Data Mining:Assignment 3

==========

Krishna Mahajan,0003572903

Q3

A)

Please find the code for both $F_{k-1}XF_{k-1}$ and $F_{k-1}XF_1$.
Module:aprioriGn.py

I maintained counter for count of candidate itemsets and count of frequent itemsets in actual **apriori** code which can be seen here
Module:apriori.py

Instructions to run the code can be seen in the attached readme.md file.

B)

In this problem i took up datasets from UCI repository which were categorical,numerical or both and correspondingly binarized the datasets removing redundant columns.I implemented binarization function in R.
Module:binarization.r

Dataset 1:Car Evaluation

This is completely categorical dataset with dimension $1728 * 7$. After removing some redundant columns i binarized this dataset and converted into csv file.(This binarized file can be seen as binarize_car.csv in ./Data/Car folder).
After executing apriori code on this dataset using the two candidate generation methods i got following results for three different threshold values of support.

$F_{k-1}XF_{k-1}$

support	candidate_itemsets	frequent_itemsets
0.01	190	84
0.1	78	18
0.3	12	1

$F_{k-1}XF_1$

support	candidate_itemsets	frequent_itemsets
0.01	281	84
0.1	64	18

support	candidate_itemsets	frequent_itemsets
0.3	12	1

All the corresponding frequent itemsets with the their corresponding support values can be seen in ./Code/Data/Car/support_*.csv files.

Dataset 2: Nursery store

This is completely categorical dataset with dimension $12960 * 9$.I binarized this dataset and converted into csv file.(This binarized file can be seen as binarize_car.csv in ./Data/Nursery folder).
After executing apriori code on this dataset using the two candidate generation methods i got following results for three different threshold values of support.

$F_{k-1}XF_{k-1}$

support	candidate_itemsets	frequent_itemsets
0.01	23880	7028
0.1	870	164
0.3	151	17

$F_{k-1}XF_1$

support	candidate_itemsets	frequent_itemsets
0.01	47372	7034
0.1	1550	164
0.3	159	17

All the corresponding frequent itemsets with the their corresponding support values can be seen in ./Code/nursery/Car/support_*.csv files.

Dataset 3:Groceries

This is actual transactional dataset of a belgian store and most practical dataset to apply Apriori Algorithm.This datasets has more than 9K transactions and over 500+ different items. After executing apriori code on this dataset using the two candidate generation methods i got following results for three different threshold values of support.

$F_{k-1}XF_{k-1}$

support	candidate_itemsets	frequent_itemsets
0.01	3927	217
0.1	178	5
0.3	168	0

$F_{k-1}XF_1$

support	candidate_itemsets	frequent_itemsets
0.01	4418	217
0.1	178	5
0.3	168	0

All the corresponding frequent itemsets with the their corresponding support values can be seen in ./Code/Data/Groceries/support_*.csv files.

As can be observed in the data $F_{k-1}XF_{k-1}$ generates less candidate itemsets and thus is more efficient than $F_{k-1}XF_1$

C)

Maximal Frequent Itemset : Frequent item set $X \epsilon F$ is maximal if it does not have any frequent supersets module:maximal_frequent_itemset.py

Closed Frequent Itemset : Frequent item set $X \epsilon F$ is closed if it has no superset with the same frequency. module:closed_frequent_itemset.py

Dataset 1:Car Evaluation

$F_{k-1}XF_{k-1}$

support	candidate_itemsets	frequent_itemsets	maximal_frequent	closed_frequent
0.01	190	84	40	78
0.1	78	18	9	18
0.3	12	1	1	1

$F_{k-1}XF_1$

support	candidate_itemsets	frequent_itemsets	maximal_frequent	closed_frequent
0.01	281	84	40	78
0.1	64	18	9	18
0.3	12	1	1	1

Dataset 2: Nursery store

$F_{k-1}XF_{k-1}$

support	candidate_itemsets	frequent_itemsets	maximal_frequent	closed_frequent
0.01	23880	7028	3294	2113
0.1	870	164	111	75
0.3	151	17	15	5

$F_{k-1}XF_1$

support	candidate_itemsets	frequent_itemsets	maximal_frequent	closed_frequent
0.01	47372	7034	3294	2113
0.1	1550	164	111	75
0.3	159	17	15	5

Dataset 3:Groceries

$F_{k-1}XF_{k-1}$

support	candidate_itemsets	frequent_itemsets	maximal_frequent	closed_frequent
0.01	3927	217	159	172
0.1	178	5	5	5
0.3	168	0	0	0

$F_{k-1}XF_1$

support	candidate_itemsets	frequent_itemsets	maximal_frequent	closed_frequent
0.01	4418	217	159	172
0.1	178	5	5	5
0.3	168	0	0	0

The above results confirm that **Candidate items > Frequent Itemsets > closed Frequent> Maximal Frequent**

d)

