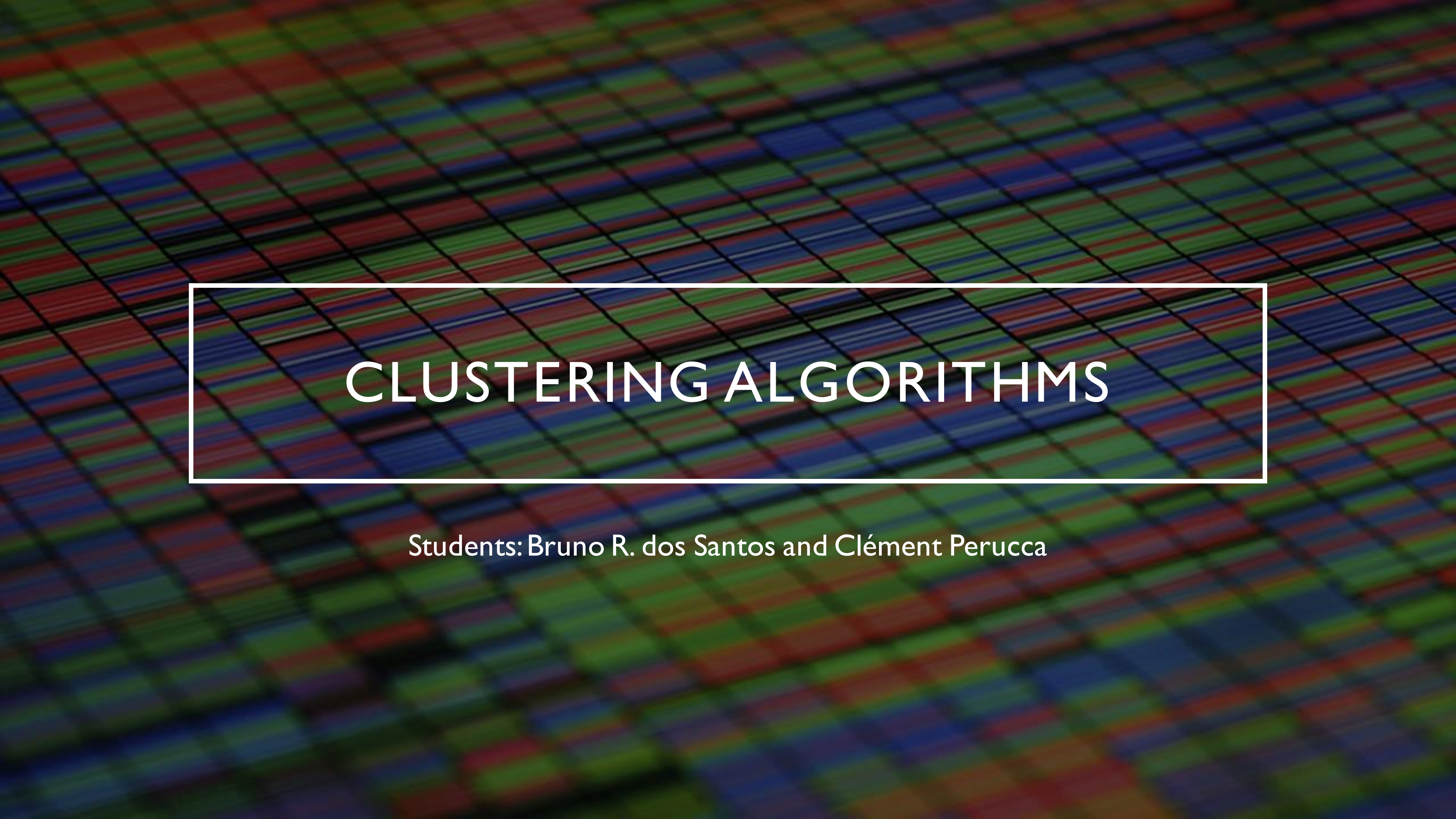
