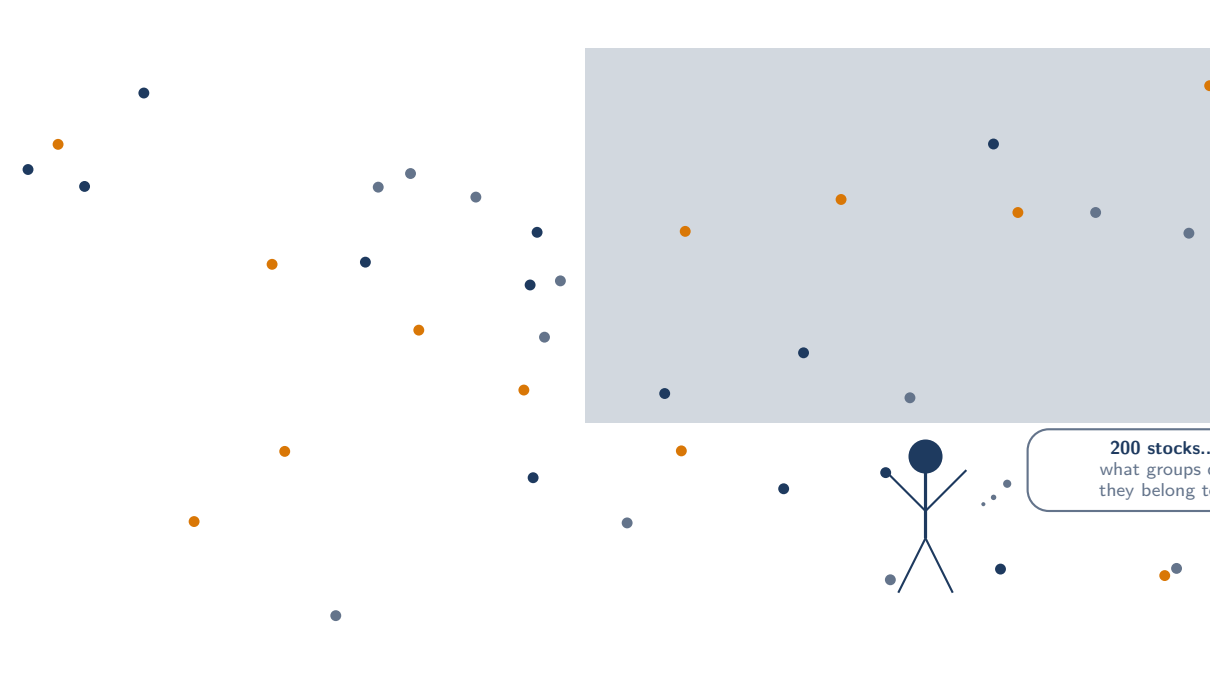
Unsupervised Learning: The Big Idea

Data Science with Python – BSc Course

25–30 Minutes

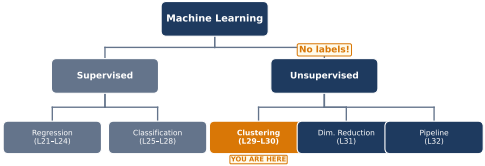

200 stocks..
what groups
they belong t

The Challenge: No Labels, Discover Structure

Why unsupervised?

- Supervised learning needs **labeled data** (each example paired with its correct answer) – labels are expensive or unavailable
- Unsupervised learning finds hidden patterns in unlabeled data
- Goal: discover structure the data reveals on its own

Read the chart: Unsupervised learning sits in the bottom branch – no labels needed.



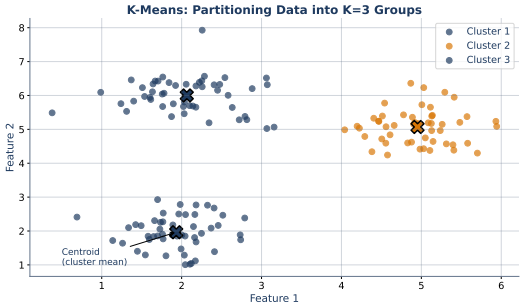
Group 500 stocks by return patterns – no one tells the algorithm which stocks "belong together."

Two Tools: Clustering and Dimensionality Reduction

Two ways to find structure

- **Clustering**: group similar observations together
- **Dimensionality reduction**: compress many features into fewer, meaningful ones
- Both reveal structure – from different angles

Read the chart: Points with similar colors are in the same cluster. Centers marked with X.



