Ton Duc Thang University
Faculty of Information Technology

MIDTERM ESSAY

Course: Mining Massive Data Sets

Duration: 03 weeks

I. Formation

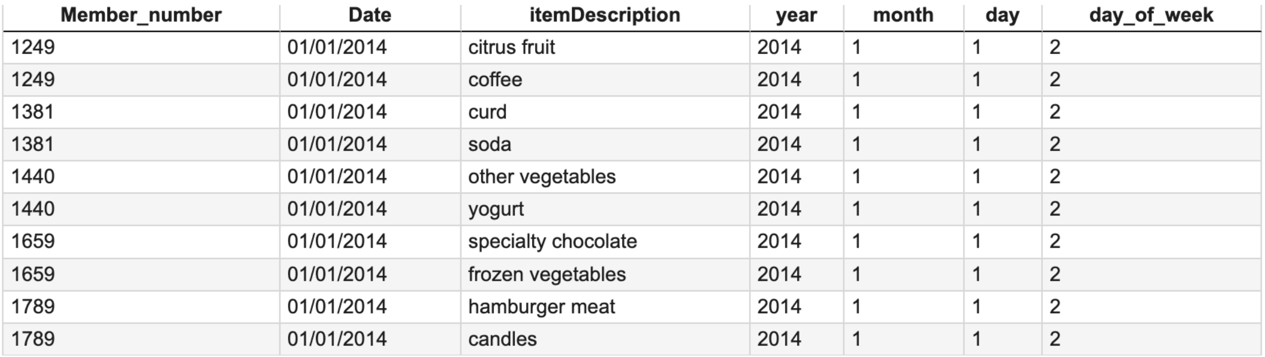
- The essay is conducted in groups with 03-05 students.
- Student groups conduct designated tasks and submit the essay by the deadline.

II. Requirements

Given **baskets.csv** file, consisting of shopping data, in which the first row is header and the remaining ones are records.

- **Member_number**: customer number
- **Date**: date in dd/mm/yyyy
- **itemDescription**: product name
- **year**: year
- **month**: month
- **day**: day
- **day_of_week**: day of week

For example,



Member_number	Date	itemDescription	year	month	day	day_of_week
1249	01/01/2014	citrus fruit	2014	1	1	2
1249	01/01/2014	coffee	2014	1	1	2
1381	01/01/2014	curd	2014	1	1	2
1381	01/01/2014	soda	2014	1	1	2
1440	01/01/2014	other vegetables	2014	1	1	2
1440	01/01/2014	yogurt	2014	1	1	2
1659	01/01/2014	specialty chocolate	2014	1	1	2
1659	01/01/2014	frozen vegetables	2014	1	1	2
1789	01/01/2014	hamburger meat	2014	1	1	2
1789	01/01/2014	candles	2014	1	1	2

baskets.csv (displayed in Google Colab)

a) **Task 1 (4.0 points): RDD**

- Use **RDD** of PySpark library to read **baskets.csv**. Then **implement**, **execute**, **save**, and **visualize** results of the following functions.

Function	*Input*	*Output*	*Processing*
f1	Path to **baskets.csv**	Print results on the screen and save them to folder **f1**	Find the list of distinct products. Results are sorted in the ascending order of product names. Print down 10 frist and 10 last products in the resulting list.
f2	Path to **baskets.csv**	Print results on the screen and save them to folder **f2**	Find the list of distinct products and their frequency of being purchased. Results are sorted in the descending order of frequency. Select top 100 products with the highest frequency, draw a bar chart to visualize their frequency.
f3	Path to **baskets.csv**	Print results on the screen and save them to folder **f3**	Find the number of baskets for each member. A basket is a set of distinct products bought by a member in a date. Results are sorted in the descending order of number of baskets. Select top 100 members with the largest number of baskets, draw a bar chart to visualize their number of baskets.
f4	Path to **baskets.csv**	Print results on the screen and save them to folder **f4**	Find the member that bought the largest number of distinct products. Print down the member number and the number of products. Find the product that is bought by the most members. Print down its name and the number of members.

- Note: do not use DataFrame in any ways and do not print down too much information in a single output cell to avoid being hidden.

b) Task 2 (2.0 points): DataFrame

- Use DataFrame (PySpark) to find out the list of baskets. A basket is a set of products bought by a member in a date. Resulting baskets are sorted in the ascending order of year, month, day.
- With the resulting DataFrame, find the number of baskets bought in each date. Draw a line chart to visualize the result.
- Save the resulting baskets in the folder **baskets.**

c) Task 3 (3.0 points): PCY

Use PySpark library to implement the PCY class to perform the corresponding algorithm.

- Constructor: receives a path to a file consisting of baskets from task 2; constant **s** is the support threshold (i.e., s = 0.3); constant c is the confidence threshold (i.e., c = 0.5).
- run(): run the algorithm. After that,
  - Save the resulted DataFrame consisting of frequent pairs to **pcy_frequent_pairs.csv**
  - Save the resulted DataFrame consisting of association rules to **pcy_association_rules.csv**.
  - Schemas of DataFrames are based on the one of **FPGrowth**.
- Note:
  - Source code must follow big data principles, avoid installing functions that contain pure data in the main memory.
  - Students may implement additionaly attributes and methods to support your work. However, ensure that source code are compact and optimal.
  - Do not use any libraries directly providing PCY implementation.

d) Task 4 (1.0 points): Report

- Student groups compose a report.
- **THERE IS NO TEMPLATE. STUDENTS ARANGE CONTENTS IN A LOGICAL STRUCTURE BY YOURSELVES.**
- The report must include below contents

  - Student list: Student ID, Full name, Email, Assigned tasks, Complete percentage.
  - Briefly present approaches to solve tasks, should make use of pseudo code/diagrams.
  - Avoid embedding raw source code in the presentation.
  - Study topics are introduced briefly with practical examples.
  - Advantages versus disadvantages
  - A table of complete percentages for each task.
  - References are presented in IEEE format.
- **Format requirements:** avoid using dark background/colorful shapes, students ensure contents are clear enough when printing in grayscale.

III. Submission

- Create a folder whose name is in the format **midterm_<Group ID>**:
  - **source.ipynb:** source code of the essay
  - **source.pdf:** exported ipython notebook.
  - **report.pdf:** report of the essay.
- Students maintain outputs of all cells in both .ipynb and .pdf files.
- Compress the folder into a zip file and submit by the deadline.

IV. Policy

- **Student groups submitting late get 0.0 points for each member.**
- **Copying source code on the internet/other students, sharing your work with other groups, etc. cause 0.0 points for all related groups.**
- **If there exist any signs of illegal copying or sharing of the assignment, then extra interviews are conducted to verify student groups' work.**

-- THE END --