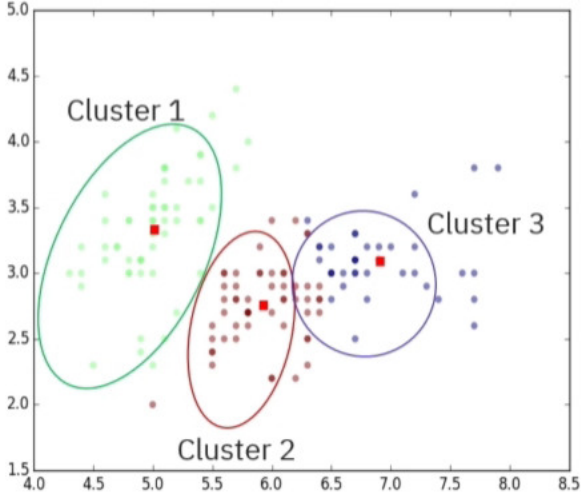
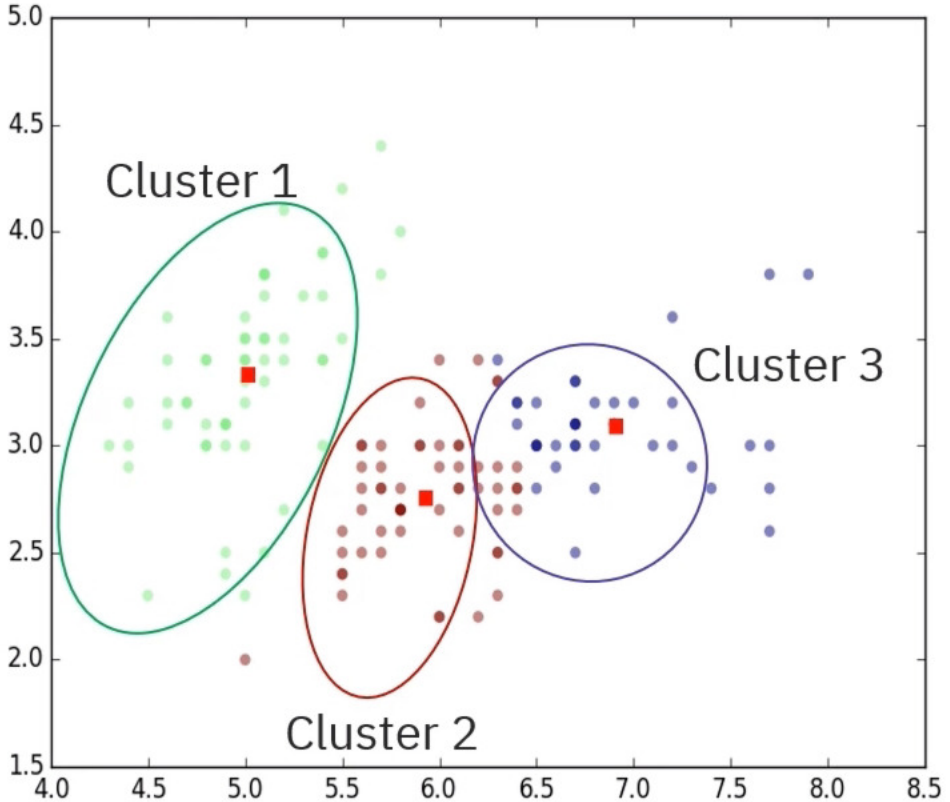
What is clustering?



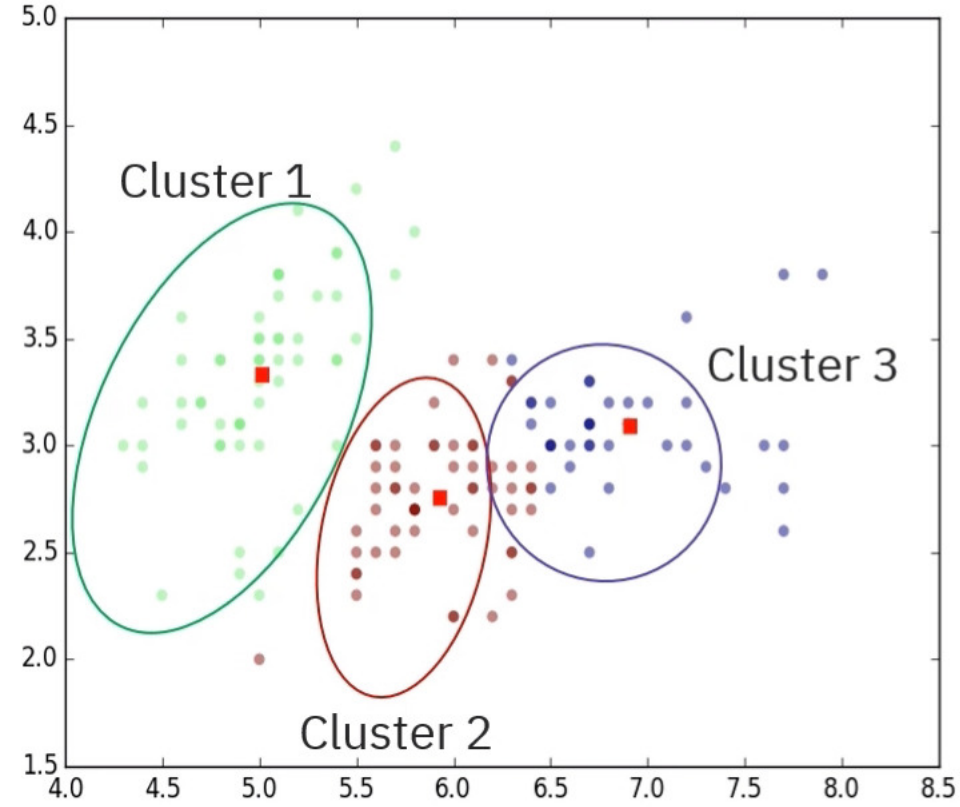
Clustering

- Machine learning technique
- Automatically groups data points based on similarities



Applications

- Identifying music genres
- Segmenting user groups
- Analyzing market segments



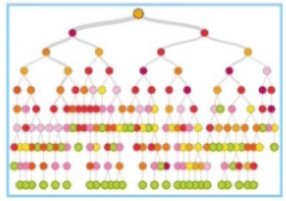
Can use one or multiple data features

Clustering and classification

Labeled dataset

Customer Id	Age	Edu	Years Employed	Income	Card Debt	Other Debt	Address	DebtIncomeRatio	Defaulted
1	41	2	6	19	0.124	1.073	NBA001	6.3	0
2	47	1	26	100	4.582	8.218	NBA021	12.8	0
3	33	2	10	57	6.111	5.802	NBA013	20.9	1
4	29	2	4	19	0.681	0.516	NBA009	6.3	0
5	47	1	31	253	9.308	8.908	NBA008	7.2	0
6	40	1	23	81	0.998	7.831	NBA016	10.9	1
7	38	2	4	56	0.442	0.454	NBA013	1.6	0
8	42	3	0	64	0.279	3.945	NBA009	6.6	0
9	26	1	5	18	0.575	2.215	NBA006	15.5	1
10	47	3	23	115	0.653	3.947	NBA011	4	0

Modeling



Decision Tree

Prediction

Data set includes customer features and loan status

Classification algorithm predicts categorical labels

Decision tree model predicts loan default status

Clustering and classification

Labeled dataset

Customer Id	Age	Edu	Years Employed	Income	Card Debt	Other Debt	Address	DebtIncomeRatio	Defaulted
1	41	2	6	19	0.124	1.073	NBA001	6.3	0
2	47	1	26	100	4.582	8.218	NBA021	12.8	0
3	33	2	10	57	6.111	5.802	NBA013	20.9	1
4	29	2	4	19	0.681	0.516	NBA009	6.3	0
5	47	1	31	253	9.308	8.908	NBA008	7.2	0
6	40	1	23	81	0.998	7.831	NBA016	10.9	1
7	38	2	4	56	0.442	0.454	NBA013	1.6	0
8	42	3	0	64	0.279	3.945	NBA009	6.6	0
9	26	1	5	18	0.575	2.215	NBA006	15.5	1

Unlabeled dataset

Customer Id	Age	Edu	Years Employed	Income	Card Debt	Other Debt	Address	DebtIncomeRatio
1	41	2	6	19	0.124	1.073	NBA001	6.3
2	47	1	26	100	4.582	8.218	NBA021	12.8
3	33	2	10	57	6.111	5.802	NBA013	20.9
4	29	2	4	19	0.681	0.516	NBA009	6.3
5	47	1	31	253	9.308	8.908	NBA008	7.2
6	40	1	23	81	0.998	7.831	NBA016	10.9
7	38	2	4	56	0.442	0.454	NBA013	1.6
8	42	3	0	64	0.279	3.945	NBA009	6.6
9	26	1	5	18	0.575	2.215	NBA006	15.5

- Resembles classification but uses unlabeled data
- Finds patterns to form clusters

Labeled dataset

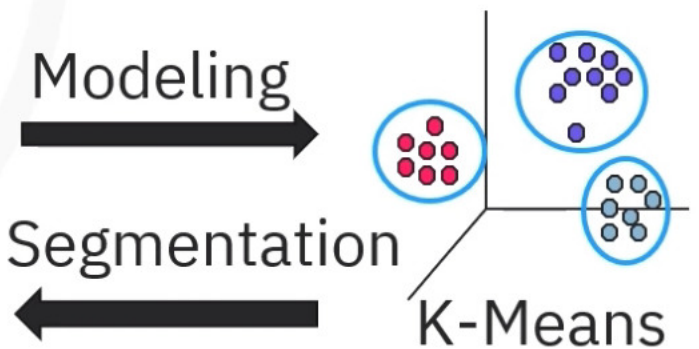
Customer Id	Age	Edu	Years Employed	Income	Card Debt	Other Debt	Address	DebtIncomeRatio	Defaulted
1	41	2	6	19	0.124	1.073	NBA001	6.3	0
2	47	1	26	100	4.582	8.218	NBA021	12.8	0
3	33	2	10	57	6.111	5.802	NBA013	20.9	1
4	29	2	4	19	0.681	0.516	NBA009	6.3	0
5	47	1	31	253	9.308	8.908	NBA008	7.2	0
6	40	1	23	81	0.998	7.831	NBA016	10.9	1
7	38	2	4	56	0.442	0.454	NBA013	1.6	0
8	42	3	0	64	0.279	3.945	NBA009	6.6	0
9	26	1	5	18	0.575	2.215	NBA006	15.5	1

Unlabeled dataset

Customer Id	Age	Edu	Years Employed	Income	Card Debt	Other Debt	Address	DebtIncomeRatio
1	41	2	6	19	0.124	1.073	NBA001	6.3
2	47	1	26	100	4.582	8.218	NBA021	12.8
3	33	2	10	57	6.111	5.802	NBA013	20.9
4	29	2	4	19	0.681	0.516	NBA009	6.3
5	47	1	31	253	9.308	8.908	NBA008	7.2
6	40	1	23	81	0.998	7.831	NBA016	10.9
7	38	2	4	56	0.442	0.454	NBA013	1.6
8	42	3	0	64	0.279	3.945	NBA009	6.6
9	26	1	5	18	0.575	2.215	NBA006	15.5

K-means model

- Segments customers by characteristics
- Identifies three distinct customer clusters



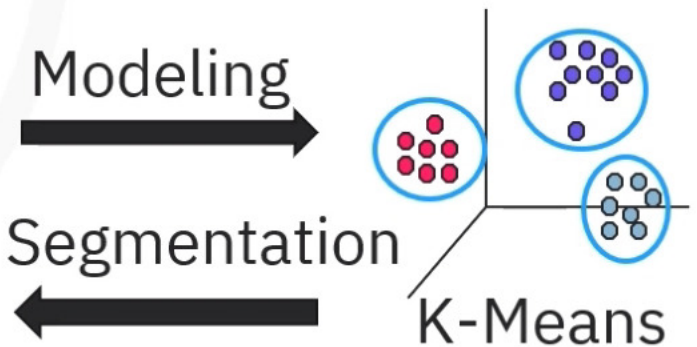
Labeled dataset

Customer Id	Age	Edu	Years Employed	Income	Card Debt	Other Debt	Address	DebtIncomeRatio	Defaulted
1	41	2	6	19	0.124	1.073	NBA001	6.3	0
2	47	1	26	100	4.582	8.218	NBA021	12.8	0
3	33	2	10	57	6.111	5.802	NBA013	20.9	1
4	29	2	4	19	0.681	0.516	NBA009	6.3	0
5	47	1	31	253	9.308	8.908	NBA008	7.2	0
6	40	1	23	81	0.998	7.831	NBA016	10.9	1
7	38	2	4	56	0.442	0.454	NBA013	1.6	0
8	42	3	0	64	0.279	3.945	NBA009	6.6	0
9	26	1	5	18	0.575	2.215	NBA006	15.5	1

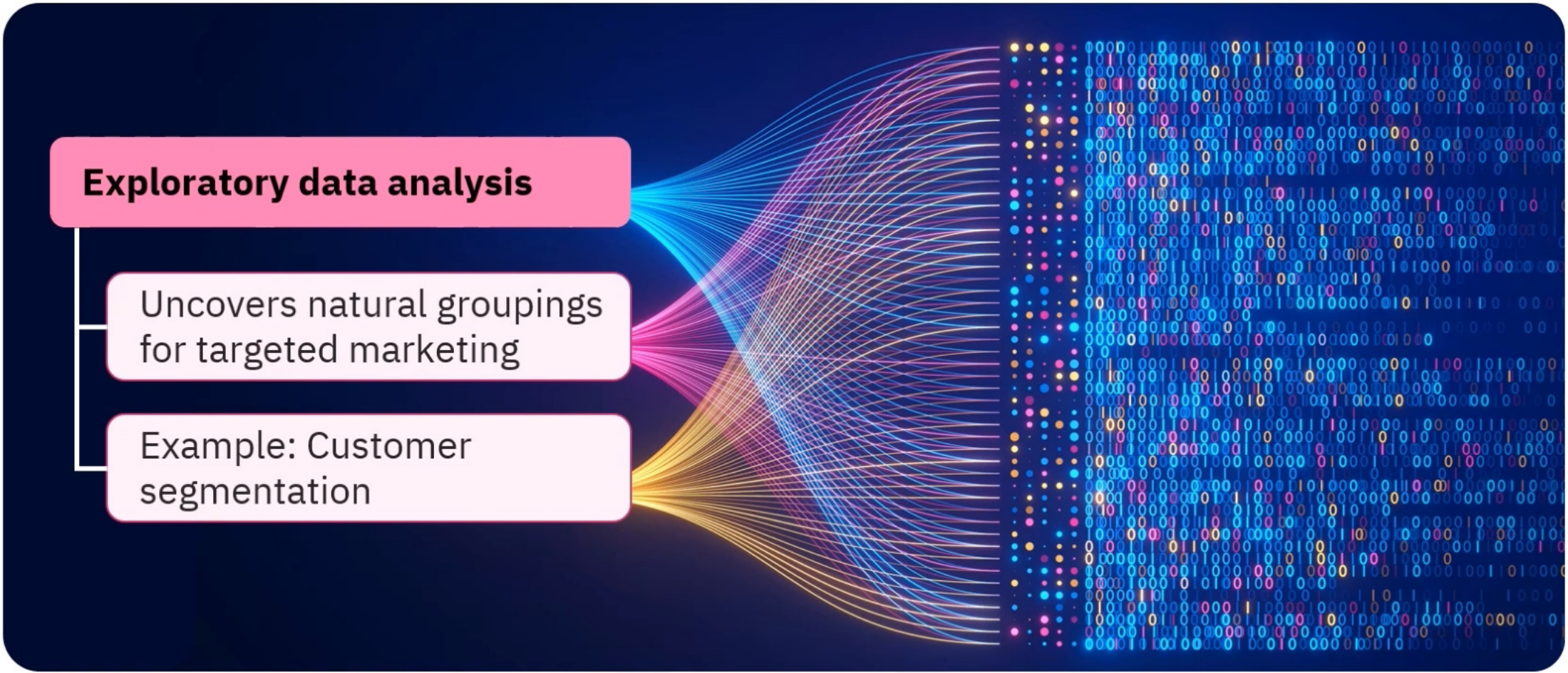
Unlabeled dataset

Customer Id	Age	Edu	Years Employed	Income	Card Debt	Other Debt	Address	DebtIncomeRatio
1	41	2	6	19	0.124	1.073	NBA001	6.3
2	47	1	26	100	4.582	8.218	NBA021	12.8
3	33	2	10	57	6.111	5.802	NBA013	20.9
4	29	2	4	19	0.681	0.516	NBA009	6.3
5	47	1	31	253	9.308	8.908	NBA008	7.2
6	40	1	23	81	0.998	7.831	NBA016	10.9
7	38	2	4	56	0.442	0.454	NBA013	1.6
8	42	3	0	64	0.279	3.945	NBA009	6.6
9	26	1	5	18	0.575	2.215	NBA006	15.5

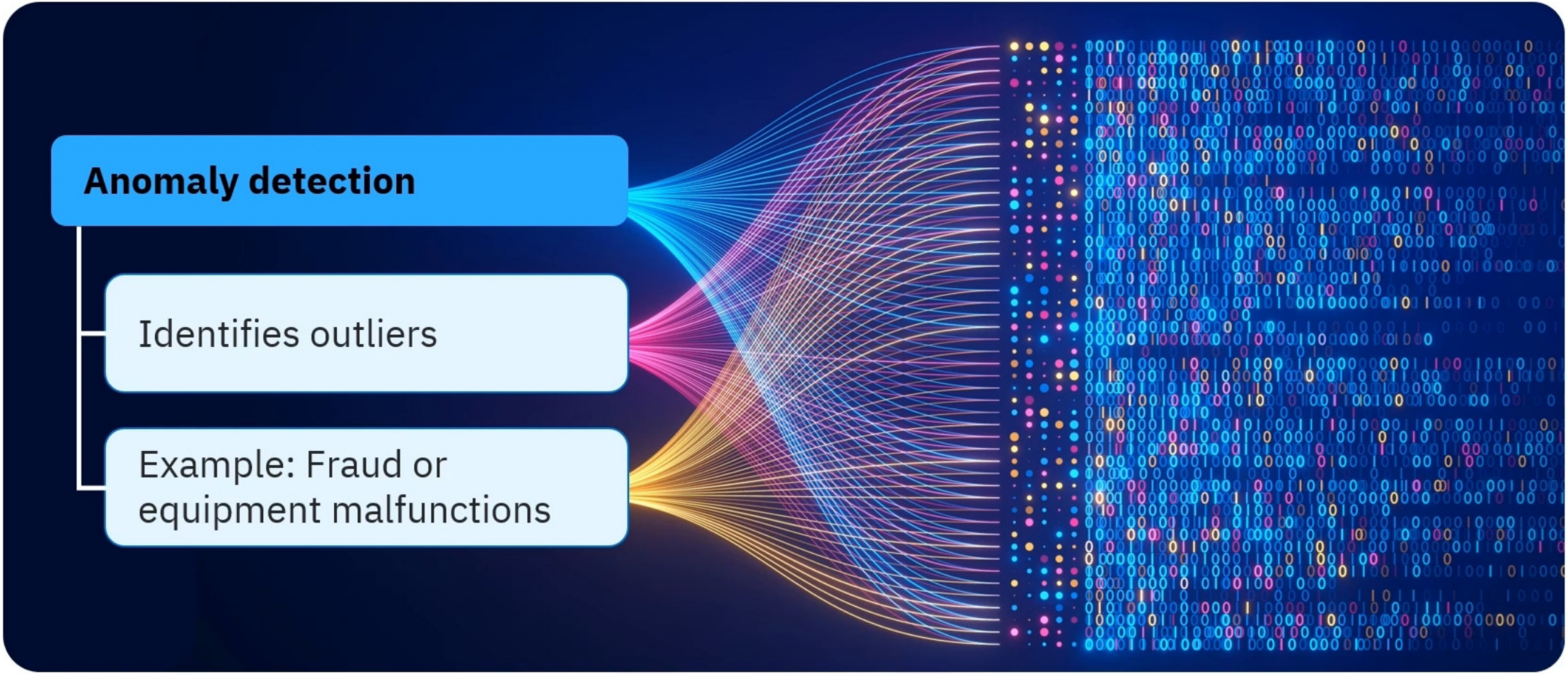
- The model operates without knowing defaults
- Default data is not part of the design

