Feature scaling
-
Importance in clustering

Week 6 – Advanced Topic 1

Objectives

- Ensure comparability between features.
- Accelerate model convergence.
- Better interpretation of model coefficients.
- Enable better interpretation.

Feature scaling

- Pattern recognition in data.
- Reduce complexity.
- Handle unlabeled data.
- Support decision-making systems.

Clustering

Theory

Feature scaling

- A preprocessing step for many maching learning algorithms.
- The range of features of data is normalized or standardized.
- To prevent any one feature from disproportionately affecting the result due to its larger numerical scale.

Common methods:

$$X_{scaled} = \frac{X_i - X_{min}}{X_{max} - X_{min}}$$

Min-Max Scaling
(Normalization)

$$X_{scaled} = \frac{X_i - X_{mean}}{\sigma}$$

Z-score normalization
(Standardization)

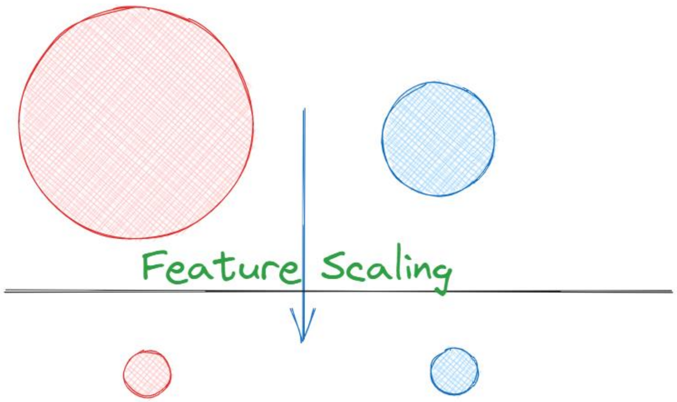

Fig.1 Feature scaling process.

(*https*://medium.com/@yashbaravaliya206/feature-scaling-and-normalization-ca484a16882a)

Theory

Clustering

- Unsupervised machine learning technique.
- Group unlabeled data based on their similarity.
- Several clustering methods.
- Used methods:
  - Kmeans
  - DBSCAN
  - Spectral

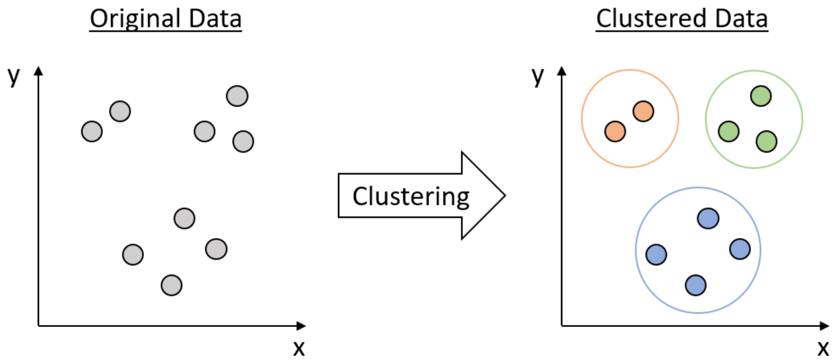

Fig.2 Clustering process.

(https://waterprogramming.wordpress.com/2022/03/16/clustering-basics-and-a-demonstration-in-clustering-infrastructure-pathways/)

Theory

Feature scaling and clustering

Distance-based clustering algorithms rely on calculating distances between the data points.

Feature scaling contributes in:

1. Equal contribution of features
2. Better distance computation
3. More accurate cluster assignments
4. Improved algorithm convergence

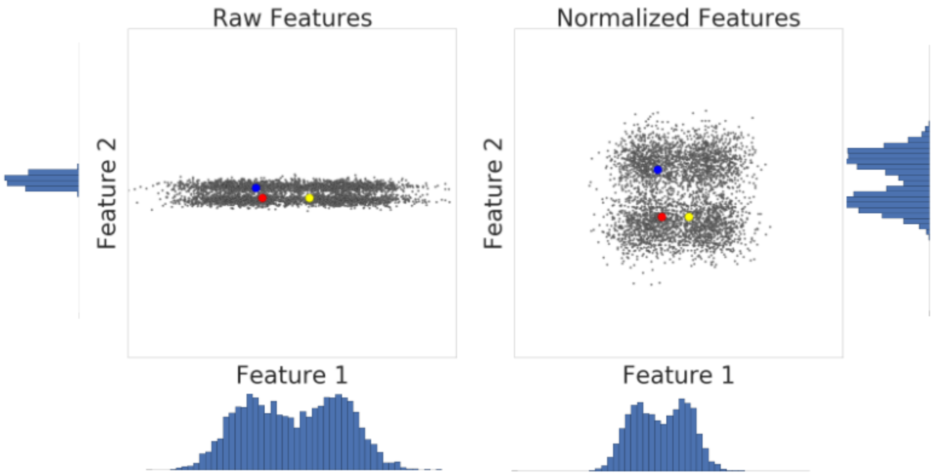

Fig.3 Feature scaling and clustering process.

https://developers.google.com/machine-learning/clustering/prepare-data

Methods

KMeans

- A method for vector quantization.
- Makes partitions of n observations into k clusters with the nearest mean.
- K-means clustering minimized within-cluster variances (squared Euclidian distance).

