Week 3: Data Normalization, Unsupervised Learning & Clustering

Heart Failure Survival Analysis

MDST Project

Winter 2026

Outline

1. Week 2 Recap
2. Data Normalization
3. Principal Component Analysis (PCA)
4. K-Means Clustering
5. Hierarchical Clustering
6. Confusion Matrix
7. Why Clustering Struggles Here
8. Summary

Quick Recap: Week 2 - Statistical Analysis

Significant Features ($p < 0.05$):

1. time ($p \approx 10^{-22}$)
2. ejection_fraction ($p \approx 10^{-6}$)
3. serum_creatinine ($p \approx 10^{-5}$)
4. age ($p \approx 10^{-5}$)
5. serum_sodium ($p \approx 10^{-3}$)

Not Significant:

- diabetes, sex, smoking
- platelets, anaemia
- creatinine_phosphokinase

Key Point: Results held after FDR correction for multiple testing.

From Statistics to Unsupervised Learning

Week 2: Statistics	**Week 3: Unsupervised**
"Which features differ between groups?"	"Can we find natural groupings without labels?"
Uses the target variable (supervised)	Ignores the target variable
Tests one feature at a time	Considers all features together

This Week's Goal: Normalize data, reduce dimensions with PCA, and cluster patients to see if the survived/died groups emerge naturally.

Why Normalize?

Problem: Features are on very different scales!

Raw Feature Ranges:

- Platelets: 25,100 – 850,000
- Age: 40 – 95
- Ejection fraction: 14 – 80
- Anaemia: 0 or 1

Why This Matters:

- PCA will be dominated by large-scale features
- Clustering distances will be skewed
- Platelets “outweighs” all other features

Solution: Put all features on the same scale!

Z-Score Standardization

Formula:

$$z = \frac{x - \mu}{\sigma}$$

What It Does:

- Subtracts the mean (μ)
- Divides by standard deviation (σ)
- Result: mean $= 0$, std $= 1$

Interpretation:

- $z = 0$: at the average
- $z = +2$: 2 std above average
- $z = -1$: 1 std below average

Key Property: Z-scoring is a **linear transformation**. It does NOT change the shape of the distribution, the rank order, or any statistical test results.

Discussion

If you run PCA on non-normalized data, PC1 explains ∼90% of the variance by itself.

Is this a good thing? Why or why not?

What is PCA?

Problem: We have 12 features. Hard to visualize or understand.

PCA finds new axes (principal components) that capture the most variance in the data.

$$X \approx Z \cdot W^T$$

- X: original data (299 × 12)
- Z: scores in PC space (299 × k)
- W: loadings (how features contribute)

Goal: Find W that minimizes the reconstruction error between X and $Z \cdot W^T$

PC1 captures the most variance, PC2 the next most, etc.

Key PCA Concepts

Explained Variance Ratio:

- How much information each PC captures
- PC1: 13.9%, PC2: 13.2%, PC3: 10.6%
- No single PC dominates (after normalization!)

Cumulative Variance:

- How many PCs for 80%? 90%?
- Rule of thumb: keep enough for ~80–90%

Loadings:

- How strongly each feature contributes to each PC
- High loading = feature is important for that PC
- Positive/negative = direction of contribution

Scores:

- Each patient's coordinates in PC space
- Used for visualization (2D scatter plot)

PCA Scores Plot

Can we see separation between survived and died in PC space?

Plot PC1 vs PC2, colored by DEATH_EVENT

Observation: The two groups overlap heavily.

No clear boundary between survived and died in the first 2 PCs.

This hints that unsupervised methods may struggle.

From PCA to Clustering

Now that we can visualize the data in lower dimensions,

can unsupervised clustering recover the survived/died groups **without using the labels**?

How K-Means Works

Algorithm:

1. Choose K (number of clusters)
2. Randomly initialize K centroids
3. Assign each point to nearest centroid
4. Recalculate centroids as cluster means
5. Repeat steps 3–4 until convergence

Properties:

- Requires specifying K upfront
- Assumes spherical clusters
- Every point is assigned to exactly one cluster
- Sensitive to initialization (use `n_init`)
- Fast, widely used

Choosing K: Elbow Method

Inertia (within-cluster sum of squares):

$$\text{Inertia} = \sum_{i=1}^{n} \|x_i - c_k\|^2$$

where c_k is the centroid of point x_i's cluster.

How It Works:

- Plot inertia vs. K
- Inertia always decreases as K increases
- Look for the "elbow" – where the curve bends
- Diminishing returns after the elbow

Limitation:

- The elbow is often ambiguous
- Inertia ALWAYS decreases (at K=n, inertia=0)
- Use silhouette score as a complement

Choosing K: Silhouette Score

Silhouette Score measures how well each point fits its cluster:

$$s = \frac{b - a}{\max(a, b)}$$

For each data point:

- a = average distance to points in **same** cluster
- b = average distance to points in **nearest other** cluster

Interpretation:

- $s = +1$: perfectly clustered
- $s = 0$: on the boundary
- $s < 0$: probably in the wrong cluster
- $s < 0.25$: weak clustering structure

Our Dataset: Silhouette score at K=2 is below 0.25, indicating weak natural cluster structure.