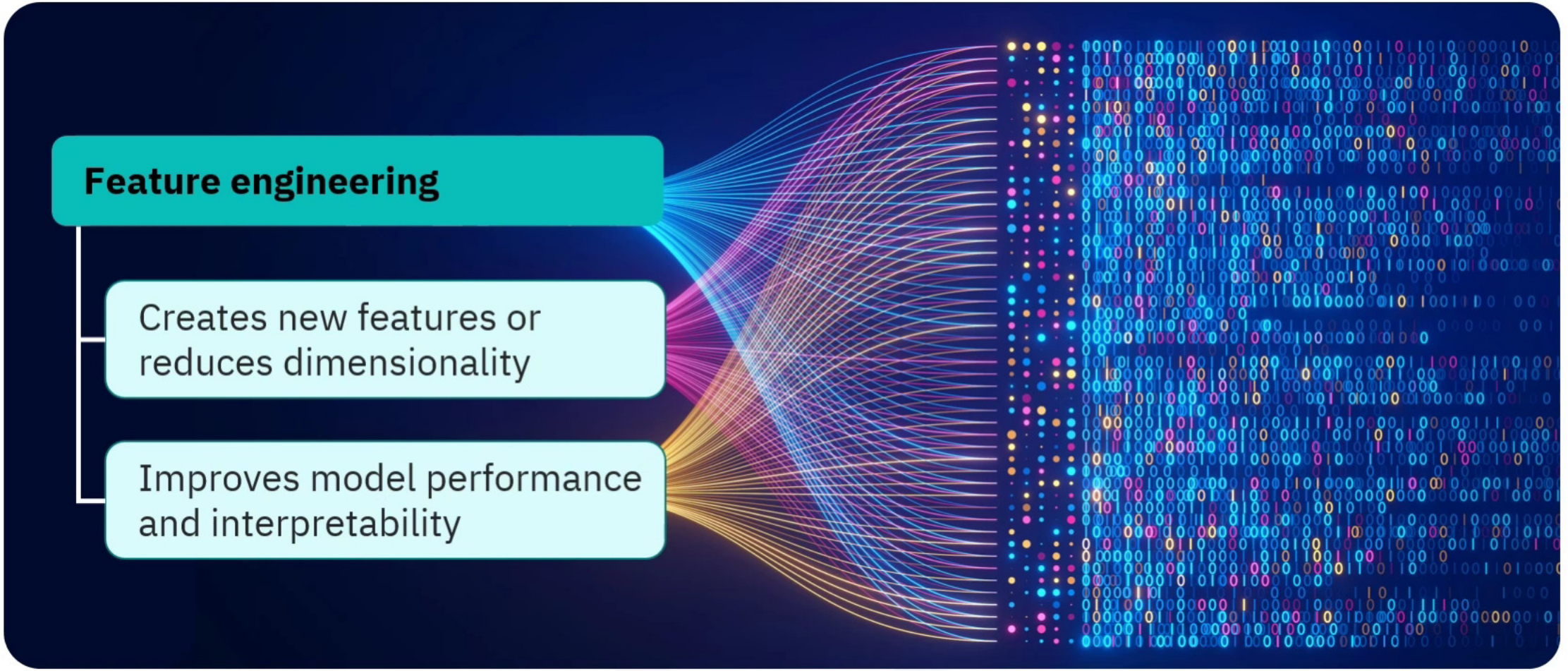


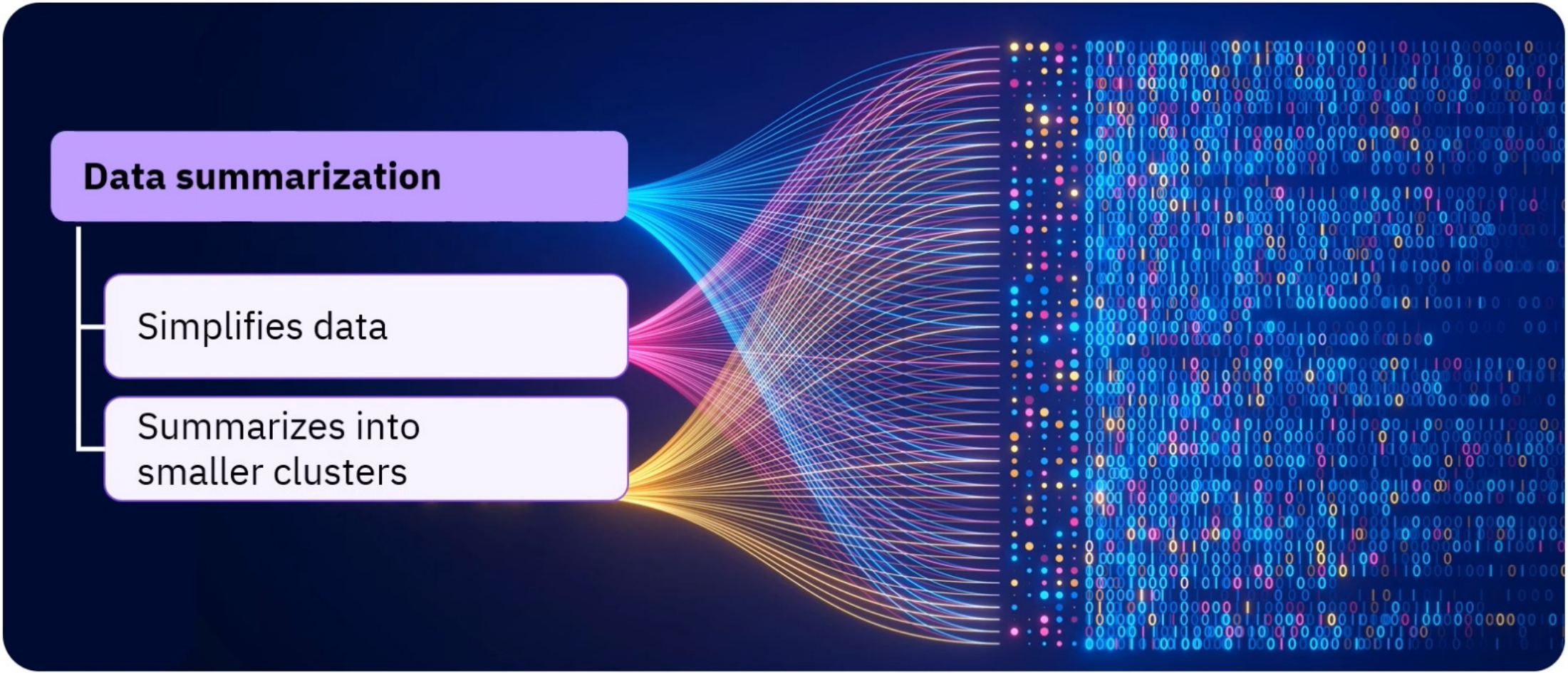
Common applications of clustering

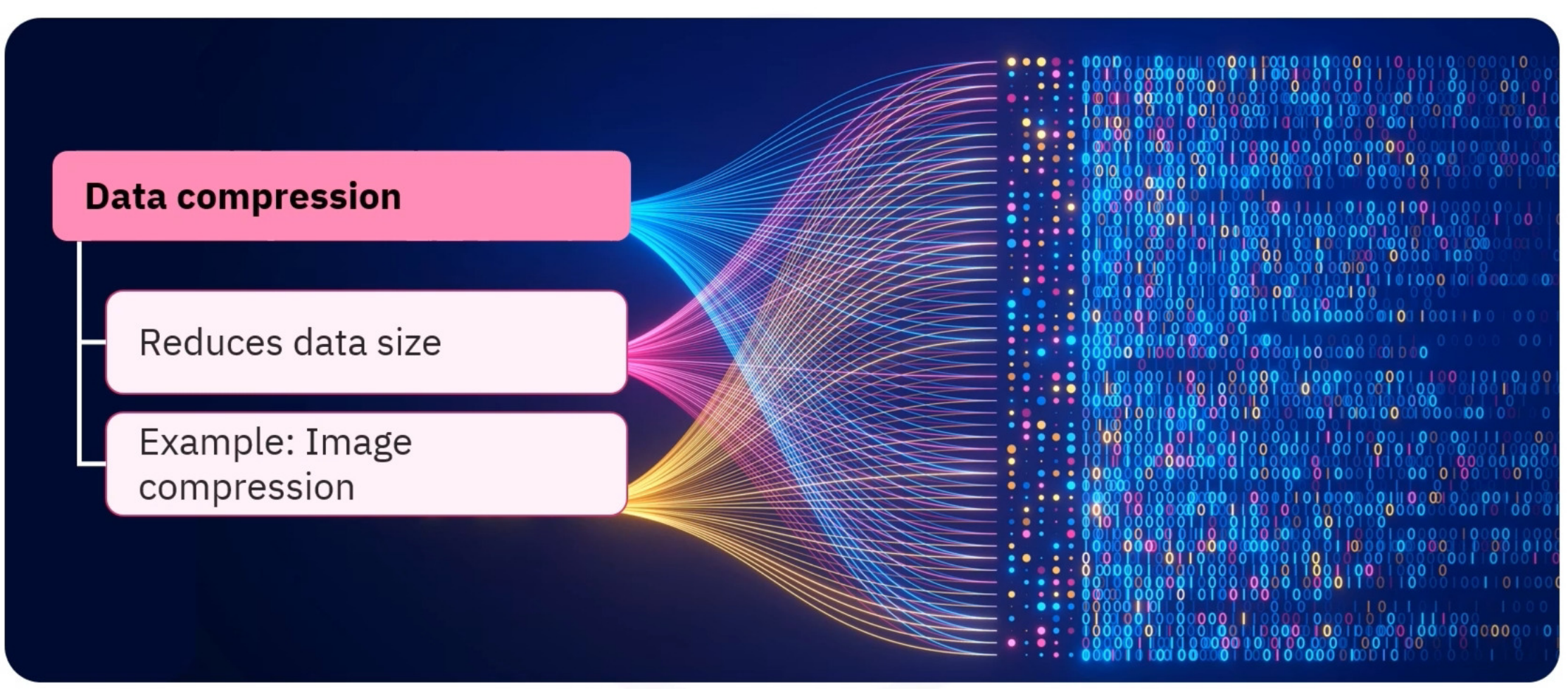


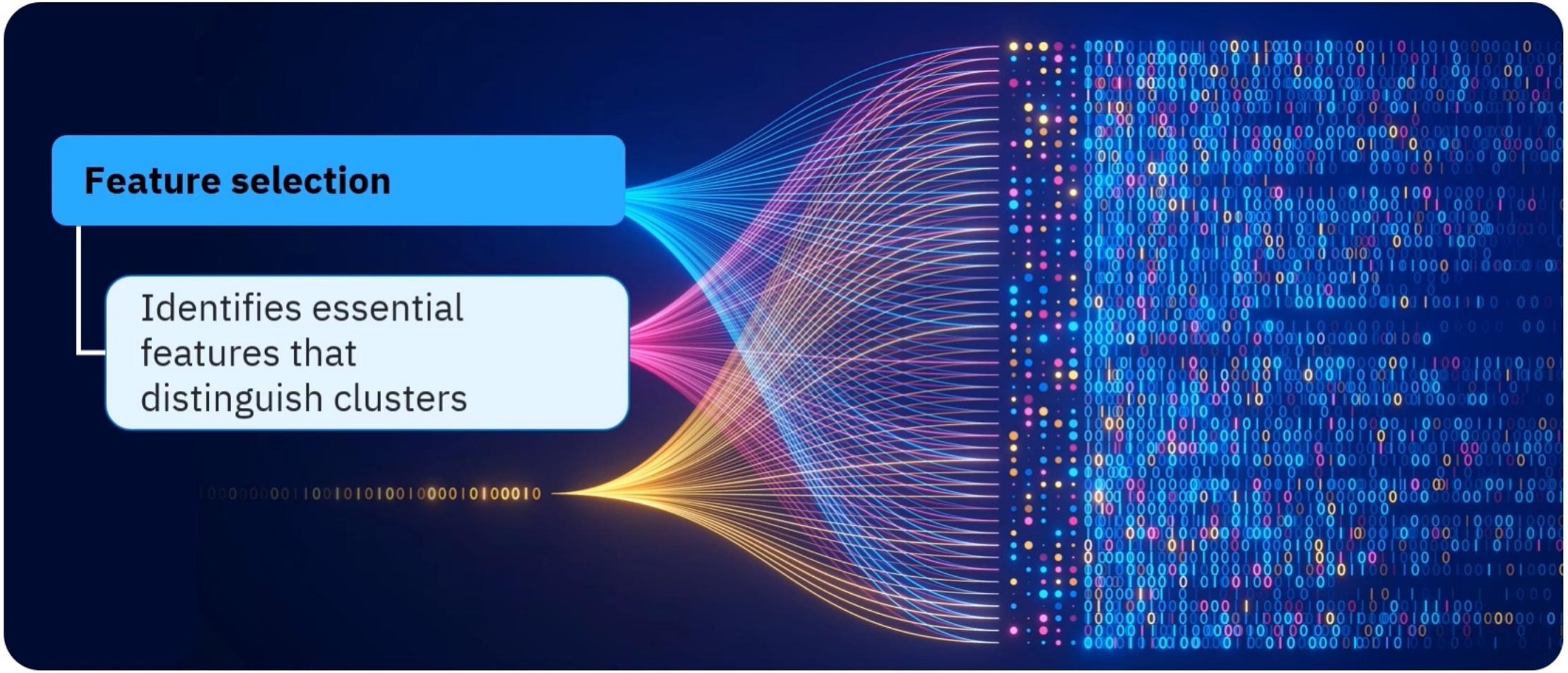

Pattern recognition
Groups objects and aids in image segmentation
Example: Detecting medical abnormalities


Anomaly detection
Identifies outliers
Example: Fraud or equipment malfunctions


Feature engineering
Creates new features or reduces dimensionality
Improves model performance and interpretability


Data summarization
Simplifies data
Summarizes into smaller clusters

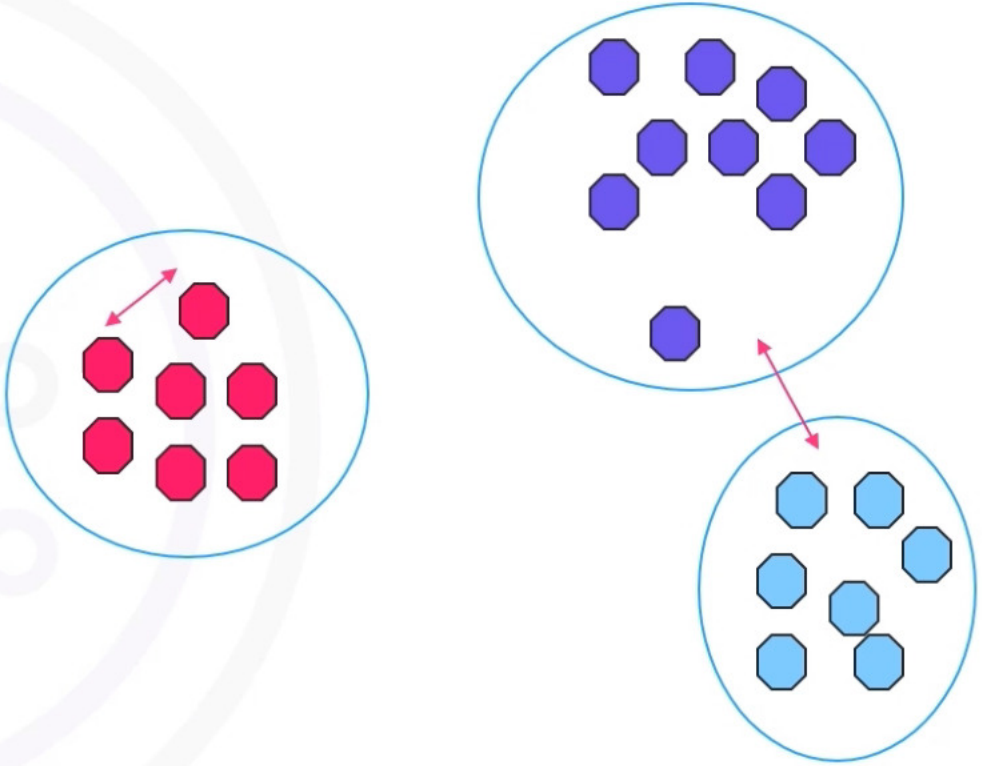

Data compression
Reduces data size
Example: Image compression

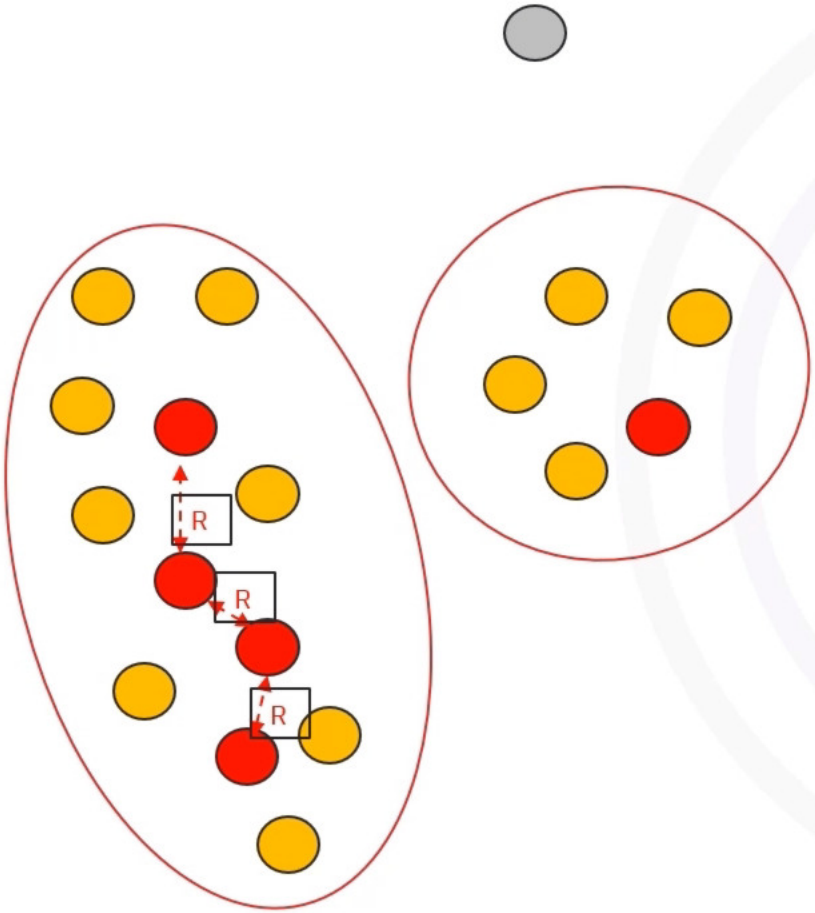

Feature selection
Identifies essential features that distinguish clusters

Types of clustering

Partition-based

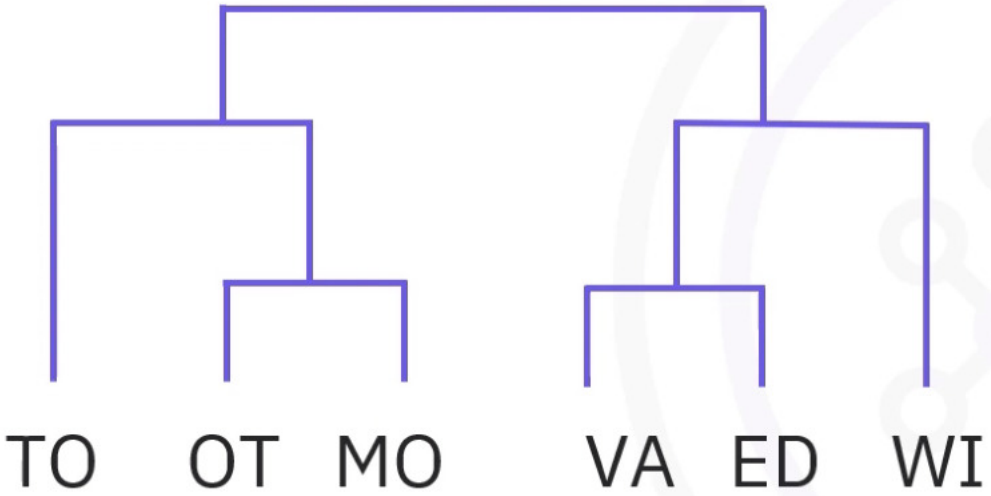
- Divides data into non-overlapping groups
- Identifies k clusters with minimal variance
- Scales well with large data sets