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x}\in S_i} \|\mathbf{x}-\boldsymbol{\mu}_i\|^2 = \arg\min_{\mathbf{S}} \sum_{i=1}^{k} |S_i| \operatorname{Var} S_i$$

Set of observations

Mean (Centroid)

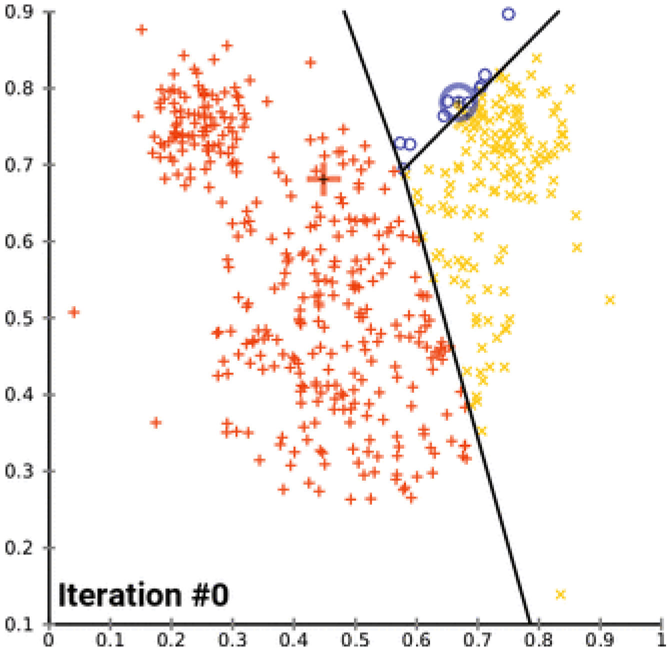

Fig.4 KMeans clustering.

Methods

DBSCAN

- Does not require one to specify the number of clusters.
- Can find arbitrarily-shaped clusters.
- Robust to outliers.
- It is designed for use with databases that can accelerate region queries, e.g. using an R* tree.

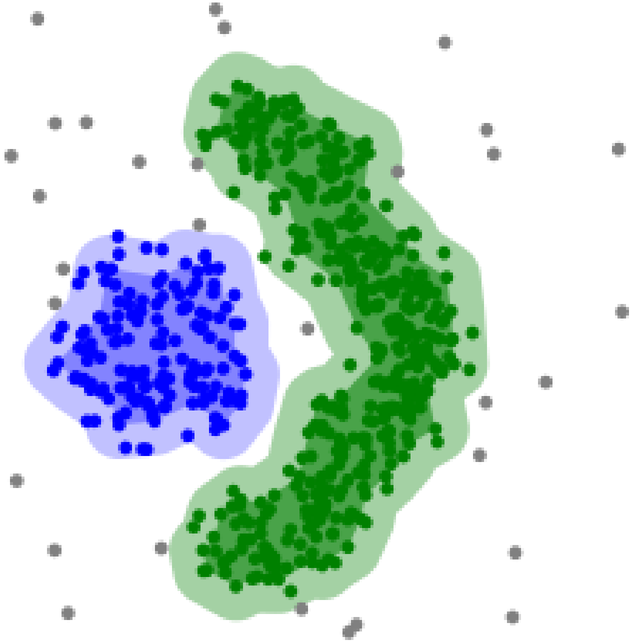

Fig.5 DBSCAN clustering.

https://en.wikipedia.org/wiki/DBSCAN

Methods

Spectral clustering

- Make use of the spectrum (eigenvalues) of the similarity matrix of the data.
- Perform dimensionality reduction before clustering in fewer dimension.
- Similarity matrix is a quantitative assessment of the relative similarity of each pair of points in the dataset.

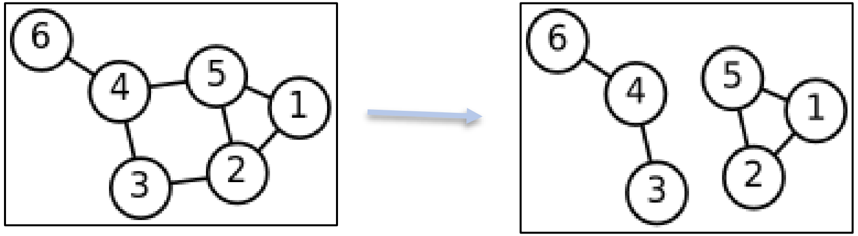

Fig.6 Spectral clustering.

https://en.wikipedia.org/wiki/Spectral_clustering

Methods

Raw dataset

- Wine dataset from scikit learn library.
- Heterogeneous values.
- Visualization based on two features (Hue- Proline).

