Project 2

The project has the following 4 tasks:

Task 1: Implement the Apriori algorithm to mine frequent itemsets.

You can refer to online code, and reuse part of online code. But you must understand the code. **If you reuse some online code, you need to tell what online code you use in your report. In the report, you need to find a way to show the result of your code is correct.**

Task 2: Use 3 datasets to run the Apriori algorithm developed from Task 1 with different min-support thresholds.

- These datasets can be obtained from any online data portal. Some example data portal: https://archive.ics.uci.edu/ml/datasets.php.
- Yelp Challenge Data
- Wikipedia data (https://en.wikipedia.org/wiki/Wikipedia:Database_download)
- https://www.data.gov/
  US-centric agriculture, climate, education, energy, finance, health, manufacturing data, …
- https://cloud.google.com/bigquery/public-data/
  BigQuery (Google Cloud) public datasets (bikeshare, GitHub, Hacker News, Form 990 non-profits, NOAA, …)
- https://www.kaggle.com/datasets
  Microsoft-owned, various (Billboard Top 100 lyrics, credit card fraud, crime in Chicago, global terrorism, world happiness, …)
- https://aws.amazon.com/public-datasets/
  AWS-hosted, various (NASA, a bunch of genome stuff, Google Books n-grams, Multimedia Commons, …)

Requirement for the dataset: At least the size of one data is larger than your available memory. The purpose of the requirement if that **the implementation of the algorithm should be disk based and is able to handle dataset of large size, i.e., it cannot assume that the data can fit into memory**. As some of you asked what would be the size of data, it is suggested that the data should be at least 4GB for experiments.

In the report, you need to 1) document the dataset description, and the URL of the dataset where you download, and 2) report the running time results with different min-sup threshold, and the number of frequent itemsets, and the largest size of frequent itemsets.

Task 3: Use the frequent itemsets as features to do clustering with TWO clustering algorithms. You can use any clustering algorithm. You can choose two datasets with labels to evaluate the methods in part 3). Here you can use different data from the data you use for Task 1 &2.

In the report, you need to document how do you do clustering, and the performance (effectiveness) of the two clustering algorithms on the different methods.

Task 4:) Advanced part: Figure out an algorithm for improving the performance of the methods in Task 3. Give the pseudocode, and report and evaluation results. You need to evaluate your improved algorithms on the same set of datasets you used in part 3.

In the report, you need to explain the algorithm you design. You can give pseudocode and explain it line by line. You also need to document the classification results of your algorithm.

Weight of grading:

- Part 1: 20%
- Part 2: 20%
- Part 3: 40%
- Part 4: 20%

You are expected to improve your problem solving, deep thinking, and self-learning ability through the project, which are very important skills to acquire in universities.

What to deliver:

Report: The final report is up to 8 A4 pages (not necessary to write 8 pages. The page limit does not include front page). The report should include the results of Part 1—Part 4 of the project, i.e., those underlined parts.

In the end of report, please include the individual contribution claims (which should be agreed by all of you) in the following format:

Member name 1: list of contributions to the project.

Up to 5 minutes video. In the short video, you can capture screen to do a demo of your code to show it works, and you can also highlight any other part of your report in the video.

Project are done in groups. Discussions with other students are allowed, but each group has to write your own code.

Code submission + report +video: Due in the end of Thursday, 23 Nov. Only softcopy is required, and submission will be through NTUlearn.

Grading will consider report + code + video

NOTE:

1. **MOSS**: Sharing code with your classmates is not acceptable!!! All programs will be screened using the Moss (Measure of Software Similarity.) system.
2. **You are not allowed to share your project code on the web publicly**.

TA for projects:

- Liu Shuai (shuai004@e.ntu.edu.sg)

If you have questions, please email all the TAs above and cc to me. TAs can only provide some consultation for projects, but you should NOT expect TAs to help to do any part of your project.