Introduction to Unsupervised Learning

Understanding Patterns and Structures in Data

Kyle Territo

CHE 4230

Learning Objectives

- Define unsupervised learning and its importance
- Differentiate between supervised and unsupervised learning
- Explore key unsupervised learning techniques: clustering and dimensionality reduction
- Understand practical applications of unsupervised learning

Review:

- Open up the command terminal on your pc (not in VSCODE)

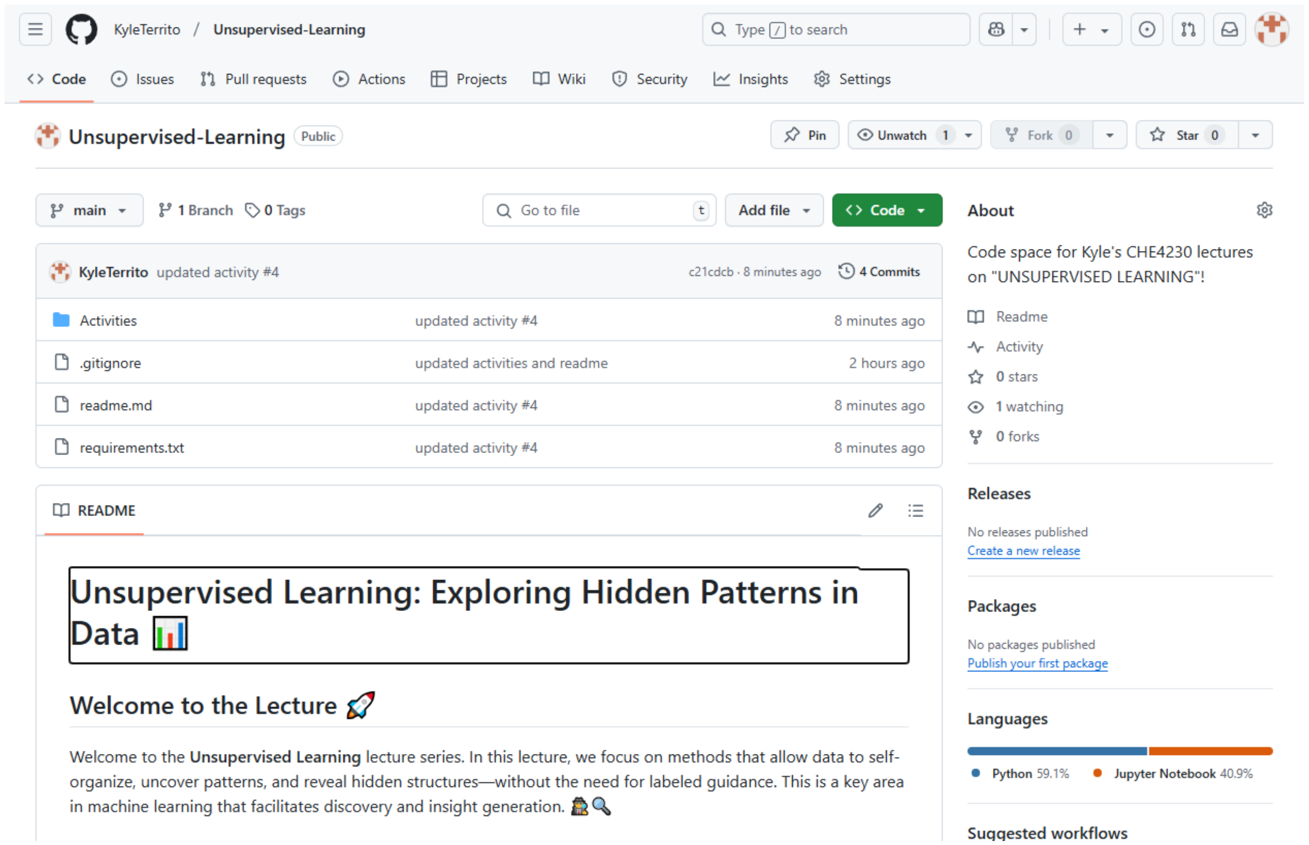
- Get comfortable with navigating to folders through the cmd line

  - cd <path>

➢ cd ..	*move up a level
➢ cd \	*go to root directory
➢ cls	*clear terminal
➢ Ctrl + C	*interrupt terminal

- Navigate to a folder where you store your repositories..

Clone the following repo here…



KyleTerrito / Unsupervised-Learning

Code · Issues · Pull requests · Actions · Projects · Wiki · Security · Insights · Settings

Unsupervised-Learning Public

Pin · Unwatch 1 · Fork 0 · Star 0

main · 1 Branch · 0 Tags · Go to file · Add file · Code

KyleTerrito updated activity #4 · c21cdcb · 8 minutes ago · 4 Commits

Activities	updated activity #4	8 minutes ago
.gitignore	updated activities and readme	2 hours ago
readme.md	updated activity #4	8 minutes ago
requirements.txt	updated activity #4	8 minutes ago

README

Unsupervised Learning: Exploring Hidden Patterns in Data 📊

Welcome to the Lecture 🚀

Welcome to the **Unsupervised Learning** lecture series. In this lecture, we focus on methods that allow data to self-organize, uncover patterns, and reveal hidden structures—without the need for labeled guidance. This is a key area in machine learning that facilitates discovery and insight generation. 🕵️🔍

About

Code space for Kyle's CHE4230 lectures on "UNSUPERVISED LEARNING"!

Readme · Activity · 0 stars · 1 watching · 0 forks

Releases: No releases published · Create a new release

Packages: No packages published · Publish your first package

Languages: ● Python 59.1% · ● Jupyter Notebook 40.9%

Suggested workflows

- We'll use this folder for our unsupervised learning lectures.
- https://github.com/KyleTerrito/Unsupervised-Learning.git

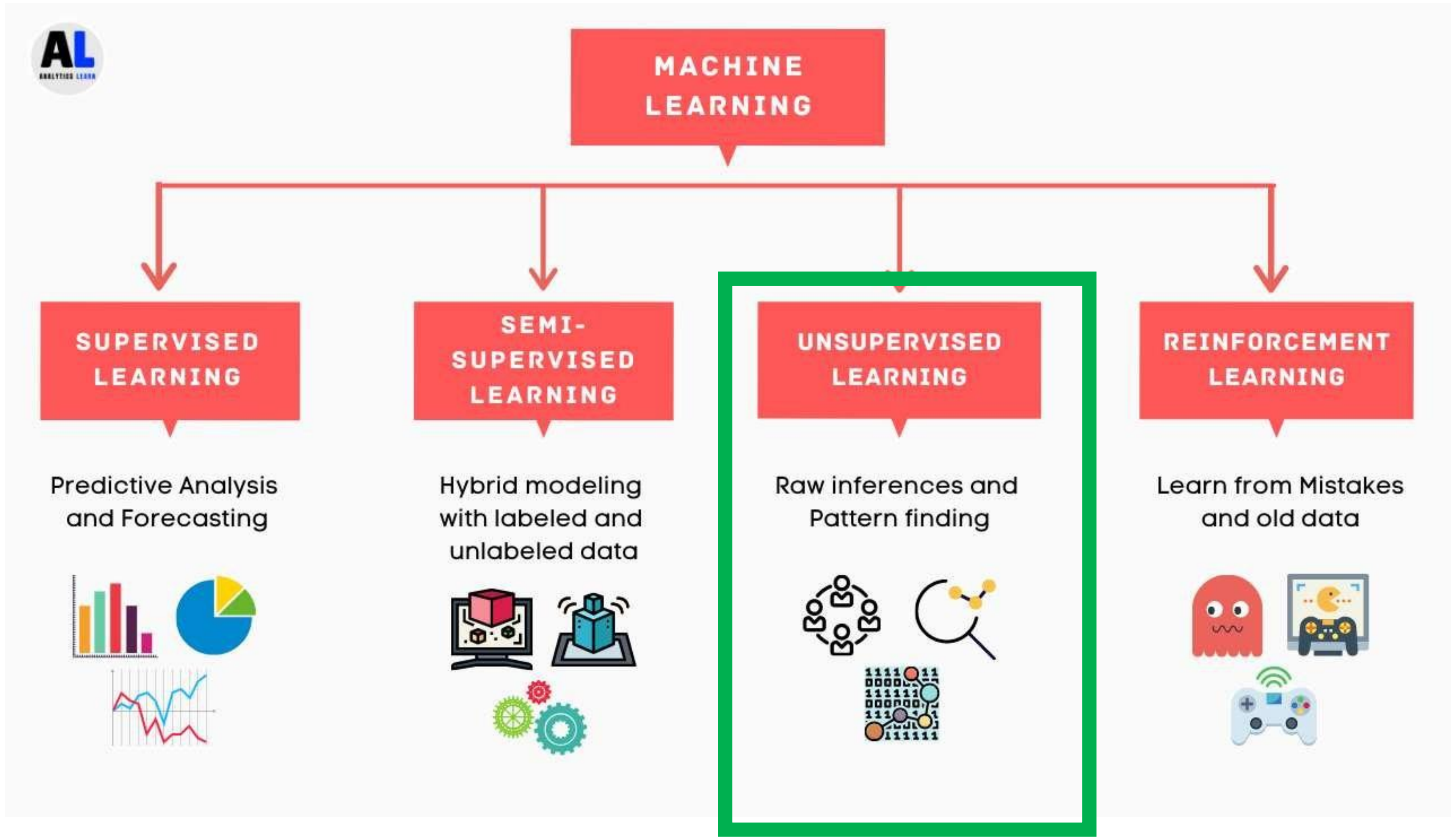
Activity #1 “Intro”:

- In many real-world situations, data does not come with clear labels or categories.
- How do we uncover hidden structures, patterns, or groupings in such datasets?
- Take a look at the provided dataset “Pyrolysis” (open in python or excel).
- What meaningful insights, trends, or groupings can you find in this data by inspecting it manually?

Activity (Discussion):

- How easy or hard was it to find meaningful insights?
  - Could you find patterns or structures in the data.
  - Discuss any challenges you faced, such as data complexity, volume, or lack of labels.
- What methods did you use to explore the data?
- What if the dataset had even more features or rows?
- What is there was a way to automate this exploration without manual effort?
- Unsupervised Learning!!

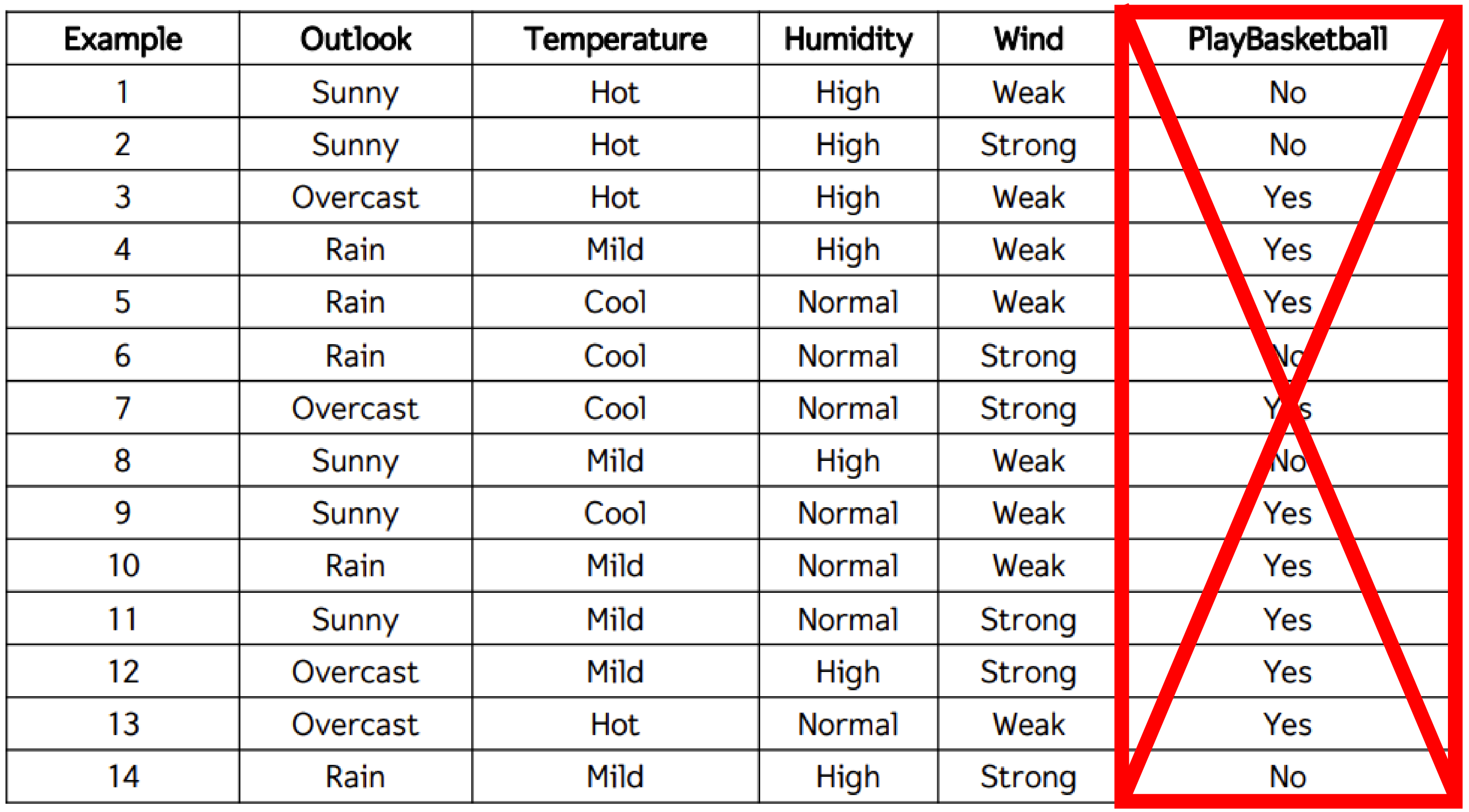
Machine Learning


MACHINE LEARNING
SUPERVISED LEARNING
Predictive Analysis and Forecasting
SEMI-SUPERVISED LEARNING
Hybrid modeling with labeled and unlabeled data
UNSUPERVISED LEARNING
Raw inferences and Pattern finding
REINFORCEMENT LEARNING
Learn from Mistakes and old data

What is unsupervised learning?

