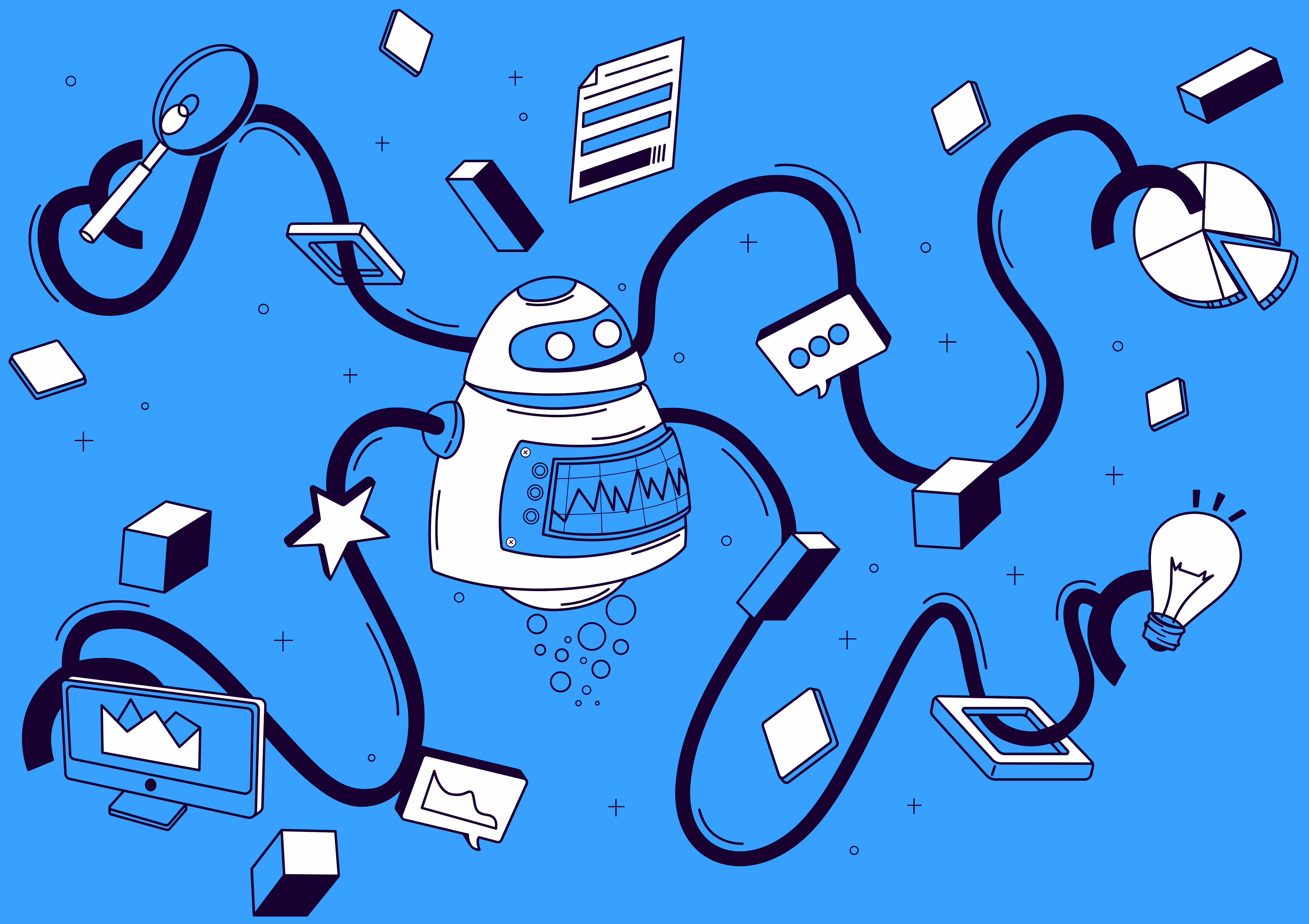
CLUSTERING

Overview



Unsupervised Machine Learning

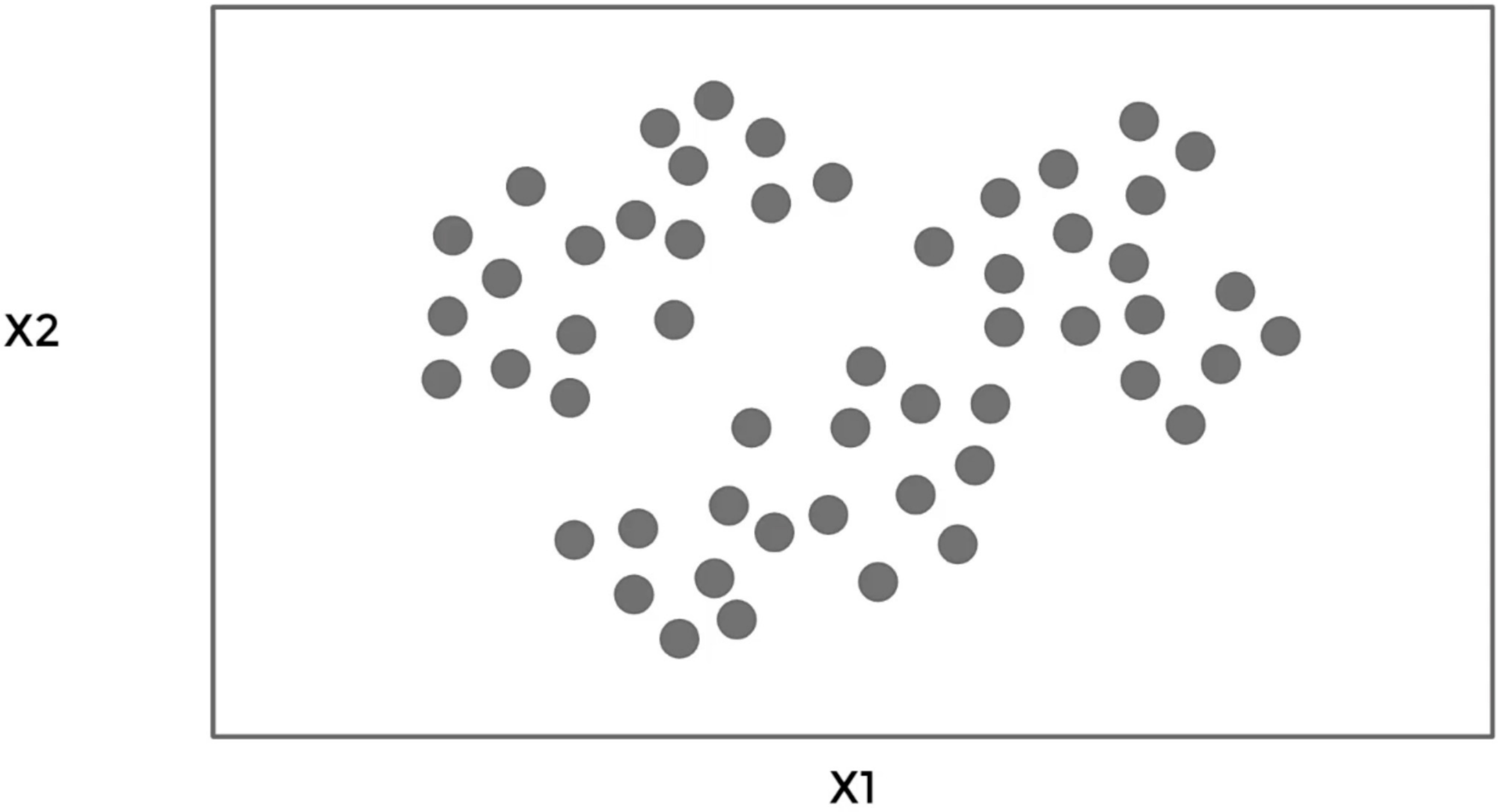
- Utilizes unlabeled data.
- No Response or dependent variable.

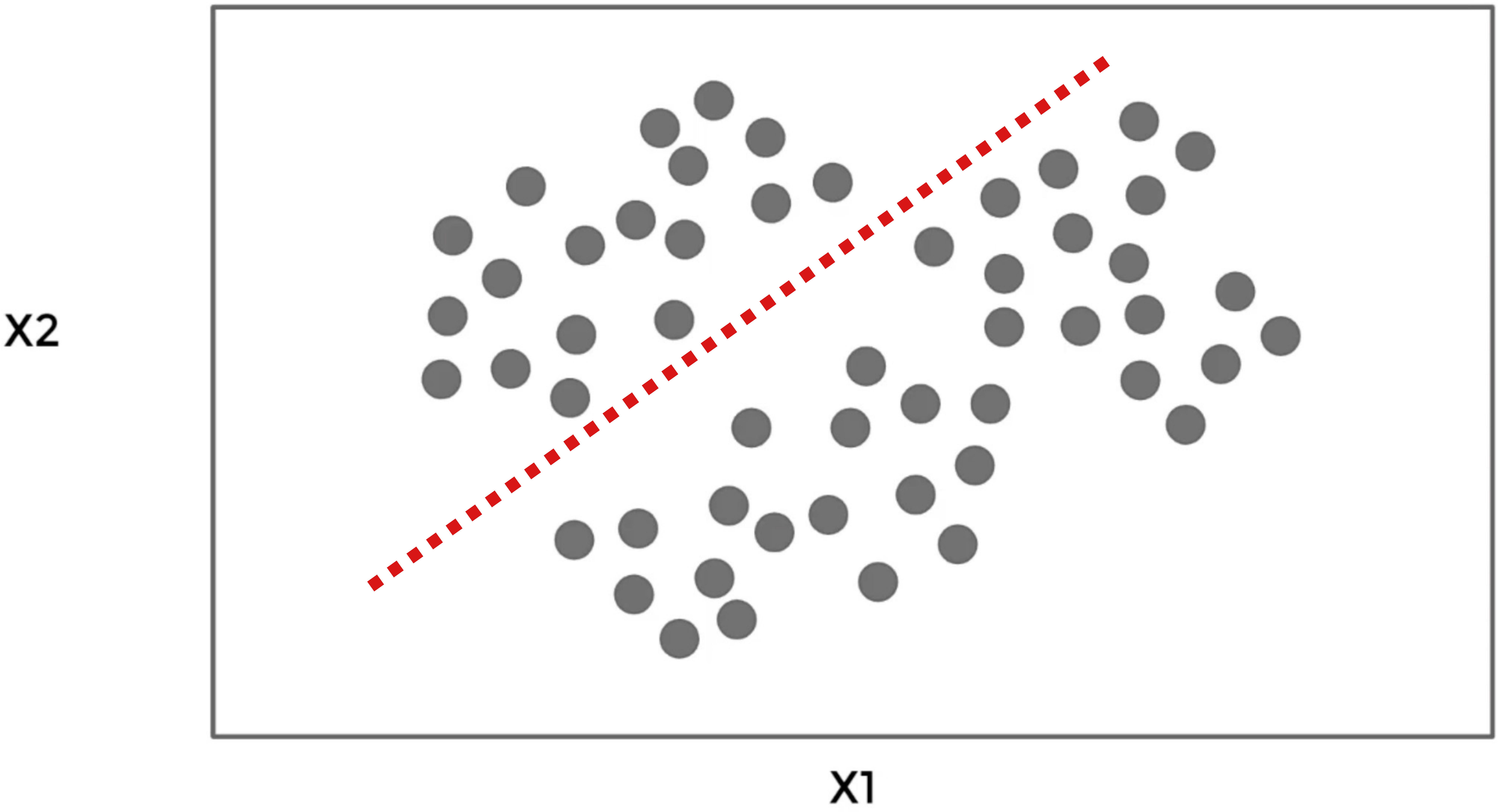
Similarity Measures

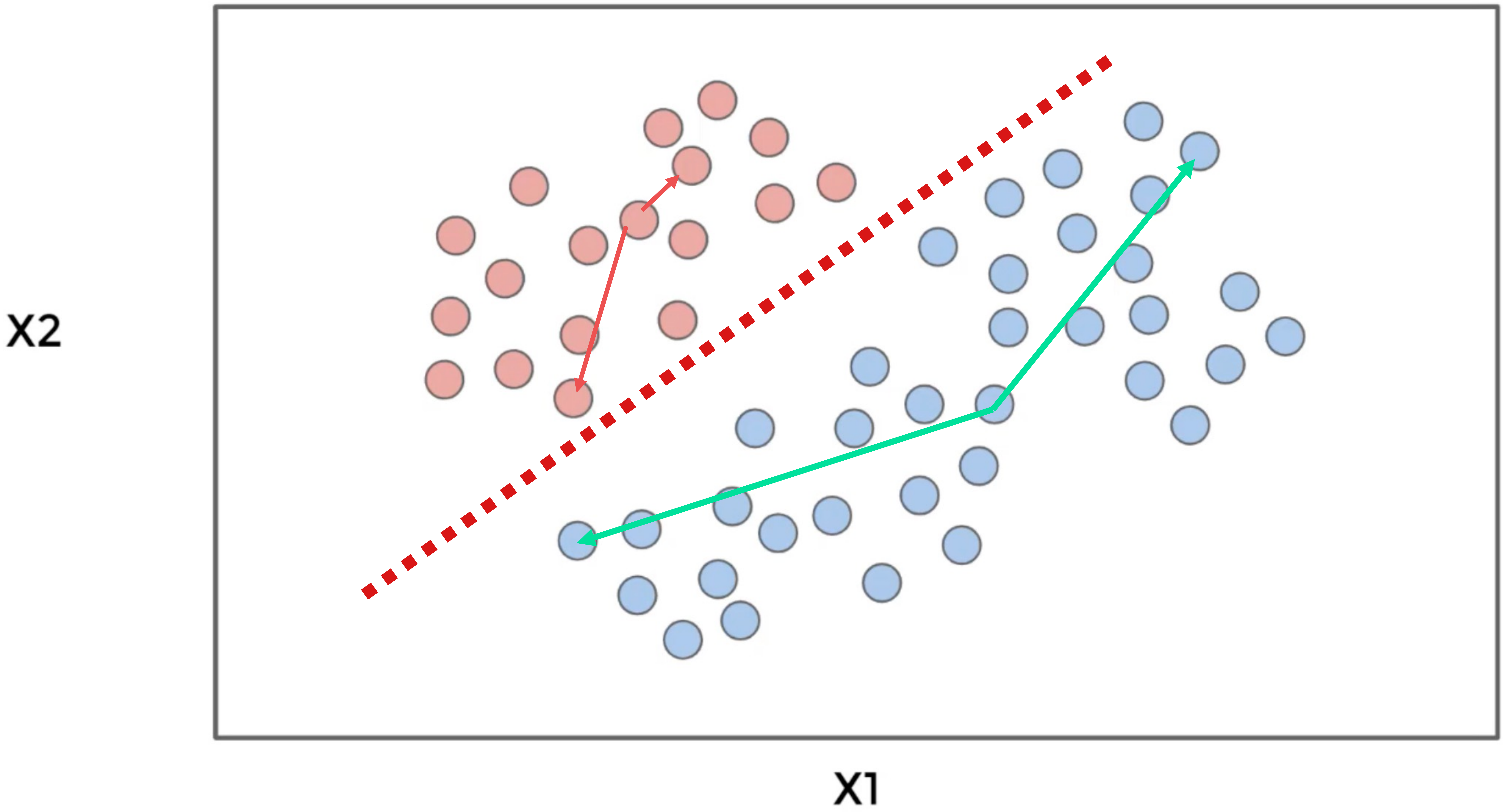
- Utilizes a similarity score to group together data points with the same characteristics.

Human Interpretation

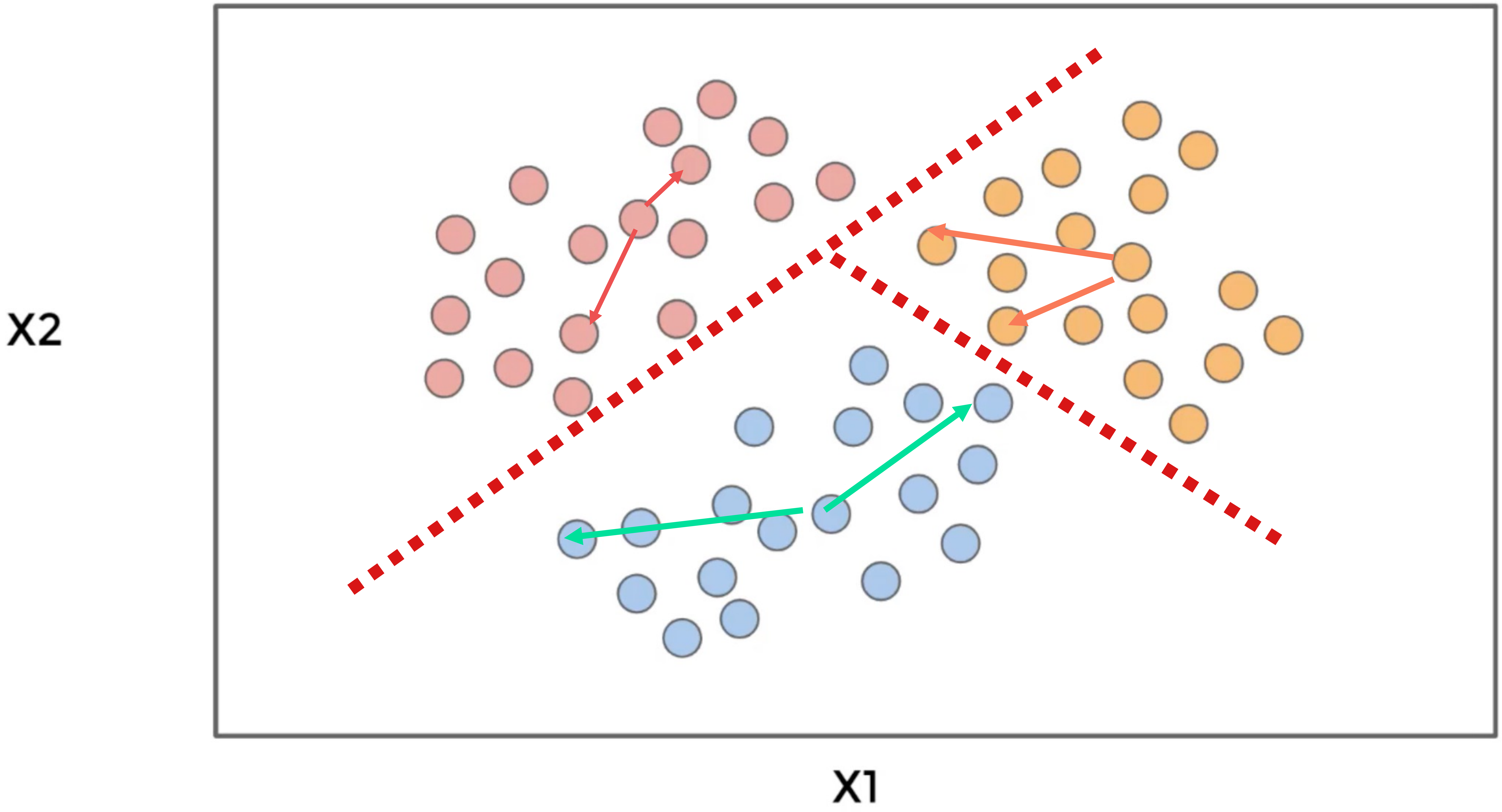
- Cluster characteristics requires interpretation based on the features used.


X2
X1


X2
X1


X2
X1

NOTE: Clustering algorithms will just group the data but will not label them.



CLUSTERING

SOME QUESTIONS TO KEEP IN MIND

Number of Clusters?

- User Inputted
- Statistical Measures

What Method?

- Distance based
- Tree based
- Density based

Performance metrics

- No ground truth.
- Statistical measures of purity

CLUSTERING

K-Means Clustering

Proposed by Hugo D. Steinhaus Polish Mathematician and Statistician

A

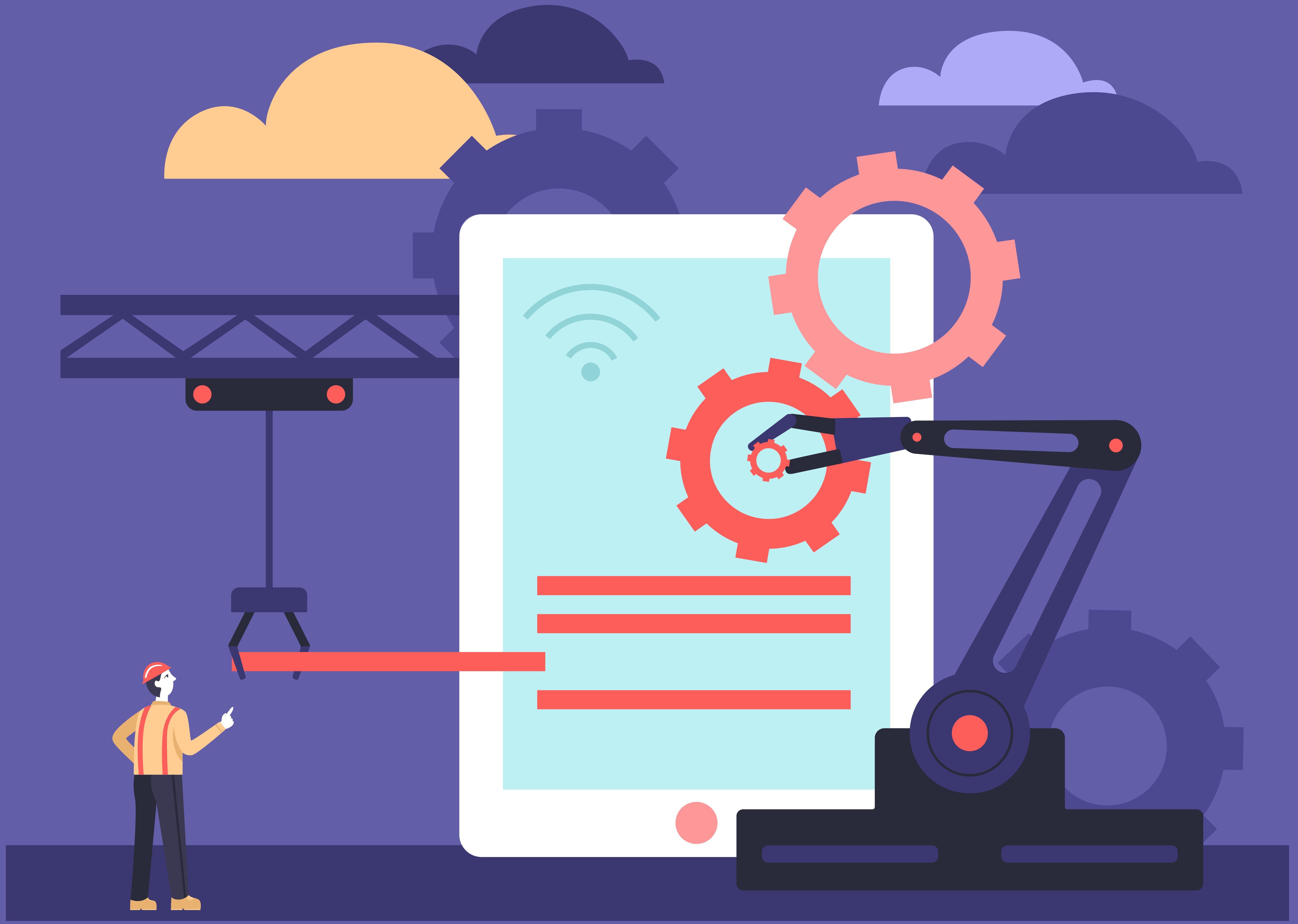
Group Assignment

Each point must belong to a group.

B

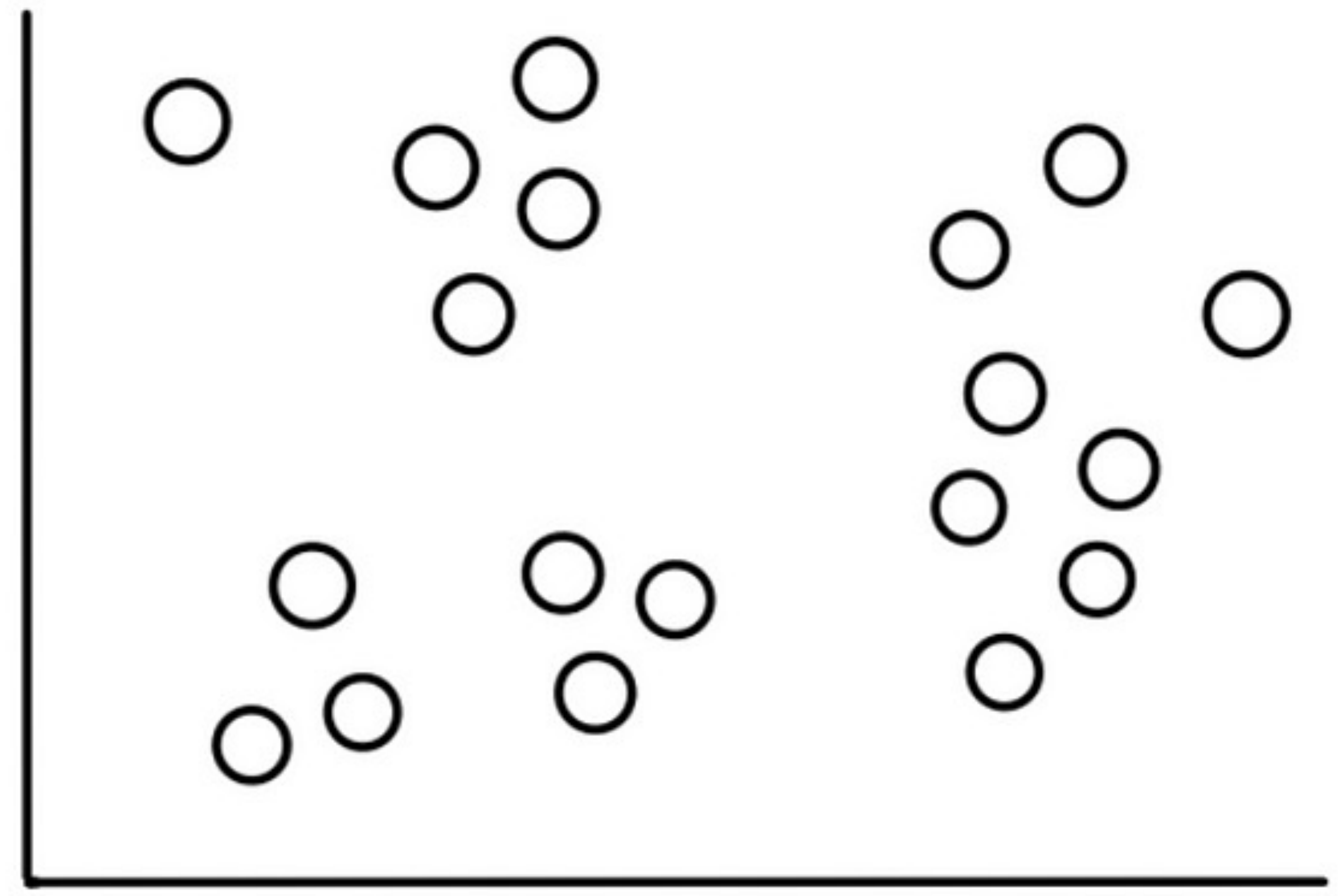
Hard Clustering

No point can be shared by two or more groups.



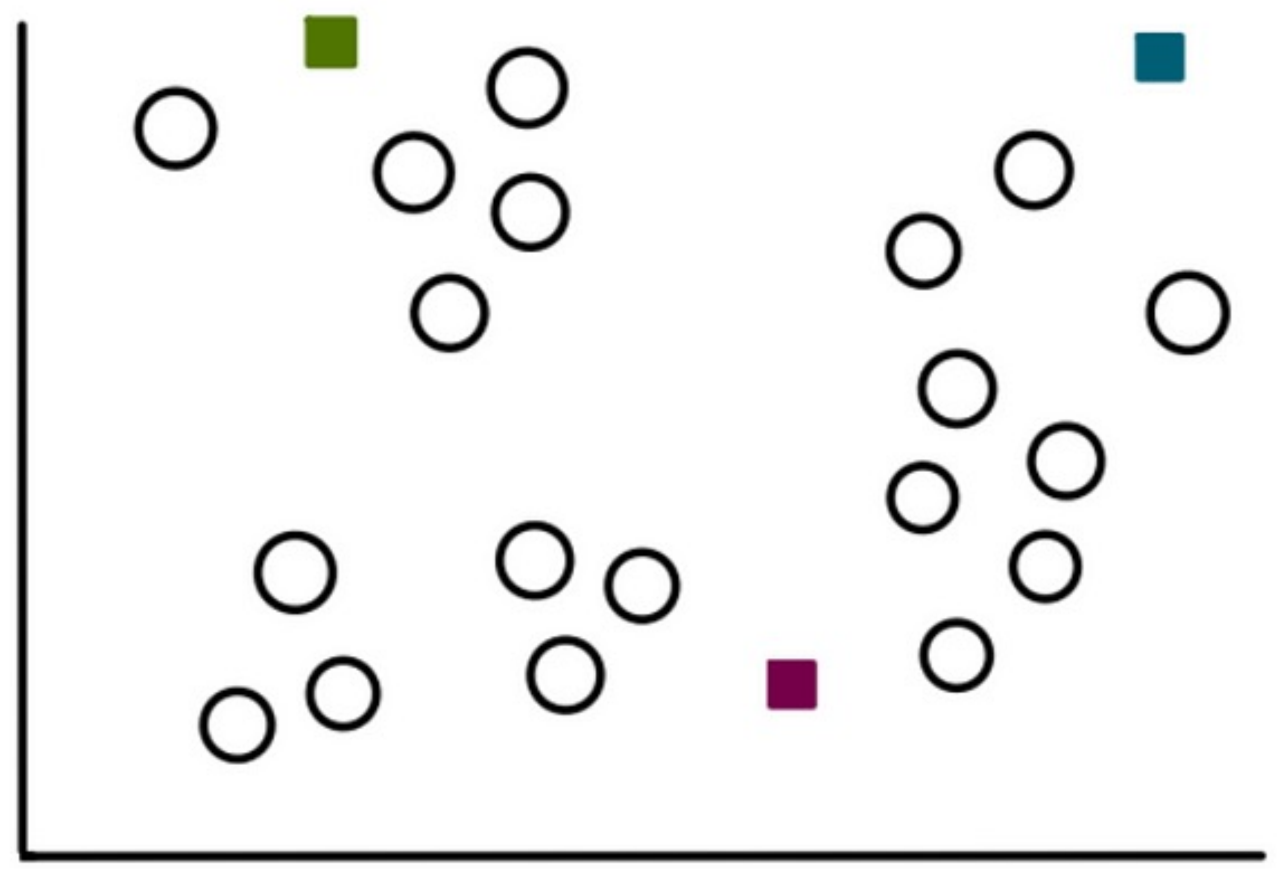
K-Means Clustering

Step 1: Choose number of cluster, k.



