Clustering and Ecosystem Pattern Analysis

Learning Outcomes

After completing this unit, students will be able to:

- Apply clustering techniques to identify patterns and groupings in environmental datasets.
- Evaluate cluster quality using internal and external validation metrics.
- Interpret ecological meaning from unsupervised models.
- Use clustering results to support conservation and ecosystem management.

Introduction

Clustering is a fundamental unsupervised learning approach that organizes data samples into groups based on similarity, without requiring predefined labels. In environmental analytics, it is particularly valuable because ecosystems often display complex and multi-dimensional structures where clear class boundaries are not evident. By identifying natural groupings within data, clustering supports tasks such as biodiversity assessment, soil chemistry analysis, and the interpretation of remote sensing observations, while also accommodating noise and irregular patterns that are common in ecological measurements [4,6,8]. Unsupervised clustering applied to remotely sensed productivity data has been linked to large-scale biodiversity patterns [2].

K-Means Clustering

K-Means partitions the dataset into k clusters by iteratively assigning points to the nearest centroid and updating centroid positions to minimize intra- cluster variance. It is one of the most widely used clustering algorithms due to its simplicity and computational efficiency.

Advantages:

- Fast and scalable to large datasets.
- Easy to implement and interpret.

Limitations:

- Requires the number of clusters k to be specified in advance.
- Sensitive to initialization and outliers.

- Works best with spherical, equally sized clusters.

Environmental applications: Detecting pollution patterns in water quality measurements [9].

A key limitation of the K-Means algorithm is the requirement to specify the number of clusters (*k*) in advance. In environmental applications, the true number of ecosystem regimes is rarely known a priori, and inappropriate choices of *k* may lead to oversimplified or fragmented representations of ecological structure.

To address this issue, internal validation techniques are commonly used to guide the selection of *k*. These methods assess clustering quality based on compactness and separation of clusters, without relying on external labels. Widely used approaches include the Elbow method, which examines the reduction of within-cluster variance as *k* increases, and Silhouette analysis, which evaluates how well each data point fits within its assigned cluster compared to neighboring clusters.

In practice, the final selection of *k* often balances quantitative validation results with ecological interpretability, ensuring that identified clusters represent meaningful environmental regimes rather than purely mathematical partitions.

Hierarchical Clustering

Hierarchical clustering creates a hierarchy of clusters, often visualized as a dendrogram, which reveals nested levels of similarity among samples. It can be agglomerative (bottom-up) or divisive (top-down). This method is especially useful for ecological studies that require understanding relationships across multiple scales.

Advantages:

- Does not require specifying the number of clusters a priori.
- Produces dendrograms that reveal multi-scale ecological structures.
- Flexible linkage criteria (single, complete, average).

Limitations:

- Computationally intensive for very large datasets.
- Sensitive to noise and scaling of variables.
- Results can be difficult to interpret if clusters are not well separated.

Environmental applications: Clustering hydrological catchments based on flow and precipitation regimes [5].

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

DBSCAN groups together points that are closely packed while marking points in low-density regions as noise. Unlike K-Means, it does not require the number of clusters to be specified in advance, and it can identify clusters of arbitrary shape.

Advantages:

- Can detect clusters of irregular shape.
- Automatically identifies noise/outliers.
- No need to predefine the number of clusters.

Limitations:

- Performance depends heavily on parameter choice (epsilon, minPts).
- Struggles with varying cluster densities.
- Less scalable than K-Means for very large datasets.

Environmental applications: Detecting pollution hotspots in air or water quality sensor networks [3, 7].

Cluster evaluation metrics

Unlike supervised classification, clustering lacks ground-truth labels in most environmental datasets. Evaluating clustering quality therefore relies on internal or external validation indices that quantify compactness, separation, or agreement with expert-defined groupings.

Silhouette Score

- Measures how similar each point is to its own cluster compared to other clusters.
- Ranges from -1 to 1; higher values indicate well-separated and cohesive clusters.
- Useful for selecting the number of clusters k in K-Means.

Davies–Bouldin Index

- Ratio of within-cluster scatter to between-cluster separation.
- Lower values indicate better clustering.
- Computationally efficient, applicable to large environmental datasets.

Calinski–Harabasz Index

- Compares between-cluster variance to within-cluster variance.
- Higher values correspond to better-defined clusters.
- Frequently used in ecological applications with multivariate data.

