Assignment 4

author: Antonio Dehesa

# 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 dataframe where the reduction technique was applied and a dataframe where it was not applied.

# Dataset

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.

# 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

datasets: Used to import the Iris dataset.

ggplot2: Used to create plots from the dataset.

factoextra: Used to create the elbow plot, as well as visualize multivariate data analyses. In this case, both the elbow plot and the PCA (Principal Component Analysis) will be used.

fpc: Used to perform the DBSCAN clustering technique.

ggfortify: Used to perform plots on PCA.

ClusterR: Used to perform the Gaussian Mixture Model technique.

Tidyverse: Used to facilitate data manipulation used in the clustering techniques and plots.

cluster: Used for some clustering algorithms.

dendextend: Used to create dendrograms, which will allow us to plot them.

ape: Used to facilitate the plotting of dendrograms, as well as their manipulation.

FactoMineR: Used to perform the PCA dimensionality reduction technique.

## Library source

datasets: source -> https://cran.r-project.org/package=dataset

ggplot2: source -> https://github.com/tidyverse/ggplot2

factoextra: source -> https://cran.r-project.org/web/packages/factoextra/index.html

ggfortify: source -> https://cran.r-project.org/web/packages/ggfortify/index.html

fpc: source -> https://cran.r-project.org/web/packages/fpc/index.html

ClusterR: source -> https://cran.r-project.org/web/packages/ClusterR/index.html

tidyverse: source -> https://cran.r-project.org/web/packages/tidyverse/index.html

cluster: source -> https://cran.r-project.org/web/packages/cluster/index.html

dendextend: source -> https://cran.r-project.org/web/packages/dendextend/index.html

ape: source -> https://cran.r-project.org/web/packages/ape/index.html

FactoMineR: source -> https://cran.r-project.org/web/packages/FactoMineR/index.html

# Plots

## K-means

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

Figure 1 Plot of K-means algorithm without using PCA.

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

Figure 2 Plot of K-means algorithm using PCA.

As we can see, the plot for the data with PCA does not help us to directly compare the results with the original dataset.

So, we could instead simply plot the iris dataset, using the results with PCA for the colors, another plot using the clusters without PCA for the colors, and the original dataset with the original clusters for comparison:

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

Figure 3 Plot of K-means cluster without PCA

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

Figure 4 Plot of K-means cluster with PCA

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

Figure 5 Plot of original classification of the Iris dataset

As we can see, the plots for the original dataset and the one after applying PCA are remarkably similar, with the only difference being the colors.

However, the plot for the clusters without applying PCA are quite different.

Therefore, we can confirm that using PCA improved the performance for this algorithm.

## DBSCAN

We can perform a similar analysis as in the previous algorithm:

Calendario

Descripción generada automáticamente

Figure 6 DBSCAN clustering without PCA

Calendario

Descripción generada automáticamente

Figure 7 DBSCAN clustering with PCA

Calendario

Descripción generada automáticamente

Figure 8 Original classification of the Iris dataset

As we can see, some observations improved without PCA, and some improved with PCA.

The one without using PCA seems to be more accurate, but it left several observations without a valid cluster, as we saw it chose 4 clusters, when the maximum we can really have is 3.

After applying PCA, we can see that the number of clusters was 3, which is consistent with our original dataset.

However, we can also see that, even though every observation received a classification, several of them received an incorrect classification.

That is why the accuracy did not seem to improve.

## Gaussian Mixture Model

We can use a similar approach as in the previous method:

Logotipo, Calendario

Descripción generada automáticamente

Figure 9 Gaussian Mixture Model clustering without PCA

Logotipo, Calendario

Descripción generada automáticamente

Figure 10 Gaussian Mixture Model with PCA

Calendario

Descripción generada automáticamente

Figure 11 Original classification of the Iris dataset

As previously seen, the accuracy between using PCA and not using it is not vastly different.

As a result, the plots for pre-PCA and post-PCA are basically the same.

We can see, despite the difference in colors for the original dataset, some observations are in different colors in the pre and post PCA plots.

This lets us know that neither of them is highly accurate.

## Agglomerative Hierarchical Clustering

Imagen que contiene Círculo

Descripción generada automáticamente

Figure 12 Circular Dendrogram for Agglomerative Hierarchical Clustering without PCA

Diagrama, Escala de tiempo

Descripción generada automáticamente

Figure 13 Dendrogram for Agglomerative Hierarchical Clustering with PCA

As we can see, the plot from before applying PCA and after applying PCA are remarkably similar, and they are both similar to the original plot, but with some differences.

Then we can perform a similar analysis as in the previous methods:

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

Figure 14 Agglomerative Hierarchical Clustering without PCA

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

Figure 15 Agglomerative Hierarchical Clustering with PCA

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

Figure 16 Original classification of the Iris dataset

As we can see, the plot from before applying PCA and after applying PCA are remarkably similar, and they are both similar to the original plot, but with some differences.

# Application outputs

The expected outputs for this application are:

The results of applying clustering techniques to a dataframe of the original dataset, and to the same dataframe after applying a dimensionality reduction technique (Principal Component Analysis).

The plots comparing the clustering technique before and after applying PCA on the dataset, as well as its accuracy compared to the original classification in the Iris dataset.

# Analysis of results

As we can see, for some of the clustering techniques, the use of PCA on the Iris dataset improved the accuracy of the classification, but in some, the difference was negligible.

The most noticeable difference was in the K-means clustering technique, as the accuracy improved from only 79 observations being correct, to 125 being correct.

The clustering technique with the least difference before and after applying PCA is DBSCAN, as both have the same number of correct classifications. However, they do have different mistaken classifications.

Therefore, we can conclude that PCA, although helpful, is not perfect, and does not mean that by using it, the clustering techniques, or any other analysis we perform will automatically improve.