Ira A. Fulton Schools of
Engineering
Arizona State University

CSE 572: Data Mining

Cluster Validation Project

Purpose

In this project, you will apply the cluster validation technique to data extracted from a provided data set.

Objectives

Learners will be able to:

- Develop code that performs clustering.
- Test and analyze the results of the clustering code.
- Assess the accuracy of the clustering using SSE and supervised cluster validity metrics.

Technology Requirements

- Python 3.11
- scikit-learn 1.3.2
- pandas 1.5.3
- numpy 1.26.3
- scipy 1.11.4
- matplotlib 3.8.2

Project Description

For this project, you will write a program using Python that takes a dataset and performs clustering. Using the provided training data set you will perform cluster validation to determine the amount of carbohydrates in each meal.

Please watch **the Cluster Validation Project introductory video** before beginning. This is located in Ed Lessons before the project's code challenge.

Note: Project details in the Overview Document were recently updated since the recording of the videos, so some directions or items may not match. Please follow the Overview Document directions to complete your project correctly.

Directions

Accessing Ed Lessons

You will complete and submit your work through Ed Lessons. Follow the directions to correctly access the provided workspace:

1. Go to the Canvas Assignment, “**Submission: Cluster Validation Project**”
2. Click the “**Load Submission…in new window**” button.
3. Once in Ed Lesson, select the assignment titled "**Submission: Cluster Validation Project**".
4. In the code challenge, first review the directions and resources provided in the description.
5. When ready, start working in the Python file “**main.py**”

Project Directions

There are two main parts to the process:

1. Extract features from Meal data
2. Cluster Meal data based on the amount of carbohydrates in each meal

Data:

Use the Project 1 data files:

- CGMData.csv
- InsulinData.csv

Step 1: Extracting Ground Truth:

1. From InsulinData.csv, take column Y (**BWZ Carb Input(grams)**) and get all the meal intake data, and derive the min and max values.
2. Discretize the meal amount in bins of size 20. In total, you should have n = (max-min)/20 bins.
3. Consider each row in the Meal Data Matrix (P x 30) that you generated in Project 2.
4. Put them in the respective bins according to their meal amount label. This will be your Ground Truth

Step 3: Performing clustering:

Use the features in your Project 2 to cluster the meal data into n clusters. Use DBSCAN and KMeans.

Step 4: Compute SSE

For each of the clusters, compute SSE values and combine them to get one SSE value for both KMeans and DBSCAN.

Step 5: Calculate the Entropy and Purity

1. You need to create two matrices one for KMeans and the other for DBSCAN which contain bins (b1,b2…bn) as the columns and clusters (C1, C2 …Cn) from KMeans and DBSCAN as rows

2. Populate the matrix by combining the cluster values that fall in the respective bins.

3. Calculate the Entropy and Purity using the formulas provided in the video.

Expected Output:

A Result.csv file which contains a 1 X 6 vector. The vector should have the following format:

SSE for Kmeans	SSE for DBSCAN	Entropy for KMeans	Entropy for DBSCAN	Purity for KMeans	Purity for DBSCAN

The Result.csv file should not have any headers, just the six values in six columns.

Submission Directions for Project Deliverables

This project will be auto-graded. You must complete and submit your work through Ed Lesson's code challenges to receive credit for the course:

1. To get started, use the "**main.py**' file provided in your workspace.

2. All necessary datasets are already loaded into the workspace.

3. Execute your code by running the "**python3 main.py**" command in the terminal to test your work.

4. When you are ready to submit your completed work, click on “**Test**” at the bottom right of the screen.

5. your work, submit it for auto-grading by clicking the “**Test”** button.

6. You will know you have completed the assignment when feedback appears for each test case with a score.

7. If needed: to resubmit the assignment in Ed Lesson

    a. Edit your work in the provided workspace
    b. Execute your code again by running the commands in the terminal
    c. Click “**Test**” at the bottom of the screen

8. Once you have finished working on the project, please submit it by clicking the **"Submit"** button at the top right corner of your submission space.

Your submission will be reviewed by the course team and then, after the due date has passed, your score will be populated from Ed Lesson into your Canvas grade.

Note:

1. Do not change the code file name; it must remain ‘**main.py**' for the auto-grader to recognize your submission.

2. When the auto-grader runs your Python file, it should generate a '**Result.csv**' file with the specified format. The **'Result.csv**' file should not include any headers and should only contain the metrics in a 1 x 6 matrix.

3. Avoid using absolute paths when accessing other files.

4. Before submitting to the grader, it is recommended to run your code file via the terminal to catch any potential runtime errors.

Evaluation

The autograder will evaluate your code based on the following criteria:

- 100 points: For developing a code in Python that takes the dataset and performs clustering.

- 40 points: For developing a code in Python that implements a function to compute SSE, entropy, and purity metrics. These two can be written in the same file.

- 60 points: Evaluate the supervised cluster validation results obtained by your code.