Adjusted Rand Index (ARI)

- External validation metric that compares clustering results to expert- labeled data or known ecological regions.
- Adjusted for chance, with 1 indicating perfect agreement and 0 indicating random partitioning.
- Valuable when validating against established ecoregion maps or conservation units.

Cluster validation ensures that discovered structures are not artifacts of the algorithm but reflect meaningful ecological patterns. In environmental studies, combining internal indices (e.g., Silhouette, Davies–Bouldin) with external validation against expert-based classifications provides robust evidence for ecological interpretations.

Case Study: Identifying ecosystem regimes using unsupervised clustering of environmental data

Objective

This case study demonstrates how unsupervised learning techniques can be used to uncover natural groupings and ecological regimes in environmental data without predefined labels. The objective is to identify regions with similar environmental characteristics and to interpret these clusters as distinct ecosystem or environmental regimes. Such pattern discovery is particularly valuable in ecological analysis, where natural boundaries are often gradual, fuzzy, or unknown.

Libraries used

The case study is implemented in Python, using widely adopted open-source libraries for data analysis, machine learning, and visualization:

- **NumPy**
  Used for numerical operations, array manipulation, and random sampling.
- **Pandas**
  Used for loading, handling, and preprocessing tabular environmental datasets.

- **Scikit-learn (sklearn).**

  Used for:

  - data preprocessing (StandardScaler),
  - clustering algorithms (K-Means),
  - dimensionality reduction (PCA),
  - internal cluster validation metrics (Silhouette Score, Davies–Bouldin Index).

- **SciPy**

  Used for hierarchical clustering and dendrogram visualization.

- **Matplotlib**

  Used for visualizing clustering results, scatter plots, PCA projections, and dendrograms.

This choice supports transparency and makes the workflow easy to run in standard environments such as Jupyter Notebook or Google Colab.

Data source

The analysis is based on the Forest Cover Type (Covertype) [1] dataset, an open-access dataset provided by the UCI Machine Learning Repository. The dataset contains environmental and topographic measurements collected from forested areas in the Roosevelt National Forest, USA. Although the dataset includes a land cover label, this information is intentionally excluded in this case study to preserve the unsupervised nature of the analysis.

Data description

Each observation represents a geographic location described by environmental variables such as:

- Elevation
- Slope
- Aspect
- Horizontal and vertical distances to hydrology
- Distance to roadways
- Hillshade indices

These variables jointly describe terrain, moisture gradients, and accessibility, which are key drivers of ecosystem differentiation.

Data preprocessing

To prepare the dataset for clustering, the following steps are applied:

- Selection of continuous environmental and topographic variables only
- Removal of the forest cover type label
- Standardization of variables to zero mean and unit variance to ensure equal contribution to distance-based clustering methods

```
import pandas as pd
from sklearn.preprocessing import StandardScaler

# Load dataset (CSV version of Covertype)
df = pd.read_csv("covtype.csv")

# Drop the cover type label (last column)
X = df.iloc[:, :-1]

# Standardize features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

pd.DataFrame(X_scaled, columns=X.columns).head()
```