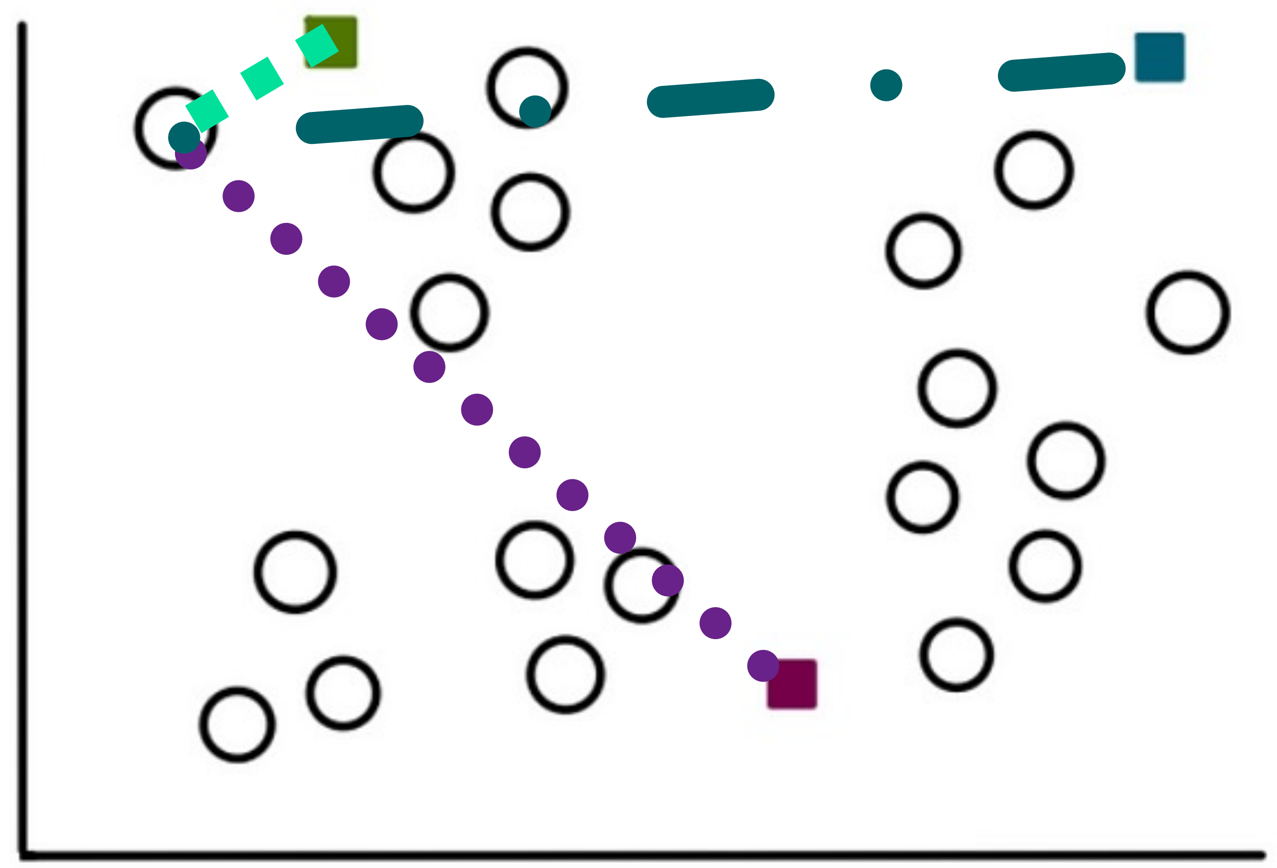
K-Means Clustering

Step 1: Choose number of cluster, k.
Step 2: Select initial centroids at random.



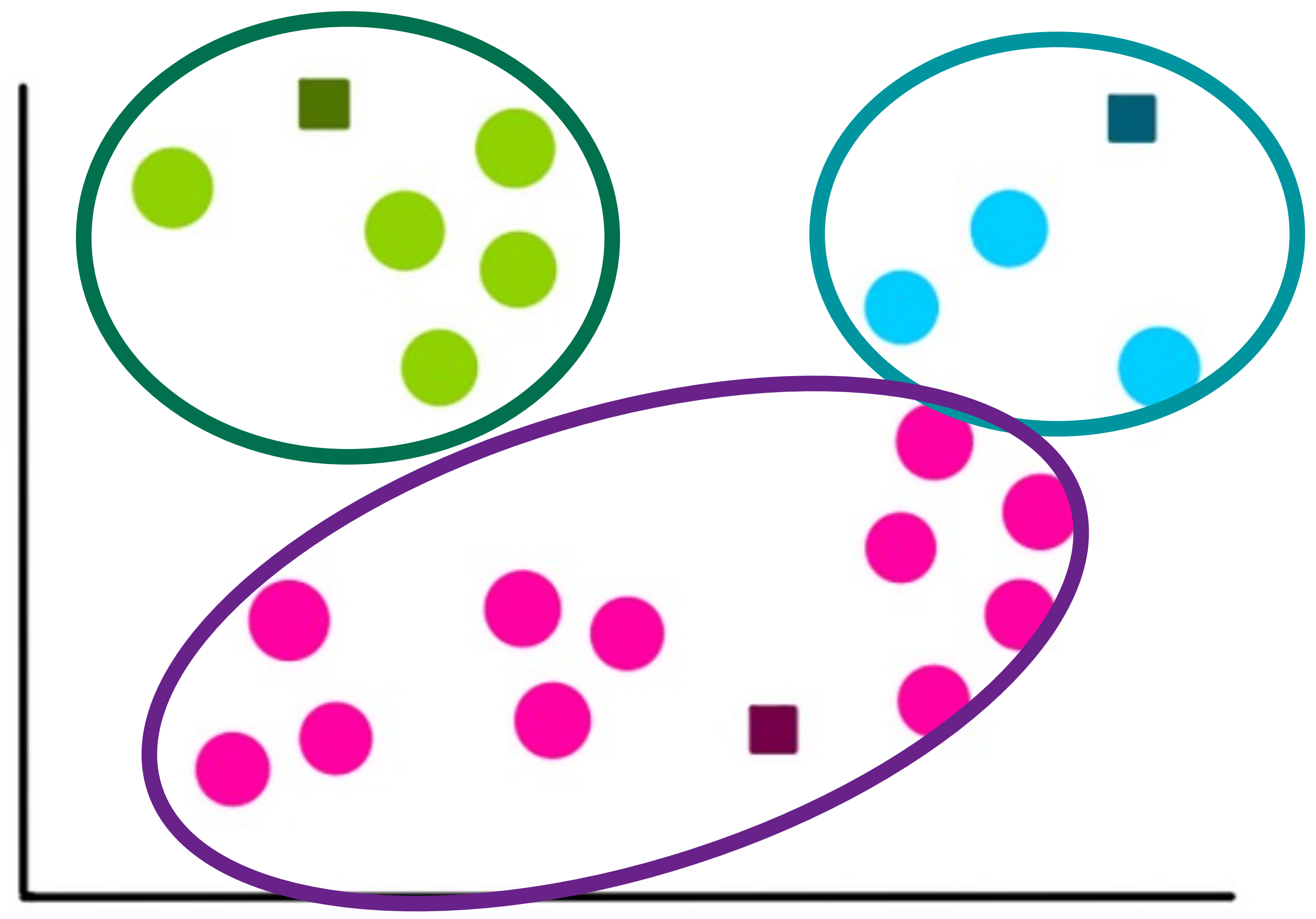
K-Means Clustering

Step 1: Choose number of cluster, k.
Step 2: Select initial centroids at random.
Step 3: Calculate the distance of each point to each centroid.



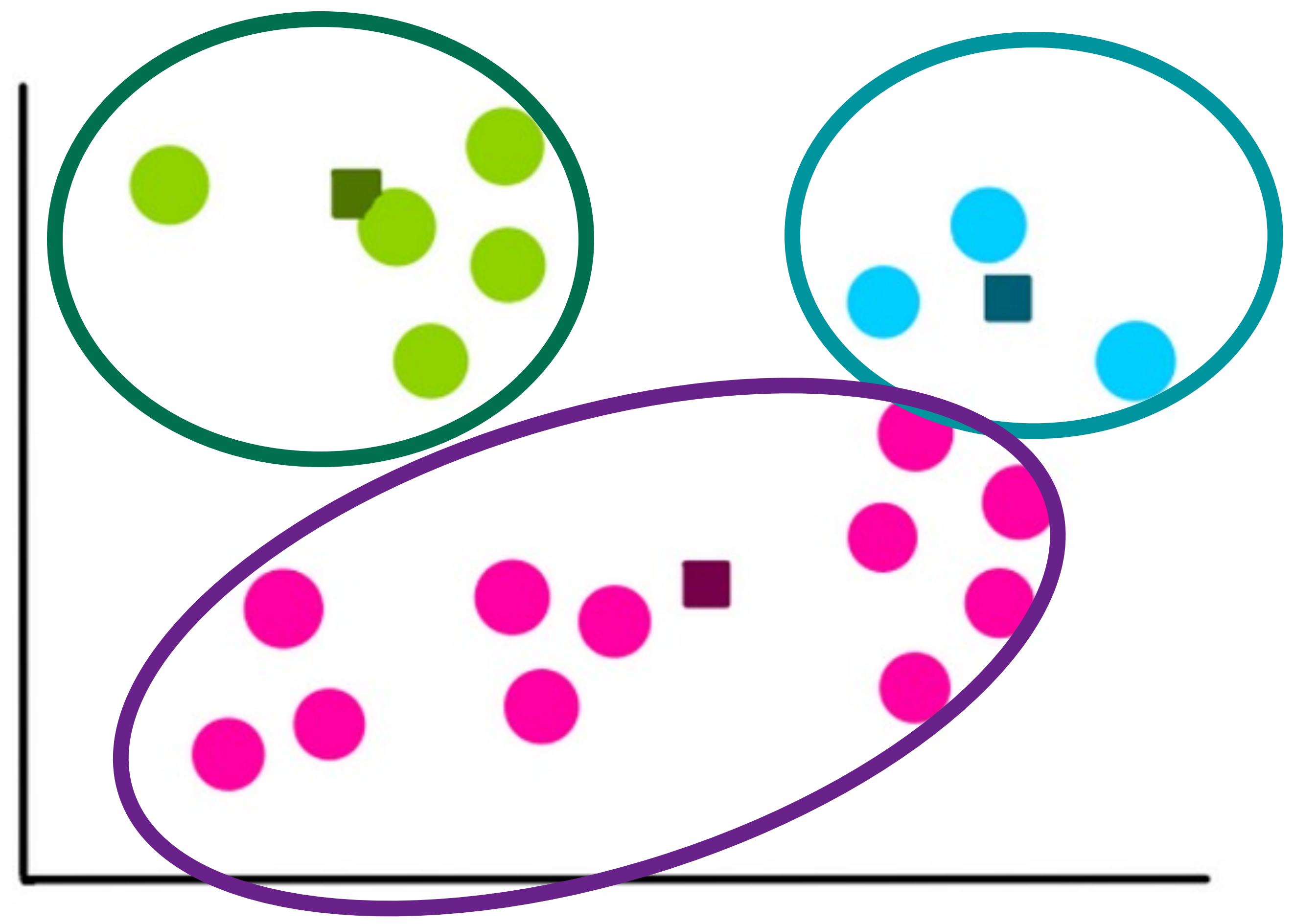
K-Means Clustering

Step 1: Choose number of cluster, k.
Step 2: Select initial centroids at random.
Step 3: Calculate the distance of each point to each centroid.
Step 4: Assign labels to points based on closest centroid.



K-Means Clustering

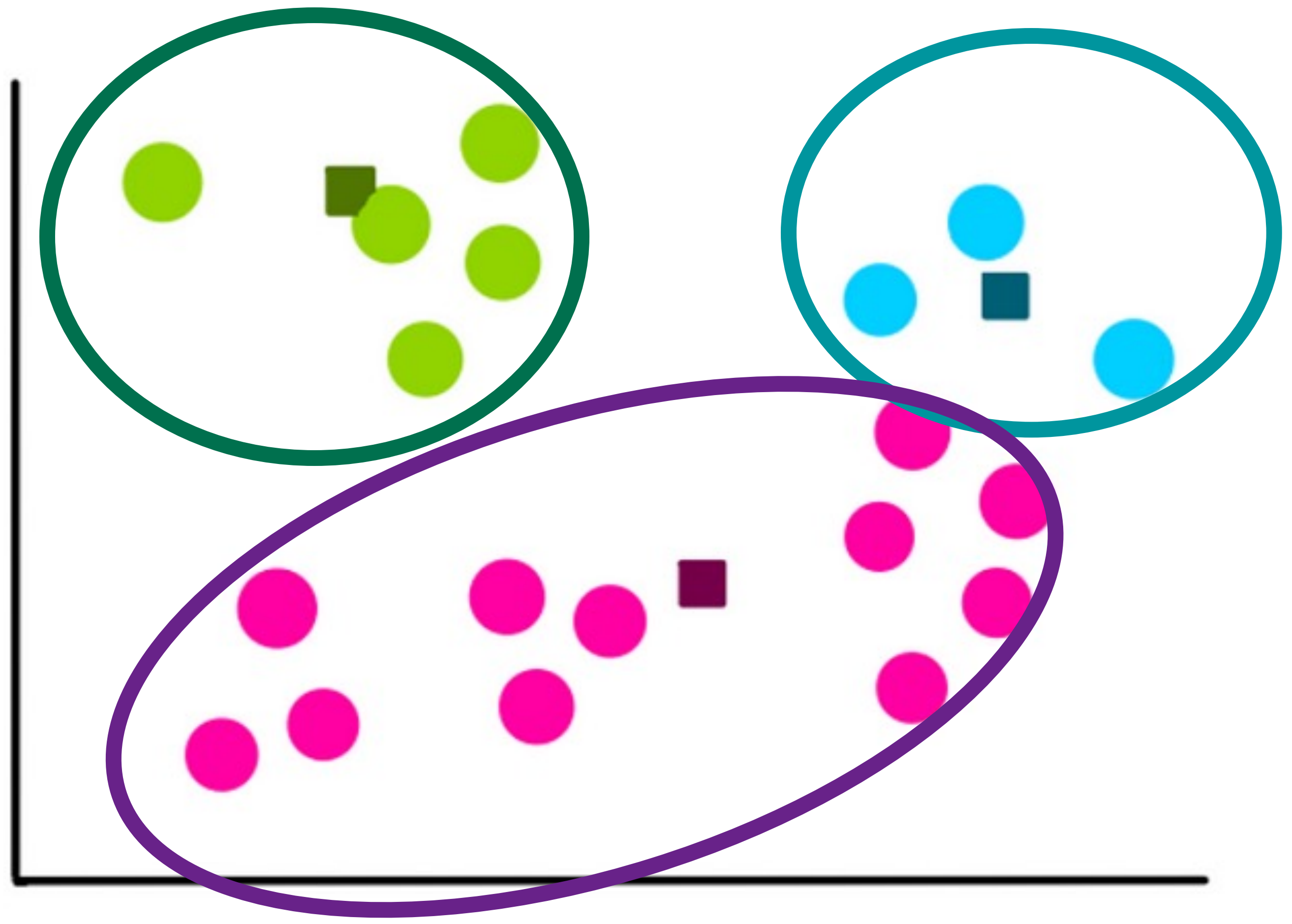
Step 1: Choose number of cluster, k.
Step 2: Select initial centroids at random.
Step 3: Calculate the distance of each point to each centroid.
Step 4: Assign labels to points based on closest centroid.
Step 5: Re-calculate position of new centroid based on groupings.



K-Means Clustering

Step 1: Choose number of cluster, k.
Step 2: Select initial centroids at random.
Step 3: Calculate the distance of each point to each centroid.
Step 4: Assign labels to points based on closest centroid.
Step 5: Re-calculate position of new centroid based on groupings.
Step 6: Evaluate cluster performance by using the Within Sum of Squares:

$$WSS = \sum_{k=1}^{K} \sum_{i=1}^{N} \left\| x_{i,k}, C_k \right\|$$

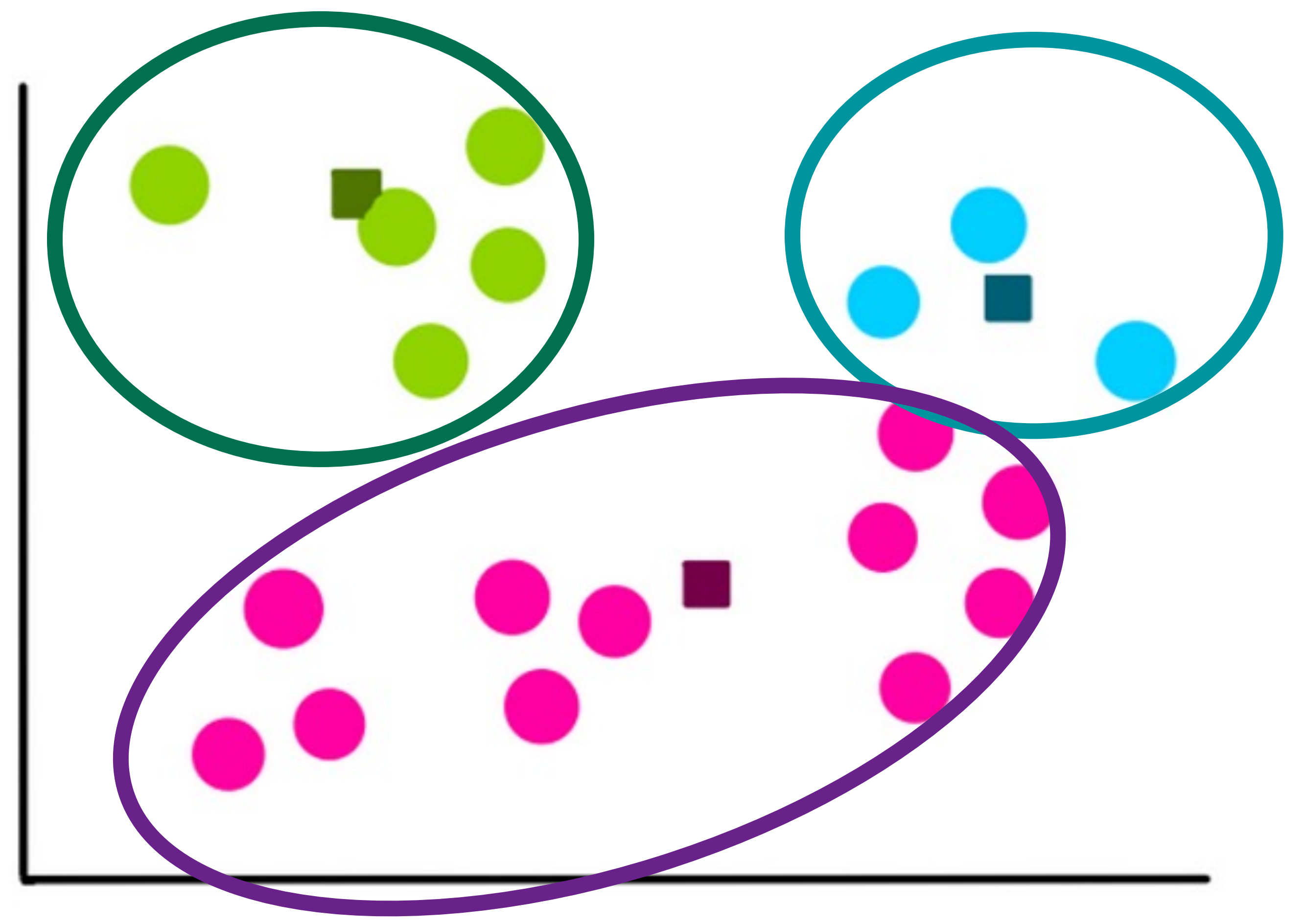


K-Means Clustering

Step 1: Choose number of cluster, k.
Step 2: Select initial centroids at random.
Step 3: Calculate the distance of each point to each centroid.
Step 4: Assign labels to points based on closest centroid.
Step 5: Re-calculate position of new centroid based on groupings.
Step 6: Evaluate cluster performance by using the Within Sum of Squares:

$$WSS = \sum_{k=1}^{K} \sum_{i=1}^{N} \left\| x_{i,k}, C_k \right\|$$

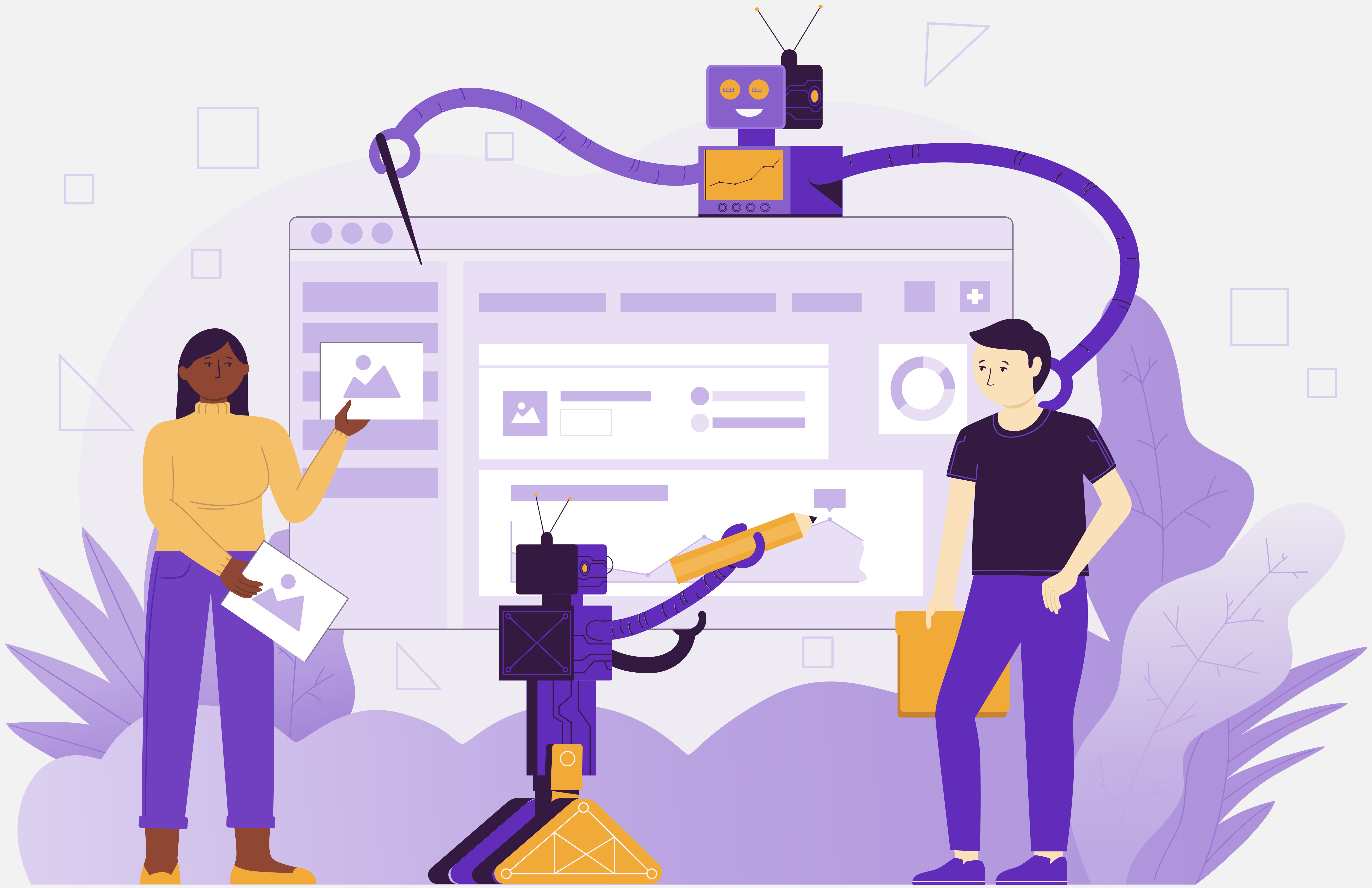
Question: Looking at the steps, what is the obvious limitations of K-means?



CLUSTERING

Hierarchical Clustering

Robert R. Sokal (Biostatistician & Entomologist)
Peter H.A. Sneath (Biostatistician & Microbiologist)



A

Advantages

- Easy to understand & Visualize.
- There is no need to choose number of prior cluster.

B

Disadvantages

- Greedy!
- The more data the longer time it takes to visualize.

C

Types

1. Agglomerative
2. Divisive

CLUSTERING

Hierarchical Clustering

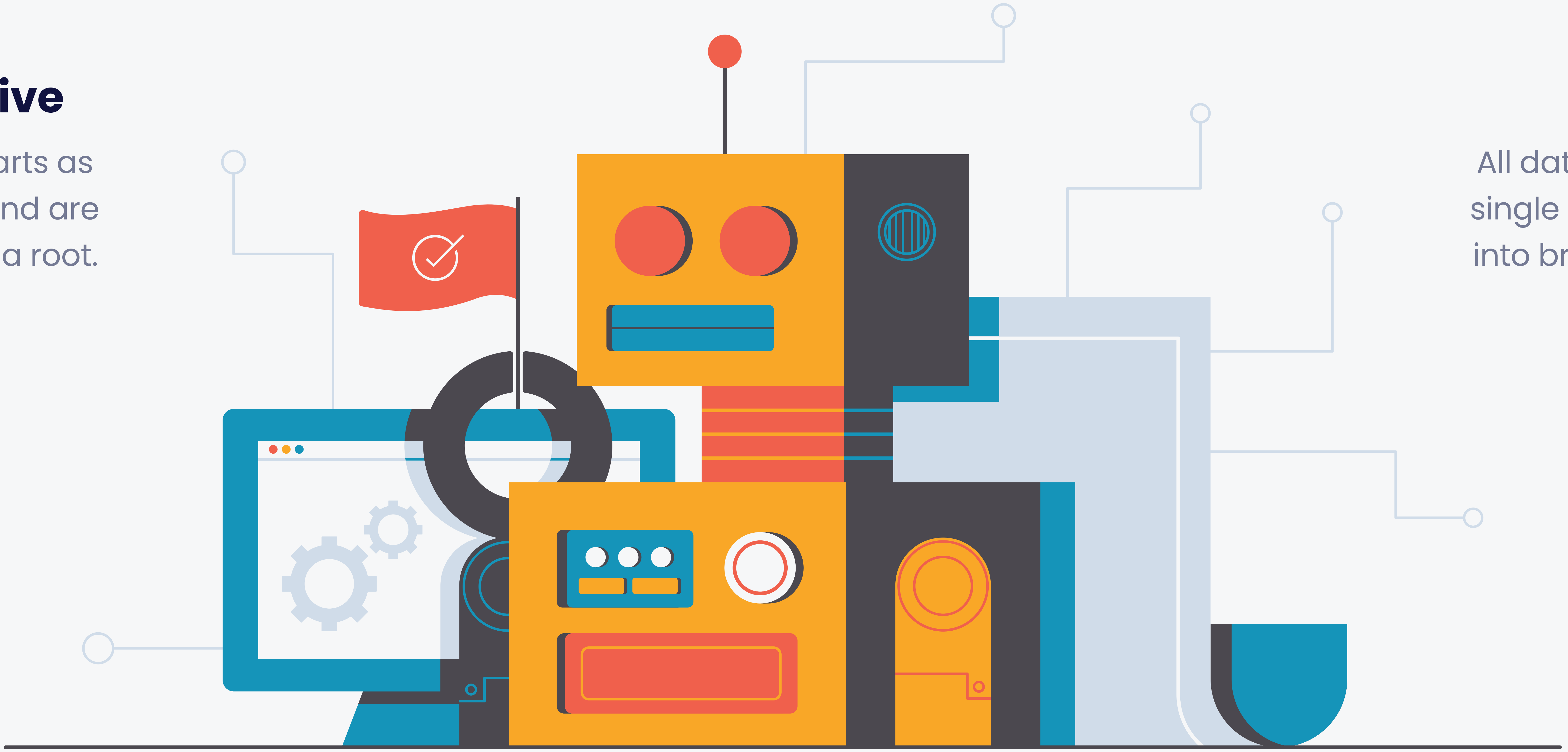
TYPE

Agglomerative

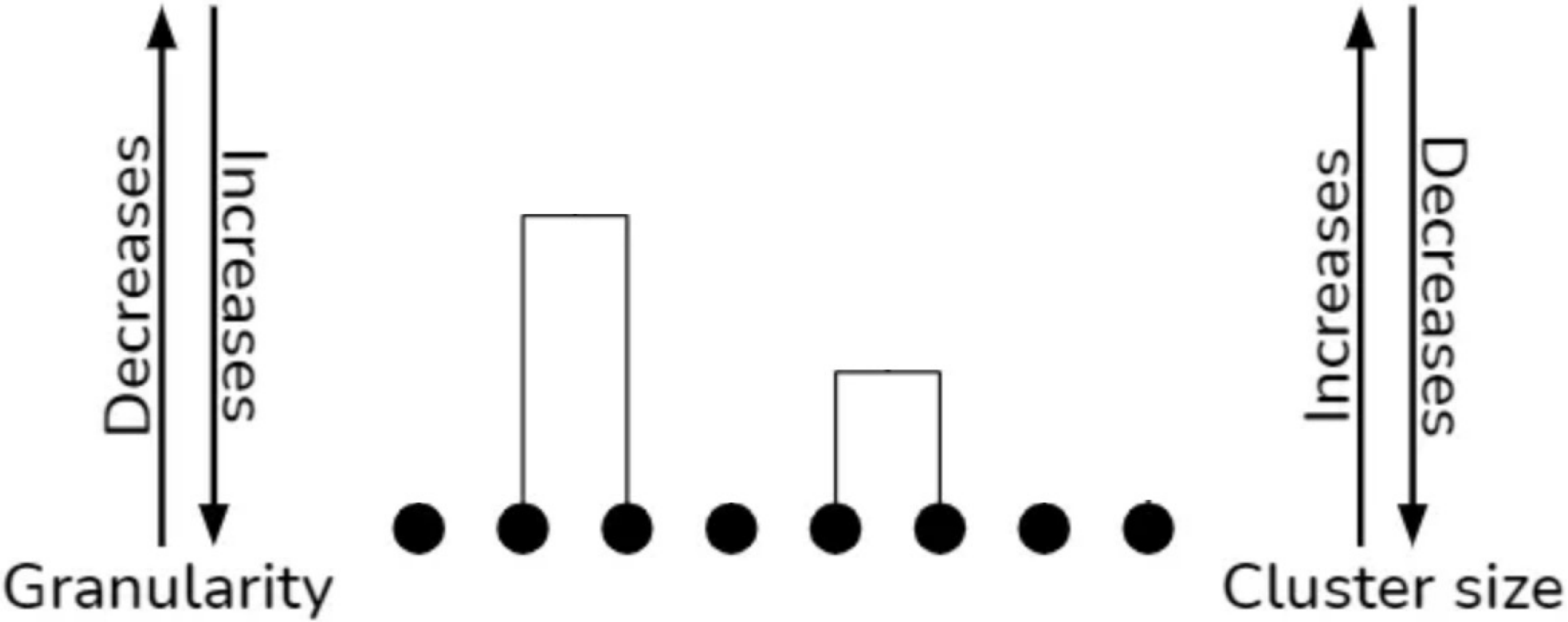
Each Data Points starts as a separate cluster and are joined together into a root.

