PODS Lab 14: Unsupervised Learning – Dimensionality Reduction and Clustering

Hamza Alshamy

Center for Data Science, NYU
ha2486@nyu.edu

05/02/25

1. Dimensionality Reduction (PCA)
2. Clustering
    - K-Means
    - DBSCAN

Reminder: Supervised vs Unsupervised Learning

Two (Major) Types of Machine Learning

1. **Supervised Learning:** Inputs (X) and label (y).
   - We train the model on the relationship between inputs and labels, then test and make predictions.
   - **Example:** Predicting house prices from square footage and number of bedrooms.
2. **Unsupervised Learning:** Only inputs (X).
   - Models learn patterns or groupings in the data without labeled outputs.
   - **Example:** Segmenting customers into distinct groups based on purchasing behavior (clustering).

Unsupervised Learning

1. **Dimensionality Reduction**: Transform high-dimensional data into a lower-dimensional space while preserving important structure.
   - Principal Component Analysis (PCA)
2. **Clustering**: Group similar data points together based on structure or proximity in feature space.
   - K-Means
   - Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

Dimensionality Reduction:

Motivation

- **Dimensionality Reduction:** technique used to simplify complex datasets by reducing the number of features (dimensions) while preserving essential information
  - $\mathbb{R}^{12} \rightarrow \mathbb{R}^3$

Dimensionality Reduction:

Motivation

- **Dimensionality Reduction:** technique used to simplify complex datasets by reducing the number of features (dimensions) while preserving essential information
  - $\mathbb{R}^{12} \to \mathbb{R}^3$
- **High-dimensional data** often contains redundant or irrelevant features.
- ***Multivariable* datasets** (many features) are harder to visualize, interpret, and model.
  - **Note:** *Multivariate* refers to multiple dependent variables

Dimensionality Reduction:

Motivation

- **Dimensionality Reduction:** technique used to simplify complex datasets by reducing the number of features (dimensions) while preserving essential information
    - $\mathbb{R}^{12} \to \mathbb{R}^3$
- **High-dimensional data** often contains redundant or irrelevant features.
- ***Multivariable* datasets** (many features) are harder to visualize, interpret, and model.
    - **Note:** *Multivariate* refers to multiple dependent variables
- **Dimensionality reduction helps us**:
    - Simplify data for **visualization** (e.g., 2D/3D plots).
    - **Improve model performance** by removing noise and redundancy.
    - **Prevent overfitting** in predictive models.
    - **Speed up** computation and training.

Dimensionality Reduction:

Principal Component Analysis (PCA)

- **Goal:** Maximize the variance captured by the principal components to retain as much information as possible while reducing dimensionality.

Dimensionality Reduction:

Principal Component Analysis (PCA)

- **Goal:** Maximize the variance captured by the principal components to retain as much information as possible while reducing dimensionality.
- PCA is a **linear transformation** technique.
- It identifies **uncorrelated components** (not necessarily statistically independent) that capture the most variance in the data.
- It is commonly used on **multivariable datasets** with many features.
- **Output:** As many components as there are original features, ranked by explained variance.

Why is PCA a Linear Transformation?

- PCA transforms the original data using **linear combinations** of the features.
- Each principal component is a projection:

$$z_i = \mathbf{w}^\top \mathbf{x} = w_1 x_1 + w_2 x_2 + \cdots + w_D x_D$$

- The full transformation (matrix form) is:

$$\underbrace{Z}_{n\times d} = \underbrace{X}_{n\times D} \underbrace{W}_{D\times d}$$

  - $X \in \mathbb{R}^{n\times D}$: centered data matrix (each row is an observation)
  - $W \in \mathbb{R}^{D\times d}$: matrix of top d eigenvectors (principal components)
  - $Z \in \mathbb{R}^{n\times d}$: lower-dimensional representation
- **Linear** means no nonlinear operations (e.g., squaring, exponentiating).
- PCA preserves linear structure but cannot capture nonlinear patterns.

Principal Component Analysis (PCA): Logic

- PCA finds an **orthonormal basis** and projects the data onto it.
    - **Orthonormal:** The vectors are both *orthogonal* (perpendicular) and of *unit length*.
- This is done by computing the **covariance matrix** and performing **eigen decomposition**.
- **Eigenvectors:** Define the new basis directions (principal components).
- **Eigenvalues:** Indicate how much variance is captured along each eigenvector.

