A3: Unsupervised learning with PCA, t-SNE, k-means, AHC and SOM

Professors: SERGIO GÓMEZ JIMÉNEZ, JORDI DUCH GAVALDÀ
Students: SEBASTIAN BUZDUGAN

Table of Contents

Environment and Language:

Programming Language: Python 3.12
Development Environment: Jupyter Notebook in Visual Studio Code This setup provides an interactive environment that's ideal for data exploration, visualization, and running machine learning algorithms.

Core Libraries and Their Uses:
Pandas (pandas): Used for data manipulation and analysis. Ideal for working with structured data, like CSV or Excel files. Functions like reading data, data cleaning, and preprocessing are handled efficiently with Pandas.
NumPy (numpy): Provides support for large, multi-dimensional arrays and matrices. Offers a wide range of mathematical functions to operate on these arrays. Essential for numerical operations and transformations on data.
Matplotlib (matplotlib.pyplot): A comprehensive library for creating static, animated, and interactive visualizations in Python. Used for plotting a wide variety of graphs (like line, bar, scatter, histograms).
Seaborn (seaborn): Based on matplotlib, it provides a high-level interface for drawing attractive and informative statistical graphics. Used for more advanced visualizations, like heatmaps and pair plots.
Plotly Express (plotly.express): A high-level API for rapid data exploration and figure generation. Useful for interactive plots and advanced visualizations.

Machine Learning and Clustering Libraries:
Scikit-learn (sklearn): Provides simple and efficient tools for predictive data analysis. Used for clustering algorithms like KMeans, Agglomerative Clustering, Spectral Clustering, and Affinity Propagation. Also offers tools for data preprocessing (like StandardScaler), dimensionality reduction (like PCA), and model evaluation (like classification_report).
SciPy (scipy.cluster.hierarchy): Used for hierarchical clustering and generating dendrograms.
MiniSom (minisom.MiniSom): An implementation of the Self-Organizing Maps (SOM). Useful for unsupervised learning and data visualization.

Datasets

Dataset 1: Synthetic Dataset (A3-data.txt)

The Synthetic Dataset, named 'A3-data.txt', is composed of 4 variables and includes a class attribute. The dataset consists of 360 patterns, offering a rich field for applying unsupervised learning techniques. In this project, the class information is not used for the learning process itself but is crucial for identifying the classes in plots, aiding in the validation and interpretation of the clustering results. Preprocessing steps, such as normalization or standardization, might have been undertaken to prepare the data for effective clustering analysis.

Dataset 2: Dry Bean Dataset (Dry_Bean_Dataset.xlsx)

The Dry Bean Dataset, housed in 'Dry_Bean_Dataset.xlsx', features 16 distinct physical attributes across 13,611 patterns, categorized into 7 unique classes: 'SEKER', 'BARBUNYA', 'BOMBAY', 'CALI', 'HOROZ', 'SIRA', and 'DERMASON'. Primarily used in unsupervised learning, the class labels facilitate post-analysis validation and visualization of clustering outcomes. Standard preprocessing like normalization ensures balanced contribution from each feature, making this dataset ideal for in-depth unsupervised learning explorations.

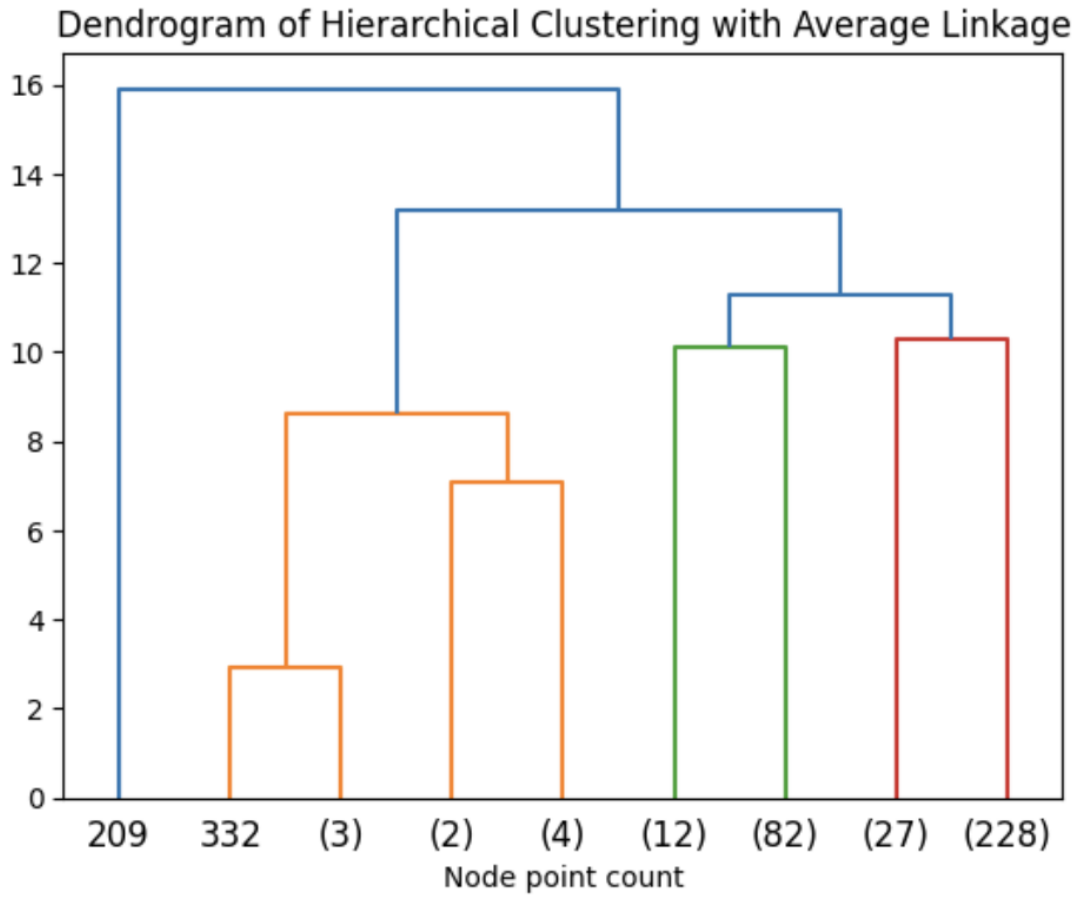
https://archive.ics.uci.edu/dataset/602/dry+bean+dataset

