

1 Fundamentals

1. Clustering: getting started

- (a) Download the synthetic clustering datasets (fruits) from the class webpage. Load the first one.
- (b) Using sklearn library, apply `KMeans` algorithm on the dataframe, with 2 clusters, on the numerical columns. You can check the documentations to see how to do it, it should be something like `clusters = KMeans(n_clusters=2).fit_predict(df[["weight","diameter"]])`.
- (c) Add the clusters found as a "cluster" column to the dataframe
- (d) To observe the clusters, we can plot (with `seaborn` library for instance, `sns.scatterplot`) with dot colors corresponding to clusters (hue="cluster"). Compare with using the `fruit` information as colors (ground truth clusters). You should see that the clusters are not the expected ones. Guess why.

2. Limits of k-means

- (a) Normalize the data, and retry on the first dataset. It should solve the problem.
- (b) Do the same with the second example. The result should not be as expected. Do you understand why?
- (c) Try to solve the issue using Gaussian Mixture (class `BayesianGaussianMixture`). Check the `covariance_type` parameter.
- (d) Also try the DBscan approach
- (e) Do the same (comparing ground truth, k-means, GM, DBScan) on the other examples. Every time, try to understand why each of the method succeed or fail. Try to play with the parameters to make the methods succeed.

3. Interpreting clusters

An important part of using clustering is to make sense of the clusters obtained.

- (a) Download the cleaned toy car dataset from the class website. Keep only the numerical values (e.g., `df.select_dtypes(include='number')`)
- (b) Use k-means with 3 clusters.
- (c) Compute the centroid (mean values for each feature), and the size for each cluster. A flexible way to proceed is to extract the clusters (`fit_predict`), add the resulting list as a new column (e.g., "cluster") in a copy of the feature dataframe, then compute statistics by cluster in that dataframe, for instance with `.groupby("cluster").agg(['mean','count'])`
- (d) If you had to give a manual label to those clusters, to describe the cars they contain, what would it be ? (e.g.: "large and old expensive cars"...)
- (e) Check the difference with and without normalization

4. Evaluation and number of clusters

- (a) Compute the silhouette score using method `silhouette_visualizer` from package `yellowbrick`, plot the silhouette score and interpret it.
- (b) We would like to find the optimal number of clusters. Apply the silhouette score method: plot the relation between k and the silhouette score, and search for a maximum value. You can use the `kelbow_visualizer`, with the `metric` option set to silhouette. (By curiosity, you can check other cluster evaluation metrics)

2 Going Further

- (a) Using this knowledge, explore the proposed Wine dataset: <https://www.kaggle.com/datasets/harrywang/wine-dataset-for-clustering>