COMP40370 Practical 3

Association Rules

Prof. Tahar Kechadi

Academic year 2019-2020

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`, making 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.

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. When doing so, only select frequent itemsets with a support of at least 20% (so, the minimum support should be 0.2).
3. Use the **FP-Growth** algorithm to generate frequent itemsets from the data. 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.

    Note: The top 2 most interesting rules may not be the top 2 rules in the result set. They are rules that provide some non-trivial, actionable knowledge based on the underlying business objectives.

Data files

- `./specs/gpa_question1.csv`: data file
- `./specs/bank_data_question2.csv`: data file
- `./specs/test_practical3.py`: Python test file to check your solutions

Expected output and submission data

Your submission should be a single archive file (zip, tar, tgz, ...) containing one folder called `output` and the following files and directories:

- `./run.py`: main Python script
- `./report.pdf`: your PDF report (2 pages maximum)
- `./output/question1_out_apriori.csv`: frequent itemsets for first question
- `./output/question1_out_rules9.csv`: association rules for first question (confidence value of 0.9)
- `./output/question1_out_rules7.csv`: association rules for first question (confidence value of 0.7)
- `./output/question2_out_fpgrowth.csv`: frequent itemsets for second question
- `./output/question2_out_rules.csv`: association rules for second question
- `./specs/`: the original `specs` folder included in the assignment archive, containing the input data and the test file

The final deadline for the submission is **Monday, 13th of October**, 2019,at **17:00**. You can submit your solution on Brightspace.

Programming requirements and tools

The assignment should be solved in Python, version 3.5 or above (3.7 is recommended). You shall use the following packages for this assignment:

- `pandas` 0.25+
- `mlxtend` 0.17

The documentation of `mlxtend` can be found here: http://rasbt.github.io/mlxtend/

In particular, the following user guides are available for the required algorithms of the assignment:

- Apriori: http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/apriori/
- FP-Growth: http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/fpgrowth/
- Association rules: http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/association_rules/

Keep in mind that you can save the dataframes generated by `apriori`, `fpgrowth` and `association_rules` straight into csv files.

When handling the data to feed into the `mlxtend` functions, remember that all the attributes should be binomial. You can take a look at the `pandas` `get_dummies` function, in case you do not want to use the encoders provided by `mlxtend`.