title: Unsupervised M.L

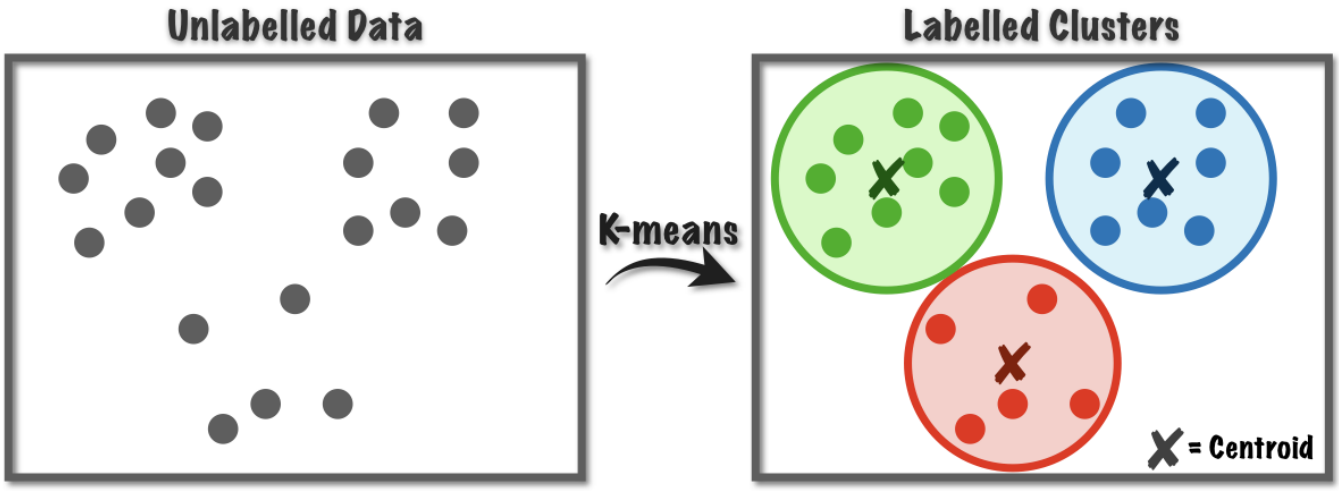
author: Wilber B. Quito, Andrea Ramirez

date: 11/12/2022

Clustering learner with K-means

The project's goal is to grouping samples linked by the optimal number of clusters using any clustering algorithm.

We picked K-means as clustering algorith and we used the Prediction Strength algorithm in order to evaluate the clustering. Prediction Strength algorithm is an option to find the optimal number of centroids, hence the optimal number of clusters. There are other options to find the optimal number of cluster with K-means, for example GAP.



Implementation

The implementation is made in two different scripts. We have the scripts *learner.py* and *preditor.R*.

learner.py

The script expects to find a file *training.csv* which should have the samples to clusterize. The implementation tries to find the optimal number of cluster between 1 and 7 included. Once the algorithm had found the optimal number of clustering, it exports the number of centroids found in the *training.csv* distribution and it's centroids into a file named *param.out*.

We had not used any library to compute the Prediction Strength, instead, we implemented from scratch the algorithm using the following equation. The implementation is in the file *compute.py* in the *learner* module.

$$\mathrm{ps}(k) = \min_{1 \le j \le k} \frac{1}{n_{kj}(n_{kj}-1)} \sum_{i \neq i' \in A_{kj}} D[C(X_{\mathrm{tr}}, k), X_{\mathrm{te}}]_{ii'}.$$

predictor.R

The script get's the output of *learner.py*, picks the centroids and reads the file *testing.csv* and assign each sample of the testing into a cluster by computing the minimun Euclidean distance between each sample and the centroids. Finally, exports the clusterization into a file named *clustering.out* where each *i* column of this file is the clustering assignation of the *i* sample of *testing.csv*.