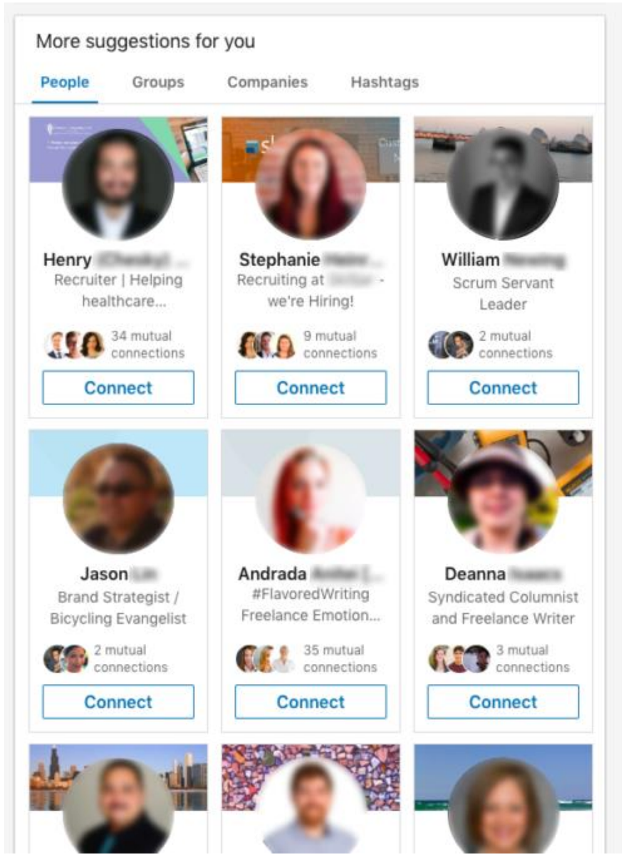
- Definition: a type of machine learning that identifies patterns and structures in data *without labelled outcomes*
- Key idea: the algorithm infers a function to describe hidden structures in data
- Comparison with Supervised Learning:
  - Supervised Learning: Input → Output (labels provided)
  - Unsupervised Learning: Input → Hidden Structure (no initial labels)

What are labels?



Example	Outlook	Temperature	Humidity	Wind	PlayBasketball
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Key Applications


More suggestions for you

People Groups Companies Hashtags

Henry [redacted]
Recruiter | Helping healthcare...
34 mutual connections
Connect

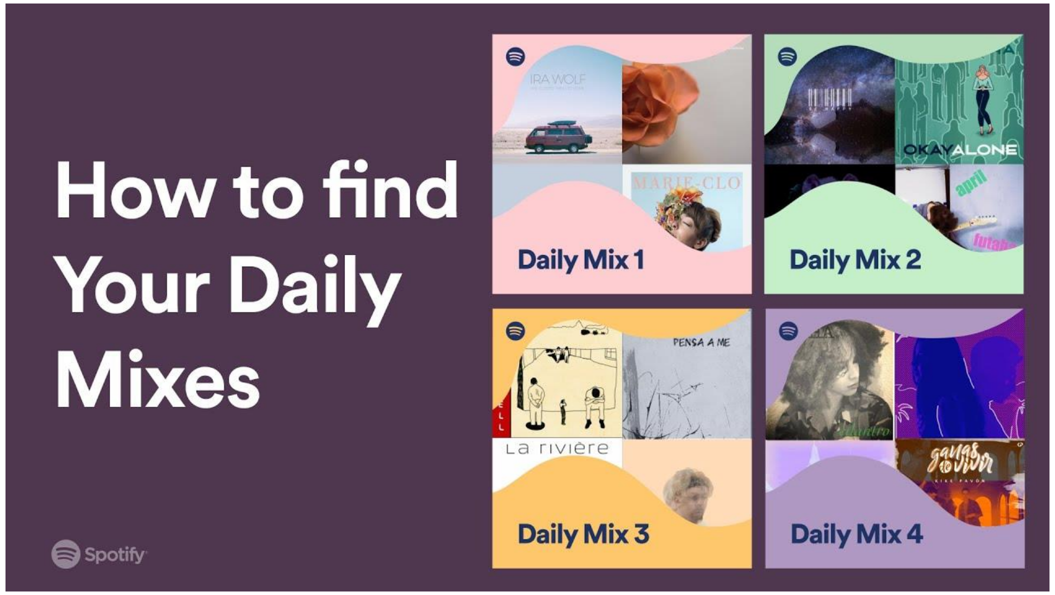
Stephanie [redacted]
Recruiting at [redacted] - we're Hiring!
9 mutual connections
Connect

William [redacted]
Scrum Servant Leader
2 mutual connections
Connect

Jason [redacted]
Brand Strategist / Bicycling Evangelist
2 mutual connections
Connect

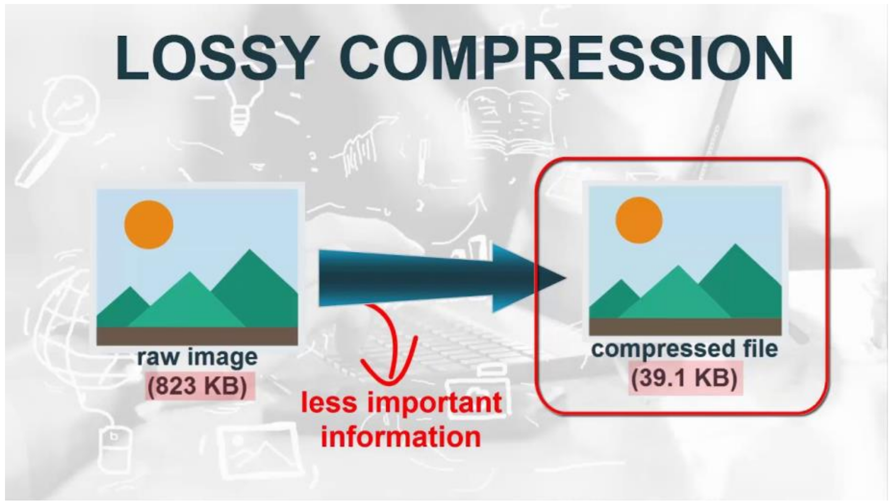
Andrada [redacted]
#FlavoredWriting Freelance Emotion...
35 mutual connections
Connect

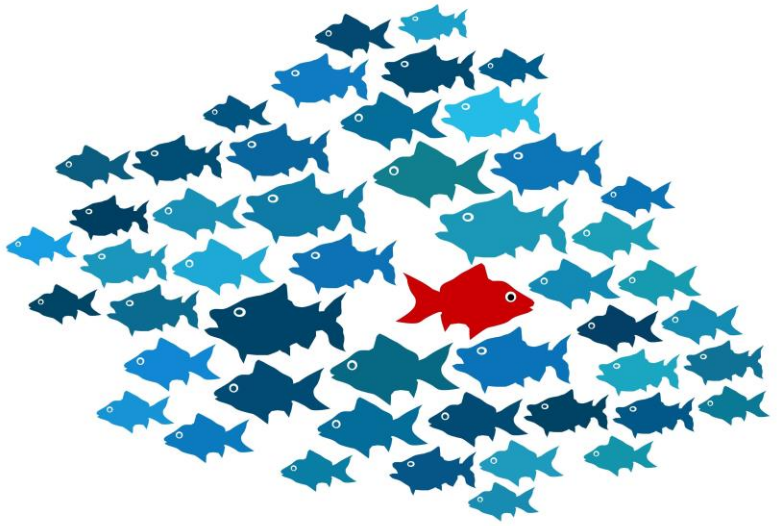
Deanna [redacted]
Syndicated Columnist and Freelance Writer
3 mutual connections
Connect



Key Applications

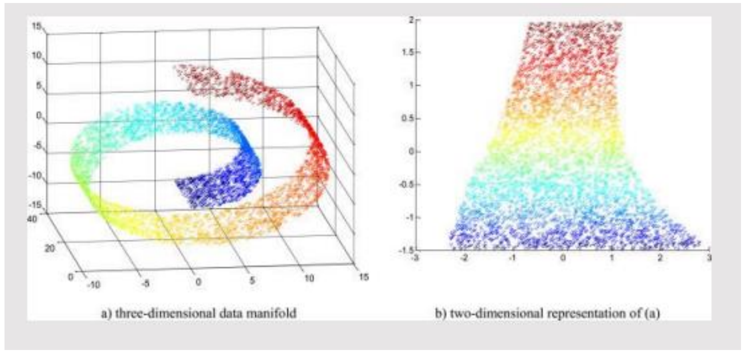
- Market segmentation
- Image/ video compression
- Anomaly detection
- Customer recommendation systems
- Speech and audio analysis



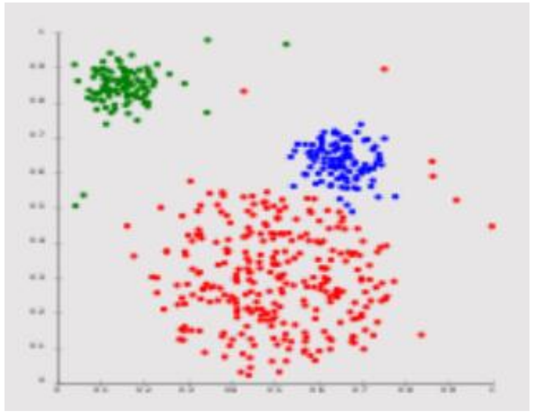


Types of Unsupervised Learning

- Dimensionality Reduction (DR)
  - Reducing the number of features while retaining essential information



- Clustering
  - Organizing data into groups based on similarity



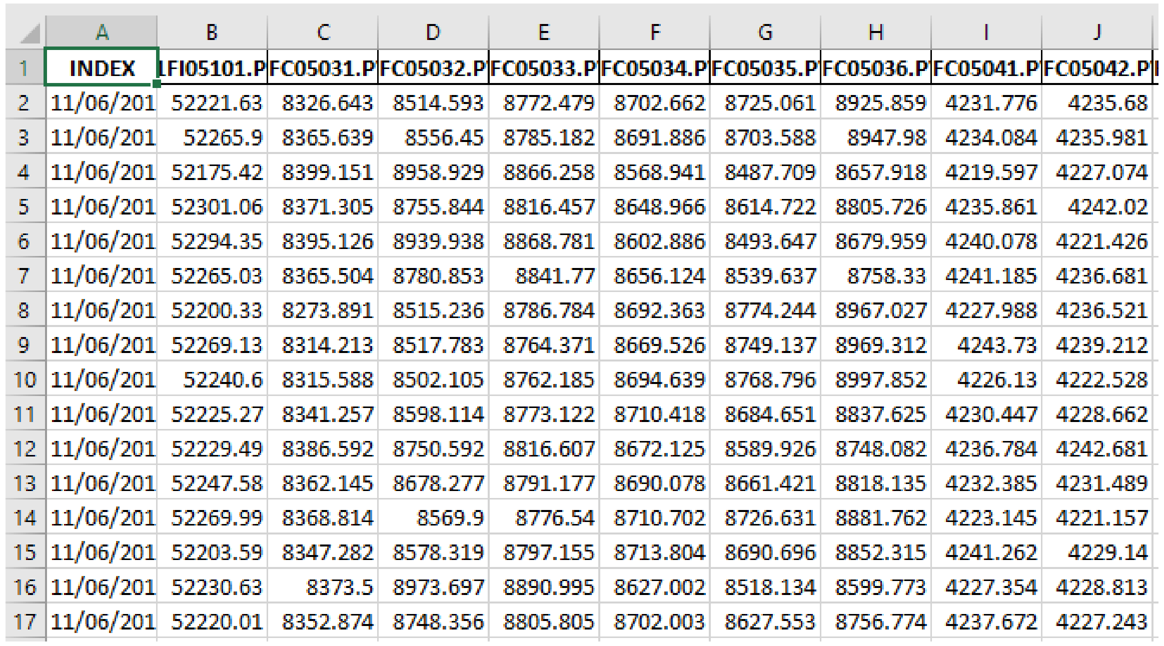
- Self-Organizing Maps (SOMs)
  - Maps high-dimensional data onto a low-dimensional grid while preserving topological relationships

