Unsupervised Learning Guide

## What is Unsupervised Learning?

\*\*Definition\*\*: Unsupervised Learning is a type of machine learning where the model identifies patterns in data without labeled outcomes.

\*\*Analogy\*\*: Like a person organizing files without knowing their labels—grouping based on similarities.

```python  
# Example: clustering with KMeans  
from sklearn.cluster import KMeans  
model = KMeans(n\_clusters=3)  
model.fit(X)  
```

## Why Unsupervised Learning?

Used when labeled data is unavailable or expensive to obtain.

Ideal for pattern discovery, customer segmentation, and anomaly detection.

## Key Concepts

- \*\*Clustering\*\*: Grouping similar data points together (e.g., K-Means, DBSCAN).

- \*\*Dimensionality Reduction\*\*: Reducing features while preserving variance (e.g., PCA, t-SNE).

- \*\*Association Rule Mining\*\*: Discovering interesting relations (e.g., Apriori).

## Clustering Techniques

- \*\*K-Means\*\*: Partitions data into k clusters based on proximity.

- \*\*DBSCAN\*\*: Groups based on density, useful for noisy data.

- \*\*Hierarchical Clustering\*\*: Builds a tree of clusters (dendrogram).

```python  
from sklearn.cluster import KMeans  
kmeans = KMeans(n\_clusters=3)  
kmeans.fit(X)  
```

## Dimensionality Reduction

- \*\*PCA\*\*: Projects data into fewer dimensions by preserving variance.

- \*\*t-SNE\*\*: For visualizing high-dimensional data in 2D/3D.

```python  
from sklearn.decomposition import PCA  
pca = PCA(n\_components=2)  
X\_reduced = pca.fit\_transform(X)  
```

## Association Rule Mining

- Finds frequent itemsets and strong rules (used in market basket analysis).

```python  
from mlxtend.frequent\_patterns import apriori  
frequent = apriori(df, min\_support=0.1, use\_colnames=True)  
```

## Applications of Unsupervised Learning

- Customer segmentation in marketing.

- Anomaly detection in fraud or network security.

- Recommendation systems.

- Document/topic clustering.

- Image compression and recognition.

## Model Evaluation Techniques

- \*\*Silhouette Score\*\*: Measures cluster quality.

- \*\*Elbow Method\*\*: Helps choose number of clusters.

```python  
from sklearn.metrics import silhouette\_score  
score = silhouette\_score(X, kmeans.labels\_)  
```

## Challenges

- No ground truth for validation.

- Interpreting clusters may be subjective.

- Scaling and preprocessing often critical.

## Best Practices

- Visualize clusters for insight.

- Try multiple clustering algorithms.

- Normalize and scale data beforehand.

- Use dimensionality reduction for large feature sets.

## Common Interview Questions

- How does K-Means clustering work?

- What is the difference between PCA and t-SNE?

- How do you evaluate clustering performance?

- Give real-world use cases of unsupervised learning.