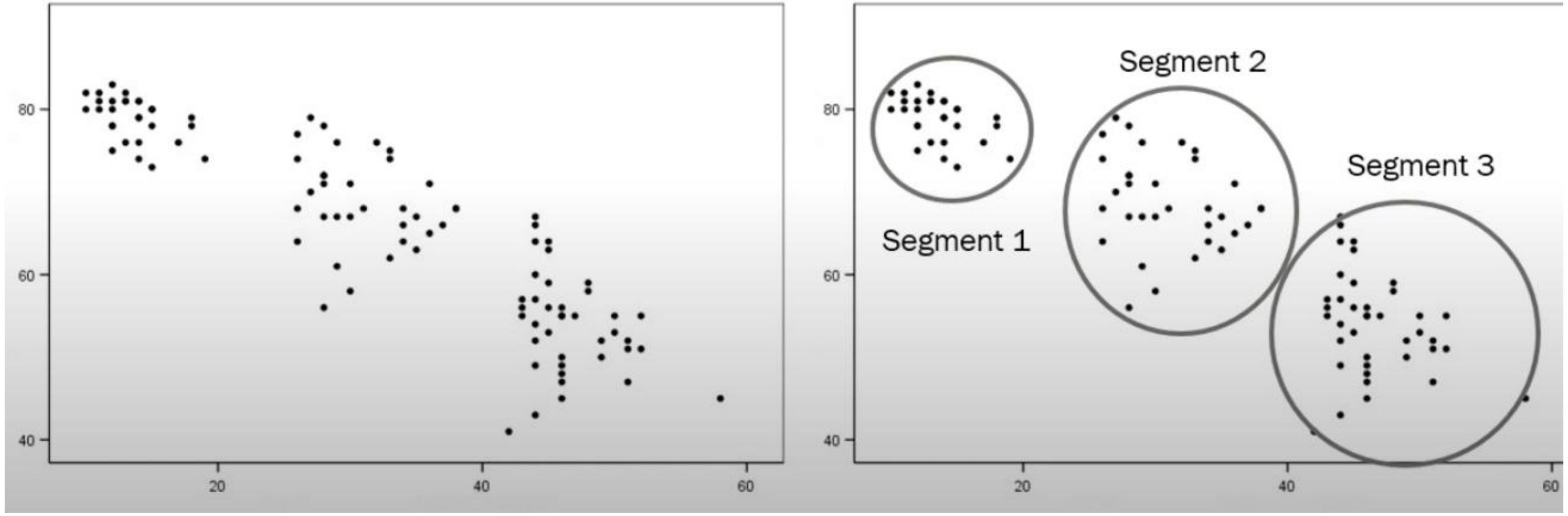


