ENDTERM PROJECT

Course: Mining Massive Data Sets

Duration: 05 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. Datasets

Data sets	Description
ratings2k.csv *(Download here)*	Product rating data set. The first line is the header. • **index**: row index • **user**: user ID • **item**: product ID • **rating**: rating (0.0-5.0) 2365 remaining lines are data samples.

III. Requirements

1) Task 1 (3.0 point(s)): Dimensionality Reduction – SVD

Implement the task in **Task01.ipynb.**

Represent each user, in the given data set, as a sparse vector of ratings.

Use a PySpark module, for example, **RowMatrix**, to discover **32** "strongest" concepts in the data set based on the SVD algorithm.

Each concept is assigned an ID in the range [0, 31].

Infer the concept ID for each user and each item. Results are stored in the two data frames: **df_concept_user** and **df_concept_item.**

Compute the portion of each concept based on the number of users and based on the number of items. Results are stored in the data frame **df_concept_portion**.

Compute the embedding of users (using U, Σ, V) and then store them in the data frame **df_embedding_user** for the next task.

2) Task 2 (3.0 point(s)): Clustering – CURE

Implement the task in **Task02.ipynb**.

Use **PySpark** to implement the Clustering Using Representatives algorithm (CURE) to cluster user embeddings in Task 01 (**df_embedding_user**).

Students implement an OOP class for the algorithm and conduct experiments with the numbers of representatives in the range [3, 8]. For each value, compute the average distance from data points the nearest representative of the cluster and then draw a bar chart to illustrate the result.

3) Task 3 (3.0 point(s)): Recommender Systems - Collaborative Filtering

Implement the task in **Task03.ipynb**.

Create an OOP class to implement the Colaborative Filtering algorithm for item recommendation using PySpark, given the data set in **rating2k.csv**. Note that each user is representated as a sparse vector of ratings.

- The constructor takes in the value of N (number of similar users) and the data set as a data frame (PySpark).
- The function predict() takes in a user (a vector of ratings) and the expected number of recommended items. It returns a data frame (PySpark) consisting of recommended items sorted in the descending order of scores.

4) Task 1 (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 **endterm_<Group 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.

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.**

-- THE END --