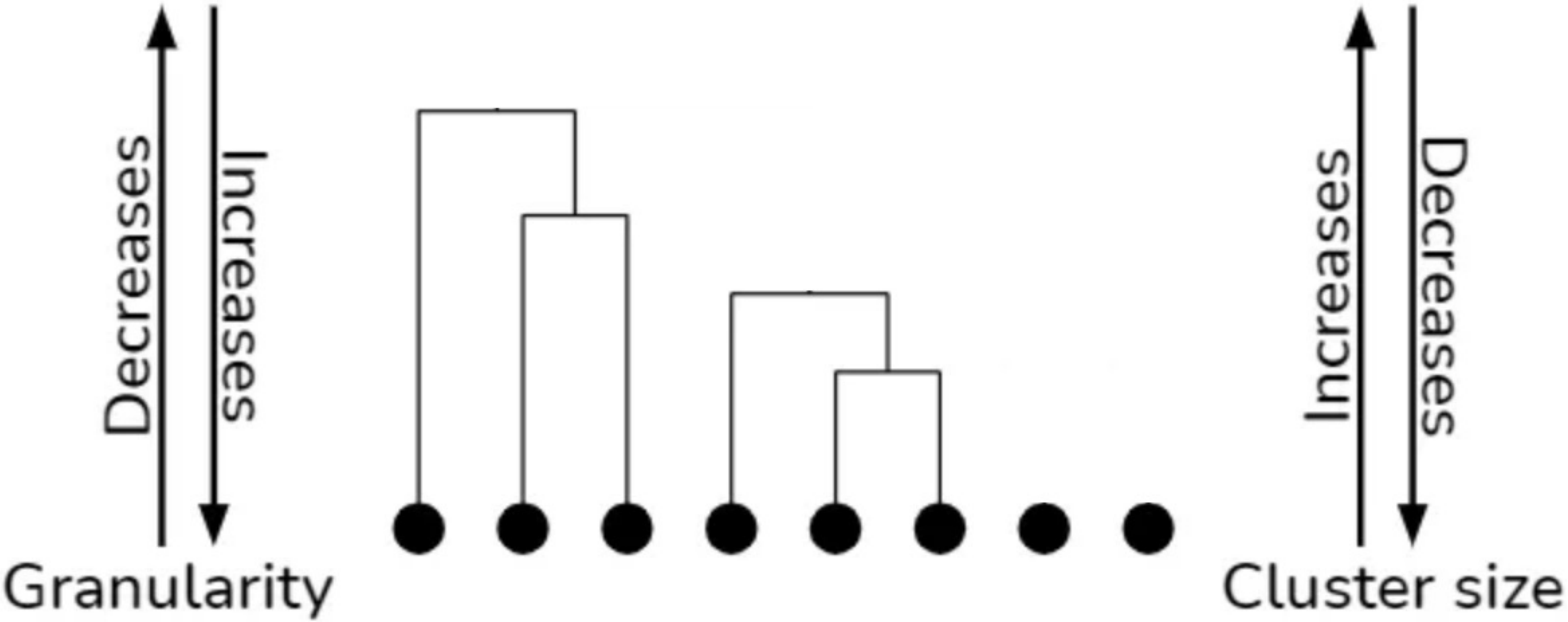
DIVISIVE

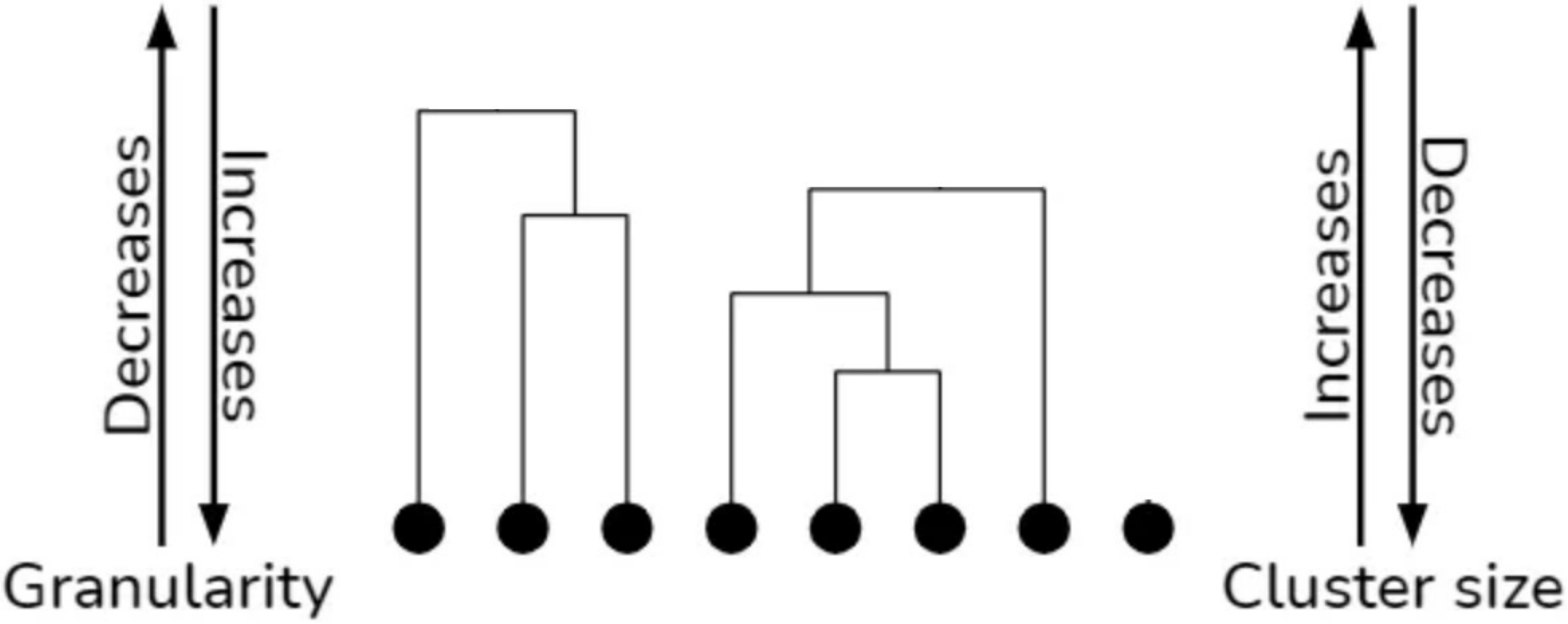
All data points starts as a single cluster and are split into branches and leaves.

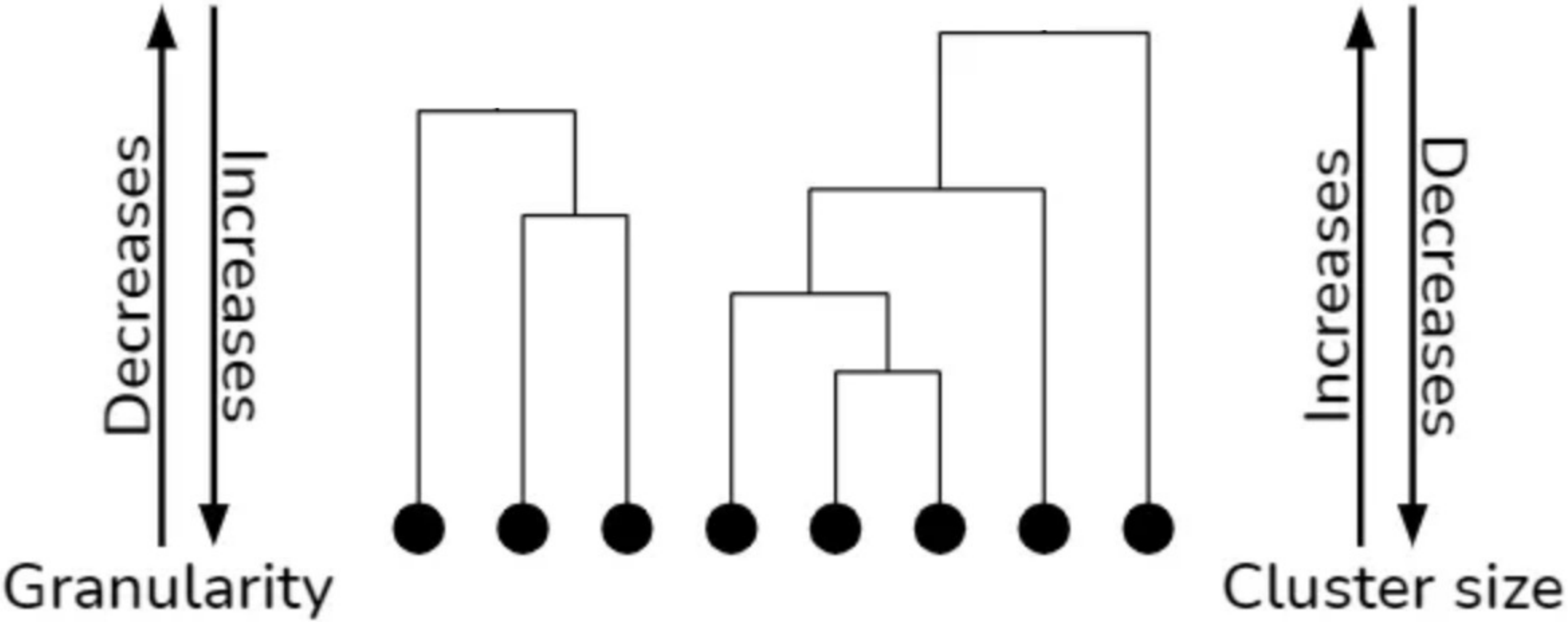


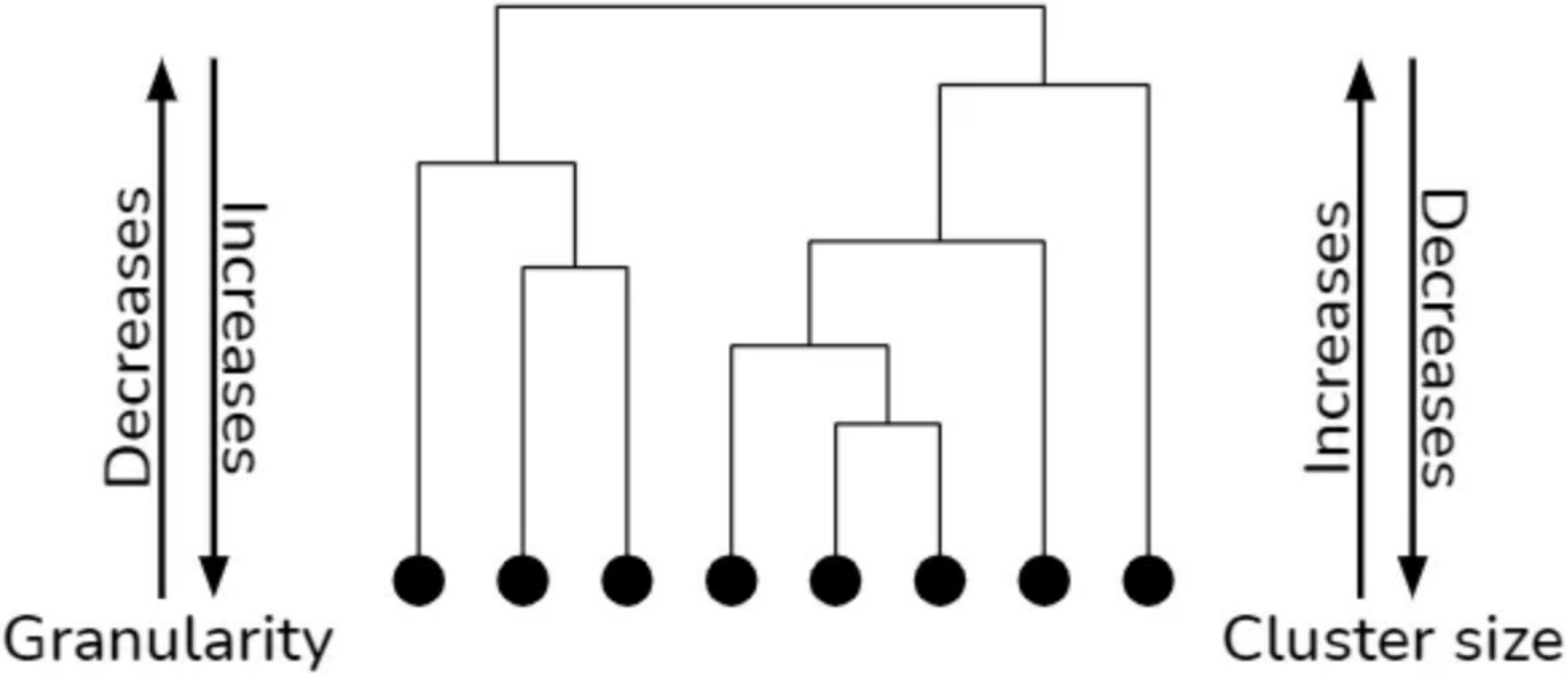

Decreases
Increases
Granularity
Increases
Decreases
Cluster size


Decreases
Increases
Granularity
Increases
Decreases
Cluster size


Decreases
Increases
Granularity
Increases
Decreases
Cluster size

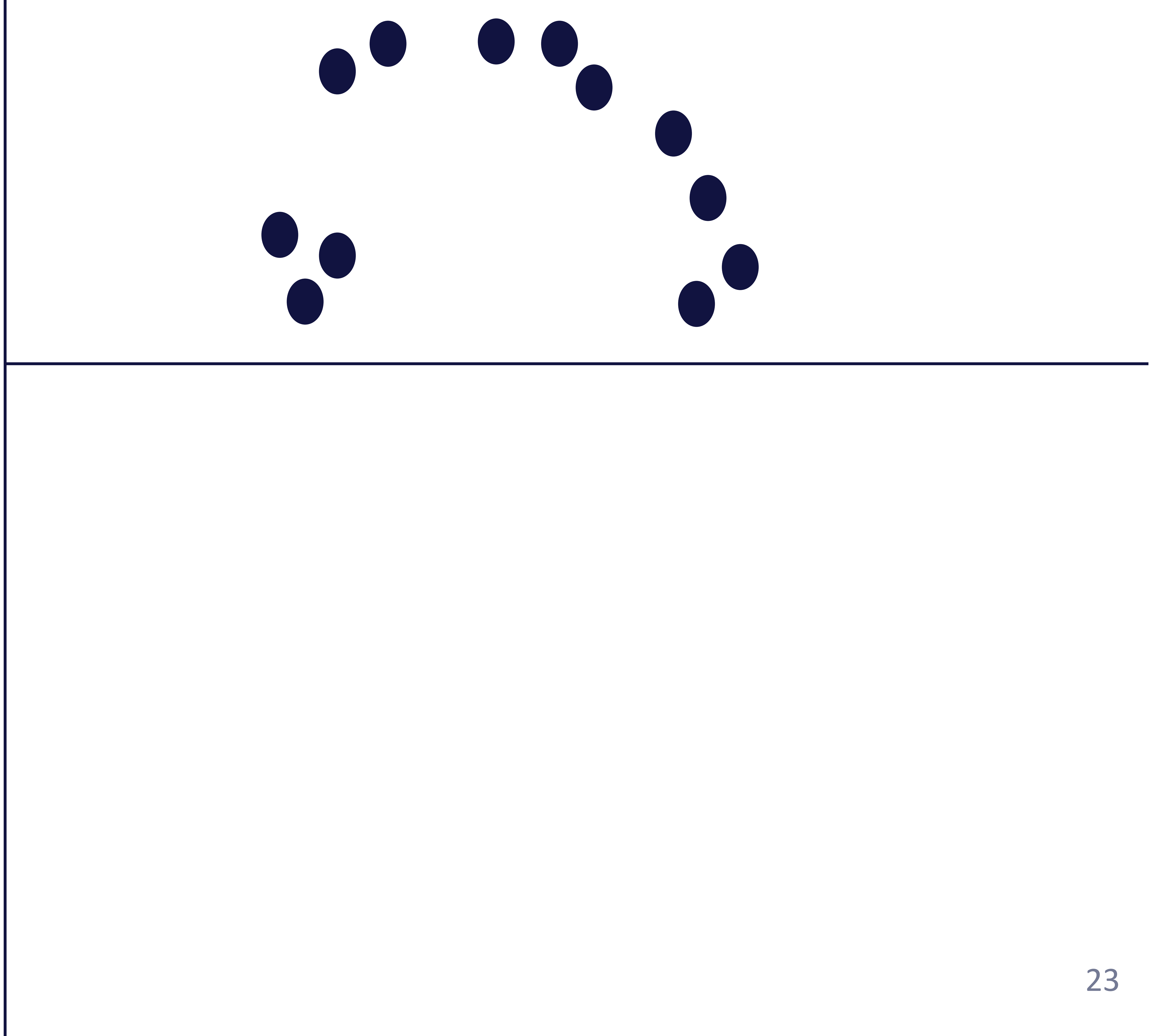

Decreases
Increases
Granularity
Increases
Decreases
Cluster size


Decreases
Increases
Granularity
Increases
Decreases
Cluster size


Decreases
Increases
Granularity
Increases
Decreases
Cluster size

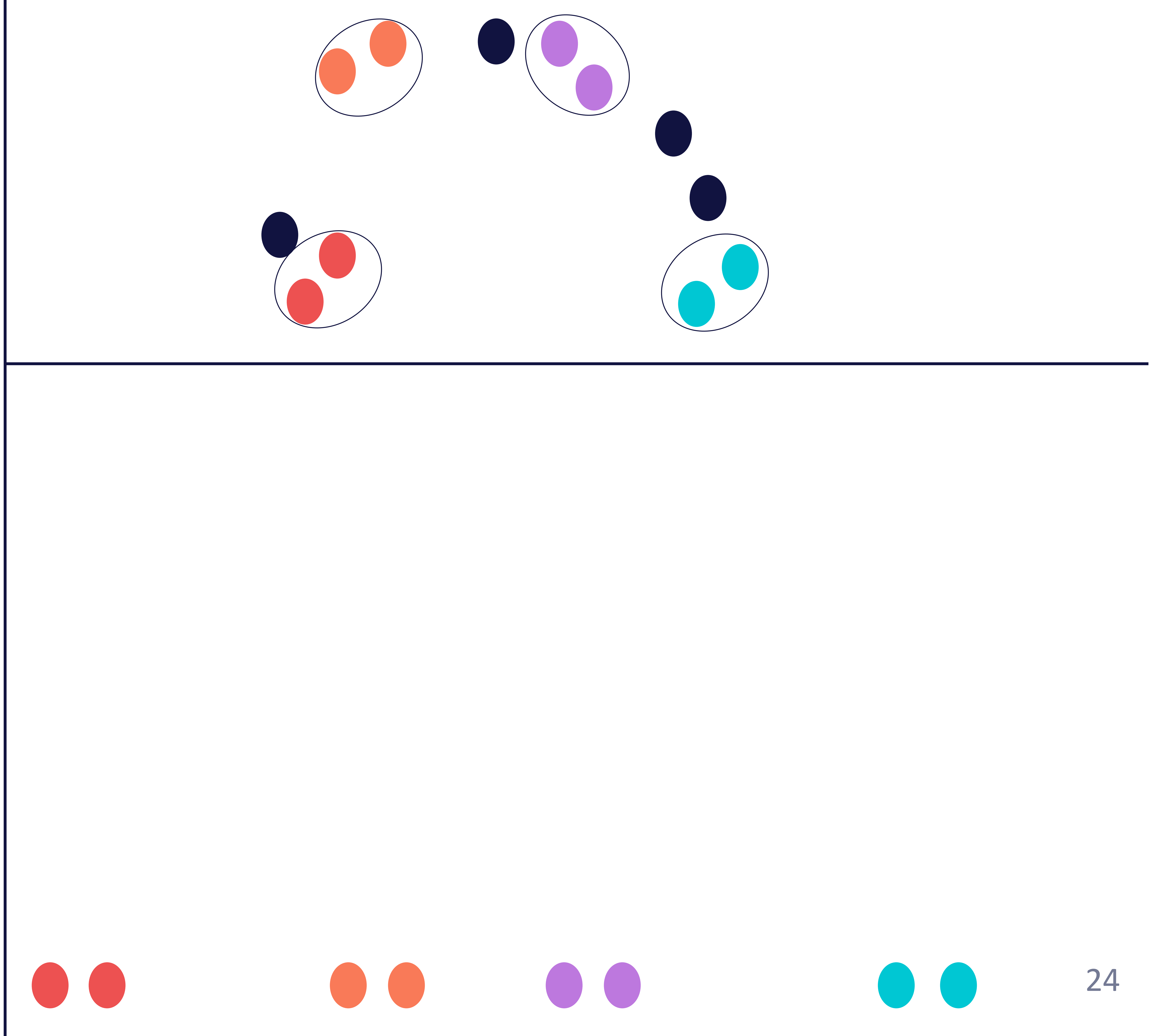
Agglomerative Hierarchical Clustering

Step 1: Find a suitable similarity metric.



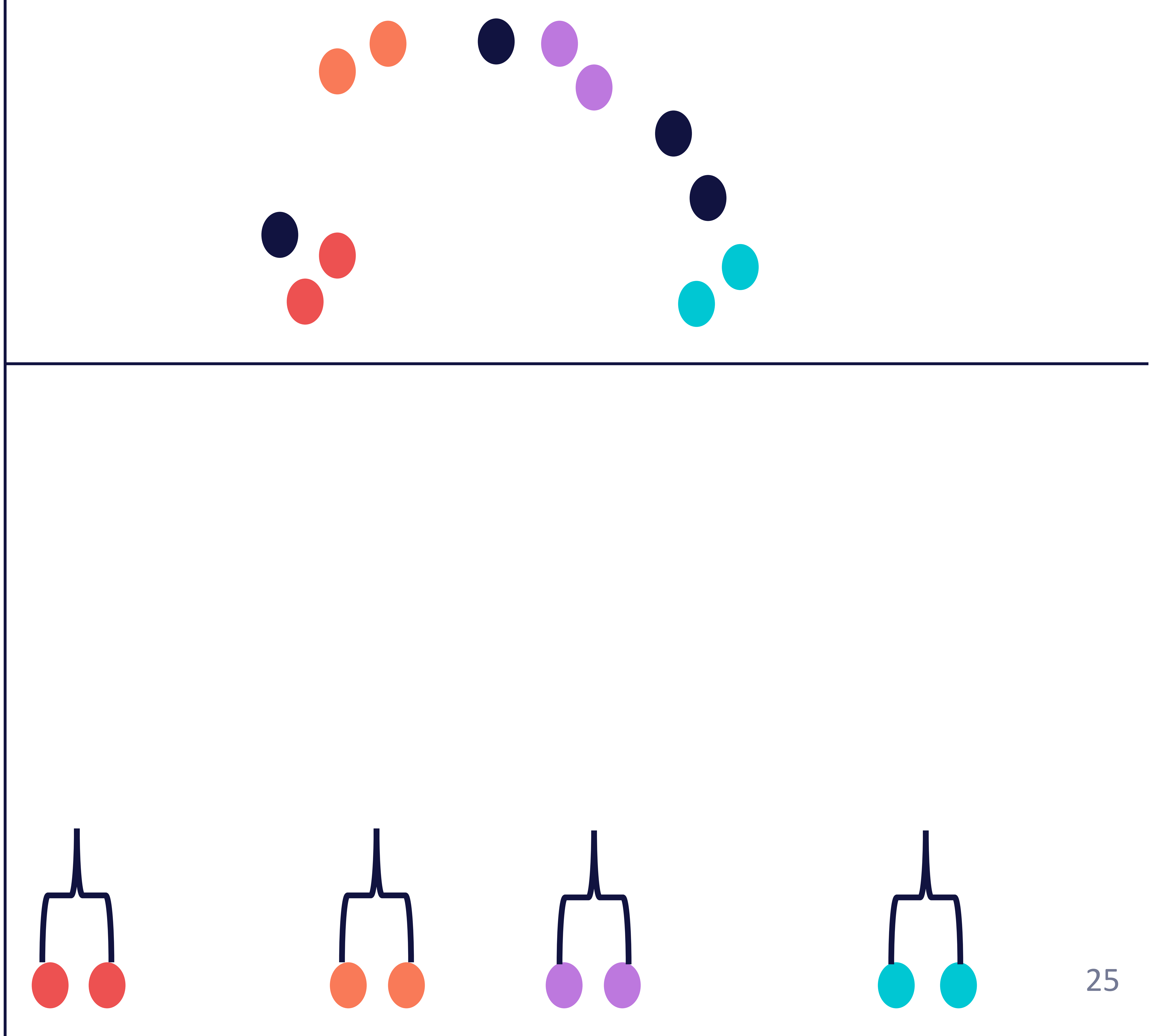
Agglomerative Hierarchical Clustering

Step 1: Find a suitable similarity metric.
Step 2: Use the similarity metric to find the closest pair.



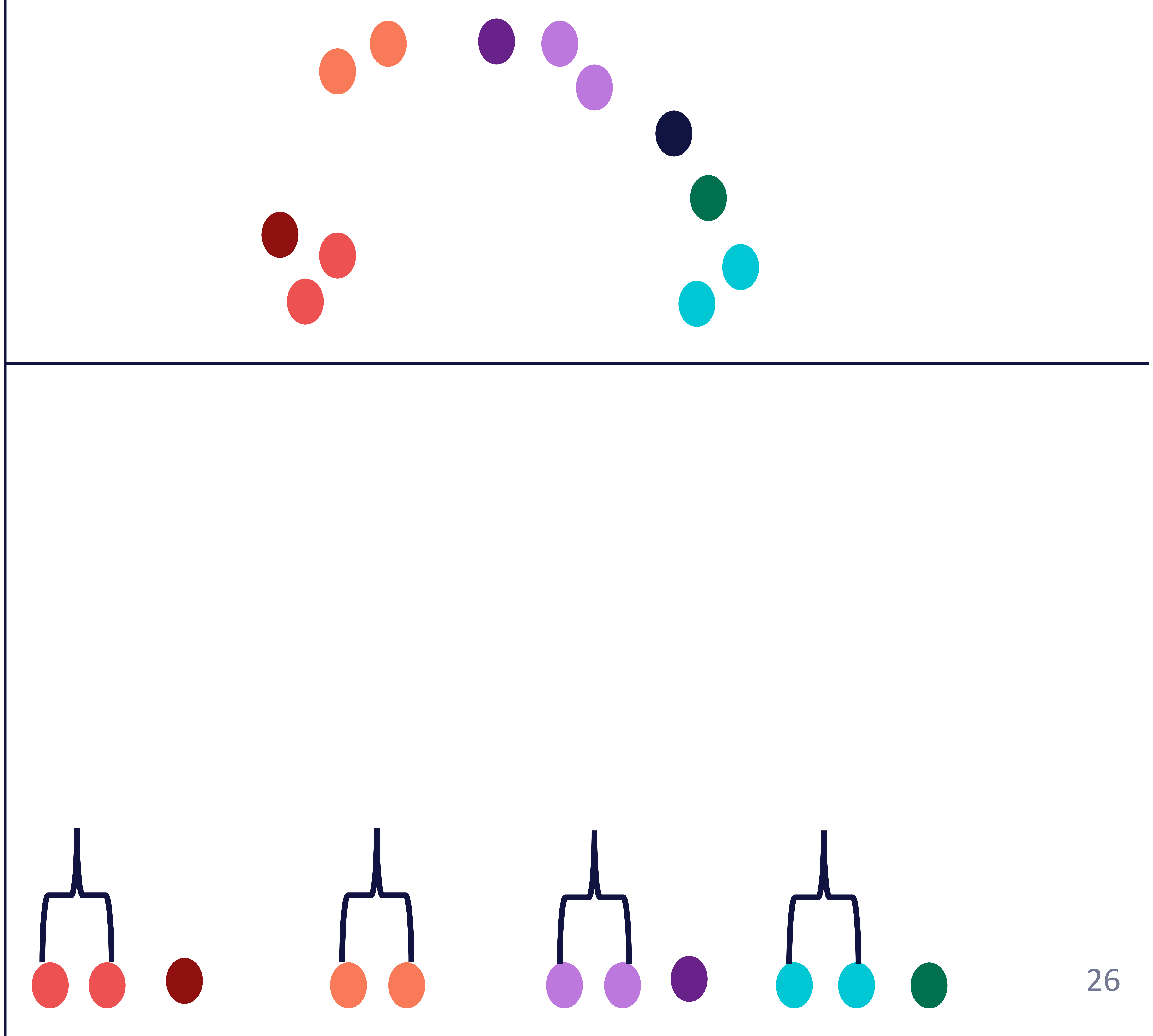
Agglomerative Hierarchical Clustering

Step 1: Find a suitable similarity metric.
Step 2: Use the similarity metric to find the closest pair.



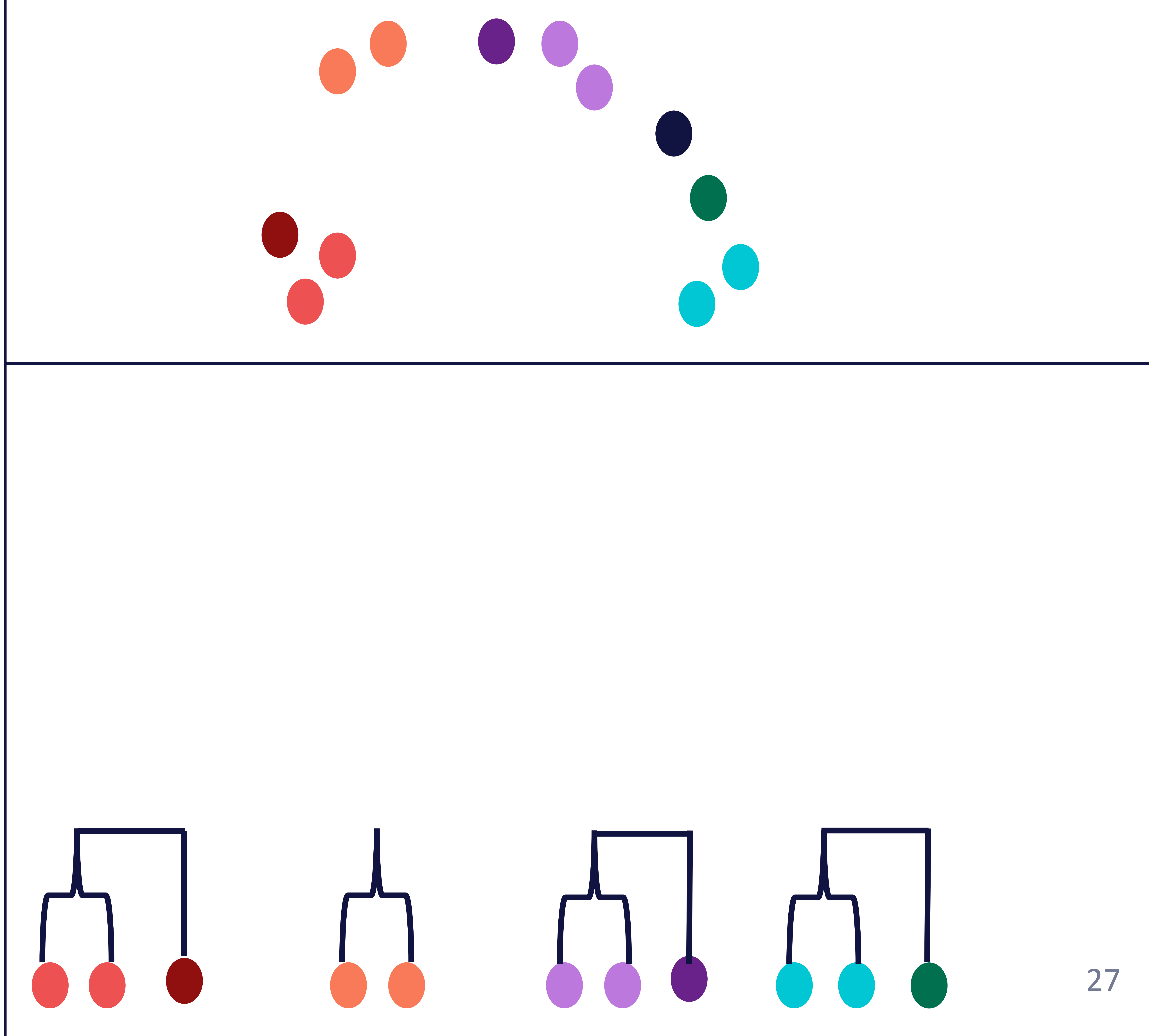
Agglomerative Hierarchical Clustering

Step 1: Find a suitable similarity metric.
Step 2: Use the similarity metric to find the closest pair.
Step 3: Iterate and look for the next point closest to a point belonging to a cluster.



