Intent of the application

The purpose of this application is to explore clustering techniques, as well as a dimensionality reduction technique (Principal Component Analysis) and compare the results in a data frame where the reduction technique was applied and a data frame where it was not applied.

Functional description

This application will take the iris dataset as an input, perform some basic transformations on the dataset, and create two different datasets from it: one in the original state, and one after performing PCA on the original dataset.

Using these datasets, the elbow method was used in order to get the ideal number of clusters for some of the clustering techniques.

Using this ideal number of clusters, the K-means clustering technique was used and compared between the original dataset and the dataset after performing PCA.

A similar process was followed to create dendrograms with the original dataset and the PCA dataset. The dendrograms are used for the agglomerative clustering technique, where the results were compared between the original dataset and the PCA dataset.

Affinity Propagation, Mean shift clustering, Spectral clustering, OPTICS, BIRCH, DBSCAN, and Hierarchical Density-Based Spatial clustering were also performed in a similar way.

Dataset to be used

The dataset to be used is the Iris dataset, for which more information can be found here: https://www.ritchieng.com/machine-learning-iris-dataset/

Mathematical background

Elbow Method

Clustering is a technique used in Data Mining to split the observations in a dataset into multiple groups.

One of the most used methods for this is K-means, which consists in splitting n observations into k clusters, according to their mean.

The elbow method is one of the most used methods to determine K. It is a graphical method, in which the percentage of explained variation is plotted as a function of the number of clusters. When an "elbow" is formed in the plot, that is the number of clusters that should be used.

Principal Component Analysis (PCA)

Dimensionality reduction method to reduce the dimensionality (variables) of large datasets, while preserving its most essential information.

It is used to reduce the complexity of algorithms, while sacrificing as little accuracy as possible.

It can also help to reduce the noise in the data.

K-means

The main objective of the k-means clustering technique is to split n observations from a dataset into k clusters, using the nearest mean (cluster center) as the criteria. The mean used as the criteria depends on the observations and their features. The k comes from the previously described term, the Elbow Method.

The mean used as the criteria gets updated with each observation that gets added to the cluster, making the mean a more “accurate” mean.

DBSCAN

Stands for Density-Based Spatial Clustering of Applications with Noise. It is a Density-based clustering algorithm.

This algorithm has two differences with the previous algorithm:

It does not need a previously calculated K for the number of clusters, as they will be automatically chosen.

It does not take into consideration the features themselves, but how close some observations are to others.

It requires two arguments: minimum number of points, and epsilon, which is the “radius” of the “circle” where the observations are densely packed.

This algorithm creates a circle of radius epsilon around each observation and counts how many observations are around it. If there are at least as much as the minimum number of points, then this observation is a core observation. If there are fewer, then it is just a border observation.

If there are no other observations close to it, then it is considered noise.

Gaussian Mixture Model

This algorithm takes the number of clusters, in this case, K.

It assumes that every observation is generated from a mixture of a finite number of Gaussian distributions with unknown parameters.

Each of these distributions is a cluster. So, this algorithm gathers every observation that belongs to the same Gaussian distribution.

Agglomerative Hierarchical Clustering

Most common type of hierarchical clustering. Also known as AGNES (Agglomerative Nesting).

It works by assigning each observation to a single cluster made of a single element.

Then the clusters get merged into the closest cluster repeatedly, until every observation is inside one single cluster.

By performing this action, the result is a tree-based representation of the dataset, which is known as a dendrogram.

It is considered a bottom-up algorithm. It is the opposite of a divisive algorithm, which starts with a single cluster for every observation, then splits them repeatedly until every observation is in their own cluster, where the cluster is made of a single element.

As the previous algorithm, it uses the distance between the observations to create the distributions or the clusters. The distance can be the Euclidean distance.

Affinity Propagation

Algorithm that identifies exemplars among data points and forms clusters of data points around these exemplars. It does not require for the number of clusters to be previously defined. Every observation is a potential exemplar, and it exchanges messages between the observations to determine which observations are good exemplars, and the number of clusters these generate.

The messages tell the other observations how good of an exemplar it is, and they decide if it is the best exemplar.

All points with the same exemplar are placed in the same cluster.

Mean Shift Clustering

This unsupervised algorithm does not require for the clusters to be previously defined. Particularly useful when clusters are not well defined and separated by linear boundaries, but mixed shaped.

