ENDTERM PROJECT

Course: Mining Massive Data Sets

Duration: 06 weeks

I. Formation

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

II. Requirements

1) Task 1 (2.0 point(s)): Hierarchical clustering in non-Euclidean spaces

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

Data

- Generate a data set of about 10000 alphabetical strings whose lengths are in the range [32, 64]. Cases are not sensitive.
- Apply the 4-shingle technique to tokenize each string into a set.
- Jaccard distance is used to measure shingle sets' dissimilarity.
- Save the generated data set in a csv file with columns: *index, string, shingles*.

Algorithms

- Implement the agglomerative algorithm (bottom-up), in form of a class, to cluster shingle sets above (each shingle set is a sample in a non-Euclidean space)
  - Algorithm structure: agglomerative clustering
  - Data space: non-Euclidean
  - Memory usage: **in-memory** algorithm
- Cluster representation and cluster distance approaches follow **the Approach 1** presented in the corresponding lecture.
- Termination condition is one of the techniques mentioned in the same lecture.

Experiments

- Conduct experiments using the implemented class and the generated data set above.
- For each cluster, compute the average distance from each sample to its clustroid and then draw a bar chart to illustrate the result.

2) Task 2 (2.0 point(s)): Linear Regression – Gold price prediction

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

Data

- Given a data set of gold prices in Vietnam, collected from 2009/08/01 to 2025/01/01, in the **gold_prices.csv** file.
- Columns include [*Date, Buy Price, Sell Price*].
- Use PySpark to read the given data set and transform it into a data frame.
- Generate samples with features and labels as below
  - Features: gold prices of 10 consecutive previous dates of the date *t*,
  - Label: the gold price of the date *t*.
- Divide samples randomly into two sets, including training and test, with the ratio 7:3.

Algorithms

- Students are allowed to use the existing Linear Regression model of PySpark instead of manually implementing.

```
pyspark.ml.regression.LinearRegression
```

Experiments

- Train and evaluate a linear regression model, using PySpark, in the two sets mentioned above.
- Draw a line chart to illustrate losses during the training process.
- Draw a bar chart to contrast the results in the training and test sets.

3) Task 3 (2.0 point(s)): CUR – Dimensionality Reduction

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

Data

- Reuse samples (*features, label*) from Task 2

Algorithms

- Use PySpark to implement the CUR algorithm, in form of a class, to reduce dimensionality of feature vectors from 10 to 5.
- Students are allowed to use the RowMatrix libray of PySpark

```
pyspark.mllib.linalg.distributed.RowMatrix
```

Experiments

- Infer the new representation (row embedding) for each feature vector in the training and test sets.
- Training and evaluate a linear regression model as in Task 2 with the two new sets.
- Draw a bar chart to contrast losses between the new and original sets.

4) Task 4 (3.0 point(s)): PageRanking – the Google algorithm

Implement the task in **Task04.ipynb**.

Data

- Given the source webpage: https://it.tdtu.edu.vn
- Crawl all sub-webpages in the given domain and store them in form of pairs

<source page URL, destination page URL>

  - **source page** → a web page in the domain
  - **destination page** → an out-link in the source page
  - Notes: excluding all static files such as images, videos, css, js, etc.
- Store URL pairs in a data frame of PySpark (large data), which is considered as an edge list representing a directed graph.

Algorithms

- Use PySpark to implement the Google algorithm for page ranking, in form of a class, according to the pseudo code in the corresponding lecture.

Experiments

- Execute the algorithm in the collected data set to discover the importance of individual pages, sort them in the descending order to figure out the most important page.

5) Task 5 (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-readable 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.

III. Submission Notice

- Create a folder whose name is like

endterm_<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 team member must submit the project individually.

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.**
- **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 --