CLUSTERING ALGORITHMS

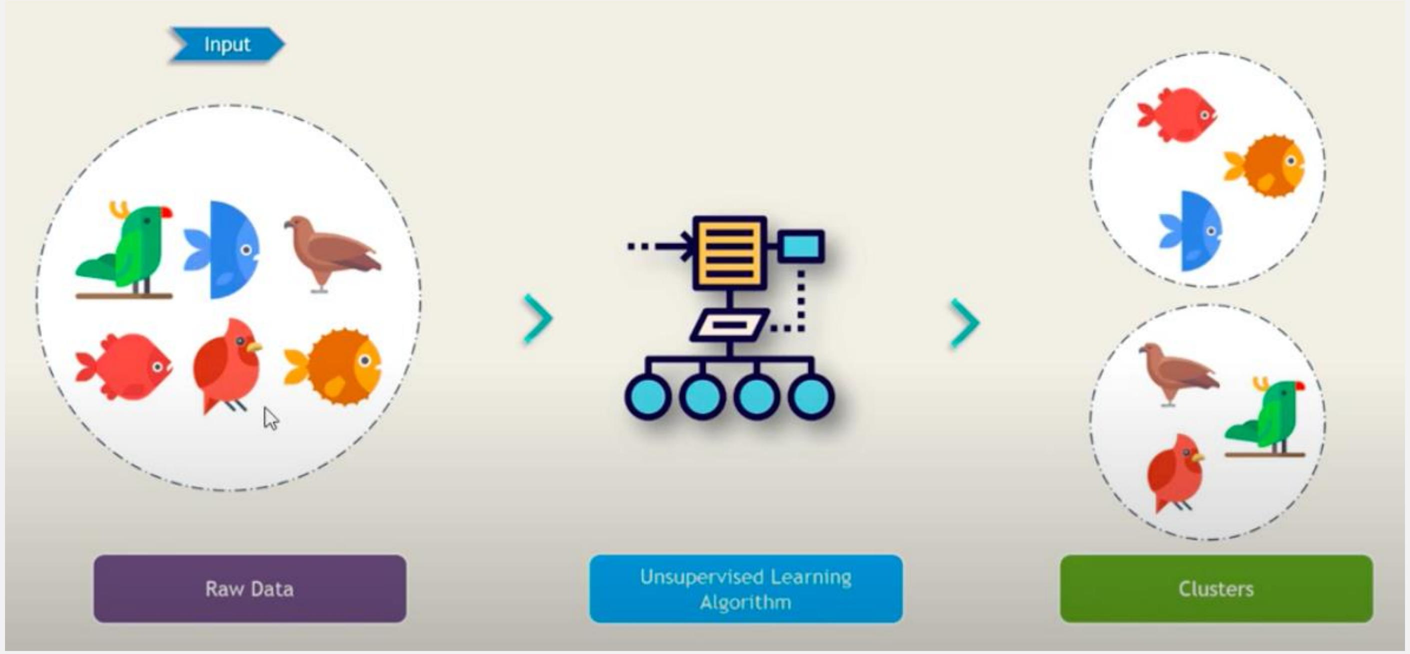
Students: Bruno R. dos Santos and Clément Perucca

WHAT IS CLUSTERING ?



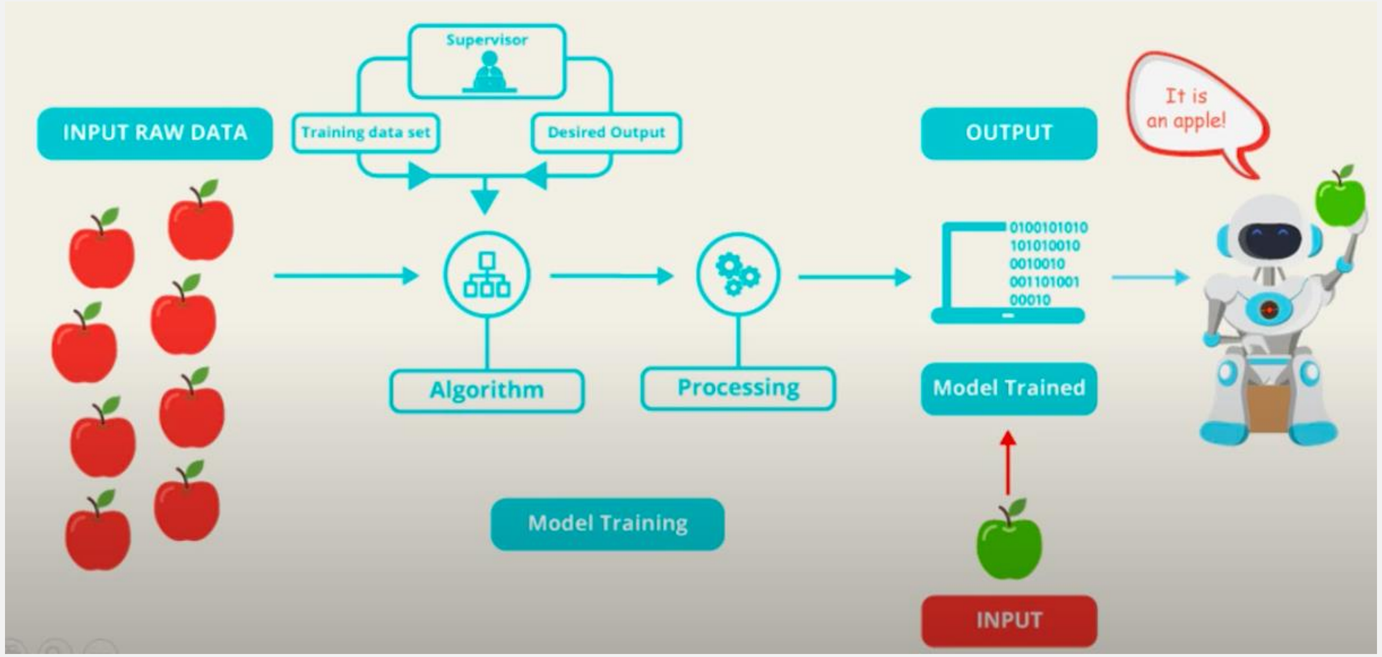
CLUSTERING OR CLASSIFICATION ?