It basically "shifts" the observations towards the mean (regional) iteratively, and the destination of each observation is the cluster to which it belongs.

Some of the advantages are that it is robust to outliers, and it does not assume a pre-defined shape.

However, it is very computationally expensive, and does not scale well.

Spectral Clustering

Algorithm that uses the spectrum (eigenvalues) of the similarity matrix of the data to perform dimensionality reduction before clustering the observations.

OPTICS Clustering

OPTICS stands for Ordering Points to Identify the Clustering Structure.

Density based algorithm, similar to DBSCAN.

It requires two parameters: epsilon, which determines the maximum distance to consider, and minpts, which would be the number of points required to consider an agglomeration as a cluster.

It is very similar to DBSCAN, but it fixes one thing: It can detect meaningful clusters in data of varying density.

BIRCH Clustering

Balanced Iterative Reducing and Clustering using Hierarchies.

Unsupervised algorithm.

It first performs a dimension reduction technique to work on the reduced dataset.

It works with dendrograms, and easily removes outliers. It splits the observations into multiple subclusters, and slowly builds up to a larger one using different clustering algorithms.

Use case

This application can be used to demonstrate the usefulness of both clustering techniques and dimensionality reduction technique with a well-known dataset, which would allow for easy reproduction.

This application is used to show the differences in clustering techniques, their advantages, disadvantages, requirements, implementation, and the difference that using PCA can have in the analysis of a dataset.

Variables

Sepal.Length: The length of the sepal, which is the outer part of the flower that encloses a developing bud.

Sepal.Width: The width of the sepal.

Petal.Length: The length of the petal of the flower, which are leaves that surround the reproductive parts of a flower.

Petal.Width: The width of the petal of the flower.

Labels

Species: Describes the species of the flower associated to the previous measurements.

Data import

In this application, there is no input needed from the user.

Proposed Libraries

Seaborn: Data visualization library based on matplotlib. Source: https://seaborn.pydata.org/. Version: 0.11.2

Matplotlib: Library for creating static, animated, and interactive visualizations in Python. Source:3.7.2 https://matplotlib.org/stable/users/release_notes.html.

Sklearn: Machine learning tool for predictive data analysis. Supports both supervised and unsupervised learning. Version: 1.0.2

Source -> https://scikit-learn.org/stable/getting_started.html

Scipy: Provides algorithms for optimization, integration, interpolation, eigenvalue problems, algebraic equations, differential equations, statistics and others. Version: 1.11.1

Source -> https://scipy.org/install/

hdbscan: It performs the HDBSCAN algorithm. Version: 0.8.30 Source -> https://pypi.org/project/hdbscan/

Numpy: Used for vectorization and indexing for scientific computing. Version: 1.25.1 Source -> https://github.com/numpy/numpy

Pandas: Data analysis and data manipulation library. Version: 2.0.3 Source -> https://pandas.pydata.org/getting_started.html

