Frequent Itemsets and A-Priori Algorithm

(from HW1 of https://web.stanford.edu/class/cs246/index.html, lecture by Jure Leskovec)

Application in product recommendations: The action or practice of selling additional products or services 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 online. 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 itemsets of size 2 and 3.

Use the online browsing behavior dataset from browsing.txt. 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. Some lines contain duplicste items. Removing or ignoring duplicates should not impact your results.

Two sanity checks are provided and they should be helpful when you progress:

(1) there are 647 frequent items after 1st pass ($|L_1| = 647$),

(2) the top 5 pairs you should produce in part (a) all have confidence scores greater than 0.985. See detailed instructions below.

(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 output.

Break ties, if any, by lexicographically increasing order on the left hand side of the rule.

(You need not use Spark for parts a) and b))

(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 output. 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.