CLUSTERING



CLUSTERING OR CLASSIFICATION ?

CLASSIFICATION



HOW IT'S WORKS ?

1. Specify the number of clusters K
2. Initialise the centroids
3. Continue iterating

    - Calculate the sum of the square of the distance between the data points and all centroids.

    - Assign each data point to the next closest cluster (centroid).

    - Calculate the centroids of the clusters by taking the average of all data points that belong to each cluster.

K MEANS - CODE

OTPA0011/kMeans (Class).py at main · bruniculos08/OTPA0011 (github.com)

DB SCAN

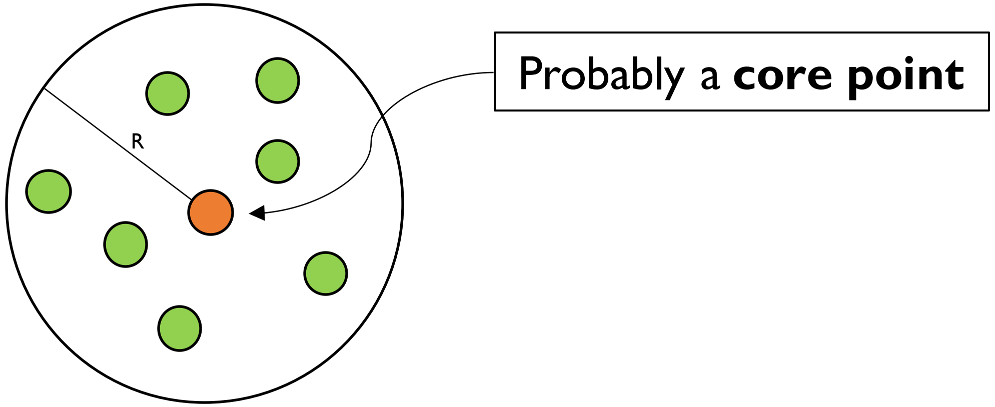
- **Density-based spatial clustering of applications with noise;**

- **Parameters:** a ratio **R** and a minimum group size **k**;

- To implement the algorithm, first, it is good to have in mind the concepts of **core points**, **border points** and **noise points**.

DB SCAN – CORE POINT

- A **core point** is a point that has at least **k** neighbors in a distance **R**;
- A **core point** always belong to a group of points (basically it is not a **noise point**);