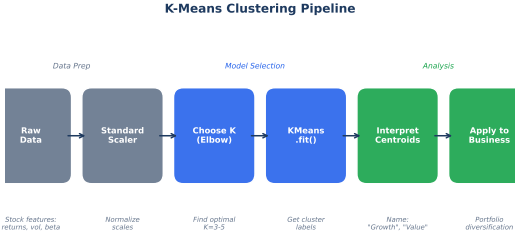
Clustering groups customers by spending; PCA compresses 50 financial ratios to 3 summary scores.

K-Means: Place Centers, Assign, Move, Repeat

The algorithm

- Pick **K** (the number of clusters you want) random centers, assign each point to the nearest
- Move each center to the mean of its assigned points
- Repeat until centers stop moving (**convergence**)

Read the chart: The flowchart shows the iterative loop: assign, update, check convergence.



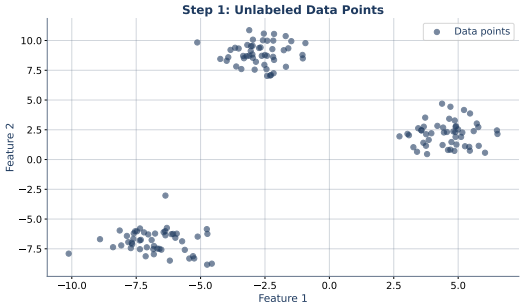
K = 3 groups 200 stocks into growth, value, and income clusters after 8 iterations.

K-Means in Action: From Raw Data to Clusters

Starting from scratch

- Step 1: raw data with no group assignments
- Each iteration refines cluster boundaries
- Final result: well-separated, compact groups

Read the chart: All points are the same color – no clusters yet. K-Means will color them.



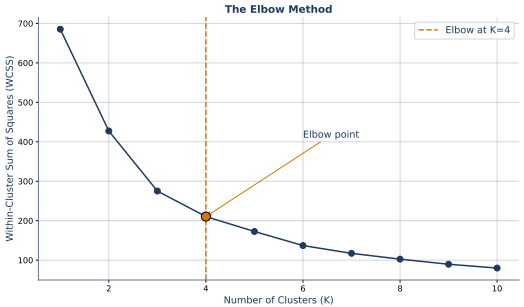
The algorithm starts from raw scatter and converges to stable clusters.

The Elbow Method: How Many Clusters?

Choosing K

- Run K-Means for K = 1, 2, 3, ... and measure **within-cluster variance** (average squared distance from each point to its center)
- The "elbow" marks where adding more clusters gives diminishing returns
- Choose K at the bend

Read the chart: Variance drops steeply until K=4, then flattens. The bend IS the elbow.



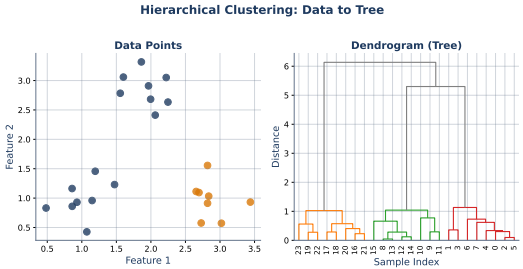
Elbow at K = 4 means 4 clusters explain the data well; K = 5 adds little.

Hierarchical Clustering: Merge From Bottom Up

Bottom-up merging

- Start with each point as its own cluster
- Merge the two closest clusters at each step
- A **dendrogram** (tree diagram showing the full merge history) lets you cut at any level for K groups

Read the chart: The tree grows upward as clusters merge. Cut at any height to get K groups.



Start with 50 individual stocks; merge closest pair at each step until 3 sector-like groups remain.

K-Means vs Hierarchical Clustering

Head-to-head comparison

- K-Means: fast, needs K upfront, assumes spherical clusters
- Hierarchical: slower, no K required, reveals nested structure
- Choose based on data size, shape, and whether you need a hierarchy

Read the chart: Side-by-side comparison shows how each method partitions the same data differently.

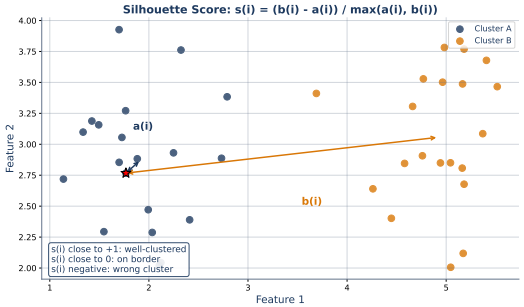
K-Means scales better; hierarchical clustering reveals richer structure.