Dimensionality Reduction

- Definition: Simplifying data by reducing its dimensions

- Popular techniques:
  - **PCA** (Principal Component Analysis)
  - **t-SNE** (t-Distributed Stochastic Neighbor Embedding)
  - **UMAP** (Uniform Manifold Approximation and Projection)
  - **PaCMAP** (Pairwise Controlled Manifold Approximation Projection)

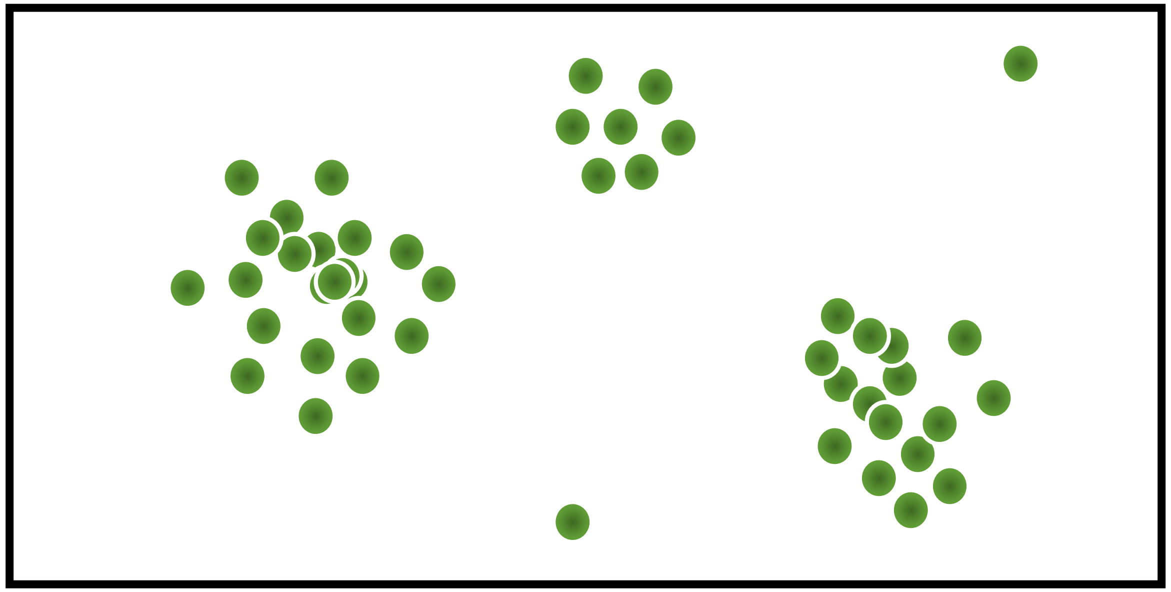
Goal of dimensionality reduction…



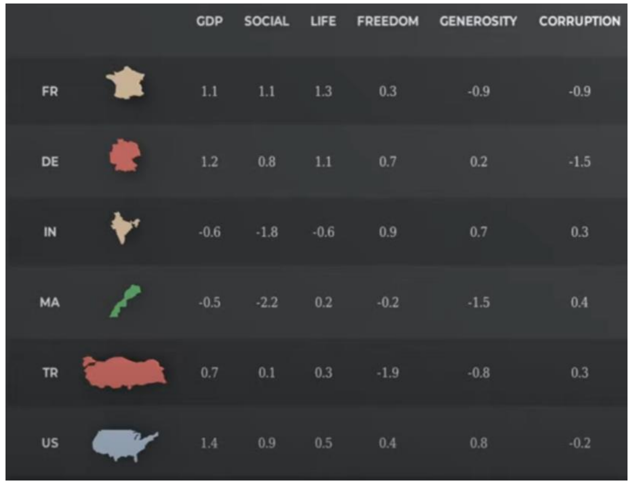
	A	B	C	D	E	F	G	H	I	J
1	INDEX	1FI05101.P	FC05031.P	FC05032.P	FC05033.P	FC05034.P	FC05035.P	FC05036.P	FC05041.P	FC05042.P
2	11/06/201	52221.63	8326.643	8514.593	8772.479	8702.662	8725.061	8925.859	4231.776	4235.68
3	11/06/201	52265.9	8365.639	8556.45	8785.182	8691.886	8703.588	8947.98	4234.084	4235.981
4	11/06/201	52175.42	8399.151	8958.929	8866.258	8568.941	8487.709	8657.918	4219.597	4227.074
5	11/06/201	52301.06	8371.305	8755.844	8816.457	8648.966	8614.722	8805.726	4235.861	4242.02
6	11/06/201	52294.35	8395.126	8939.938	8868.781	8602.886	8493.647	8679.959	4240.078	4221.426
7	11/06/201	52265.03	8365.504	8780.853	8841.77	8656.124	8539.637	8758.33	4241.185	4236.681
8	11/06/201	52200.33	8273.891	8515.236	8786.784	8692.363	8774.244	8967.027	4227.988	4236.521
9	11/06/201	52269.13	8314.213	8517.783	8764.371	8669.526	8749.137	8969.312	4243.73	4239.212
10	11/06/201	52240.6	8315.588	8502.105	8762.185	8694.639	8768.796	8997.852	4226.13	4222.528
11	11/06/201	52225.27	8341.257	8598.114	8773.122	8710.418	8684.651	8837.625	4230.447	4228.662
12	11/06/201	52229.49	8386.592	8750.592	8816.607	8672.125	8589.926	8748.082	4236.784	4242.681
13	11/06/201	52247.58	8362.145	8678.277	8791.177	8690.078	8661.421	8818.135	4232.385	4231.489
14	11/06/201	52269.99	8368.814	8569.9	8776.54	8710.702	8726.631	8881.762	4223.145	4221.157
15	11/06/201	52203.59	8347.282	8578.319	8797.155	8713.804	8690.696	8852.315	4241.262	4229.14
16	11/06/201	52230.63	8373.5	8973.697	8890.995	8627.002	8518.134	8599.773	4227.354	4228.813
17	11/06/201	52220.01	8352.874	8748.356	8805.805	8702.003	8627.553	8756.774	4237.672	4227.243

27 features…

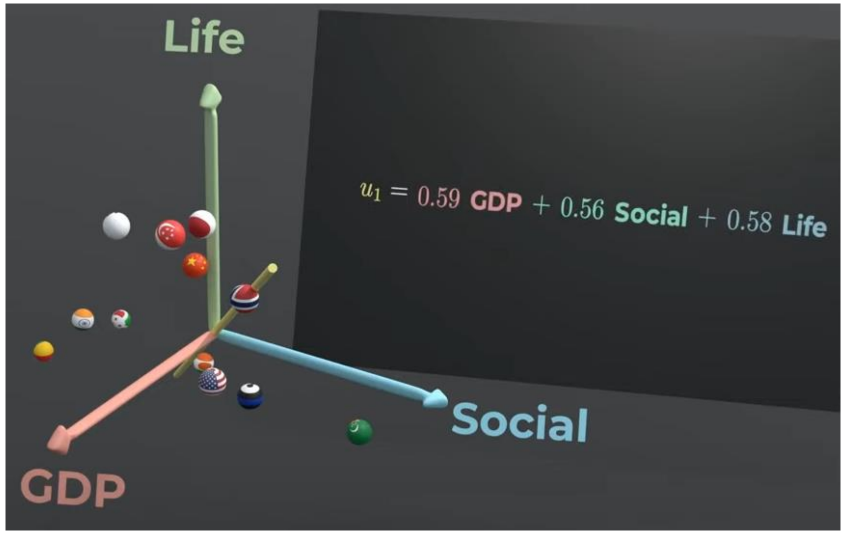
2 features ?

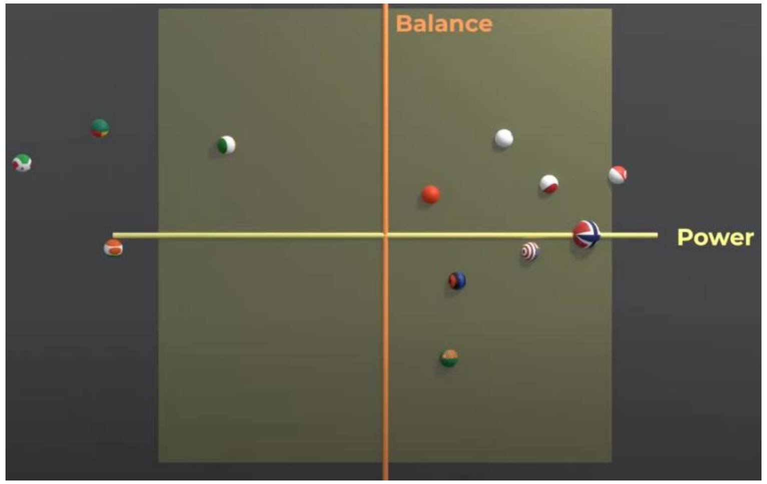


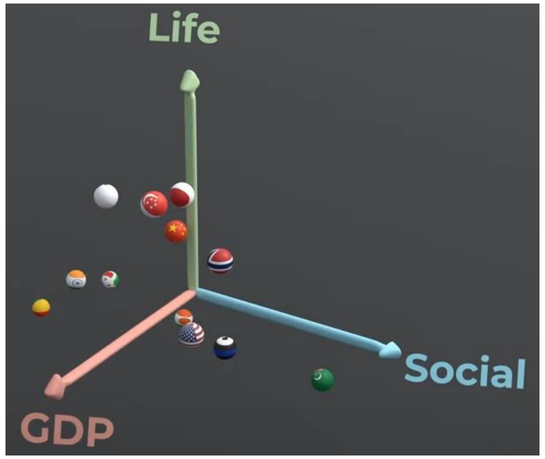
Dimensionality Reduction

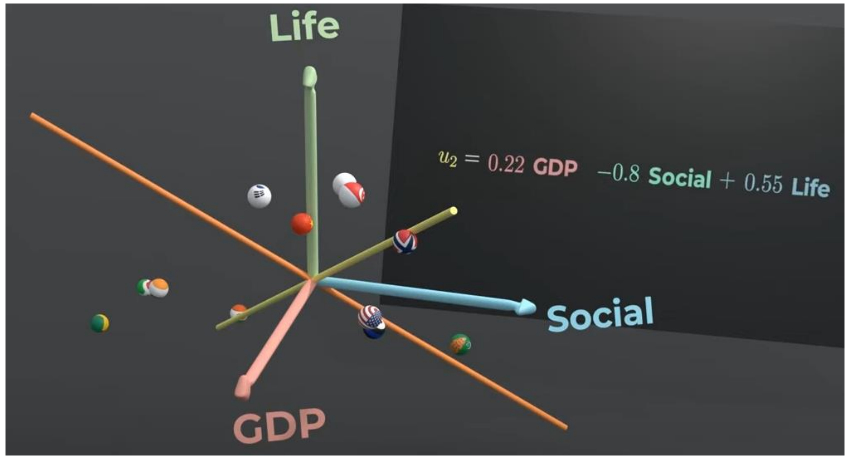


		GDP	SOCIAL	LIFE	FREEDOM	GENEROSITY	CORRUPTION
FR		1.1	1.1	1.3	0.3	-0.9	-0.9
DE		1.2	0.8	1.1	0.7	0.2	-1.5
IN		-0.6	-1.8	-0.6	0.9	0.7	0.3
MA		-0.5	-2.2	0.2	-0.2	-1.5	0.4
TR		0.7	0.1	0.3	-1.9	-0.8	0.3
US		1.4	0.9	0.5	0.4	0.8	-0.2









[Principal Component Analysis (PCA) - YouTube]

Principle Component Analysis (PCA)

- Objective: Transform data into a smaller set of uncorrelated components

- Steps:
  1. Compute the mean-centered data
  2. Compute the covariance matrix
  3. Calculate eigenvectors and eigenvalues
  4. Project data onto top eigen vectors

$$X_c = X - \bar{X}$$

$$\Sigma = \frac{1}{n-1} X_c^T X_c$$

$$\Sigma v = \lambda v$$

- Pros: Effective for large datasets
- Cons: Linear technique, may lose interpretability

$$Solve: \det(\Sigma - \lambda I)$$

t-SNE

- **t**-Distributed **S**tochastic **N**eighbor **E**mbedding

- How it works:

    1. Converts pairwise similarities into probabilities in high dimensions
    2. Maps these probabilities to a low-dimensional space while preserving relative similarities
    3. Optimizes positions to minimize divergence between high and low dimensional distributions

- Pros: excellent for capturing local structure and visualizing clusters
- Cons: computationally intensive and non-deterministic (random)