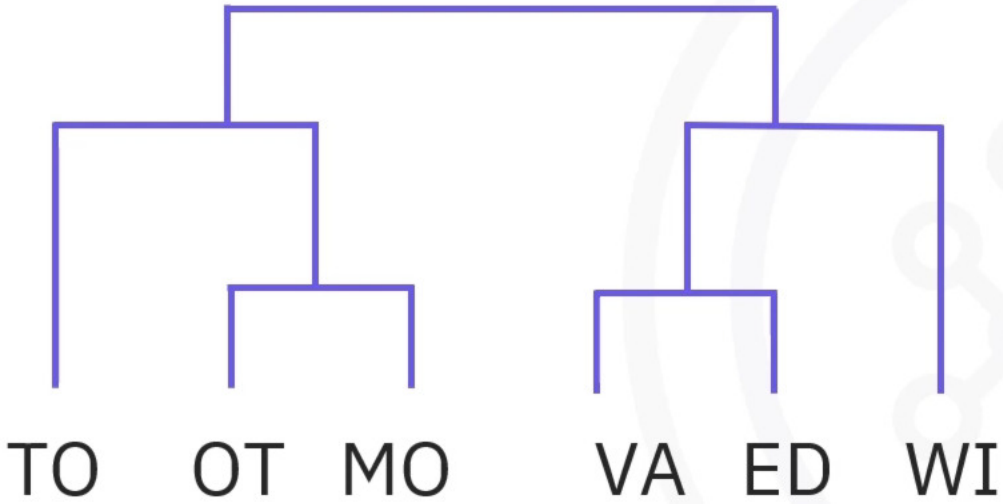
Density-based

- Creates clusters of any shape
- Suitable for irregular clusters
- Example: DBSCAN algorithm



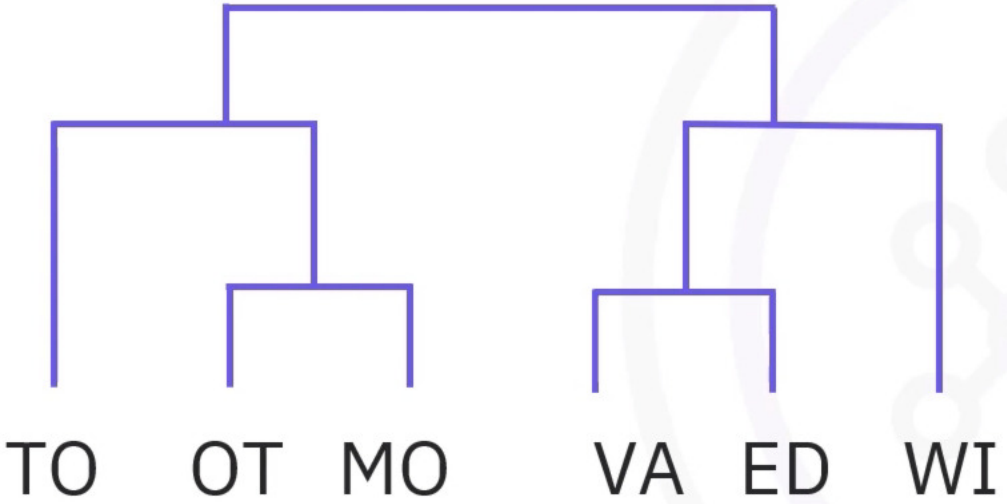
Hierarchical clustering

- Organizes data into nested clusters
- Contains smaller sub-clusters
- Generates a dendrogram
- Reveals relationships between clusters



Agglomerative: Merges clusters

Divisive: Splits clusters



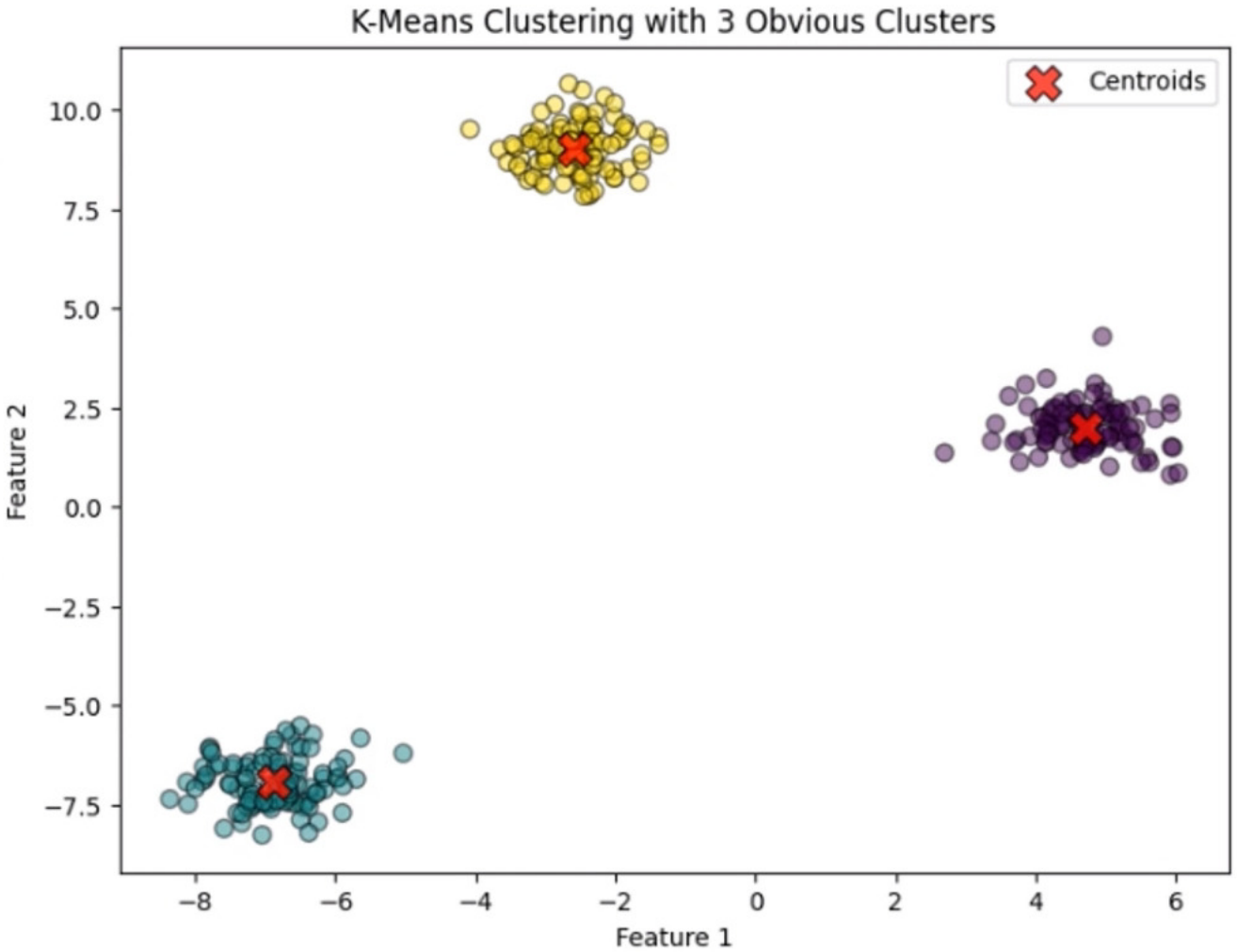
Intuitive for small to mid-sized data sets