Comparing unsupervised learning algorithms

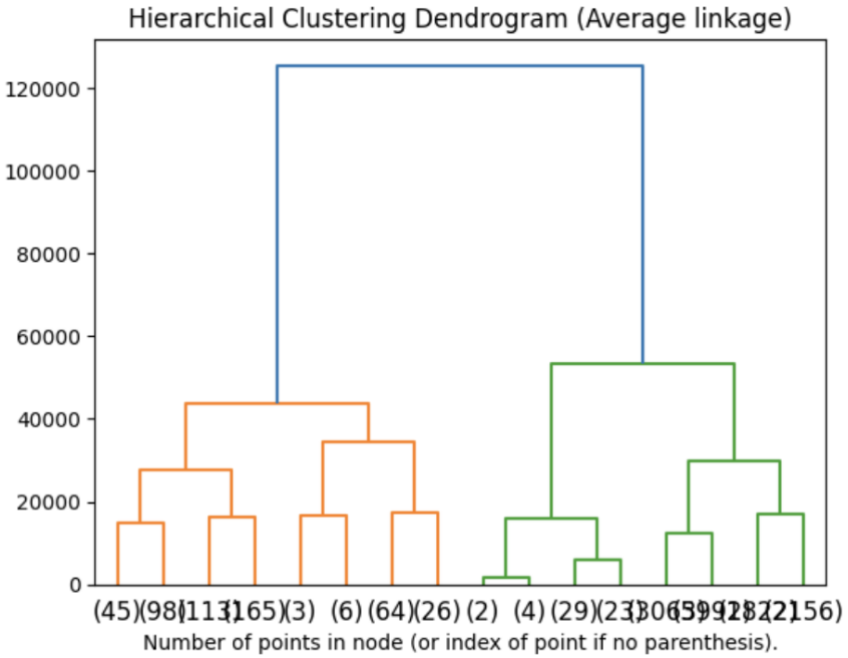
Agglomerative Hierarchical Clustering

AHC is a method of cluster analysis which seeks to build a hierarchy of clusters. It's a "bottom-up" approach: each observation starts as its own cluster, and pairs of clusters are merged as one moves up the hierarchy. This method builds a dendrogram, representing the nested levels of clusters, which aids in understanding the data's structure and deciding the number of clusters by cutting the dendrogram at a suitable level.

Dataset A3-data.txt plot for the average linkage:

