Data Mining Assignment 1 – Frequent Pattern Mining

Problem Setting: You have received the dataset 'income_levels.xlsx'. This data consists of information of individuals from the US, aged between 17 and 93 years old. Along the age of these people the data contains information about their education level, their occupation, their marital status and so forth. Some Sociology researchers have hired you to analyse this data for them. In the basic data exploration phase you will use frequent pattern mining algorithms to do so.

Task 1: Data Inspection and Preparation

a) Before you start doing any analysis on the data at hand it is crucial that you understand it. On a later page of this document you can find description of all the columns in the data, make sure to have read it before continuing with the rest of the assignment

b) In the data there are some missing values. Based on the given data description, write code that imputes these missing values in a way that you find fitting. Describe your reasoning in the report.

c) As you can see the features "age", "workinghours" and "education" have a wide range of possible values. It can be useful to *bin/categorize* these features, meaning that you can group values like "age = 42" and "age = 44", into one overarching category like "age = Between 40 and 45". Later, when you apply some frequent pattern mining algorithm on the data, this will help in extracting a more concise set of patterns. You can find more about this kind of Data Preprocessing in your textbook or online. Apply some data categorization on the features "age", "workinghours" and "education" and provide in your report motivation behind your chosen categories.

Note, that there is no clear "right" way of choosing your number/types of categories. It is more important that you can motivate your choices and are aware of their possible disadvantages.

Task 2: Search for Association Rules

You are now going to explore the dataset by finding frequent patterns that occur in it. Use an algorithm like Apriori to extract frequent itemsets and generate association rules with high support and confidence. You can implement some algorithm yourself, but we encourage you to make use of existing implementations/libraries like apyori for Python (https://pypi.org/project/apyori/)

a) Play around with the algorithm and run it for different values for the "minimum support" and "minimum confidence" you want your rules to have. What do you notice about the number of rules you can find and the nature of these rules. Describe and explain your findings in the report!

b) The Sociology researchers that have hired you are interested to study the population differences in Men and Women regarding the type of work they do, their education level, their income level, etc. Extract rules that have "sex = Male" or "sex = Female" as their

consequence. Find rules with high support, high confidence or both and describe some patterns that you have found (for each sex at least 3 distinct ones). Which associations are there between the sex of people and the other features in the data? Which of these patterns did you expect? Do you find the "support" of rules more informative or their "confidence"? Do you notice any differences between the rule support/confidence with "sex = Male" or "sex = Female" in their consequence? Describe your findings in the report

Some Notes on your Code: You may implement your code in any programming language of your choice, but we do recommend Python as this is the most common language used for Data Mining tasks. We encourage you to use programming libraries, both for data preprocessing and for the implementation of the apriori algorithm. You might find these libraries useful:

- `pandas` for data preprocessing
- `apyiori` for an implementation of the apriori algorithm
- `matplotlib` or `seaborn` for generating figures

Handing in instructions: Put all your findings in a report of about 2-3 pages (there is no hard page limit, but be concise). The deadline for handing in your report and the code is 29 March. Please upload both in a zip-file on Blackboard.
Be clear and concise in your writing and refer to the grading rubric on the last page to see what is expected from you!

Questions: In case you have any questions specific to the assignment, please send an email to daphne.lenders@uantwerpen.be

For any general course related questions, or questions about association rules please refer to the lecturers.

Data Description

age – the age of an individual person, ranges between 17 and 93

workclass – this describes the workclass of a person (e.g. governmental or private)

education – this column ranges between 1 and 24 and describes the numeric encoding of the highest education level the person has received. This is what the numeric codes mean:

01 - No schooling completed
02 - Nursery school, preschool
03 - Kindergarten
04 - Grade 1
05 - Grade 2
06 - Grade 3
07 - Grade 4
08 - Grade 5
09 - Grade 6
10 - Grade 7
11 - Grade 8
12 - Grade 9
13 - Grade 10
14 - Grade 11
15 - 12th grade - no diploma
16 - Regular high school diploma
17 - GED or alternative credential
18 - Some college, but less than 1 year
19 - 1 or more years of college credit, no degree
20 - Associate's degree
21 - Bachelor's degree
22 - Master's degree
23 - Professional degree beyond a bachelor's degree
24 - Doctorate degree

marital status – the marital status of the person

occupation – describes the occupational sector of a person (e.g. Health or Education)

workinghours – numeric measure of a person's average workinghours per week, ranges from 1 to 99

sex – a person's sex (unfortunately, the collectors of this dataset didn't consider non-binary sexes)

ability to speak english – numeric encoding of a persons' ability to speak English (only given for non-native English speakers)

blank – Not Applicable (person is a native English speaker)
1 – very well 2 – well 3 – not well 4 – not at all

gave birth this year – describes whether the person gave birth to a baby over the last 12 months

income – describes if a person has a high or low income (whereas low income means an income of less than 50.000USD a year)

	Less than 7	7 to 10	10 to 13	14 to 17	18 to 20
Layout	Layout of the report is sloppy, report is hard to read, there are many typos		Layout is okay, but at some parts there are typos or section headings, figure titles etc. are missing	The layout of the report is good, there are barely any typos	The layout is excellent
Writing Style	The report is very confusing; the writing style is below average	At places the report is not very clear; there is a lack of clear structure	The text is overall clear although some parts could be improved.	Most of the text is easy to follow, findings are explained in a clear way	The text is very clearly and concisely written, illustrative examples and figures are not overused, but added where needed
Task 1	Task was not completed	Data was preprocessed in an unlogical way, no clear motivation behind preprocessing choices given	Data was preprocessed in an okay way, but motivation behind choices is missing/not very clear	Preprocessing was done well and motivation (as well as possible disadvantages of the chosen approach) are well explained	Preprocessing was done well and motivation behind choices is excellent (e.g. backed up by statistical measures/figures)
Task 2 a)	Task was not completed	There are some errors in the implementation, not many observations are made about the effect of 'min_support' and 'min_confidence' on the generated rules	The algorithm has been implemented correctly, some basic observations about the effect 'min_support'/'min_confidence' are given	Correct algorithm implementation, the student gives interesting observations about the effect 'min_support'/' min_confidence' and shows clear understanding about why effects occur	Correct algorithm implementation; analysis of results are excellent and also some Figures and illustrative Examples are also provided
Task 2 b)	Task was not completed	There are some mistakes in the execution of the task, analysis of the results is minimal	Task has been executed correctly. Basic analysis of the results is given	Task was executed correctly, the student gives a good motivation for their selection of 'interesting'	Task was executed correctly, and analysis of results is excellent. The discussion even goes

				rules, rest of analysis is interesting as well	beyond the questions that were asked in the assignment
Code	Code raises many errors	Code raises some errors or is very unclear	Code runs but lacks clear structure and readability, only little documentation is given	Code is readable and sufficiently documented	Code is very readable and well-documented. It is structured in a way that only by (un)commenting single lines, the code for the different tasks can be run