Task 4:

**Learning Objectives**:

1. Learn to use popular clustering algorithms, namely K-means, and DBSCAN
2. Learn how to summarize and interpret clustering results
3. Learn to write analysis and evaluation functions which operate on the top of clustering algorithms and clustering results
4. Learning how to interpret unsupervised data analysis results

**Datasets**: In the project we will use the Complex9 and the Cleaned Pima Indians Diabetes dataset. The Complex9 dataset is a 2D dataset which can be found at <http://www2.cs.uh.edu/~ceick/UDM/DataSets/Complex9.txt>

and Cleaned Pima Indian Diabetes is an 8D dataset which can be found at:

<http://www2.cs.uh.edu/~ceick/DM/Pid-clean.csv>

More about the original Pima Indian Dataset can be found at: [Pima Indians Diabetes Database | Kaggle](https://www.kaggle.com/uciml/pima-indians-diabetes-database); however, you will use a cleaned version of the dataset in which some null values in the original dataset have been removed from the original dataset.

However, we will ignore the 4th and 5th attributes (SkinThickness and Insulin) and will z-score the dataset before clustering the dataset; we call this dataset *ZPID[[1]](#footnote-1)* in the following. The last attribute of each dataset denotes a class variable which should be ignored when clustering the data sets—however, the class variable will be used in the post analysis of the clusters generated by running K-means, and DBSCAN.

**Task 4 Subtasks:**

1. Write an function[[2]](#footnote-2) purity(a,b,outliers=FALSE) that computes the purity of a clustering result based on an apriori given set of class lables, where *a* gives the assignment of objects in O to clusters, and *b* is the “ground truth”. Purity is defined as follows: Let

O be a dataset

X={C1,…,Ck} be a clustering of O with Ci ⊆O (for i=1,…,k), C1∪…∪Ck ⊆O and Ci∩Cj=∅ (for i≠ j)

PUR(X)= (number\_of\_majority\_class\_examples(X)/(total\_number\_examples\_in\_clusters(X))

If the used clustering algorithm supports outliers, outliers should be ignored in purity computations; if you use R-clustering algorithms, you can assume that cluster 0 contains all the outliers, and clusters 1,2,…,k represent “true” clusters. If the parameter outliers is set to FALSE, the function just returns a floting point number of the observed purity, if parameter outliers is set to T the function returns a vector: (<purity>,<percentage\_of\_outliers); e.g. if the function returns (0.98, 0.2) this would indicate that the purity is 98%, but 20% of the objects in dataset O have been classified as outliers.\*

1. Run[[3]](#footnote-3) K-means for k=9 and k=13 for the Complex9 dataset. Visualize the obtained two clusterings. Also compute their purity using the function you developed in Task a. Assess if the K-means clustering was able to “rediscover” the natural clusters of the Complex 9 dataset, captured by the 3rd attribute of the dataset \*\*
2. Run2 K-means for k=3 for the ZPID dataset. Compute the purity of the obtained clustering result; also create box plots for all 6 attributes of the obtained 3 clusters and report their centroids. We also recommend to create six boxplots for the whole dataset to have a reference point to interpret the boxplots of the three clusters. Finally, summarize based on the obtained boxplots and centroids what kind of objects each of the 3 clusters contains. \*\*\*\*\*
3. Try to obtain a DBSCAN clustering for the ZPID dataset, having between 2 and 15 clusters with less than 20% outliers. Report its purity! Don’t worry if the reported clustering has a low purity. \*\*
4. Develop a search procedure that looks for the “best” clustering by exploring different settings for the (MinPoints, epsilon) parameters of DBSCAN for the Complex9 dataset. The procedure maximizes purity of the obtained clustering, subject to the following constraints:
   * 1. There should be between 2 and 15 clusters
     2. The number of outliers should be 10% or less.

The procedure returns the “best” DBSCAN clustering found and the accomplished purity as its result[[4]](#footnote-4); please limit the number of tested (MinPoints, epsilon)-pairs to 3000 in your implementation! Explain how your automated parameter selection method works and demonstrate your automated procedure using an example!

Apply the procedure you developed to the Complex9 dataset and report the best clustering you found. Are you happy with the obtained solution? \*\*\*\*\*

If you did not succeed in writing the function that seeks of the optimal DBSCAN clustering, you can manually seek for the best clustering for the Complex 9 dataset and report it; also report how you searched manually for it. \*\*\*

Extra credit: Apply your search procedure also to the cleaned Pima Indian Diabetes Dataset and report the clusters of the best result and what purity you accomplished. \*

Remark: Number of \*’s indicate a tentative assessment of the number of points allocated to each subtask: more \*’s mean more points!

**Deliverables for Task 4:**

1. A Report[[5]](#footnote-5) which contains all deliverables for the 5 subtasks of Task 4.
2. The report should have an Appendix which describes how to run the procedure that you developed for Task e, if you developed such a procedure.
3. Another Appendix which contains the software/code you developed as part of Task 4.

1. ZPID has six attributes and a class variable. [↑](#footnote-ref-1)
2. This function could be an R-function, a Python function or any other function. You might find some implementation of this function online; it is okay to use those implementations, as long as you acknowledge in your report what you use, and not all software you find on the internet is running properly. [↑](#footnote-ref-2)
3. Run k-means 20 times and use the clustering with the lowest SSE! [↑](#footnote-ref-3)
4. It should report the number of clusters obtained and the percentage of outliers as well. [↑](#footnote-ref-4)
5. Single-spaced; please use an 11-point or 12-point font! [↑](#footnote-ref-5)