PCA Example: Wine Dataset

- **Dataset:** Each row represents a wine sample.
- **13 features** per sample (e.g., Alcohol, Flavonoids, Color Intensity).
- **No outcome variable:** This is a fully unsupervised setting.

Alcohol	Malic Acid	...	Color Intensity	Proline
14.23	1.71	...	5.64	1065
13.20	1.78	...	4.38	1050
13.16	2.36	...	5.68	1185
14.37	1.95	...	7.80	1480
13.24	2.59	...	4.32	1195

- **Goal:** Reduce the data to a lower-dimensional space that captures the most variance, enabling visualization and pattern discovery.

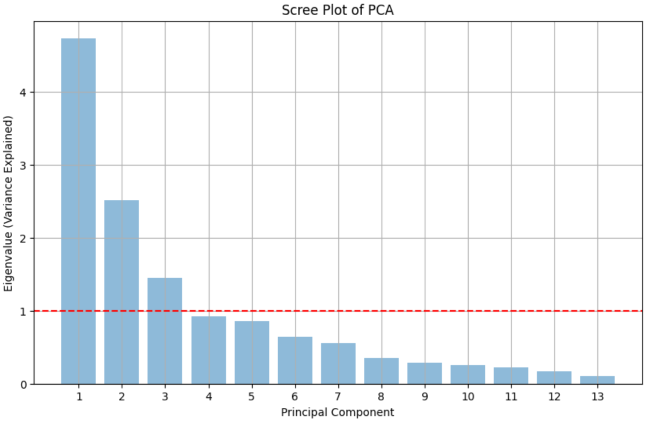
Principal Component Analysis (PCA):

Determining the Number of Principal Components (PCs)

1. **Elbow Criterion:** Identify the point where the explained variance starts to level off (the "elbow") in the scree plot.
2. **Kaiser Criterion:** Retain all components with eigenvalues greater than 1.
3. **Variance Threshold:** Select the smallest number of PCs that cumulatively explain a desired amount of variance (e.g., 90%).

Principal Component Analysis (PCA):

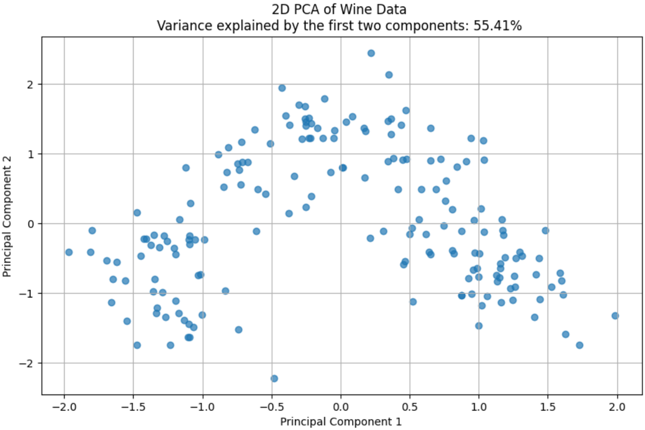
Determining the Number of Principal Components (PCs)



- **Elbow:** Retain first two (?) components (variance begins to flatten).
- **Kaiser:** Retain the first three components

From 13 Features to 2 Components:

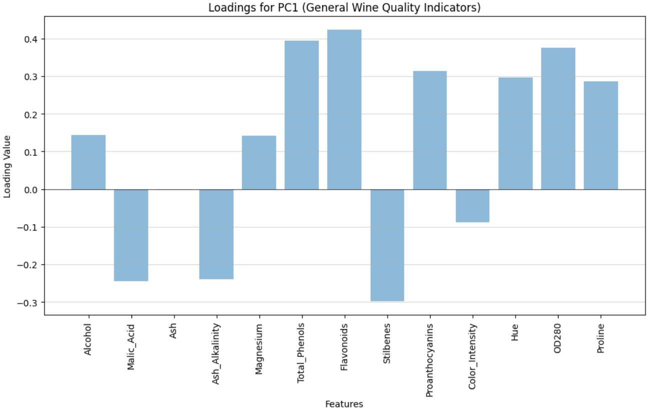
Visualization



- 2D projection captures **55.41% of total variance**.
- Structure in the plot suggests potential groupings or patterns.