Silhouette Score: Measuring Cluster Quality

Quantifying quality

- Measures how similar a point is to its own cluster vs the nearest other cluster
- Ranges from -1 (wrong cluster) to $+1$ (perfect fit)
- Average across all points summarizes overall quality

Read the chart: Each bar is one point. Long bars = well-assigned. Short or negative = misassigned.



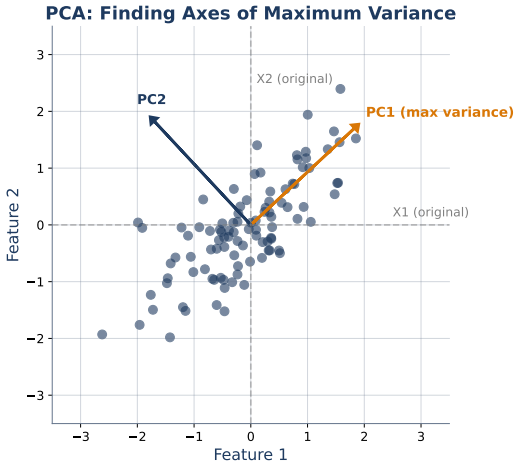
Silhouette = 0.71 (strong clusters) vs. 0.25 (points sit between groups).

PCA: Compress 50 Features to 3

Dimensionality reduction

- **Principal Component Analysis** finds directions of maximum variance
- Projects high-dimensional data onto fewer, informative axes
- Keeps the most important information while discarding noise

Read the chart: The arrow shows the direction of maximum variance. PCA aligns the new axis with it.



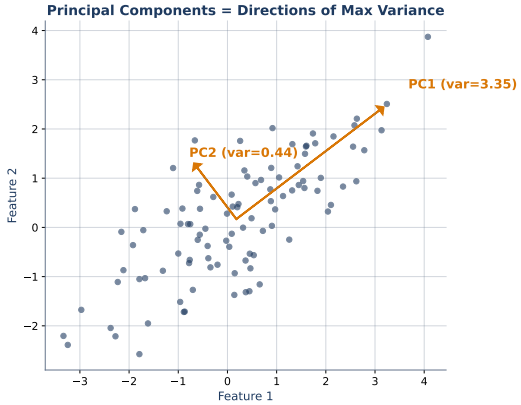
50 financial ratios compressed to 3 PCs that capture 87% of variation across 200 stocks.

Principal Components: Directions of Maximum Variance

What are components?

- PC1 captures the most variance, PC2 the second-most, and so on
- Each component is **orthogonal** (perpendicular, meaning completely uncorrelated) to all others
- The first few components often capture 80–90% of total variance

Read the chart: Two arrows show PC1 (long) and PC2 (short). PC1 captures more spread.



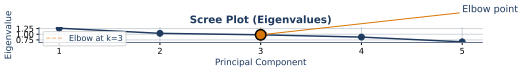
PC1 = "market direction" (captures 60%), PC2 = "sector rotation" (captures 15%).

Scree Plot: How Many Components to Keep?

Deciding component count

- Plot the **eigenvalue** (the amount of variance each component explains) in descending order
- Look for the "elbow" – components after it add little information
- Analogous to the elbow method in clustering

Read the chart: Bars drop steeply then flatten. The elbow tells you where to cut.



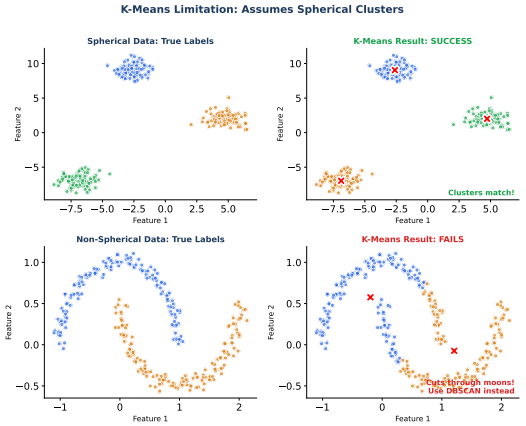
Eigenvalues 12.3, 4.1, 1.8, 0.9, ... – keep the first 3 (elbow at component 4).

K-Means Limitations: Non-Spherical Clusters

When K-Means struggles

- K-Means assumes clusters are round and equally sized
- Fails on elongated, ring-shaped, or uneven clusters
- Alternatives: **DBSCAN** (clusters by local density), **GMM** (soft probabilistic clusters), **spectral** (uses graph connectivity)

Read the chart: K-Means forces circular boundaries onto crescent-shaped data. The fit is poor.