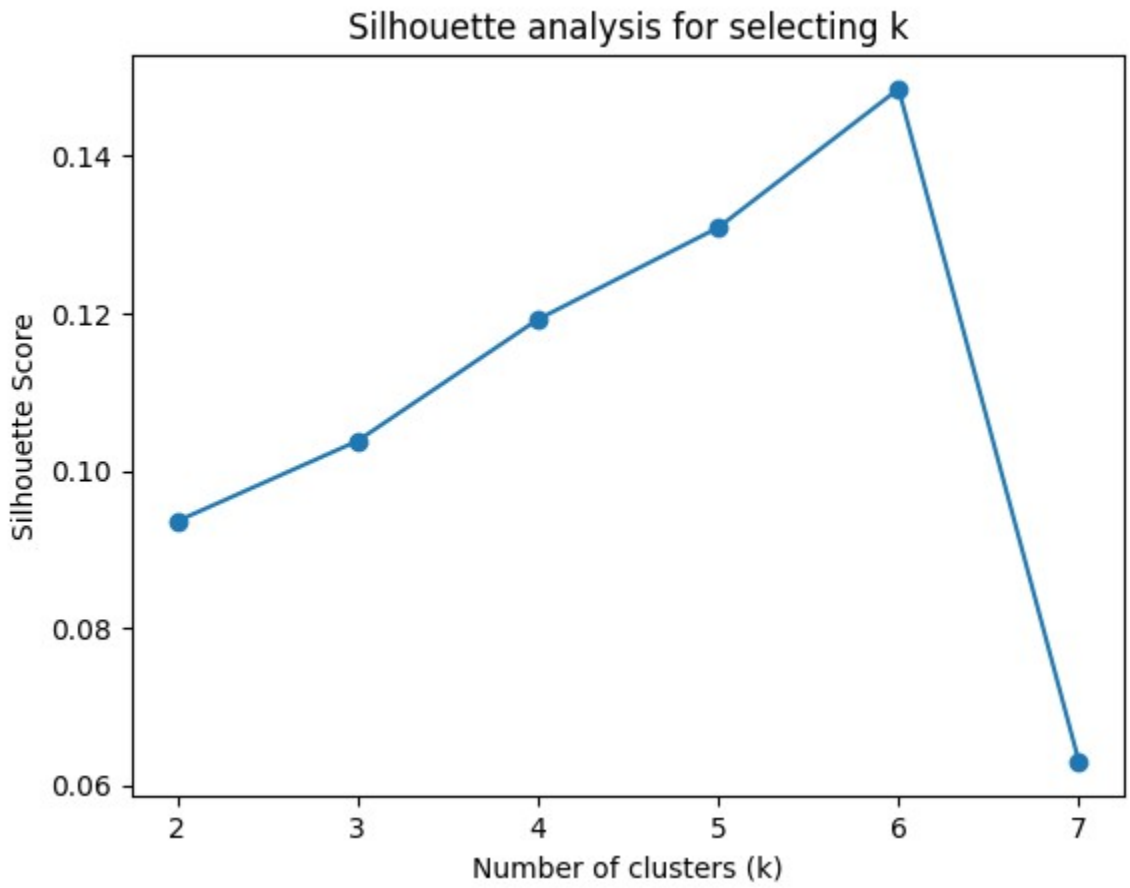
Clustering methods

Two complementary clustering techniques are applied:

K-Means clustering

K-Means is used to partition the dataset into a predefined number of clusters based on environmental similarity. Before applying K-Means clustering, the number of clusters was examined using internal validation techniques. The Silhouette analysis was employed to evaluate clustering quality across different values of *k*. These analyses support the selection of a cluster number that balances separation, compactness, and ecological interpretability.

```python
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
import matplotlib.pyplot as plt

# Range of k values to test
K_range = range(2, 8)

silhouette_scores = []

for k in K_range:
    print(f"Computing Silhouette Score for k = {k} ...")

    kmeans = KMeans(n_clusters=k, random_state=42)
    labels = kmeans.fit_predict(X_scaled)

    score = silhouette_score(X_scaled, labels)
    silhouette_scores.append(score)

# Plot results
plt.figure()
plt.plot(K_range, silhouette_scores, marker='o')
plt.xlabel("Number of clusters (k)")
plt.ylabel("Silhouette Score")
plt.title("Silhouette analysis for selecting k (full dataset)")
plt.show()
```



Silhouette analysis indicated higher scores for k = 6, suggesting improved cluster separation from a purely statistical perspective. However, in environmental applications, the optimal number of clusters is not determined solely by internal validation metrics. Ecosystems are governed by continuous gradients rather than sharply separated classes, and increasing the number of clusters may fragment ecologically coherent regimes into smaller, less interpretable units.