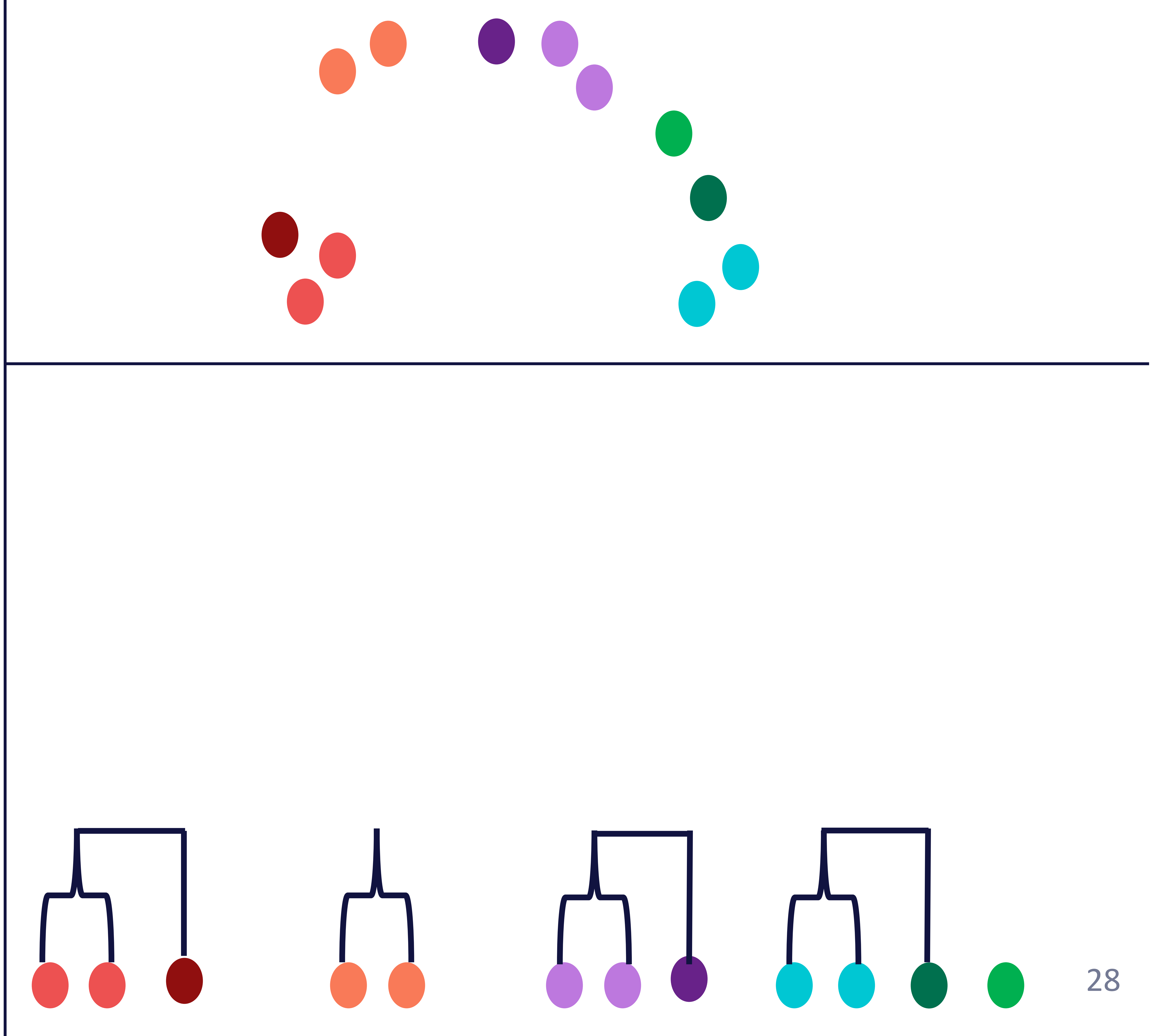
Agglomerative Hierarchical Clustering

Step 1: Find a suitable similarity metric.
Step 2: Use the similarity metric to find the closest pair.
Step 3: Iterate and look for the next point closest to a point belonging to a cluster.



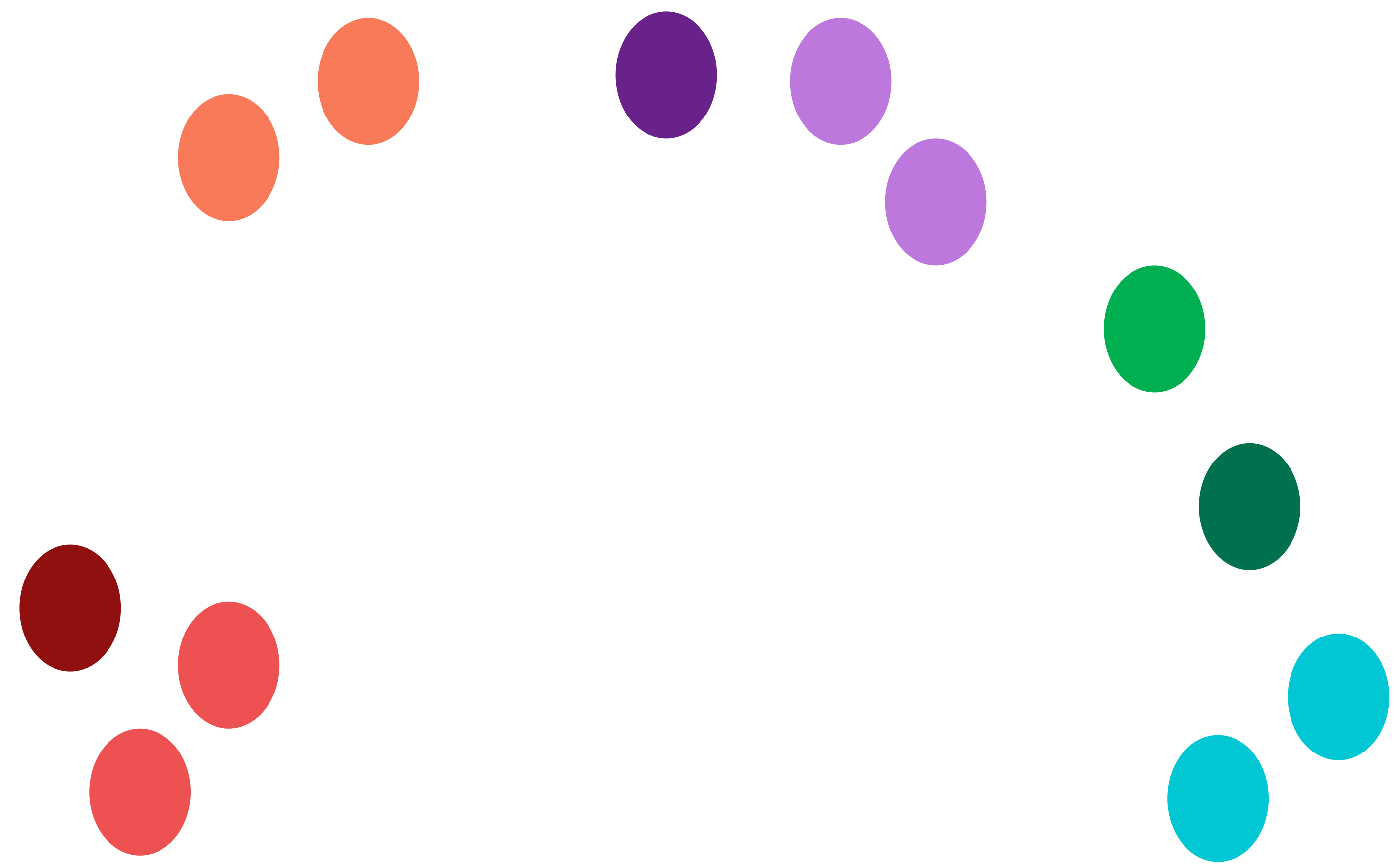
Agglomerative Hierarchical Clustering

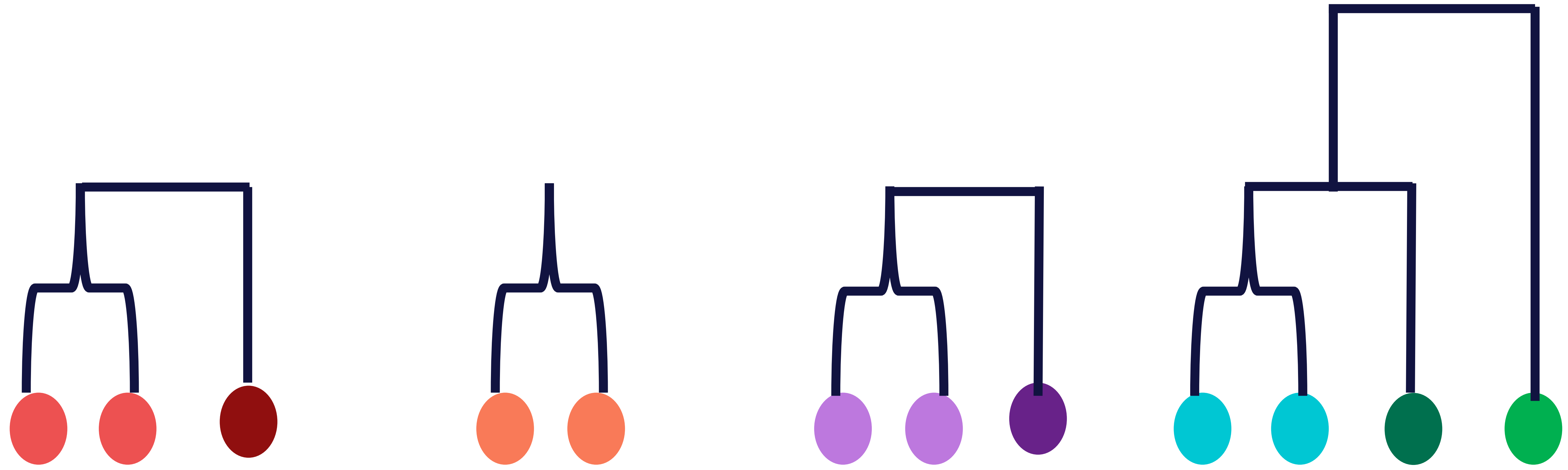
Step 1: Find a suitable similarity metric.
Step 2: Use the similarity metric to find the closest pair.
Step 3: Iterate and look for the next point closest to a point belonging to a cluster.
Step 4: Iterate and look for the next point closest to a point belonging to a cluster.



Agglomerative Hierarchical Clustering

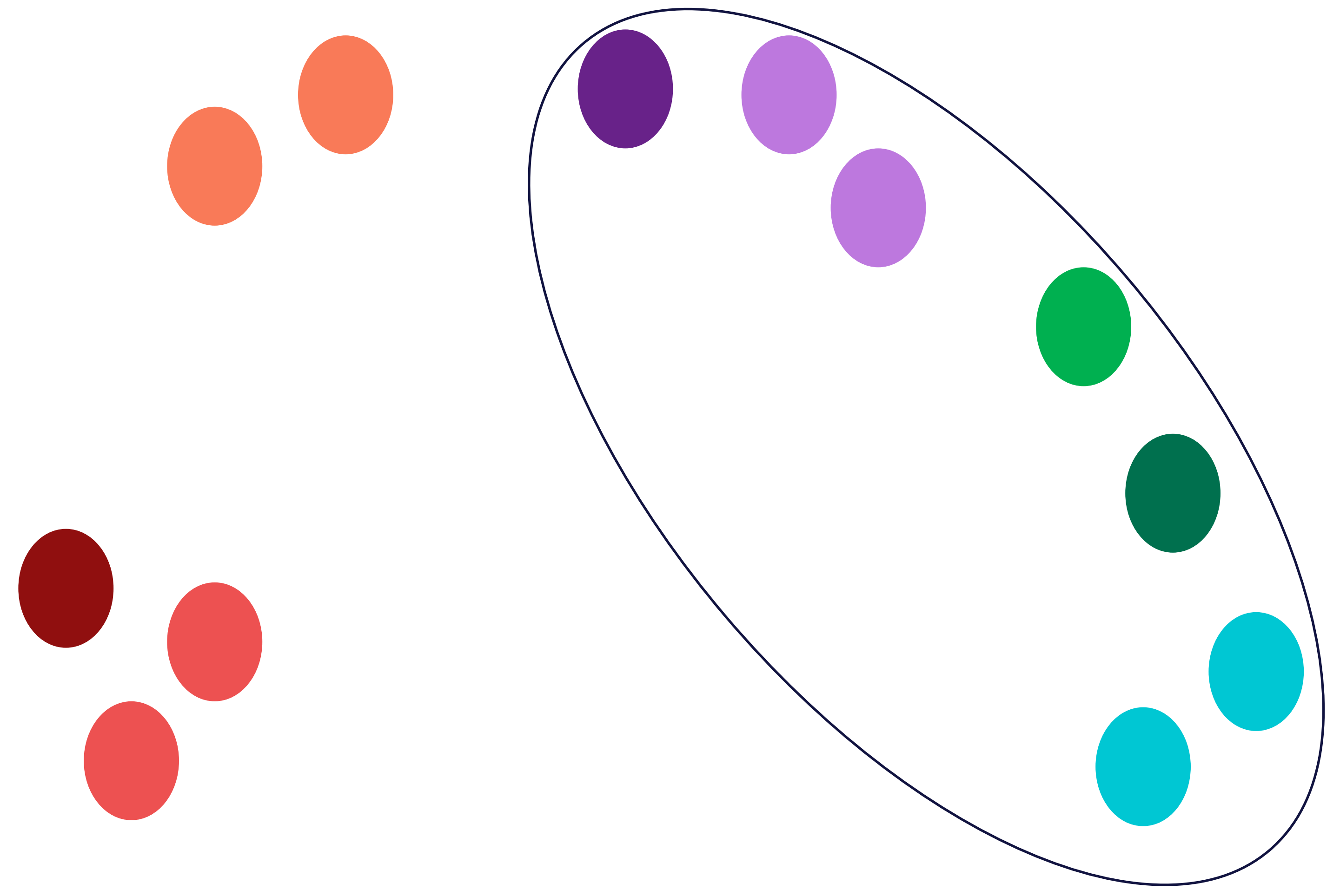
Step 1: Find a suitable similarity metric.
Step 2: Use the similarity metric to find the closest pair.
Step 3: Iterate and look for the next point closest to a point belonging to a cluster.
Step 4: Iterate and look for the next point closest to a point belonging to a cluster.

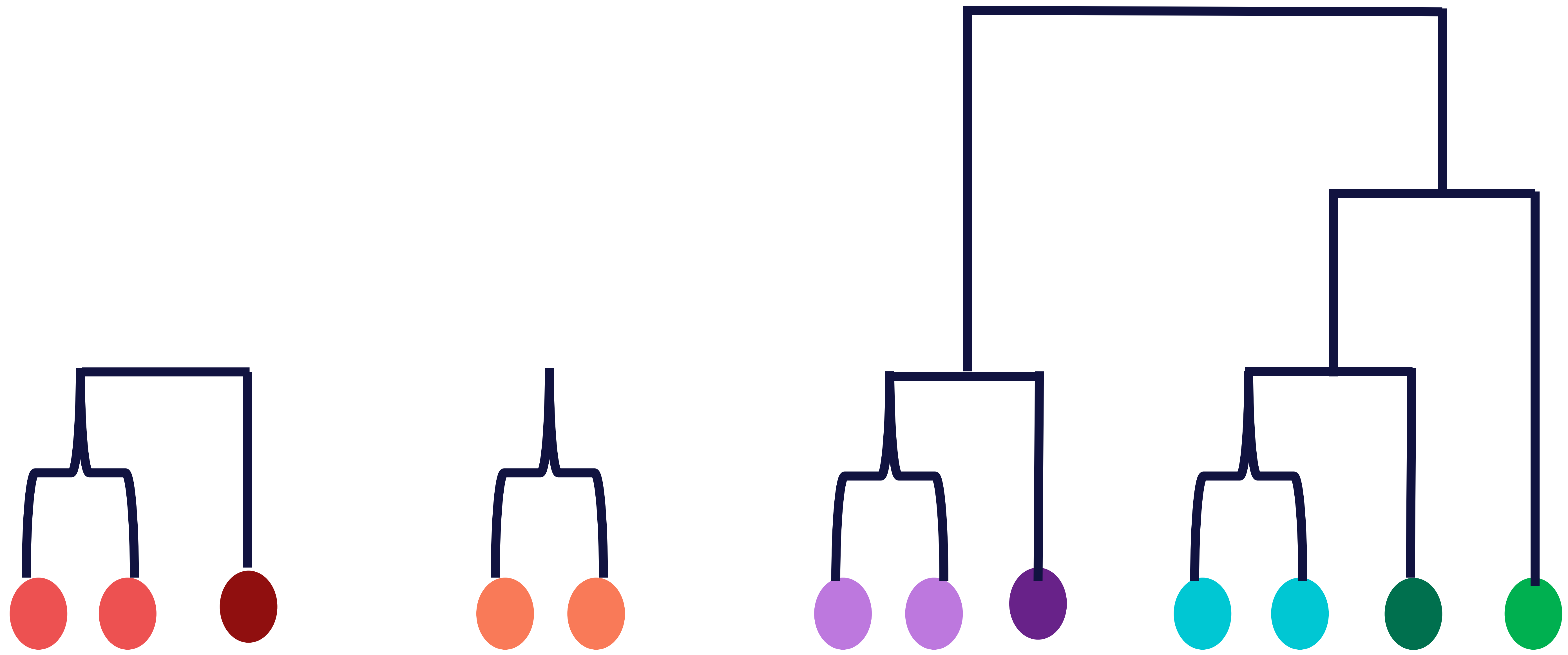




Agglomerative Hierarchical Clustering

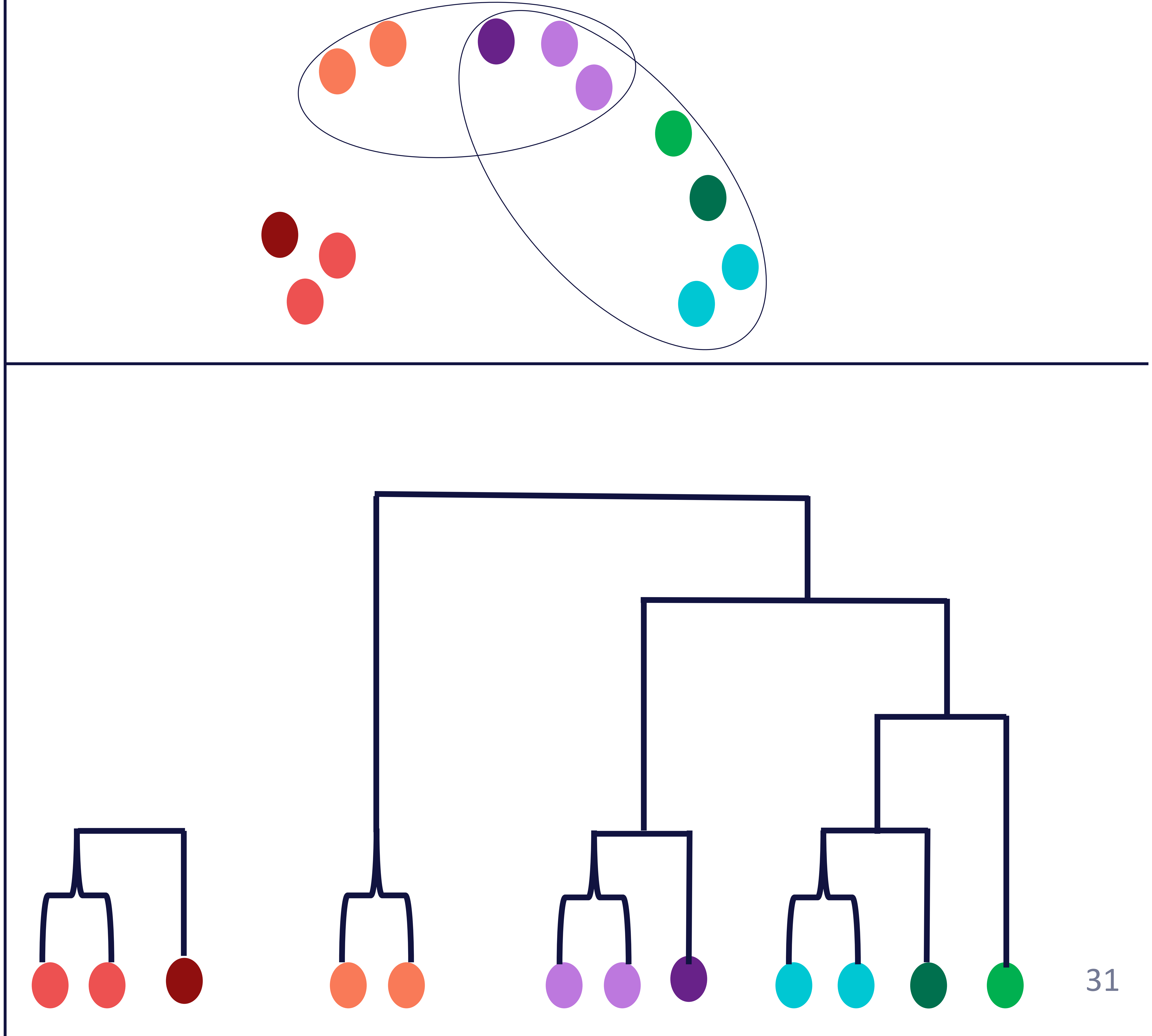
Step 1: Find a suitable similarity metric.
Step 2: Use the similarity metric to find the closest pair.
Step 3: Iterate and look for the next point closest to a point belonging to a cluster.
Step 4: Iterate and look for the next point closest to a point belonging to a cluster.
Step 5: Iterate and look for the next point closest to a point belonging to a cluster.





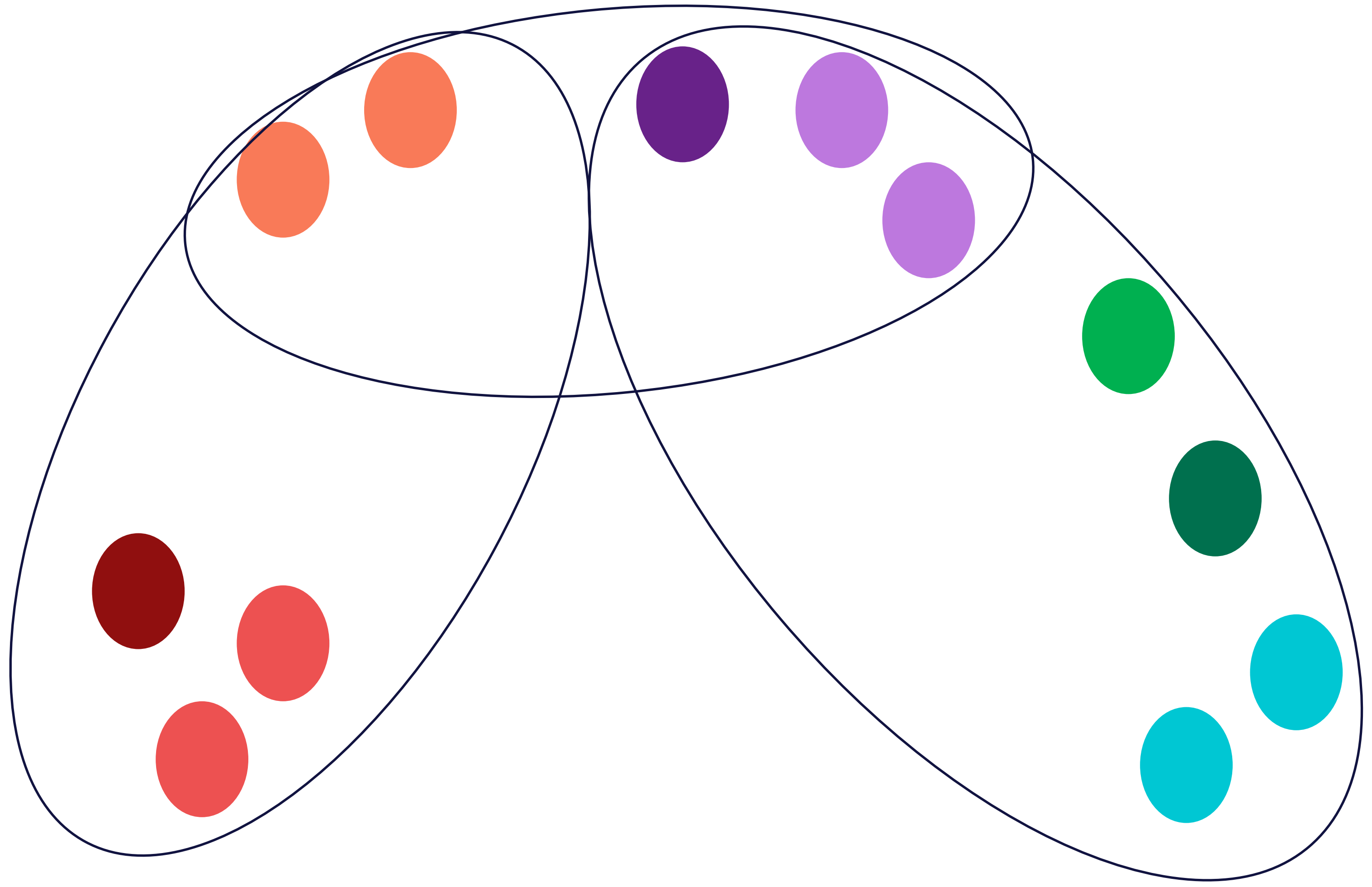
Agglomerative Hierarchical Clustering

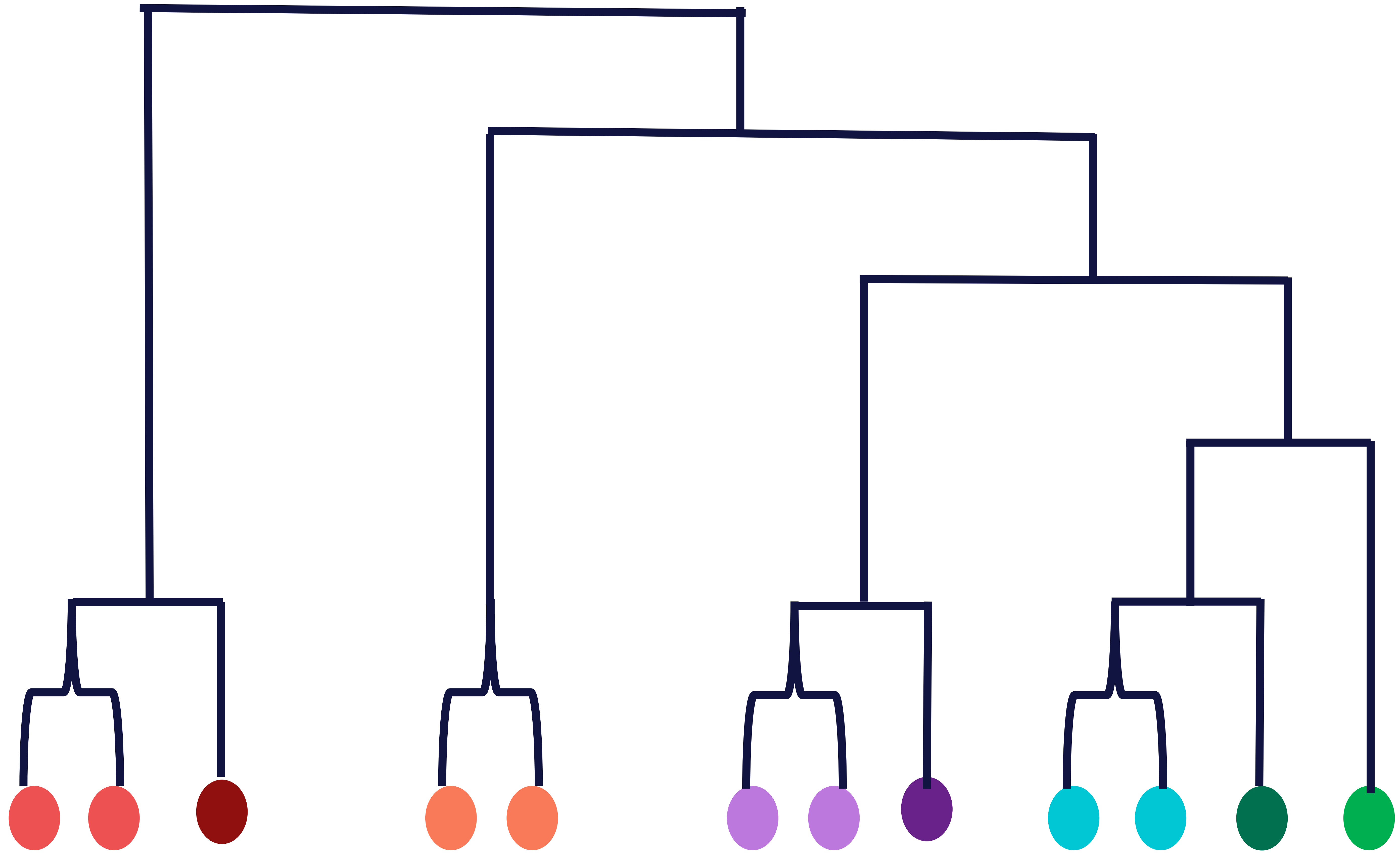
Step 1: Find a suitable similarity metric.
Step 2: Use the similarity metric to find the closest pair.
Step 3: Iterate and look for the next point closest to a point belonging to a cluster.
Step 4: Iterate and look for the next point closest to a point belonging to a cluster.
Step 5: Iterate and look for the next point closest to a point belonging to a cluster.
Step 6: Iterate and look for the next point closest to a point belonging to a cluster.



Agglomerative Hierarchical Clustering

Step 1: Find a suitable similarity metric.
Step 2: Use the similarity metric to find the closest pair.
Step 3: Iterate and look for the next point closest to a point belonging to a cluster.
Step 4: Iterate and look for the next point closest to a point belonging to a cluster.
Step 5: Iterate and look for the next point closest to a point belonging to a cluster.
Step 6: Iterate and look for the next point closest to a point belonging to a cluster.
Step 7: Iterate and look for the next point closest to a point belonging to a cluster.