Partition-based clustering

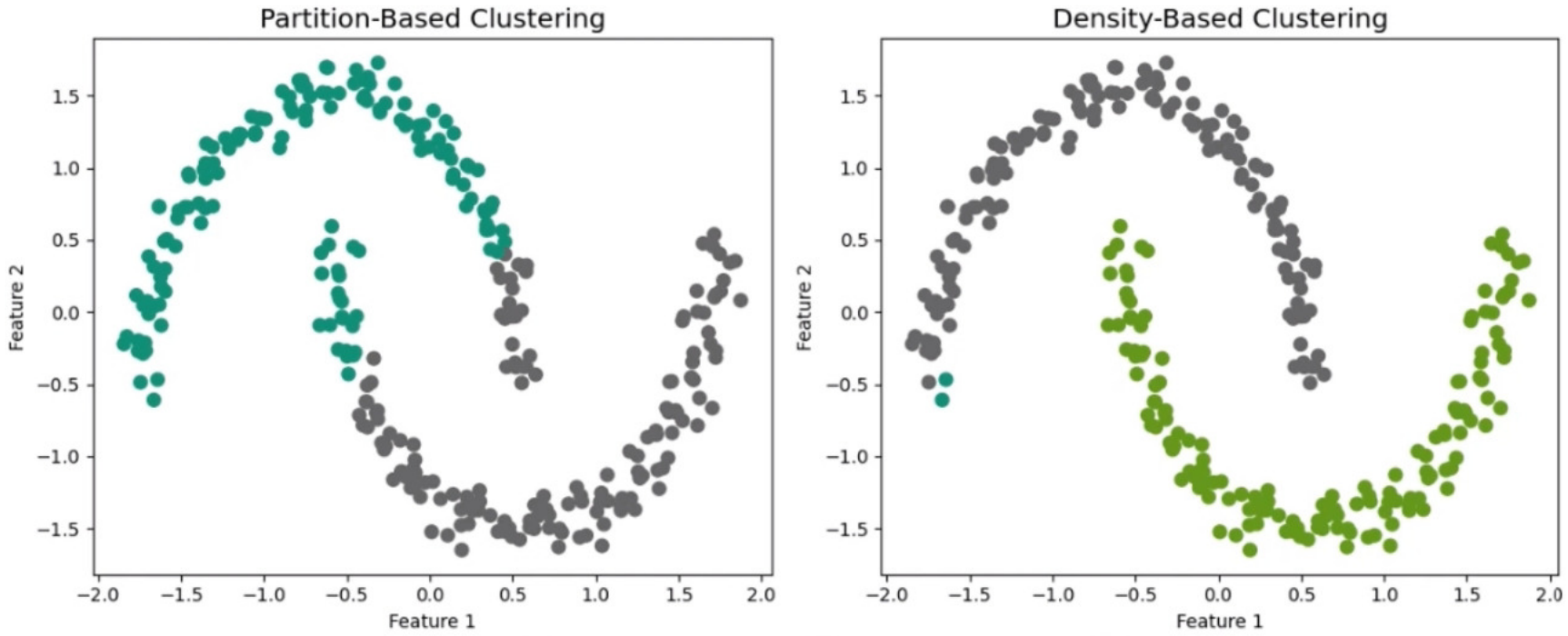
Uses the "make blobs" function

Generates three color-coded clusters

Displays clusters in a scatterplot



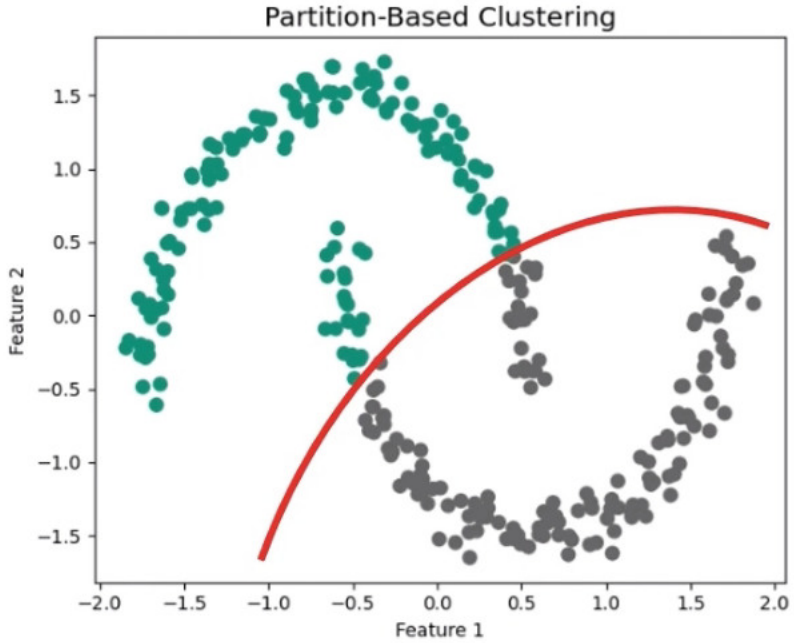
Partition and density-based clustering

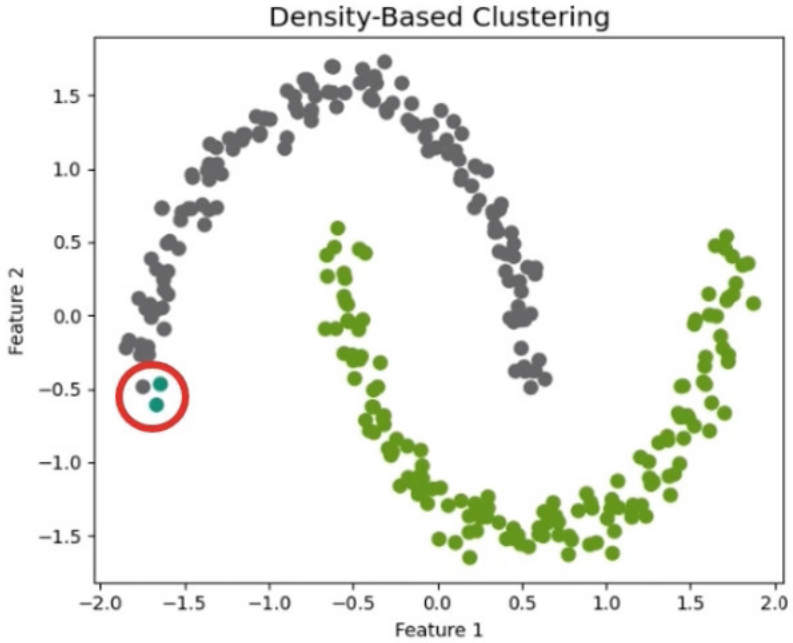


Uses the "make moons" function

Generates interlocking half-circles

Distinguishes clusters using color





Partition-based clustering struggles with separation

Partitions data along a red curve

Density-based clustering separates distinct shapes

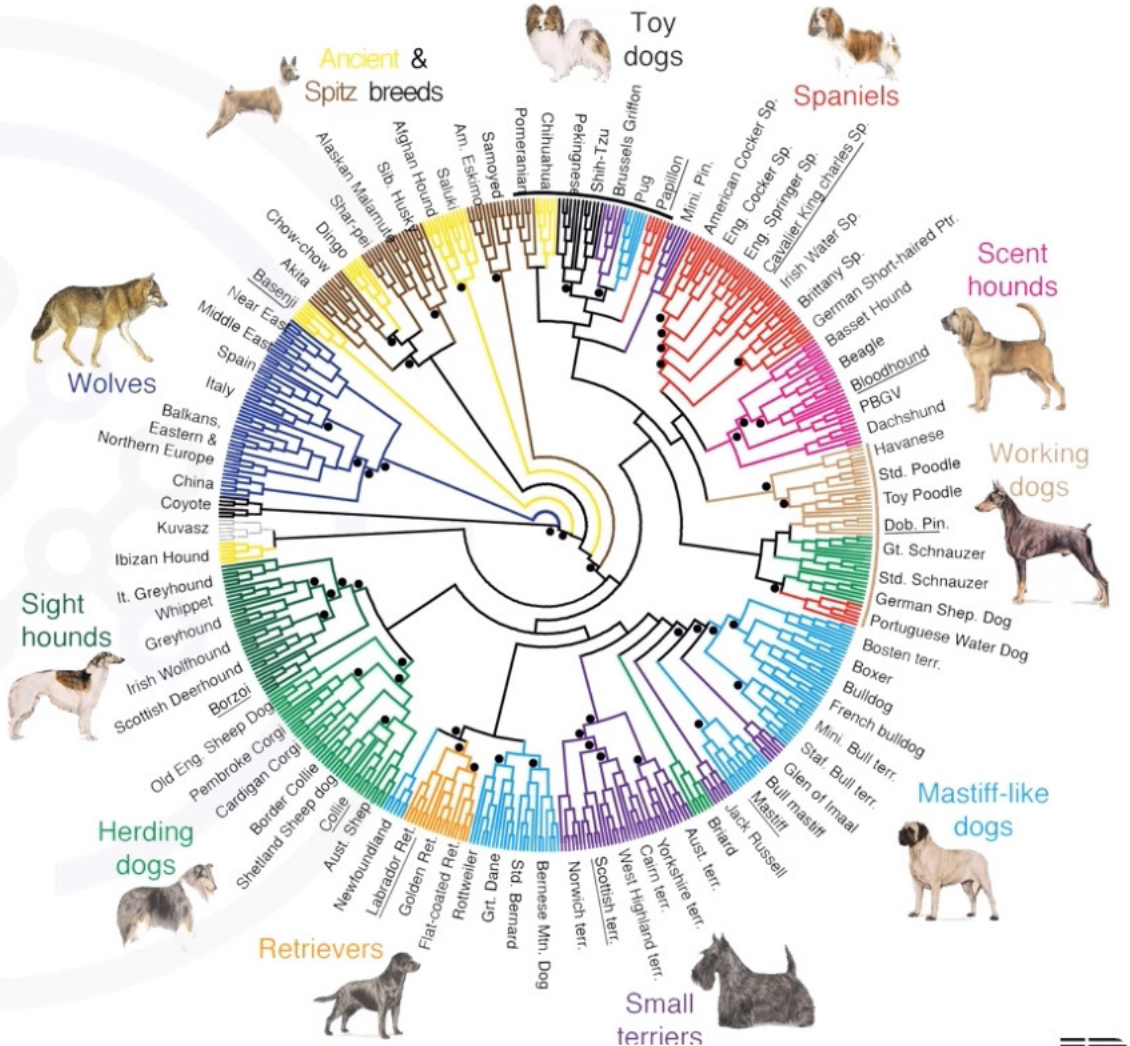
Creates third cluster of three points

Hierarchical clustering

Presents genetic data from over 900 dogs

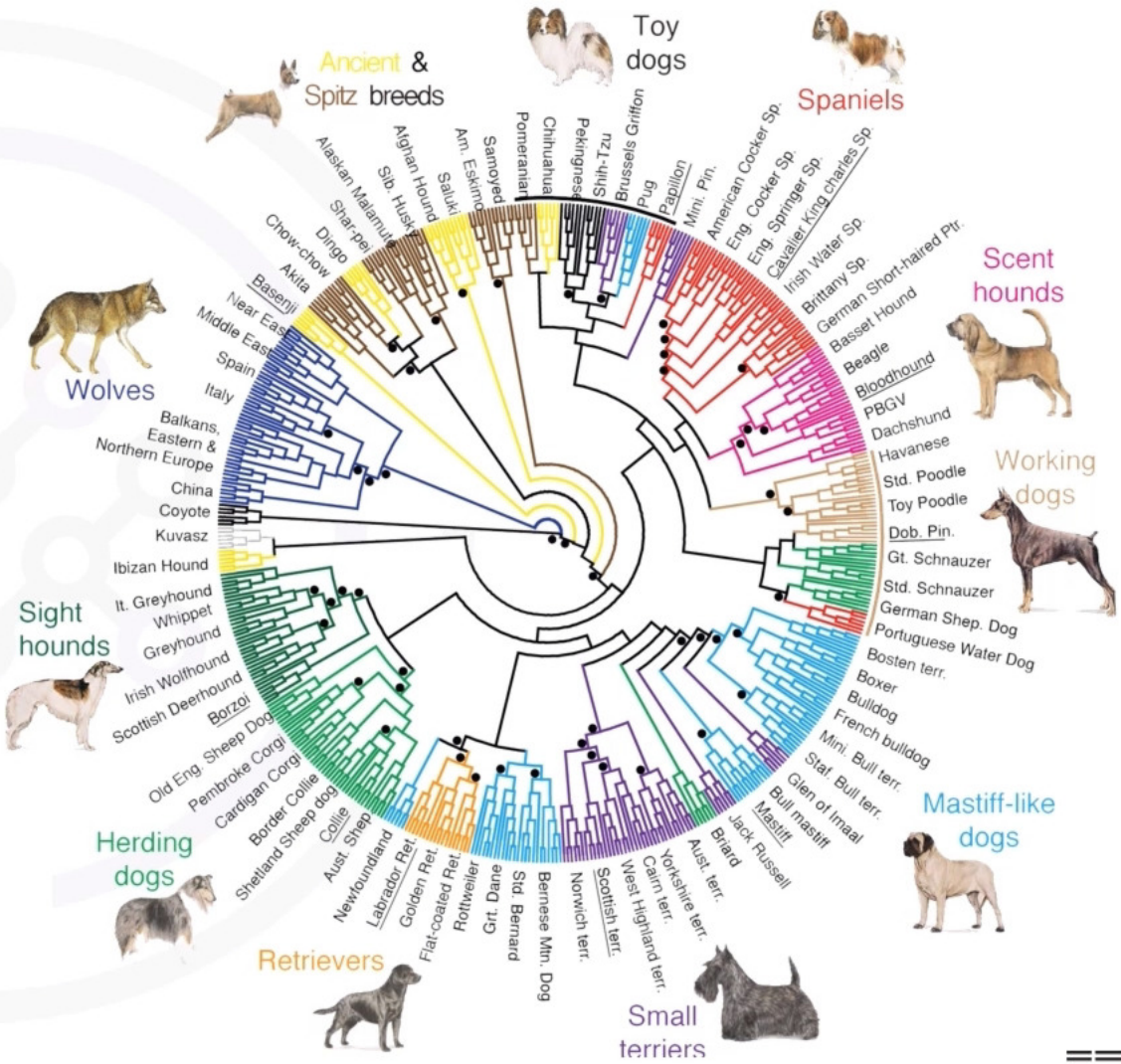
Analyzed 48,000 genetic markers

Illustrates hierarchical clustering of animal groups

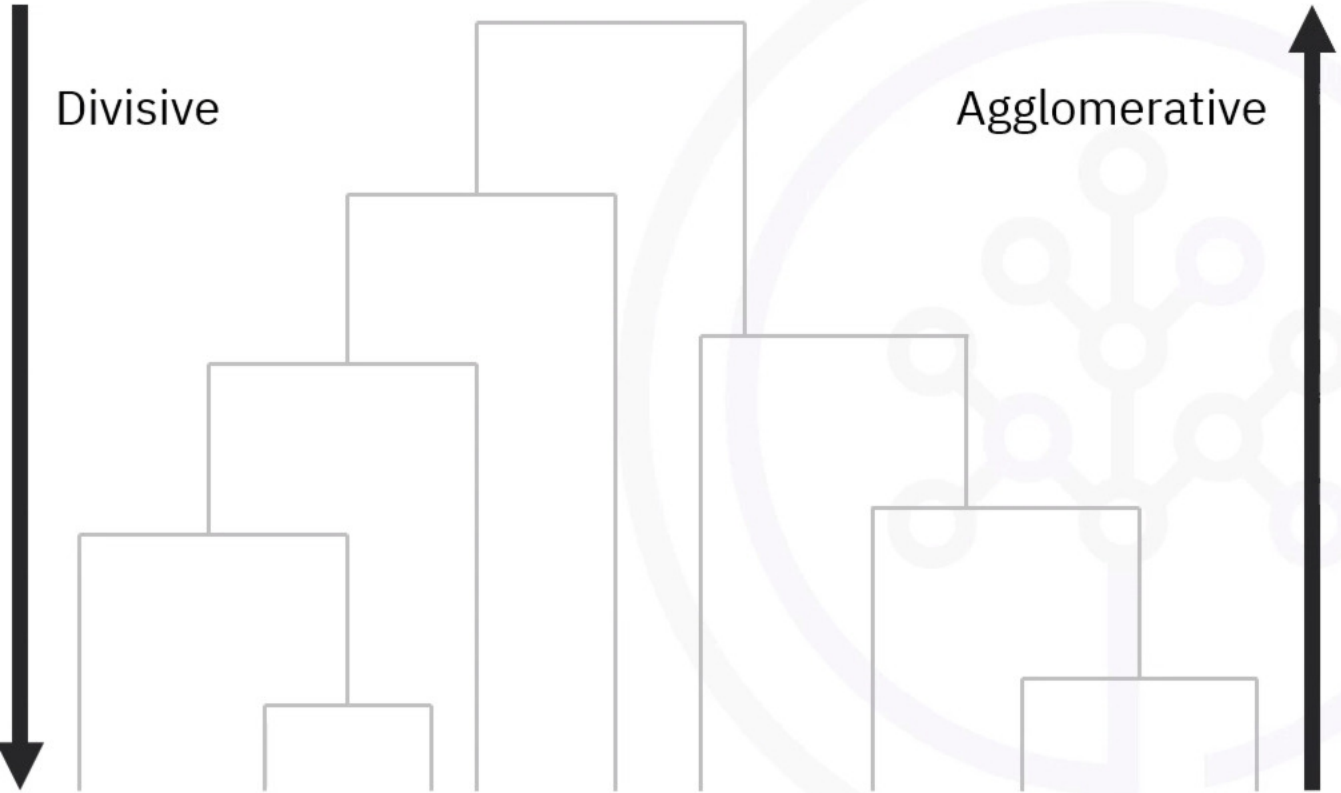


Groups animals based on genetic similarities

Displays tree-like structure of nested clusters

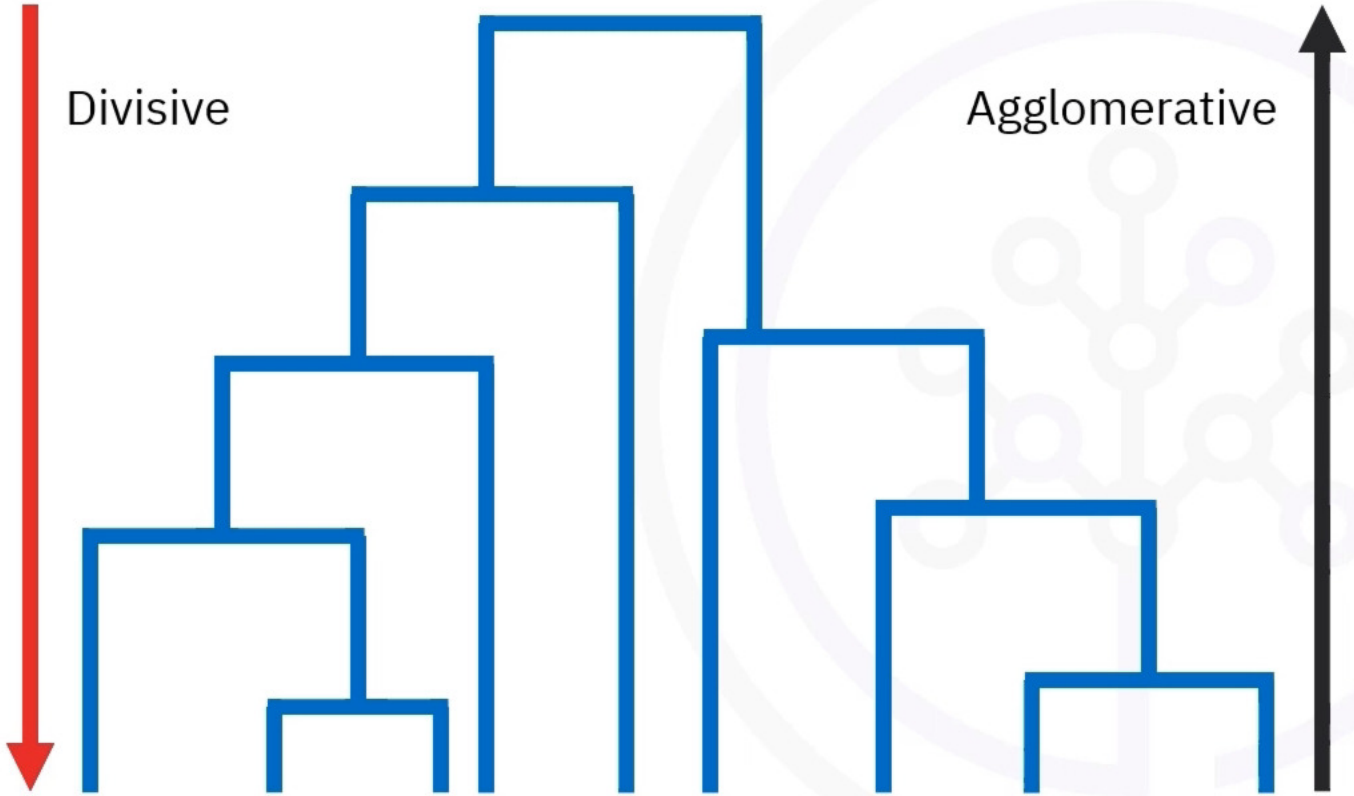


Hierarchical clustering



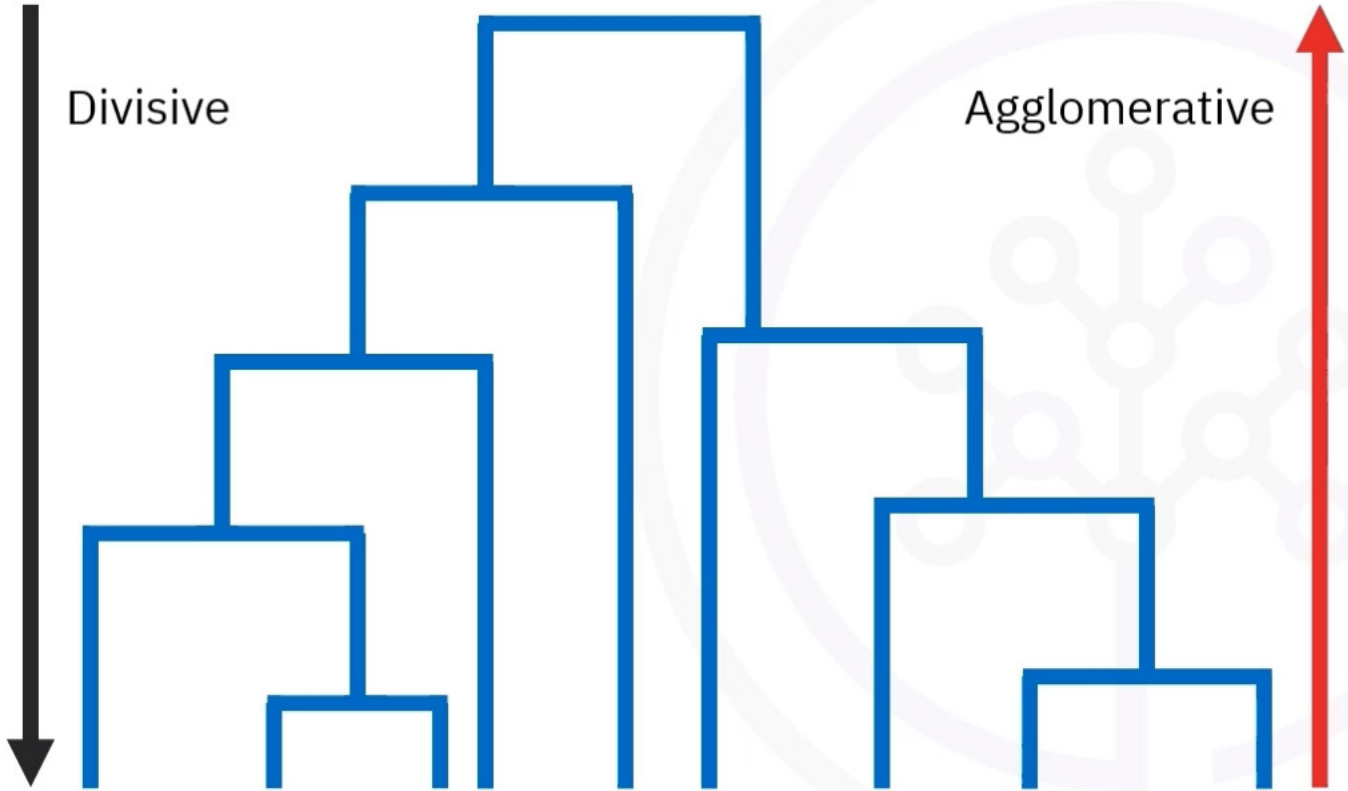
Divisive

Agglomerative



Divisive clustering

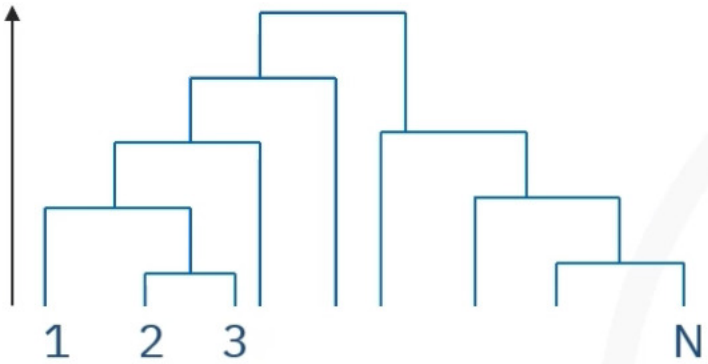
- Uses a top-down approach
- Starts with a single root cluster
- Iteratively splits into smaller child clusters



Agglomerative clustering

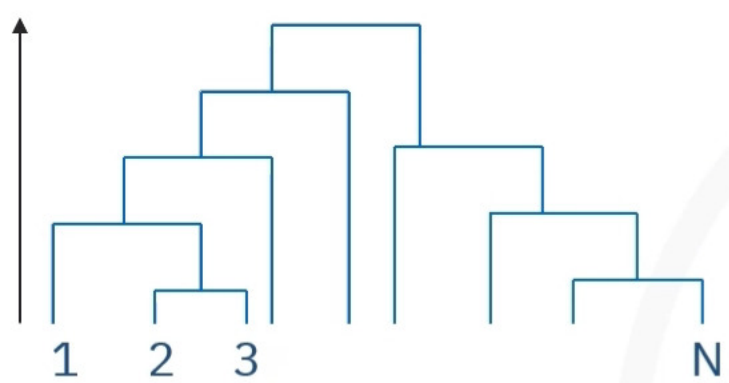
- Employs a bottom-up approach
- Starts with individual clusters
- Merges similar clusters into larger parents

Agglomerative hierarchical clustering



$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{bmatrix}$$

- Uses a bottom-up clustering approach
- Select a metric to measure distance
- Measure distance between cluster centroids

