The first part of any clustering method consists of parsing the input data in order to have it in a format in which the clustering algorithm is able to work with.

In the case of this project the data is loaded from *arff* files using the function loadarff contained in the library scipy. When a database is loaded it is a list of lists where each row corresponds to an instance and each column corresponds to one of the attributes, except for the final column where there are specified the real class of each instance.

The functionality of the parsing function designed in this project is to split the initial data into a list of lists containing only the instances and its attributes and another list which contains the classes.

Once the split has been done, it is not returned directly, there are some preprocess made on the data, iterating the attributes (columns):

* If the attribute is numerical, the ‘*nan’s* are substituted by the average of the column. If the most common value in the attribute is *‘nan’*, the column is erased because there is not enough data to take it into account for the clustering.

After that, the column is normalized in a way that the minimum value is 0 and the maximum is 1, so the maximum distance is 1.

* If the attribute is nominal, the ‘*?’s* are substituted by the most common value of the column. If the most common value in the attribute is *‘?’*, the column is erased because there is not enough data to take it into account for the clustering.

Finally, the parser function returns the preprocessed data and the classes.