**ASSOCIATION RULES**

**Question 1: Association rules with Apriori**

The file ./specs/gpa question1.csv contains data scholar data related to a selected sample of students. There might be interesting rules that can be extracted from this file.

1. Filter out the count attribute as this will not be included in the rule

generation.

2. Use the Apriori algorithm to generate frequent itemsets from the input data. When doing so, only select frequent itemsets with a support of at least 15% (so, the minimum support should be 0.15). How many frequent itemsets are produced? How big are they? Include this information in your report.

3. Save the generated itemsets in ./output/question1 out apriori.csv,

making sure to include the support column.

4. Using these frequent itemsets, generate a first batch of association rules with a minimum confidence of 0.9. How many rules are produced? For each rule, include a short description in your report.

5. Save the generated rules in ./output/question1 out rules9.csv, mak-

ing sure to include the support and confidence columns.

6. Generate a second batch of association rules, but this time use a minimum confidence of 0.7. How many rules are produced this time? Again, shortly describe the outcome in your report.

7. Save the generated rules in ./output/question1 out rules7.csv in the

same format as the previous rule batch.

The file gpa1\_question1.csv contains data scholar data related to a selected sample of students.

Library used: pandas and mlxtend.

The input data is fetched from the input file using the pandas function read\_csv() and stored in a storage structure DataFrame named ‘dframe’.

***Q1a.*** Column named as count is dropped from the dataframe using the function drop() function of pandas and saved in dataframe itself.

***Q1b.***

* get\_dummies() of pandas is used to convert all the categorical columns to numerical values as some Machine Learning algorithms cannot operate on Categorical data. So the data should be binomial(0/1 or True/False).
* Apriori, an algorithm for frequent item set mining and association rule learning over relational databases. It identifies the frequent individual items in the database and extend them to larger item sets as long as those item sets appear sufficiently often in the database.
* The support of an itemset X, supp(X) is the proportion of transaction in the database in which the item X appears. It signifies the popularity of an itemset.
* Apriori algorithm is applied on the dataframe with min support 15% and frequent itemsets are generated. Total frequent itemset are 20. Every itemset consists of 2 unique itemset.

***Q1c.*** A new csv file is saved having support and their corresponding itemset using the pandas function to\_csv() with arguments: File path and indexing as false(to ignore the indexing in the file) named question1 out.csv.

***Q1d.***

* Association rule learning is a rule-based machine learning method for discovering co relations between itemsets.
* Confidence signifies the likelihood of item Y being purchased when item X is purchased.
* Association rule with minimum support 90% are generated using association\_rule() of mlxtend-> frequent\_patterns. Total rules generated are 1. It signifies that if age is in the range (21-25) then the status will be most likely Junior with support 16%.

**Q1e.** A new csv file is saved having support, confidence and antecedent, consequent using the pandas function to\_csv() with arguments: File path and indexing as false(to ignore the indexing in the file) named question1 out\_rules9.csv.

**Q1f.** Another association rule with minimum support 70% are generated using association\_rule() of mlxtend-> frequent\_patterns. Total rules generated are 3.

* Major\_Philosophy -> Age(26-30). It signifies that if Major is Philosophy then the age will be most likely in the range(26-30) with support 20% and confidence(assurance that Age will be 26-30 if major is Philosophy) as 71%.
* Status\_PhD-> Age(26-30). It signifies that if status is PhD then the age will be most likely in the range(26-30) with support 16% and confidence(assurance that Age will be 26-30 if status is PhD) as 80%.
* Age(26-30)->Status\_Junior. It signifies that if age is in between (26-30) then the status will be most likely Junior with support 16% and confidence(assurance that status is Junior if Age is in be 26-30) as 100%.

**Q1g.** A new csv file is saved having support, confidence and antecedent, consequent using the pandas function to\_csv() with arguments: File path and indexing as false(to ignore the indexing in the file) named question1 out\_rules7.csv.

