INFO-H515 - Big Data
Distributed Data Management and Scalable Analytics
Project 2021-2022

Théo Verhelst, Daniele Lunghi, Antonios Kontaxakis
Gianluca Bontempi, Dimitris Sacharidis

April 2022

The project counts for 50% of your final grade (i.e. 10/20). This project has to be developed by a team of 3 students registered to the class. Any project returned by a team composed by a different number of students will not be considered. The project shall be completed independently and it shall represent the sole efforts of the team submitting the assignment. The result of another team efforts, or the copy of another team efforts (current, or past, semester(s)), is considered academic dishonesty and will be punished accordingly.

1 Goal

The goals of the project are:

- To participate to the RecSys 2022 Challenge on recommender systems by implementing and assessing a recommender system pipeline in a distributed and scalable way.
- To implement data preprocessing, feature engineering, feature selection, predictive modeling, assessment of the impact of feature selection, and use the implemented pipeline to submit to the RecSys competition.
- To report your analyses and results as a Jupyter notebook.

2 RecSys challenge

The RecSys 2022 challenge focuses on fashion recommendation for online shopping. It is organized by Dressipi (an AI company focusing on fashion), B. Ferwerda (Jönköping University), S. Kalloori (ETH Zürich), and A. Srivastava (IIM Jammu). Given user sessions, purchase data and content data about items, the goal is to predict which item will be bought at the end of the session. The session information contains the list of items viewed up to and not including the purchased item. Each item is described by a set of categorical attributes (e.g. color or length of sleeves), which can vary between item types. More details on the dataset can be found here. Particularly effective, novel, otherwise interesting contributions to the competition are invited to submit to the ACM RecSys 2022 conference, and prizes can be awarded to such submissions.

3 The team

A project team has to be composed by exactly 3 students registered to the class. Projects submitted by teams composed by a different number of students or by a student not officially registered will not be considered. The team must be registered on the RecSys challenge website (under the tab "My team"), and for organization purpose, the team must also be registered on Université Virtuelle (UV) using the "Group Choice for Project" tool. The teams must be formed at the latest one week after the publication of the project assignment, that is, **no later than the 27th of April 2022**. If you are unable to form a team of three by this date, get in touch with the assistant (theo.verhelst@ulb.be), who will group the remaining students.

4 Tasks

The team will have to:

1. Implement in the Spark language a pipeline for data preprocessing, including missing value imputation, normalization (if required) and feature engineering (at least 20 features). This procedure must return a Spark RDD containing the dataset for further analysis and must be detailed in the notebook. The documentation must contain the list of engineered variables and the motivation of their choice. The use of visualizations and tables to provide a better understanding of the data and the usage of formulas and pseudo-code to describe the feature engineering procedure is strongly encouraged. Note one third of the score will be attributed on the basis of the quality of the documentation.

2. Implement two scalable feature selection algorithms, a ranking algorithm and a forward feature selection. The implementation should not use existing feature selection code and rely only on basic Map/Reduce functionalities of Spark; the use of existing code to implement a learner (e.g. for the sake of cross-validation) is allowed. This step must return, for each feature selection algorithm, a Spark RDD containing the set of the most relevant features. The implementation must be detailed in the notebook.

3. Implement a model to return the prediction required by the competition. Students are free to reuse existing code available in open access Spark libraries. This step should also provide some quantitative assessment of the quality of the set of features returned in the previous step. The learning procedure must be detailed in the notebook. The text should justify the choice of this procedure, assess its accuracy with respect to the one developed in the point 2 and discuss the results. The use of figures, formulas, tables and pseudo-code to describe the combination of this novel procedure is strongly encouraged. Note one third of the score will be attributed on the basis of the quality of the documentation. On the basis of the described procedure, the team must compute the predictions for the competition and submit them via the RecSys website. The name of the team should appear in the official leaderboard of the competition.

4. (Bonus) Assess the scalability (in terms of executors) of the solution. The student must use tables and graphics in the report to illustrate the scalability results and justify the implementation choices.

5 Specifications

Tasks 1 and 2 use only conventional Spark map-reduce functions (the ones introduced in the second lecture).

For the Task 3, the team is free to employ other learning methods, either already available online, or coded. The quality of the classification models during the selection process should be assessed by using classification accuracy.

6 Deliverables

The student team will deliver:

1. The implementations of the tasks presented above in a notebook.

2. A report presenting

   (a) A description of the overall architecture, and why it is scalable
   (b) Experimental results in terms of predictions accuracy
   (c) Experimental results in terms of scalability (bonus).

3. A video of max 10 minutes to present this project. The presentation should address the main points illustrated in the Jupyter notebook. Each of the three main tasks of the project must be presented by a different student of the team.

Rules for project submission

To be read carefully!

1. The assignment should be made by teams of **exactly** three students. The team composition has to be finalized no later than the **27th of April 2022** on the RecSys challenge website and at https://forms.gle/Wtp8NAG2xiqp1tDf6.
2. The assignment will be graded on the implementation,the report and the video presentation.
3. The code should be **commented**.
4. The assignment will be handed in through the dedicated Homework module on the Virtual University.
5. All the deliverables will be put in a single archive `INFOH515_<STUDENT_ID>_<LAST_NAME>.zip` where <STUDENT_ID> and <LAST_NAME> should be replaced by the actual student id and last name of the student submitting the assignment for the group. One submission per group is sufficient. The archive should include:
   - **Python Jupyter Notebook** (`*.ipynb`)
   - **Report** (`*.pdf`)

   N.B. The report should contain a link to the video of presentation. Given the size of the videos and the format of the video, the video needs to be stored on a different platform than the Virtual University (e.g. Microsoft Stream, Youtube). In case of problems, get in touch with the assistant (`theo.verhelst@ulb.be`) to find an alternative solution.
6. Your project should be submitted on the UV no later than **11PM, the 29th of May 2022**.
7. All the projects submitted after the deadline will be
   - Penalized of one point if submitted **before 11PM, the 30th of May 2022**.
   - Not graded (0/10) if submitted **after 11PM, the 30th of May 2022**.
8. Sharing of code is not allowed (you may, however, verbally discuss ideas on how to tackle the project).
9. This project counts for 50% of your grade (10 points). This project **shall be completed as a team and it shall represent your sole efforts**. The result or the copy of another team efforts (current, or past, semester(s)), is considered academic dishonesty. Plagiarism, in the sense of copy-pasting from existing reports or code is a serious issue.
10. Each project producing any error during its execution will receive a grade of 0/10.