CS 634 Data Mining
Midterm Term Project

Dr. Yasser Abduallah
Department of Computer Science
New Jersey Institute of Technology

General Submission Rules

- Embed your last name and first name in your project file name. For example, if your name is John Smith, your file name should read: smith_john_finaltermproj.zip

- Your project will automatically lose **10** points if the above submission rules are violated.

- Submit your project file in Canvas under Midterm Term Project Submission Site before the due time. The project file in Canvas is considered as the final version.

- No late project is accepted. A project is late if it is not submitted in Canvas before the due time. Zero points will be given to the late project.

- **NOTE: Pay attention to the project description front page in Canvas because it may have additional information and requirements, etc..**

Project Grading

- The grades will be posted on Canvas when they are completed.

- Note: There is a limit on the file size in Canvas and in NJIT's email box. So, keep your project file small to avoid any problem that may occur when submitting the file in Canvas.

- The project file must contain the **all source code (Jupyter if applicable and Python files) in running state, data sets,** and documentation including **screenshots**. The screenshots are used to demonstrate the running situation of your program, particularly how the program executes and produces output based on different input data and user-specified parameter values, if applicable.

- Project documentation/report should state how to run your program, any required packages and how to install them as if someone without any knowledge of can follow your instructions to replicate and run your program.

- The code should be running without any editing by the TA or me.

- Implementation, complexity of the code, code style, clarity of the report, and more are taken into consideration.

- Copying and sharing code with peers is prohibited and will result in 0 points for all parties that are involved.

- *Do not share or copy code from your peers or other resources. Your task is to implement the algorithm from scratch by yourself.*

Midterm Term Project Part 1

- This is a single person project. *Do not share or copy code from your peers.*
- Make sure to follow the submission rules when submitting your project.
- **The Project Part 1 Details**:
  1. Create 10 (or any number of, not less than 5) items usually seen in Amazon, K-mart, or any other supermarkets (e.g. diapers, clothes, etc.).
  2. Create a database of at least 20 transactions each containing some of these items. Save the transaction in a CSV file.
  3. Repeat (1) by creating 4 additional, different databases each containing at least 20 transactions.
  4. Note: You can create these transactions and datasets manually, download them from the net, or use the examples I will provide. In any case, add a note to your report where and how you built your data sets.
  5. The items and transaction must not be random so that your code is deterministic.

Midterm Term Project Part 2

- **The Project Part 2 Details**:
  - Implement the brute force method to generate frequent items and generate association rules.
  - The brute force method for finding frequent itemsets works as follows. Enumerate and generate all possible 1-itemsets and 2-itemsets. There are 30 items, so there are 435 possible 2-itemsets totally. Check to see whether each possible 1-itemset/2-itemset is frequent. Then enumerate and generate all possible 3-itemsets. There are 4060 possible 3-itemsets totally. Check to see whether each possible 3-itemset is :frequent. Keep on doing so until you see none of the possible k-itemsets is :frequent for some k, at which point the brute force method terminates without generating (k+ 1)-itemsets.

Midterm Term Project Part 3

- ✓ Use an existing Apriori implementation from Python libraries/packages to verify the results from your brute force algorithm implementation.
- ✓ Use Python existing package for fpgrowth (as known as fp-tree algorithm) to generate the items and rules.
- ✓ Compare the results from your brute-force, Apriori, and FP-Tree/Growth.
- ✓ Do the three algorithms produce the same results?
- ✓ Report the timing performance for all three algorithms as well.
- ✓ Which one is faster?

So, for all three algorithms, generate and print out all the association rules and the input transactions for each of the 5 transactional databases you created/used. The data set selection, support, and confidence *must* be user-specified parameters, so the output should show different rules with respect to different databases and different support/confidence.

Make sure to show multiple support and confidence results for each data set. You should prompt the user only once for the input and reuse for the three algorithms in each run.

The items and transactions must be clear and easy to identify. Your program should show the performance time for each algorithm.

Midterm Term Project Part 4

- Github & Jupyter Notebook.
  - After you finish your code in development and testing and make sure it works, and prepare the report (meaning all heavy lifting job is done 😊), Create a Github repository in https://github.com/. Your account must be with your NJIT email not your personal email (unless if you have to, but indicate that in your report as well).
  - Load your project to the repository.
  - Create Jupyter notebook for your work to show the output, for more info visit https://jupyter.org/
  - Give me ya54@njit.edu access as a collaborator to your repository. (If we have a grader, you give him/her access too).
  - Add Github link to your repository to your report.

NOTES:

- ✓ If you need help with Github and/or Jupyter notebook, let me know.
- ✓ Jupyter can be used as an IDE to edit and develop your code, but you must save it as Python at the end to generate the source code. Jupyter source code by itself will not be accepted as the project source code.
- ✓ Colab is not considered Github repository nor Jupyter notebook. You must create Jupyter notebook in Github.