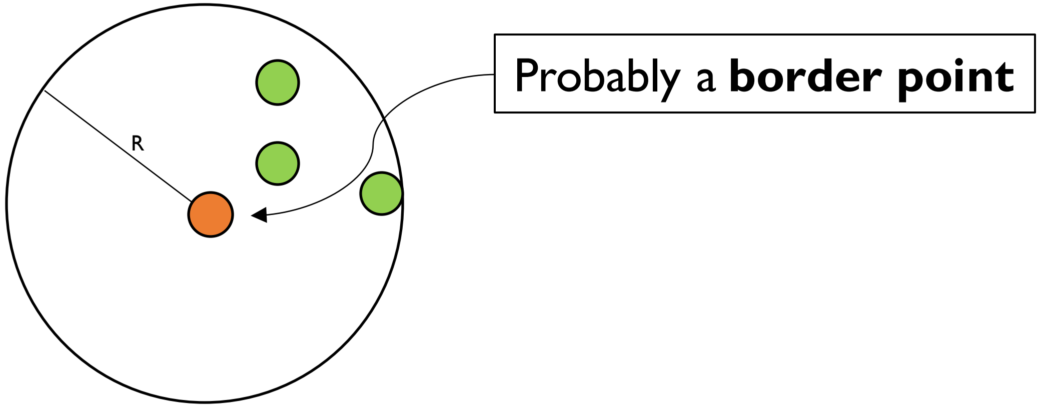


Probably a **core point**

DB SCAN – BORDER POINT

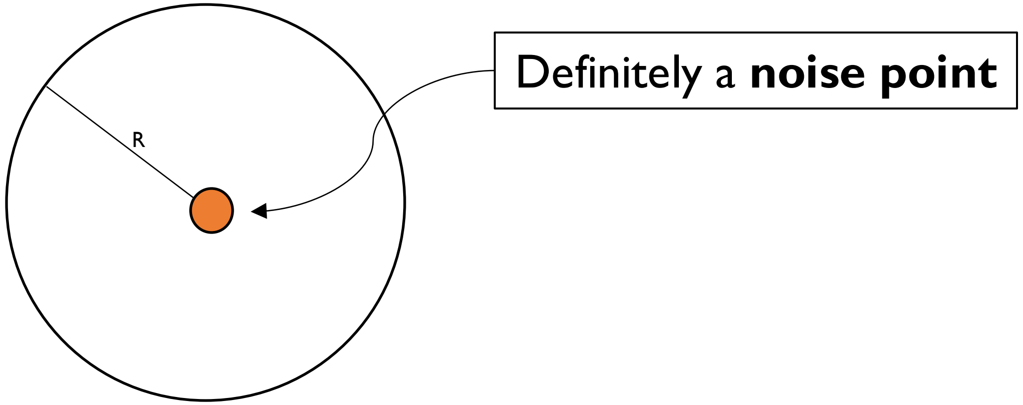
- A **border point** is a point that has fewer than **k** neighbors in a distance **R,** but this point has at least one **core point** as neighboor;

- A **border points** belongs to the cluster of his **core point** neighbors;



DB SCAN – NOISE POINT

- A **noise point** is a point that has fewer than **k** neighbors in a distance **R,** and none of these points is **core point**;



DB SCAN – PSEUDOCODE

```
DB-Scan(data, R, k):
1. Create a list of clusters
2. Chose any point $\beta$ from data that has not been selected yet
3. Create a cluster, set $\beta$ as selected and add to this cluster the selected point $\beta$ and the return of DB-ScanBuild(data, $\beta$, R, k)
4. Add the result cluster to the list of clusters
5. If there still exists unselected points in the data list, go back to step 2
```

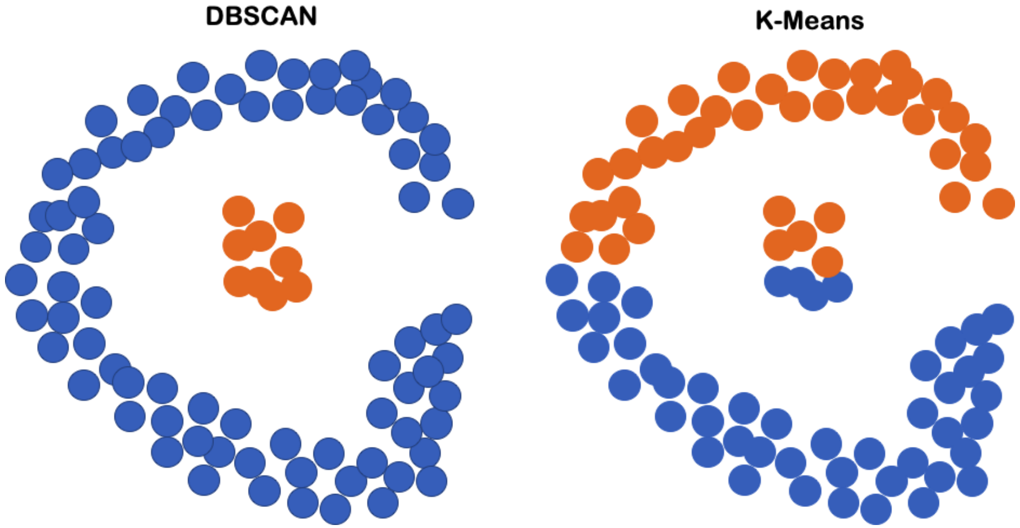
DB SCAN – PSEUDOCODE

```
DB-ScanBuild(data, $\beta$, R, k):
1. Create a empty list for the neighbors of $\beta$
2. For each poin $\mu$ in data, if $\mu$ is in a distance R from $\beta$ than put this point in the neighbors list
3. If the length of the list is less then k, return an empty list, else go to step 4
4. Create a new cluster and add to this cluster the result of DBScanBuild(data, $\mu$, R, k) for each point $\mu$ in the list
5. Return the new cluster concatenated with the neighbors list
```

