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

III. Requirements

1) Task 1 (3.0 point(s)): A-Priori Algorithm for Frequent Customers

- Store data in HDFS and then implement a Hadoop MapReduce program (Java) to discover groups of customers going shopping in the same date.
- Implement the A-Priori algorithm to identify frequent customer pairs in form of 02 Hadoop MapReduce programs (Java), each one corresponding to a pass.

2) Task 2 (3.0 point(s)): PCY Algorithm for Frequent Items

- Store data in Google Drive, using PySpark DataFrames to identify baskets (sets of items bought a customer in a date).
- Implement the PCY algorithm to identify frequent pairs and generate association rules based on a given support threshold s and a confidence threshold c. Describe, in details, the hashing function and bucket management.
- The algorithm is organized in form of OOP classes to support software deployment. Refer to the FPGrowth class of PySpark for examples.

3) **Task 3 (3.0 point(s)): MinHashLSH for Similar Dates**
   - Store data in Google Drive and use PySpark DataFrames for this task.
   - Two dates (day, month, year) are said to be similar if and only if they share at least *50%*, Jaccard similarity, distinct items bought by customers.
   - ***Approach 1:*** Students apply the MinHashLSH algorithm to discover similar pairs of dates whose Jaccard similarity is at least *s* provided as a parameter. `pyspark.ml.feature.MinHashLSH` is allowed in this case to avoid manual implementation.
   - ***Approach 2:*** Students implement another approach in which all possible pairs of dates are evaluated to discover similar pairs of dates as mentioned above.
   - Finally, draw a chart to contrast the running time of the two approaches for *s* in range [0.0, 1.0], step = 0.1.

4) **Task 4 (1.0 point(s)): Report**
   - Student groups compose the project report using the IEEE conference proceeding template.
   - Recommended editor: Overleaf.
   - Selective contents:
     - *Title*: the project title
     - *Authors*: group member's information, the lecturer is appended as the last author.
     - *Abstract*: summarize the project requirements, approaches, experimental results, and levels of completion.
     - Each following section presents a task in the project, with a meaningful and human-readible title. Briefly introduce the approach to tackle the problem and illustrate results with related figures/tables, etc.
     - "*Contributions*" section: individual tasks, individual completion levels (0%-100%).
     - "*Self-evaluation*" section: self-evaluate task completion and estimate scores.
     - "*Conclusion*" section: summarize the project requirements, approaches, experimental results, and levels of completion.
   - References are in the IEEE format.
   - Maximal length is 05 pages.

IV. Submission Notice

- Create a folder whose name is like **midterm_<Group ID>_<Your Student ID>**:
  - **Source/:** consists of the project source code, each task is implemented in an individual sub-directory, preserving the outputs of all cells in ipynb files, output files as well.
  - **Report/:** report source (exported from Overleaf), **report.pdf** file.
- Compress the folder as a zip file and submit by the deadline.

- Every member submits the project to the elearning system.

V. 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.**
- **Evaluation scores of individual tasks are only recorded if and only if the student group give a reasonable presentation and justification to avoid cheating by AI tools, rental of doing the project, imbalance contributions, missing discussing, cooperating of group members in the project, etc.**
- **AI tools are forbidden in the project.**

-- THE END --