UMAP

- **U**niform **M**anifold **A**pproximation and **P**rojection

- How it works:
  1. Constructs a weighted graph of nearest neighbors in high dimensions
  2. Optimizes the graph layout in lower dimensions

- Pros: Faster than t-SNE, preserves more global structure
- Cons: Results sensitive to parameter tuning

PaCMAP

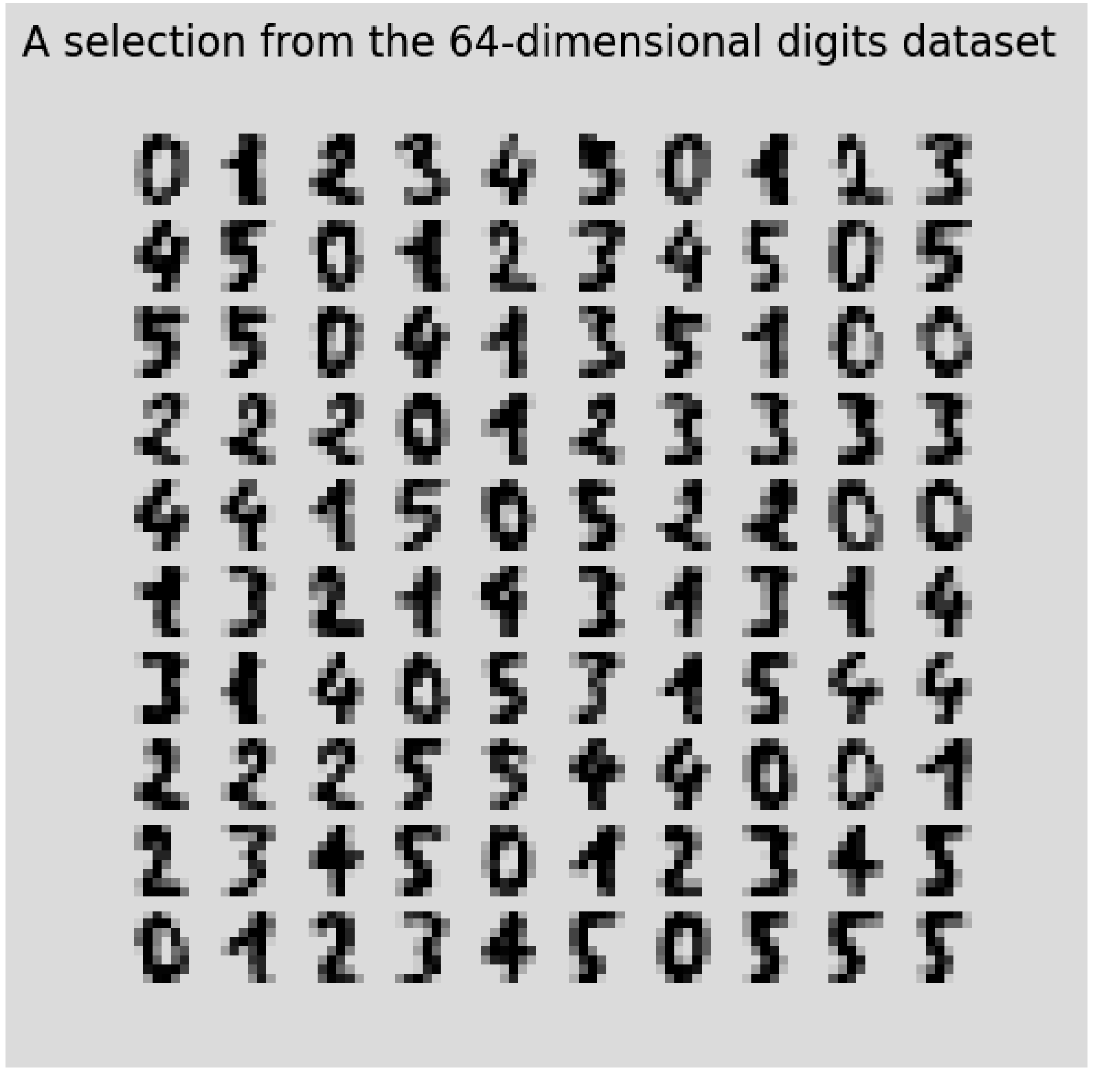
- **Pa**irwise **C**ontrolled **M**anifold **A**pproximation and **P**rojection

- How it works:
  1. Focusses on three types of point pairs: neighbors (local), mid-range, and distant points
  2. Optimizes a weighted combination of pairwise distances

- Pros: Balances local and global structure effectively
- Cons: sensitivity to hyperparameters, computational cost

Activity #2 “Digits Data”

- Go to **Activity 2** in the folder. Here there is a code for DR methods.
- This example uses a dataset called “Digits”
- It contains 1,797 images of handwritten digits (8x8 pixels) flattened into 64 features



BUT, the data structure is preserved differently…

- Types of data preservation:

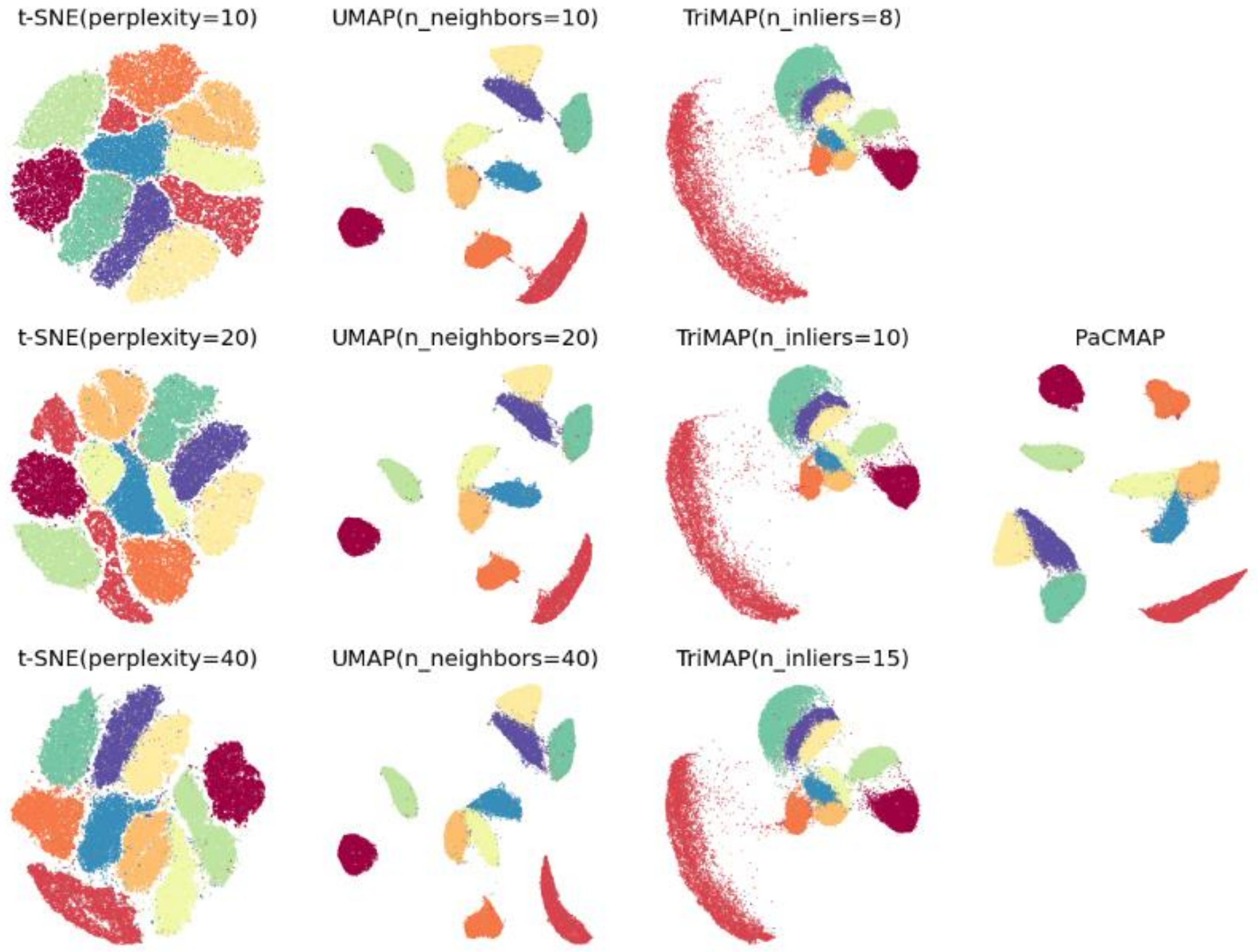
  1. Global Structure
  2. Local Structure
  3. Local-Global Structure

Type	Description	Example
LOCAL	long-distance relationships	PCA
GLOBAL	local neighborhoods	t-SNE
LOCAL-GLOBAL	balance between the two	UMAP, PaCMAP

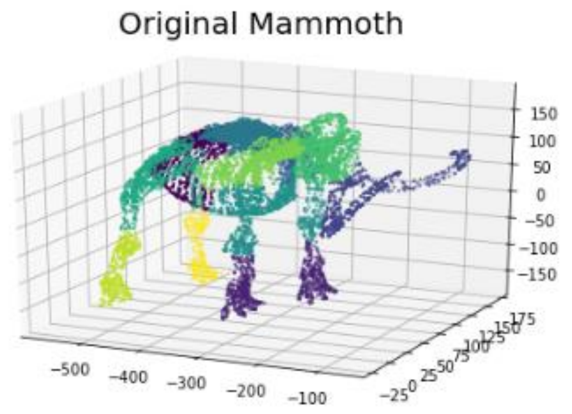
- **Global structure** focuses on retaining the **overall geometric relationships** and variance patterns across the entire dataset.

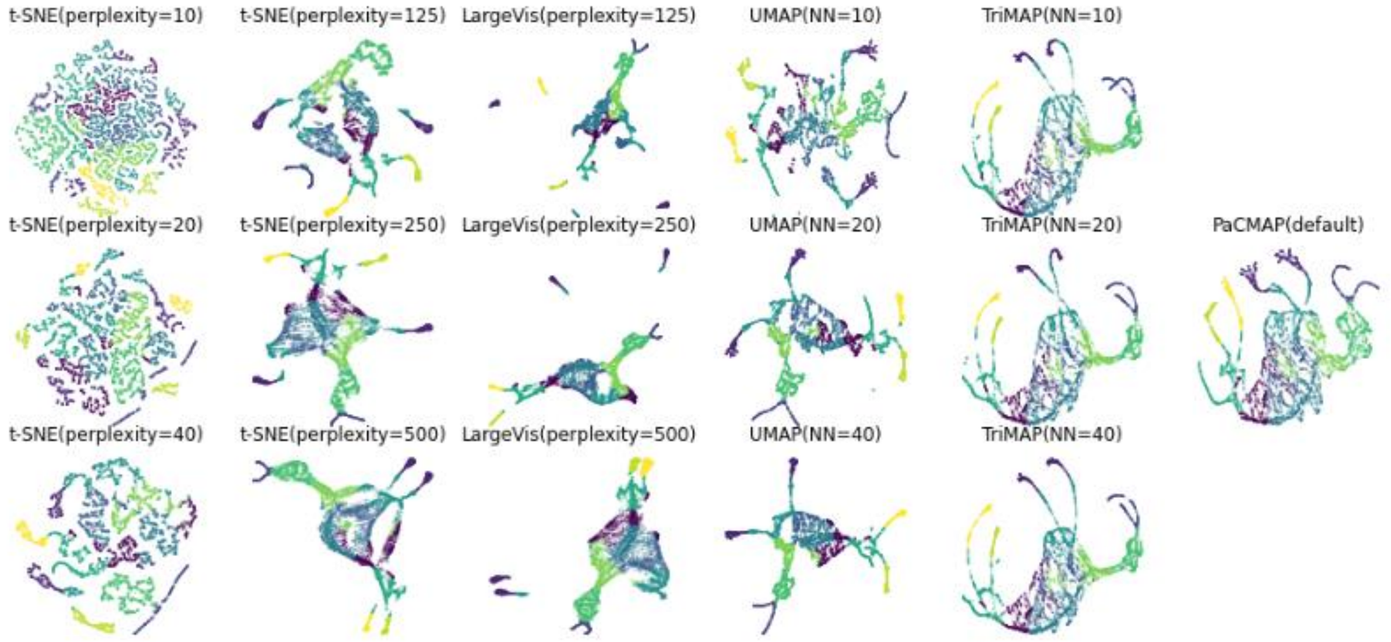
- **Local structure** maintains the **proximity and relationships of nearby points** in the reduced dimensional space

Benchmarks (MNIST)



Benchmarks (Mammoth)





Clustering

- Definition: Grouping similar data points together

- Types:
  - Partition based
  - Hierarchical based
  - Density based

- Popular algorithms:
  - K-Means (partition based)
  - Hierarchical Clustering (hierarchical based)
  - DBSCAN (density based)