===============================================================================

**Question 2: Association rules with FP-Growth**

The file ./specs/bank data question2.csv contains customer records from the marketing department of a financial firm. The data contains the following fields:

id : a unique identification number

age : age of customer in years (numeric)

sex : MALE / FEMALE

region : inner city / rural / suburban / town

income : income of customer (numeric)

married : if the customer is married - YES / NO

children : number of children (numeric)

car : if the customer owns a car - YES / NO

save acct : if the customer has a saving account - YES / NO

current acct : if the customer has a current account - YESY / NO

mortgage : if the customer has a mortgage - YES / NO

pep : if the customer signed for a Personal Equity Plan after the last mailing YES / NO

1. Filter out the id attribute as this will not be include in the rule generation.

2. Discretize the numeric attributes into 3 bins of equal width, the filter out

the original attributes.

3. Use the FP-Growth algorithm to generate frequent itemsets from the data. When doing so, only select frequent itemsets with a support of at least 20% (so, the minimum support should be 0.2). How many frequent itemsets are produced? How big are they? Include this information in your report.

4. Save the generated itemsets in ./output/question1 out fpgrowth.csv

5. Using the obtained frequent itemsets, generate association rules. Experiment with different confidence values, selecting a value that produces at least 10 rules. What is this value? Include it in your report.

6. Save the generated rules in ./output/question2 out rules.csv

7. Select the top 2 most interesting rules and for each specify the following in your report:

-an explanation of the pattern and why you believe it is interesting based on the business objectives of the company;

-any recommendations based on the discovered rule that might help the company to better understand behavior of its customers or in its marketing campaign.

The file ./specs/bank\_data\_question2.csv contains customer records from the marketing department of a financial firm.

Library Used: frequent\_patterns->fpgrowth (mlxtend)

The input data is fetched from the input file using the pandas function read\_csv() and stored in a storage structure DataFrame named ‘dframe\_Q2’.

***Q2a.*** Column named as id is dropped from the dataframe using the function drop() function of pandas and saved in dataframe itself.

***Q2b.*** There are 3 numerical attributes in the dataframe(age,children and income). All are discretized using the cut() of pandas into 3 bins of equal width. Further the columns Age, Income and Children in the dataset are replaced using the binned one.

***Q2c.*** The FP-Growth Algorithm is another way to find frequent itemsets without using candidate generations, thus improving performance. Frequent itemsets are generated with a minimum support value 20%. Total frequent itemset are 231. The itemset consists of max 4 unique itemset.

***Q2d.*** A new csv file is saved having support with corresponding itemset using the pandas function to\_csv() with arguments: File path and indexing as false(to ignore the indexing in the file) named question2\_out\_fpgrowth.csv.

**Q2e.** Association rule with that generates at least 10 rules using association\_rule() of mlxtend-> frequent\_patterns. A confidence value of 80% yields 8 rules. Confidence less than 80% yields a minimum of 10 rules. So the value of confidence is 0.7 that yields 119 rules.

**Q2f.** A new csv file is saved having support, confidence and antecedent, consequent using the pandas function to\_csv() with arguments: File path and indexing as false(to ignore the indexing in the file) named question2\_out\_rules.csv.

**Q2g.** The top 2 interesting rules are:

1. sex\_FEMALE and pep\_NO -> current\_act\_YES

This is interesting because the lift value here (0.99) is < 1 as it implies that having a current account is independent of the antecedents (female with No Equity Plan).

1. current\_act-YES and age(17.951-34.333) -> income(4956.094, 24386.1)

This is interesting because the confidence value here (71%). It implies that there are 71% likelihood that if you have current account, having age in between 17 and 34 approx then you must have income in between 4956.094 and 24386 approx. Although this is not the case practically.

* Recommendation based on the discovered rules: Instead of sex\_FEMALE and pep\_NO -> current\_act\_YES, Association could be sex\_FEMALE and pep\_YES-> married\_YES. OR car\_YES and married\_YES So, pep\_YES.

===============================================================================