Plots

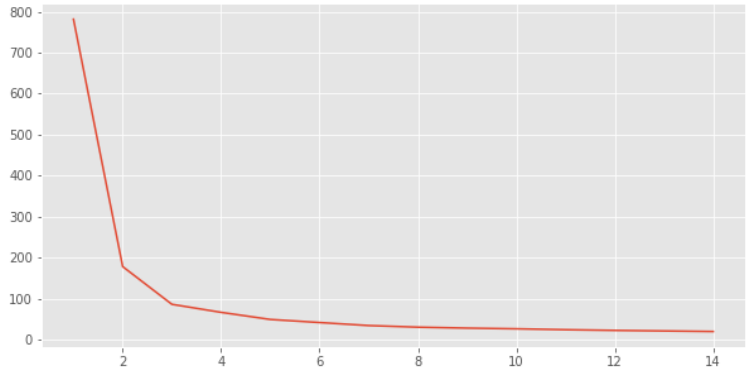

Figure 1Elbow method – plot

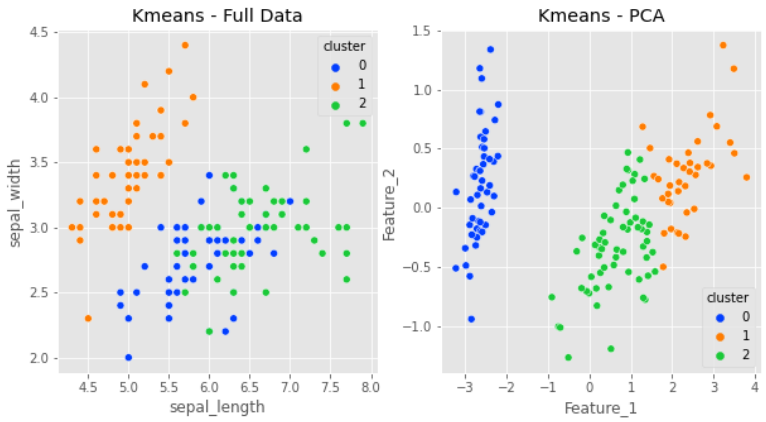

Figure 2 K-means clustering - Full data & PCA

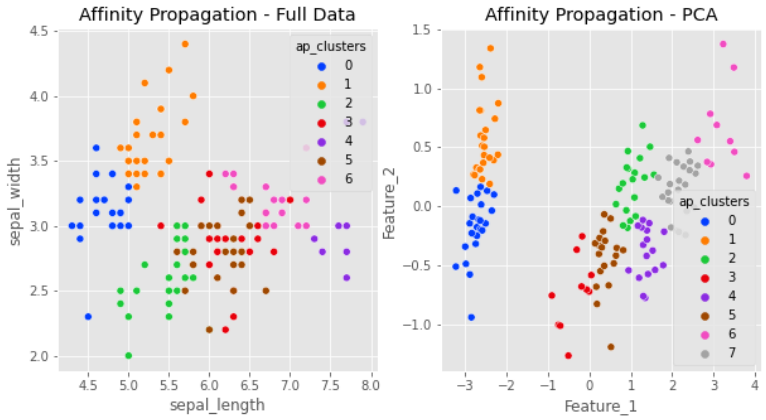

Figure 3 Affinity Propagation - Full data & PCA

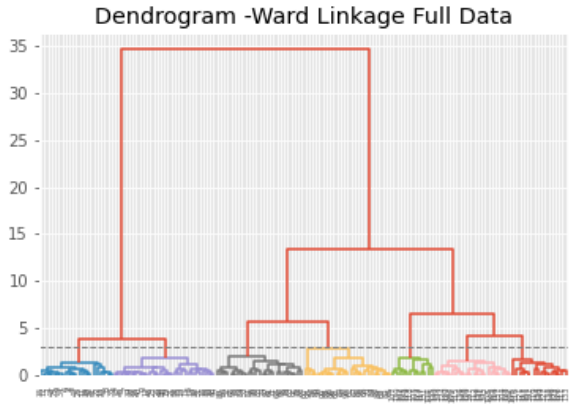

Figure 4 Dendrogram - Ward Linkage Full data

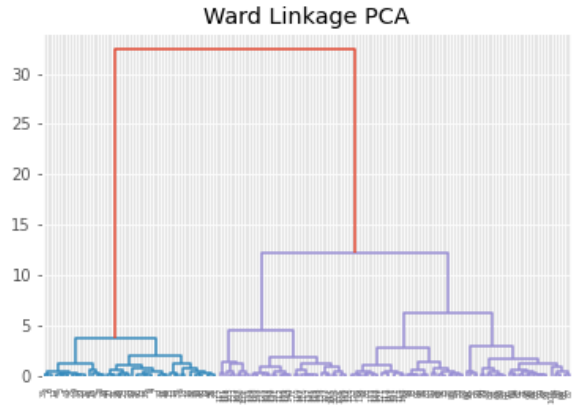

Figure 5 Ward Linkage – PCA

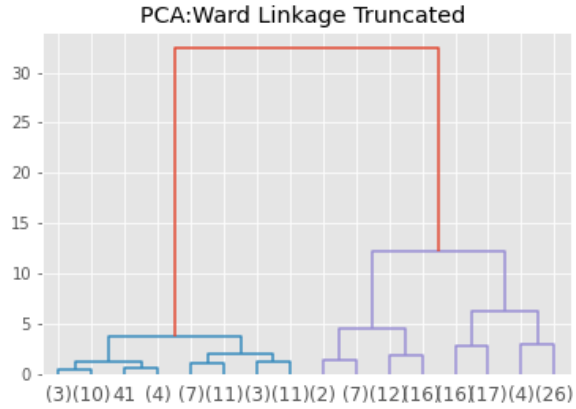

Figure 6 PCA-Ward Linkage Truncated

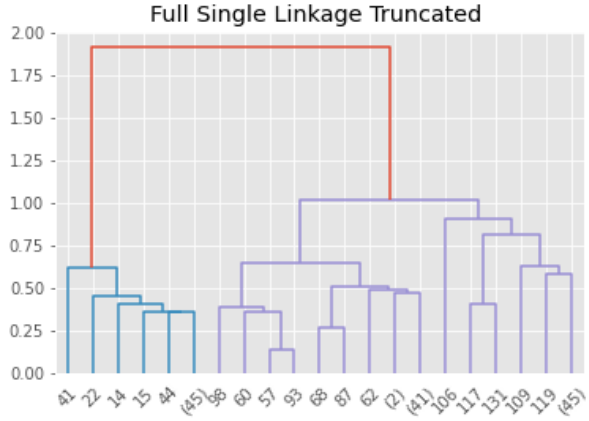

Figure 7 Full Data Single Linkage Truncated

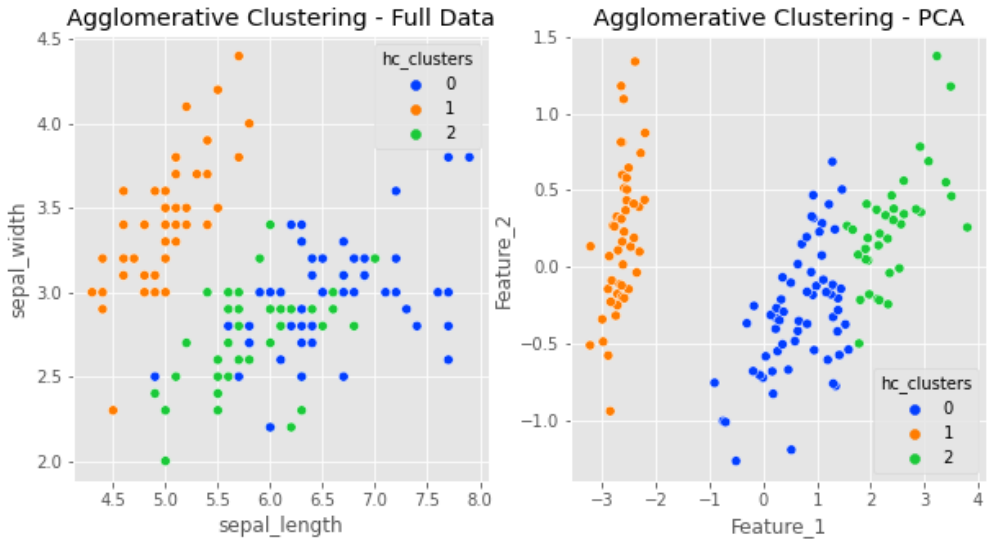

Figure 8 Agglomerative Clustering - Full Data & PCA

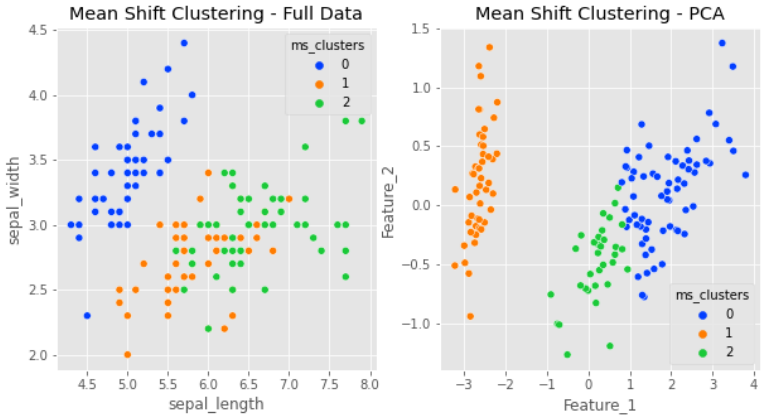

Figure 9 Mean Shift clustering - Full Data & PCA

