CptS 315: Introduction to Data Mining
Programming Assignment 1
(PA1)

Instructions

- Supported programming languages: Python
- Store all the relevant files in a folder and submit the corresponding zipfile (.zip).
- Please include a ‘readme.txt’ file on full instructions to run your program, within this folder. A list of all libraries used should also be included in this file.
- This folder should have a script file named (Not needed if Windows OS used for development)
  run_code.sh
  - Executing this script should do all the necessary steps required for executing the code including compiling, linking, and execution
- Assume relative file paths in your code.
- Some examples
  - Unix/ Linux OS environments:
    ‘‘./filename.txt’’ or ‘‘../hw1/filename.txt’’
  - Windows OS:
    Keep data files within the same folder as your program(s).
- You should submit your zip file to Blackboard by the stated due date.

Programming Assignment Explanation

Product Recommendations: The action or practice of selling additional products or ser-vices to existing customers is called cross-selling. Giving product recommendation is one of the examples of cross-selling that are frequently used by online retailers. One simple method to give product

recommendations is to recommend products that are frequently browsed together by the customers.

Suppose we want to recommend new products to the customer based on the products they have already browsed on the online website. Write a program using the A-priori algorithm to find products which are frequently browsed together. Fix the support to s =100 (i.e., product pairs need to occur together at least 100 times to be considered frequent) and find item sets of size 2 and 3.

Use the online browsing behavior dataset provided with this homework. Each line represents a browsing session of a customer. On each line, each string of 8 characters represents the id of an item browsed during that session. The items are separated by spaces.

a) Identify pairs of items (X, Y) such that the support of {X, Y}is at least 100. For all such pairs, compute the confidence scores of the corresponding association rules: X⇒Y, Y⇒X. Sort the rules in decreasing order of confidence scores and list the top 5 rules in the writeup. Break ties, if any, by lexicographically increasing order on the left hand side of the rule.

b) Identify item triples (X, Y, Z) such that the support of {X, Y, Z}is at least 100. For all such triples, compute the confidence scores of the corresponding association rules: (X, Y) ⇒ Z, (X, Z) ⇒ Y, (Y, Z) ⇒ X. Sort the rules in decreasing order of confidence scores and list the top 5 rules in the writeup. Order the left-hand-side pair lexicographically and break ties, if any, by lexicographical order of the first then the second item in the pair.

c) **Output Format:**

• The output of your program should be dumped in a file named "output.txt" in the following format:

OUTPUT A
FRO11987 FRO12685 0.4325

FRO11987 ELE11375 0.4225
FRO11987 GRO94758 0.4125
FRO11987 SNA80192 0.4025
FRO11987 FRO18919 0.4015
OUTPUT B
FRO11987 FRO12685 DAI95741 0.4325
FRO11987 ELE11375 GRO73461 0.4225
FRO11987 GRO94758 ELE26917 0.4125
FRO11987 SNA80192 ELE28189 0.4025
FRO11987 FRO18919 GRO68850 0.4015

Explanation:

–Line 1 should have “Output A”
–Next five lines should have the top five rules with decreasing confidence scores for part (a) of the programming question. Format:

< item1> < item2> <confidence > meaning {item1} ⇒ item2

–Line 7 should have “Output B”
–Next five lines should have the top five rules with decreasing confidence scores for part (b) of the programming question. Format:
< item1> < item2> < item3> < confidence > meaning {item1, item2} ⇒ item3

• Unix/ Linux OS: Make sure the output.txt file is dumped when you execute the script

run_code.sh

- In Windows OS, your program should create this file, output.txt, within the same folder as your program and data.