Unsupervised Learning Models

Unsupervised learning is used when we don't have labeled data. The algorithm identifies patterns, clusters, or associations within the data without predefined categories.

Here are the most common **unsupervised learning models** with **when to use them** and **code examples**:

1. K-Means Clustering

👉 **When to Use?**

- When you want to group similar data points together.
- Useful when you don't know the categories beforehand.
- Examples: Customer segmentation, image compression, document clustering.

Python Code Example

```
from sklearn.cluster import KMeans

import numpy as np

# Sample dataset

X = np.array([[1, 2], [1, 4], [1, 0],

              [4, 2], [4, 4], [4, 0]])

# Applying K-Means with 2 clusters

kmeans = KMeans(n_clusters=2, random_state=0, n_init=10)

kmeans.fit(X)
```

```python
print("Cluster Centers:\n", kmeans.cluster_centers_)

print("Labels:", kmeans.labels_)
```

2. Hierarchical Clustering

👉 **When to Use?**

- When you want a **tree-like structure** of data grouping (dendrogram).
- Works well for small datasets where you want to analyze how clusters are formed step by step.
- Examples: Gene expression analysis, social network analysis.

Python Code Example

```python
import numpy as np

import scipy.cluster.hierarchy as sch

import matplotlib.pyplot as plt

from sklearn.cluster import AgglomerativeClustering


# Sample dataset

X = np.array([[1, 2], [1, 4], [1, 0],

              [4, 2], [4, 4], [4, 0]])


# Plot dendrogram
```

```
plt.figure(figsize=(5, 3))

dendrogram = sch.dendrogram(sch.linkage(X, method='ward'))

plt.show()

# Applying Hierarchical Clustering

hc = AgglomerativeClustering(n_clusters=2, affinity='euclidean',
linkage='ward')

y_hc = hc.fit_predict(X)

print("Cluster Labels:", y_hc)
```

3. Principal Component Analysis (PCA)

👉 **When to Use?**

- When you have **high-dimensional data** and want to reduce its complexity.
- Used for **feature reduction** while preserving essential information.
- Examples: Image compression, noise reduction, speeding up ML models.

Python Code Example

```
from sklearn.decomposition import PCA

import numpy as np

# Sample dataset (3 features)

X = np.array([[2.5, 3.5, 4.0], [3.2, 4.5, 5.1],

              [1.8, 2.8, 3.2], [2.9, 3.9, 4.4]])
```

```
# Reducing dimensions to 2

pca = PCA(n_components=2)

X_reduced = pca.fit_transform(X)


print("Reduced Data:\n", X_reduced)

print("Explained Variance Ratio:", pca.explained_variance_ratio_)
```

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

👉 **When to Use?**

- When data has **arbitrary shapes** and noise.
- Unlike K-Means, it does **not** require predefining the number of clusters.
- Examples: Anomaly detection, customer segmentation with varying densities.

Python Code Example

```
from sklearn.cluster import DBSCAN

import numpy as np

# Sample dataset

X = np.array([[1, 2], [2, 3], [2, 2],

              [8, 8], [9, 8], [8, 9]])
```

```
# Applying DBSCAN

dbscan = DBSCAN(eps=1.5, min_samples=2)

clusters = dbscan.fit_predict(X)


print("Cluster Labels:", clusters)
```

Summary of When to Use Each Model

Model	When to Use?
K-Means Clustering	When you need to **group similar items** into predefined clusters. Works well when the number of clusters is known.
Hierarchical Clustering	When you need to **visually understand cluster formation** (dendrogram). Works well for smaller datasets.
PCA (Principal Component Analysis)	When you want to **reduce dimensionality** while keeping important features. Good for speeding up models.
DBSCAN	When the data has **irregular shapes** and varying densities. Best for **anomaly detection** and **data with noise**.