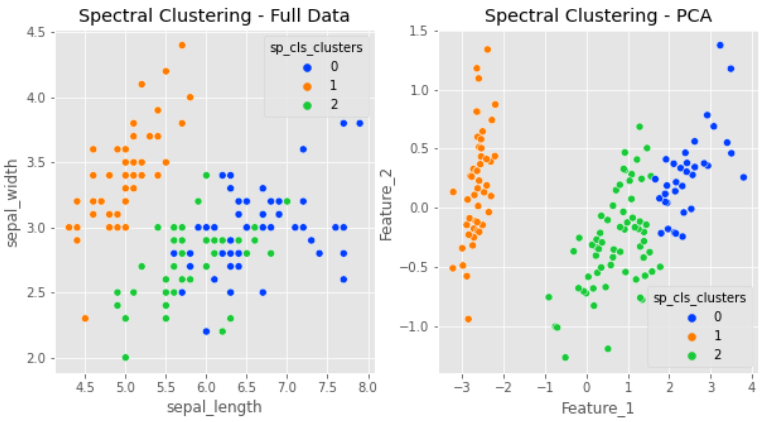

Figure 10 Spectral Clustering - Full Data & PCA

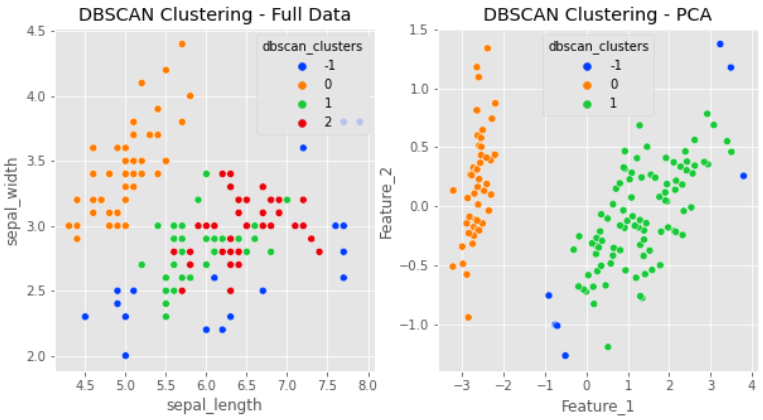

Figure 11 DBSCAN Clustering - Full Data & PCA

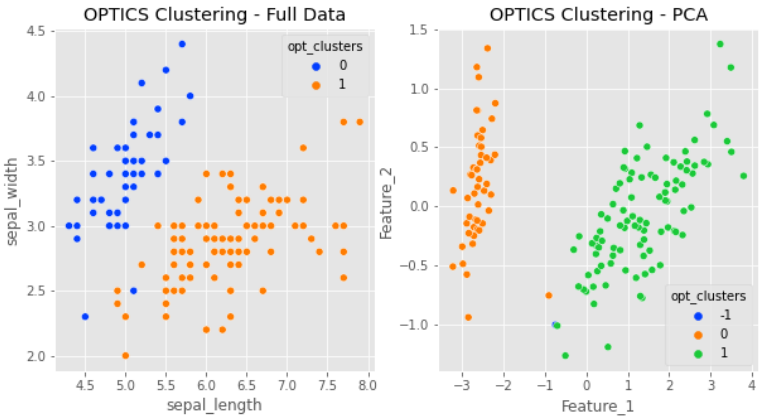

Figure 12 OPTICS Clustering - Full Data & PCA

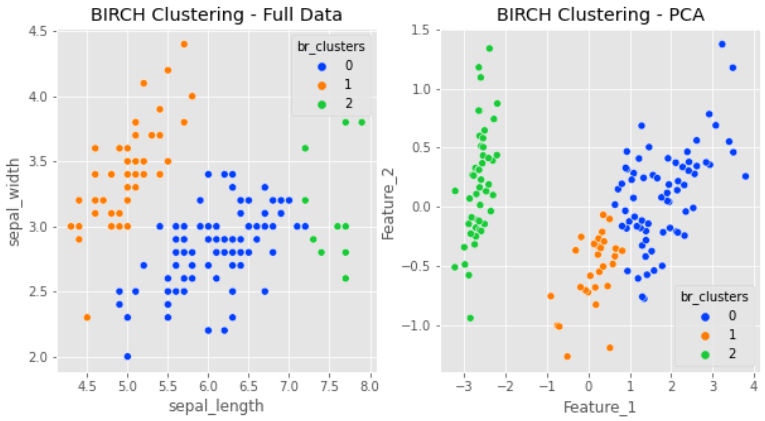

Figure 13 BIRCH Clustering - Full Data & PCA

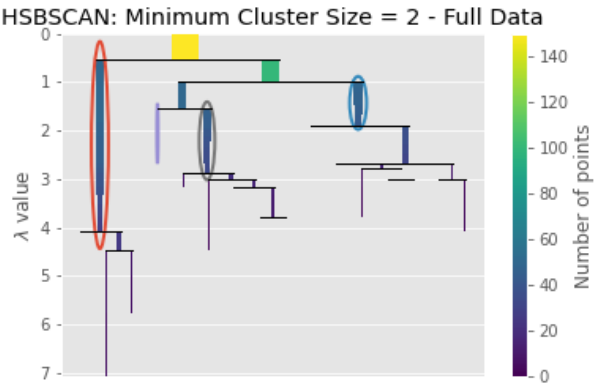

Figure 14 HSDSCAN - Minimum Cluster size 2 - Full Data

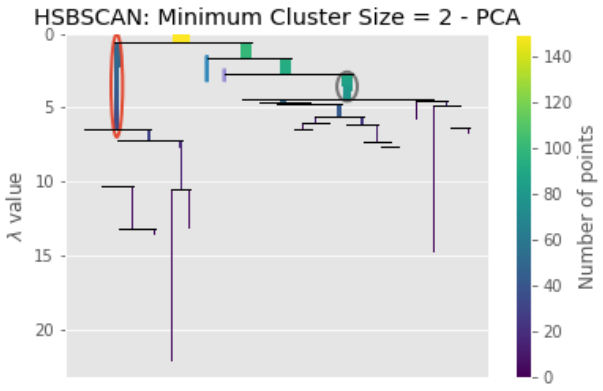

Figure 15 HSDSCAN - Minimum Cluster size 2 – PCA

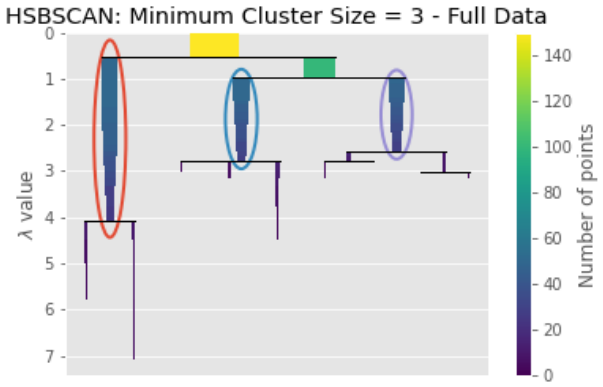

Figure 16 HSDSCAN - Minimum Cluster size 3 - Full Data

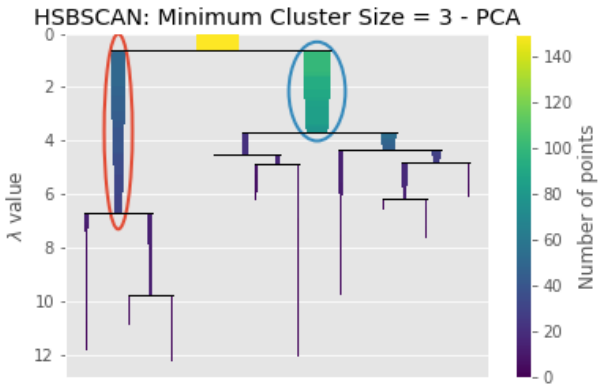

Figure 17 HSDSCAN - Minimum Cluster size 3 – PCA

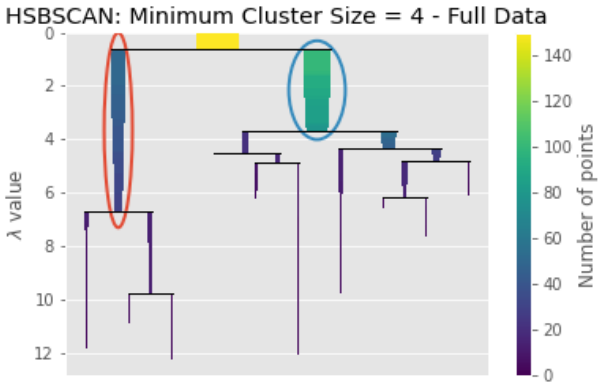