The code for confidence based rules can be found here
module:mining.py

Dataset 1:Car Evaluation

support	confidence	rules_generated	pruned_rules
0.01	0.25	175	59
0.01	0.50	175	10
0.01	0.75	175	3
0.1	0.25	16	11
0.1	0.50	16	8
0.1	0.75	16	3
0.3	0.25	0	0

support	confidence	rules_generated	pruned_rules
0.3	0.50	0	0
0.3	0.75	0	0

Dataset 2: Nursery store

support	confidence	rules_generated	pruned_rules
0.01	0.25	47372	7034
0.01	0.5	34462	754
0.01	0.75	34460	4
0.1	0.25	281	270
0.1	0.50	281	65
0.1	0.75	281	2
0.3	0.25	2	2
0.3	0.50	2	2
0.3	0.75	2	2

Dataset 3:Groceries

support	confidence	rules_generated	pruned_rules
0.01	0.25	296	75
0.01	0.5	296	2
0.01	0.75	296	2
0.1	0.25	0	0
0.1	0.50	0	0
0.1	0.75	0	0
0.3	0.25	0	0
0.3	0.50	0	0
0.3	0.75	0	0

e)

To generate the top 10 rules i sorted(max to min) rules based on confidence for each support value.
For a particular support value the rule with more confidence 'll be a better rule then the rule with lesser confidence.

Dataset 1:Car Evaluation
Top 10 rules for this dataset for each support and each threshold in above table can be in found in folder **(.Data/Car/rules/confidence)** where each csv files shows the top 10 rules.

Dataset 2: Nursery store
Top 10 rules for this dataset for each support and each threshold in above table can be in found in folder **(.Data/nursery/rules/confidence)** where each csv files shows the top 10 rules.

Dataset 3:Groceries Top 10 rules for this dataset for each support and each threshold in above table can be in found in folder **(.Data/Groceries/rules/confidence)** where each csv files shows the top 10 rules.

On my Observation for top 10 rules on Groceries dataset (Most practical dataset) i got following rules from the above support and confidence values

rules	support	confidence
'onions' ——> 'other vegetables'	1%	0.510638298
'beef' ——>,'whole milk'	1%	0.504716981
'curd'——>,'whole milk'	1%	0.491916859
'butter'——->,'whole milk'	1%	0.488272921
'root vegetables'——>'whole milk'	1%	0.478316327
'beef'——->,'other vegetables'	1%	0.466981132
'root vegetables'——>,'other vegetables'	1%	0.464285714
'domestic eggs'——->,'whole milk'	1%	0.45979021
'tropical fruit'——>'whole milk'	1%	0.447272727

All these rules seems to be very relevant.1% support seems low but considering this transaction are over 6 months duration 1% 'll be very significant. Rules for Other support Confidence combination can be found in csv files.

f)

Lift is one more parameter of interest in the association analysis. Lift is nothing but the ratio of Confidence to Expected Confidence,Using the above example, expected Confidence in this case means, "confidence, if buying A and B does not enhance the probability of buying C."For the supermarket example the Lift = Confidence/Expected Confidence.Hence, Lift is a value that gives us information about the increase in probability of the then (consequent) given the if (antecedent) part.
A lift ratio larger than 1.0 implies that the relationship between the antecedent and the consequent is more significant than would be expected if the two sets were independent. The larger the lift ratio, the more significant the association

The code for lift based rules can be found here
module:mining_lift.py

Dataset 1:Car Evaluation

support	Lift	rules_generated	pruned_rules
0.01	0.85	175	156
0.01	0.90	175	142
0.01	0.95	175	135
0.1	0.85	16	16
0.1	0.90	16	16

support	Lift	rules_generated	pruned_rules
0.1	0.95	16	12
0.3	0.85	0	0
0.3	0.90	0	0
0.3	0.95	0	0

Dataset 2: Nursery store

support	confidence	rules_generated	pruned_rules
0.01	0.85	54589	54295
0.01	0.90	54589	53654
0.01	0.95	54160	52493
0.1	0.85	281	281
0.1	0.90	281	281
0.1	0.95	281	277
0.3	0.85	2	2
0.3	0.90	2	2
0.3	0.95	2	2

Dataset 3:Belgian store

support	confidence	rules_generated	pruned_rules
0.01	0.85	296	296
0.01	0.90	296	296
0.01	0.95	296	296
0.1	0.85	0	0
0.1	0.90	0	0
0.1	0.95	0	0
0.3	0.85	0	0
0.3	0.90	0	0
0.3	0.95	0	0

To generate the top 10 rules i sorted(max to min) rules based on Lift for each support value.
For a particular support value the rule with more Lift 'll be a better rule then the rule with lesser confidence.

Dataset 1:Car Evaluation Top 10 rules for this dataset for each support and each threshold in above table can be in found in folder **(.Data/Car/rules/lift)** where each csv files shows the top 10 rules.

Dataset 2: Nursery store
Top 10 rules for this dataset for each support and each threshold in above table can be in found in folder

(.Data/nursery/rules/lift) where each csv files shows the top 10 rules.

Dataset 3:Groceries Top 10 rules for this dataset for each support and each threshold in above table can be in found in folder **(.Data/Groceries/rules/lift)** where each csv files shows the top 10 rules.