LAB 4: Clustering

Assignment 1

(a) Generate 4 Gaussian clusters in two dimensional space given by two numeric variables in range [-10, +10]. The Gaussian distributions of the clusters must have different means and the same standard deviation, initially set to 0.6. The total number of instances for all the four clusters is 300.
Hint: use function `make_blobs` from `sklearn.datasets.samples_generator`.

(b) Run *k*-means clustering algorithm on the data obtained in (a) and visualize the clusters and their centers for *k* in {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}.
Hint: use function `KMeans` from `sklearn.cluster`. Set parameter `random_state` to `None.`
Print the contingency tables of the clustering solutions.
Hint: use function `contingency_matrix` from **`sklearn.metrics.cluster.`**

(c) Plot the sum of square error (SSE) for *k* in {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}. Does the plot indicate that the natural number of clusters is 4?

(d) Repeat (a)-(c) when you generate the clusters with the standard deviation of 0.1 and 2.5. Do the SSE plots indicate that the natural number of clusters is 4?

(e) Repeat (d) with another cluster-center initialization by setting the parameter `random_state` of `KMeans` to an integer number. When do (not) you receive clustering solutions similar to those obtained for `random_state`=`None`? Can you explain why? Can you propose an extension of the k-means initialization that results in similar clustering solutions?

Assignment 2

(a) Load and print the vertebrate.csv data.

(b) Run single-link, max-link and average-link hierarchical clustering on this data. Vizualize the hierarchies. Indicate the hierarchy that in your view is most natural given the data.
Hint: To run hierarchical clustering use function `hierarchy.linkage` from `scipy.cluster`. To visualize `hierarchy.dendrogram` from `scipy.cluster`.

Assignment 3

(a) Load and visualize the chameleon.csv data.

(b) Run the DBSCAN method on this data for eps=15.5 and min_samples=5. Vizualize the clustering solutions.
Hint: To run the DBSCAN method hierarchical clustering use function `DBSCAN` from `sklearn.cluster`.

(c) Experiment with the sensitivity of the DBSCAN method for eps in [0.1, 20] with step of 0.5 and min_samples in [1, 20] with step of 2.