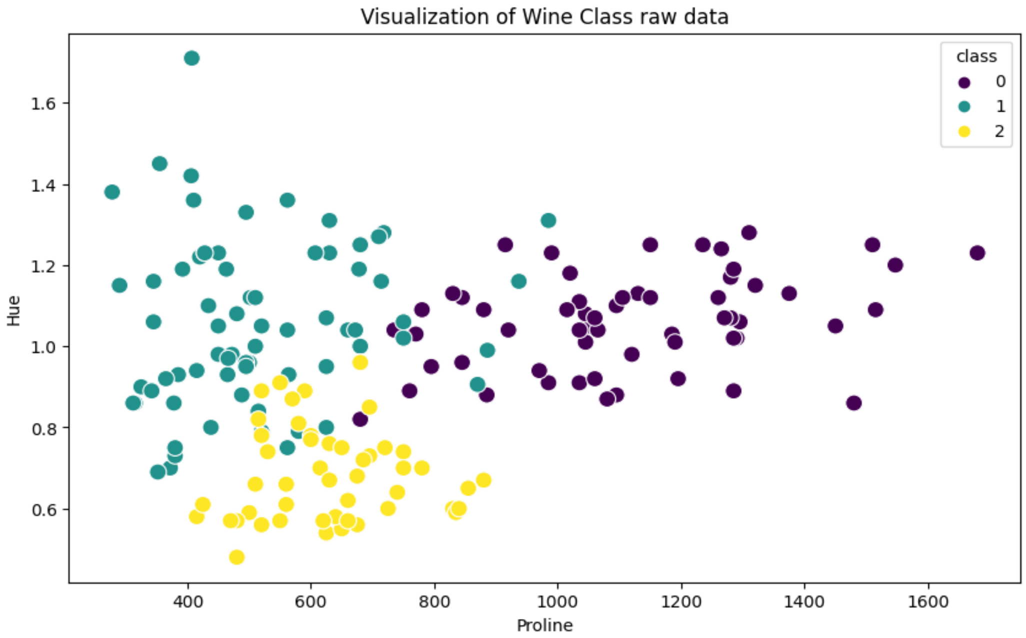

Fig.7 Wine raw dataset.

Results

KMeans

- Unscaled clustering mainly based on Proline
- Similar clustering for Standarized and MinMax scaled data

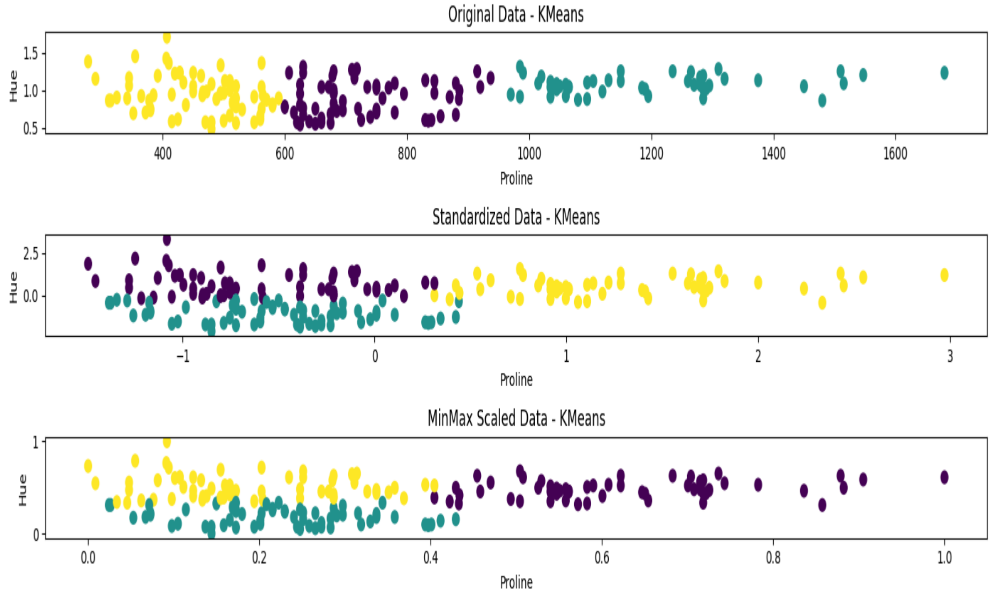

Fig.8 KMeans results from wine dataset.