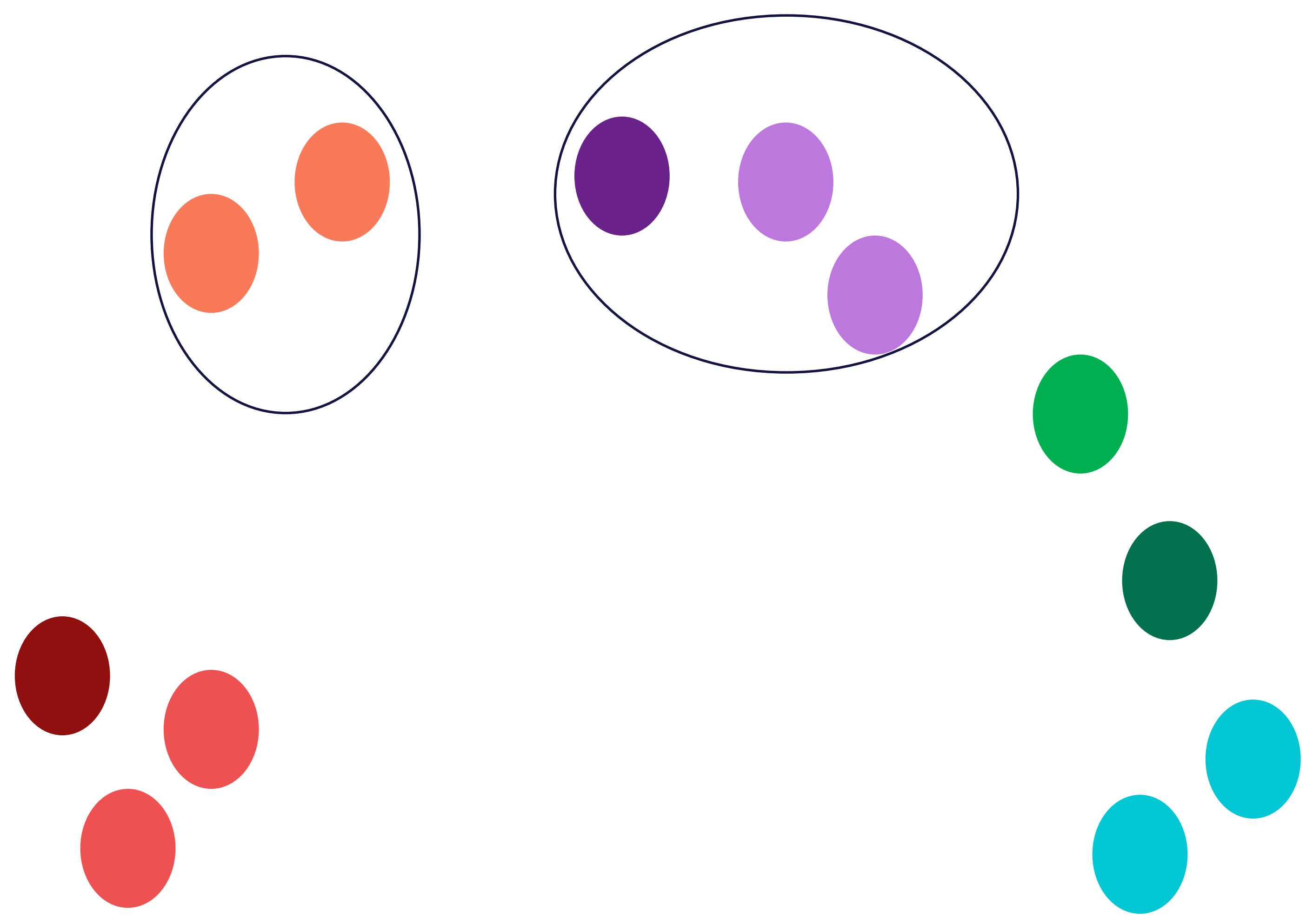


Agglomerative Hierarchical Clustering

Cluster Distance Type:

Single Linkage:

- Merges cluster if they are close somewhere.
- $D_{min}\left(C_i, C_j\right) = D_{min}(p_i, p_j), \forall\ p_i \in C_i, p_j \in C_j$

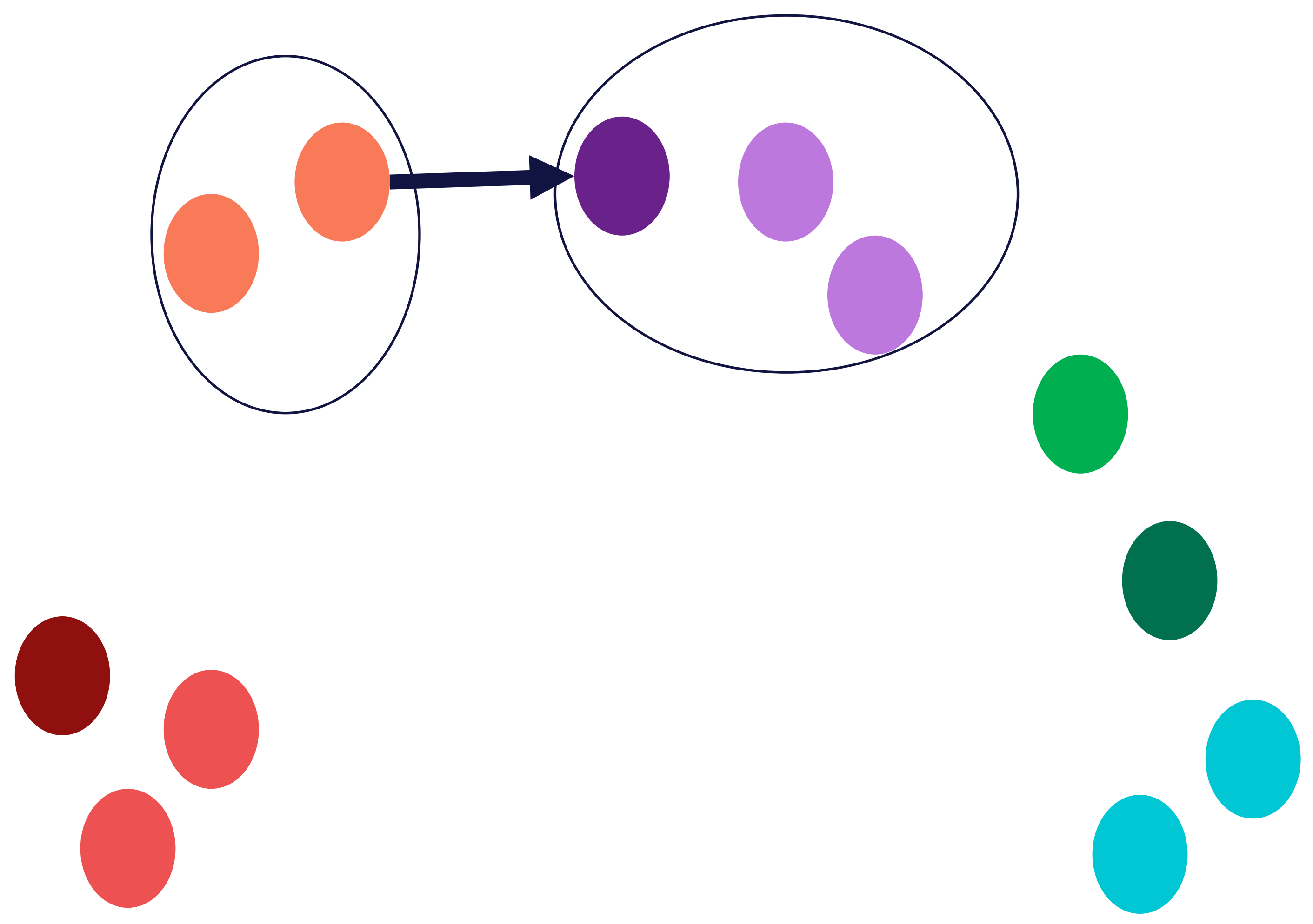


Agglomerative Hierarchical Clustering

Cluster Distance Type:

Single Linkage:

- Merges cluster if they are close somewhere.
- $D_{min}(C_i, C_j) = D_{min}(p_i, p_j), \forall\ p_i \in C_i, p_j \in C_j$

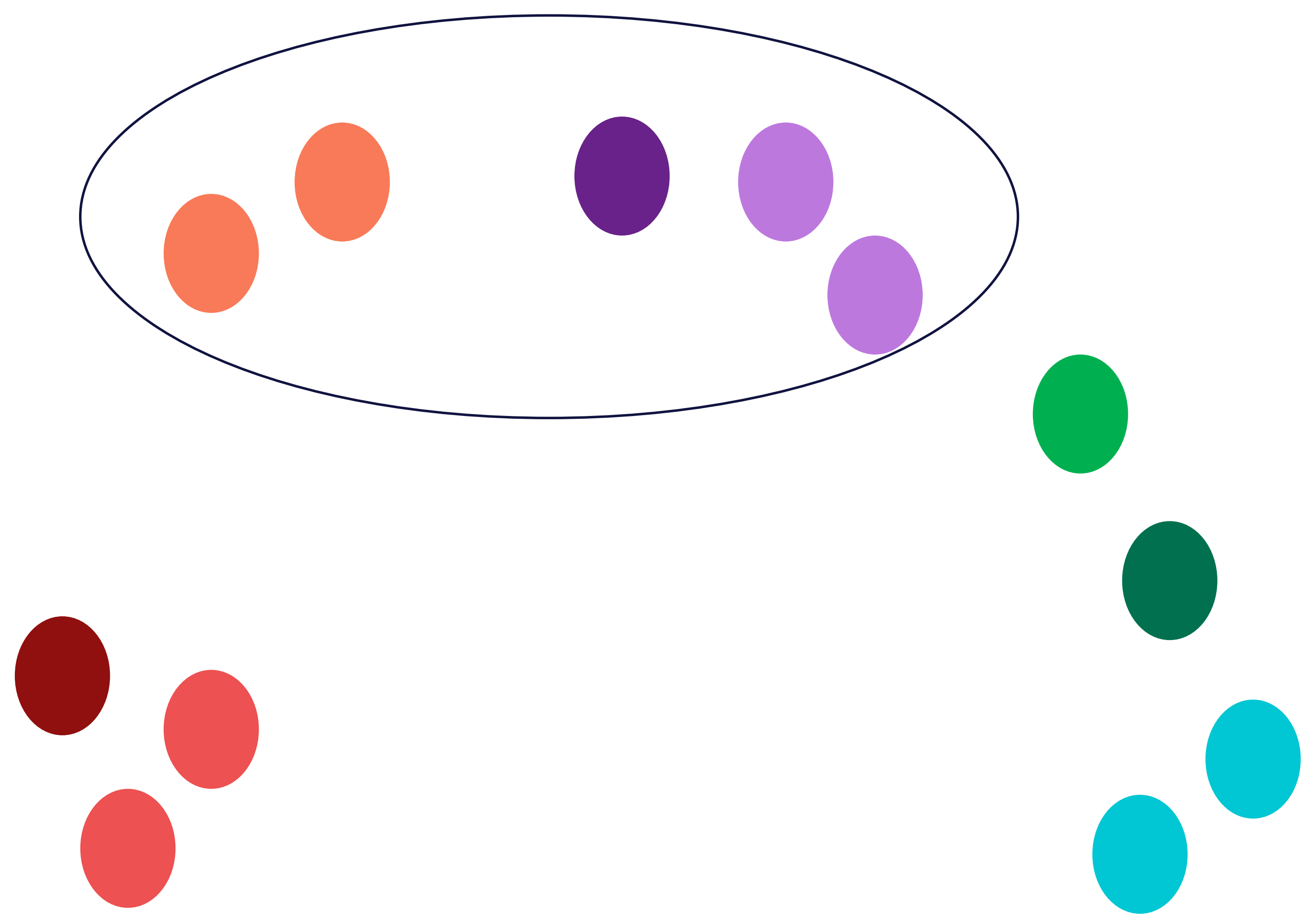


Agglomerative Hierarchical Clustering

Cluster Distance Type:

Single Linkage:

- Merges cluster if there are points that are close somewhere.
- $D_{min}\left(C_i, C_j\right) = D_{min}(p_i, p_j), \forall\ p_i \in C_i, p_j \in C_j$
- Produces a spanning tree

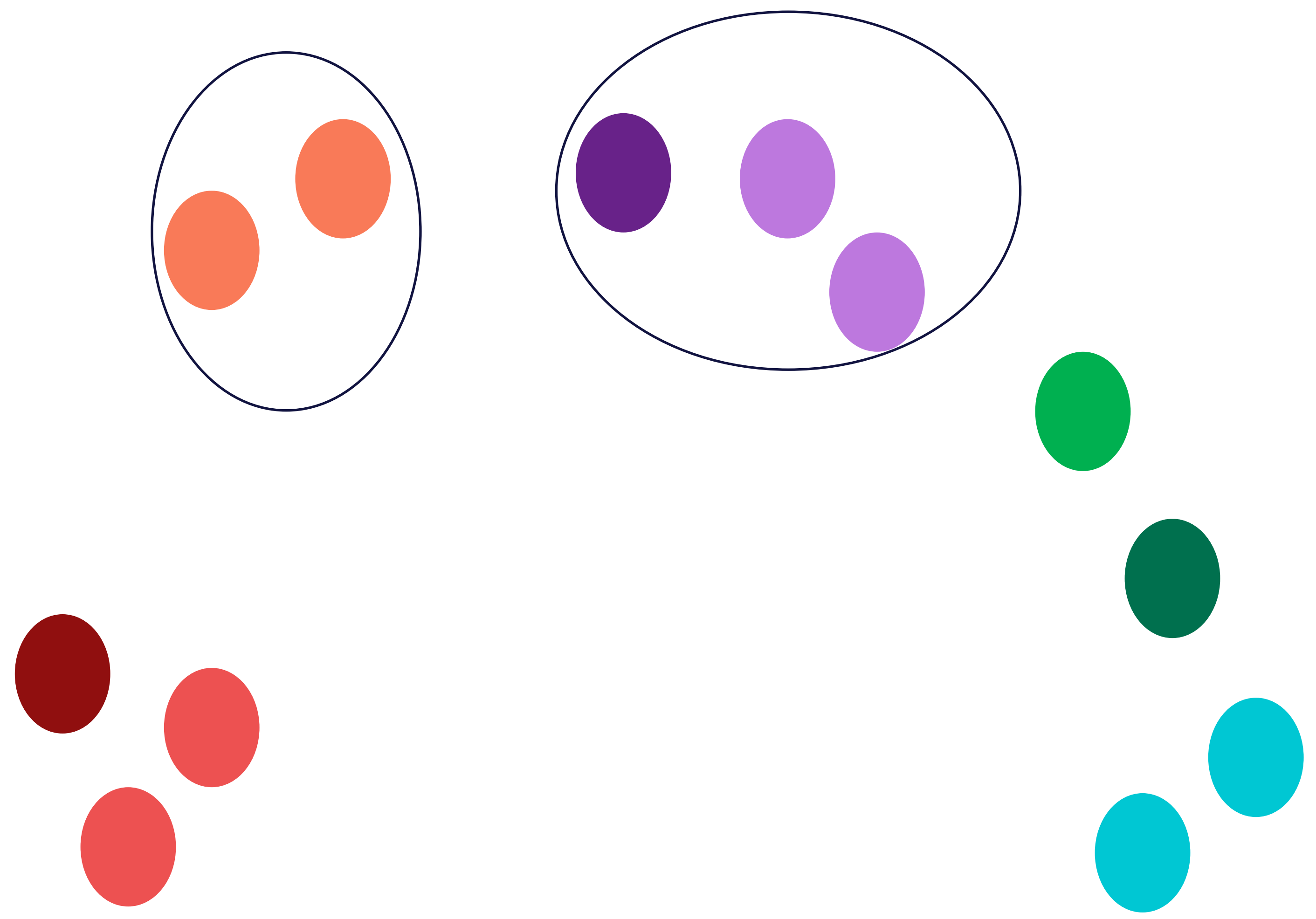


Agglomerative Hierarchical Clustering

Cluster Distance Type:

Complete Linkage:

- Merges cluster if they are close somewhere.
- $D_{max}\left(C_i, C_j\right) = D_{max}(p_i, p_j), \forall\ p_i \in C_i, p_j \in C_j$

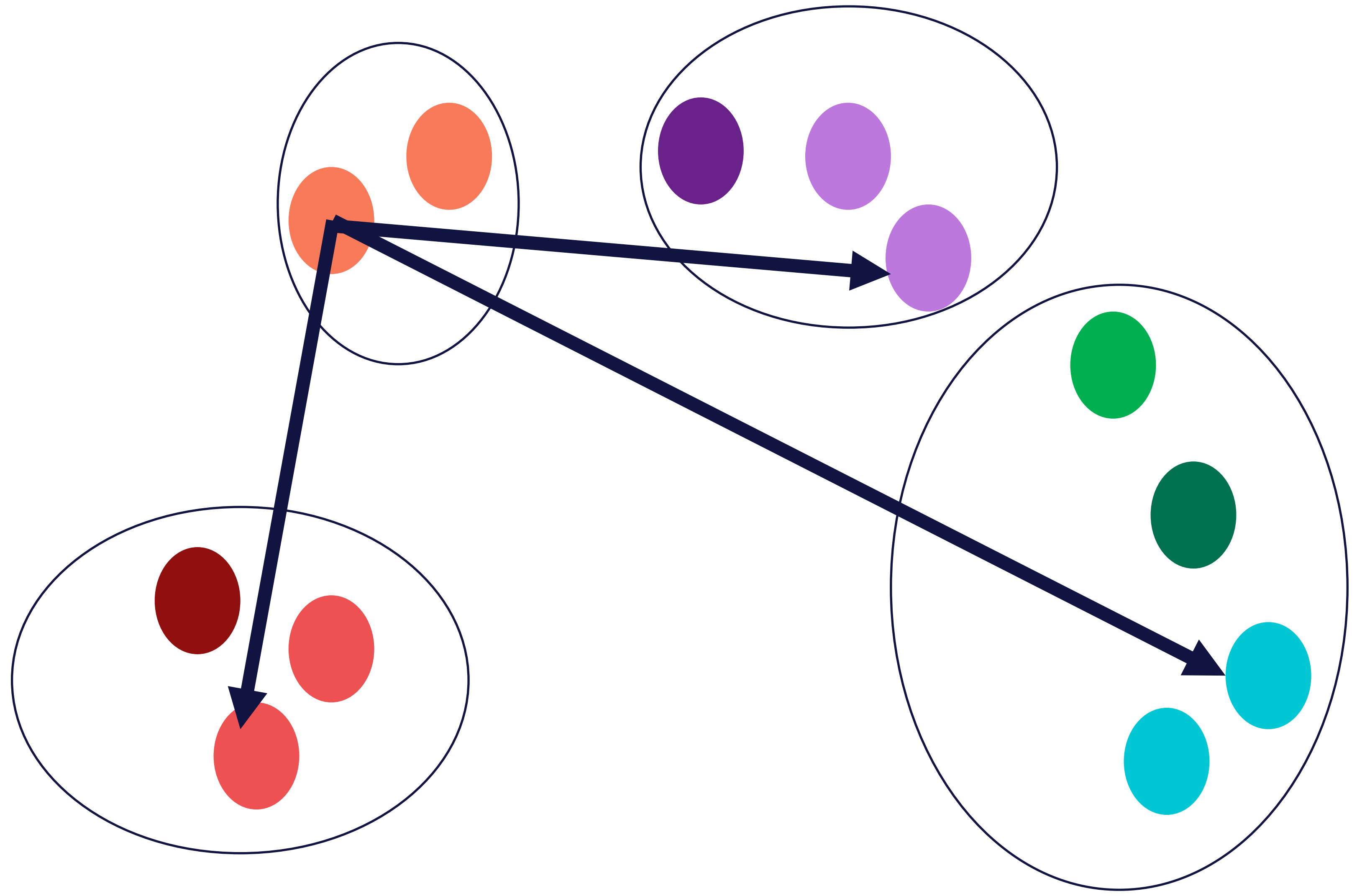


Agglomerative Hierarchical Clustering

Cluster Distance Type:

Complete Linkage:

- Merges cluster if they are close somewhere.
- $D_{max}\left(C_i, C_j\right) = D_{max}(p_i, p_j), \forall\ p_i \in C_i, p_j \in C_j$

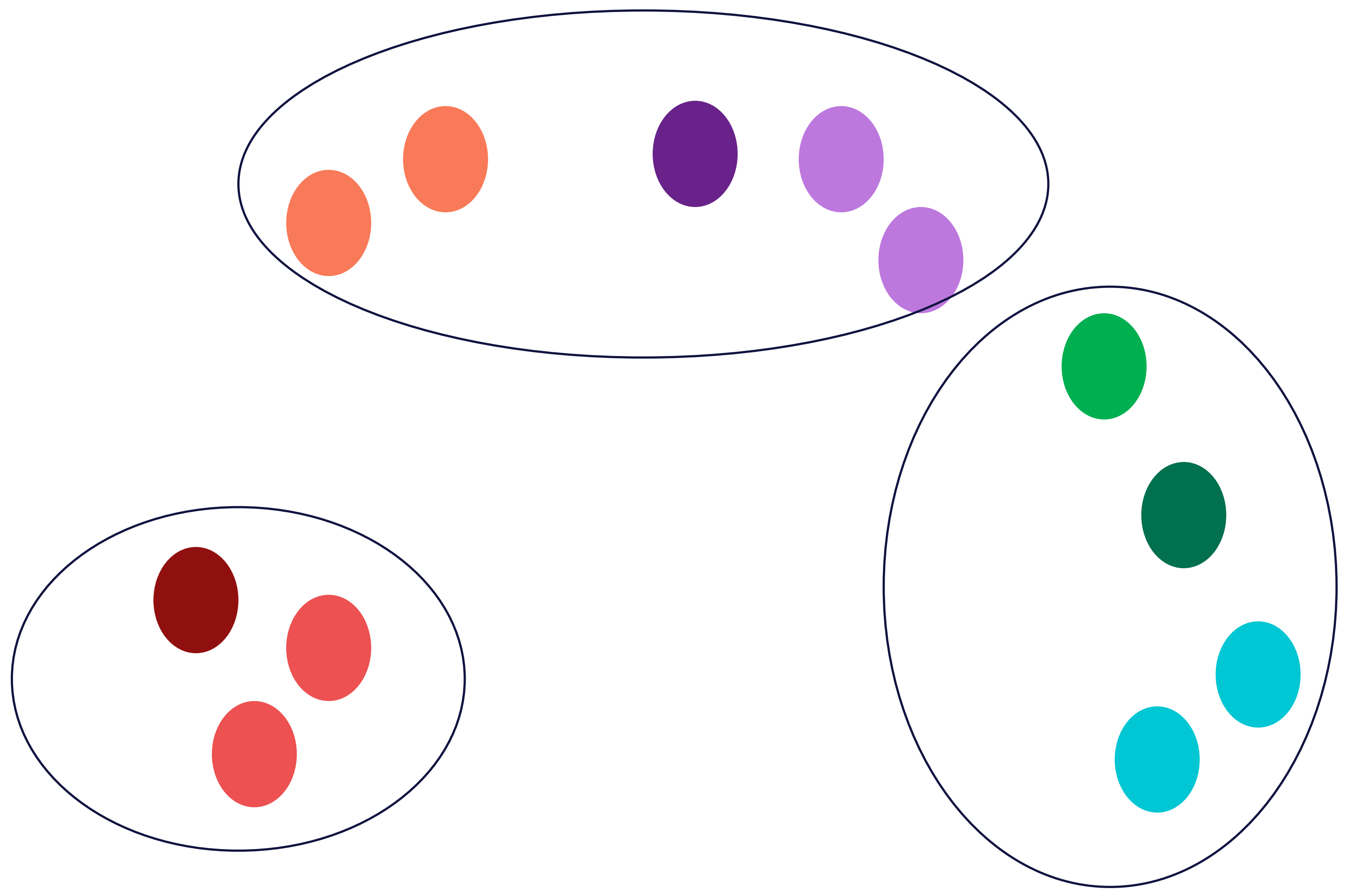


Agglomerative Hierarchical Clustering

Cluster Distance Type:

Complete Linkage:

- Merges cluster if they are close everywhere.
- $D_{max}(C_i, C_j) = D_{max}(p_i, p_j), \forall\ p_i \in C_i, p_j \in C_j$
- Find $Min[D_{max}(C_i, C_j)]$
- Forces “spherical cluster” (why?)

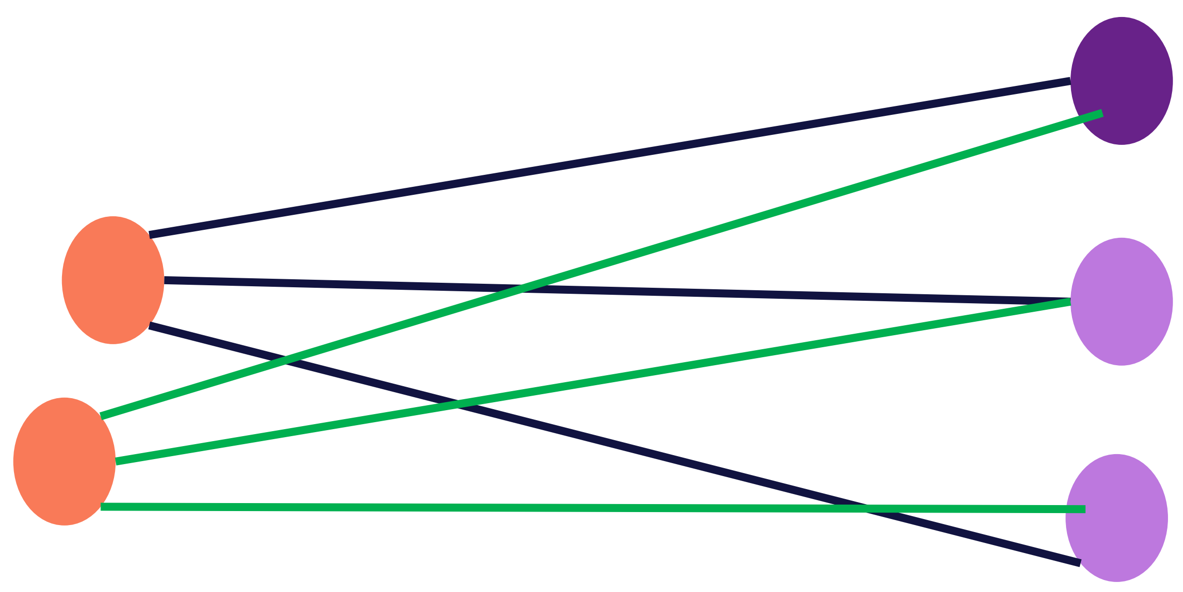


Agglomerative Hierarchical Clustering

Cluster Distance Type:

Average Linkage:

- Average of all pairwise distance.
- $D_{Ave}(C_i, C_j) = \frac{1}{n_{C_i}} \frac{1}{n_{C_j}} \sum_{p_i \in C_i} \sum_{p_j \in C_j} D(p_i, p_j)$ $, \forall\, p_i \in C_i, p_j \in C_j$
- Less affected by outliers.

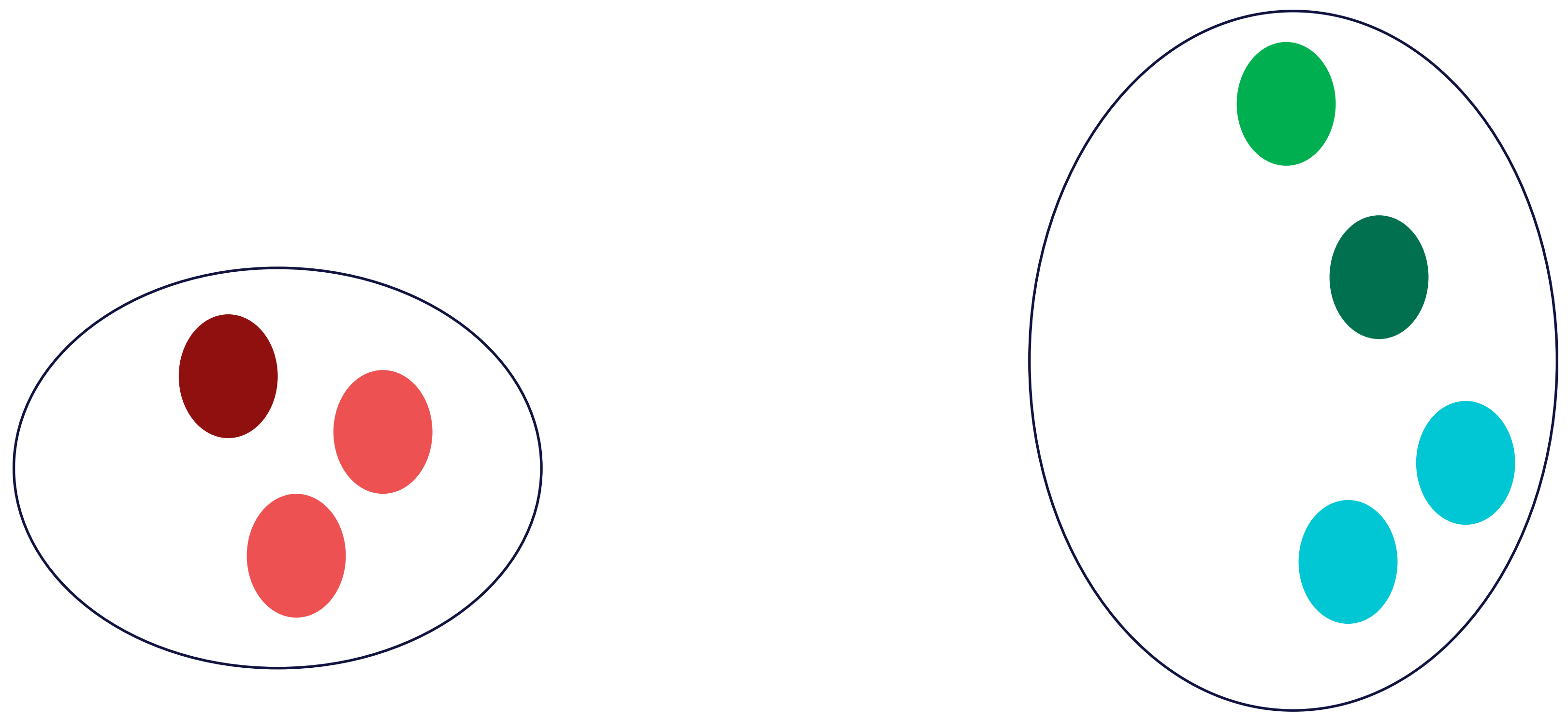


Agglomerative Hierarchical Clustering

Cluster Distance Type:

Centroid:

- Distance between centroids of two cluster.
- $D_{Centroid}\left(C_i, C_j\right) = D\left(\frac{1}{n_{C_i}}\sum_{p_i \in C_i}\overrightarrow{p_i}, \frac{1}{n_{C_j}}\sum_{p_j \in C_j}\overrightarrow{p_j}\right), \forall\ p_i \in C_i, p_j \in C_j$