Principal Component Analysis (PCA):

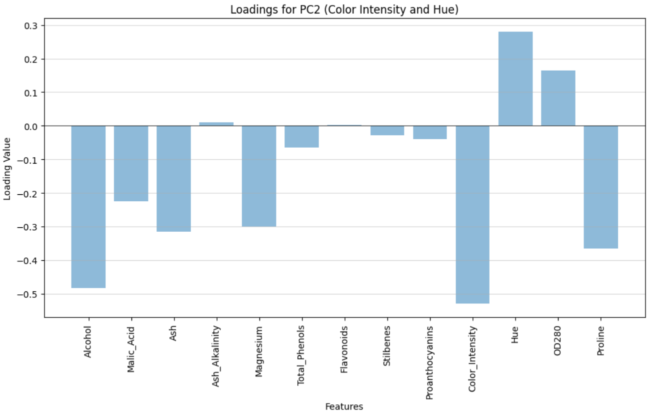
How to Interpret the Principal Components – Loading Matrix



- **Loadings** = feature contributions to PC1
- **High absolute value** → strong influence
- **Positive/Negative loadings** → increase/decrease PC1

Principle Component Analysis (PCA):

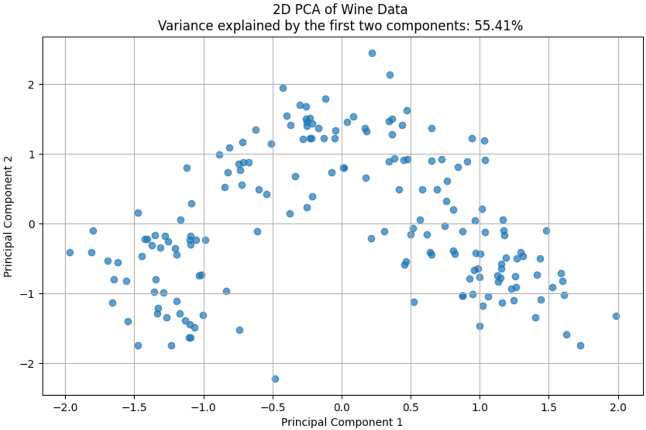
How to Interpret the Principle Components? – Loading Matrix



- PC2 contrasts wines high in **color intensity** with those high in **hue**
- Likely reflects a color-related axis – e.g., darker vs. lighter wines.

From 13 Features to 2 Components:

Visualization



- Structure in the plot suggests potential groupings or patterns.
- Notice any clusters?

Clustering:

Motivation

- Clustering techniques are essential for identifying distinct groups or clusters within a dataset

Clustering:

Motivation

- Clustering techniques are essential for identifying distinct groups or clusters within a dataset
- **Goal:** Ensure that the points within each cluster are similar to each other but dissimilar from points in other clusters

Clustering:

Motivation

- Clustering techniques are essential for identifying distinct groups or clusters within a dataset
- **Goal:** Ensure that the points within each cluster are similar to each other but dissimilar from points in other clusters
- This helps reveal hidden structure and natural groupings in the data.
- **Examples:**
  - categorizing customers based on shopping behavior
  - segmenting geographic areas based on land use.

Clustering:

Motivation

- Clustering techniques are essential for identifying distinct groups or clusters within a dataset
- **Goal:** Ensure that the points within each cluster are similar to each other but dissimilar from points in other clusters
- This helps reveal hidden structure and natural groupings in the data.
- **Examples:**
  - categorizing customers based on shopping behavior
  - segmenting geographic areas based on land use.

Two (amongst many) Types of Clustering:

1. K-means
2. Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

K-Means

- **Idea:** Group data points into K clusters.
- **Goal**: Find cluster centers (centroids) that **minimize distortion** – the total squared distance between each point and its nearest centroid.

Distortion (Minimizing within-cluster variance)

$$\text{Distortion} = \sum_{k=1}^{K} \sum_{x \in S_k} \|x - \mu_k\|^2$$

where x is a data point, S_k is the set of data points in cluster k, and μ_k is the centroid of cluster k.

K-Means

- **Idea:** Group data points into K clusters.
- **Goal**: Find cluster centers (centroids) that **minimize distortion** – the total squared distance between each point and its nearest centroid.