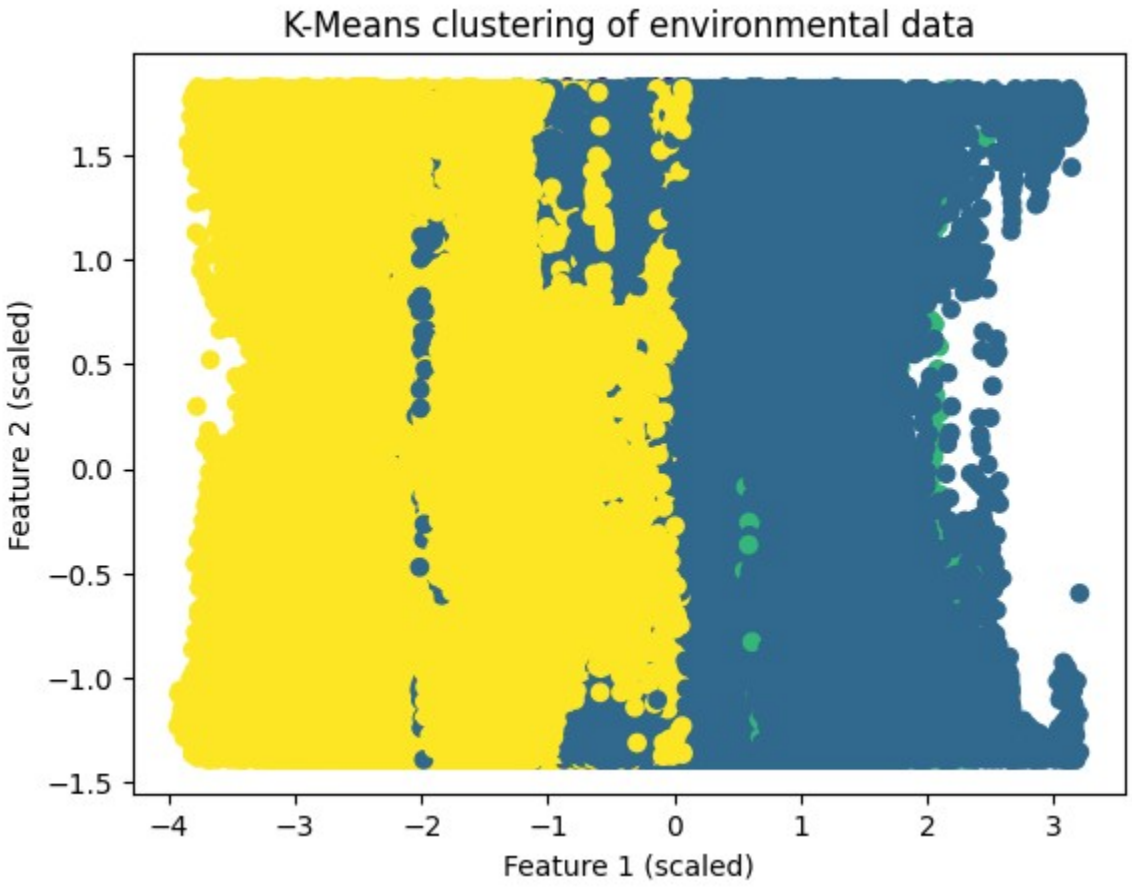
Therefore, a smaller number of clusters (k = 4) was selected to balance statistical clustering quality with ecological interpretability and pedagogical clarity. This choice enables the identification of broad and meaningful ecosystem regimes that align with dominant environmental gradients, while avoiding over-partitioning that could obscure practical understanding and application

```python
from sklearn.cluster import KMeans

# Apply K-Means
kmeans = KMeans(n_clusters=4, random_state=42)
clusters = kmeans.fit_predict(X_scaled)

# Add cluster labels to dataframe
df["Cluster"] = clusters

df[["Cluster"]].value_counts()
```