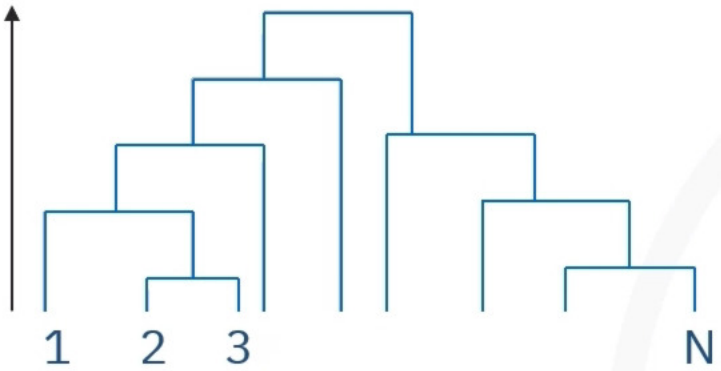


$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{bmatrix}$$

Initialize N clusters with single data points

Compute a distance matrix for clusters

Display distances between each pair of points

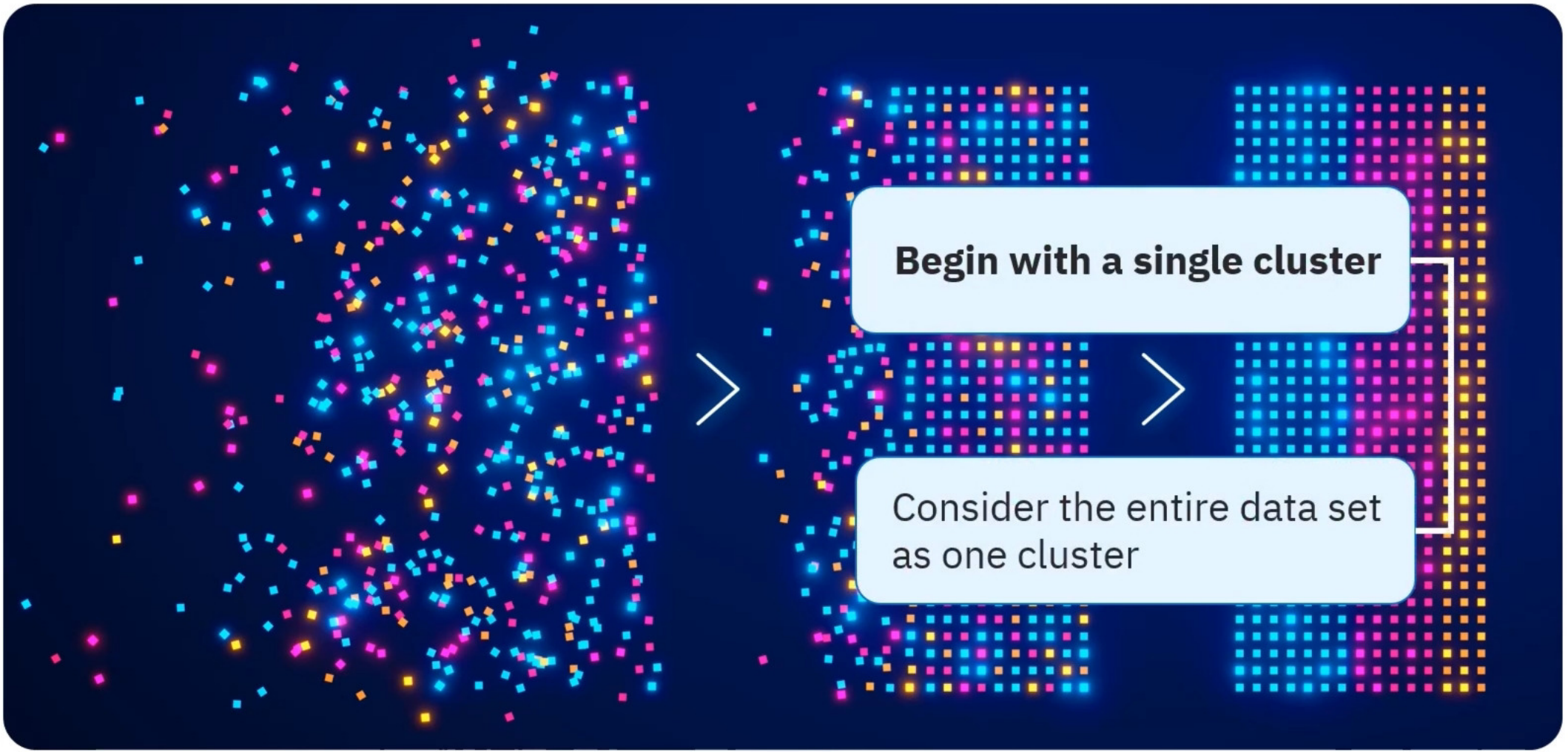


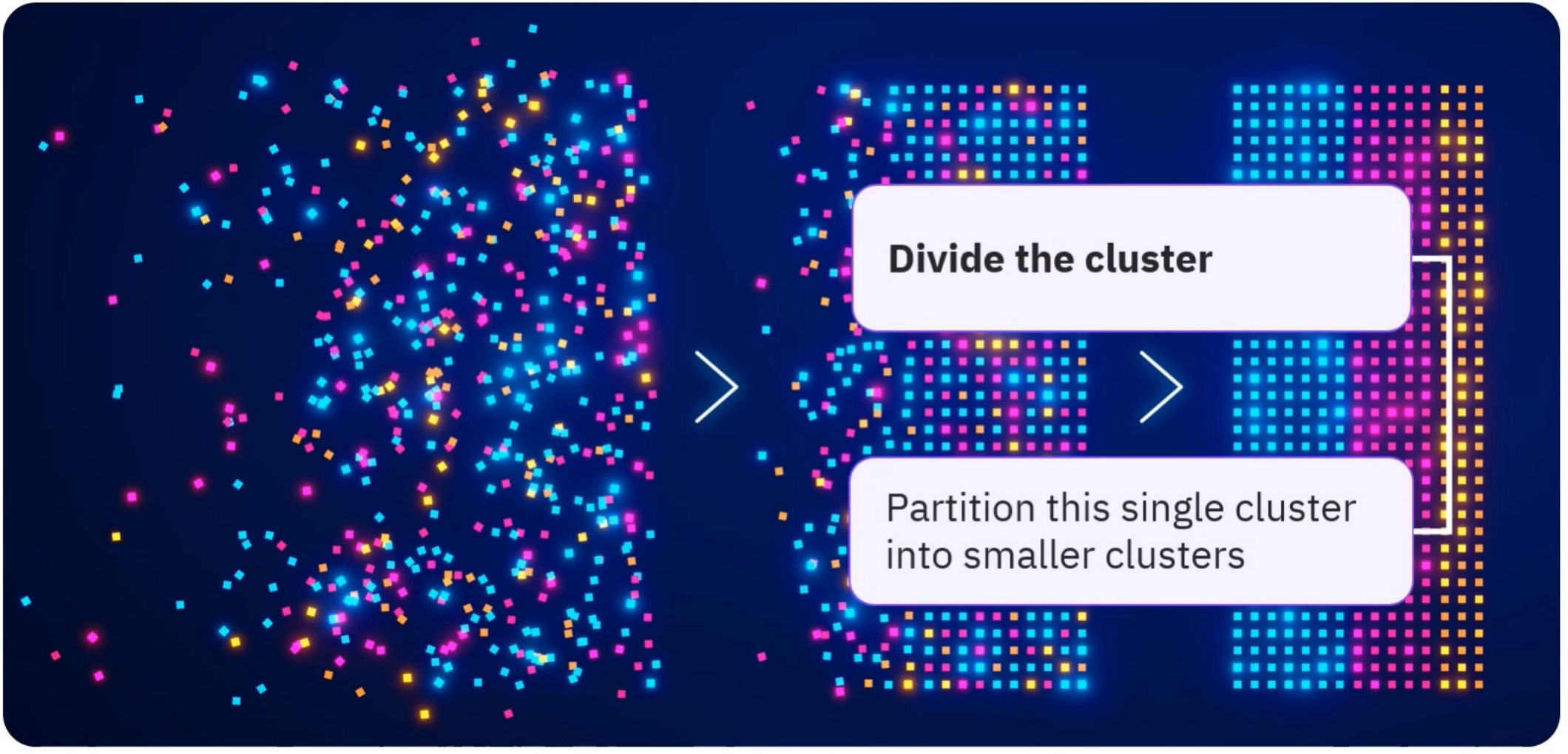
$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{bmatrix}$$

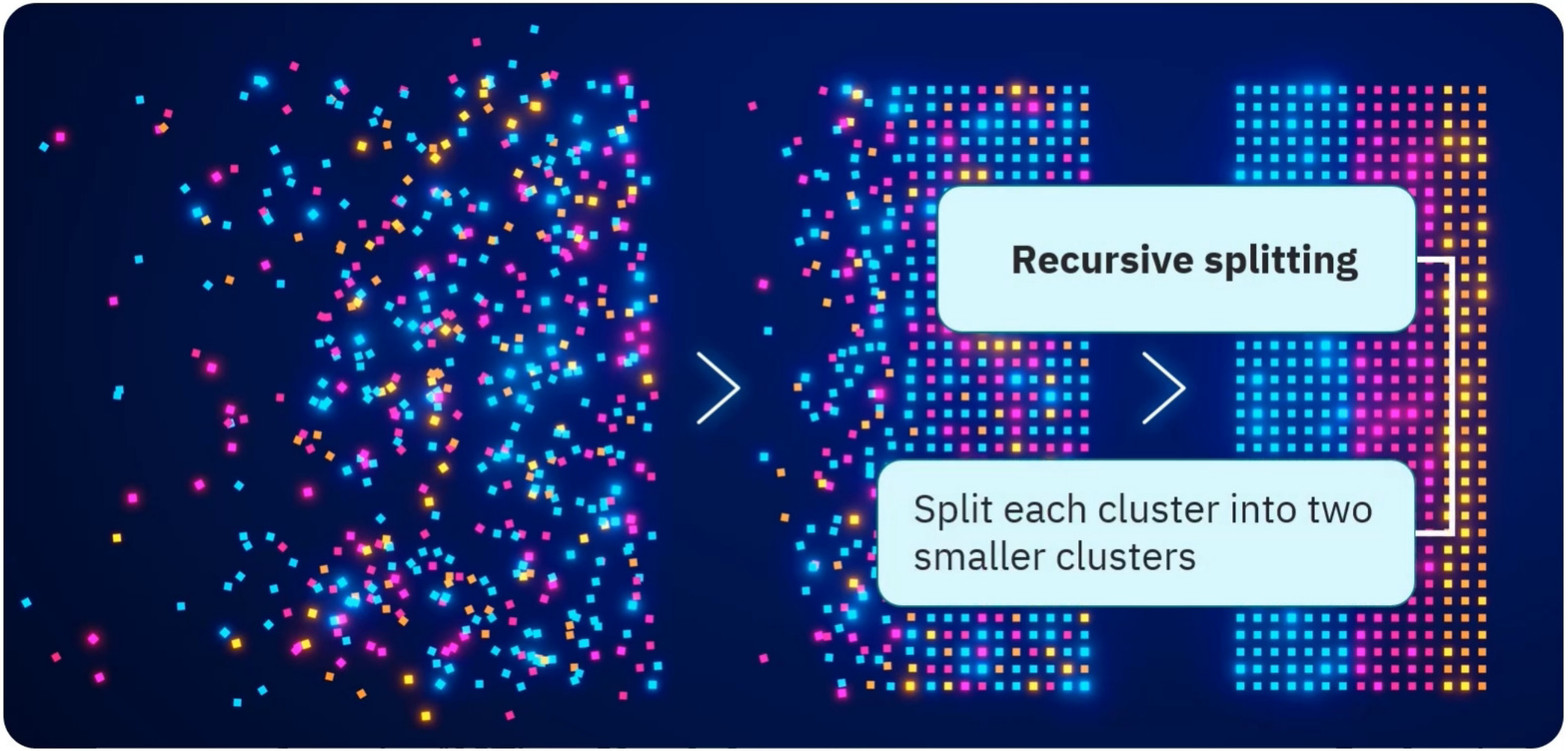
Merge the two closest clusters

Update the proximity matrix

Divisive hierarchical clustering



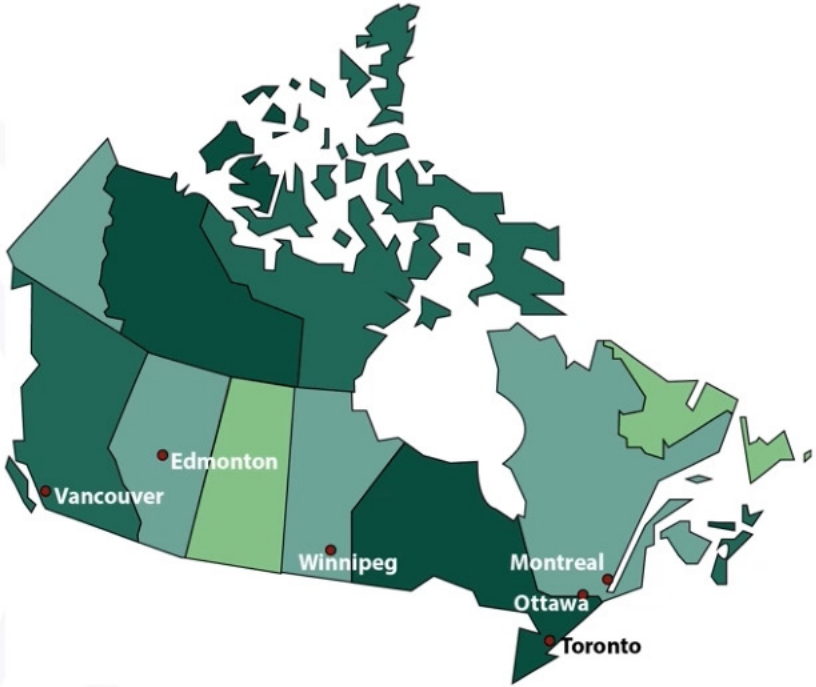

Divide the cluster
Partition this single cluster into smaller clusters


Recursive splitting
Split each cluster into two smaller clusters


Stopping criterion
Defined as a specific number of clusters

Agglomerative clustering

	TO	OT	VA	MO	WI	ED
TO		351	3363	505	1510	2699
OT			3543	167	1676	2840
VA				3690	1867	819
MO					1824	2976
WI						1195
ED						

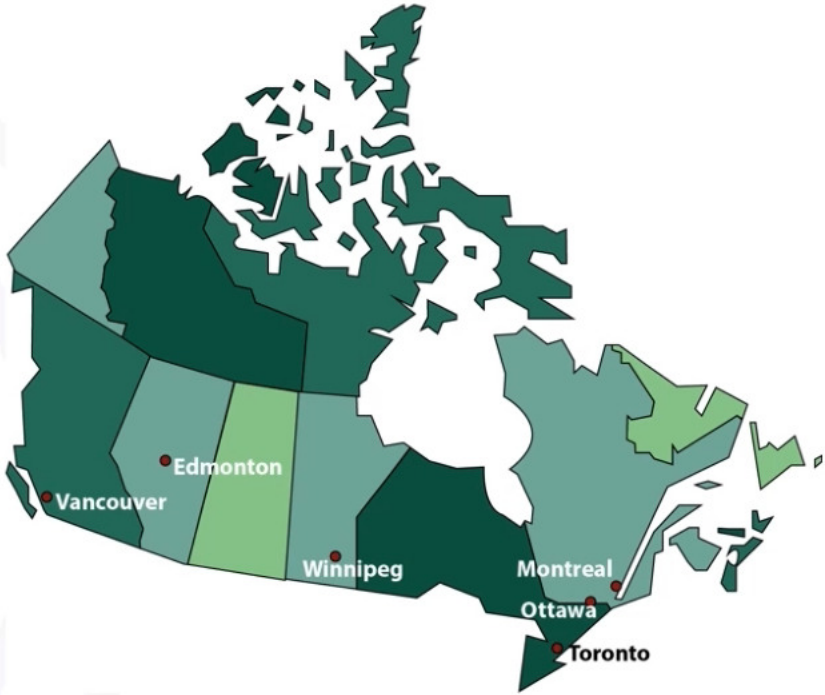


Group six Canadian cities by distances

Use a distance matrix to represent distances

TO OT MO VA ED WI

	TO	OT	VA	MO	WI	ED
TO		351	3363	505	1510	2699
OT			3543	167	1676	2840
VA				3690	1867	819
MO					1824	2976
WI						1195
ED						



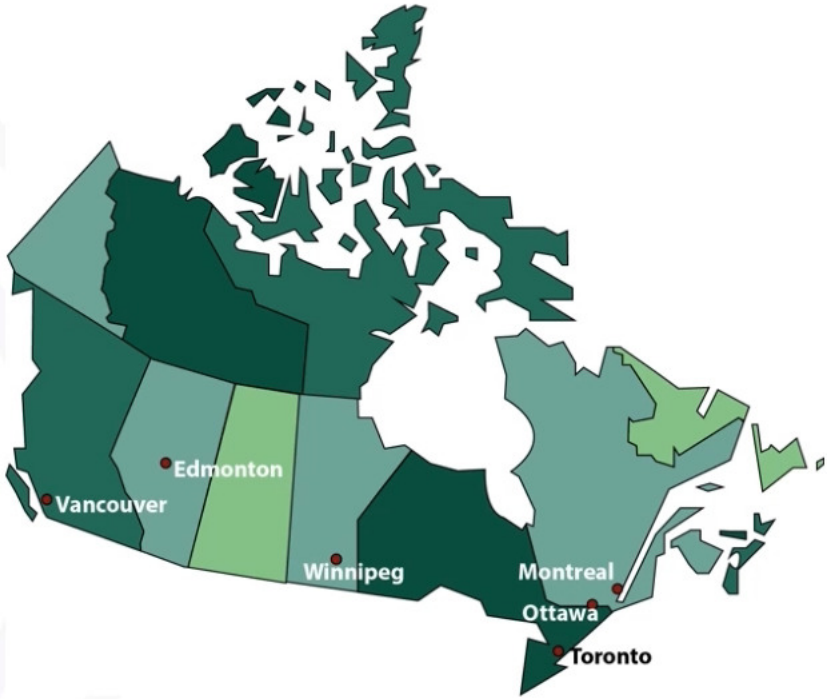
Starts with six clusters for cities

Identifies clusters to merge based on distance

Shows closest clusters in the distance matrix

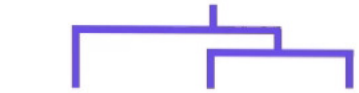
TO OT MO VA ED WI

	TO	OT	VA	MO	WI	ED
TO		351	3363	505	1510	2699
OT			3543	167	1676	2840
VA				3690	1867	819
MO					1824	2976
WI						1195
ED						



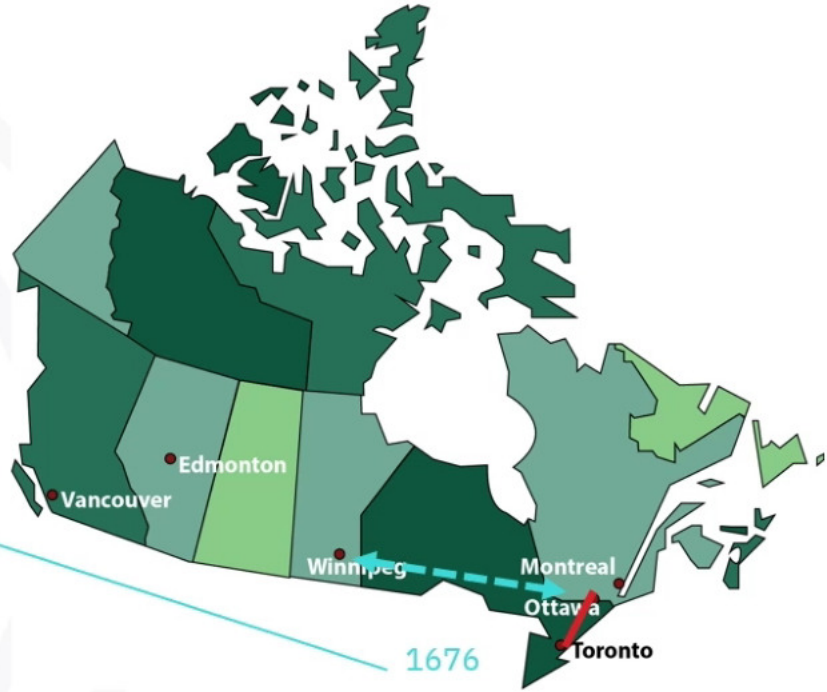
Combines clusters into the next parent cluster