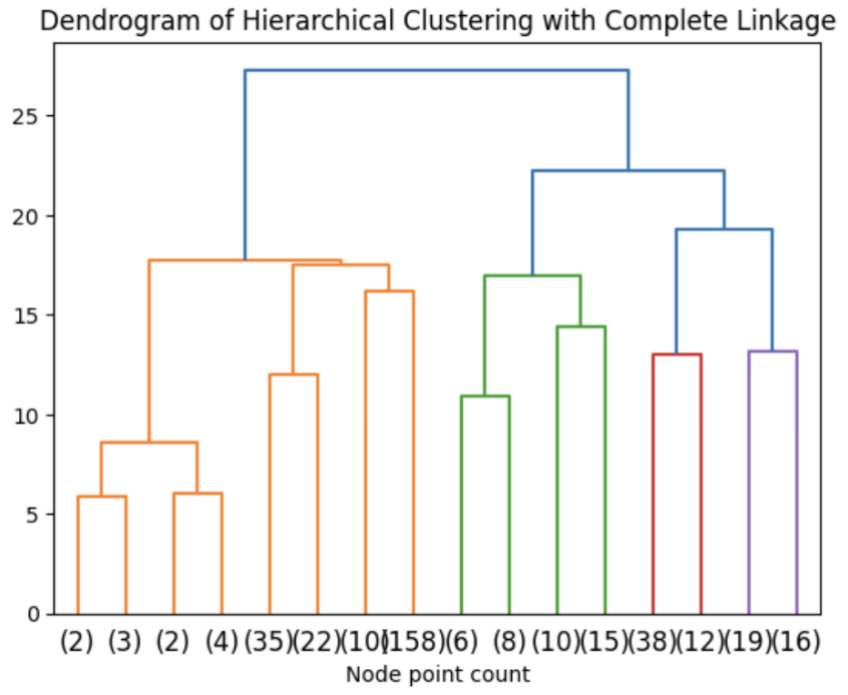


Dataset Dry_Bean_data for the average linkage:

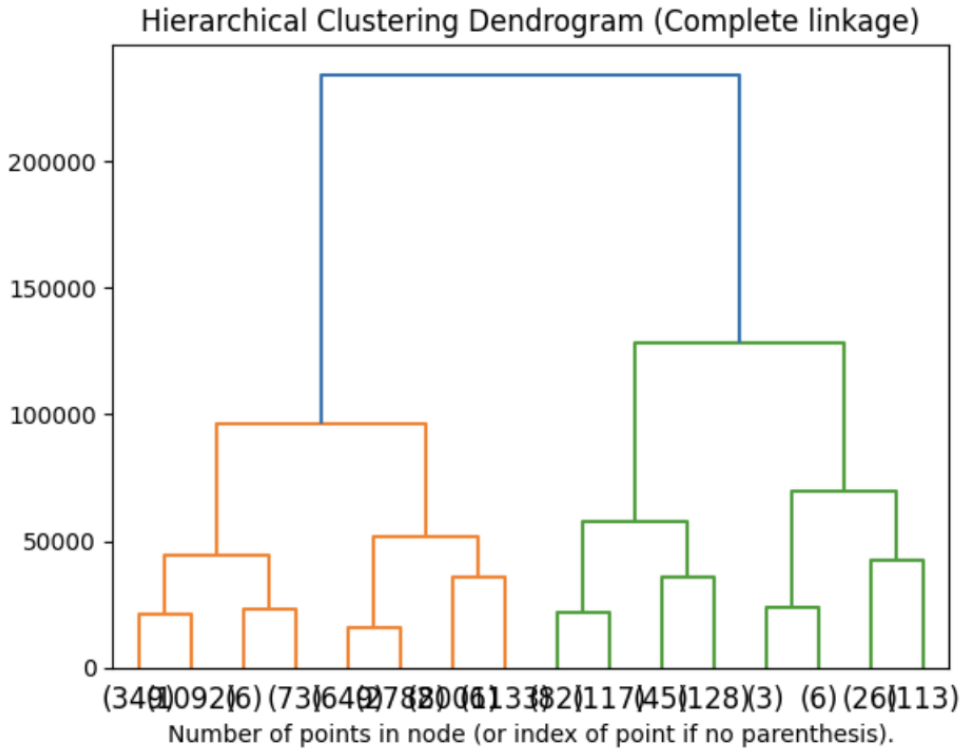


In AHC with average linkage, the Synthetic Dataset (A3-data) demonstrates uniform clustering with merge distances ranging from about 8.03 to 15.91, indicating closely related groups. In contrast, the Dry Bean Dataset exhibits a more complex pattern, with merge distances spanning from approximately 16,247 to 125,455, suggesting a diverse and heterogeneous cluster formation. This comparison highlights the Synthetic Dataset's relative homogeneity and the Dry Bean Dataset's intricate clustering complexity.

Dataset A3-data.txt plot for the complete linkage:



Dataset Dry_Bean_data for the complete linkage:

