

Faculty of Science

**Course**: CSCI 2000U Scientific Data Analysis

**Tutorial:** #5 and #6

**Topic:** Frequent Itemsets

**Description**

Your task it to conduct market-basket analysis by developing the frequent itemsets algorithm. The goal is to find frequent pairs and triples of elements (and in some cases quadruples).

**Programming Language**

The implementation of the algorithm should be in the Python programming language. First implement a standard version of the frequent itemset algorithm that does not use the Apriori technique (**Tutorial 3**), see the SQL implementation from the lecture notes, Lecture 3.

Python installation:

https://www.python.org/downloads/

You can alternatively use online environment to avoid installation

http://www.tutorialspoint.com/execute\_python\_online.php

**Dataset**

The retail dataset contains the (anonymized) retail market basket data (~88200 baskets) from an anonymous Belgian retail store (retail.dat). The data are provided ’as is’.

Note that since the dataset was anonymized the preprocessing step to map text labels into integers is done for you. (Working with integers is more efficient than textual data as it is saves the main memory.)

Use Notepad++ or other software rather than Notepad to open the file for the correct formatting. Note that each row in the file represents different transaction.

You can find the datasets on Blackboard.

**Experiments**

Perform the scalability study for finding frequent pairs and triples of elements for retail dataset by dividing the data into the chunks and measuring the time performance. Provide the figures. You can choose the threshold on your own.

Report top 10 answers.

**Additional Task**

Extend the code to compute frequent itemsets of four items over the movie data, still using support threshold .03. If your code is correct, you should find 20 itemsets of two, 14 itemsets of three, and 3 itemsets of four. The movie dataset (*movies.dat*) contains information mapped to integers. (You can find information about movie in *movies.info*) Report if the number of found itemsets is the same as provided as a gold standard.

No scalability study is necessary for this task.

**Tutorial 4**

Implement Apriori version of the frequent itemset algorithm (see the SQL implementation of Apriori from Lecture 7.). Perform the scalability study experiment over the retail dataset and compare the results to the standard non-Apriori implementation.