K-Means

- Definition: Grouping similar data points together using centroids

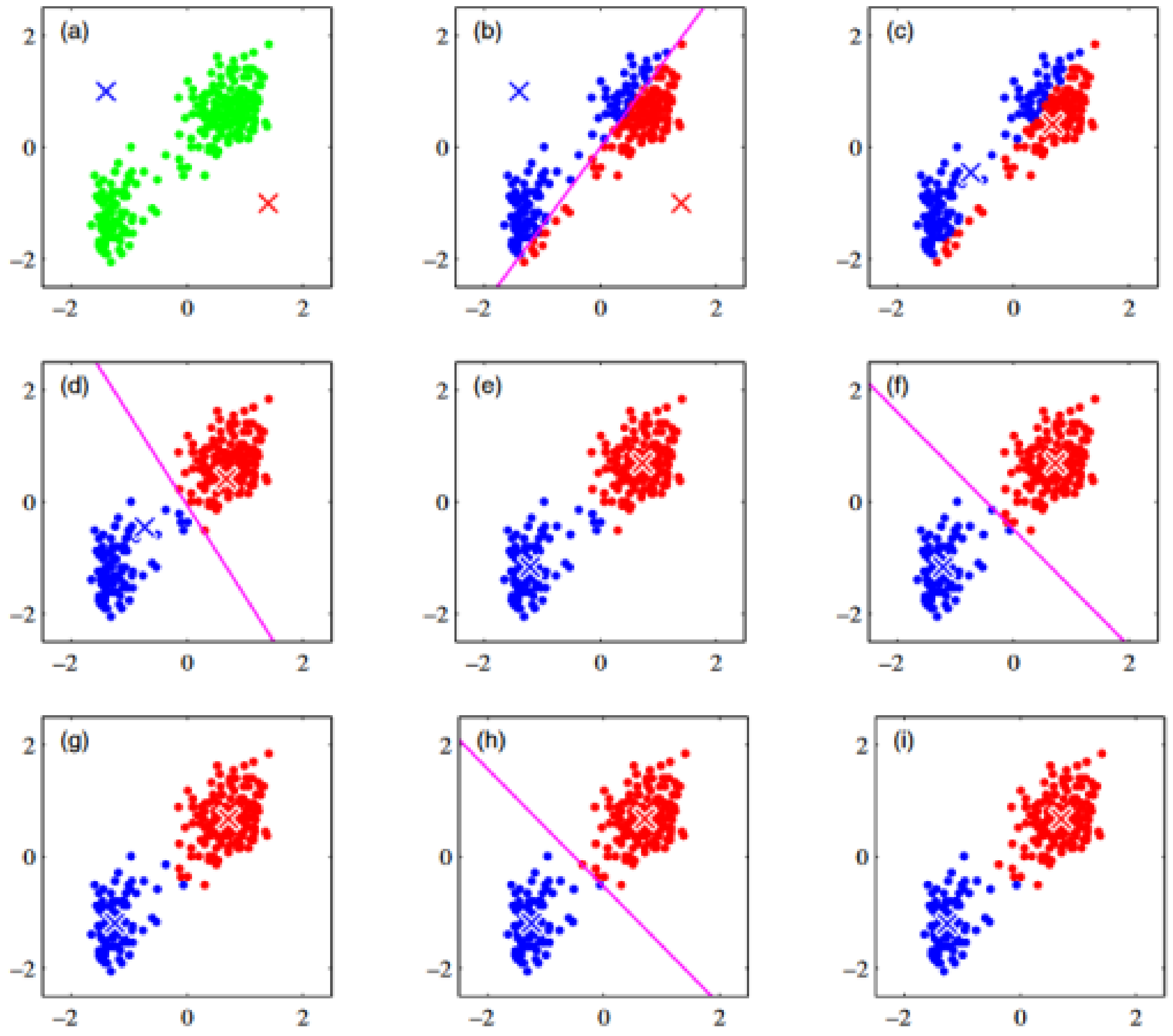
- How it works:
  - Initialization- select k random points as initial centroids
  - Assignment step- assign each data point to the nearest centroid (based on Euclidean distance)
  - Update step- calculate the new centroid for each cluster (mean of all points in the cluster)
  - Repeat- iterate between assignment and update steps until centroids stabilize

- Pros: simple, efficient
- Cons: sensitive to initialization, have to select k number of clusters

K-Means

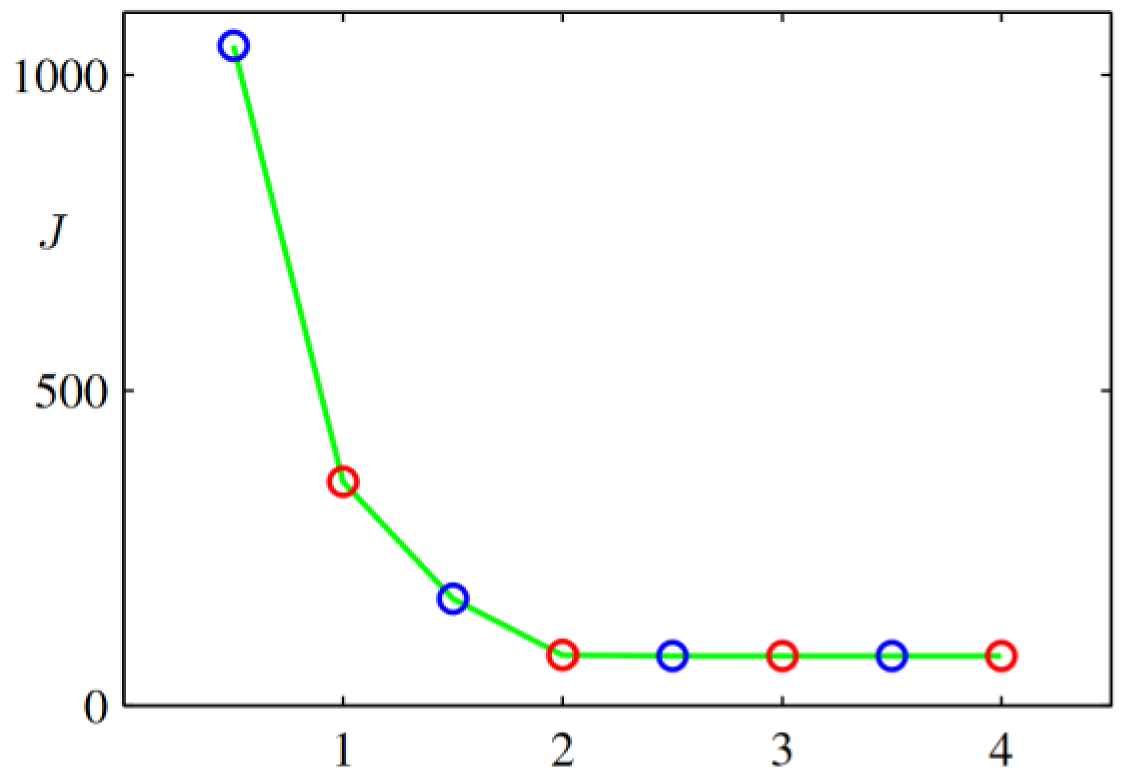
- After each iteration, the centroids move closer to the center of the expected clusters.

- Points are reassigned each step until convergence.

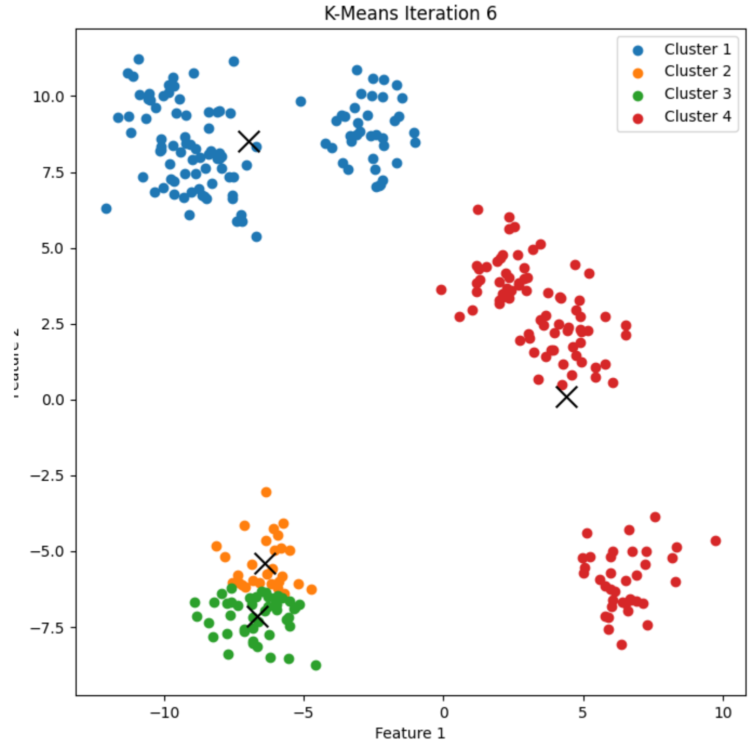


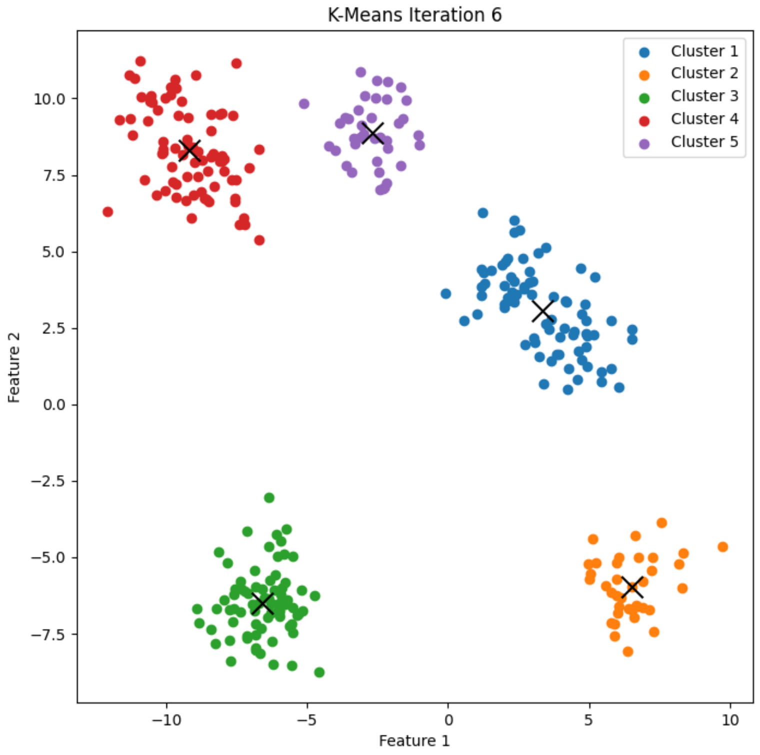
K-Means

- Convergence is governed by an error function that maximizes the distance between clusters and minimizes distance of points within clusters.



Activity #3 “K-Means Demo”





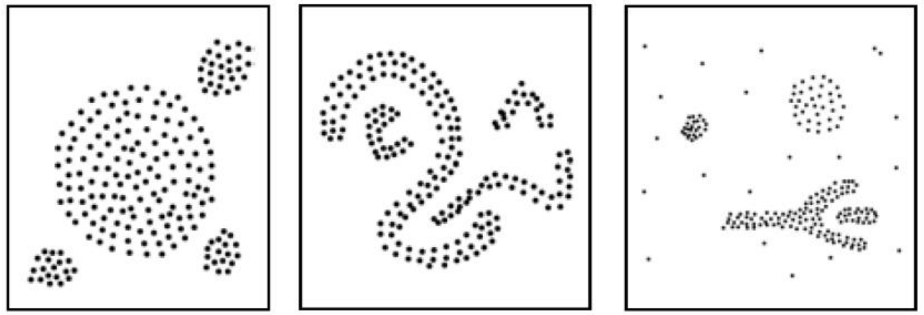
DBSCAN

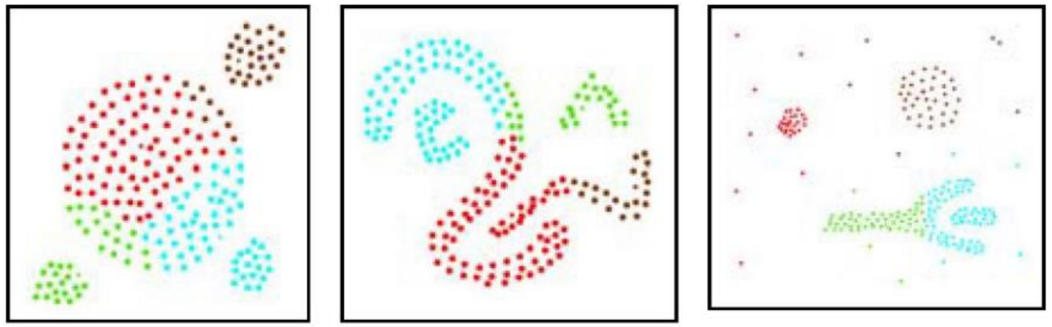
- **D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise

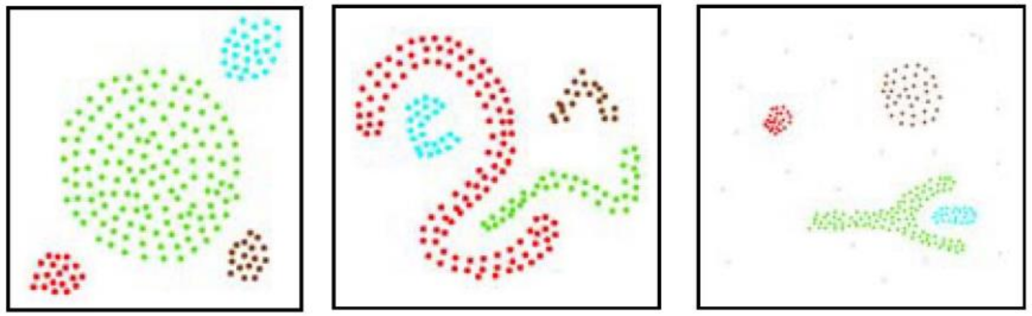
- How it works:
  1. Define a neighborhood radius (ε) and a minimum number of points (β) required to form a dense region
  2. Start with an unvisited point:
     - If it has at least β neighbors within ε, it becomes a core point and starts forming a cluster
     - Expand the cluster by recursively adding points that are unreachable from core points
     - Points that do not meet the density criteria are marked as noise
  3. Clusters grow based on density-connectivity

- Pros:
  - Can detect clusters of arbitrary shapes,
  - Handles noise and outliers
  - No need to specify number of clusters
- Cons:
  - Sensitive to ε and β
  - Struggles with clusters of varying densities
  - Computationally expensive

Centroid-based vs Density-based





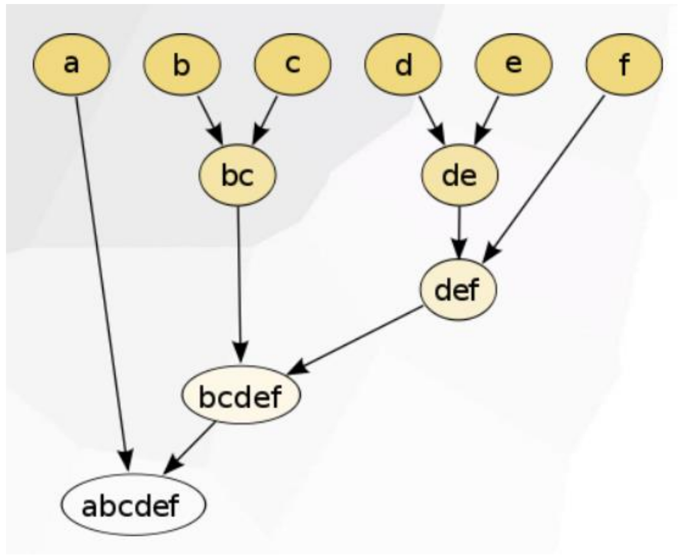


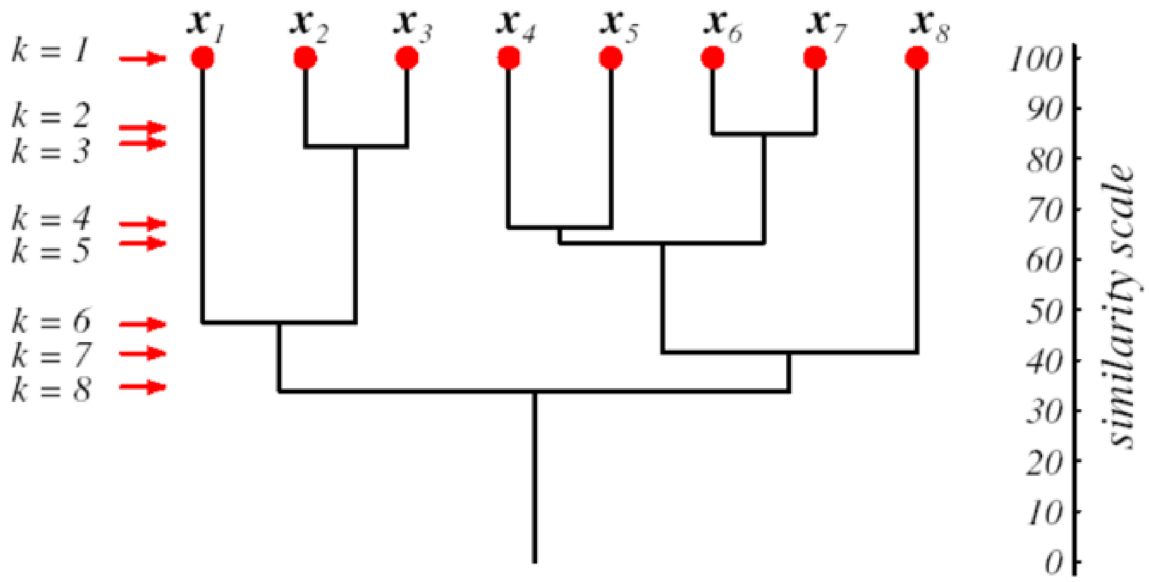
Hierarchical Clustering

- Types:
    1. Agglomerative (bottom-up)
    2. Divisive (top-down)

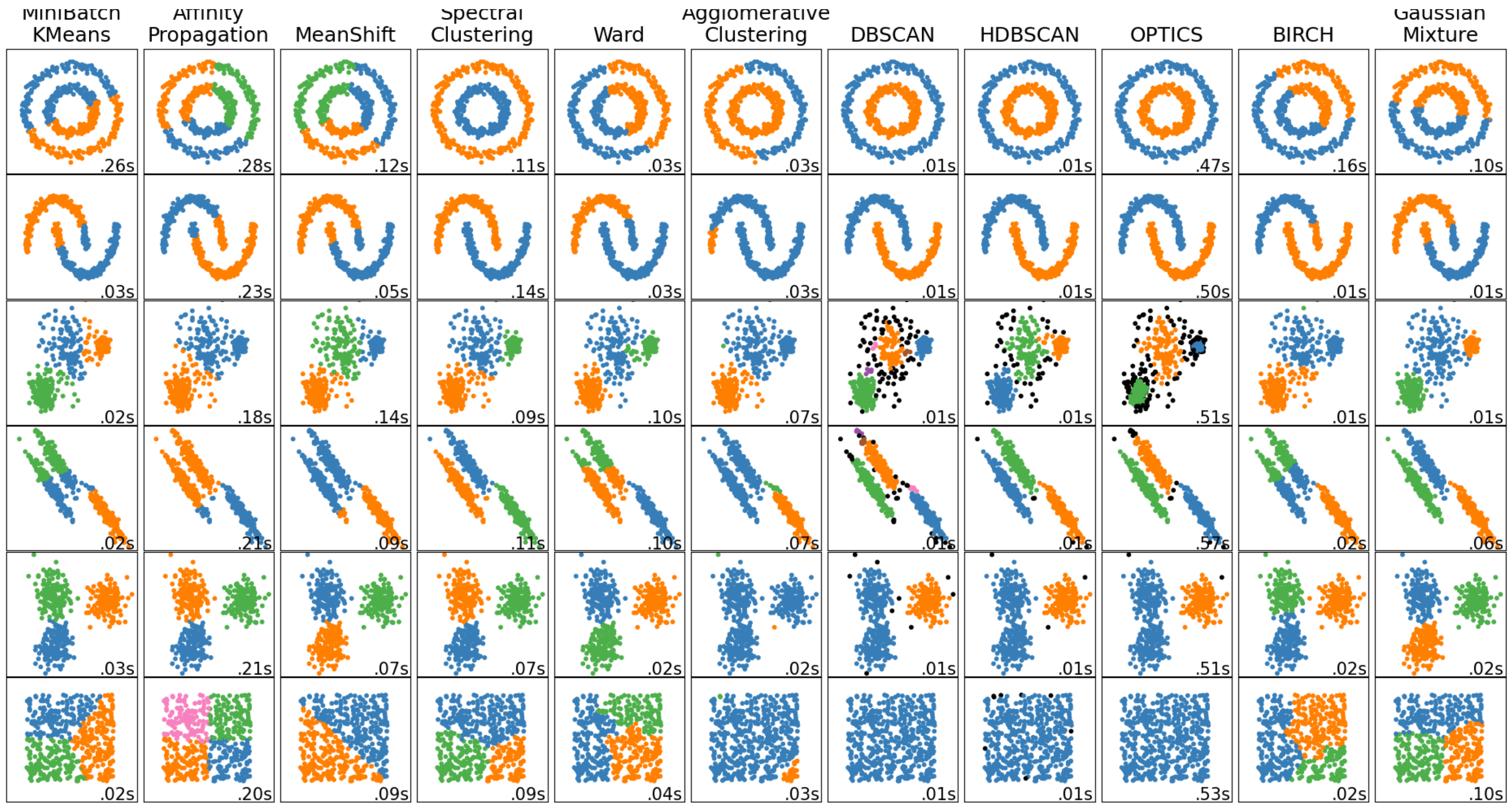
- Key idea: Build a tree (dendrogram) based on associations between group members. Determine number of clusters based on group associations.

- Pros: Does not require specifying k
- Cons: Computationally expensive





Clustering Examples

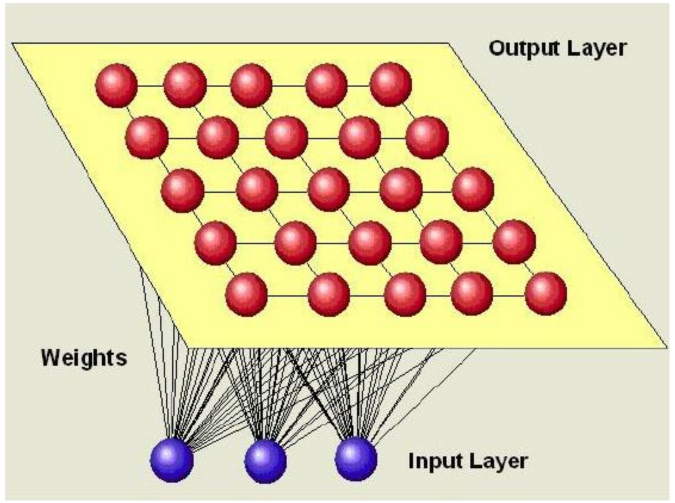


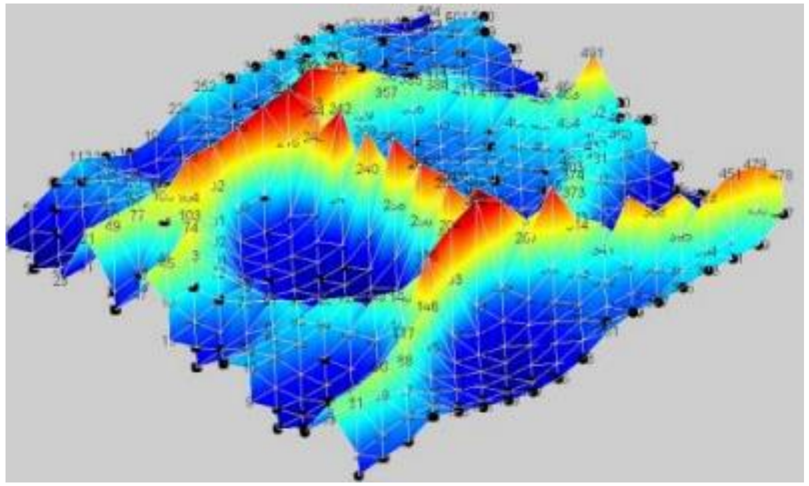
Self-Organizing Maps (SOMs)

•**What It Is**: SOMs are a type of unsupervised neural network that organize and simplify complex, high-dimensional data into a 2D grid, where similar data points are placed closer together.