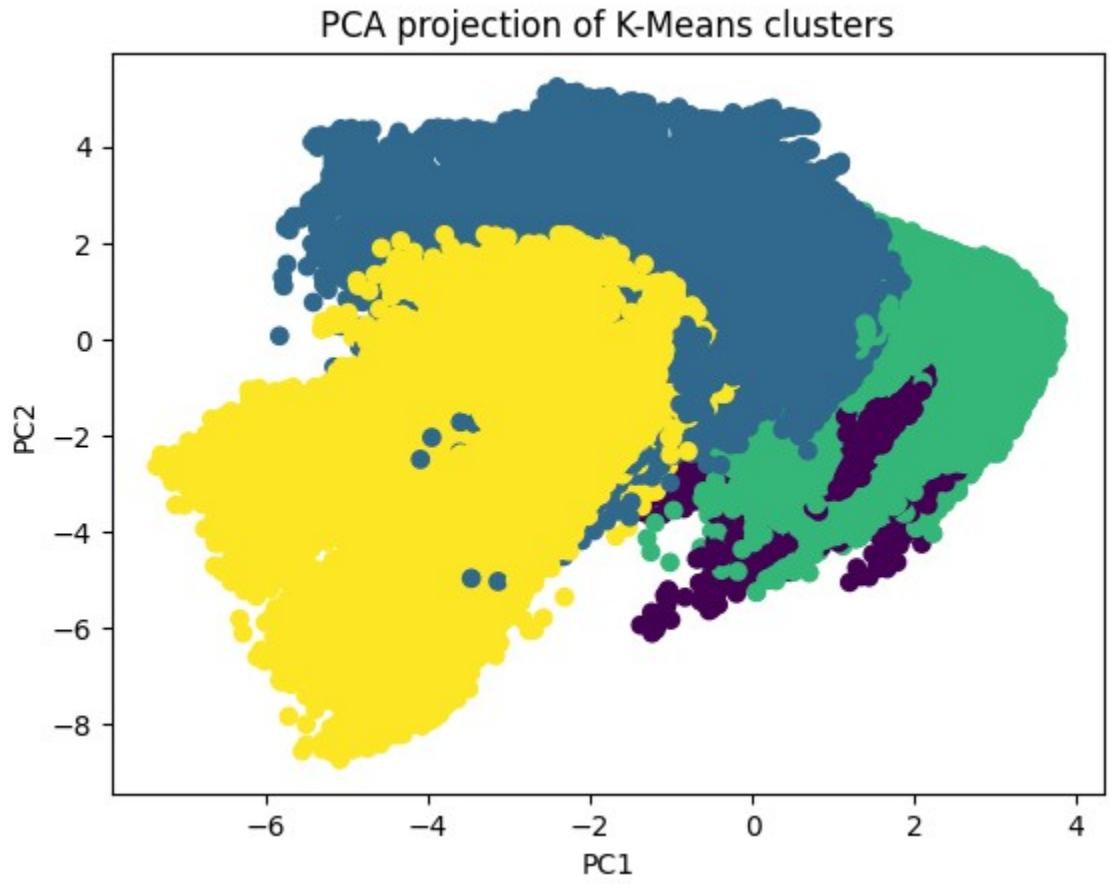
✓ 1.9s

```
Cluster
1          236975
2          228173
3           84547
0           31317
Name: count, dtype: int64
```

```python
import matplotlib.pyplot as plt

plt.figure()
plt.scatter(X_scaled[:, 0], X_scaled[:, 1], c=clusters)
plt.xlabel("Feature 1 (scaled)")
plt.ylabel("Feature 2 (scaled)")
plt.title("K-Means clustering of environmental data")
plt.show()
```



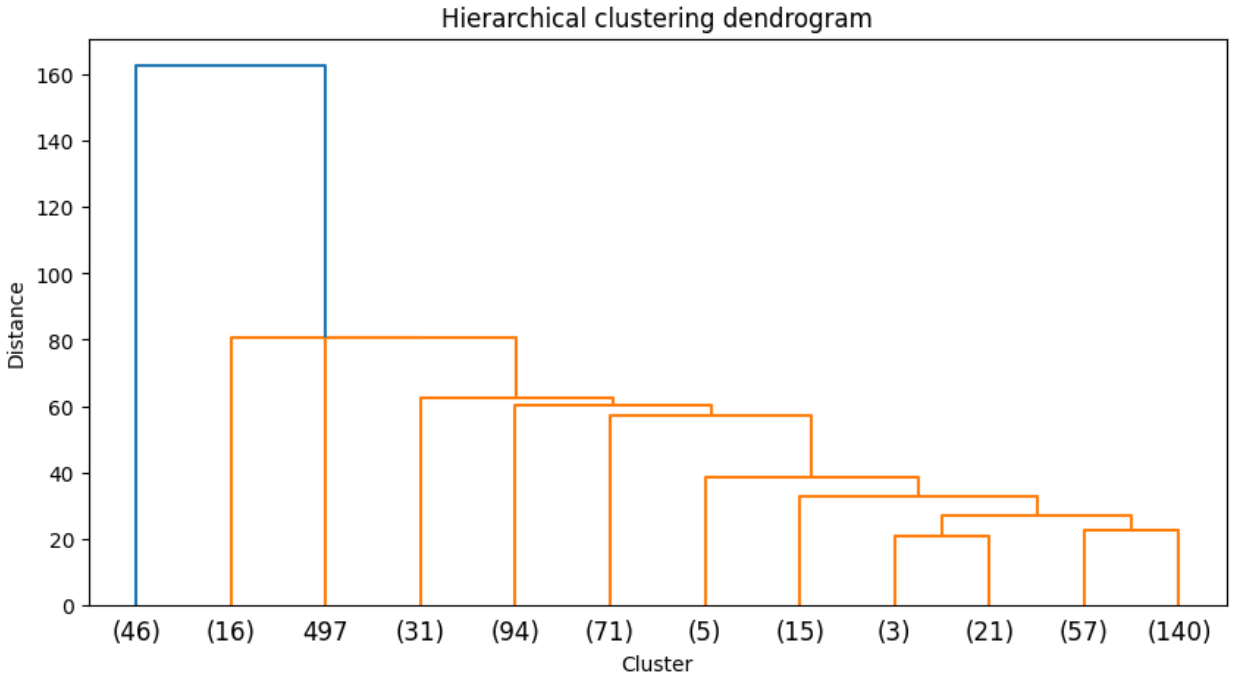

```python
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

# Apply PCA to reduce to 2 dimensions
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)

plt.figure()
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=clusters)
plt.xlabel("PC1")
plt.ylabel("PC2")
plt.title("PCA projection of K-Means clusters")
plt.show()
```