Figure 18 HSDSCAN - Minimum Cluster size 4 - Full Data

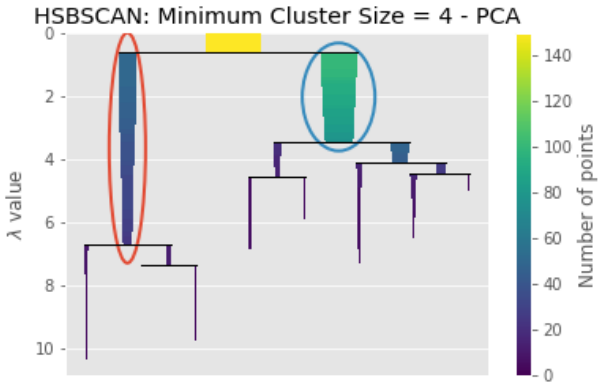

Figure 19 HSDSCAN - Minimum Cluster size 4 – PCA

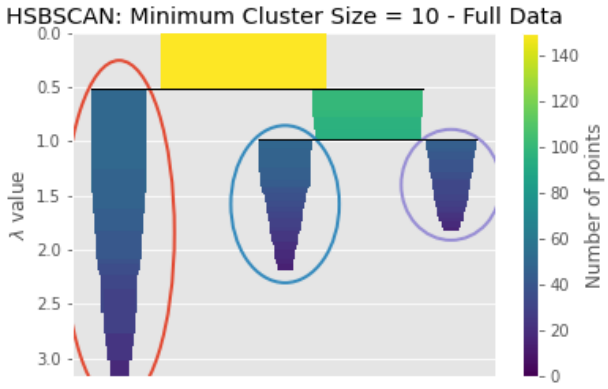

Figure 20 HSDSCAN - Minimum Cluster size 10 - Full Data

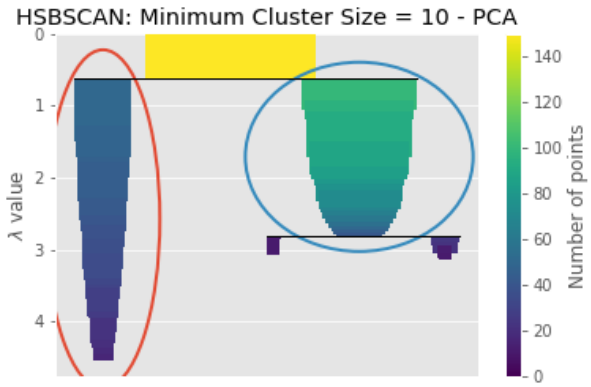

Figure 21HSDSCAN - Minimum Cluster size 10 – PCA

Conclusion

We can see that the use of PCA for K-means clustering improved the accuracy, which can be demonstrated with the observations that match the right classification.

In K-means, when using the full dataset, the accuracy was only of 49/150 = 0.32

But if PCA is used, then the accuracy goes to 67/150 = 0.44

The accuracy is not high, but it is higher than with the full dataset.