Distortion (Minimizing within-cluster variance)

$$\text{Distortion} = \sum_{k=1}^{K} \sum_{x \in S_k} \|x - \mu_k\|^2$$

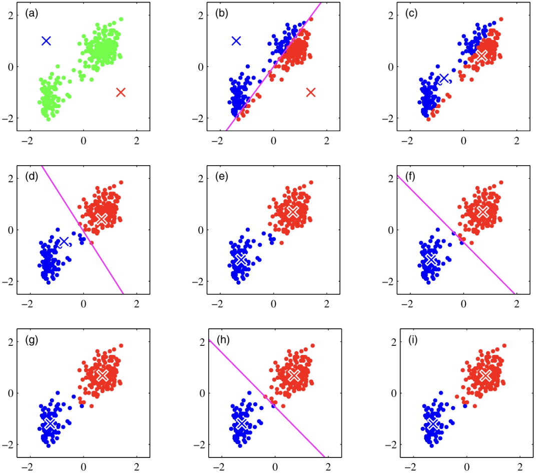
where x is a data point, S_k is the set of data points in cluster k, and μ_k is the centroid of cluster k. **Two-Step Algorithm (iterative):**

1. **Assignment Step:** Assign each point to the nearest centroid.
2. **Update Step:** Recalculate each centroid as the mean of its assigned points.

Note: The data does *not* move, the centroids do!

K-Means: Algorithm

1. Random initialization of centroids (X, X)
2. Assign each point to the nearest centroid
3. Recalculate each centroid as the mean of its assigned points
4. Repeat until assignments no longer change



From *Pattern Recognition and Machine Learning* by Christopher Bishop, p. 426 (PDF)

K-Means:

How to Choose the Optimal number of Clusters?

- K-means requires the number of clusters (K) to be predefined

K-Means:

How to Choose the Optimal number of Clusters?

- K-means requires the number of clusters (K) to be predefined

Two Ways to Choose the Number of Clusters

1. **Elbow Method:** Plotting the total within-cluster distances for various values of K and selecting the point where the decrease in distance becomes negligible, forming an "elbow"

K-Means:

How to Choose the Optimal number of Clusters?

- K-means requires the number of clusters (K) to be predefined

Two Ways to Choose the Number of Clusters

1. **Elbow Method:** Plotting the total within-cluster distances for various values of K and selecting the point where the decrease in distance becomes negligible, forming an "elbow"
2. **Silhouette Method:** Compute the *Silhouette Coefficient* for each point:

Silhouette Coefficient

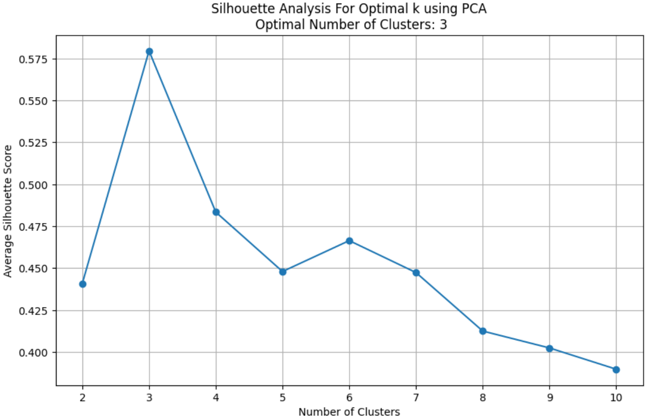
For a data point i, the silhouette coefficient is:

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

- Values near +1 indicate that the point is well-placed within its cluster.

K-Means: Wine Dataset

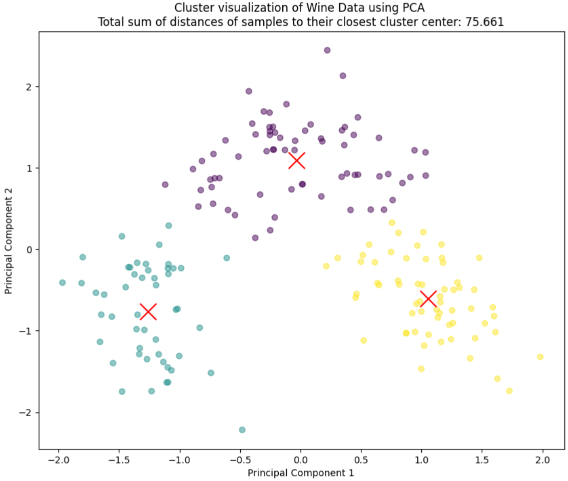
Silhouette Score for Optimal Number of Clusters



- Highest silhouette score occurs at $K = 3$, suggesting this is the optimal number of clusters.