Results

DBSCAN

- Poor clustering
- Hard to tune
- Different tuning for different scaling

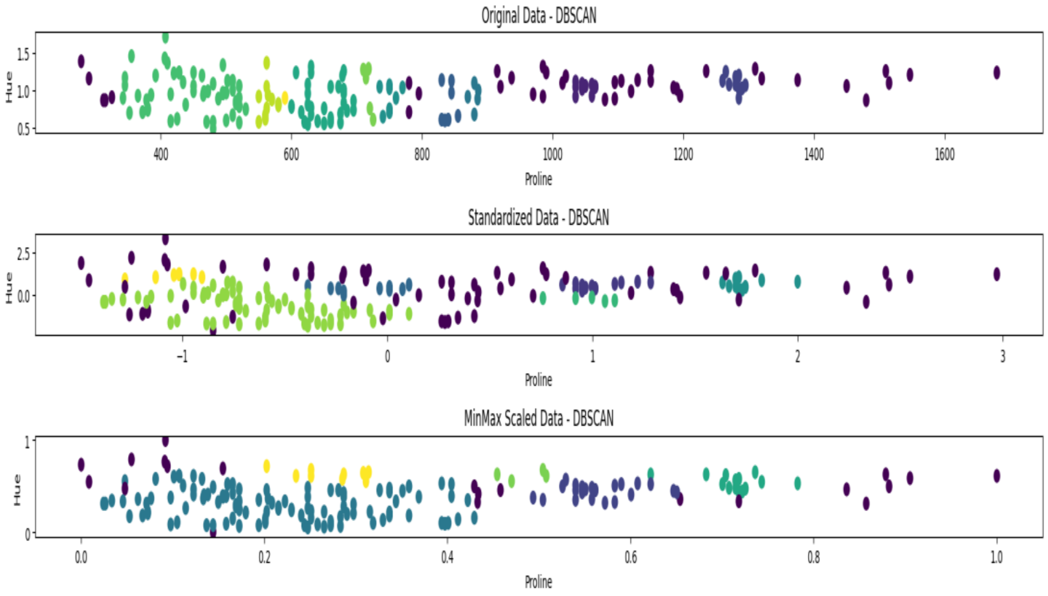

Fig.9 DBSCAN results from wine dataset.

NTNU

Results

Spectral clustering

- Unscaled clustering mainly based on Proline
- Similar clustering for standarized and MinMax scaled data

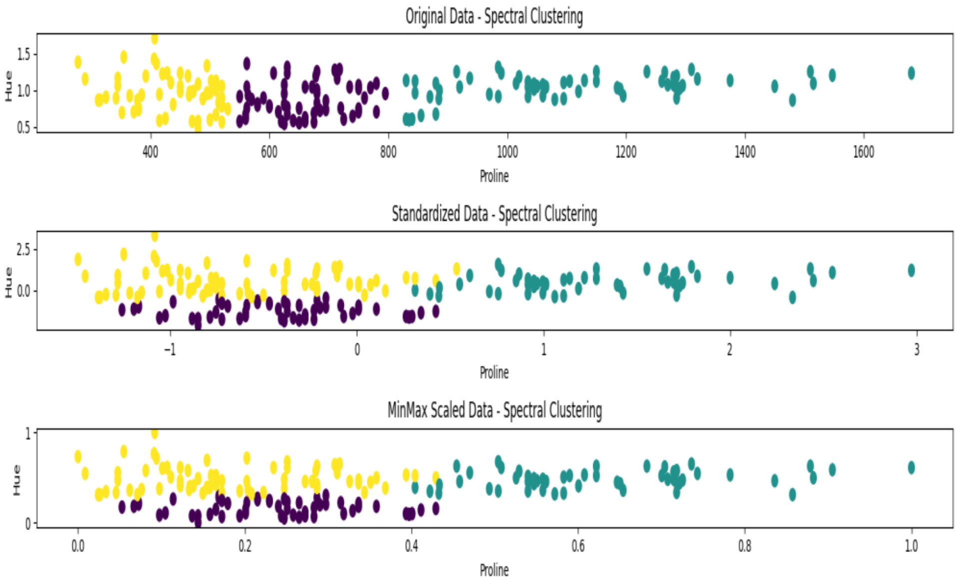

Fig.10 Spectral clustering results from wine dataset.

Conclusions

Feature scaling and clustering
- Scaling equalizes feature importance
- Improved algorithm performance
- Easier convergence for optimization

KMeans
- Number of clusters must be predefined
- Sensitive to initialization and scaling
- Works well for larger datasets

DBSCAN
- No predefined number of clusters
- Identifies arbitrary-shaped clusters
- Robust to noise

Spectral clustering
- Good for non-convex data
- Computationally expensive
- Handles complex structures
- Requires predefined number of clusters

Thank you!