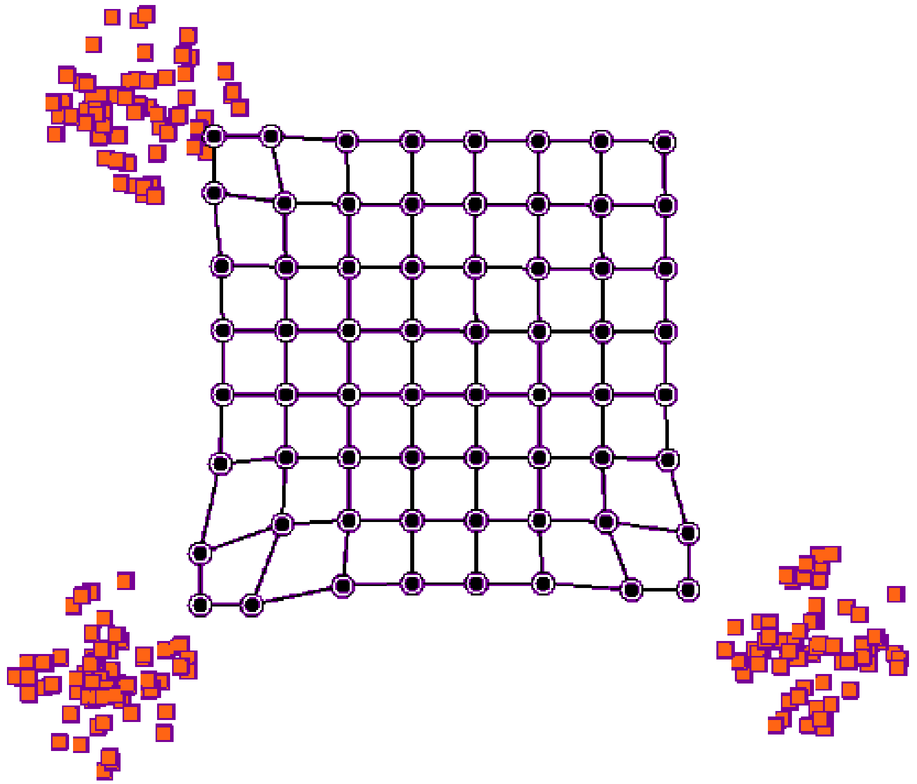
•**How It Works**: SOMs identify the grid point (node) most similar to a data point, called the Best Matching Unit (BMU). Then, the BMU and its neighbors are adjusted to resemble the data point more closely, gradually shaping the grid to reflect the data's structure.

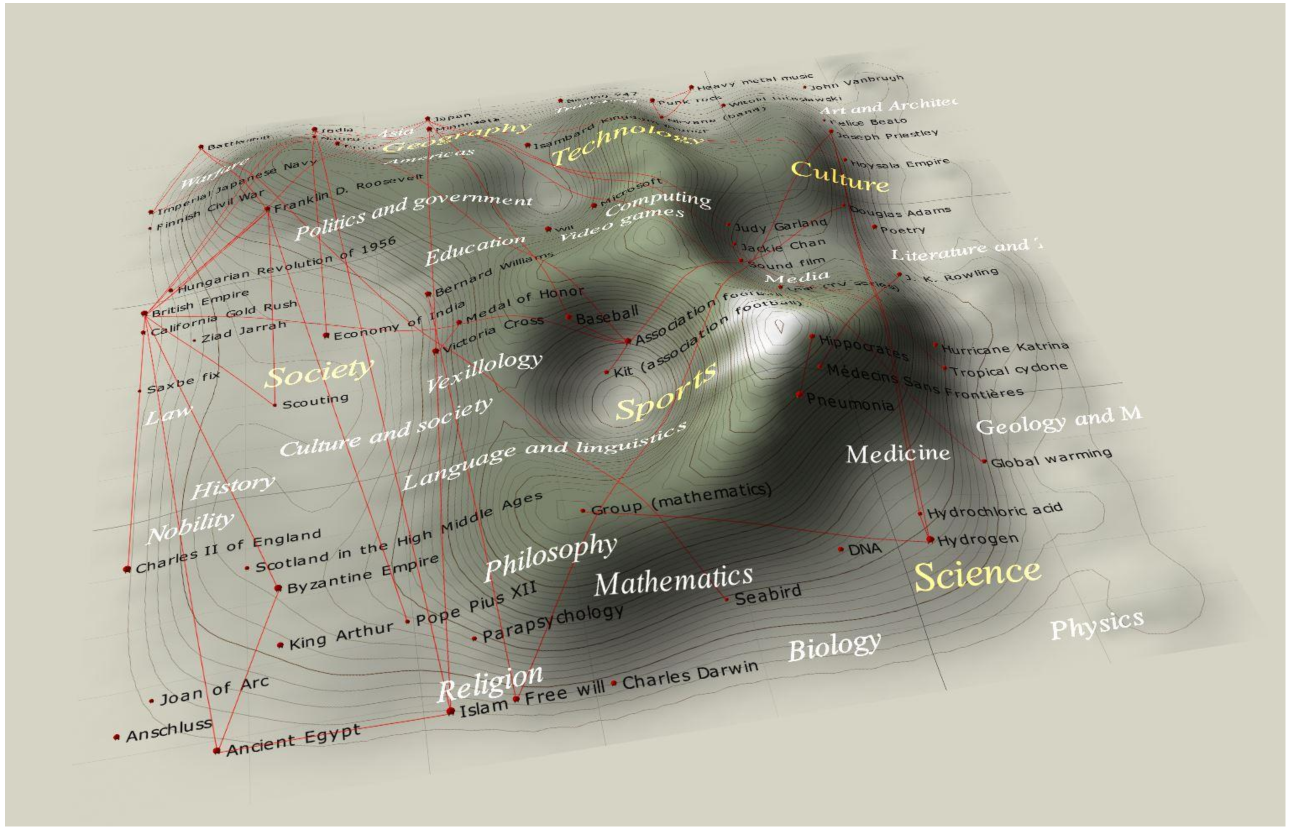
•**Why It's Useful**: SOMs preserve topological relationships, meaning similar data points are mapped near each other on the grid. This makes it easy to visualize patterns, clusters, and relationships in data while retaining the original structure and similarities.





SOMs



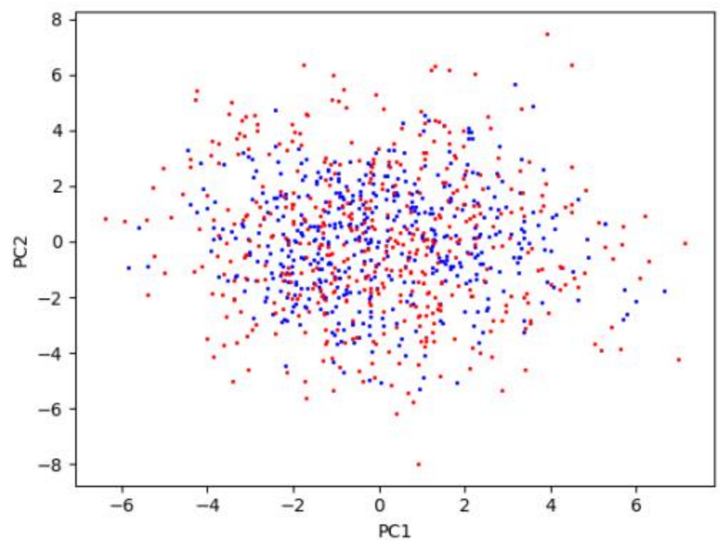


Warfare
Geography
Technology
Culture
Politics and government
Computing
Video games
Education
Literature and
Media
Society
Vexillology
Sports
Law
Culture and society
Language and linguistics
Medicine
Geology and M
History
Nobility
Philosophy
Mathematics
Science
Religion
Biology
Physics
Franklin D. Roosevelt
Hungarian Revolution of 1956
British Empire
California Gold Rush
Ziad Jarrah
Economy of India
Medal of Honor
Victoria Cross
Baseball
Scouting
Hippocrates
Hurricane Katrina
Tropical cyclone
Médecins Sans Frontières
Pneumonia
Global warming
Group (mathematics)
Hydrochloric acid
Hydrogen
DNA
Seabird
Charles II of England
Scotland in the High Middle Ages
Byzantine Empire
King Arthur
Pope Pius XII
Parapsychology
Joan of Arc
Anschluss
Ancient Egypt
Islam
Free will
Charles Darwin
Judy Garland
Jackie Chan
Sound film
Douglas Adams
Poetry
J. K. Rowling
Saxbe fix
Finnish Civil War
Imperial Japanese Navy

Challenges in Unsupervised Learning

- Lack of labeled data for validation
  - How can the results be validated?
- Choice of hyperparameters
- Interpretability of results
- Computational complexity

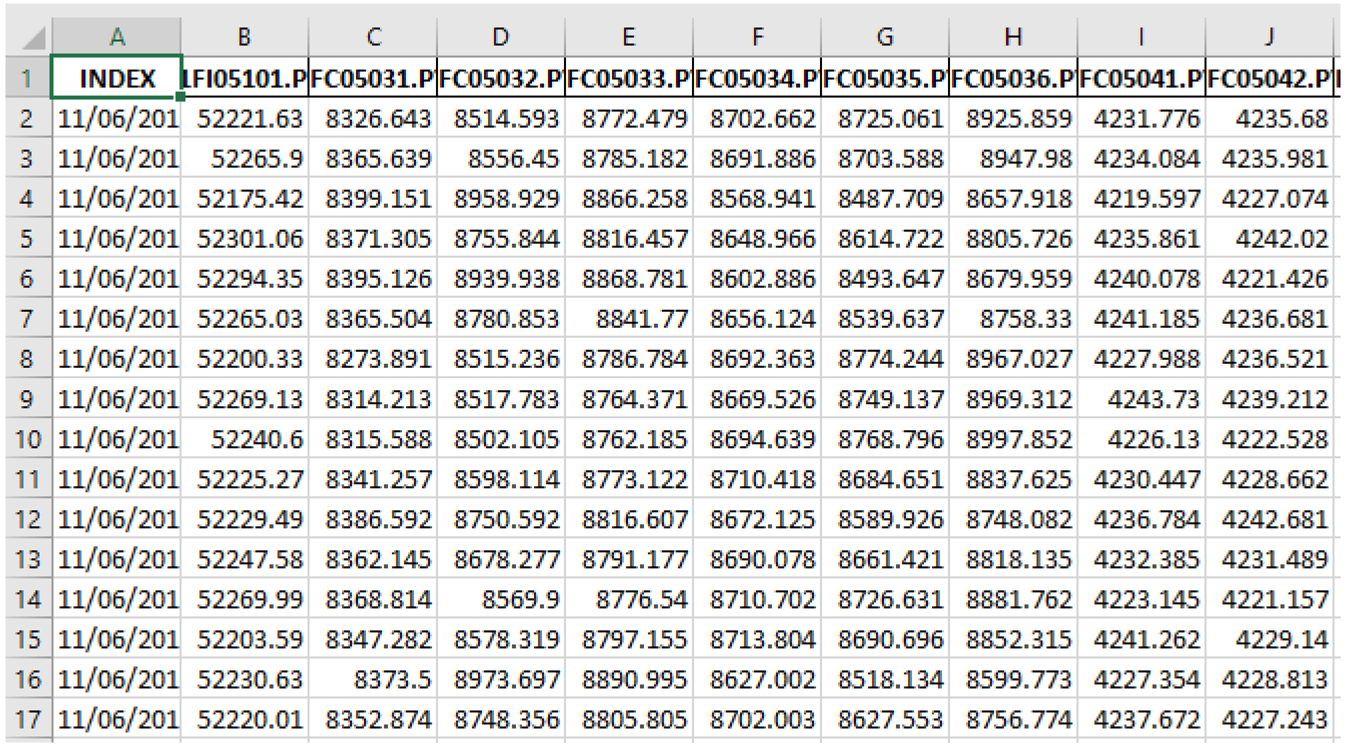
Does this mean anything?



Practical Application

- **<u>Fault Detection and Process Monitoring</u>**
  - DR and clustering techniques can effectively analyze chemical plant data, which is typically large, multivariate time series data.
  - DR reduces the features in the data such that essential patterns and features become more apparent– making it easier to analyze complex relationships.
  - Clustering helps uncover distinct operational states such as:
    - Normal operating conditions
    - Downtimes
    - Faults or anomalies
  - These insights can be used to monitor live plant data, enabling real-time fault detection, process optimization, and enhanced decision making.

