CMPT 459 Spring 2021
Data Mining
Martin Ester
TAs: Arash Khoeini and
Madana Krishnan Vadakandara Krishnan

[Total Marks:100]

The aim of this assignment is to implement DBSCAN, which is a density-based clustering algorithm, using Python programming language, and to test it on a household power consumption dataset. DBSCAN pseudocode is provided on page 268 of the lecture slides.

Household Dataset

This dataset contains 525600 measurements of electric power consumption in one house located in Sceaux (7km from Paris, France) in 2007. Attributes are as follows:

- **date**: Date in format dd/mm/yyyy
- **time**: time in format hh:mm:ss
- **global_active_power**: household global minute-averaged active power (in kilowatt)
- **global_reactive_power**: household global minute-averaged reactive power (in kilowatt)
- **voltage**: minute-averaged voltage (in volt)
- **global_intensity**: household global minute-averaged current intensity (in ampere)
- **sub_metering_1**: energy sub-metering No. 1 (in watt-hour of active energy). It corresponds to the kitchen, containing mainly a dishwasher, an oven and a microwave (hot plates are not electric but gas powered).
- **sub_metering_2**: energy sub-metering No. 2 (in watt-hour of active energy). It corresponds to the laundry room, containing a washing-machine, a tumble-drier, a refrigerator and a light.

- **sub_metering_3**: energy sub-metering No. 3 (in watt-hour of active energy). It corresponds to an electric water-heater and an air-conditioner.

Tasks

a. Preprocessing [15 marks]

The Household dataset contains missing values that you need to handle. Moreover, you should normalize the data. You can also apply any other preprocessing that you think is helpful. **Please note that you should explain all the preprocessing you have done in your report file to receive full marks.**

b. Implementation of DBSCAN [45 marks]

Implement the DBSCAN algorithm. If an object is density-reachable from two clusters, then it should be assigned to both clusters [10 marks]. The implementation takes the dataset as input (along with other necessary attributes) and returns a file with an additional attribute "cluster label" [10 marks]. Cluster labels should start from 0, and noise objects should be labelled as -1 [5 marks]. Your implementation has to include a method, named "*fit*", which gets input data and returns a list of cluster labels. **You will not get the marks for this part if your implementation does not have a method with exact same name and same functionality** [20 marks]**.**

c. Test your DBSCAN implementation on the household dataset. [40 marks]

Use the heuristic approach taught in the class or some other approach to determine reasonable values for the parameters epsilon and MinPts. Explain how you chose the parameters and include your k-distance-diagram in your report file [25 marks]. Provide the statistics of your resulting clustering: how many clusters, how many objects per cluster. Discuss how good the resulting clustering is [15 marks].

Hints:

- Running DBSCAN algorithm on this dataset takes approximately 18 minutes on a Intel core i5 processor.

- You may want to use a smaller subset of this data for debugging purposes, and then run your implementation on the whole dataset only after you made sure it works correctly. Note that this is only for debugging and your report has to be based on the **whole data.**
- You may want to remove *Date* and *Time* columns. Or you may want to become more creative! It's your choice.

[IMPORTANT] You should submit two files:

- A report file: *[studentID].pdf*
- A Python file: *[studentID].py*

Deadline: 23:59 pm PST on March 15th.
You will lose %10 of the marks for submissions after this deadline, as long as it's not more than 24 hours late. You will lose all the marks for submissions after that.

Libraries: You can use libraries including math, numpy, scipy, random, etc.
You MUST provide YOUR OWN code for the DBSCAN algorithm and for all the tasks specified in this assignment. These MUST be implemented from scratch i.e. not using scikit-learn or other libraries. You will be marked on the correctness of your implementation.