7 k-means

You can use external libraries for linear algebra operations but you are expected to write your own algorithms.

7.1 Exercise 1

- Download the `breast_cancer.csv` dataset (original data available at `https://archive.ics.uci.edu/ml/datasets/Breast+Cancer`) and preprocess it by using `sklearn.preprocessing.OrdinalEncoder` to properly deal with the categorical variables.
- Write your own function to compute the Mutual Information Criterion.
- Compute the Mutual Information between the covariates and the response variable (stored in the last column).

Which features appear to be the most significant?

7.2 Exercise 2

- Use the dataset `s3.txt` available in the `Datasets` folder.
- Write your own implementation of the k-means clustering algorithm.
- Test your implementation with 10 different initializations and $k = 15$.
- Plot the clustering results for which the loss is, respectively, the highest and the lowest.

Notes

k-means can be quite slow if not programmed correctly. Of course, the goal of the exercise is not to produce the most efficient code (ie. it's ok if it is a little slow) but for you to understand the mechanisms of the algorithms. In practical application, you would likely use Cython as programming tool other than Python directly (or use already-made implementations).

Nonetheless, some useful notes you could take into account while programming are the following:

- use `numpy`
- try to use vectorization when possible instead of for loops, as it increases the efficiency of the code