Pyrolysis Reactor



	A	B	C	D	E	F	G	H	I	J
1	INDEX	LFI05101.P	FC05031.P	FC05032.P	FC05033.P	FC05034.P	FC05035.P	FC05036.P	FC05041.P	FC05042.P
2	11/06/201	52221.63	8326.643	8514.593	8772.479	8702.662	8725.061	8925.859	4231.776	4235.68
3	11/06/201	52265.9	8365.639	8556.45	8785.182	8691.886	8703.588	8947.98	4234.084	4235.981
4	11/06/201	52175.42	8399.151	8958.929	8866.258	8568.941	8487.709	8657.918	4219.597	4227.074
5	11/06/201	52301.06	8371.305	8755.844	8816.457	8648.966	8614.722	8805.726	4235.861	4242.02
6	11/06/201	52294.35	8395.126	8939.938	8868.781	8602.886	8493.647	8679.959	4240.078	4221.426
7	11/06/201	52265.03	8365.504	8780.853	8841.77	8656.124	8539.637	8758.33	4241.185	4236.681
8	11/06/201	52200.33	8273.891	8515.236	8786.784	8692.363	8774.244	8967.027	4227.988	4236.521
9	11/06/201	52269.13	8314.213	8517.783	8764.371	8669.526	8749.137	8969.312	4243.73	4239.212
10	11/06/201	52240.6	8315.588	8502.105	8762.185	8694.639	8768.796	8997.852	4226.13	4222.528
11	11/06/201	52225.27	8341.257	8598.114	8773.122	8710.418	8684.651	8837.625	4230.447	4228.662
12	11/06/201	52229.49	8386.592	8750.592	8816.607	8672.125	8589.926	8748.082	4236.784	4242.681
13	11/06/201	52247.58	8362.145	8678.277	8791.177	8690.078	8661.421	8818.135	4232.385	4231.489
14	11/06/201	52269.99	8368.814	8569.9	8776.54	8710.702	8726.631	8881.762	4223.145	4221.157
15	11/06/201	52203.59	8347.282	8578.319	8797.155	8713.804	8690.696	8852.315	4241.262	4229.14
16	11/06/201	52230.63	8373.5	8973.697	8890.995	8627.002	8518.134	8599.773	4227.354	4228.813
17	11/06/201	52220.01	8352.874	8748.356	8805.805	8702.003	8627.553	8756.774	4237.672	4227.243

27 features

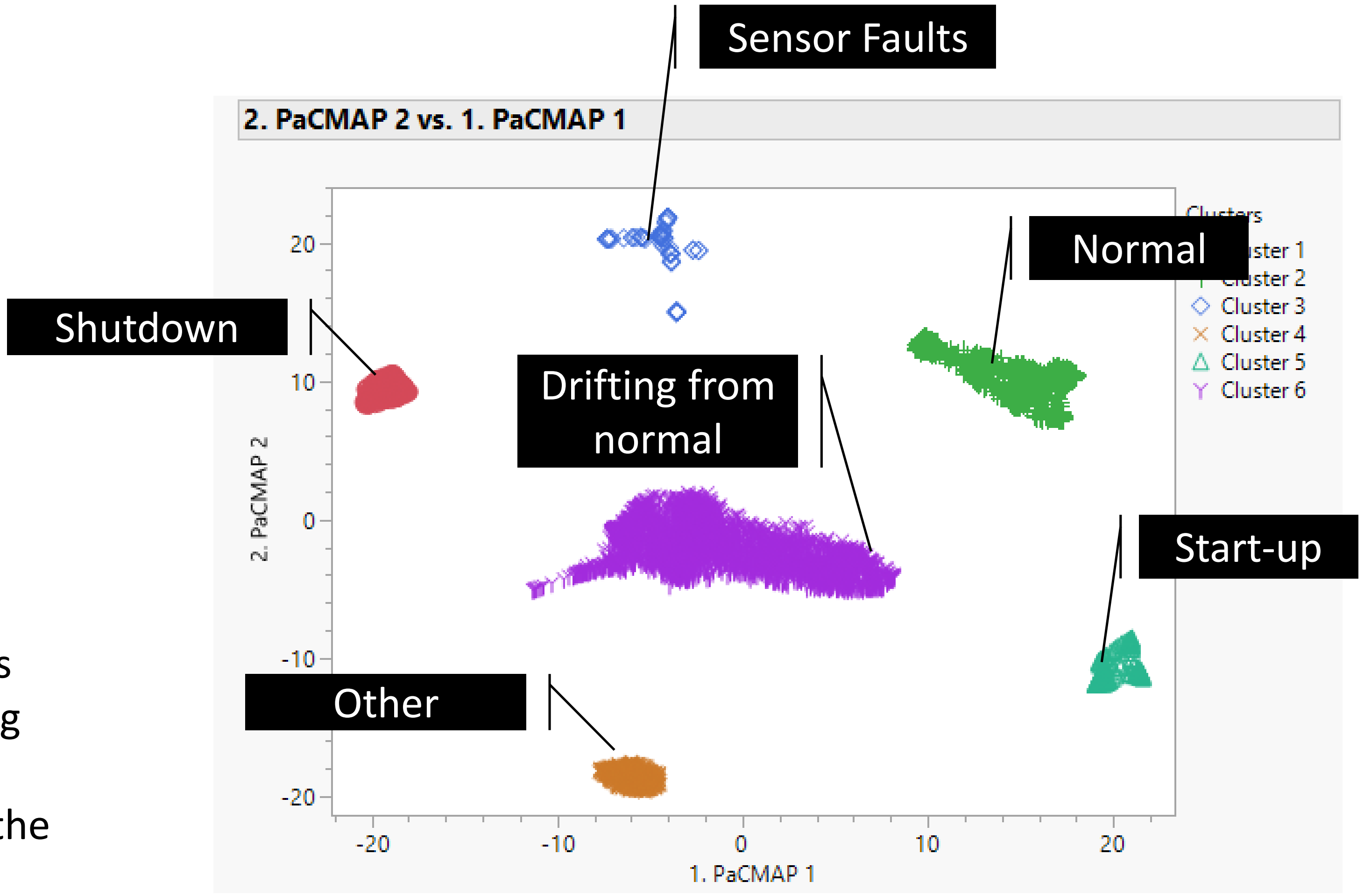
6670 rows of data (3 months of data)

Pyrolysis Reactor

After reducing the data to 2D using PaCMAP, six distinct clusters are identified using DBSCAN.

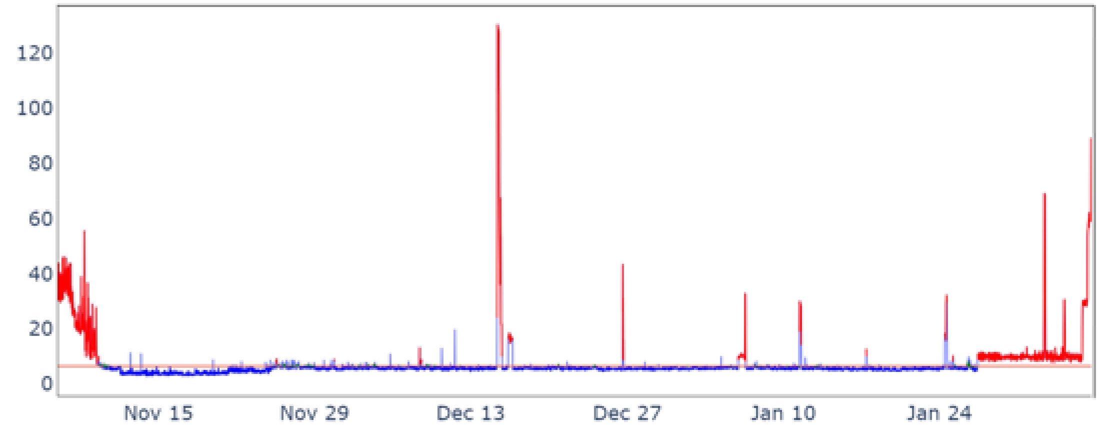
Although the clusters may not be immediately interpretable, each one corresponds to a specific operating condition.

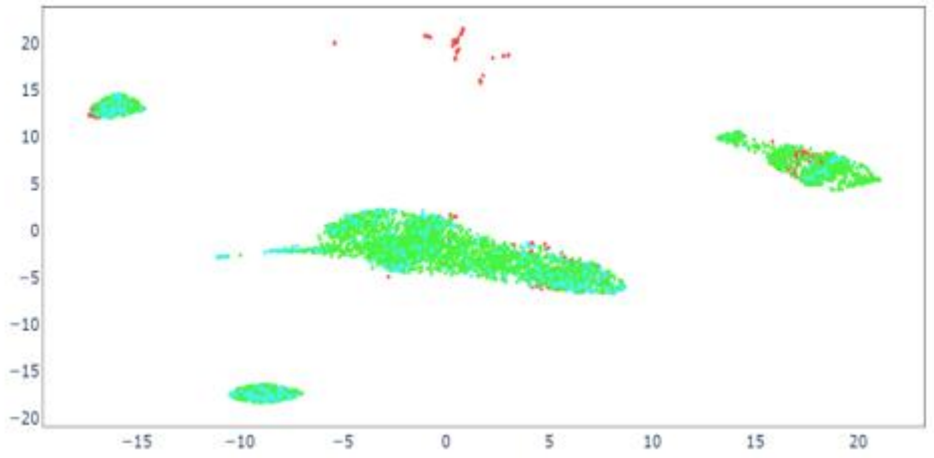
To validate these findings, the clusters can be matched with known operating conditions by comparing their timestamps, with confirmation from the industrial source.

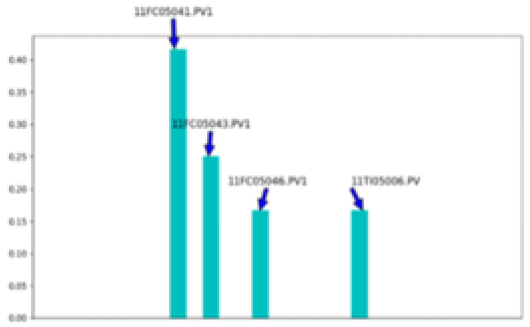


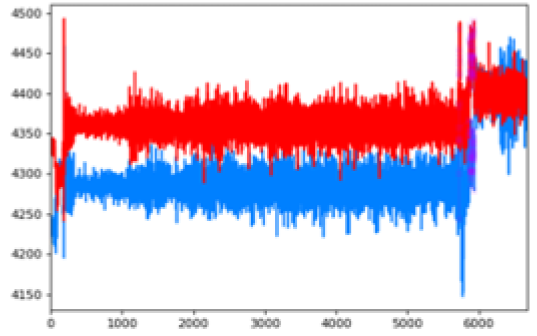
Pyrolysis Reactor

By utilizing the offline model in an online monitoring framework, faults can be detected and feature contribution plots can be created to find the fault contributors.





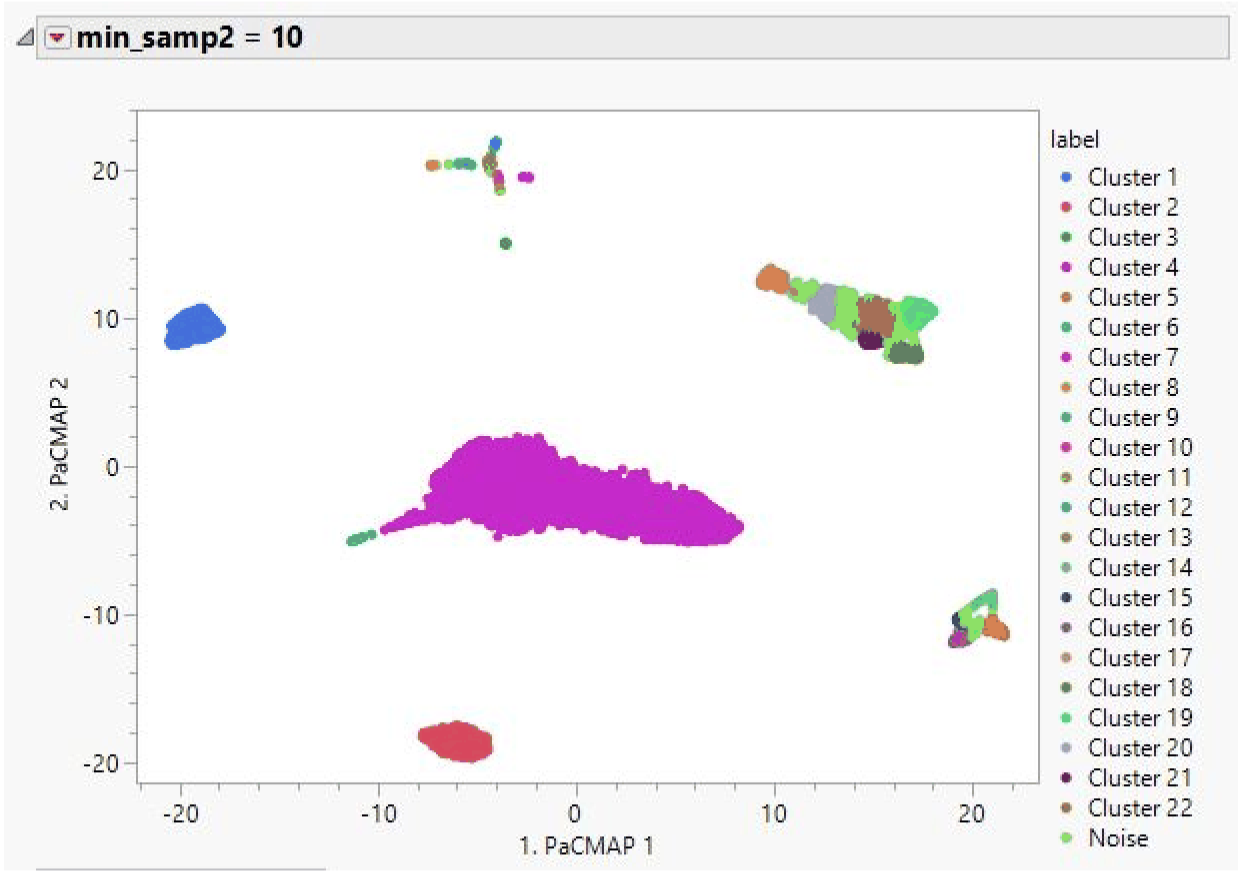




Fastman-JMP

- Show examples
- Show sensitivity analysis

Sensitivity Analysis


min_samp2 = 10