Visualizes hierarchy with a dendrogram



TO OT MO VA ED WI

	TO	OT/MO	VA	WI	ED
TO		351	3363	1510	2699
OT/MO			3543	1676	2840
VA				1867	819
WI					1195
ED					



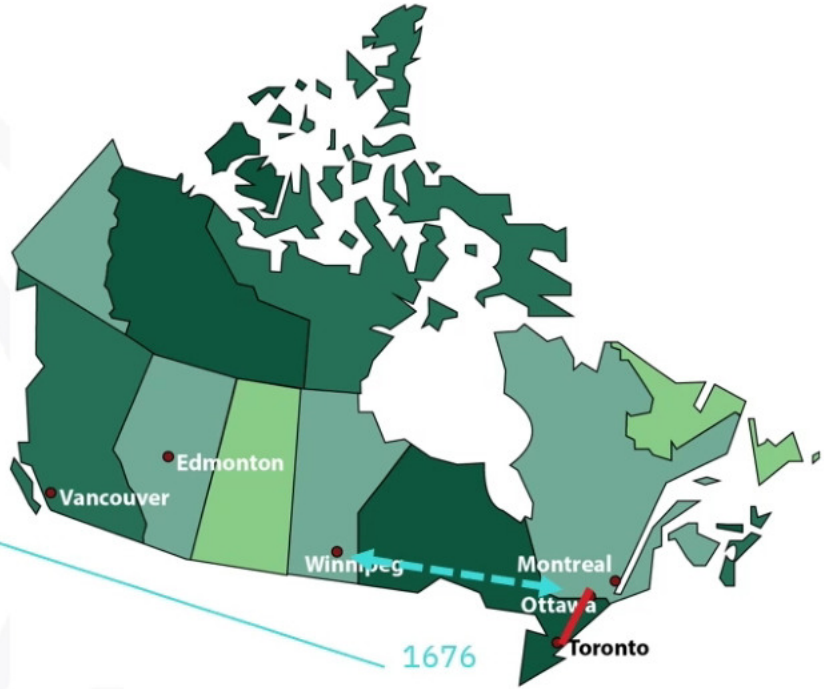
Distance matrix combines Montreal and Ottawa rows

Creates the Ottawa-Montreal cluster

Updates distances to the new cluster

Calculates midpoint between the two cities

	TO	OT/MO	VA	WI	ED
TO		351	3363	1510	2699
OT/MO			3543	1676	2840
VA				1867	819
WI					1195
ED					



Look for the closest clusters again

Form the Toronto-Ottawa-Montreal cluster

Visualize hierarchy with a dendrogram

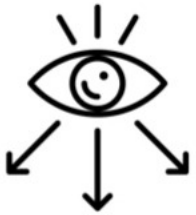
	TO/OT/MO	VA	WI	ED
TO/OT/MO		3543	1676	2840
VA			1867	819
WI				1195
ED				

	TO/OT/MO	VA/ED/WI
TO/OT/MO		1676
VA/ED/WI		

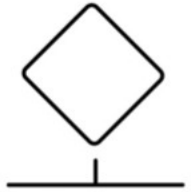
Recap

- Explain the concept of clustering
- Apply k-means clustering to segment customers
- Explain how density-based clustering works with irregular clusters
- Explain hierarchical clustering and dendrogram generation
- Explain strategies for hierarchical clustering
- Analyze agglomerative hierarchical clustering's bottom-up approach
- Analyze divisive hierarchical clustering's top-up approach

What is k-means clustering?



An iterative, centroid-based clustering algorithm



Partitions data into similar groups based on distance between centroids

Divides data into
k non-overlapping clusters

K-clusters have minimal
variances around centroids
and maximal dissimilarity
between clusters

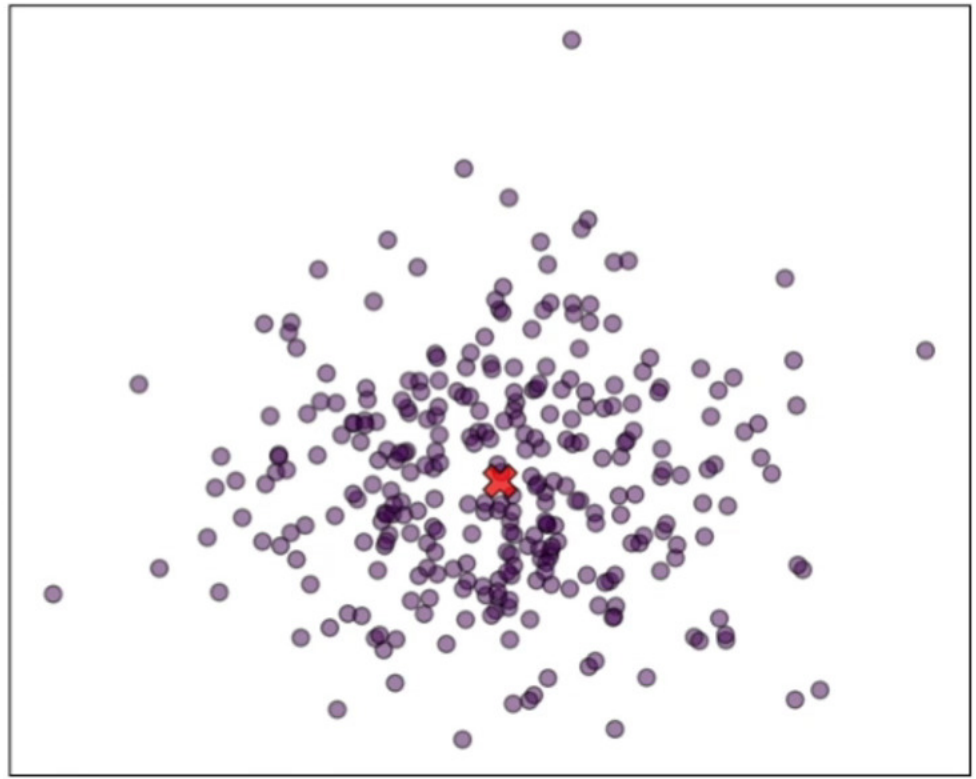
K-means algorithm

Data points nearest to centroid grouped together

Higher k = Smaller clusters with greater detail

Lower k = Larger clusters with less detail

Cluster and its centroid



1. **Initialize the algorithm:**

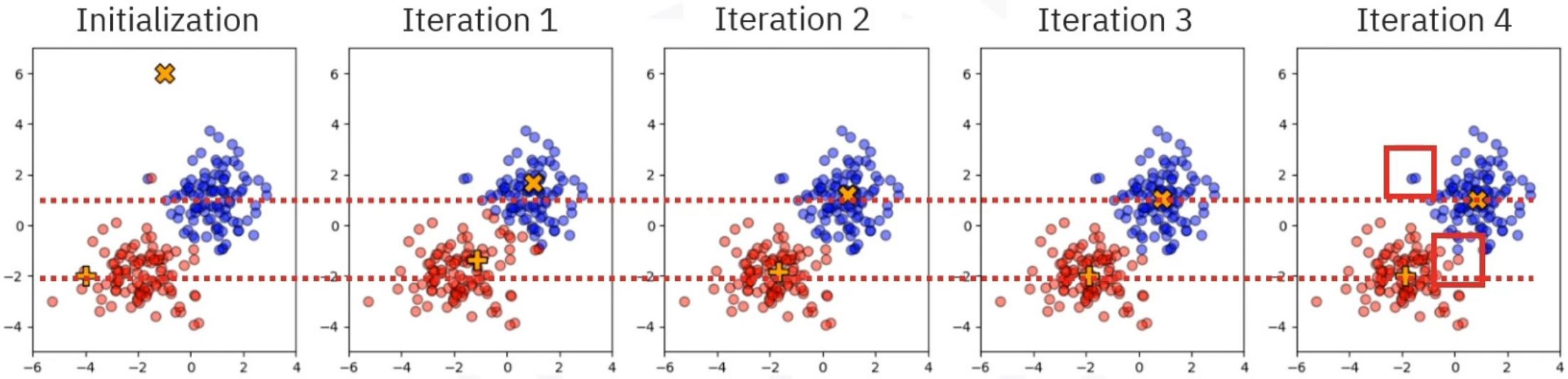
- Select the number of clusters, k
- Randomly select k centroids

2. **Iteratively assign points to clusters and update centroids:**

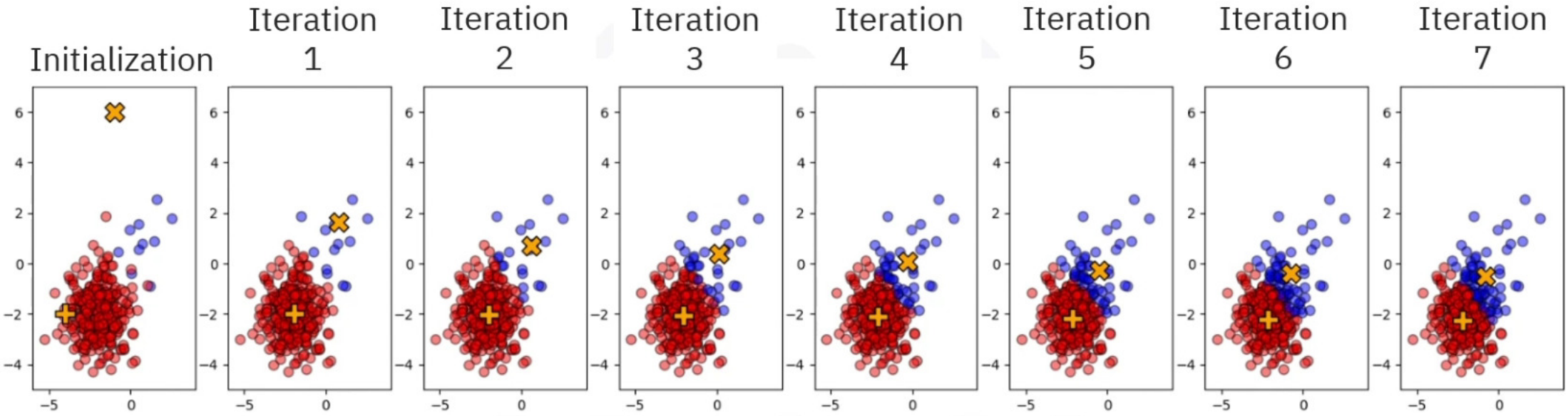
- Compute distance matrix
- Assign each point to cluster with nearest centroid
- Update cluster centroids as the mean position

3. **Repeat until centroids stabilize or max iterations reached**

K-means clustering in action

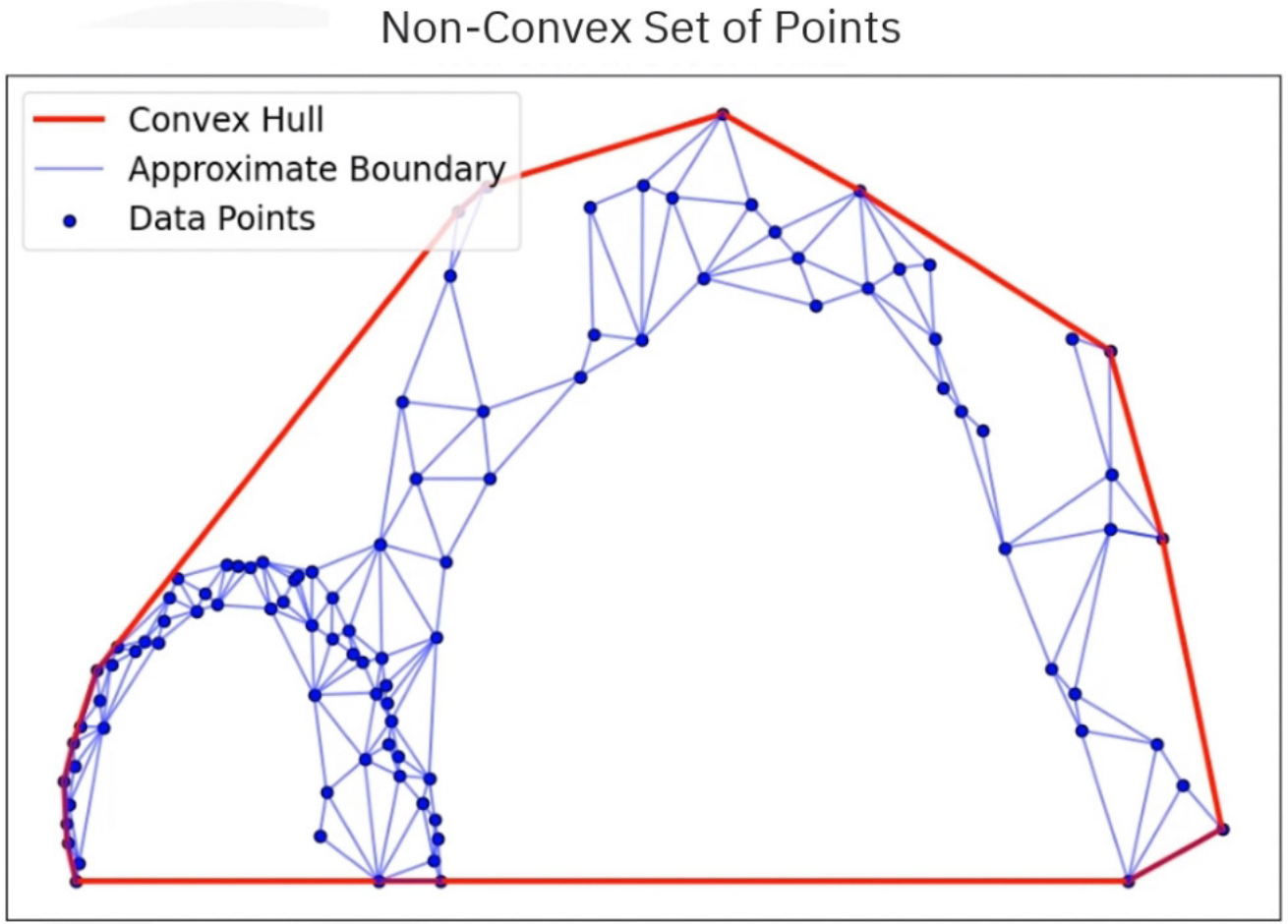


K-means failure with imbalanced clusters



K-means clustering considerations

- Assumes convex clusters
- Assumes balanced cluster sizes
- Sensitive to outliers and noise
- Scales well to big data



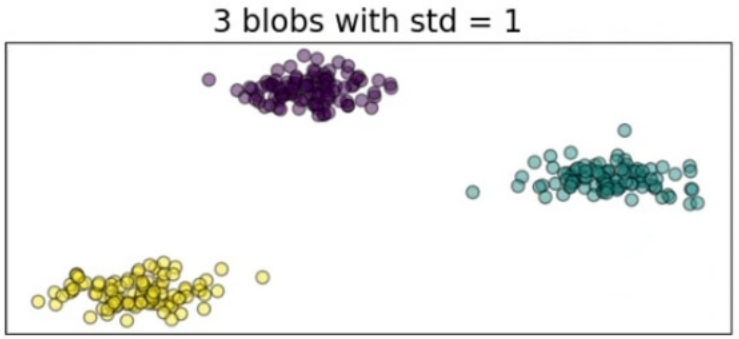
K-means optimization

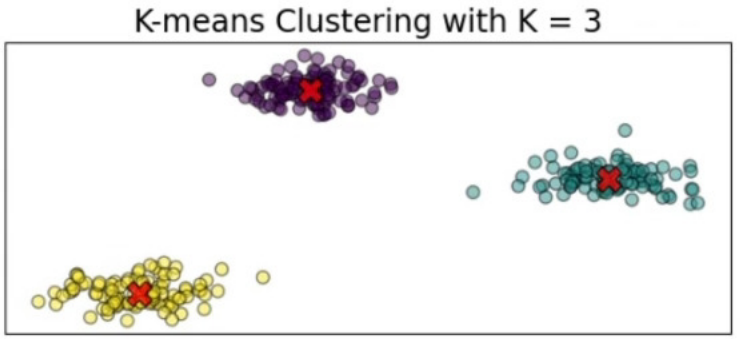
Goal: Minimize within-cluster sum of squares:

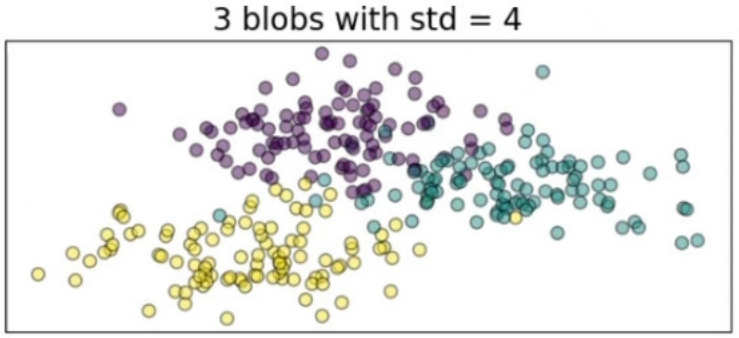
$$\sum_{i=1}^{K} \sum_{x \in C_i} \|x - \mu_i\|^2$$

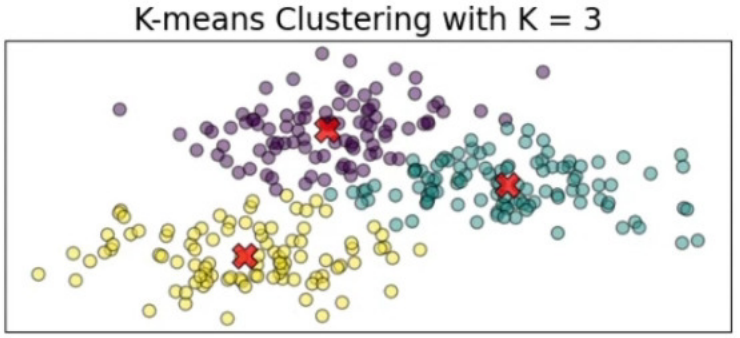
- K = Number of clusters
- C_i = i th cluster
- x = Data point
- μ_i = Centroid of cluster C_i
- $\|x - \mu_i\|^2$ = Squared distance between x and its cluster centroid