Hierarchical clustering

Hierarchical agglomerative clustering is applied to explore nested relationships among sampling locations. Dendrograms are used to visualize multi-scale ecological structures and to reveal how local patterns aggregate into broader ecosystem groupings.

```
from scipy.cluster.hierarchy import linkage, dendrogram

# Use a subset for visualization
X_subset = X_scaled[:500]

linked = linkage(X_subset, method="ward")

plt.figure(figsize=(10, 5))
dendrogram(linked, truncate_mode="lastp", p=12)
plt.title("Hierarchical clustering dendrogram")
plt.xlabel("Cluster")
plt.ylabel("Distance")
plt.show()
```



Cluster evaluation

Cluster quality is assessed using internal validation indices:

- Silhouette Score
- Davies–Bouldin Index

These metrics help ensure that the identified clusters are compact, well-separated, and not artifacts of random partitioning.

```
from sklearn.metrics import silhouette_score, davies_bouldin_score

sil_score = silhouette_score(X_scaled, clusters)
db_score = davies_bouldin_score(X_scaled, clusters)

sil_score, db_score
```

The computation of internal validation metrics such as the Silhouette Score and the Davies–Bouldin Index can be computationally demanding for large-scale environmental datasets, as these metrics rely on pairwise distance calculations. Given the size of the dataset used in this case study, the evaluation of clustering quality was therefore performed on a representative random subset of the data. This approach preserves the underlying data structure while ensuring computational feasibility and is a common practice in large-scale clustering analyses. The resulting metrics provide reliable comparative insights into clustering quality without requiring exhaustive computation over the full dataset.

```
import numpy as np
from sklearn.metrics import silhouette_score, davies_bouldin_score

# Subsample for faster evaluation
np.random.seed(42)
sample_idx = np.random.choice(len(X_scaled), size=5000, replace=False)

X_sample = X_scaled[sample_idx]
clusters_sample = clusters[sample_idx]

sil_score = silhouette_score(X_sample, clusters_sample)
db_score = davies_bouldin_score(X_sample, clusters_sample)

sil_score, db_score
```

The computation of internal validation metrics such as the Silhouette Score and the Davies–Bouldin Index can be computationally demanding for large-scale environmental datasets, as these metrics rely on pairwise distance calculations. Given the size of the dataset used in this case study, the evaluation of clustering quality was therefore performed on a representative random

subset of the data. This approach preserves the underlying data structure while ensuring computational feasibility and is a common practice in large-scale clustering analyses. The resulting metrics provide reliable comparative insights into clustering quality without requiring exhaustive computation over the full dataset.

```
import numpy as np
from sklearn.metrics import silhouette_score, davies_bouldin_score

# Subsample for faster evaluation
np.random.seed(42)
sample_idx = np.random.choice(len(X_scaled), size=5000, replace=False)

X_sample = X_scaled[sample_idx]
clusters_sample = clusters[sample_idx]

sil_score = silhouette_score(X_sample, clusters_sample)
db_score = davies_bouldin_score(X_sample, clusters_sample)

sil_score, db_score
```

Interpretation and ecological meaning

The resulting clusters can be interpreted as distinct ecosystem regimes, such as:

- Low-elevation vs. high-elevation forest environments
- Moist vs. dry terrain conditions
- Accessible vs. remote mountainous regions

By comparing the clusters with known environmental gradients, students can associate each group with characteristic ecological conditions rather than predefined land cover classes.

