Name: Ahmed Boutar NetID: aboutar URID: 31401418

Project1 report:

The requirement for this project is to implement the Apriori algorithm with one improvement. I chose to implement it while taking into account transaction reduction, where the program would ignore any itemset that has a support\_count less than the minimum support\_count , which I have set to be 1000 based on the count the 1-itemsets displayed.

I decided to get rid of the attributes: hours worked per week, total gain, and total loss. I realized that total gain and total loss don’t matter much in the scope of this database and the conclusions we are looking for. I decided to take off hours worked per week in order to avoid any confusion with the attribute age, whose elements are also numbers.

I followed the pseudo-code included in the book. However, the implementation is very different than that. In fact, the hardest thing is manipulating sets and dictionaries in Python, and alternating with them. The biggest obstacle that I found was that dictionaries can’t hold sets as keys. Therefore, I used dictionaries to hold the support\_count of the different itemsets, where the keys are a string representation of the itemset being dealt with. Sets are used for practically everything else. I had to use frozensets, which, in the essence, are the same as sets, but can be appended to another set. This is the only way that I found to create a set of sets, without having to use mapping.

The algorithm works as follows:

1. Generate the 1-frequent itemset by looping through all the elements of the database, storing their respective support counts in a dictionary, then store the keys (the itemsets) in an array after deleting elements that have a support count < 1000
2. The rest of the algorithm follows what the textbook mentions
   1. I have included a couple of other functions in order to facilitate the work. For example, the function find\_subsets takes returns all the subsets of a candidate, whose lengths are less than that of the candidate’s. The other function is subsets, which checks if a candidate is a subset of a transaction, increments the count in a dictionary, and returns it back to the main function.

After running this algorithm on the database adult.csv, we can conclude that the frequent type of workers is a worker that is white, male, earns over 50K, from the US, married, has a spouse that is alive, high-school grads, and work in the private sector. I think that these conclusions are not surprising except for the level of education as we would expected white males (most likely privileged) to have gone through some college.