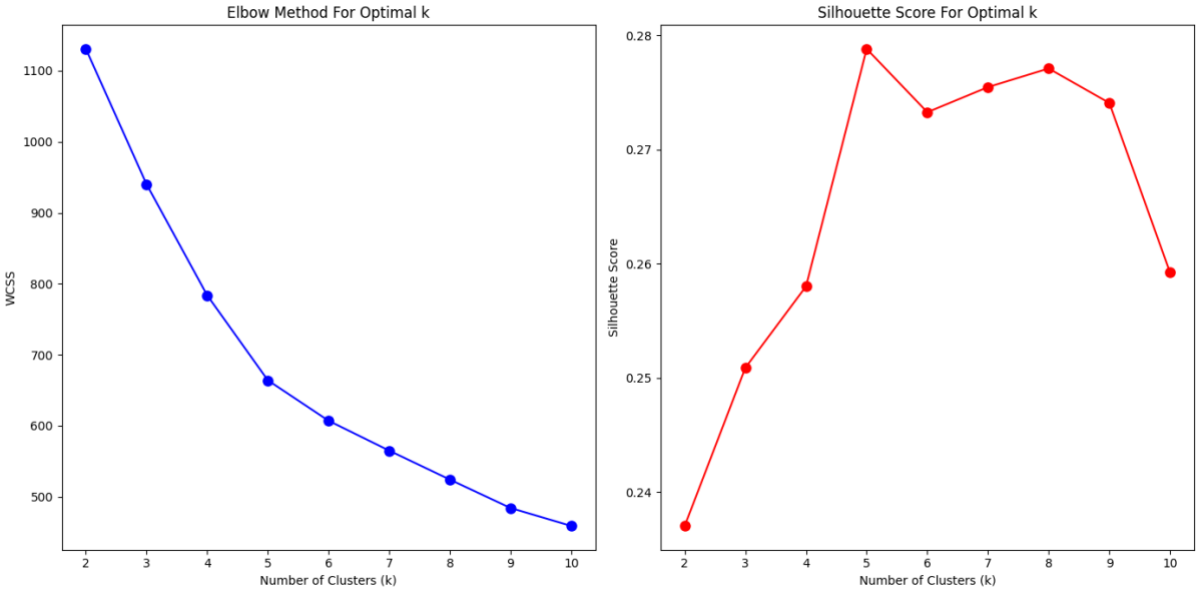


In Agglomerative Hierarchical Clustering with complete linkage, the Synthetic Dataset (A3-data) presents a relatively homogeneous clustering, evident from its merge distances ranging from about 13.15 to 27.31. This range suggests tighter groupings and more similarity within clusters. On the other hand, the Dry Bean Dataset displays a starkly different clustering behavior with merge distances extending from approximately 35,835 to 234,201. These wider merge distances underscore a highly varied and complex data structure, indicating a significant disparity within clusters. This juxtaposition of the Synthetic Dataset's cohesive clustering against the Dry Bean Dataset's diverse and intricate cluster formations accentuates the unique characteristics inherent in each dataset.

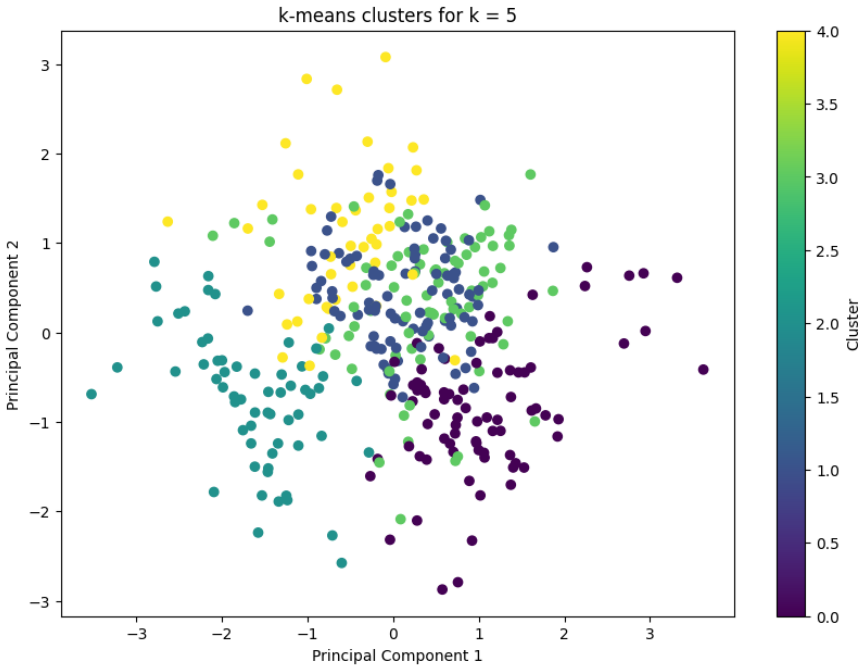
k-means

This algorithm is a popular unsupervised machine learning technique used for cluster analysis in data mining and statistics. It aims to partition a set of observations into 'k' clusters, where each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.