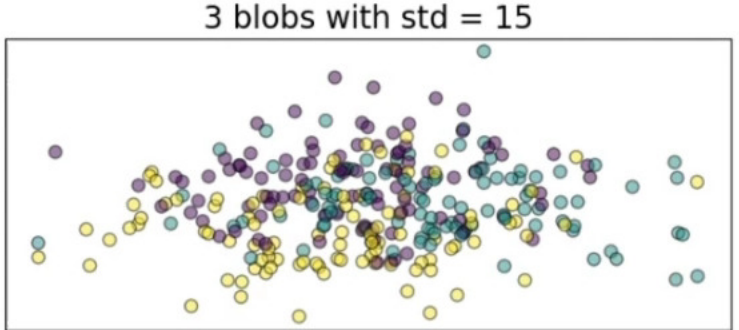
K-means experiments: K=3

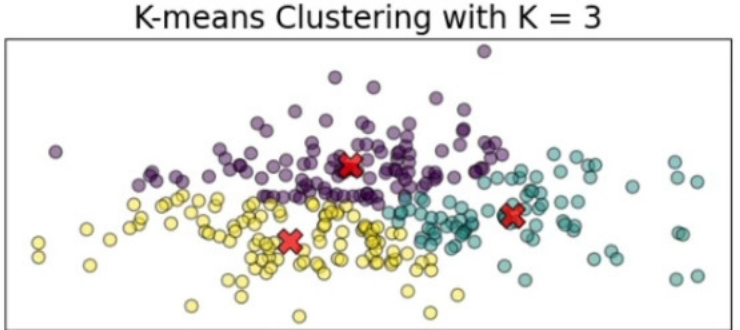




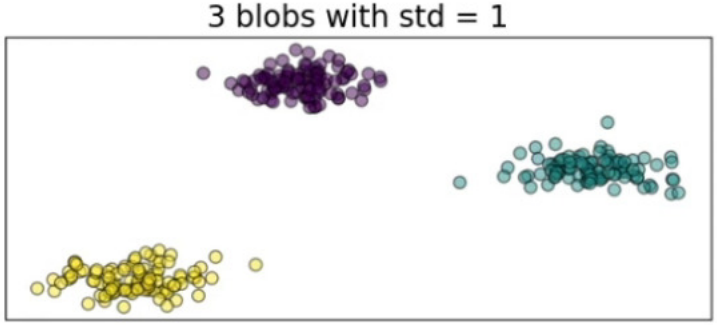


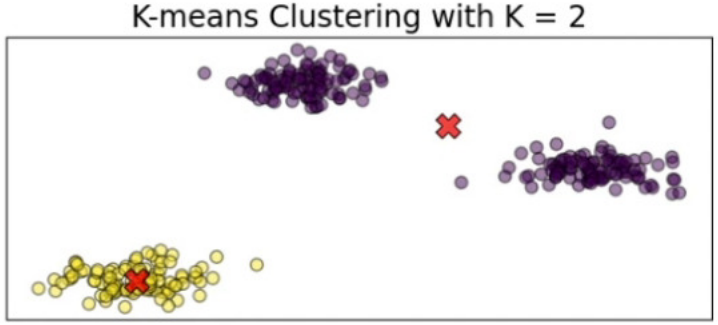


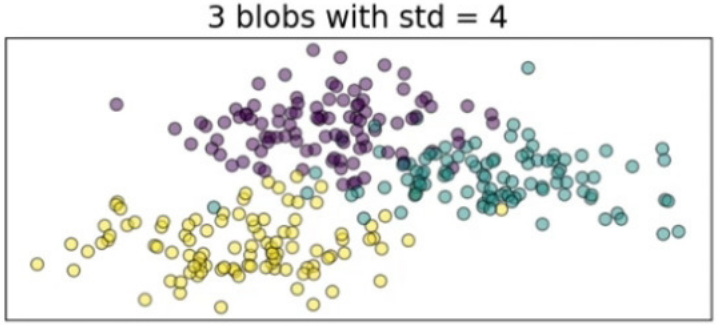


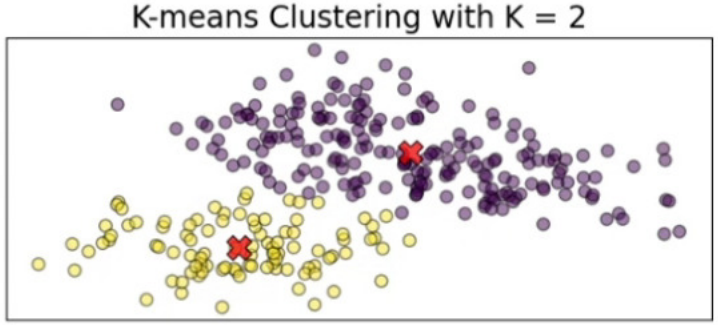


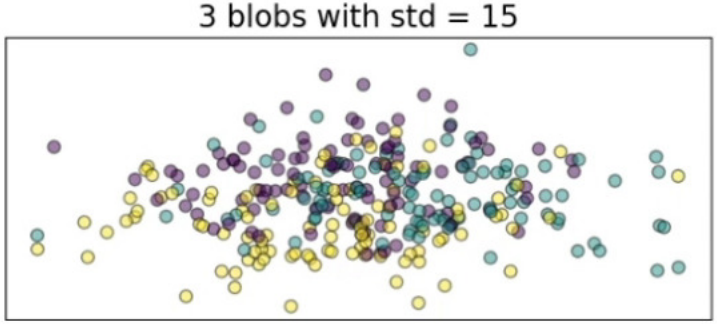
K-means experiments: K=2

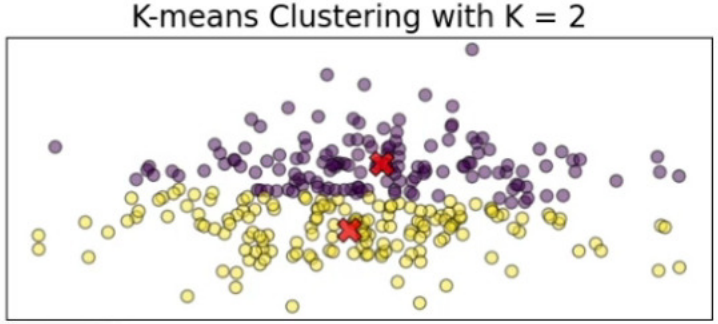






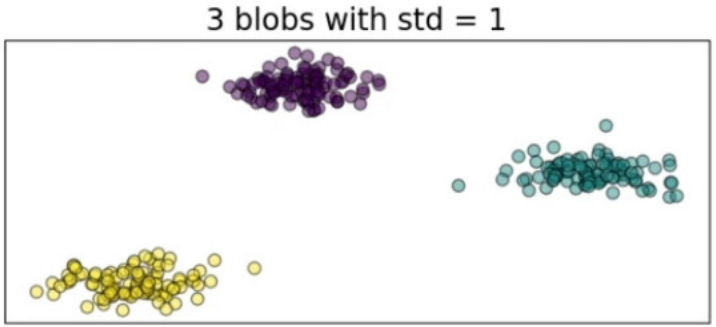


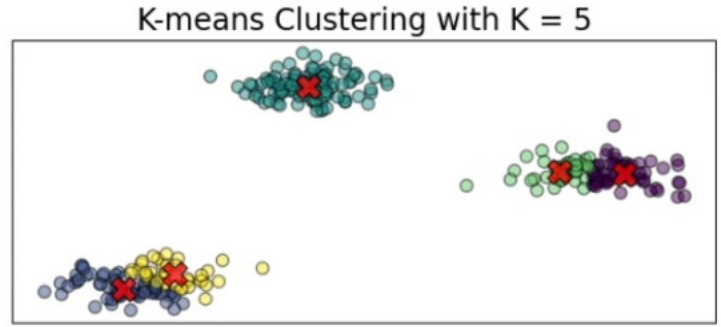


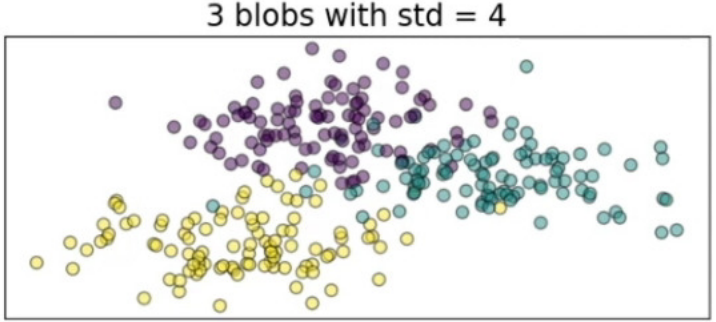


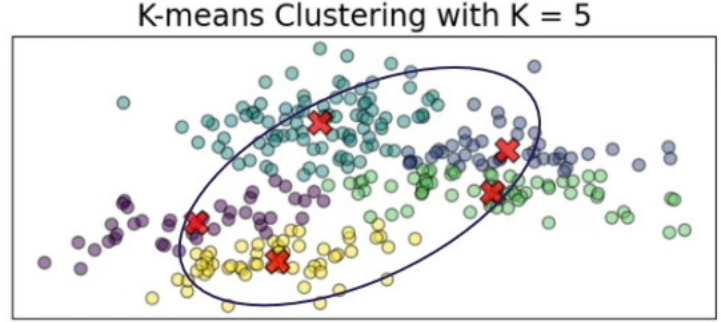
Centroids are merging

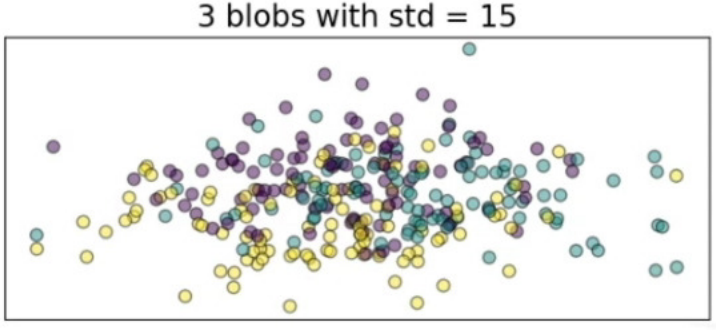
K-means experiments: K=5

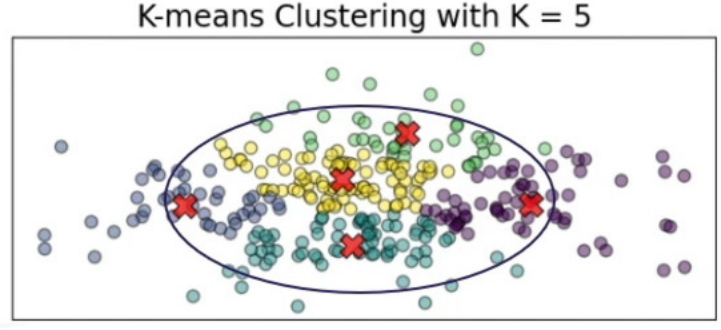








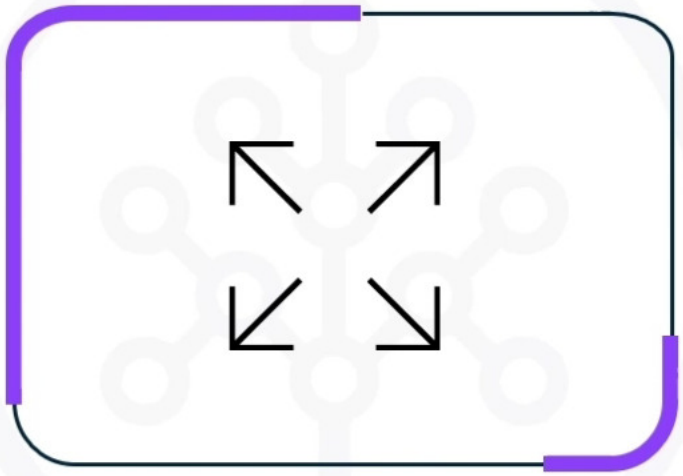




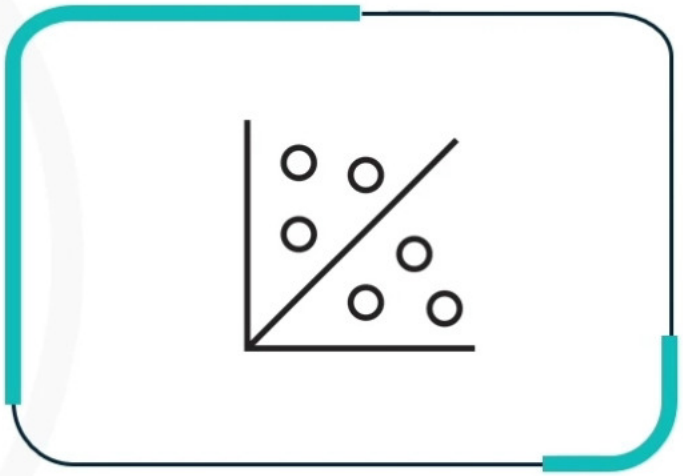
Determining k

Choosing k is feasible when:

Data is separable

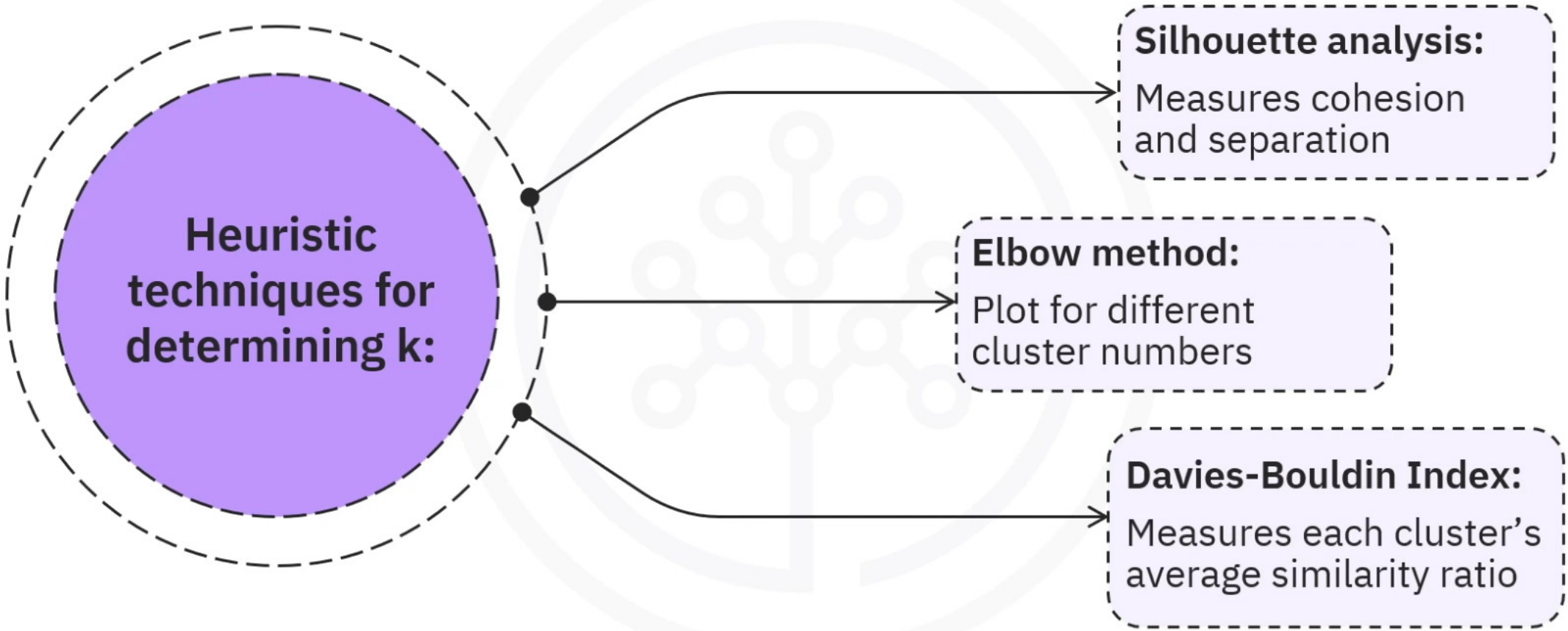


Difficult to visualize for high-dimensional spaces



Consider scatterplots between variable pairs to check for separability

Determining k



Recap

K-means:

- Iterative, centroid-based clustering algorithm
- Partitions data set into similar groups
- Clustering algorithm categorizes data points into clusters
- Doesn't perform well on imbalanced clusters and assumes that clusters are convex
- Objective: Minimize within-cluster variance
- Heuristic techniques for gauging k-means performance:
  - Silhouette analysis
  - Elbow method
  - Davies-Bouldin index

DBSCAN clustering

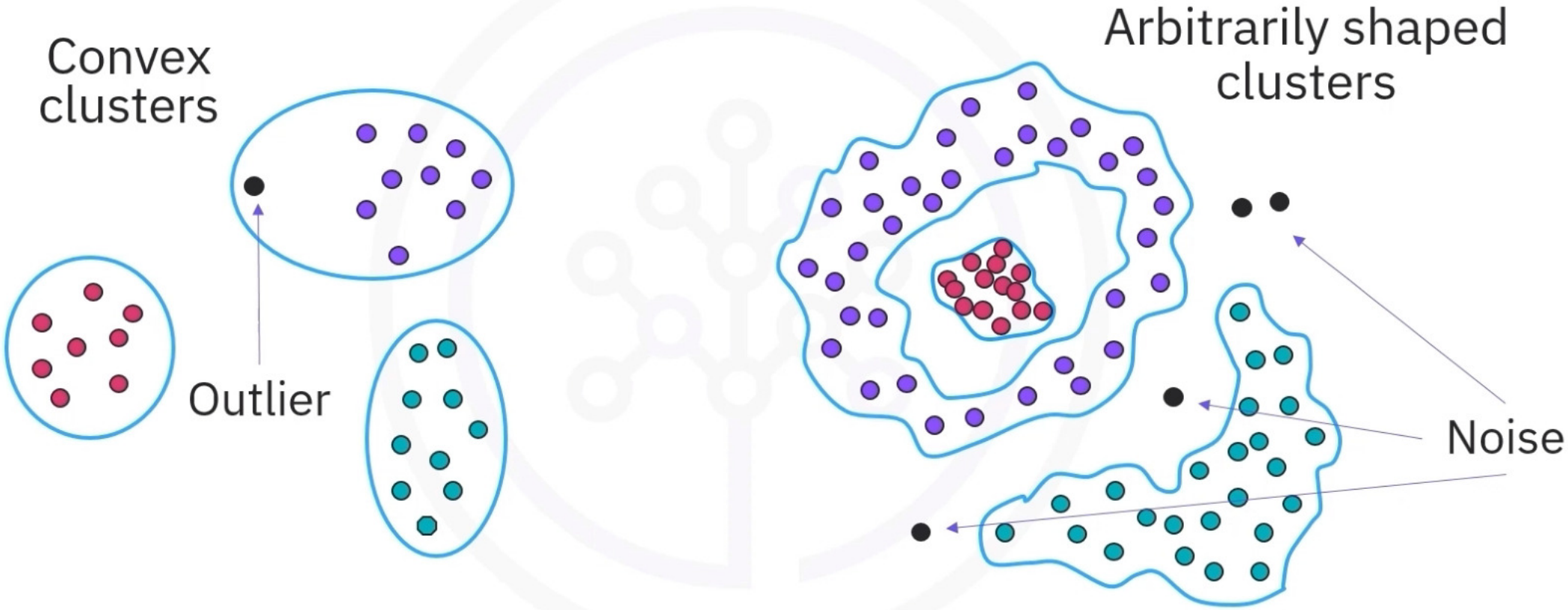
- Density-based spatial clustering algorithm
- Creates clusters centered around spatial centroids
- User provides density value
- Defines neighborhoods of clusters with specified density

