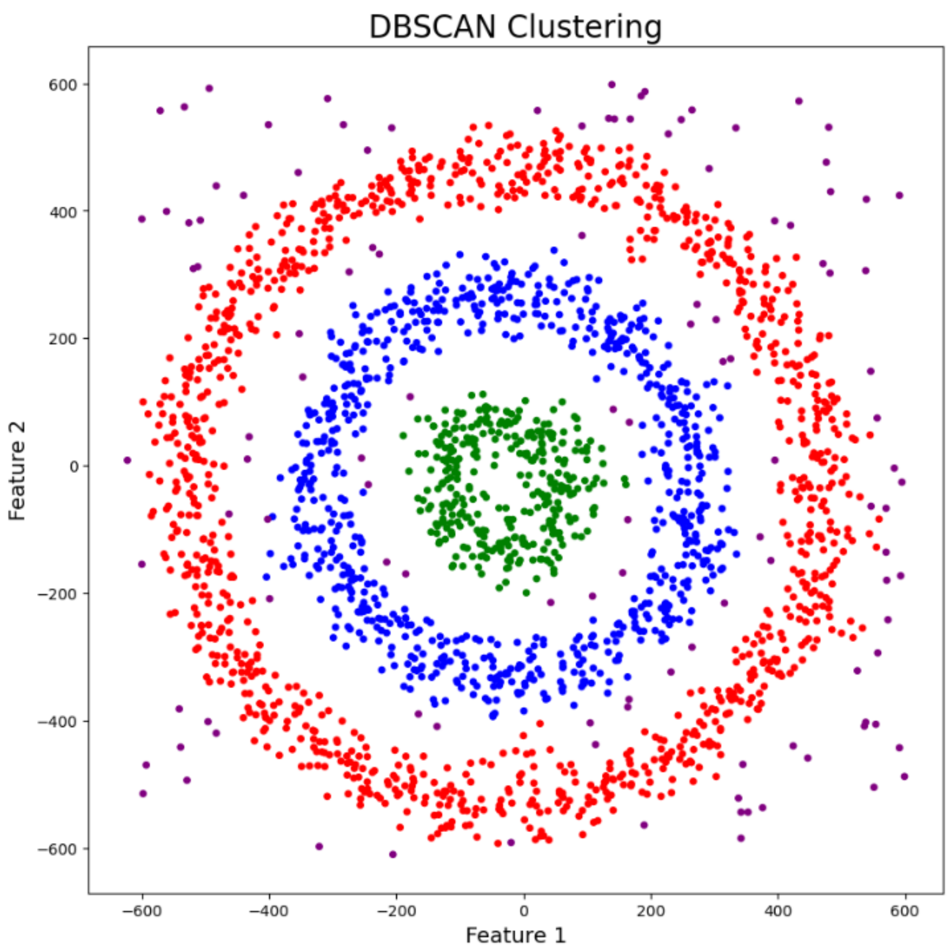
Identifying the cluster using labels count

```
df['DBSCAN_opt_labels']=dbscan_opt.labels_
df['DBSCAN_opt_labels'].value_counts()
```

```
 0    1096
 1     760
 2     325
-1     119
Name: DBSCAN_opt_labels, dtype: int64
```

