In-Class Exercises: Unsupervised Learning

Work through these during the lecture. Ask if you get stuck.

Data Science with Python – BSc Course

Exercise 1: K-Means by Hand

8 min | Pen & Paper

Six data points in 2D:

Point	x	y
A	1	1
B	2	1
C	1.5	2
D	7	8
E	8	7
F	8	9

We run K-Means with $K = 2$ and initial centroids $\mu_1 = (1, 1)$ and $\mu_2 = (8, 8)$.

Iteration 1

(a) Compute the Euclidean distance from each point to both centroids. Use $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.

(b) Assign each point to the nearest centroid. Which points go to Cluster 1? Which to Cluster 2?

(c) Recompute each centroid as the mean of its assigned points.

$$\mu_1^{\text{new}} = \left(\frac{?+?+?}{3}, \frac{?+?+?}{3}\right) \qquad \mu_2^{\text{new}} = \left(\frac{?+?+?}{3}, \frac{?+?+?}{3}\right)$$

Iteration 2

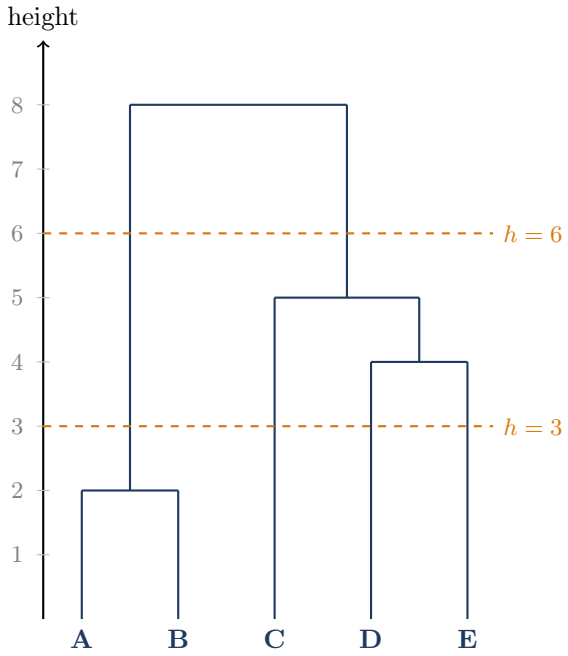
(d) Repeat the assignment step with the new centroids. Do any points change clusters?

(e) What does it mean when no points change clusters?

Exercise 2: Read the Dendrogram

5 min | Conceptual

The dendrogram below is built from five items A–E (the same kind of tree shown in the lecture, Hierarchical Clustering – Dendrogram). The vertical axis is the merge distance; each horizontal bar marks the height at which two clusters join.



(a) If you cut the dendrogram at height $h = 3$, how many clusters do you get? List which items are in each cluster.

(b) If you cut at height $h = 6$, how many clusters?

(c) Which two items (or sub-clusters) merge *first* (at the lowest height)? What does a low merge height tell you about those items?

(d) How does the dendrogram help you choose the number of clusters without running K-Means multiple times?

Exercise 3: PCA – Keep or Drop?

7 min | Conceptual

A dataset with 5 features produces these eigenvalues after PCA:

Component	PC1	PC2	PC3	PC4	PC5
Eigenvalue	4.2	1.8	0.5	0.3	0.2

(a) Compute the explained variance ratio for each component. Fill in the table:

Component	PC1	PC2	PC3	PC4	PC5
Expl. Variance (%)					
Cumulative (%)					

(b) Apply the **Kaiser rule** (keep components with eigenvalue > 1). How many components do you keep?

(c) Apply the **elbow rule** (look for the bend in the scree plot). Where is the elbow? How many components?

(d) Apply the **90% rule** (keep enough components to capture at least 90% of total variance). How many components?

(e) Do the three rules agree? What would you recommend and why?

Exercise 4: Which Method?

5 min | Discussion

Match each dataset to the most appropriate unsupervised method. Write your reasoning — there may be more than one defensible answer.

Dataset	Description	Best Method?
A	10,000 customers, 5 features (age, income, spending score, account balance, transactions). You want exactly 4 marketing segments.	
B	200 stocks. You want to see which stocks are most similar and build a tree showing the hierarchy of relationships.	
C	Credit-card transactions. Most are legitimate, but some are fraudulent. You do not know which ones. Fraudulent transactions look different from normal ones.	
D	A dataset with 50 features. You need to reduce it to 3 features for visualization and downstream modeling.	

Methods to choose from: K-Means, Hierarchical Clustering, DBSCAN, PCA, GMM, t-SNE

Bonus: For dataset C, why would K-Means be a poor choice?

Exercise 5: Build a Pipeline

10 min | Python (Optional)

Complete the blanks in this `sklearn` pipeline that scales the data, reduces dimensions with PCA, and clusters with K-Means.

```
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import ________  # (1) Which scaler?
from sklearn.decomposition import ________  # (2) Which reducer?
from sklearn.cluster import ________        # (3) Which clusterer?
import numpy as np

# Synthetic data: 500 samples, 10 features
np.random.seed(42)
X = np.random.randn(500, 10)

# Build the pipeline
pipe = Pipeline([
    ('scaler',  ________(________)),  # (4) Instantiate scaler
    ('pca',     ________(________)),  # (5) Reduce to 3 components
    ('cluster', ________(________)),  # (6) Find 4 clusters
])

# Fit and get cluster labels
labels = pipe._________(X)            # (7) Which method?

print("Cluster sizes:", np.bincount(labels))
print("PCA explained variance:",
      pipe.named_steps['pca'].explained_variance_ratio_)
```

(a) Fill in blanks (1)–(7).

(b) Why is it important that the scaler comes *before* PCA and K-Means in the pipeline?

(c) If you used this pipeline inside `cross_val_score`, what would happen if you had fitted the scaler *outside* the pipeline on the full dataset first?

Exercise 6: Cluster Real Stocks

10 min | Python (Bonus)

Use K-Means to cluster synthetic stock data and interpret the results.

```python
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans

np.random.seed(42)

# Generate synthetic stock features
stocks = pd.DataFrame({
    'annual_return': np.concatenate([
        np.random.normal(0.05, 0.02, 20),   # stable
        np.random.normal(0.25, 0.08, 15),   # growth
        np.random.normal(0.50, 0.15, 10),   # speculative
    ]),
    'volatility': np.concatenate([
        np.random.normal(0.10, 0.03, 20),
        np.random.normal(0.25, 0.05, 15),
        np.random.normal(0.45, 0.10, 10),
    ]),
    'market_cap_B': np.concatenate([
        np.random.normal(100, 30, 20),
        np.random.normal(30, 10, 15),
        np.random.normal(5, 3, 10),
    ]),
})

# Step 1: Scale the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(stocks)

# Step 2: Fit K-Means with K=3
km = KMeans(n_clusters=3, random_state=42, n_init=10)
stocks['cluster'] = km.fit_predict(X_scaled)

# Step 3: Examine cluster profiles
print(stocks.groupby('cluster').mean().round(3))
```

(a) Run the code. What does each cluster represent? Give each cluster a descriptive name (e.g., "Blue Chips", "Growth Stocks", "Speculative Bets").

(b) Look at the centroid values via `km.cluster_centers_`. Remember these are in *scaled* space. How would you convert them back to the original scale?

(c) What would happen if you forgot the `StandardScaler` step? Which feature would dominate the clustering and why?