COMP40370 Practical 6

Clustering

Prof. Tahar Kechadi

Academic year 2020-2021

Question 1

The file `specs/question_1.csv` contains coordinates of 2-dimensional points. Write a Python script that:

1. Using all the attributes, performs the k-means algorithm for three clusters. If using `sklearn`, set a fixed random state of 0.
2. Save the input data with an extra column that contains the labels generated by KMeans into a file called `output/question_1.csv`. The new column should be called `cluster`.
3. Plots the clustering results into `output/question_1.pdf`. Make sure that clusters are marked with different colors.

Discuss the obtained clustering results in your report.

Question 2

The file `specs/question_2.csv` contains data related to nutritional content of several cereal brands.

1. Discard the columns `NAME`, `MANUF`, `TYPE`, and `RATING`.
2. Run the k-means algorithm using 5 clusters as target, 5 maximum runs, and 100 maximum optimization steps. Keep the random state to 0. Save the cluster labels in a new column called `config1`.
3. Run k-means again, but this time use 100 maximum runs and 100 maximum optimization steps. Again, use a random state of 0. Save the cluster labels in a new column called `config2`.
4. Are the clustering results obtained with the first configuration different from the results obtained with the second configuration? Explain your answer in your report.

5. Run the clustering algorithm again, but this time use only 3 clusters. Save the generated cluster labels in a new column called `config3`.

6. Which clustering solution is better? Discuss it in your report.

7. Save the input data with the newly generated columns into a file called `output/question_2.csv`

Question 3

The file `specs/question_3.csv` contains coordinates of 2-dimensional points. Write a Python script to perform the following tasks.

1. Discard the `ID` column, the use the `X` and `Y` coordinates to run the k-means algorithm to detect 7 clusters. Use 5 maximum runs, and 100 maximum optimization steps. Keep a random state of 0. Save the cluster labels into a new column called `kmeans`. Discuss the cluster results in your report.

2. Plot the generated clusters in a file called `./output/question_3_1.pdf`.

3. Normalize the `X` and `Y` columns in a range between 0 and 1, then use the DBSCAN algorithm to cluster the points again. Use a value of 0.04 for *epsilon*, and use 4 minimum points for neighborhood evaluation. Save the generated plot in a file called `./output/question_3_2.pdf`, and save the cluster labels into a new column called `dbscan1`.

4. Execute DBSCAN again, but this time use a value of 0.08 for *epsilon*. Plot the generated clusters in a file called `./output/question_3_3.pdf`, and save the cluster labels into a new column called `dbscan2`.

5. Save the data with the cluster labels in a file called `./output/question_3.csv`

6. Discuss the different clustering solutions in your report. Which solution is the best? What is the reason behind the differennces in the results?

Data files

- `./specs/question_1.csv`: data file for the first question
- `./specs/question_2.csv`: data file for the second question
- `./specs/question_3.csv`: data file for the third question
- `./specs/test_practical6.`: test suit to check your results

Expected output and submission data

Your submission should be a single archive file (zip, tar, tgz, ...) containing one folder called `output` and the following files and directories:

- `./run.py`: main Python script
- `./report.pdf`: your PDF report (4 pages maximum)
- `./output/question_1.csv`: cluster results for first question
- `./output/question_2.csv`: cluster results for second question
- `./output/question_3.csv`: cluster results for third question
- `./output/question_1.pdf`: cluster plot for first question
- `./output/question_3_1.pdf`: cluster plot for third question (k-means)
- `./output/question_3_2.pdf`: cluster plot for third question (DBSCAN, first configuration)
- `./output/question_3_3.pdf`: cluster plot for third question (DBSCAN, second configuration)
- `./specs`: folder with the original assignment files

The final deadline for the submission is **Sunday, 15th of November**, 2020, at **19:00**. You can submit your solution on Brightspace.

Grading

The grading for the assignment will be assigned as follows:

- Question 1: **20%**
- Question 2: **25%**
- Question 3: **25%**
- Report quality and content, code quality, submission format: **30%**

Programming requirements and tools

The assignment should be solved in Python, version 3.6 or above (3.7 is recommended). You shall use the following packages for this assignment:

- `pandas`
- `matplotlib`
- `sklearn`

In particular, the following user guides are available for the required algorithms of the assignment:

- k-means:
  https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
- DBSCAN:
  https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html
- Examples on how to generate the scatter plots for questions 1 and 3:
  https://matplotlib.org/3.1.1/gallery/lines_bars_and_markers/scatter_with_legend.html