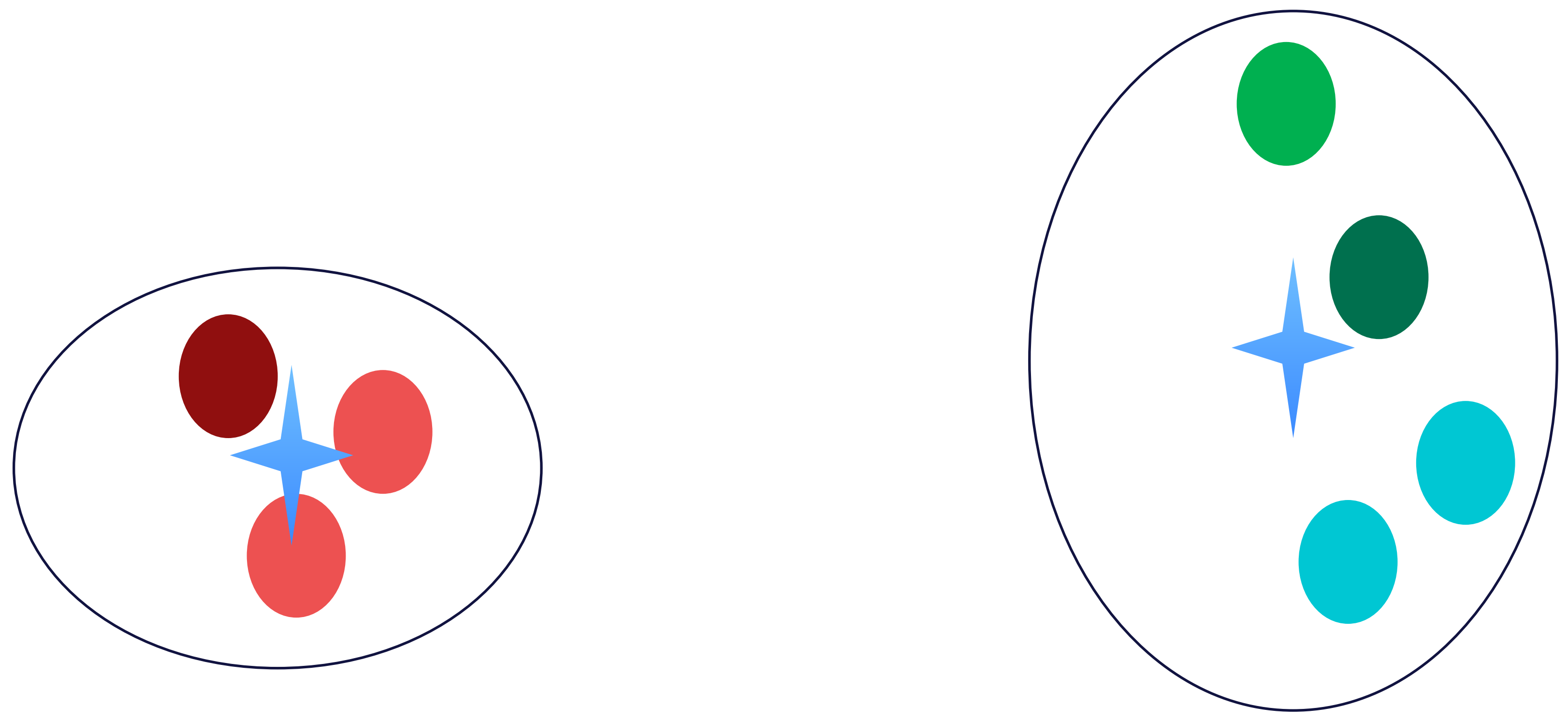


Agglomerative Hierarchical Clustering

Cluster Distance Type:

Centroid:

- Distance between centroids of two cluster.
- $D_{Centroid}\left(C_i, C_j\right) = D\left(\frac{1}{n_{C_i}}\sum_{p_i \in C_i}\overrightarrow{p_i}, \frac{1}{n_{C_j}}\sum_{p_j \in C_j}\overrightarrow{p_j}\right), \forall\ p_i \in C_i, p_j \in C_j$

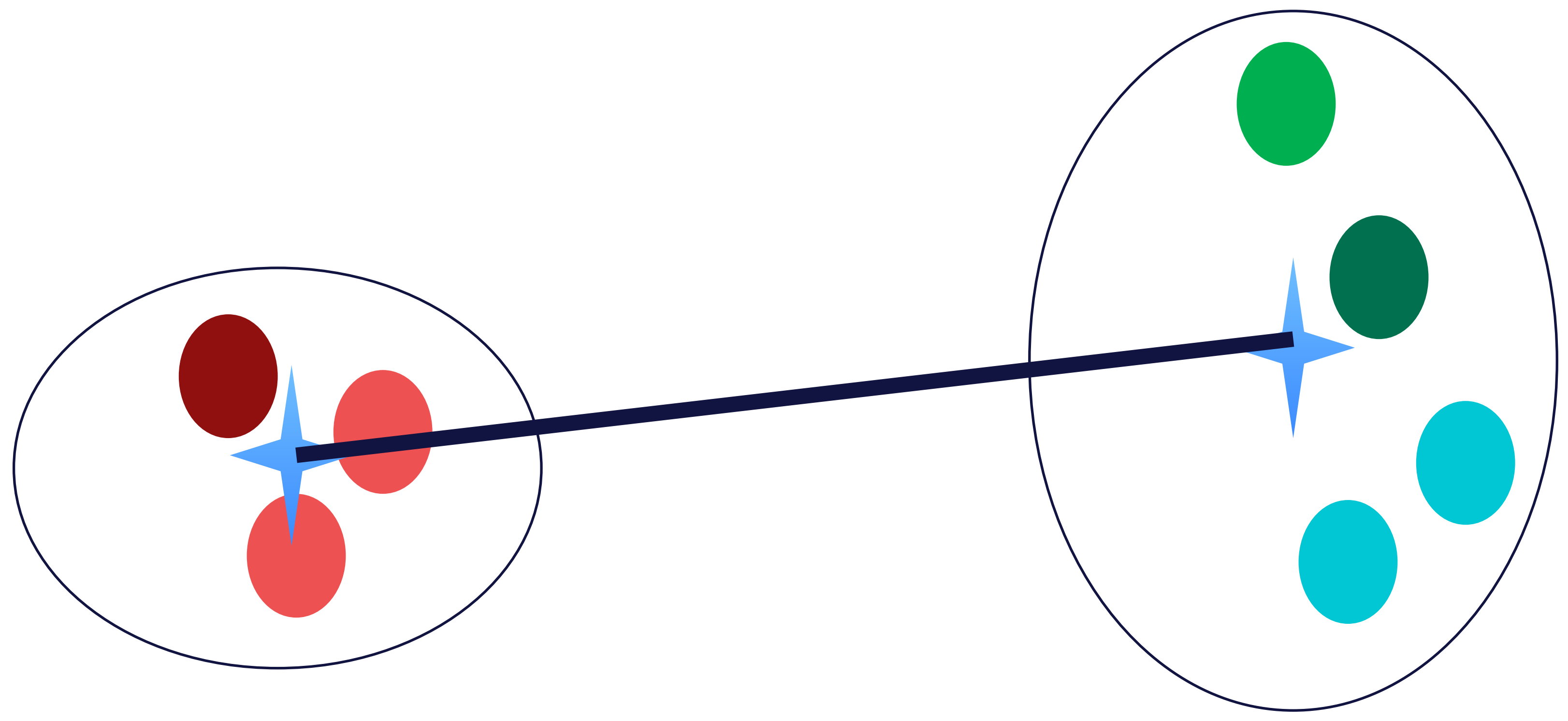


Agglomerative Hierarchical Clustering

Cluster Distance Type:

Centroid:

- Distance between centroids of two cluster.
- $D_{Centroid}\left(C_i, C_j\right) = D\left(\frac{1}{n_{C_i}} \sum_{p_i \in C_i} \overrightarrow{p_i}, \frac{1}{n_{C_j}} \sum_{p_j \in C_j} \overrightarrow{p_j}\right), \forall\, p_i \in C_i, p_j \in C_j$



Agglomerative Hierarchical Clustering

Cluster Distance Type:

Ward's Methods: Read about Ward's Method

?

?

DBSCAN

Density-Based Spatial Clustering of Applications with Noise

Parameters

- ε (epsilon): A distance threshold to be considered into a cluster.
- Min. Points: Number of points to consider a region as highly dense.

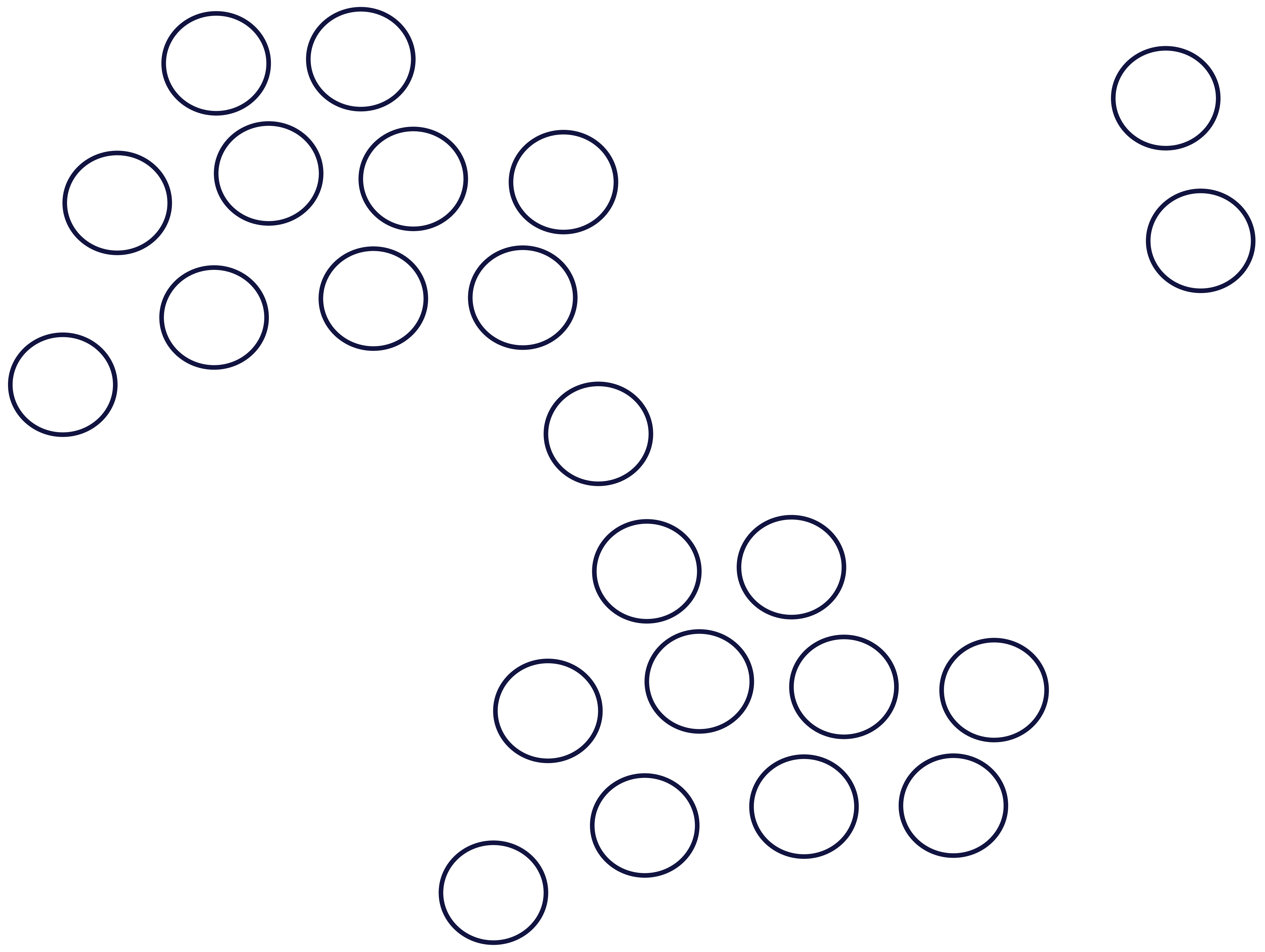
Point Classification

- Core Points: If the conditions of the parameters are met.
- Border Points: Points that are ε from the core points but does not meet the Min. Points requirements
- Noise Points: Parameter requirements are not met.

Performance metrics

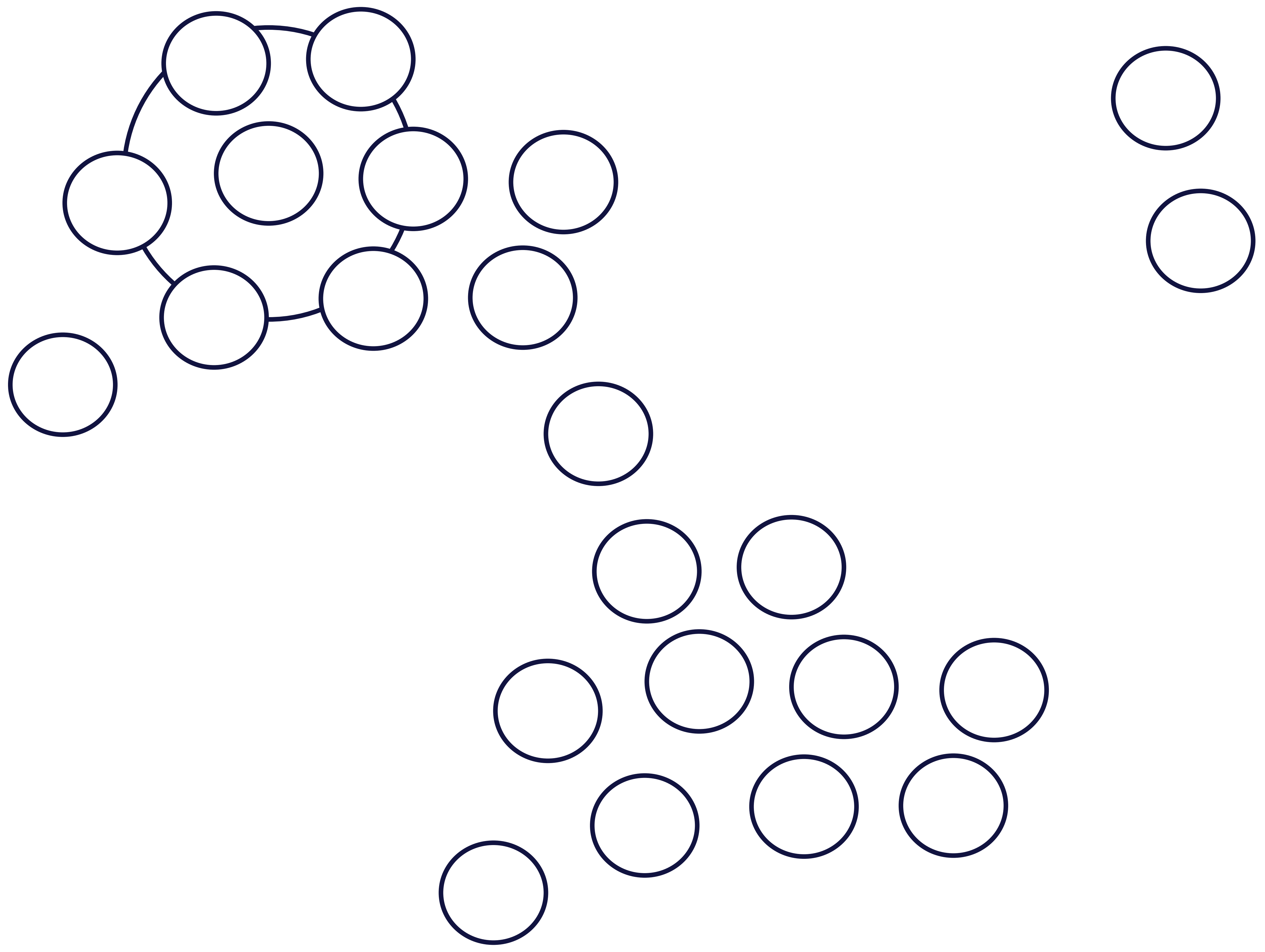
- No ground truth.
- Statistical measure of purity

Step 1: Choose an ε for the radius of a circle, and choose a minimum number of points to consider a cluster, say = 4.



Step 1: Choose an ε for the radius of a circle and choose a minimum number of points to consider a cluster, say = 4.

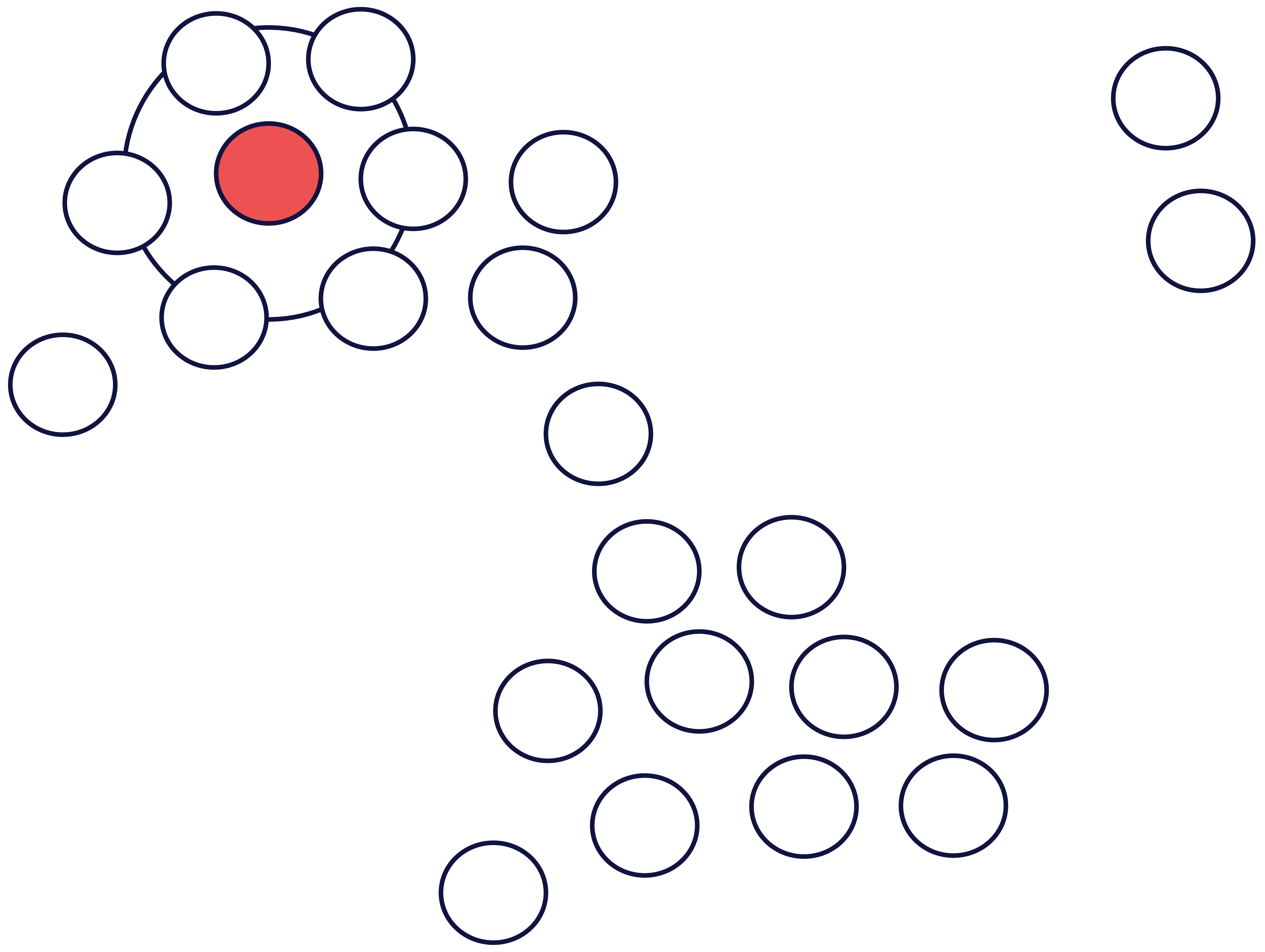
Step 2: Randomly pick a point, draw a circle with radius ε, then check the number of intersected points.



Step 1: Choose an ε for the radius of a circle and choose a minimum number of points to consider a cluster, say = 4.

Step 2: Randomly pick a point, draw a circle with radius ε, then check the number of intersected points.

Step 3: If the conditions are satisfied (min.point <= intersected points), then it’s a core point.

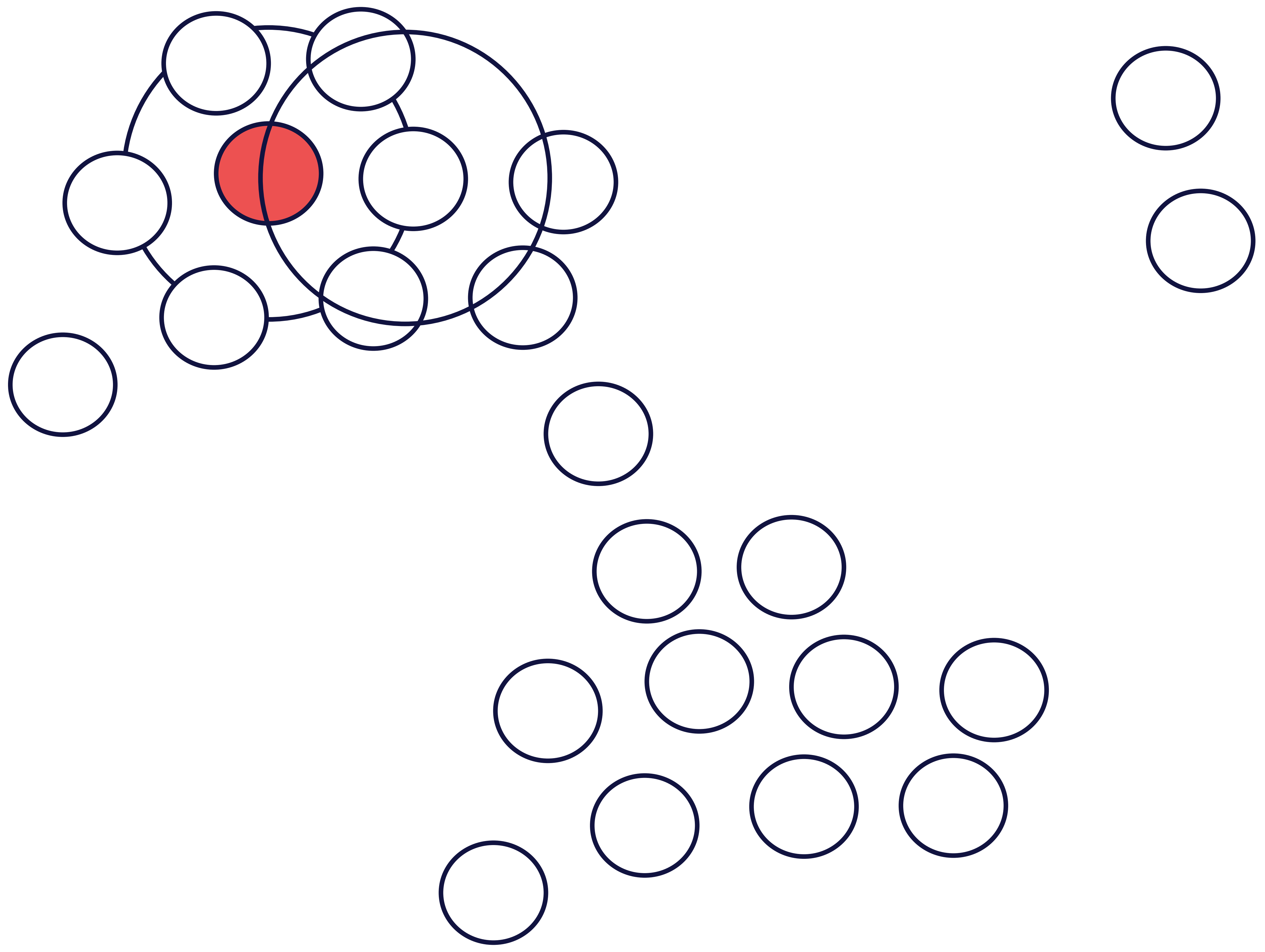


Step 1: Choose an ε for the radius of a circle and choose a minimum number of points to consider a cluster, say = 4.

Step 2: Randomly pick a point, draw a circle with radius ε, then check the number of intersected points.

Step 3: If the conditions are satisfied (min.point <= intersected points), then it’s a core point.

Step 4: Repeat steps 2 and 3 to all points.

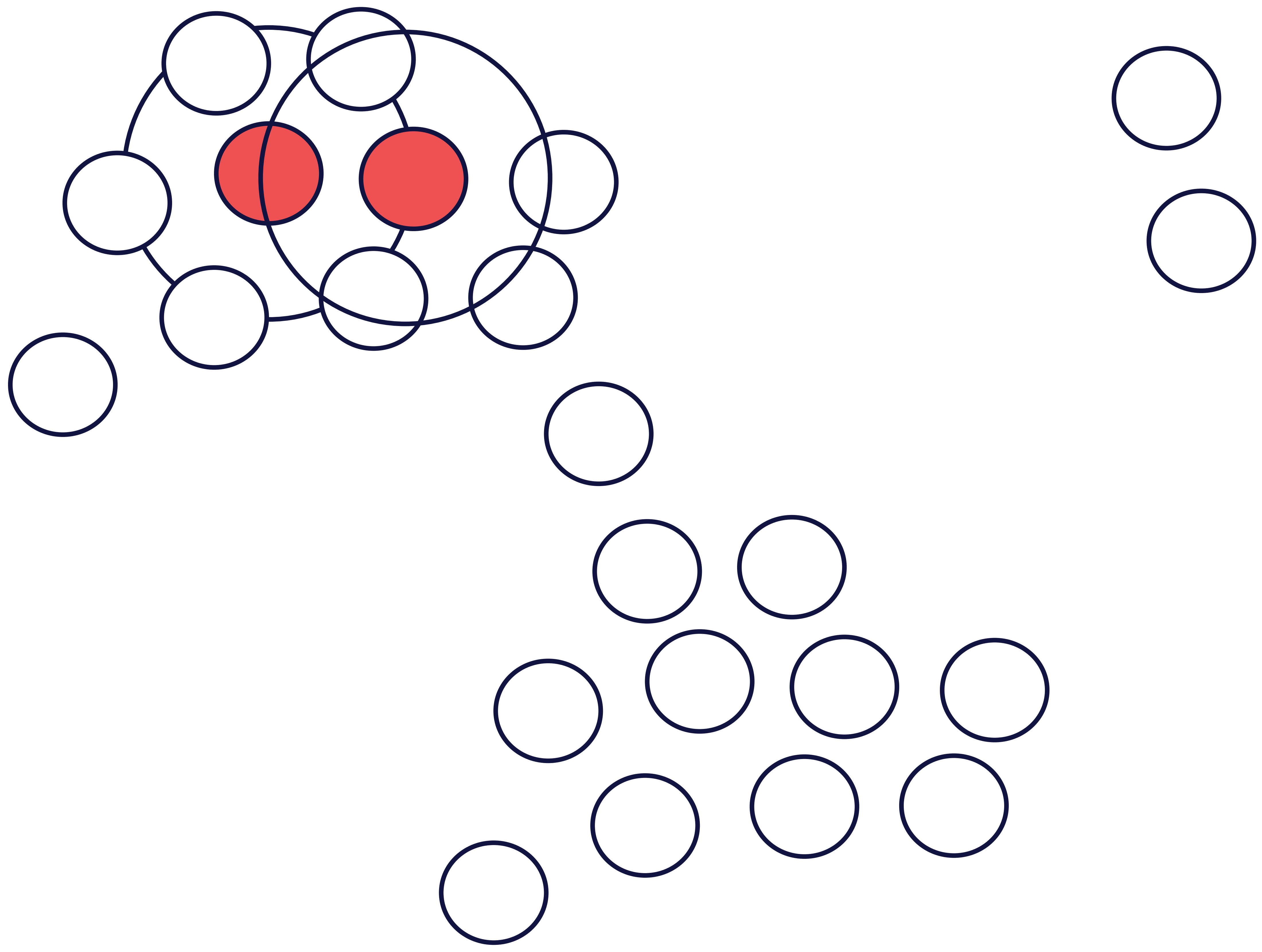


Step 1: Choose an ε for the radius of a circle and choose a minimum number of points to consider a cluster, say = 4.

Step 2: Randomly pick a point, draw a circle with radius ε, then check the number of intersected points.

Step 3: If the conditions are satisfied (min.point <= intersected points), then it's a core point.

Step 4: Repeat steps 2 and 3 to all points.

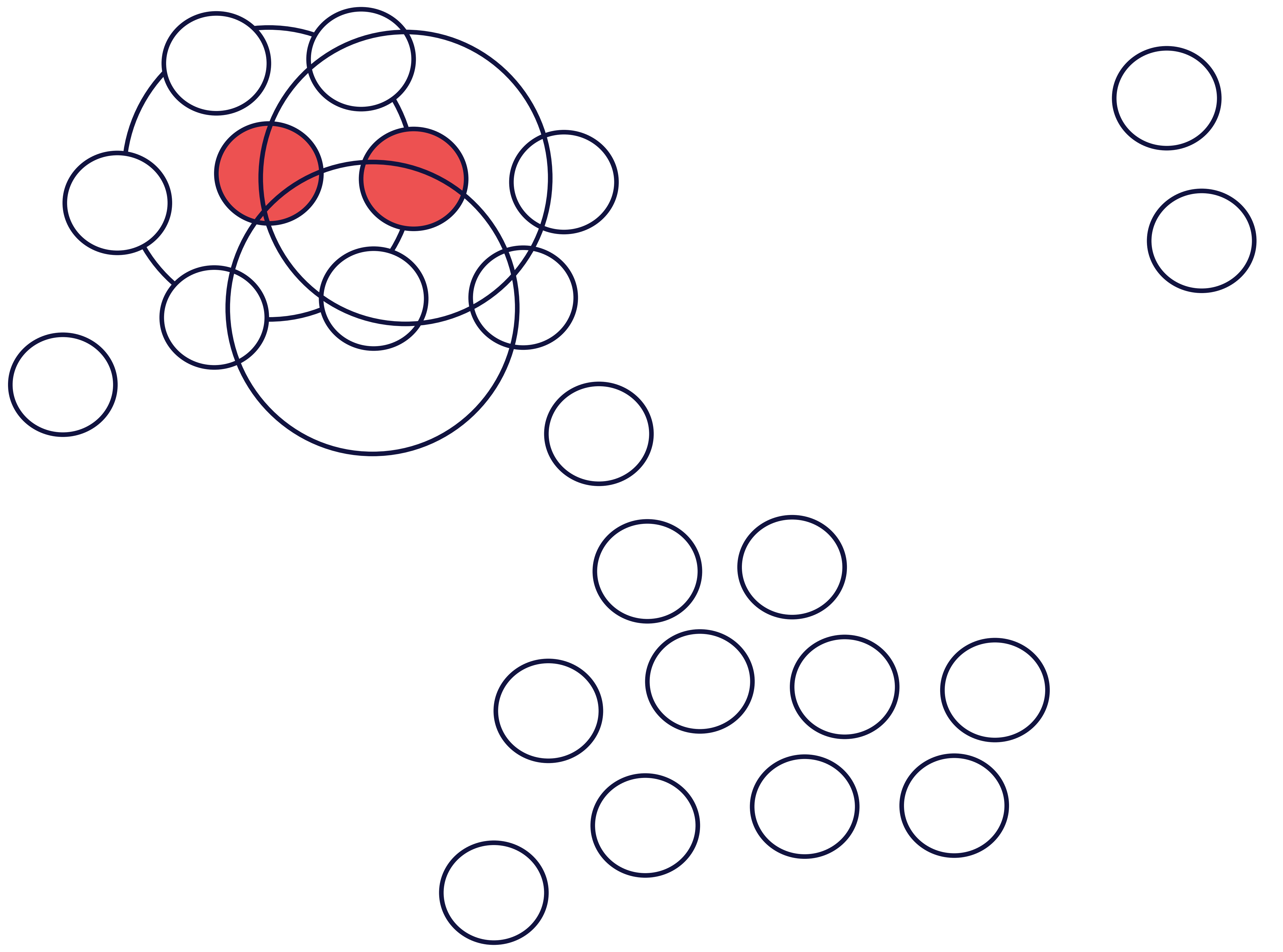


Step 1: Choose an ε for the radius of a circle and choose a minimum number of points to consider a cluster, say = 4.

Step 2: Randomly pick a point, draw a circle with radius ε, then check the number of intersected points.

Step 3: If the conditions are satisfied (min.point <= intersected points), then it's a core point.

Step 4: Repeat steps 2 and 3 to all points.

