Module 02: Clustering & PCA — Cheat Sheet

ML for Innovation Research — Prof. J. Osterrieder

K-Means Algorithm

1. Choose k (number of clusters).
2. Initialise k centroids randomly.
3. **Assign** each point to its nearest centroid.
4. **Update** each centroid to the mean of its members.
5. Repeat steps 3–4 until convergence.

Choosing k

- **Elbow method:** plot inertia vs. k; look for the "elbow."
- **Silhouette score:** quantitative measure (see below).
- **Domain knowledge:** does the number make sense?

Silhouette Score

For each sample i:

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

$a(i)$ mean intra-cluster distance
$b(i)$ mean nearest-cluster distance

Range	Interpretation
$+1$	Perfect separation
≈ 0	Overlapping clusters
-1	Misclassified
> 0.5	Good structure
0.25–0.5	Weak structure

Principal Component Analysis (PCA)

- **Purpose:** reduce dimensions while retaining maximum variance.
- Finds orthogonal axes (principal components) ranked by variance explained.
- PC1 captures the most variance, PC2 the second-most, etc.
- **Variance explained ratio** tells you how much information each PC retains.
- Common rule: keep enough PCs to explain ≥80–90% of total variance.

UMAP vs. PCA

PCA	UMAP
Linear	Non-linear
Fast, deterministic	Slower, stochastic
Global structure	Local structure preserved
Good for preprocessing	Better for visualisation
Interpretable axes	Axes not interpretable

Key Code Snippets

```
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.metrics import silhouette_score
from sklearn.preprocessing import StandardScaler

# Always scale before clustering/PCA
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# K-Means
km = KMeans(n_clusters=4, random_state=42,
            n_init=10)
labels = km.fit_predict(X_scaled)
sil = silhouette_score(X_scaled, labels)

# PCA for visualisation
pca = PCA(n_components=2)
X_2d = pca.fit_transform(X_scaled)
print(pca.explained_variance_ratio_)
```

Common Pitfalls

- Forgetting to **scale** features before K-Means or PCA.
- Using PCA components as cluster *inputs* (cluster on original scaled data, then project for plotting).
- Interpreting PCA axes as single original features.