Discovers clusters of any shape, size, or density

Distinguishes between data points that are part of a cluster and noise

Useful for data with outliers or when cluster number is unknown

Centroid and density-based clustering

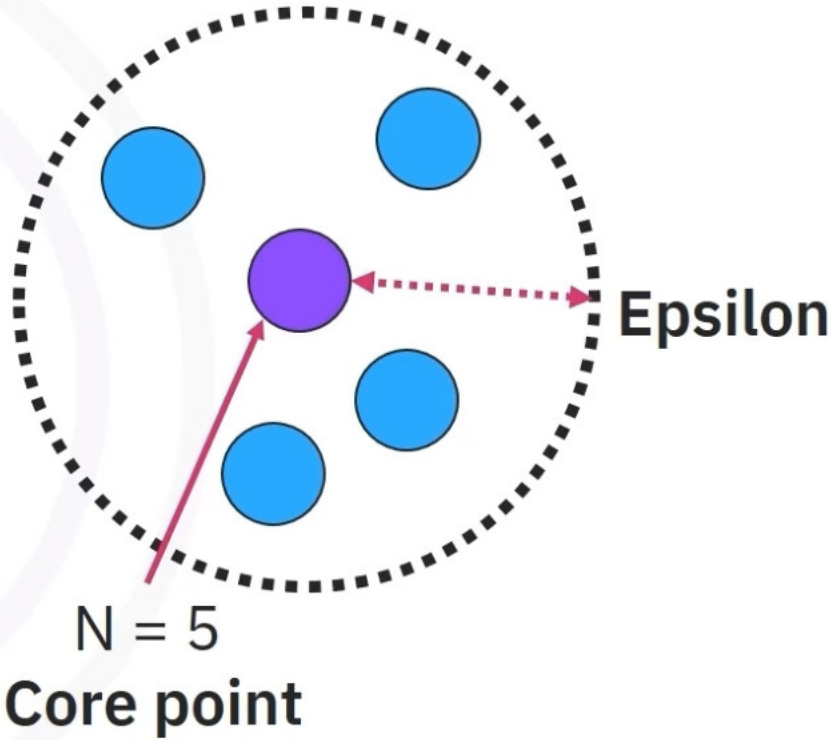


DBSCAN algorithm

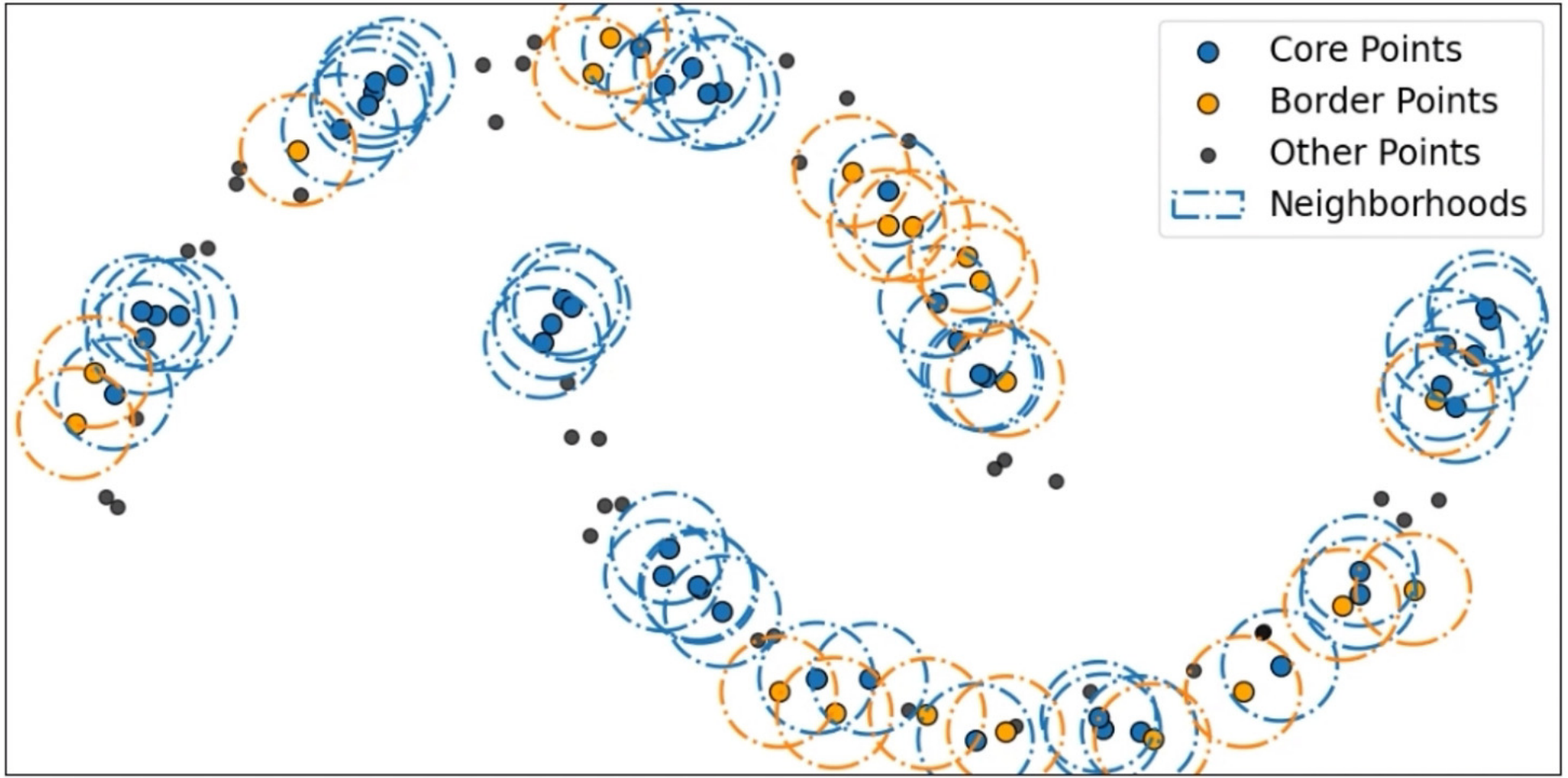
Parameters: N, epsilon

Every data point is labelled as one of the following:

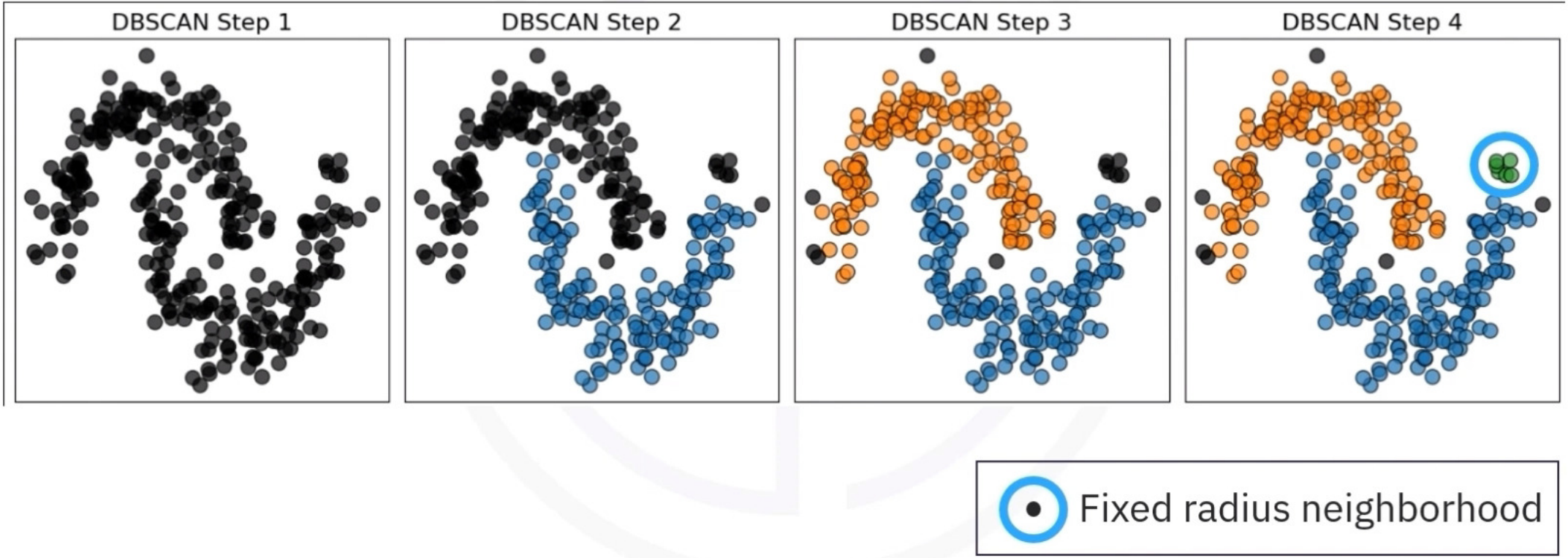
- **Core point:** Has at least N points within the epsilon neighborhood
- **Border point:** Non-core points within a core-point neighborhood
- **Noise point:** Isolated from all core-point neighborhoods
- **Clusters:** Core points and border points
- **Noise:** Non-clustered points



Core and border points

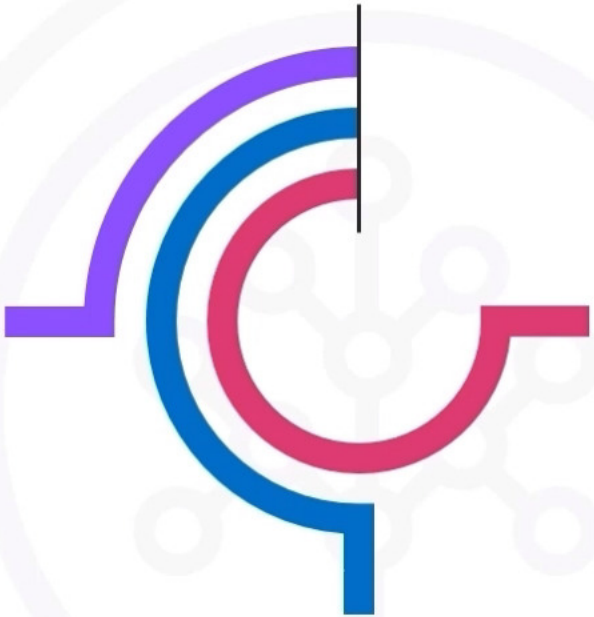


DBSCAN experiment


DBSCAN Step 1
DBSCAN Step 2
DBSCAN Step 3
DBSCAN Step 4
Fixed radius neighborhood

HDBSCAN clustering

Cluster stability: Cluster persistence over a range of neighborhood sizes



Improves clustering for variable-density and noisy data

HBDSCAN locally adjusts neighborhood radii for cluster stability

HDBSCAN algorithm

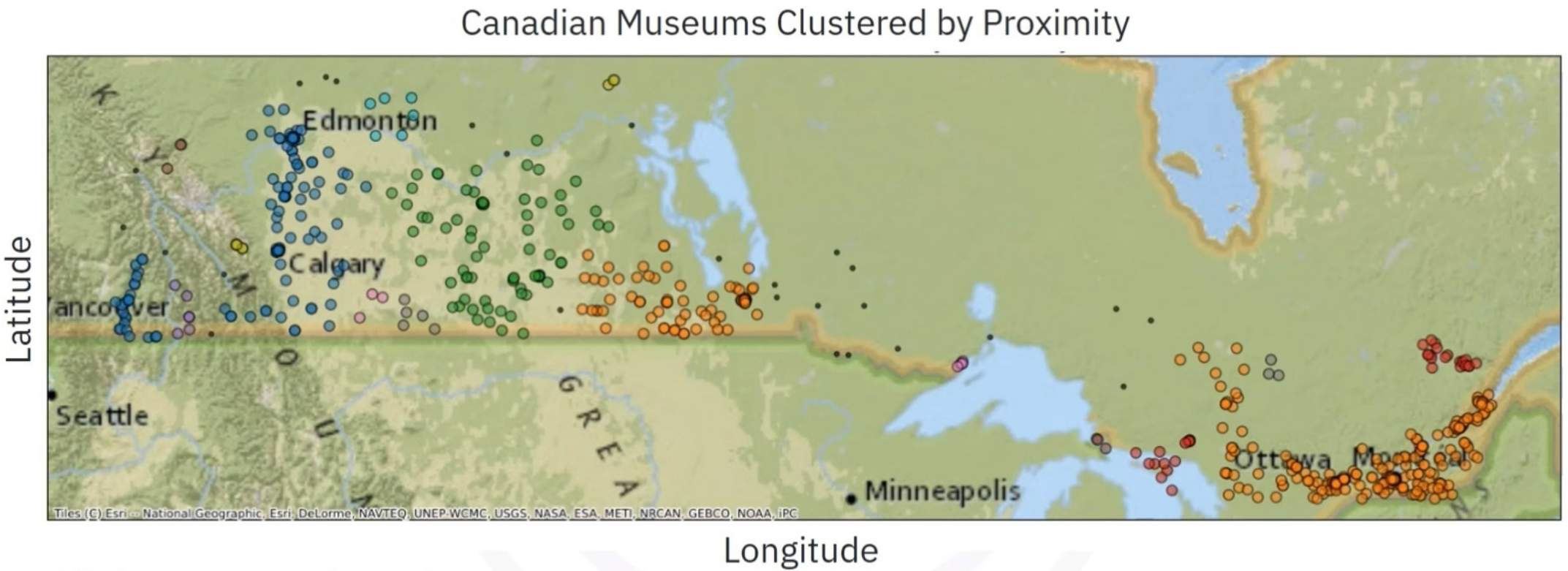
Combination of agglomerative and density-based clustering

Starts by identifying each point as its own cluster

Progressively agglomerates clusters into a hierarchy

Gets simplified into a condensed tree with stable clusters

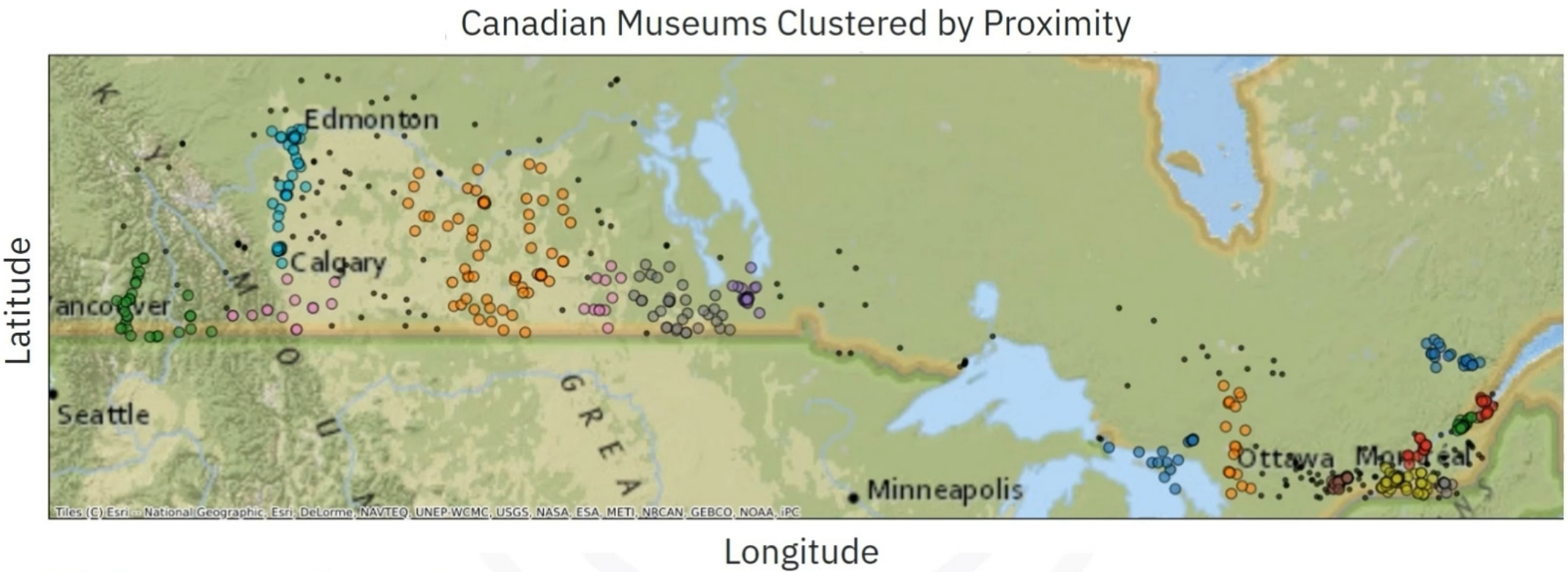
Real-world DBSCAN test



Minimum Samples = 3

Epsilon = 0.75

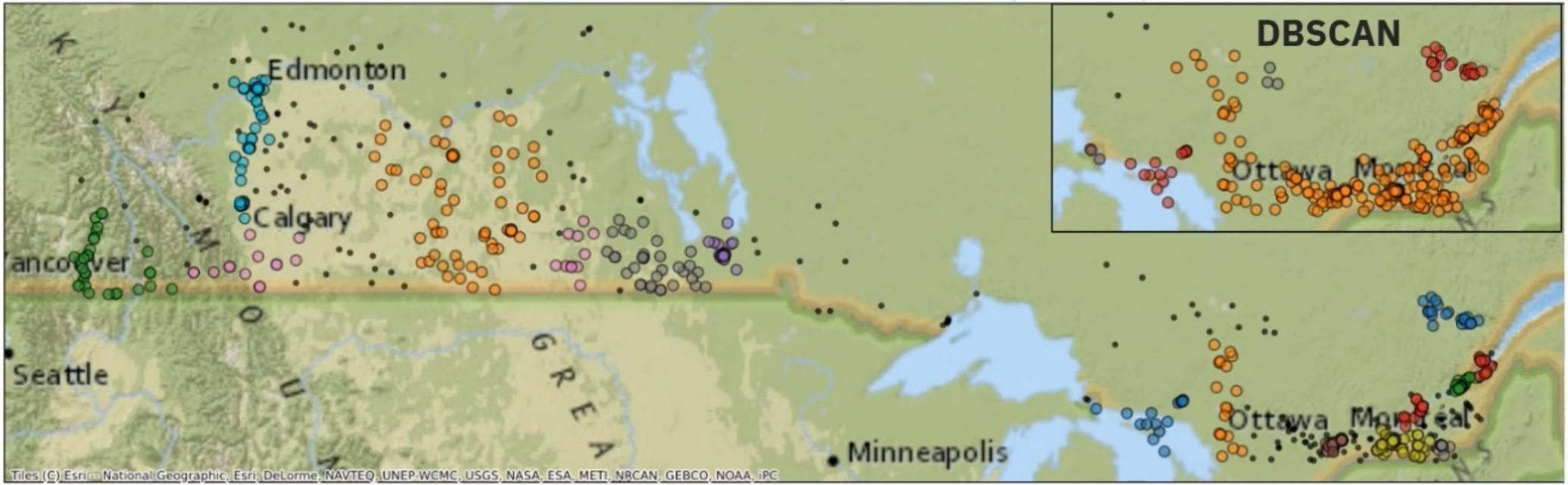
HDBSCAN adjusted test



Minimum samples = 10

Minimum size = 3

Canadian Museums Clustered by Proximity



Latitude

Longitude

Minimum samples = 10

Minimum size = 3

Recap

- DBSCAN creates clusters with a density value provided by the user
- Density-based clustering works by identifying regions of relatively high density
- DBSCAN is not iterative
- HDBSCAN doesn’t require any parameters to be set and uses cluster stability
- Cluster stability is the persistence of a cluster over a range of distance thresholds