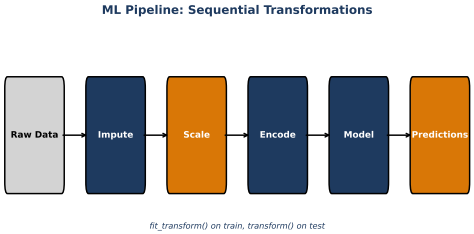
When clusters are not spherical, K-Means struggles – know your data shape.

Pipelines: Combine Preprocessing and Model

Chaining steps safely

- A pipeline chains steps: scale, reduce, cluster in one object
- Prevents **data leakage** (accidentally using test-set statistics during training)
- Makes code cleaner and reproducible

Read the chart: Arrows show data flowing through sequential steps. Each step transforms the data.



StandardScaler → PCA → K-Means in one pipeline, fit only on training data.

When to Use What

Criterion	K-Means	Hierarchical	PCA
Goal	Group points	Group points	Reduce features
Needs K upfront	Yes	No	No
Speed (large data)	Fast	Slow	Fast
Output	Cluster labels	Dendrogram + labels	Components
Cluster shape	Spherical	Any	N/A
Interpretable	Moderate	High	Low
Best for	Segmentation	Taxonomy	Compression

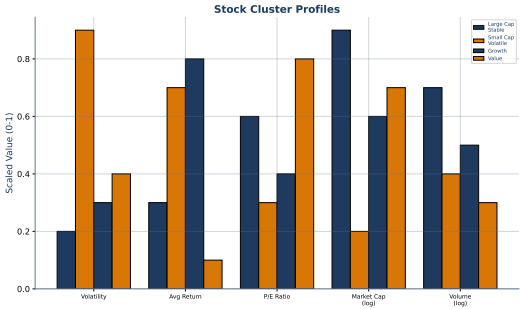
Choose the method that matches your goal: grouping or compressing.

Finance: Stock Cluster Profiles

Data-driven grouping

- Cluster stocks by return, volatility, and volume patterns
- Discover sector-like groups without using sector labels
- Useful for portfolio diversification and **regime detection** (identifying market phases like bull/bear)

Read the chart: Each cluster profile shows average return and volatility. Clusters differ meaningfully.



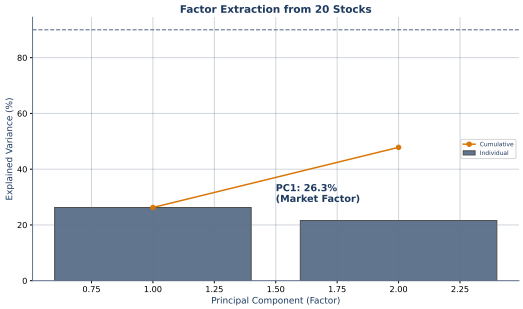
Clustering stocks reveals hidden groups that may differ from traditional sector labels.

Finance: Factor Discovery with PCA

Extracting latent factors

- Apply PCA to a universe of stock returns
- PC1 often resembles the market factor, PC2 a size or value tilt
- Data-driven factors complement traditional Fama-French models

Read the chart: Bar chart shows how much variance each PC explains. PC1 dominates.

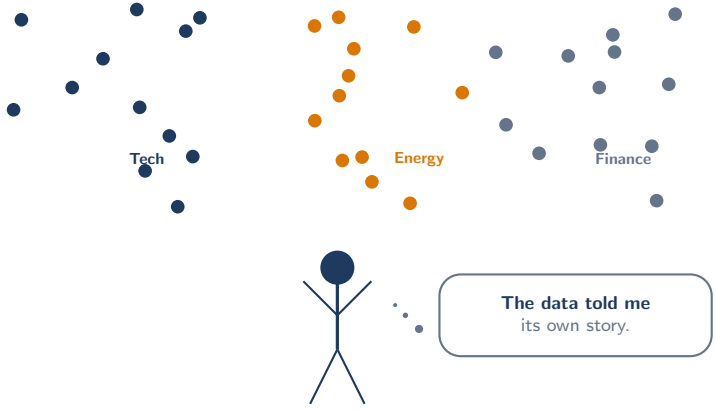


PCA extracts latent factors from return data – the data tells you what drives prices.

Key Takeaways

1. Unsupervised learning finds structure in data **without labels**
2. K-Means clusters by iterating assignment and center updates
3. The elbow method and silhouette score guide cluster count selection
4. Hierarchical clustering reveals nested group structures via dendrograms
5. PCA compresses features by projecting onto directions of maximum variance
6. Pipelines chain preprocessing and modeling for reproducible workflows

These tools let data reveal its own story – no labels required.


Tech
Energy
Finance
The data told me
its own story.