K-Means

Wine Dataset Example (2D PCA Projection)



- **Final result:** $K = 3$ clusters with minimal within-cluster distortion

Assessment of K-means

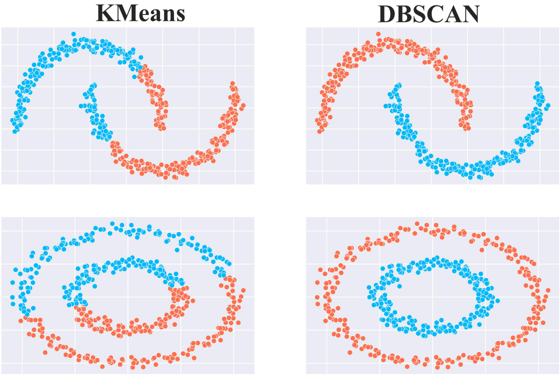
Pros:

1. Simple
2. Fast
3. (Usually) converges to a solution

Cons

1. **Cluster Shape Assumption:** K-means assumes clusters are spherical and evenly sized, which may not hold in real-world datasets
2. Sensitive to outliers
3. Includes noise into clusters

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



- **K-Means fails** on non-spherical or nested clusters.
- **DBSCAN** handles complex shapes and identifies noise points.

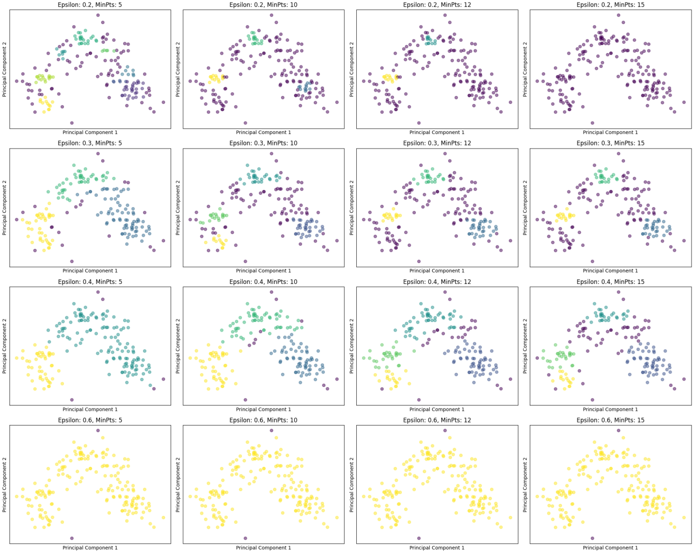
DBSCAN: Framework (Click on LINK to visualize).

- DBSCAN's approach, based on density rather than distance, allows it to discover clusters with arbitrary shapes.

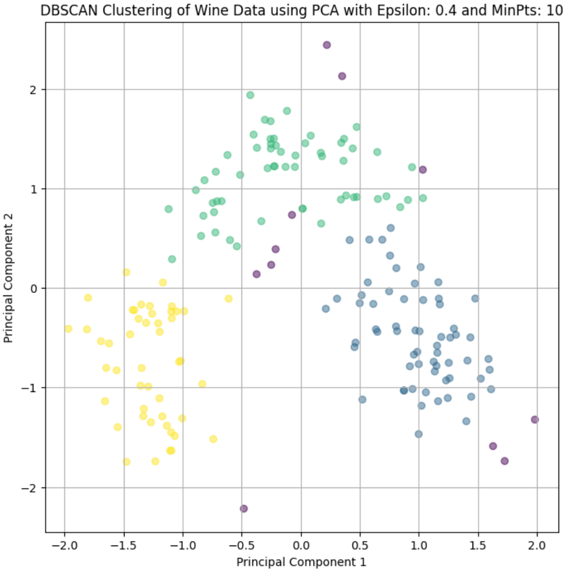
Two Hyperparameters:

1. **Epsilon (ε):** Defines the radius of neighborhoods around each data point
   - A *higher* ε might merge distinct clusters, leading to fewer, larger clusters.
   - A *lower* ε might result in not capturing all relevant points in a cluster, increasing the number of clusters
2. **MinPts (Minimum Points):** Specifies the minimum number of points required to form a dense region
   - A *higher MinPts* value increases the density required to form a cluster
   - A *lower MinPts* value decreases the density requirement,

DBSCAN on Wine Dataset: Effect of Epsilon and MinPts



DBSCAN on Wine Dataset with Ranges of Epsilon and MinPts


DBSCAN Clustering of Wine Data using PCA with Epsilon: 0.4 and MinPts: 10
Principal Component 2
Principal Component 1