Relevance for ecosystem management

The discovered clusters provide an objective basis for:

- Ecosystem classification and mapping
- Identification of environmentally similar regions for conservation planning
- Supporting land-use management decisions in data-scarce or poorly mapped areas

This case study illustrates how clustering complements expert knowledge by revealing data-driven ecological structures that may not align with administrative boundaries.

Exercise Tasks

- Download the Forest Cover Type dataset (Covertype) from the UCI Machine Learning Repository [1].
- Explore the dataset and perform an initial descriptive analysis of the environmental and topographic variables.
- Select a subset of relevant continuous variables (e.g., elevation, slope, hydrological distances) and justify the selection.
- Standardize the selected variables and examine how scaling affects distance-based clustering.
- Use internal validation techniques (Elbow method and Silhouette analysis) to explore different values of the number of clusters (*k*).
- Apply K-Means clustering using the selected value of *k* and analyze the resulting cluster structure.
- Perform hierarchical clustering on a reduced subset of the data and visualize multi-scale relationships using a dendrogram.
- Compare the clustering results obtained from K-Means and hierarchical methods, highlighting similarities and differences.
- Interpret the clusters in terms of dominant environmental gradients rather than predefined land-cover classes.
- Reflect on the limitations of clustering for ecosystem analysis and discuss how results could complement expert-based ecosystem classification and management decisions.

Assessment

What is the main purpose of applying clustering techniques in ecosystem pattern analysis?

A. Predict future environmental variables
B. Assign samples to predefined risk classes
C. Identify natural groupings and patterns in unlabeled environmental data
D. Reduce dimensionality for visualization only

Which limitation is most characteristic of the K-Means clustering algorithm in environmental datasets?

A. Inability to handle large datasets
B. Requirement to predefine the number of clusters (k)
C. Inability to work with numerical variables
D. Inability to detect compact clusters

Which clustering method is best suited for detecting pollution hotspots of irregular shape while identifying noise?

A. K-Means
B. Hierarchical clustering
C. DBSCAN
D. Logistic Regression

Why are internal validation metrics such as the Silhouette Score commonly used in clustering analysis?

A. Because clustering always has ground-truth labels
B. To predict future ecosystem changes
C. To evaluate cluster compactness and separation without labeled data
D. To optimize supervised classifiers

How can clustering results support ecosystem management and conservation decisions?

A. By replacing ecological field observations
B. By providing exact forecasts of ecosystem change
C. By revealing functional groups and ecological gradients relevant for management
D. By eliminating uncertainty in environmental data

Additional Sources

Machine Learning - A First Course for Engineers and Scientists, Andreas Lindholm, Niklas Wahlström, Fredrik Lindsten, and Thomas B. Schön

Deep Learning with Python, Second Edition, François Chollet

Applications of Deep Neural Networks with Keras, Jeff Heaton

Understanding Machine Learning: From Theory to Algorithms, Shai Shalev-Shwartz and Shai Ben-David

Bibliography

[1] Blackard, J. (1998). Covertype [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C50K5N.

[2] Coops, N. C., Kearney, S. P., Bolton, D. K., & Radeloff, V. C. (2018). Remotely-sensed productivity clusters capture global biodiversity patterns. Scientific Reports, 8, 16261.

[3] Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD) (pp. 226–231).

[4] Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. ACM Computing Surveys, 31 (3), 264–323.

[5] Jehn, F. U., Bestian, K., Breuer, L., Kraft, P., & Houska, T. (2020). Using hydrological and climatic catchment clusters to explore drivers of catchment behavior. Hydrology and Earth System Sciences, 24(3), 1081–1100.

[6] Legendre, P., & Legendre, L. (2012). Numerical Ecology (3rd ed.). Elsevier.

[7] Shetty, C., Seema, S., Sowmya, B. J., Nandalike, R, Supreeth, S., Dayananda, P., Rohith, S., Ranjan, R., & Goud, V. (2024). A Machine Learning Approach for Environmental Assessment on Air Quality and Mitigation Strategy. Journal of Engineering, 2024(1), 2893021.

[8] Tan, P.-N., Steinbach, M., Karpatne, A., & Kumar, V. (2018). Introduction to Data Mining (2nd ed.). Pearson.

[9] Zou, K. (2022). The Dynamic Detection of Water Quality Monitoring and Pollution Prevention and Control. In Journal of Physics: Conference Series (Vol. 2152, No. 1, p. 012017). IOP Publishing.