K-Means Results: Confusion Matrix

How well do K-Means clusters (K=2) match the true labels?

Result: The clusters don't cleanly map to survived/died.

K-Means finds groupings based on feature similarity, but feature similarity $\neq$ same outcome.

Why? The survived and died groups overlap in feature space. Patients who died can look very similar to patients who survived based on their clinical measurements.

Hierarchical (Agglomerative) Clustering

Key Differences from K-Means:

- **Bottom-up**: starts with each point as its own cluster
- Merges the closest pair at each step
- No need to specify K upfront
- Produces a **dendrogram** (tree)
- Deterministic (no random initialization)

Linkage Methods:

- **Ward**: minimizes within-cluster variance (balanced)
- **Complete**: max distance between clusters (compact)
- **Average**: mean distance (compromise)
- **Single**: min distance (can chain)

Reading a Dendrogram

The Dendrogram shows the full merge history:

How to Read It:

- X-axis: samples (or cluster sizes)
- Y-axis: distance at which clusters merge
- **Long vertical lines** = well-separated clusters
- **Short vertical lines** = similar clusters merging

Choosing K:

- Draw a horizontal line at a chosen height
- Count how many vertical lines it crosses
- That's the number of clusters
- Look for the biggest "gap" in distances

Advantage over K-Means: You can explore different numbers of clusters from a single computation!

K-Means vs Hierarchical: Comparison

K-Means	**Hierarchical**
Must specify K before running	K chosen after seeing dendrogram
Random initialization (non-deterministic)	Deterministic (always same result)
Fast ($O(nK)$ per iteration)	Slower ($O(n^2 \log n)$)
Every point assigned to a cluster	Full merge history preserved
Assumes spherical clusters	Linkage method controls cluster shape

Evaluating Clusters with a Confusion Matrix

Since we know the true labels, we can compare clusters to reality:

How to Read It:

- Rows: true outcome (Survived/Died)
- Columns: cluster assignment (0/1)
- Diagonal = agreement
- Off-diagonal = disagreement

For Our Dataset:

- Neither K-Means nor Hierarchical cleanly separates the groups
- Both produce roughly similar results
- This tells us something important...

Discussion

Why do K-Means and hierarchical clustering fail to cleanly separate the survived and died groups?

Think about: feature overlap, what clustering optimizes, and the nature of clinical outcomes.

Why Unsupervised Methods Struggle on This Dataset

The Problem:

- Survived and died patients **overlap** heavily in feature space
- Strongest correlation with death is only $\sim$0.29
- No clean decision boundary exists
- Mortality depends on **complex interactions**, not just distance

What This Means:

- Clustering finds structure by **similarity**
- But similar features $\neq$ same outcome
- We need methods that **use the labels**
- This motivates **supervised learning**!

Key Insight: Unsupervised methods explore data structure.
Supervised methods predict specific outcomes.
They answer different questions.

Key Takeaways

1. **Normalization** is essential before PCA and clustering – prevents large-scale features from dominating
2. **PCA** reduces dimensions while preserving variance; cumulative variance plots help choose the number of components
3. **K-Means** partitions data into K clusters; use **elbow method** and **silhouette scores** to choose K
4. **Hierarchical clustering** provides a dendrogram – no need to choose K upfront
5. **Confusion matrices** compare unsupervised clusters to known labels
6. Unsupervised methods **cannot** cleanly recover the survived/died groups in this dataset

Next Week: Supervised Learning

- **Train/Test Split** – evaluating models fairly
- **Logistic Regression** – the simplest classifier
- **Random Forest** – ensemble of decision trees
- **ROC Curves & AUC** – measuring classification performance

Key Difference: Supervised methods *use the labels* during training, which is why they can learn patterns that clustering cannot find.

Discussion

Does normalizing the data change the results of a t-test or Mann-Whitney U test?

Why or why not?

Hint: Think about what z-scoring does to the difference in means and the standard error.

Exercises

1. Run PCA on **non-normalized** data. What happens and why?
2. Run PCA on normalized data **excluding the time column**. How do the loadings change?
3. Run K-Means with K=3 and plot the top 2 features colored by cluster
4. Compare K-Means clusters vs. true labels side-by-side on the top 2 features
5. Try different **linkage methods** for hierarchical clustering. How do the dendrograms change?
6. Re-run Week 2 statistical tests on normalized data. Are the p-values different?

Resources:

- Scikit-learn PCA: `https://scikit-learn.org/stable/modules/decomposition.html`
- Scikit-learn Clustering: `https://scikit-learn.org/stable/modules/clustering.html`
- StatQuest PCA: `https://www.youtube.com/watch?v=FgakZw6K1QQ`