Data Mining

Midterm Project

Dr. Yasser Abduallah, Professor

Department of Computer Science

New Jersey Institute of Technology

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_midtermproj.doc or <name>.pdf or <name>.zip, <name>.tar

Make sure to include the source code and the data set of your project as well as document any additional packages required to run your program.

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

This is an individual student project.

Submit your project file in Canvas under Midterm Project Submission Site before the due date and 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 date and time. Zero points will be given to the late project.

Late Project Policy

A project is late if it is not submitted to Canvas before the deadline. If you turn in your project *n* days late, your total point will be deducted by (50 × *n*) points. For example, suppose you turn in your project 1 day late (if you turn in your project after the deadline on the due date, it is also considered as 1 day late). Then, you lose (50 × 1) = 50 points automatically, and your total point is 50 points. Further, suppose you lose 10 points in documentation. Thus, you receive (50 –10) = 40 points in total.

For all late submissions of the project, they must be emailed to me at ya54@njit.edu
The email subject and the file name in the email must both be called
Lastname_Firstname_midtermproj.doc
(where you should fill in your last name and first name).

Note: Each student should submit one midterm project only. If the student has submitted his/her midterm project (even incomplete) in Canvas, the student is NOT allowed to send another midterm project to ya54@njit.edu . Your project will automatically lose 80 points if this rule is violated.

Midterm Project – Part 1

Create 30 items usually seen in Amazon, K-mart, or any other supermarkets (e.g., diapers, clothes, etc.).

(1) Create a database of 20 transactions each containing some of these items. The information can be stored in a file, or a DBMS (e.g., ORACLE).

(2) Repeat (1) by creating 4 additional, different databases each containing 20 transactions.

Using Apriori, generate and print out all the association rules and the input transactions for each of the 5 transaction databases you created (support and confidence should be user-determined parameter values, so the output should show different support and confidence values).

Midterm Project – Part 2

- Implement the brute force method and compare the brute force method with the Apriori algorithm on each of the 5 transaction databases you created. Present computation (CPU or clock) time to demonstrate that the Apriori algorithm is faster than the brute force method on each of the 5 transaction databases. The brute force method and Apriori algorithm should output the same association rules on each database.

- 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 Project – Part 2 Cont'd

- Use an existing Apriori implementation from Python libraries/packages to verify the results.
- Use the Python package/library as is to verify the result.
- You don't need to re-implement the algorithm. This is only for verification and sanity check.

Platforms

- ✓ **Programming language**:

  This is a **Python** based project. If you want to use a different programing language you must consult with me first.

- ✓ **Operating system is open:**

Any one of the following is allowed: Windows, Linux, Mac OS, Ubuntu etc.

- ✓ **Hardware is open:**

Any one of the following is allowed:

PC, Laptop, Mac etc.

If you know one language you can certainly learn another one very quick, it’s just matter of syntax differences

Project Grading

- There is a limit on the file size in Canvas. So, keep your project file small to avoid any problem that may occur when submitting the file in Canvas.

- The project file should contain the data, source code 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.

- **The data and source code must be physically presented and placed in the project file.** A link to some website containing the data/source code is NOT permitted.

- Both the Apriori algorithm and brute force method must be implemented from scratch. Using existing commands, libraries, or packages is NOT permitted.

Important Note: Your code must be in a working stage without any manual editing by me or the TA.