COMS E6111 Advanced Database Systems
Fall 2015

Project 3

a. **Name and UNI**
Bahul Jain - bkj2111
Parth Parekh - prp2121

b. **A list of all the files that you are submitting**
run.sh - Shell script which calls run.py python script.
run.py - Main Python Script which runs the project.
Integrated-Dataset.csv – The dataset that contains the all the relevant columns from which we extract associations rules.
Legend.csv – Contains the legend for all the codes in the dataset.
test.db – The sqlite3 database that we have used to operate on the dataset

c. **Choice of Dataset**
Link: https://data.cityofnewyork.us/Housing-Development/Complaint-Problems/a2nx-4u46

Complaints received from building residents are one of the most common issues that a building management has to deal with. Deciding what kind of resources, and employees are required to maintain the condition of buildings and to address to most frequently occurring problems in such households. The dataset we chose contains a list of complaints regarding problems that happen in such households. Each of the complaints contain three important fields that can be used to extract some valuable information :

1. Space Type: Describes the area in the house where problem happens.
2. Type: The urgency of the problem.
3. Major Category: A high level description of the problem.

Finding associations among these attributes could help building management companies and landlords address some of the most frequently occurring household maintenance issues in more informed and efficient ways. This will also help in deciding the priority of the issues to be fixed and efficient job scheduling.

The integrated dataset is nothing but the original dataset with all the irrelevant columns removed. The only columns we have chosen for our analysis is:

$$SpaceTypeID \rightarrow space\_id$$
$$TypeID \rightarrow type\_id$$
$$MajorCategoryID \rightarrow category\_id.$$

From space_id column we eliminated ids – 58, 59, 62, and 65 since these ids represented the space types already represented by other ids. Also in the type_id column we eliminated type_id – 9 since it represents 'emergency' which was already represented by type_id – 1. Also these ids barely contribute to the entire dataset, hence we have not lost any valuable information.

We have formed a legend showing what every id represents actually.

d. **A clear description of how to run your program**
To run the program on clic machine, run the run.sh file using the command:
/home/bkj2111/ADBS/Project3/run.sh

If you are running the code from local computer, run the following command from the project directory:
python run.py

Note: All the parameters are taken in the command line after the program starts executing.
These parameters include: Integrated Dataset csv file, Minimum Support, Minimum Confidence.

e. **A clear description of the internal design of your project**
The program takes 3 inputs:
1. **Integrated Dataset CSV file** – This is the csv file which includes the dataset and associations are found out using Apriori algorithm from this file.
2. **Minimum Support** – Minimum support threshold which removes all the itemsets which have less than the minimum support as it is too weak to be considered.
3. **Minimum Confidence** – Minimum confidence threshold which is the ratio of support of the LHS of the association with the RHS of the association.

The first part of the implementation was to generate large itemsets. The second part involves checking for the associations in the final large itemset.

Part 1: Generation of large item sets
We first find out the candidate sets and then use a minimum support constraint to generate large itemsets. We do this step for all the possible lengths of the set. In our case, we have 3 attributes in the dataset, therefore we find the candidate sets for sets of size 1, 2 and 3.

To find out candidate sets, we do the following 2 steps:
A) **Merge**: In this step, we take all the possible combinations of the sets and perform the following steps on these combinations:
   a. These combinations are then merged in a set. So if we are working to find the candidate sets of size 3, we merge all the combinations of large item sets of size 2.
   b. Next step here is size check. If union (merge) of 2 sets of size 2 gives a set of size 4, it cannot be used for generating candidate sets of size 3 and therefore eliminated. Therefore size check validation helps in removing extra sets.
   c. Next validation check on the remaining sets is to remove duplicate sets. There is a possibility of duplicate sets after the merging step and therefore, duplicates are removed.
   d. The final check in this step is the duplicate column check. No set should have 2 values from the same column (attributes) and hence they are eliminated.

B) **Prune**: In this step, we work with the surviving sets from the merge step. Every set here is broken down into a set of size n-1 and all these sets are checked whether they belong in the large item set (L_{n-1}) of the previous iteration or not. If any of the combinations do not exist in the item set, then the set in consideration is pruned. The output of this step forms the candidate set for a particular size.

After the prune step, these sets are checked for minimum support, if a set passes this constraint, then it is added to the large item set for a particular size. All these large item sets for every size are merged to form a final large item set which ends the implementation of part 1.

Part 2) Forming associations.
We require only one item on the right hand side of the association. Therefore, for every item in the large item set produced in Part 1, we form combinations to check for all the possible associations and check whether an association satisfies the minimum confidence criteria. If a particular association satisfies the minimum confidence criteria, it is added to the high confidence association rules set.

f. Command line specification

We implemented the Apriori algorithm and passed 0.05 as the minimum support and 0.5 as the minimum confidence. The output is stored in 'example-run.txt' file. The results generated are interesting. One of the high confidence (99.7%) association rules states that if the space_type (area) is an 'entire building' and the category of the problem is 'heat/hot water', then it is an 'emergency' situation.

Another association states that if the area is 'kitchen', then it is generally not an emergency. These associations help in allocating resources and assigning priority to a particular problem or a complaint. If the complaint includes kitchen, then low priority and less number of workers is acceptable as it is not an emergency. However, for hot water related issues in the entire building, high priority should be assigned as it is an emergency.