clusters = labels count = 3

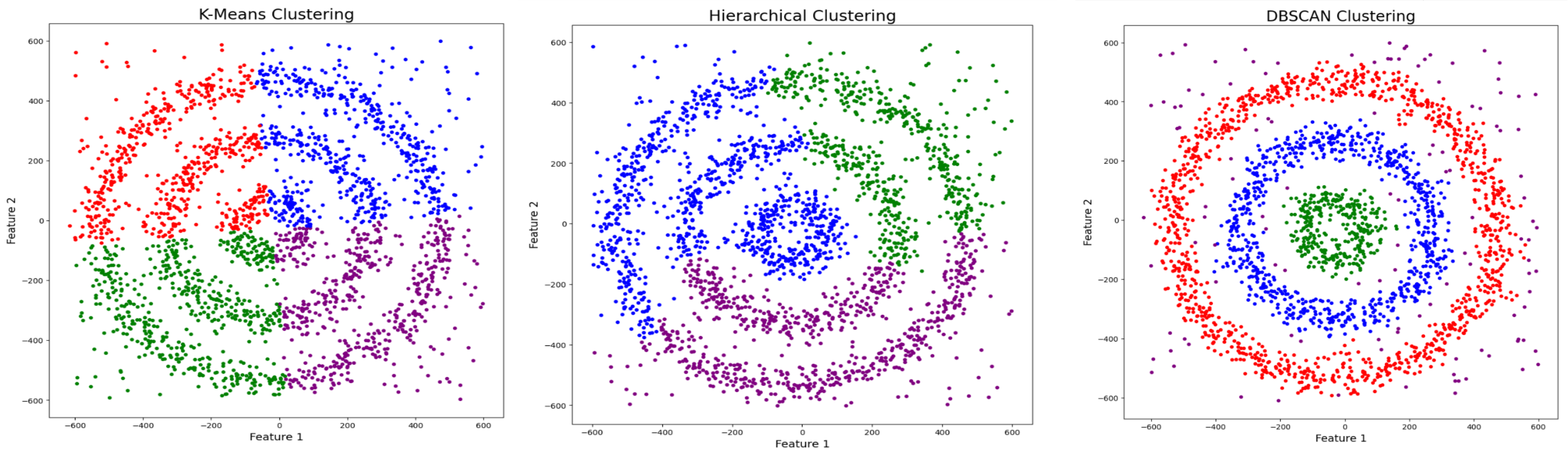
Note : Label = -1 indicates Noise




```
# Plotting the resulting clusters
plt.figure(figsize=(10,10))
plt.scatter(df[0],df[1],c=df['DBSCAN_opt_labels'],cmap=matplotlib.colors.ListedColormap(colors),s=15)
plt.title('DBSCAN Clustering',fontsize=20)
plt.xlabel('Feature 1',fontsize=14)
plt.ylabel('Feature 2',fontsize=14)
plt.show()
```


[Source:https://www.analyticsvidhya.com/blog/2020/09/how-dbscan-clustering-works/]

Comparison of all clustering methods:



[Source:https://www.analyticsvidhya.com/blog/2020/09/how-dbscan-clustering-works/]

Thank you!

Hope you have a sweet day (a bloody sweet day)

References

[1] S. Rajbanshi, “Everything you need to know about Machine Learning,” *Analytics Vidhya*, Mar. 25, 2021. https://tinyurl.com/czks8dn3 (accessed Nov. 10, 2023).

[2] “Clustering with DBSCAN, Clearly Explained!!!,” *www.youtube.com*. https://youtu.be/RDZUdRSDOok (accessed Aug. 14, 2023).

[3] A. Sharma, “How to Master the Popular DBSCAN Clustering Algorithm for Machine Learning,” *Analytics Vidhya*, Sep. 07, 2020. https://tinyurl.com/46my4tcf (accessed Nov. 10, 2023).

[4] “DBSCAN Clustering Algorithm Solved Numerical Example in Machine Learning Data Mining Mahesh Huddar,” *www.youtube.com*. https://tinyurl.com/h7f948b6 (accessed Nov. 10, 2023).

[5] “DBSCAN Clustering Algorithm Explained Simply,” *www.youtube.com*. https://youtu.be/Lh2pAkNNX1g (accessed Nov. 10, 2023).

[6]“Martin Ester,” *Wikipedia*, Jun. 22, 2023. https://en.wikipedia.org/wiki/Martin_Ester (accessed Nov. 10, 2023).

[7] “Hans-Peter Kriegel,” *Wikipedia*, Jan. 30, 2023. https://en.wikipedia.org/wiki/Hans-Peter_Kriegel (accessed Nov. 10, 2023).

[8] “Joerg Sander, PhD - Directory@UAlberta,” *apps.ualberta.ca*. https://apps.ualberta.ca/directory/person/jsander (accessed Nov. 10, 2023).