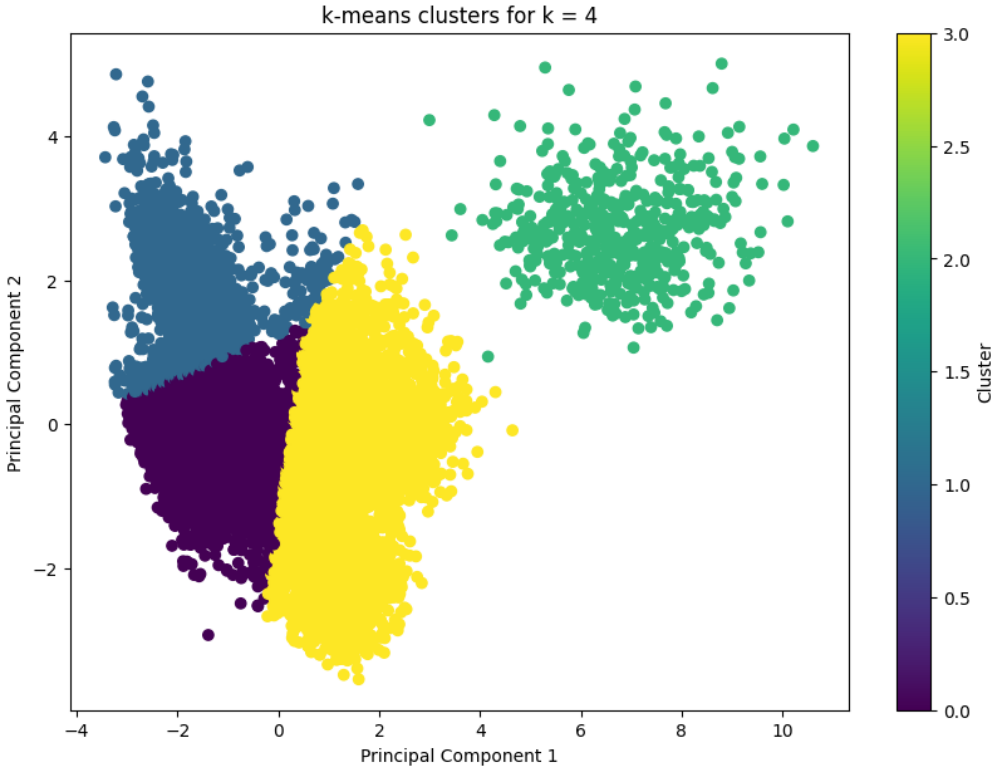
I have used 2 methods for both datasets to identify the best clusters value that I can take for k. For the A3 I have used k=5 while for Dry-Bean k=4.



A3-data Clusters (k=5): The first image, presumed to be the A3 dataset, shows a scatter plot with five different clusters. The clusters are color-coded and spread throughout the plot, suggesting that k-means has identified distinct groupings within the data. However, the clusters appear to overlap somewhat, particularly in the center of the plot where several clusters meet. This suggests that while there may be some structure to the data, the boundaries between some of the clusters are not very clear-cut. This could be a sign that some data points do not have strong membership to one particular cluster over another, or that the true underlying structure of the data is not perfectly captured by k-means with k=5.



Dry Bean Dataset Clusters (k=4): The second image is a scatter plot of what is labeled as the Dry Bean dataset with four distinct clusters. The clusters here are more well-defined than in the first image, with less overlap between them. This suggests that the Dry Bean dataset has a clearer structure, which is better captured by k-means clustering with k=4. The separation between clusters suggests that the data points within each cluster have more in common with each other than with points in other clusters. Additionally, the clusters are of varying densities and spread, which could reflect different subgroup sizes within the dataset and varying degrees of cohesion within each cluster.



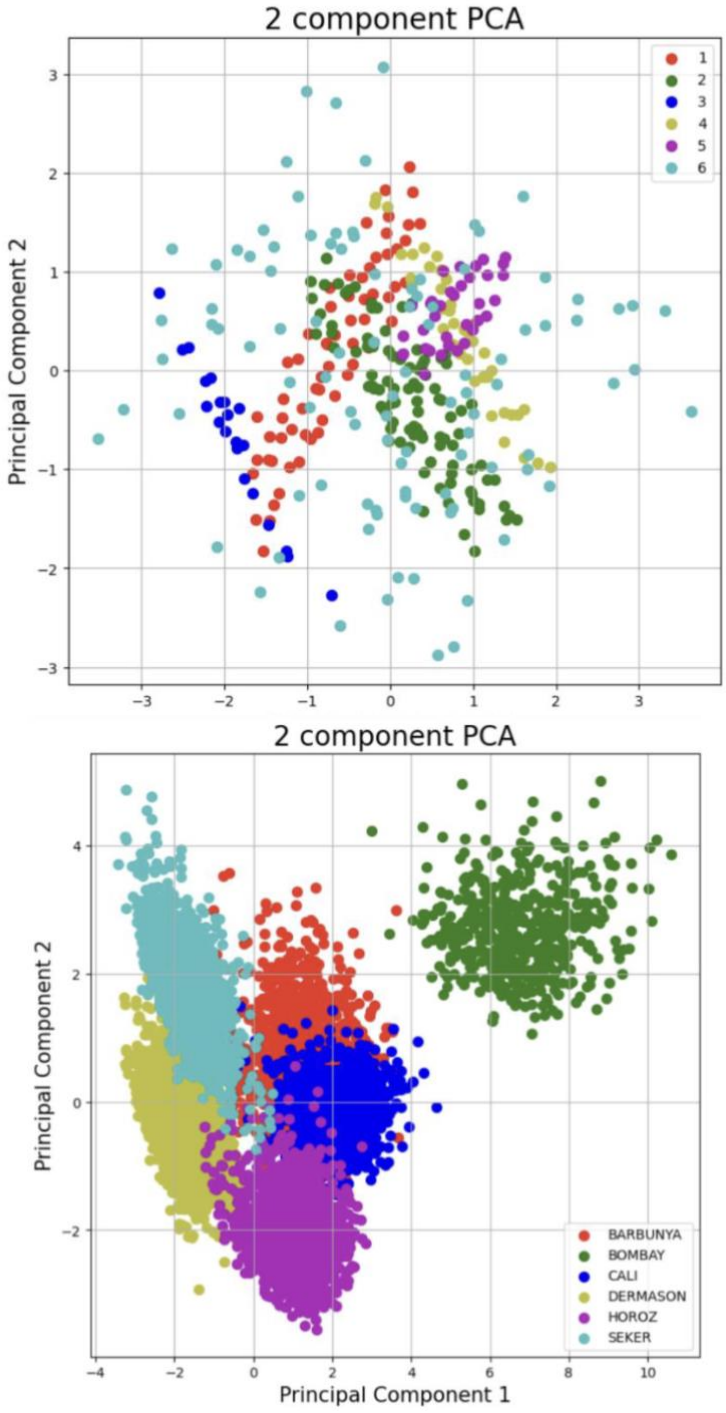
When comparing the K-means clustering results of the Dry Bean Dataset with those of the Synthetic A3 Dataset, several notable differences emerge. The Dry Bean Dataset below, characterized by its physically measured features such as 'Area' and 'Perimeter', exhibits centroids with larger values in a six-dimensional space, indicating a complex clustering pattern reflective of the diverse physical attributes of beans.