DB SCAN – CODE

OTPA0011/DB-Scan (Class).py at main · bruniculos08/OTPA0011 (github.com)

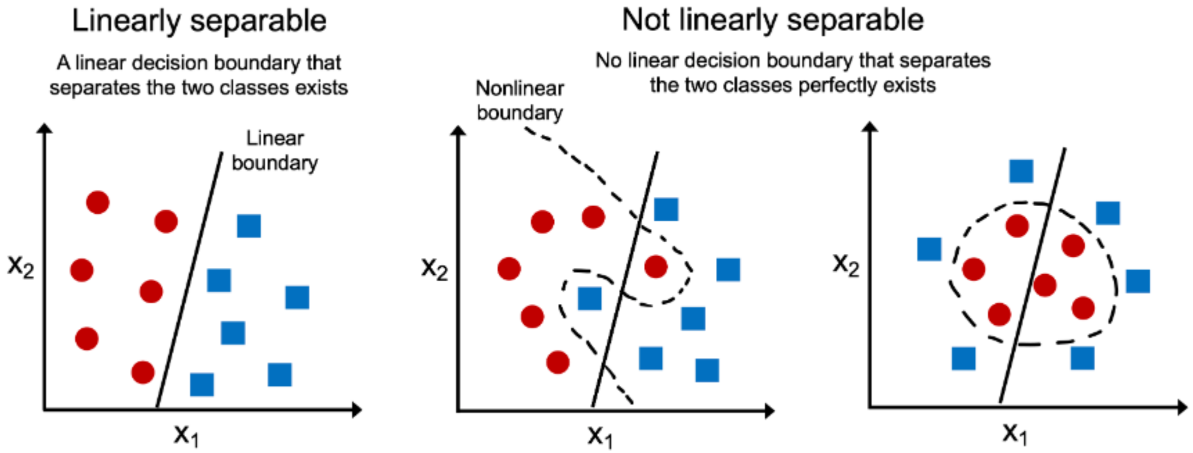
COMPARING



Source: Yufeng, 2022.

COMPARING

- K-Means is clustering algorithm used for linear datasets and DB-Scan is used for non-linear datasets.


Linearly separable
A linear decision boundary that separates the two classes exists
Linear boundary
x_2
x_1
Not linearly separable
No linear decision boundary that separates the two classes perfectly exists
Nonlinear boundary
x_2
x_1
x_2
x_1

Source: Kumar, 2022.

REFERENCES

- MUND, Kumar Shritam; How does DBSCAN clustering algorithm works? 26 de julho de 2019. Disponível em: <https://shritam.medium.com/how-dbscan-algorithm-works-2b5bef80fb3>. Acesso em 3 de dezembro de 2022.
- KUMAR, Ajitesh; Linear vs Non-Linear Data: How to Know? 31 de julho de 2022. Disponível em: <https://vitalflux.com/how-know-data-linear-non-linear/>. Acesso em 5 de dezembro de 2022.
- YUFENG; Understanding DB-Scan and Implementation with Python. 22 de janeiro de 2022. Disponível em: <https://towardsdatascience.com/understanding-dbscan-and-implementation-with-python-5de75a786f9f/>. Acesso em 3 de dezembro de 2022.