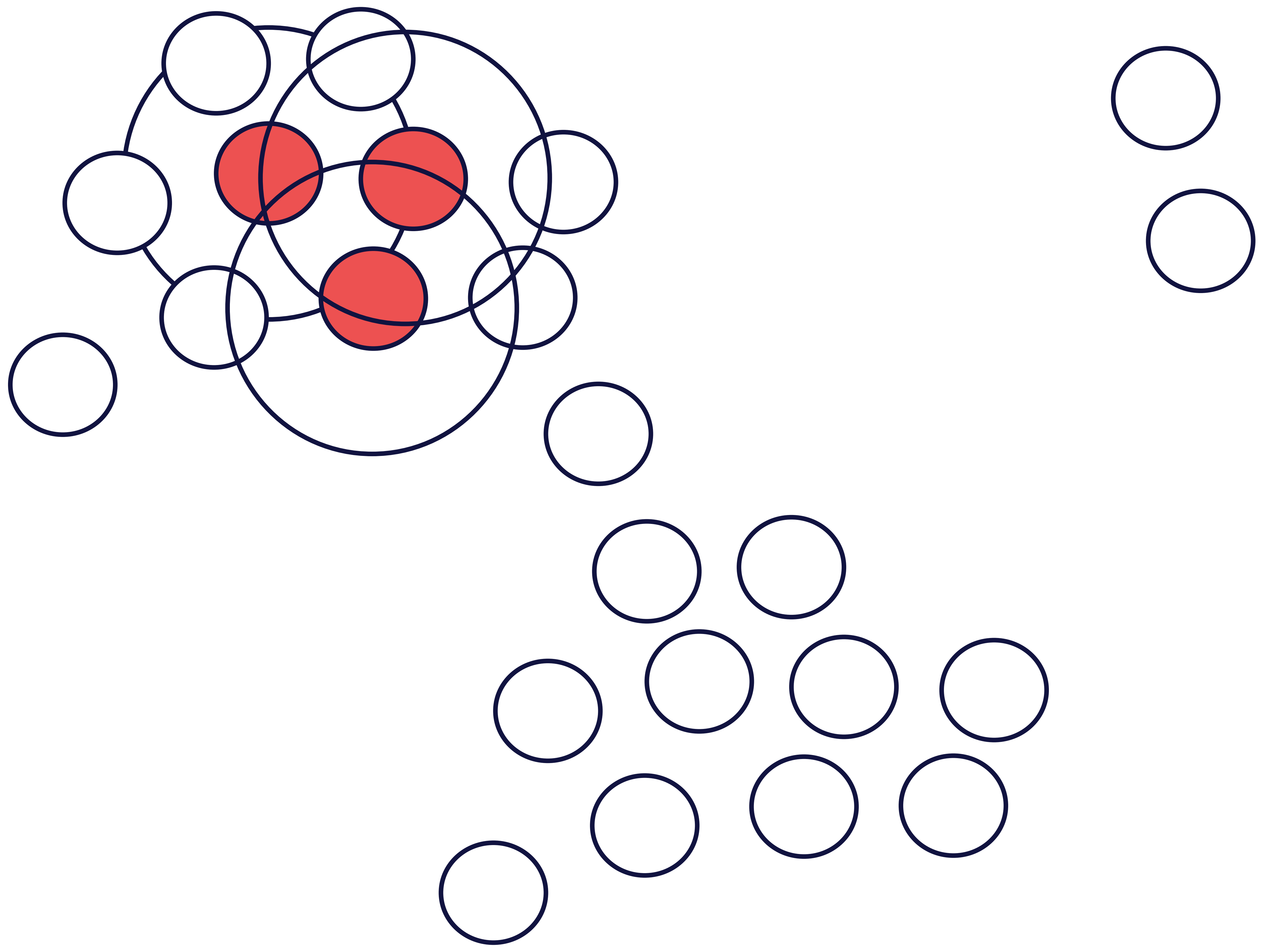


Step 1: Choose an ε for the radius of a circle and choose a minimum number of points to consider a cluster, say = 4.

Step 2: Randomly pick a point, draw a circle with radius ε, then check the number of intersected points.

Step 3: If the conditions are satisfied (min.point <= intersected points), then it’s a core point.

Step 4: Repeat steps 2 and 3 to all points.

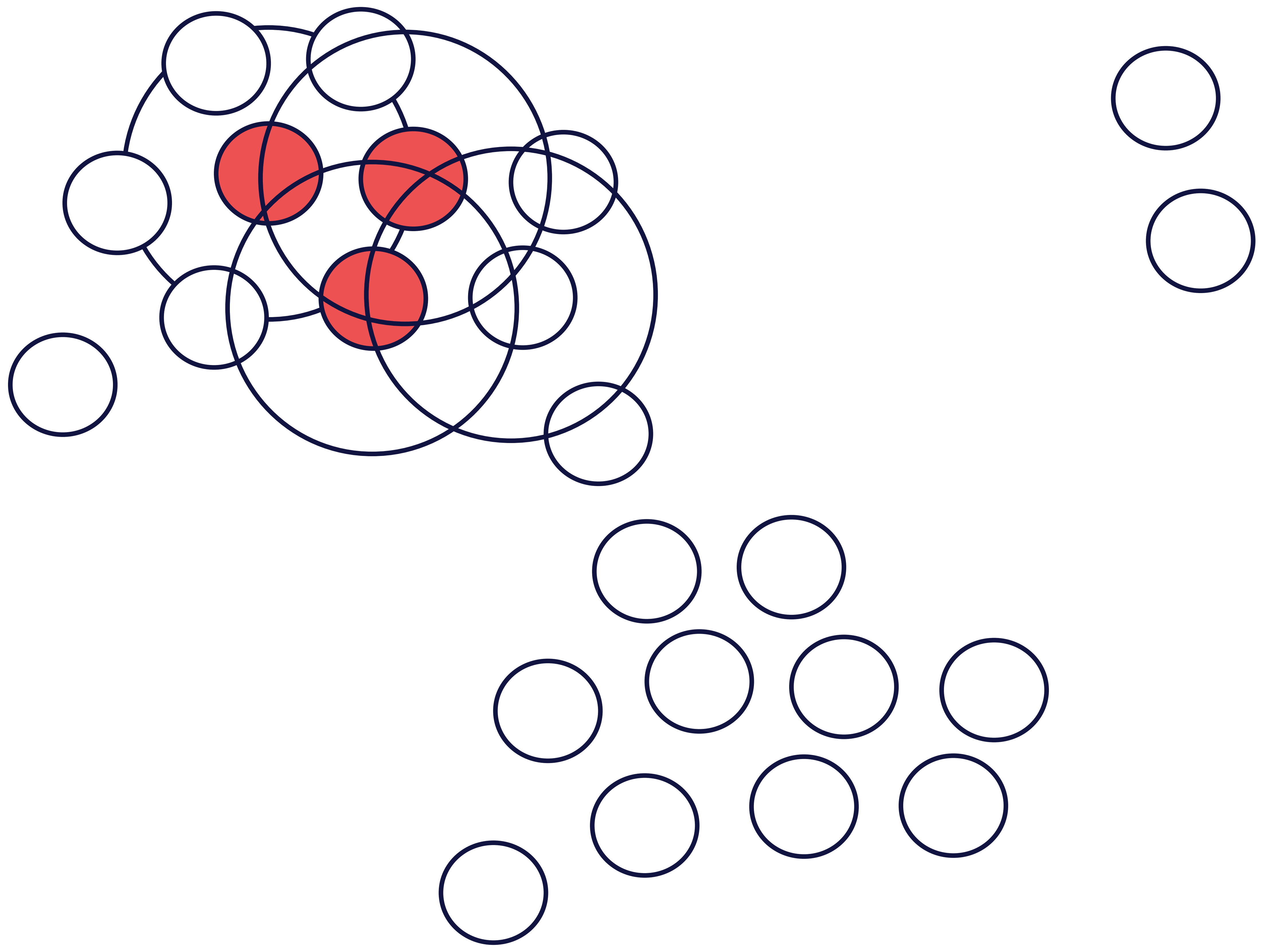


Step 1: Choose an ε for the radius of a circle and choose a minimum number of points to consider a cluster, say = 4.

Step 2: Randomly pick a point, draw a circle with radius ε, then check the number of intersected points.

Step 3: If the conditions are satisfied (min.point <= intersected points), then it’s a core point.

Step 4: Repeat steps 2 and 3 to all points.

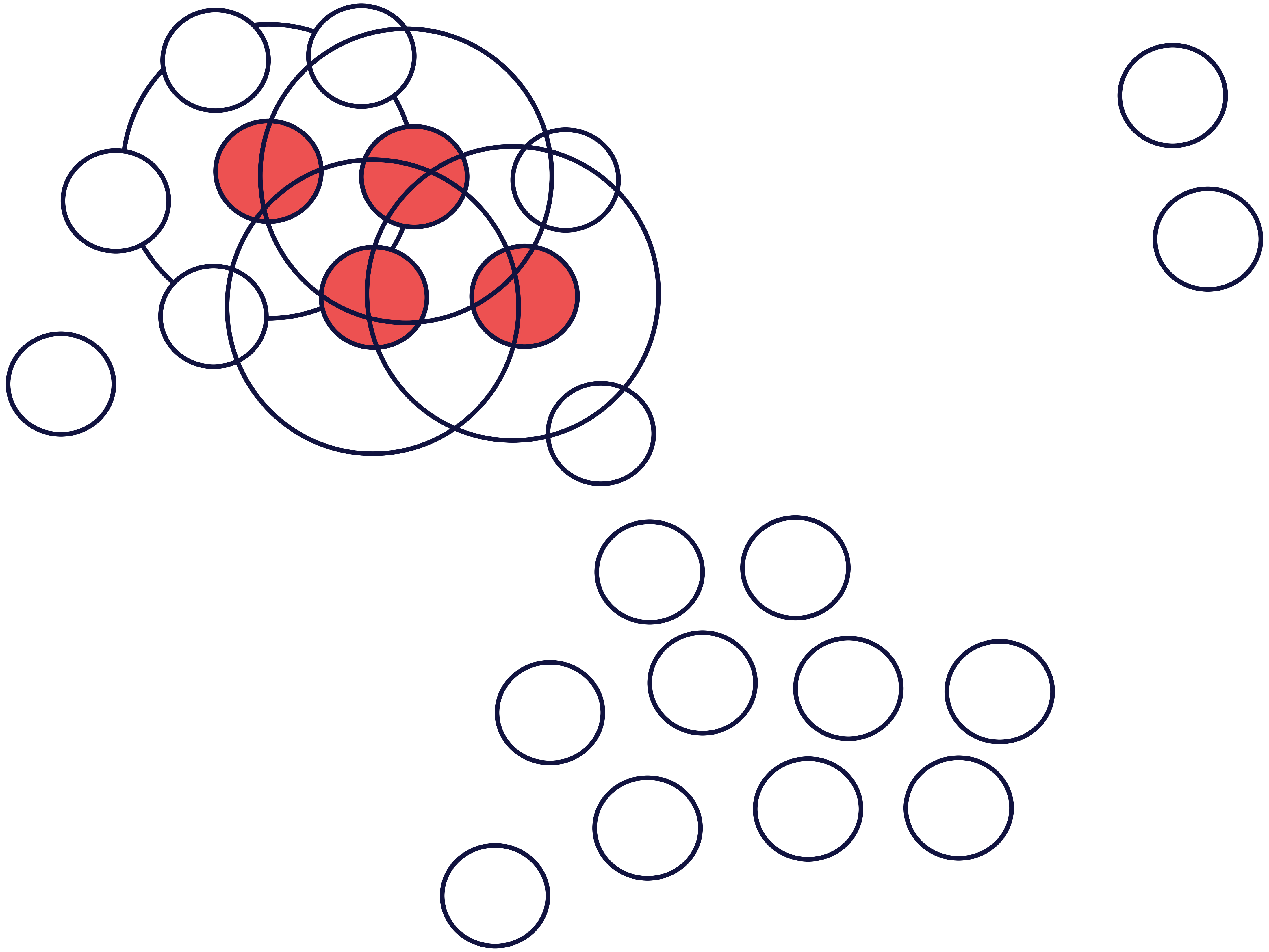


Step 1: Choose an ε for the radius of a circle and choose a minimum number of points to consider a cluster, say = 4.

Step 2: Randomly pick a point, draw a circle with radius ε, then check the number of intersected points.

Step 3: If the conditions are satisfied (min.point <= intersected points), then it's a core point.

Step 4: Repeat steps 2 and 3 to all points.

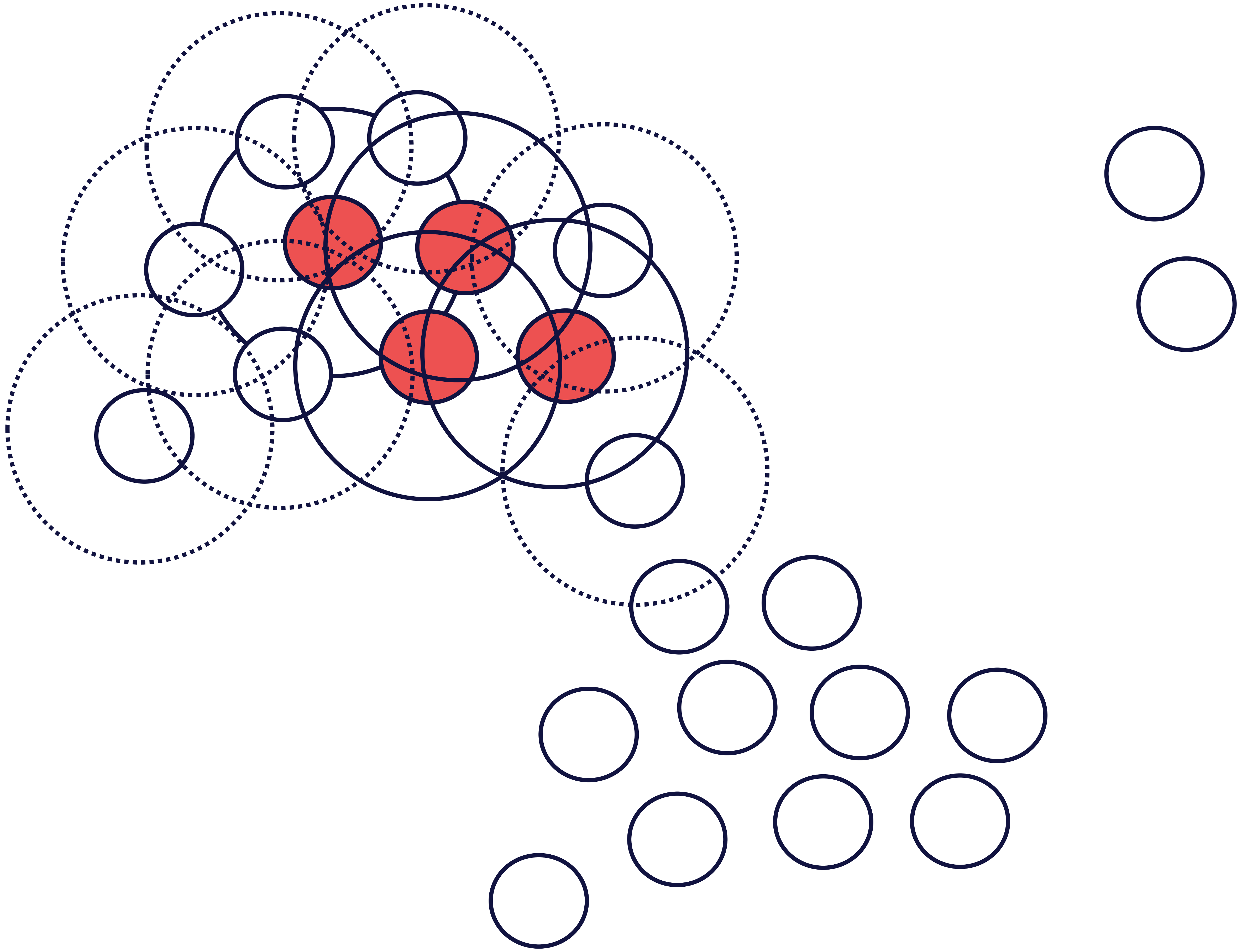


Step 1: Choose an ε for the radius of a circle and choose a minimum number of points to consider a cluster, say = 4.

Step 2: Randomly pick a point, draw a circle with radius ε, then check the number of intersected points.

Step 3: If the conditions are satisfied (min.point <= intersected points), then it’s a core point.

Step 4: Repeat steps 2 and 3 to all points.



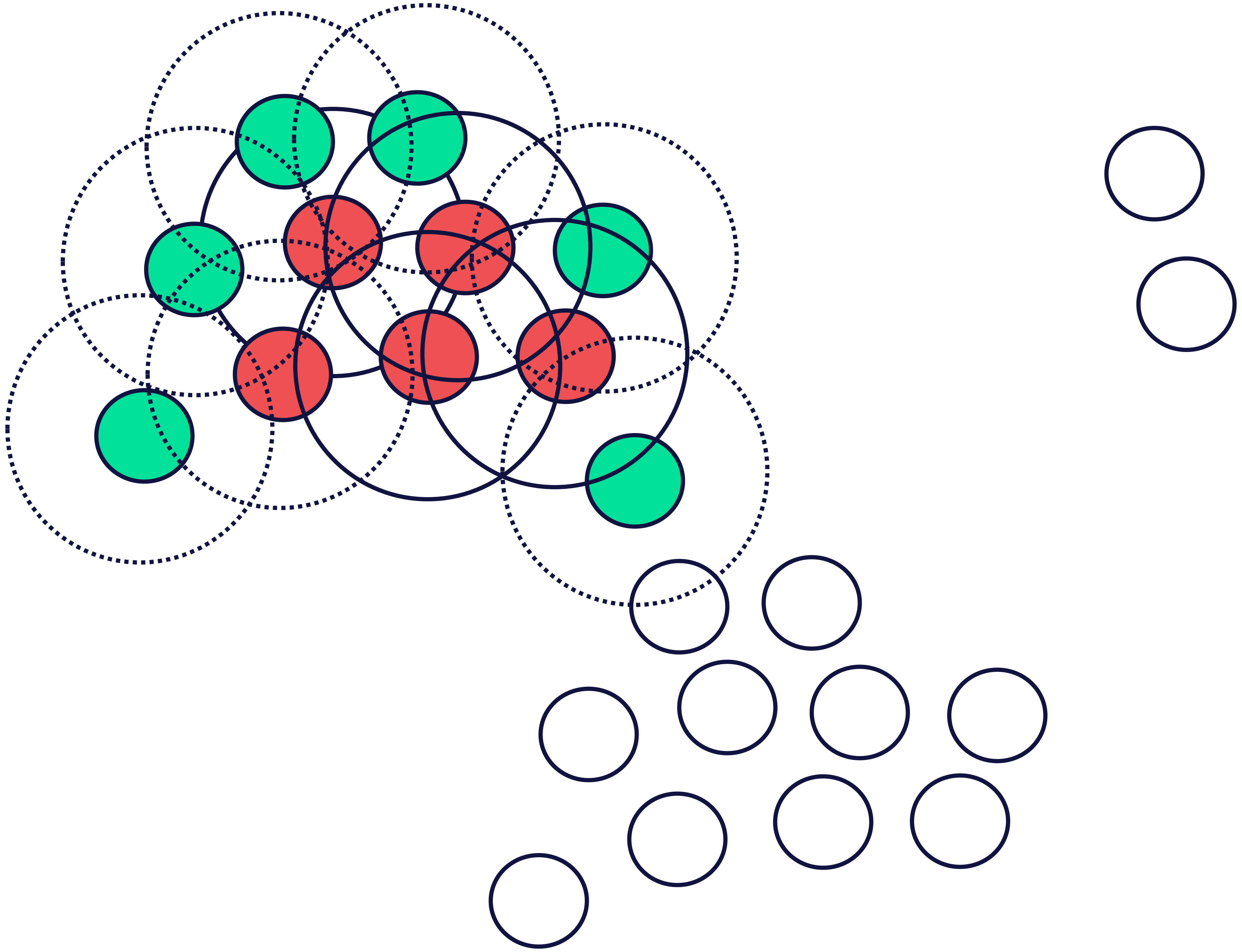
Step 1: Choose an ε for the radius of a circle and choose a minimum number of points to consider a cluster, say = 4.

Step 2: Randomly pick a point, draw a circle with radius ε, then check the number of intersected points.

Step 3: If the conditions are satisfied (min.point <= intersected points), then it’s a core point.

Step 4: Repeat steps 2 and 3 to all points.

Step 5: Points that do not satisfy the min. point requirements but are near core points are called border points.



Step 1: Choose an ε for the radius of a circle and choose a minimum number of points to consider a cluster, say = 4.

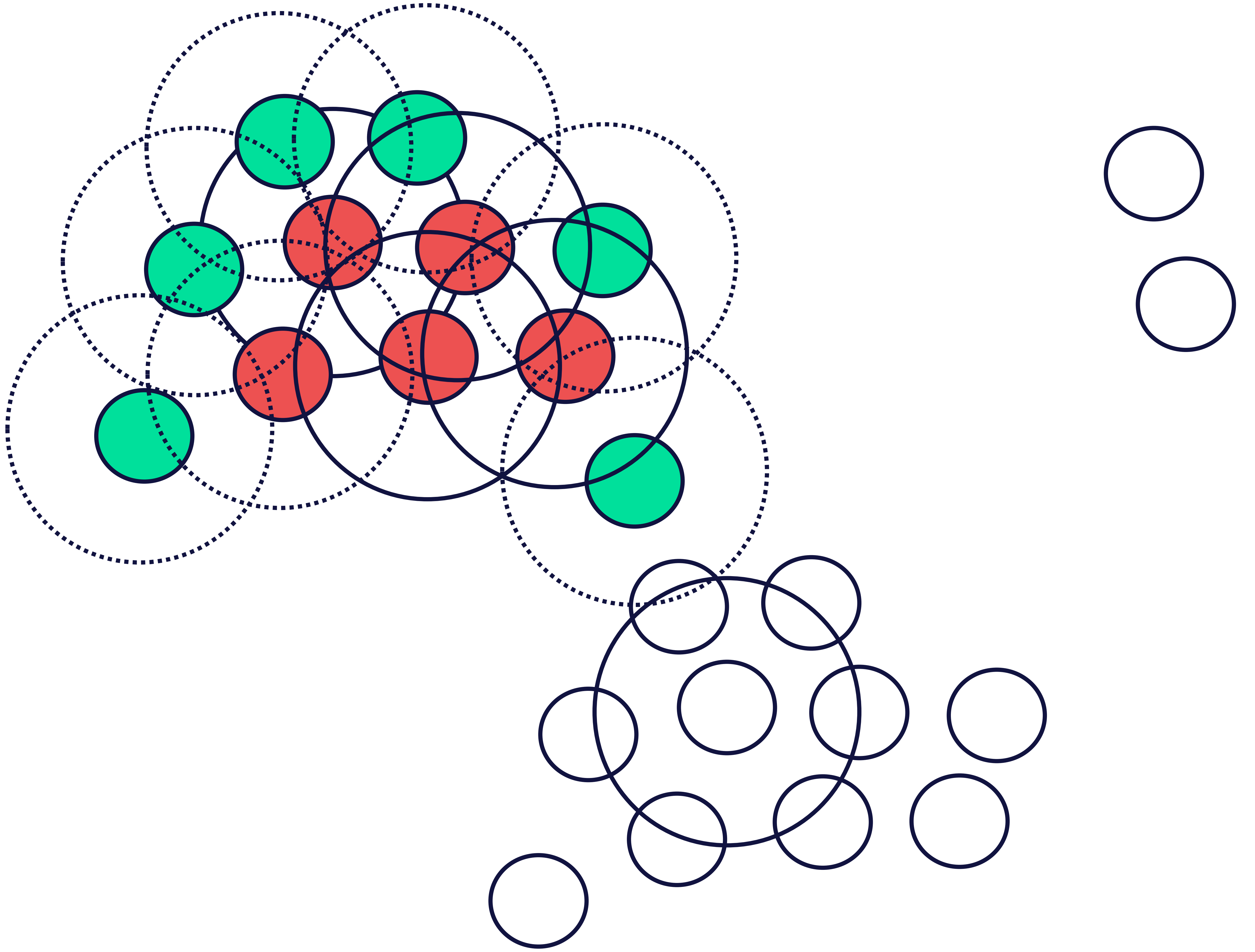
Step 2: Randomly pick a point, draw a circle with radius ε, then check the number of intersected points.

Step 3: If the conditions are satisfied (min.point <= intersected points), then it's a core point.

Step 4: Repeat steps 2 and 3 to all points.

Step 5: Points that do not satisfy the min. point requirements but are near core points are called border points.

Step 6: Perform steps 2 to 3 to all points.



Step 1: Choose an ε for the radius of a circle and choose a minimum number of points to consider a cluster, say = 4.

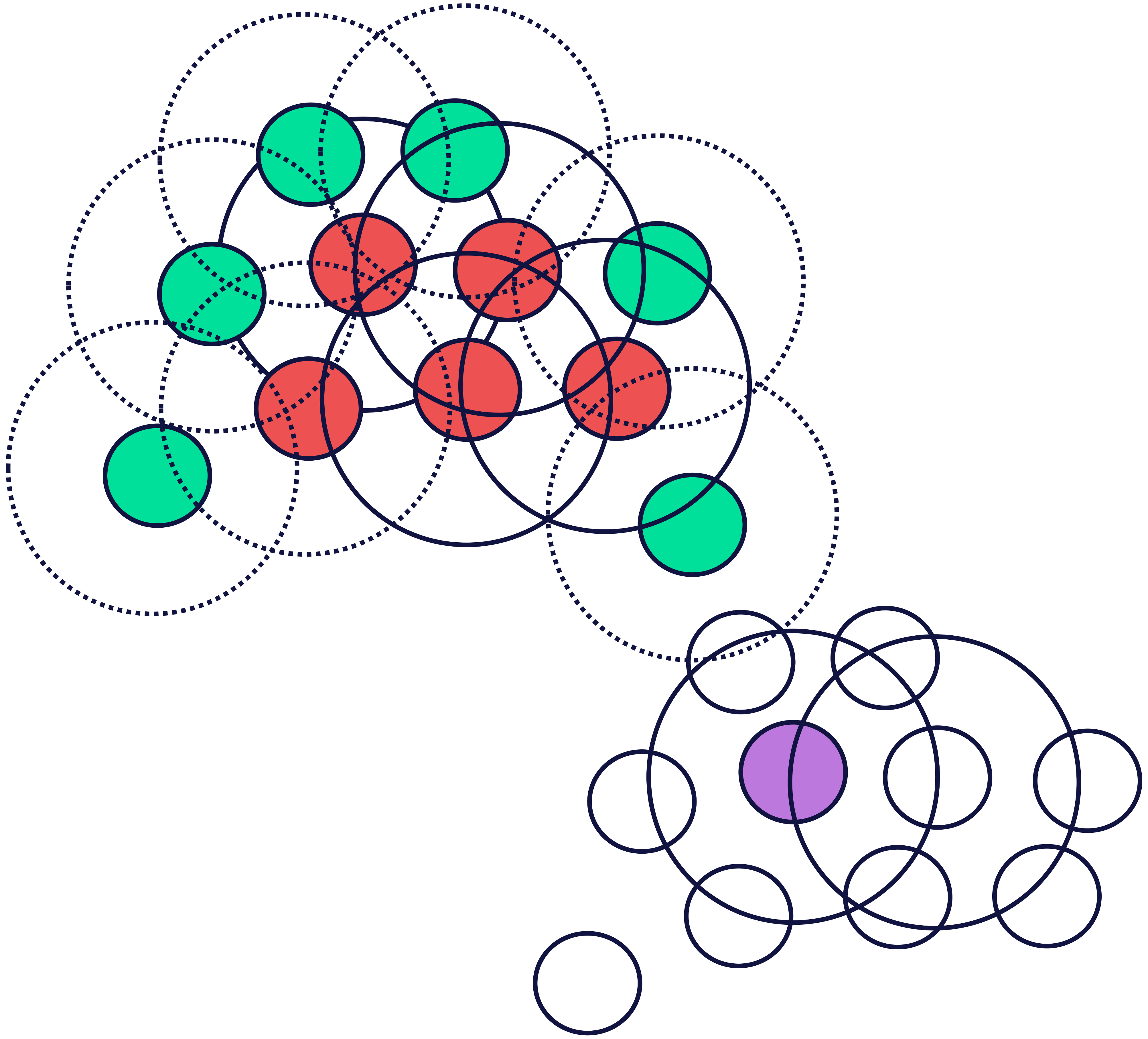
Step 2: Randomly pick a point, draw a circle with radius ε, then check the number of intersected points.

Step 3: If the conditions are satisfied (min.point <= intersected points), then it's a core point.

Step 4: Repeat steps 2 and 3 to all points.

Step 5: Points that do not satisfy the min. point requirements but are near core points are called border points.

Step 6: Perform steps 2 to 3 to all points.



Step 1: Choose an ε for the radius of a circle and choose a minimum number of points to consider a cluster, say = 4.

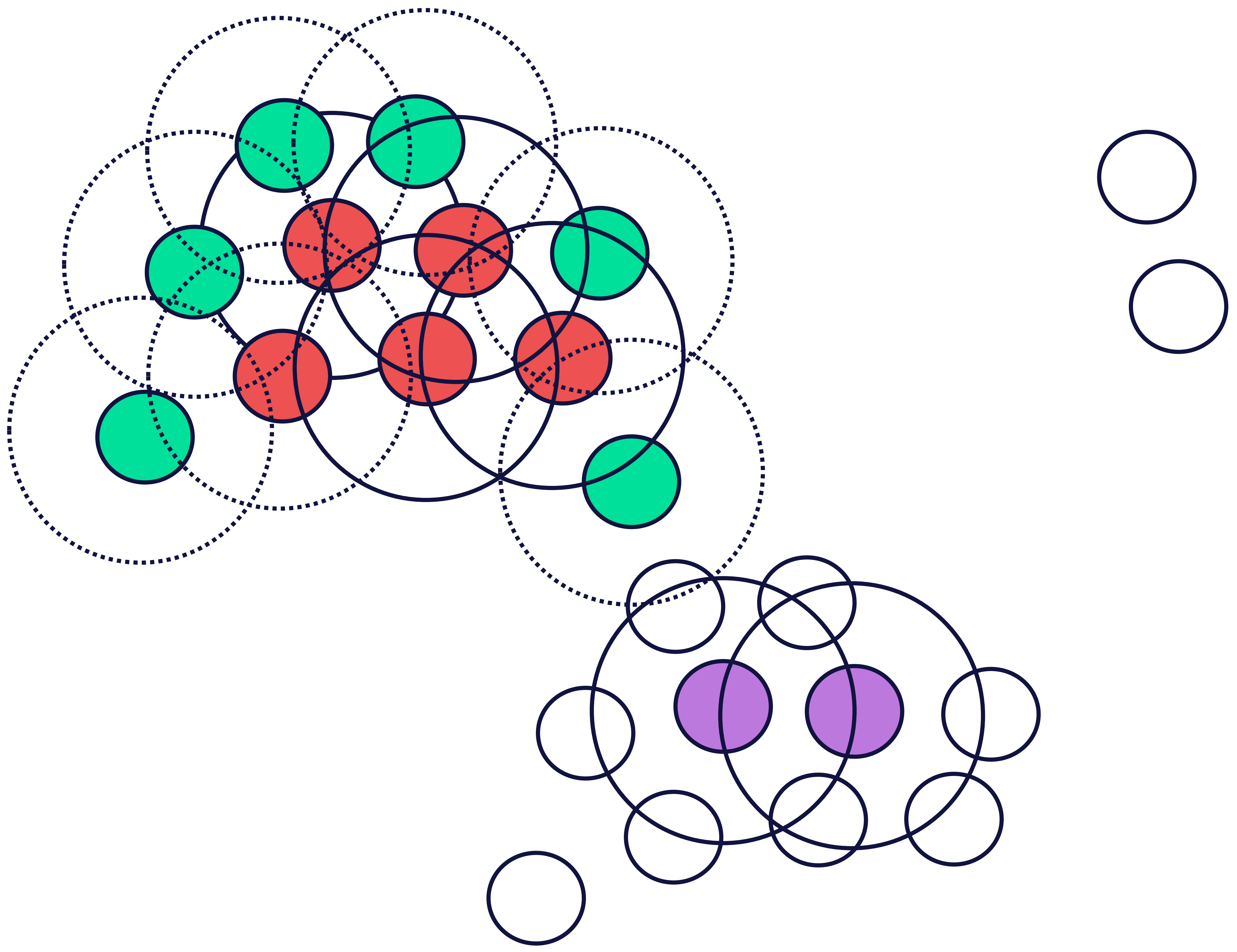
Step 2: Randomly pick a point, draw a circle with radius ε, then check the number of intersected points.