In contrast, the Synthetic A3 Dataset's centroids as can be seen in the figure below, spread across four dimensions, are smaller, suggesting a more normalized or abstract feature space. This difference in scale and dimensionality underlines the distinct nature of the datasets: the Dry Bean Dataset captures real-world physical variations in beans, while the Synthetic A3 Dataset seems to represent synthesized or categorical groupings. The clustering in the Dry Bean Dataset is likely identifying different bean types based on physical traits, whereas in the Synthetic A3 Dataset, it discerns patterns within a more conceptual feature space. Despite these differences, both datasets share a common aspect in the clustering process, where the number of clusters is determined by the unique classes present, providing a structured approach to uncovering inherent groupings in each dataset.

Principal Component Analysis

PCA is a statistical technique used in the field of machine learning and data analysis to emphasize variation and bring out strong patterns in a dataset. It's often used as a tool in exploratory data analysis and for making predictive models. PCA is commonly used for dimensionality reduction by transforming a large set of variables into a smaller one that still contains most of the information in the large set.

2D Projection of the A3-data vs Dry_Bean_Dataset:

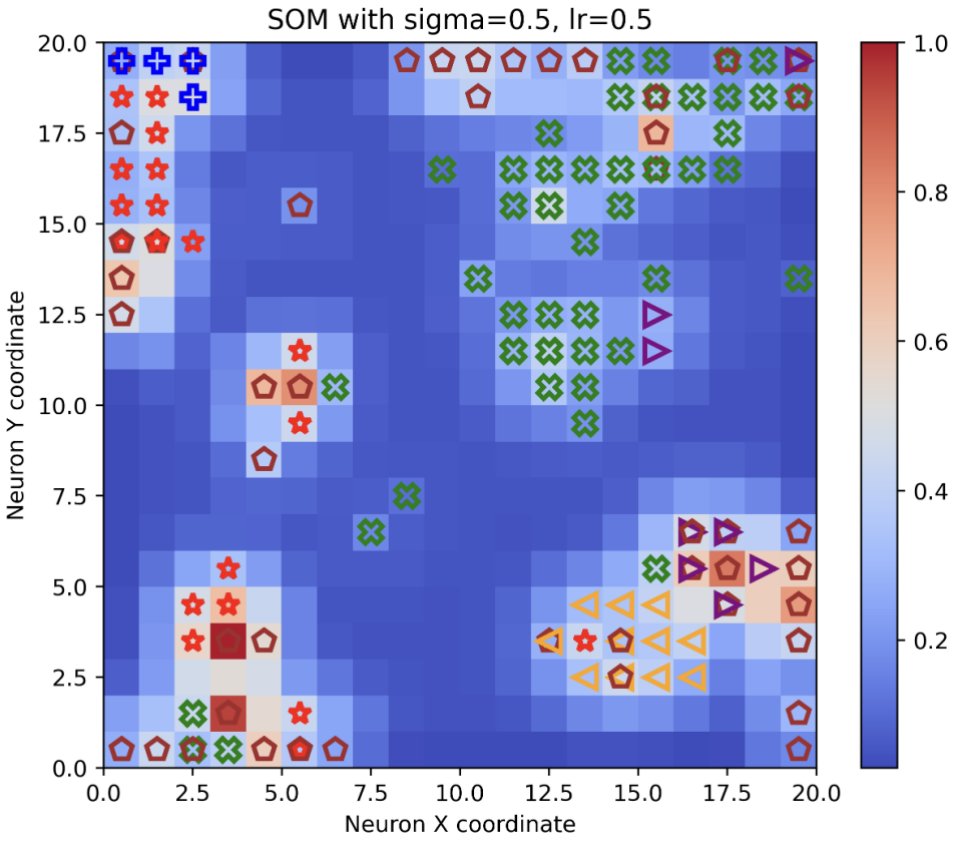


- **Dataset Complexity and Scale:** The Dry Bean Dataset (2nd one) is substantially larger and more complex, with a higher number of classes and a broader range of features compared to the Synthetic A3 Dataset (1st one).
- **PCA Distribution:** The spread of PCA values in the Dry Bean Dataset might be more dispersed due to the dataset's complexity and the variety of physical characteristics of beans. In contrast, the Synthetic A3 Dataset, with fewer data points and classes, might show a more compact distribution in the PCA-transformed space.
- **Insights Gained:** PCA effectively reduces the dimensionality of both datasets while retaining the essential variance. For the Synthetic A3 Dataset, this could mean capturing synthesized or transformed features, while for the Dry Bean Dataset, it involves distilling key physical attributes of different bean types.