Unfortunately, the use of PCA in Affinity Propagation had no real effect, as the accuracy for both the original dataset and the dataset with PCA is low, as the algorithm incorrectly guessed that the number of clusters should be 6 and 7, respectively.

The original number of clusters is 3, so anything that falls outside these 3 clusters is wrong.

The results for Agglomerative Clustering are similar to K-means clustering, as the number of clusters is not decided by the algorithm but passed in as a parameter. The results for PCA were almost identical to the results for PCA in K-means clustering.

Meanwhile, the results for mean shift clustering and spectral clustering are almost identical, both for the original dataset and the PCA dataset.

Unfortunately, the results for DBSCAN were poor, for both the original dataset and the PCA dataset.

The results for the original dataset returned 4 clusters, while the results for PCA returned 3 clusters.

Although the PCA results returned the right number of clusters, the actual results are not correct, so we cannot really say that PCA improved the accuracy by much.

The same can be said about OPTICS clustering. The results for the original dataset returned only 2 clusters, while the results for the PCA dataset returned the right number of clusters, but the actual results only use 2 of those clusters.

As a conclusion, we can see that the use of PCA improves the accuracy of some algorithms, despite it not being the main function of PCA.

In our dataset, PCA was useful, as the columns represent lengths and widths for flowers, which are usually correlated, which favors PCA.