Step 3: If the conditions are satisfied (min.point <= intersected points), then it's a core point.

Step 4: Repeat steps 2 and 3 to all points.

Step 5: Points that do not satisfy the min. point requirements but are near core points are called border points.

Step 6: Perform steps 2 to 3 to all points.



Step 1: Choose an ε for the radius of a circle and choose a minimum number of points to consider a cluster, say = 4.

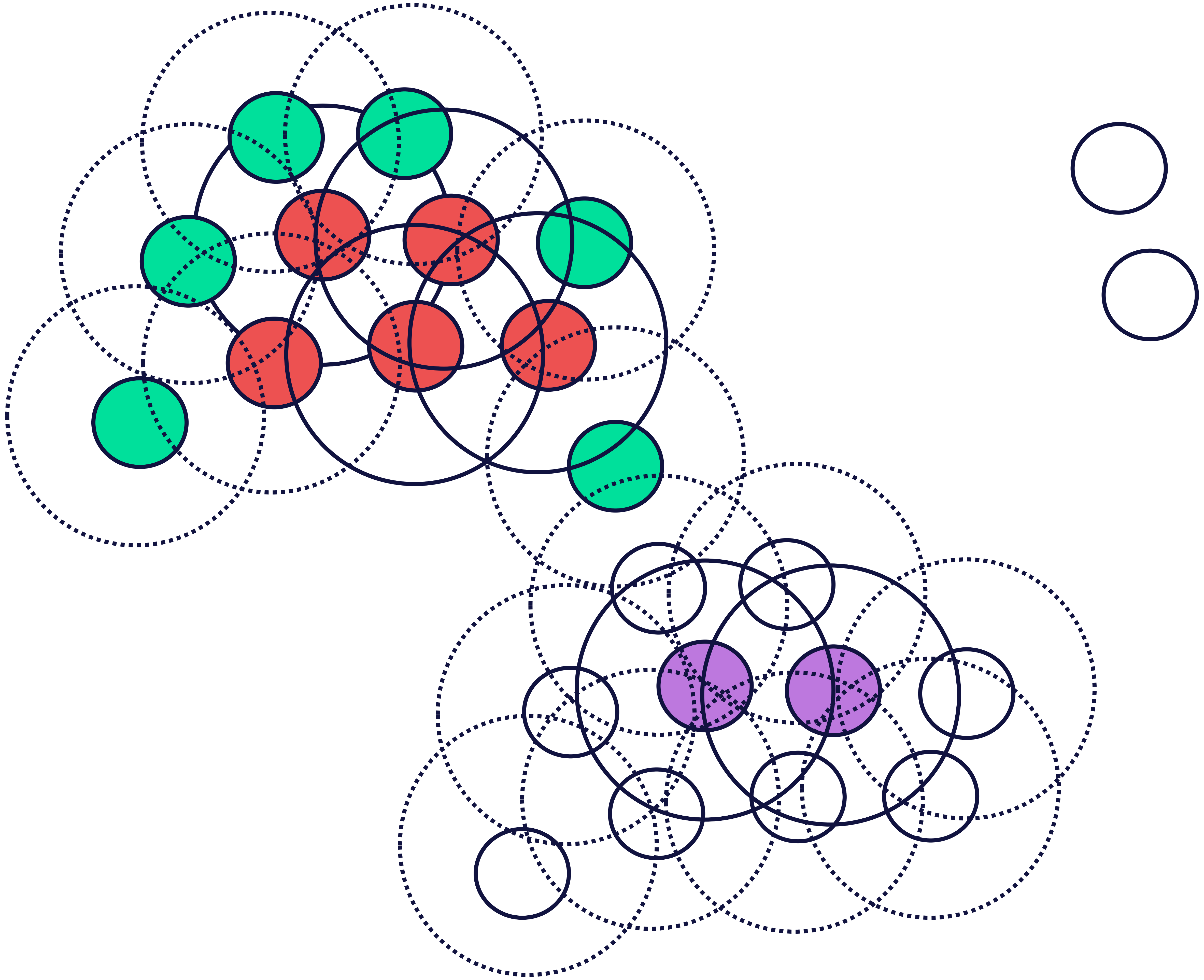
Step 2: Randomly pick a point, draw a circle with radius ε, then check the number of intersected points.

Step 3: If the conditions are satisfied (min.point <= intersected points), then it’s a core point.

Step 4: Repeat steps 2 and 3 to all points.

Step 5: Points that do not satisfy the min. point requirements but are near core points are called border points.

Step 6: Perform steps 2 to 3 to all points.



Step 1: Choose an ε for the radius of a circle and choose a minimum number of points to consider a cluster, say = 4.

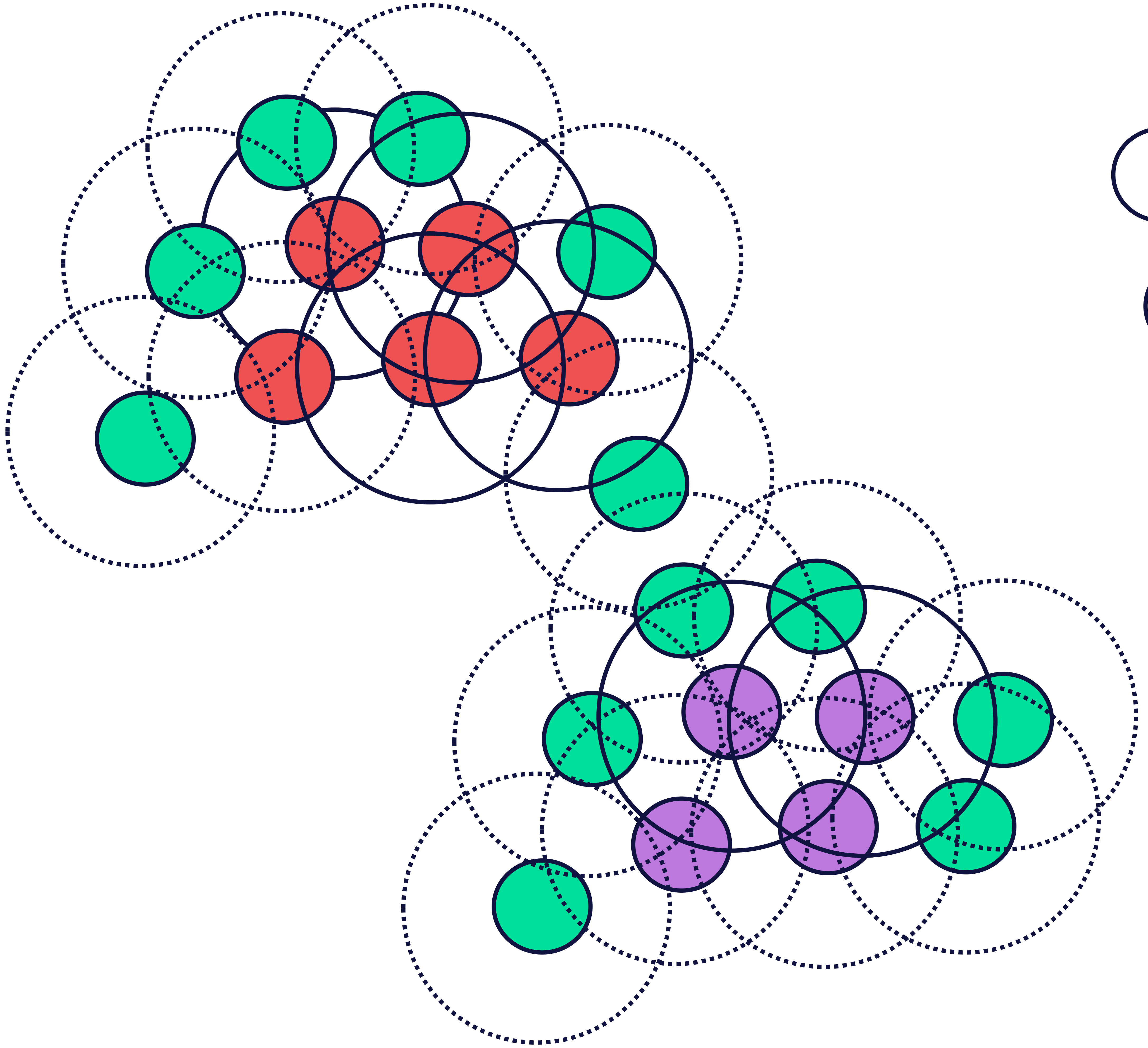
Step 2: Randomly pick a point, draw a circle with radius ε, then check the number of intersected points.

Step 3: If the conditions are satisfied (min.point <= intersected points), then it’s a core point.

Step 4: Repeat steps 2 and 3 to all points.

Step 5: Points that do not satisfy the min. point requirements but are near core points are called border points.

Step 6: Perform steps 2 to 3 to all points.



Step 1: Choose an ε for the radius of a circle and choose a minimum number of points to consider a cluster, say = 4.

Step 2: Randomly pick a point, draw a circle with radius ε, then check the number of intersected points.

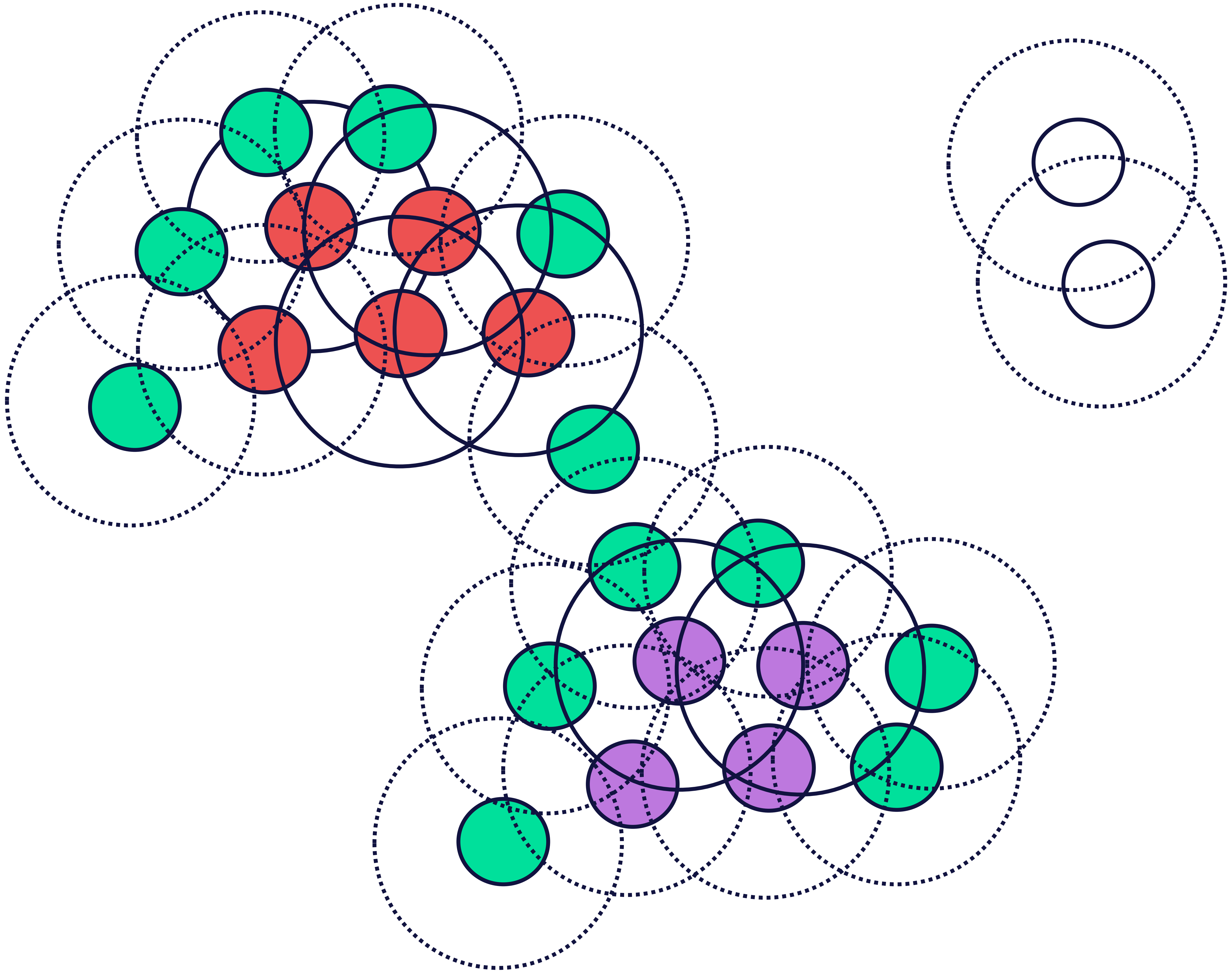
Step 3: If the conditions are satisfied (min.point <= intersected points), then it's a core point.

Step 4: Repeat steps 2 and 3 to all points.

Step 5: Points that do not satisfy the min. point requirements but are near core points are called border points.

Step 6: Perform steps 2 to 3 to all points.

Step 7: Points that are near border points or away from other points are considered noise.



Step 1: Choose an ε for the radius of a circle and choose a minimum number of points to consider a cluster, say = 4.

Step 2: Randomly pick a point, draw a circle with radius ε, then check the number of intersected points.

Step 3: If the conditions are satisfied (min.point <= intersected points), then it's a core point.

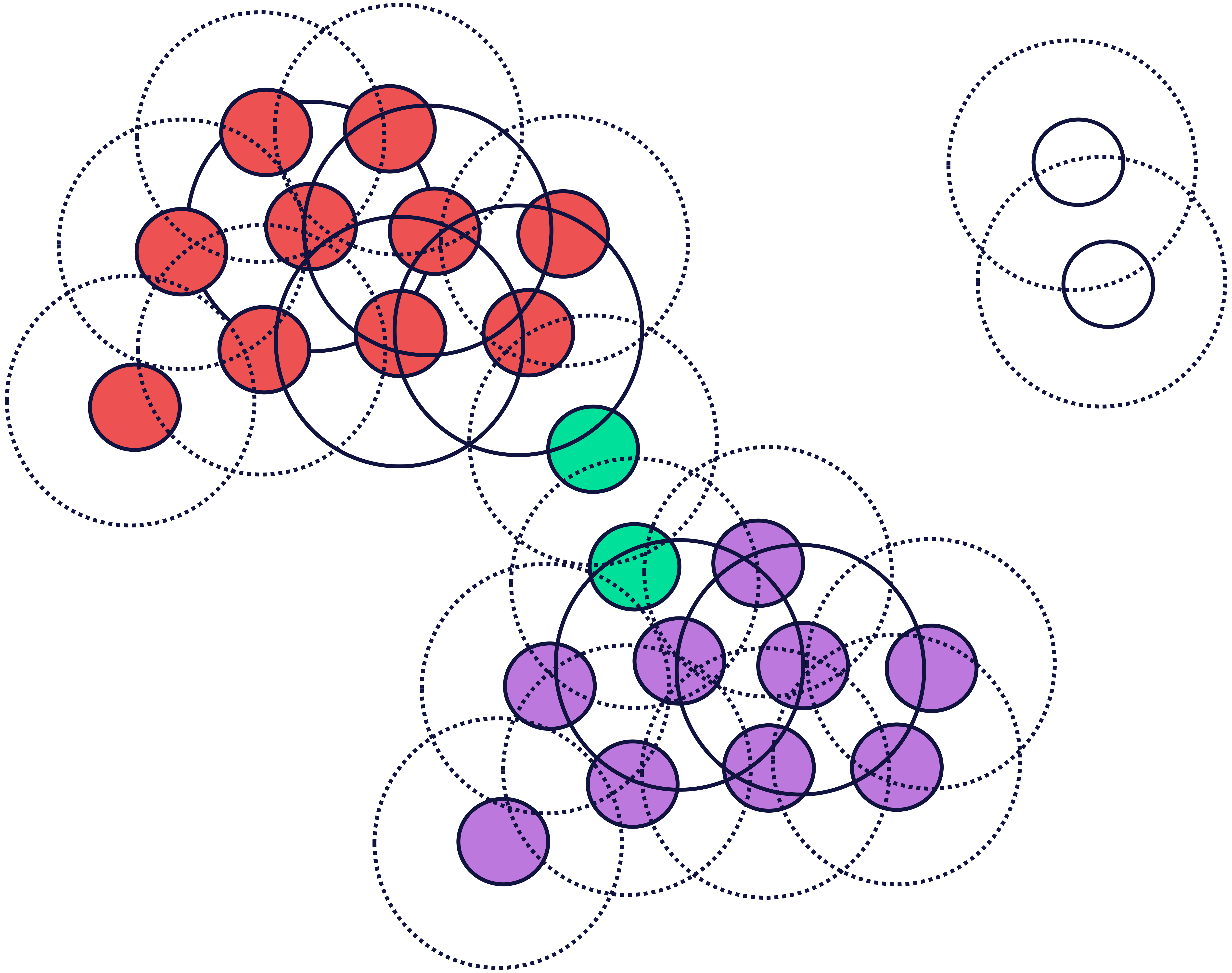
Step 4: Repeat steps 2 and 3 to all points.

Step 5: Points that do not satisfy the min. point requirements but are near core points are called border points.

Step 6: Perform steps 2 to 3 to all points.

Step 7: Points that are near border points or away from other points are considered noise.

Step 8: Perform cluster assignment based on core points.



Step 1: Choose an ε for the radius of a circle and choose a minimum number of points to consider a cluster, say = 4.

Step 2: Randomly pick a point, draw a circle with radius ε, then check the number of intersected points.

Step 3: If the conditions are satisfied (min.point <= intersected points), then it’s a core point.

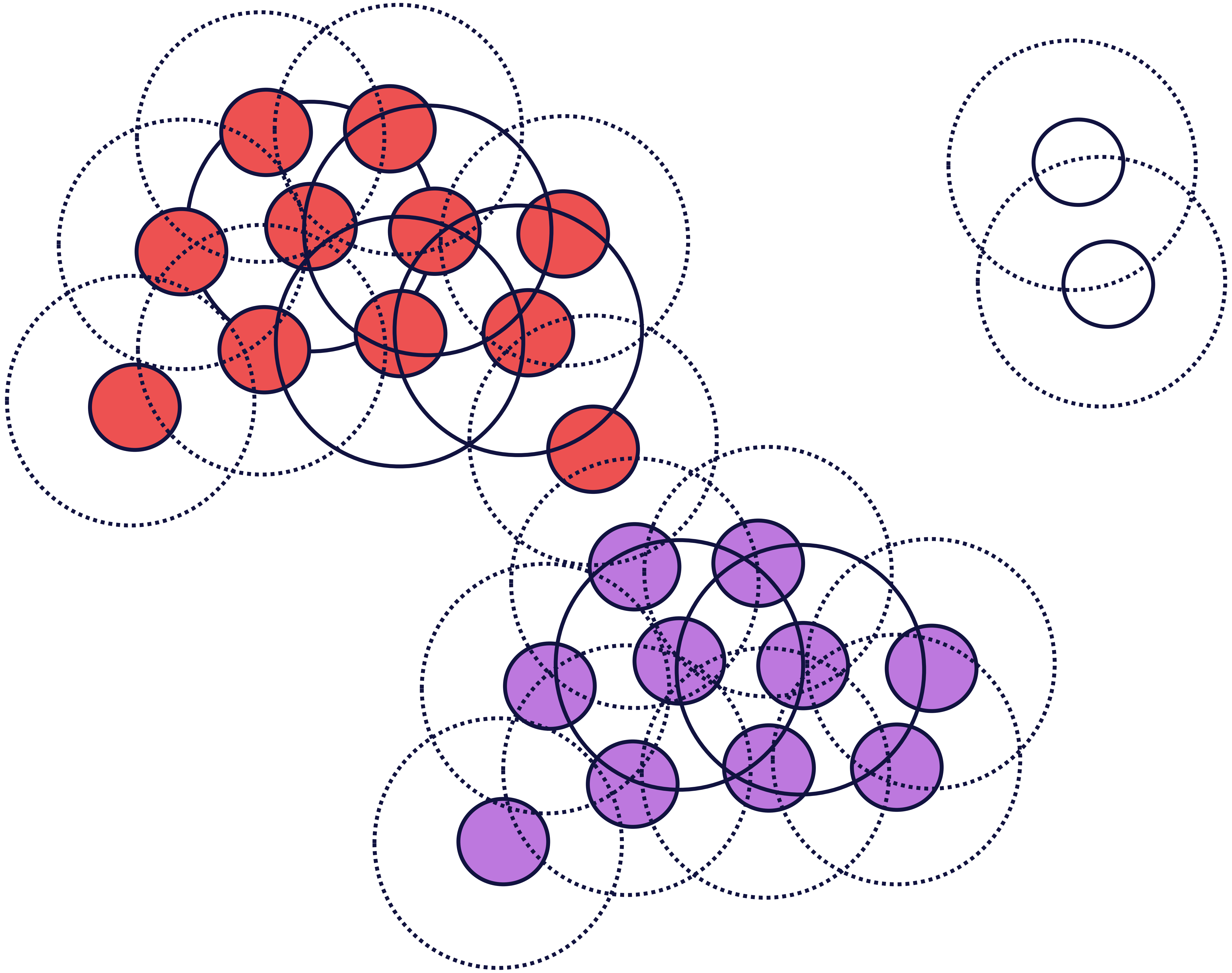
Step 4: Repeat steps 2 and 3 to all points.

Step 5: Points that do not satisfy the min. point requirements but are near core points are called border points.

Step 6: Perform steps 2 to 3 to all points.

Step 7: Points that are near border points or away from other points are considered noise.

Step 8: Perform cluster assignment based on core points.



Step 1: Choose an ε for the radius of a circle and choose a minimum number of points to consider a cluster, say = 4.

Step 2: Randomly pick a point, draw a circle with radius ε, then check the number of intersected points.

Step 3: If the conditions are satisfied (min.point <= intersected points), then it's a core point.

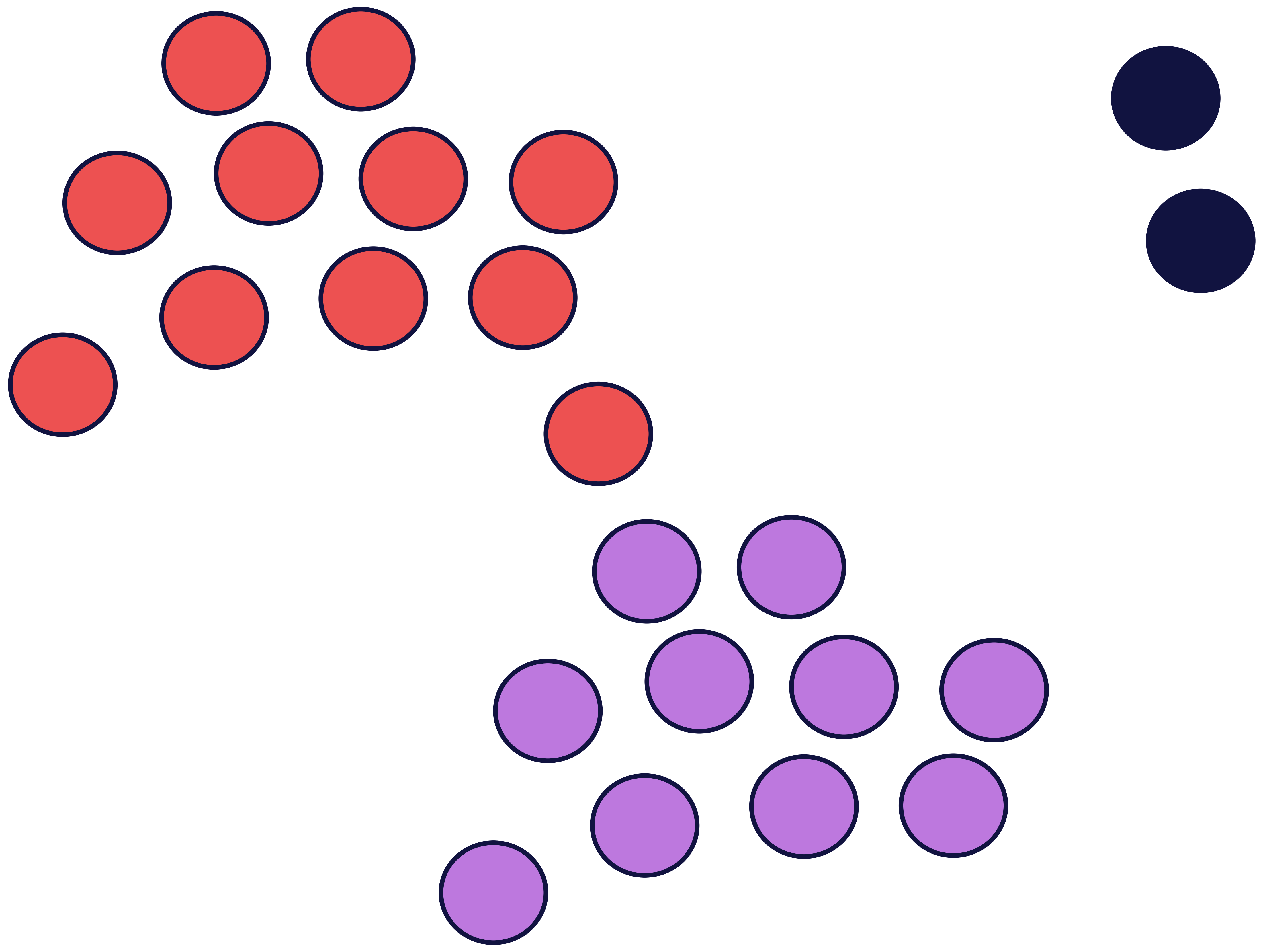
Step 4: Repeat steps 2 and 3 to all points.

Step 5: Points that do not satisfy the min. point requirements but are near core points are called border points.

Step 6: Perform steps 2 to 3 to all points.

Step 7: Points that are near border points or away from other points are considered noise.

Step 8: Perform cluster assignment based on core points.



Technical Questions

Improving the Performance of Clustering

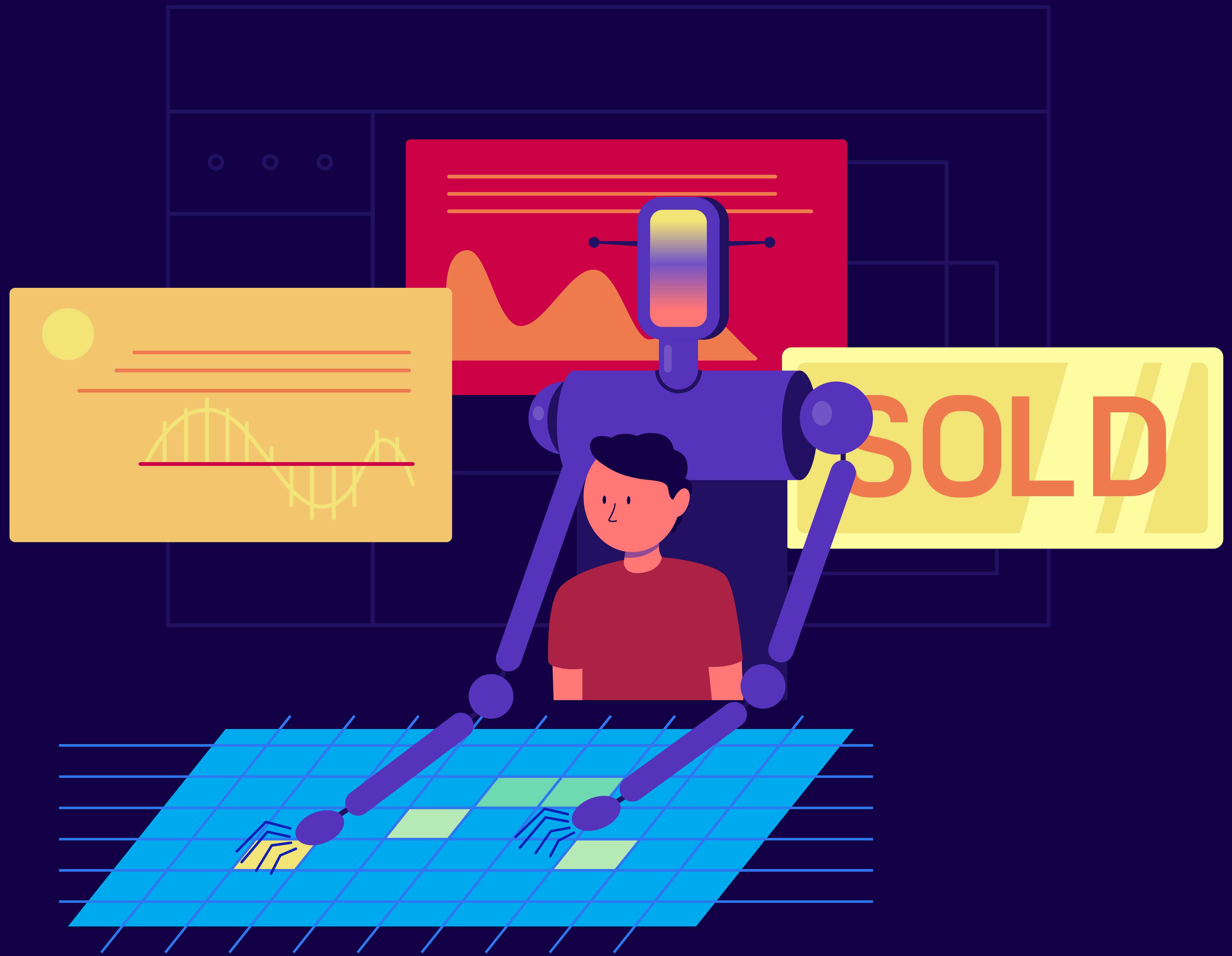
How Clusters?

1. Knee or Elbow Method
2. Subject Matter Expertise.

Evaluation Metrics

1. Internal Evaluation
These metrics evaluate the quality of a clustering solution without reference to external data (no ground truth are available). They generally assess how compact the clusters are (cohesion) and how separate or distinct the clusters are from one another (separation).

2. External Evaluation
These metrics compare the clustering results to an external standard, often a ground truth label set. They are useful when the true labels are known, providing a way to measure how closely the clustering matches the actual distribution



Active Areas of Research

What's the Gap?

Complex Data Structures

1. Purely categorical data.
2. Mixed Data.
3. High Dimensional Data.
4. Multi-modal Data

Ensembled Methods

Increase robustness and reliability

New Validation Methods

1. Categorical Clustering.
2. Non-Distance based methods.

Deep Learning Methods

Development of novel architectures for clustering