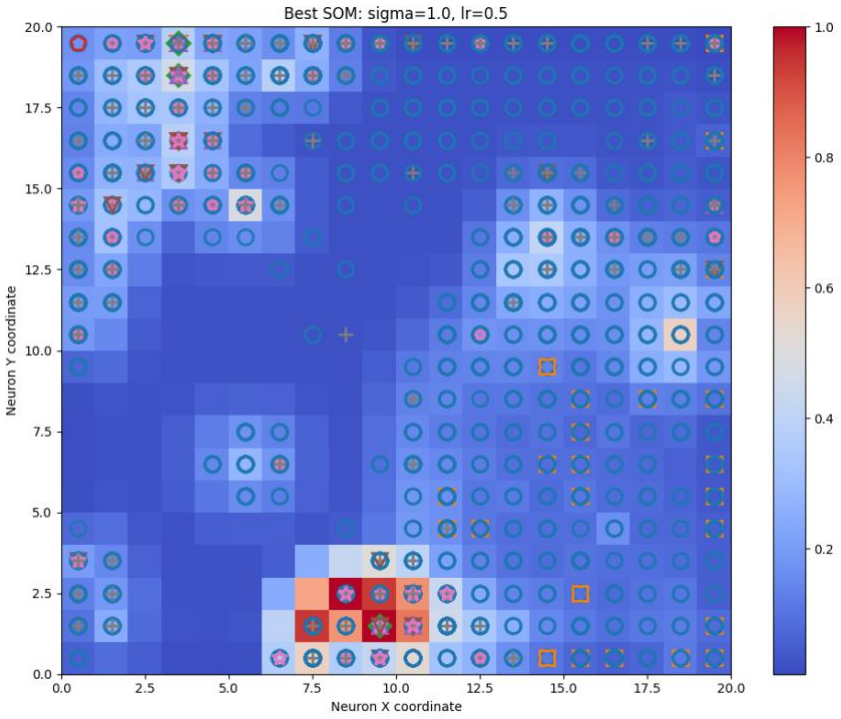
Self-Organizing Maps

SOM**,** also known as Kohonen maps, are a type of unsupervised neural network algorithm used for data visualization and dimensionality reduction. Developed by Teuvo Kohonen in the 1980s, SOMs enable the visualization of complex, high-dimensional data in a lower- dimensional (typically two-dimensional) space. They are particularly useful for identifying inherent patterns, clustering, and feature mapping in the data.

SOM plots for A3-text compared to the Dry_Bean_Dataset:



Compared to this (the Dry-Bean dataset):



When comparing the SOMs for the A3 and Dry Bean datasets, the key differences that stand out are the clarity of the clusters and the distribution of the features across the component planes. The Dry Bean SOM exhibits more defined clusters and component plane patterns, which suggests a dataset with more distinct groupings and a set of features that effectively differentiate between those groups. On the other hand, the A3 dataset shows more overlap and less distinction in both the U-matrix and component planes, indicating a dataset with a more complex or less well-defined structure.

The observed differences in SOM behavior could be due to inherent differences in the datasets, such as the number of features, the scale of the measurements, the presence of noise, or the intrinsic data distributions. The choice of SOM parameters, such as the learning rate, neighborhood function, and the number of neurons, could also impact the outcome of the visualizations.

The U-matrix visualized in the image above show the distances between neighboring neurons of the trained Self-Organizing Map (SOM). Each square in the grid corresponds to a neuron, and the color indicates the average distance to its immediate neighbors. Here's how to interpret the U-matrix:

- The color scale on the right side of the image shows a gradient from blue to red, with blue representing shorter distances (high similarity or density) and red representing larger distances (low similarity or sparsity).
- These likely represent clusters of similar data points. Neurons within these areas are similar to each other, suggesting a concentration of data points with similar features.

- These indicate boundaries between different clusters where the neurons are less similar to each other. The reddish areas can be interpreted as separations between different groups or classes in the data.
- The various markers (*, X, P, >, etc.) are overlaid on the U-matrix and colored according to their class. Each marker represents a data point that has been mapped to the winning neuron on the SOM grid.
- You provided several quantization error values, with the lowest being 0.20542249857455117. This suggests that, on average, data points are reasonably close to their winning neurons, indicating a good representation of the dataset by the SOM.
- The topographic error is given as 0.1388888888888889, which is relatively low. This suggests that the map preserves the topology of the dataset well, with most data points having their first and second-best matching units as adjacent neurons.

t-Distributed Stochastic Neighbor Embedding

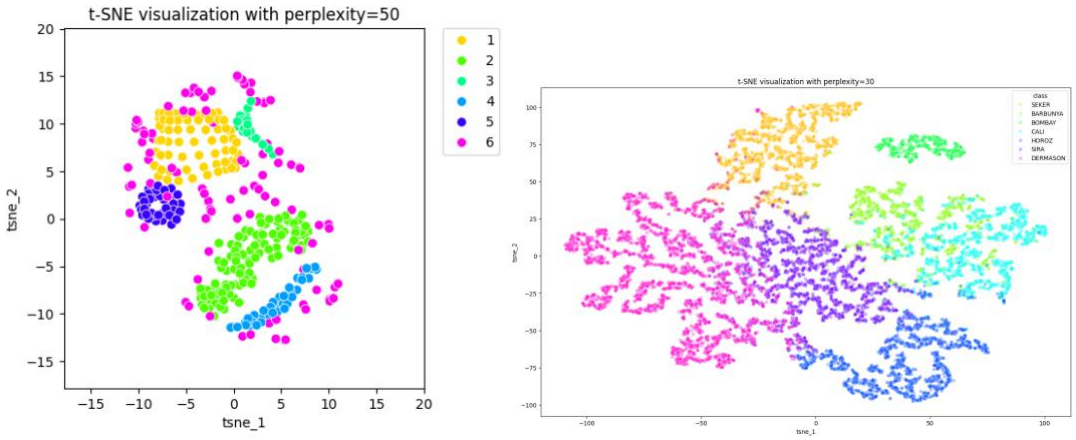
t-SNE is known for being computationally intensive and sometimes slow, especially on large datasets. However, if you're experiencing an unusually long or seemingly endless loop, it might be due to the size of the dataset or the specific parameters of t-SNE.

Computation Time: The dry bean dataset takes significantly longer (around 61 seconds) to process with t-SNE compared to the A3-text synthetic dataset (around 0.96 seconds). This difference is largely due to the difference in the number of samples.

Mean Sigma: The mean sigma (bandwidth) for the dry bean dataset is 0.066541, much smaller than the mean sigma for the A3-text synthetic dataset, which is 0.692581. A smaller sigma in the dry bean dataset suggests that the data points are more densely packed or that there's less variability in the distances between points.

KL Divergence: The final KL divergence after 1000 iterations for the dry bean dataset is 0.858700, and for the A3-text dataset, it is 0.481037. A lower KL divergence indicates a better fit of the t-SNE model to the data. This suggests that t-SNE may have been able to find a more coherent structure in the A3-text dataset than in the dry bean dataset.

Visualization and Interpretation: The final aspect, which is crucial for t-SNE, is how the data is visually represented in the 2D space and how well the t-SNE results capture the underlying structure of the data. Different datasets can lead 11 to very different visualizations, with some showing clear clusters while others might display more overlap or less distinct groupings.I have played with a few perplexity parameters to find the best results, you can see that in the images below for both datasets:



It's important to note that t-SNE is a stochastic algorithm, meaning it can produce slightly different results every time it's run, especially if the perplexity and learning rate parameters are changed. The interpretation of t-SNE results relies heavily on visual inspection and should be done with an understanding of the dataset's characteristics and the algorithm's nature.

Conclusion

This report dives into various unsupervised learning techniques, employing PCA, t-SNE, k-means, AHC, and SOM to analyze two distinct datasets: the Synthetic A3-data and the Dry Bean Dataset. Each technique brought a unique lens to the data, revealing underlying structures and patterns. PCA and t-SNE effectively reduced dimensionality, highlighting key features and variances. The clustering techniques, especially k-means and AHC, uncovered intrinsic groupings, illustrating both synthetic and natural classifications within the datasets. SOM, with its topological preservation, offered a distinct view, mapping complex multidimensional data into comprehensible formats. This study not only demonstrated the versatility and power of unsupervised learning in extracting hidden patterns and relationships but also provided a foundation for further exploration and application in diverse fields such as biology, linguistics, and marketing. The findings underscore the potential of these techniques in uncovering insights from unlabelled data, paving the way for innovative approaches in data analysis and interpretation.

GITHUB: https://github.com/sebastianbuzdugan/A3-NeuralNetworks